ARMOURED: Adversarially Robust Models using Unlabeled data by Regularizing Diversity

Anonymous authors
Paper under double-blind review

Abstract

Adversarial attacks pose a major challenge for modern deep neural networks. Recent advancements show that adversarially robust generalization requires a huge amount of labeled data for training. If annotation becomes a burden, can unlabeled data help bridge the gap? In this paper, we propose ARMOURED, an adversarially robust training method based on semi-supervised learning that consists of two components. The first component applies multi-view learning to simultaneously optimize multiple independent networks and utilizes unlabeled data to enforce labeling consistency. The second component reduces adversarial transferability among the networks via diversity regularizers inspired by determinantal point processes and entropy maximization. Experimental results show that under small perturbation budgets, ARMOURED is robust against strong adaptive adversaries. Notably, ARMOURED does not rely on generating adversarial samples during training. When used in combination with adversarial training, ARMOURED achieves state-of-the-art robustness against $\ell_\infty$ and $\ell_2$ attacks for a range of perturbation budgets, while maintaining high accuracy on clean samples. We demonstrate the robustness of ARMOURED on CIFAR-10 and SVHN datasets against state-of-the-art benchmarks in adversarial robust training.

1 Introduction

Modern deep neural networks have met or even surpassed human-level performance on a variety of image classification tasks. However, they are vulnerable to adversarial attacks, where small, calculated perturbations in the input sample can fool a network into making unintended behavior, e.g., misclassification. Szegedy et al. [2014] Biggio et al. [2013]. Such adversarial attacks have been found to transfer between different network architectures (Papernot et al., 2016) and are a serious concern, especially when neural networks are used in real-world applications. As a result, much work has been done to improve the robustness of neural networks against adversarial attacks (Miller et al. [2020]). Of these techniques, adversarial training (AT) (Goodfellow et al., 2015; Madry et al., 2018) is widely used and has been found to provide the most robust models in a recent evaluation study (Dong et al. [2020]). Nonetheless, even models trained with AT have markedly reduced performance on adversarial samples in comparison to clean samples. Models trained with AT also have worse accuracy on clean samples when compared to models trained with standard classification losses. Schmidt et al. (2018) suggest that one reason for such reductions in model accuracy is that training adversarially robust models requires substantially more labeled data. Due to the high costs of obtaining such labeled data in real-world applications, recent work has explored semi-supervised AT-based approaches that are able to leverage unlabeled data instead (Uesato et al., 2019; Najafi et al., 2019; Zhai et al., 2019; Carmon et al., 2019).

Orthogonal to AT-based approaches that focus on training robust single models, a few works have explored the use of diversity regularization for learning adversarially robust classifiers. These works rely on encouraging ensemble diversity through regularization terms, whether on model predictions (Pang et al., 2019) or model gradients (Dabouei et al., 2020), guided by the intuition that diversity amongst the model ensemble will make it difficult for adversarial attacks to transfer between individual models, thus making the ensemble as a whole more resistant to attack.
In this work, we propose ARMOURED: Adversarially Robust MOdels using Unlabeled data by REgularizing Diversity, a novel algorithm for adversarially robust model learning that elegantly unifies semi-supervised learning and diversity regularization through a multi-view learning framework. ARMOURED applies a pseudo-label filter similar to co-training (Blum & Mitchell, 1998) to enforce consistency of different networks’ predictions on the unlabeled data. In addition, we derive a regularization term inspired by determinantal point processes (DPP) (Kulesza & Taskar, 2012) that encourages the two networks to predict differently for non-target classes. Lastly, ARMOURED maximizes the entropy of the combined multi-view output on the non-target classes. We show in evaluations on CIFAR-10 and SVHN that ARMOURED outperforms existing approaches in defending against strong adaptive adversaries as long as the perturbations are within small $\ell_\infty$ or $\ell_2$ norm-bounded balls. Notably, unlike previous semi-supervised methods, ARMOURED does not use adversarial samples during training. Furthermore, we show that in combination with AT, ARMOURED achieves state-of-the-art robustness on a range of perturbation budgets while offering significantly higher clean accuracy than competing methods.

In summary, the major contributions of this work are as follows:

1. We propose ARMOURED, a novel semi-supervised method based on multi-view learning and diversity regularization for training adversarially robust models.
2. We perform an extensive comparison, including standard semi-supervised learning approaches in addition to methods for learning adversarially robust models.
3. We show that ARMOURED+AT achieves state-of-the-art adversarial robustness while maintaining high accuracy on clean data.

2 RELATED WORK

To set the stage for ARMOURED, in this section, we briefly review adversarially robust learning and semi-supervised learning - two paradigms in the literature that are related to our work.

2.1 ADVERSARIALY ROBUST LEARNING

Adversarial attacks: We consider attacks where adversarial samples stay within a $\ell_p$ ball with fixed radius $\epsilon$ around the clean sample. In this setting, the two standard white-box attacks are the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) that computes a one-step perturbation that maximizes the cross entropy loss function, and Projected Gradient Descent (PGD) (Madry et al., 2018), a stronger attack that performs multiple iterations of gradient updates to maximize the loss; this may be seen as a multi-step version of FGSM. Auto-PGD attack (APGD) (Croce & Hein, 2020) is a parameter-free, budget-aware variant of PGD which aims at better convergence. However, robustness against these gradient-based attacks may give a false sense of security due to gradient-masking. This phenomenon happens when the defense does not produce useful gradients to generate adversarial samples (Athalye et al., 2018). Gradient-masking is known to affect PGD by preventing its convergence to the actual adversarial samples (Tramèr & Boneh, 2019). There exists gradient-based attacks such as Fast Adaptive Boundary attack (Croce & Hein, 2019) (FAB) which is invariant to rescaling, thus is unaffected by gradient-masking. FAB minimizes the perturbation norm as long as misclassification is achieved. Another useful tool is Square attack (Andriushchenko et al., 2020), which is a black-box score-based attack that relies on random search without using any gradient information. Finally, AutoAttack (Croce & Hein, 2020) is a strong ensemble adversary which applies four attacks sequentially (APGD with cross entropy loss, followed by targeted APGD with difference-of-logits-ratio loss, targeted FAB, then Square).

Adversarial training: Adversarial training (AT) is a popular approach that performs well in practice (Dong et al., 2020). Madry et al. (2018) formulate AT as a min-max problem, where the model is trained with adversarial samples found via PGD. Variants of this method such as TRADES (Zhang et al., 2019b) and ALP (Kannan et al., 2018) further decompose the error into natural error and boundary error for higher robustness. Zhang et al. (2019a), Wang et al. (2019) theoretically prove the convergence of AT. Two drawbacks of AT are its slow training due to adversarial example generation requiring multiple gradient computations, and the significant reduction in model accuracy on clean samples. Several recent works have focused on speeding up AT (Zhang et al., 2019a; Qin et al., [2020]).
ARMoured addresses the second limitation, enabling significantly improved performance on clean samples.

Semi-supervised adversarial training: Schmidt et al. (2018) showed that adversarial robust generalization requires much more labeled data. To relieve the annotation burden, several semi-supervised adversarially robust learning (SSAR) methods have been developed to exploit unlabeled data instead. Uesato et al. (2019) introduced unsupervised adversarial training, a simple self-training model which optimizes a smoothness loss and a classification loss using pseudo-labels. Carmon et al. (2019) revisited the Gaussian model by Schmidt et al. (2018) and introduced robust self-training (RST), another self-training model that computes a regularization loss from unlabeled data, either via adversarial training or stability training. Zhai et al. (2019) applied a generalized virtual adversarial training to optimize the prediction stability of their model in the presence of perturbations. Najafi et al. (2019) proposed a semi-supervised extension of the distributionally robust optimization framework by Sinha et al. (2018). They replace pseudo-labels with soft-labels for unlabeled data and train them together with labeled data. It is worth noting that all of these four state-of-the-art SSAR methods apply AT in their training procedure.

Diversity regularization: Diversity regularization is an orthogonal direction to AT that has the potential to further improve the performance of AT. In earlier work, Pang et al. (2018) showed that for a single network, adversarial robustness can be improved when the features learned for different classes are diverse. Pang et al. (2019) further developed this concept by introducing Adaptive Promoting Diversity regularization (ADP). Given an ensemble of neural network classifiers, ADP promotes diversity among non-target predictions of the networks. ADP is inspired by determinantal point processes (Hough et al., 2006), an elegant statistical tool to model repulsive interactions among items of a fixed ground set; applications to machine learning are reviewed in (Kulesza & Taskar, 2012). Dabouei et al. (2020) enforce diversity on the gradients of individual networks in the ensemble instead of their predictions. We note that unlike ARMOURED, the methods described here are developed for the fully-supervised setting, and are not able to utilize unlabeled data.

2.2 Semi-supervised Learning

Semi-supervised learning: Semi-supervised learning (SSL) is an effective strategy to learn from low-cost unlabeled data. There is considerable recent work in this practically relevant and active research area; we will not be able to cover all these works here. Existing SSL methods can be broadly categorized into three groups: consistency-based, graph-based, and generative models. Recent methods, such as Mean Teacher (Tarvainen & Valpola, 2017) and MixMatch (Berthelot et al., 2019), are consistency-based as this approach can be adapted to generic problems and have superior performance in practice. The key idea behind consistency-based methods is that model predictions on different augmentations of the same input should be consistent.

Multi-view learning: Multi-view learning is a SSL paradigm that is capable of representing diversity in addition to consistency. A dataset is considered to have multiple views when its data samples are represented by more than one set of features and each set is sufficient for the learning task. In this setting, a multi-view method assigns one modeling function to each view and jointly optimizes the functions to improve generalization performance (Zhao et al., 2017). By analyzing various multi-view algorithms, Xu et al. (2013) summarized consensus and complementary as the two underpinning principles of multi-view learning. The consensus principle states that a multi-view technique must aim at maximizing the prediction agreement on different views, similar to the consistency-based SSL methods discussed above. The complementary principle states that in order to make improvement, each view must contain some information that the other views do not carry, that the views should be sufficiently diverse. This principle has been applied to boost generalization capability in regular SSL (Qiao et al., 2018) and learning with label noise (Han et al., 2018). In this paper, we argue that multi-view complementarity also plays a critical role in improving adversarial robustness, by reducing the transferability of adversarial attacks across different views.

3 The ARMOURED Method

In this section, we introduce ARMOURED, our proposed semi-supervised adversarially robust learning method. To utilize both labeled and unlabeled data, ARMOURED adopts a multi-view
framework where multiple networks output different predictions (posterior probabilities, which we will refer to as deep views) on the same input image. The networks are then co-optimized by a single loss function computed on the deep views. We adhere to both the consensus and complementary principles of multi-view learning by ensuring that the deep views maximize their consensus on the target class (the ground truth class for labeled examples), but complement each other on the non-target classes. To determine a “target” class for unlabeled samples, ARMoured applies a matching filter to pick out a target class based on agreement between views. Since our method is designed for adversarial robustness, we place a greater emphasis on the complementary principle. More concretely, we introduce two levels of complementarity: (i) among the deep views via a regularizer based on DPP and (ii) among the non-target classes via an entropy regularization applied on the combined multi-view output. Following this, we will describe ARMoured in detail. Pseudocode detailing the training procedure is provided in Appendix A.1.

**Overview:** We describe the general $M$-view model. Consider a semi-supervised image classification task on input image $x$ and target label $y$ from one of $K$ classes, $y \in \{1, 2, \ldots, K\}$. In each minibatch, our training data consists of a labeled set $L = \{(x_i, y_i)\}_{i=1}^{n_l}$ and an unlabeled set $U = \{x_i\}_{i=1}^{n_u}$. For each input image $x$, we apply random augmentations to generate $M$ different augmented images $\{x^m\}_{m=1}^M$. Let $\{N^m\}_{m=1}^M$ be architecturally similar neural networks with respective parameters $\{\theta^m\}_{m=1}^M$. Each network takes the corresponding augmented input and produces predictions $f^m(x) = N^m(x^m, \theta^m) \forall m = 1, \ldots, M$. Due to the different augmentations and network parameters, each output $f^m(x)$ can be treated as one deep view of the original image $x$. Finally, we compute a loss function on these deep views and backpropagate to optimize the parameters

$$L(x, y) = L_{CE}(x, y) + \lambda_{DPP}L_{DPP}(x, y) + \lambda_{NEM}L_{NEM}(x, y) \quad (1)$$

where $\lambda_{DPP}$ and $\lambda_{NEM}$ are model hyperparameters. We describe each component of the overall loss function, $L_{CE}$, $L_{DPP}$ and $L_{NEM}$ in the following. At inference time, $M$ outputs are combined to produce a single prediction. Since our networks possess similar learning capability, the final prediction is computed by averaging the deep views: $f(x) = \frac{1}{M} \sum_{m=1}^M f^m(x)$. Figure 1 illustrates the ARMoured multi-view framework for the dual-view scenario.

**Cross-entropy loss ($L_{CE}$) and pseudo-label filter:** For each labeled sample, we minimize the standard cross-entropy loss $L_{CE}(x, y) = -\sum_{m=1}^M \log f^m(x)$. While one may train each deep view independently using only the labeled data, the fact that augmented inputs are generated from the same original image enables us to add an additional constraint – that the deep views should agree with each other even on unlabeled samples. Hence, when all $M$ networks assign the highest probability to the same class, we can be confident about their prediction on the sample. We denote such sample as a stable sample and define a pseudo-label $\hat{y}$ for it as $\hat{y} = \arg \max_{y \in \{1, 2, \ldots, K\}} f^m \forall m = 1, \ldots, M$. This pseudo-labeling technique has its roots in co-training [Blum & Mitchell [1998]], a multi-view technique that conforms to the consensus principle. After a stable sample is confirmed, it is treated as a labeled sample and the cross-entropy loss $L_{CE}$ applies. We recompute pseudo-labels for each minibatch to avoid making incorrect pseudo-labels permanent.
DPP regularization ($\mathcal{L}_{\text{DPP}}$): Suppose that the number of deep views is smaller than the number of classes, i.e., $(M < K)$. Let $F$ be the $K \times M$ matrix formed by stacking the deep views horizontally, i.e., $F = [f^1, f^2, \ldots, f^M]$. Furthermore, let $S$ be a $K \times K$ positive semidefinite kernel matrix that measures the pairwise similarity among the classes. For each sample, we extract $F_{\bar{y}}$ and $S_{\bar{y}}$ as the submatrices of $F$ and $S$ that correspond to the non-target classes. Let $\hat{F}_{\bar{y}}$ denote the normalized $F_{\bar{y}}$, where each column is scaled to unit length. Inspired by determinantal point processes (Kulesza & Taskar, 2012), ARMOURED minimizes the following loss:

$$\mathcal{L}_{\text{DPP}}(x, y) = -\log \left( \det \left( \hat{F}^\top_{\bar{y}} S_{\bar{y}} \hat{F}_{\bar{y}} \right) \right).$$

This loss is minimized at $\hat{F}_{\bar{y}} = \tilde{F}$, where $\tilde{F}$ is the horizontal concatenation of the first $M$ dominant eigenvectors of $S_{\bar{y}}$; a proof is provided in Appendix A.2. Since eigenvectors are always orthogonal, $\mathcal{L}_{\text{DPP}}$ encourages the deep views to make diverse predictions on non-target classes. If the kernel matrix is predefined, this result allows us to interpret the non-target predictions implied by the DPP regularizer. Specifically, if the kernel $S$ is constructed by a similarity measure over the classes, then a clustering effect will be observed, where similar classes are “preferred” by the same view. On the other hand, we can also inject prior knowledge or encourage desired behavior by designing a custom kernel. Exploitation of prior knowledge can be beneficial to generalization, especially when labeled training data are limited.

We note that our DPP regularizer generalizes the ensemble diversity regularizer of ADP (Pang et al., 2019), that uses the identity matrix as its kernel ($S \equiv I$). If we decompose the kernel matrix such that $S = \Phi^\top \Phi$, then our DPP regularizer is equivalent to the ADP regularizer applied on a linear transformation $\Phi F_{\bar{y}}$ of the non-target predictions. Again, this linear transformation is another way to regulate the deep views, and can either be learned or predefined. Figure 2 illustrates the difference between the predictions from baseline model vs from ARMOURED models with different kernels.

Non-target entropy maximization ($\mathcal{L}_{\text{NEM}}$): Besides the multi-view diversity, we further propose an entropy regularizer that encourages larger margins among non-target classes in the final predictions $f(x)$. Specifically, let $f_{\bar{y}}$ be the $(K - 1) \times 1$ vector of non-target predictions, and $\hat{f}_{\bar{y}}$ be the normalized vector where the elements sum up to 1. We propose to maximize the entropy defined over the normalized non-target predictions. Our entropy regularizer is therefore defined as the negative entropy $\mathcal{L}_{\text{NEM}}(x, y) = -H(\hat{f}_{\bar{y}}) = \sum_{k=1}^{K-1} \hat{f}_{\bar{y}} \log \hat{f}_{\bar{y}}$. This loss is minimized when all elements of $\hat{f}_{\bar{y}}$ are equal to $\frac{1}{K-1}(1 - f_k)$. Intuitively, this regularizer acts as a balancing force on the non-target predictions. It prevents ARMOURED from assigning high probability to any of the incorrect classes. We note that $\mathcal{L}_{\text{NEM}}$ differs from the entropy maximization technique adopted in Pang et al. (2019) that encourages a uniform distribution over all $K$ classes. Although our regularizer is similar to the complement objective proposed by Chen et al. (2019),
we extend this technique to semi-supervised learning and provide more theoretical insight – we show that entropy maximization increases a lower bound on the average (logit) margin under mild assumptions (Theorem A.2 in Appendix A.3).

4 Experiments

4.1 Experimental Setup

Dataset: We evaluate ARMOURED on the CIFAR-10 and SVHN datasets. We use the official train/test splits (50k/10k labeled samples) for CIFAR-10 (Krizhevsky et al. (2009)) and reserved 5k samples from the training samples for a validation set. In our semi-supervised setup, the label budget is either 1k or 4k; remaining samples from training set are treated as unlabeled samples. For the SVHN dataset (Netzer et al. (2011)), our train/validation/test split is 65,932 / 7,325 / 26,032 samples. We use only 1k samples as the label budget in our semi-supervised setup for SVHN. For simplicity, we will refer to our semi-supervised setup with limited label budget as “Dataset-semi-budget”, e.g., CIFAR-10-semi-4k, SVHN-semi-1k.

Adversarial attacks: To evaluate robustness, we apply the following adversaries: (i) Auto-PGD (APGD) attack (Croce & Hein, 2020) with cross entropy loss; (ii) Fast Adaptive Boundary attack (FAB) attack (Croce & Hein, 2019); (iii) Square attack (Andriushchenko et al., 2020); and (iv) AutoAttack (Croce & Hein, 2020). We use the official implementation released by Croce & Hein in our experiments. For $\ell_\infty$ attacks, the range of perturbation budgets is $\epsilon \in \{1/255, 2/255, 4/255, 8/255\}$. For $\ell_2$ attacks, the range is $\epsilon \in \{0.1, 0.2, 0.3, 0.5\}$.

Backbone network and training: To enable fair comparison between different methods, the same Wide ResNet (Oliver et al., 2018) backbone is used for all methods. Specifically, we implement “WRN-28-2” with depth 28 and width 2 along with batch normalization, leaky ReLU activation and Adam optimizer. We train each method with 3 random seeds, for 600 epochs on CIFAR-10 and 380 epochs on SVHN. Learning rate is decayed by a factor of 0.2 after the first 400k iterations.

ARMOURED training: We implement three variants based on the dual-view model shown in Figure 1 that differ only in choice of diversity kernel. ARMOURED-I is our standard model that uses the Identity matrix as its diversity kernel. ARMOURED-H uses a Hand-crafted binary matrix intended to group the classes into two predefined clusters. On CIFAR-10, these are “vehicles” (airplane, ship, truck, automobile) and “animals” (bird, cat, deer, dog, frog, horse). On SVHN, we split the digits into “edgy” (1, 4, 7) and “curvy” (0, 2, 3, 5, 6, 8, 9). The third variant is ARMOURED-F, which uses a Feature-based kernel. From a pre-trained SSL model, we extract for each class the average feature vector over labeled training samples; feature vectors are extracted right before the first linear layer. Combining vectors from all classes, we obtain a feature matrix $B$ and compute the feature-based kernel as $S = B^\top B$. For our ablation study and other analyses, we also train a Baseline model without $L_{DPP}$ and $L_{NEM}$, ARMOURED-B.

ARMOURED+AT training: We notice that multi-view diversity is an orthogonal approach to AT, and that successful defenses against large-perturbation attacks always rely on AT (Croce & Hein, 2020). Therefore, we hope to combine the best of both worlds by implementing AT as a wrapper method for ARMOURED. The ARMOURED+AT training procedure consists of three steps. First, for each batch of semi-supervised data, we apply the inference procedure of ARMOURED to generate predictions for unlabeled data, which are then used to generate pseudo-labels. Second, for each input sample in the batch, we compute one adversarial sample using either the true label (if the sample is labeled) or the pseudo-label (if the sample is unlabeled). This step applies a 7-step PGD $\ell_\infty$ attack with total $\epsilon = 8/255$ and step size $\epsilon/4$. Third, we execute the training procedure of ARMOURED on the adversarial samples and the original labels. The pseudo-labels computed from the first step are now dropped, so that the training data are still semi-supervised.

Comparison benchmarks: We test ARMOURED against a wide range of benchmarks: ADP (Pang et al., 2019), Mean Teacher (MT) (Tarvainen & Valpola, 2017), MixMatch (Berthelot et al., 2019), the method of Zhai et al. (2019) that we denote as ARG, RST (Carmon et al., 2019) (RST has two variants, we implemented RST-std). Among them, ADP is a supervised AR method, MT and MixMatch are SSL methods, while ARG and RST are state-of-the-art SSAR methods. In addition, we combine MT and MixMatch with adversarial training (AT) (Madry et al., 2018) in an
attempt to boost their adversarial robustness. Since AT requires label information, the class with the highest predicted posterior on an unlabeled sample is used as its pseudo-label. For the methods that apply AT (including MT+AT, MixMatch+AT, ARG and RST), we use a 7-step PGD $\ell_\infty$ attack to generate adversarial samples for training with total $\epsilon = 8/255$ and step size $\epsilon/4$.

4.2 RESULTS

Each result contains mean and standard deviation statistics computed from three independent runs with different random data seeds used for selecting labeled samples.

**Range of perturbation budgets (Figure 3) Table B.6** We plot the robustness against AutoAttack of five competing methods: MT+AT, ARG, RST, ARMOURED-F and ARMOURED-F+AT when the perturbation budget $\epsilon$ is gradually increased. ARMOURED-F obtains high accuracies on clean data as well as for small perturbation budgets. For $\ell_\infty$ attacks with $\epsilon = 1/255$ or $\ell_2$ attacks with $\epsilon = 0.1$, ARMOURED-F achieves 64.70% and 65.07% robust accuracies respectively, this is the highest performance among all competing methods. From Figure 3 we can also see that ARMOURED-F+AT consistently yields high performance, comparable to other SSAR methods over a broad range of $\epsilon$.

**Results on CIFAR-10 (Table 1)**: Table 1 shows numerical results on CIFAR-10-semi-4k, the full table is in the Supplement. ARMOURED-F surpasses all other methods on clean accuracy by a large margin (15%-20%). Furthermore, ARMOURED-F is very robust against FAB and Square attacks, which suggests that gradient-masking is less likely to exist in the final model. APGD $\ell_\infty$ is more successful on ARMOURED-F than on ARG and RST (by 7%), while the numbers from APGD $\ell_2$ do not show big differences. Interestingly, AutoAttack can impact ARMOURED-F heavily, which may be due to its ensemble style of applying sequential attacks. On the other hand, the ARMOURED-F+AT model approaches state-of-the-art performance against AutoAttack, comparable to RST and slightly worse than ARG (by 2%-5%). It outperforms ARG and RST when evaluated against individual attacks: APGD (by 4%-5%), FAB (by 25%-40%) and Square (by 15%-40%). In spite of being trained on adversarial samples, ARMOURED-F+AT still maintains a relatively higher clean accuracy than ARG (by 8%) and RST (by 17%). These results suggest that ARMOURED-F+AT is able to combine the advantages of both techniques.

**Results on SVHN**: To be completed.
We choose the ARMOURED-F model trained on CIFAR-10-semi-4k as the “complete” model. For component to the performance of ARMOURED-F against AutoAttack with small perturbation budgets. complete model, especially under larization terms are removed from the total loss. This model consistently performs worse than the worse than the complete model (by 1%). With ARMOURED-B, both diversity and entropy regu-

tropy of the averaged prediction f

ARMOURED-F+AT is more robust than ARMOURED-F under larger-perturbation attacks, but is model shows the importance of unlabeled data towards improving adversarial robustness. Finally, trained without unlabeled data. The huge performance gap between this model and the complete

demonstrated the high performance of ARMOURED against strong adaptive attacks under small

perturbation budgets. Remarkably, ARMOURED improves clean accuracy by a signifi-

cant margin when compared with existing SSAR methods. The strong empirical performance of ARMOURED+AT suggests that it is possible to learn adversarially robust models even in the low-
labeled data regime while upholding a reasonable accuracy on clean samples. Extending this method to exploit more than two views or alternative custom kernels for the DPP regularizer could result in further performance gains.

5 CONCLUSION

In this work, we presented ARMOURED, a novel method for learning adversarially robust models that unifies semi-supervised learning and diversity regularization in a multi-view framework. We demonstrated the high performance of ARMOURED against strong adaptive attacks under small perturbation budgets. When combined with AT, ARMOURED shows robustness against a wider range of perturbation budgets. Remarkably, ARMOURED improves clean accuracy by a significant margin when compared with existing SSAR methods. The strong empirical performance of ARMOURED+AT suggests that it is possible to learn adversarially robust models even in the low-labeled data regime while upholding a reasonable accuracy on clean samples. Extending this method to exploit more than two views or alternative custom kernels for the DPP regularizer could result in further performance gains.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square at-


Ablation study (Table[2]: We perform an ablation study to investigate the contribution of each component to the performance of ARMOURED-F against AutoAttack with small perturbation budgets. We choose the ARMOURED-F model trained on CIFAR-10-semi-4k as the “complete” model. For w/o L_{DPP}, we remove the diversity regularizer from equation (1). This model performs marginally worse than the complete model (by 1%). With ARMOURED-B, both diversity and entropy regularization terms are removed from the total loss. This model consistently performs worse than the complete model, especially under \ell_2 attacks. For w/ H(f), we replace H(f_{\mu}) in L_{\text{NEM}} by the entropy of the averaged prediction f over all classes, similar to the term used by Pang et al. (2019). This model achieves comparable results to ARMOURED-F. The model w/o U IS ARMOURED-F trained without unlabeled data. The huge performance gap between this model and the complete model shows the importance of unlabeled data towards improving adversarial robustness. Finally, ARMOURED-F+AT is more robust than ARMOURED-F under larger-perturbation attacks, but is less robust for small perturbations.

Table 1: Benchmark Results against AutoAttack on CIFAR-10-semi-4k

<table>
<thead>
<tr>
<th>Attack (\epsilon)</th>
<th>MT+AT</th>
<th>ARG</th>
<th>RST</th>
<th>ARMOURED-F</th>
<th>ARMOURED-F+AT</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Attack</td>
<td>64.77 ± 0.28</td>
<td>67.07 ± 0.22</td>
<td>58.82 ± 0.28</td>
<td>81.09 ± 0.69</td>
<td>75.27</td>
</tr>
<tr>
<td>\ell_{\infty}</td>
<td>APGD(8/255)</td>
<td>24.66 ± 3.07</td>
<td>30.97 ± 1.80</td>
<td>30.65 ± 0.35</td>
<td>23.18 ± 1.72</td>
</tr>
<tr>
<td>FAB(8/255)</td>
<td>24.86 ± 2.94</td>
<td>33.92 ± 4.83</td>
<td>25.83 ± 0.18</td>
<td>68.34 ± 1.38</td>
<td>66.83</td>
</tr>
<tr>
<td>Square(8/255)</td>
<td>29.76 ± 3.00</td>
<td>35.41 ± 1.55</td>
<td>28.98 ± 0.17</td>
<td>72.18 ± 0.82</td>
<td>68.19</td>
</tr>
<tr>
<td>AutoAttack(8/255)</td>
<td>24.12 ± 3.15</td>
<td>30.01 ± 1.85</td>
<td>25.37 ± 0.16</td>
<td>9.25 ± 1.32</td>
<td>24.69 ± 4.34</td>
</tr>
<tr>
<td>\ell_2</td>
<td>APGD(0.5)</td>
<td>40.47 ± 1.71</td>
<td>41.73 ± 0.67</td>
<td>43.07 ± 0.38</td>
<td>40.21 ± 2.38</td>
</tr>
<tr>
<td>FAB(0.5)</td>
<td>40.23 ± 1.65</td>
<td>41.26 ± 0.67</td>
<td>39.97 ± 0.56</td>
<td>70.23 ± 1.11</td>
<td>67.45</td>
</tr>
<tr>
<td>Square(0.5)</td>
<td>51.8 ± 1.19</td>
<td>54.26 ± 0.56</td>
<td>48.78 ± 0.28</td>
<td>76.31 ± 0.78</td>
<td>70.38</td>
</tr>
<tr>
<td>AutoAttack(0.5)</td>
<td>39.42 ± 2.49</td>
<td>40.87 ± 0.66</td>
<td>39.79 ± 0.59</td>
<td>25.54 ± 2.10</td>
<td>39.55 ± 3.93</td>
</tr>
</tbody>
</table>

Table 2: Ablation Study Results of ARMOURED-F against AutoAttack.

<table>
<thead>
<tr>
<th>Norm Budget \epsilon</th>
<th>ARMOURED-F w/o L_{DPP}</th>
<th>ARMOURED-B w/ H(f)</th>
<th>w/o U</th>
<th>ARMOURED-F+AT</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ell_{\infty} 1/255</td>
<td>65.23</td>
<td>64.13</td>
<td>62.25</td>
<td>65.51</td>
</tr>
<tr>
<td>8/255</td>
<td>8.66</td>
<td>7.69</td>
<td>6.10</td>
<td>9.83</td>
</tr>
<tr>
<td>\ell_2</td>
<td>0.1</td>
<td>65.58</td>
<td>65.32</td>
<td>15.99</td>
</tr>
</tbody>
</table>


Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.


APPENDICES

A ARMOURED ALGORITHM

A.1 Detailed Training Pseudocode

**Algorithm 1: ARMOURED Minibatch Training**

**Input:** A minibatch of labeled samples $\mathbf{L} = \{(x_i, y_i)\}_{i=1}^{n_L}$ and unlabeled samples $\mathbf{U} = \{x_i\}_{i=1}^{n_U}$, kernel matrix $S$; random augmentation $\eta(x)$; hyperparameters $\lambda_{\text{DPP}}$, $\lambda_{\text{NEM}}$

**Output:** Networks $\{N^m\}_{m=1}^M$ with updated parameters $\{\theta^m\}_{m=1}^M$

for $i = 1, \ldots, n_U$ do
  for $m = 1, \ldots, M$ do
    $x_i^m = \eta(x_i)$
    $f^m(x_i) = N^m(x_i, \theta^m)$
  end
  if $y_i = \arg\max_{k=1,\ldots,K} f_k^m(x_i) \forall m = 1, \ldots, M$ then
    add $(x_i, y_i)$ to $\mathbf{L}$
    remove $x_i$ from $\mathbf{U}$
  end
end

for $i = 1, \ldots, n_L$ do
  for $m = 1, \ldots, M$ do
    $x_i^m = \eta(x_i)$
    $f^m(x_i) = N^m(x_i, \theta^m)$
  end
  compute $L(x_i, y_i) = L_{\text{CE}}(x_i, y_i) + \lambda_{\text{DPP}}L_{\text{DPP}}(x_i, y_i) + \lambda_{\text{NEM}}L_{\text{NEM}}(x_i, y_i)$
end
compute $L = \sum_{i=1}^{n_L} L(x_i, y_i)$
backpropagate $L$ to optimize $\{\theta^m\}_{m=1}^M$

A.2 Optima of DPP Regularizer

For simplicity, we find the maximum of the negative loss $L_{\text{DPP}}(x, y)$, which is defined as

$$Q(x, y) = \det\left(\hat{F}_y^T S_{\backslash y} \hat{F}_y\right)$$

Since $S_{\backslash y}$ is a principal submatrix of $S$, it is also positive semidefinite. We can decompose $S_{\backslash y}$ as follows: $S_{\backslash y} = VDV^T$, where $V$ is a square matrix whose $k$-th column is the eigenvector $v_k$ of $S_{\backslash y}$, and $D$ is a diagonal matrix whose $(k, k)$-th element $\lambda_k$ is the $k$-th largest eigenvalue of $S_{\backslash y}$.

The gradient of $Q$ with respect to $F_{\backslash y}$ is given by Petersen & Pedersen (2012) as

$$\frac{\partial Q}{\partial F_{\backslash y}} = 2\det(\hat{F}_y^T S_{\backslash y} \hat{F}_y)S_{\backslash y} \hat{F}_y(\hat{F}_y^T S_{\backslash y} \hat{F}_y)^{-1}$$

Let $\hat{F}_*$ be the horizontal concatenation of the first $M$ eigenvectors, i.e., $\hat{F}_* = [v_1, v_2, \ldots, v_M]$. Notice that $\hat{F}_y^T S_{\backslash y} \hat{F}_* = D_M$, where $D_M$ is the $M \times M$ leading principal submatrix of $D$. We evaluate the gradient at $\hat{F}_*$ as follows

$$\left.\frac{\partial Q}{\partial F_{\backslash y}}\right|_{\hat{F}_*} = 2\det(\hat{F}_*^T S_{\backslash y} \hat{F}_*) S_{\backslash y} \hat{F}_*(\hat{F}_*^T S_{\backslash y} \hat{F}_*)^{-1}$$

$$= 2\det(D_M) S_{\backslash y} \hat{F}_* D_M^{-1}$$

$$= 2\det(D_M) D_M \hat{F}_* D_M^{-1}$$

$$= 2\det(D_M) \hat{F}_*$$
Interestingly, since $D_M$ is a diagonal matrix, $\det(D_M)$ equals the product of the first $M$ eigenvalues of $S_{\bar{y}}$. This product is also nonnegative because $S_{\bar{y}}$ is positive semidefinite. Therefore, the gradient at $\tilde{F}_s$ is a nonnegative scaling of $\tilde{F}_s$ itself. Since $\tilde{F}_s$ is normalized to unit length, adding this gradient does not update it any further, i.e., the angular gradient at $\tilde{F}_s$ is zero. As shown by Cover & Thomas (1988), given a fixed positive semidefinite kernel, the determinant in equation (2) is a concave function of $\tilde{F}_s$. Thus, $\tilde{F}_s$ is a maximum of $Q$.

Note that $\tilde{F}_s$ is not the only maximum. Let $R$ be a $M \times M$ orthogonal matrix, so that $\tilde{F}_s R$ is a rotation of $\tilde{F}_s$. Then, $\tilde{F}_s R$ is also a maximum of $Q$, because

$$(\tilde{F}_s R)^\top S_{\bar{y}} (\tilde{F}_s R) = R^\top (\tilde{F}_s^\top S_{\bar{y}} \tilde{F}_s) R = R^\top D_M R = D_M = \tilde{F}_s^\top S_{\bar{y}} \tilde{F}_s \quad (8)$$

This means that a family of maxima exists for $Q$, which includes $\tilde{F}_s$ and its orthogonal transformations in the $M$-dimensional subspace spanned by $\tilde{F}_s$.

For example, when $M = 2$, objective $Q$ is maximized at $\tilde{F}_s = [v_1, v_2]$

$$\det \left( \begin{bmatrix} v_1^\top \\ v_2^\top \end{bmatrix} S_{\bar{y}} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \right) = \det \left( \begin{bmatrix} v_1^\top S_{\bar{y}} v_1 & v_1^\top S_{\bar{y}} v_2 \\ v_2^\top S_{\bar{y}} v_1 & v_2^\top S_{\bar{y}} v_2 \end{bmatrix} \right) = \det \left( \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \right) = \lambda_1 \lambda_2 \quad (9)$$

Any rotation of $(v_1, v_2)$ in the 2-dimensional plane spanned by them is also a maximum.

### A.3 Analysis of Non-Target Entropy Maximization

For the ease of expression, we denote $g(x)$ as the unnormalized logits of $f(x)$, the Lipschitz constant $L_N$ as the scalar satisfying,

$$||g(x) - g(x + \epsilon)||_2 \leq L_N ||\epsilon||_2 \quad (10)$$

The guarded adversarial area (Tsuzuku et al. (2018)) is defined as the hypersphere satisfying the following condition, where $\epsilon$ is the largest perturbation radius measured in $\ell_p$ distance

$$\forall \epsilon : ||\epsilon||_p \leq c \Rightarrow f_y(x + \epsilon) \geq \max f_{\bar{y}}(x + \epsilon) \quad (11)$$

The max/average logit gap is the gap between target class logit and maximal/average non target class logit,

$$\text{maxgap}(x) = g_y(x) - \max_{k \neq y} g_k(x), \quad \text{avggap}(x) = g_y(x) - \text{avg}_{k \neq y} g_k(x) \quad (12)$$

We start by introducing the following lemma which is related to Proposition 1 of Tsuzuku et al. (2018).

**Lemma A.1** For any adversarial perturbation $\epsilon$ smaller than the logit gap divided by the Lipschitz constant, it is guaranteed the class prediction does not change.

**Proof.** Lemma A.1 can be written as the following,

$$\text{maxgap}(x) = g_y(x) - \max_{k \neq y} g_k(x) \geq \sqrt{2L_N} ||\epsilon|| \Rightarrow g_y(x + \epsilon) - \max_{k \neq y} \{g_k(x + \epsilon)\} \geq 0 \quad (13)$$

A proof of Lemma A.1 is the same with the proof for Proposition 1 of Tsuzuku et al. (2018).

This lemma suggests that it is possible to increase the robustness, the guarded adversarial area $\epsilon$, by either decreasing the Lipschitz constant and/or increasing the logit gap. It is often acknowledged, as with the analysis in Tsuzuku et al. (2018), that the Lipschitz constant for large neural network is very hard to quantify. Instead we find it is easier to enlarge the logit gap by non-target entropy maximization and reveal a relation between them as follows,

**Theorem A.2** The non-target entropy $H(\tilde{f}_{\bar{y}})$ is a lower bound of average logit gap plus a constant.

The entropy maximization term will encourage a uniform distribution over non-target classes, i.e. $\text{maxgap}(x) \approx \text{avggap}(x)$. By referring to Lemma A.1, this theorem suggests maximizing non-target entropy $H(\tilde{f}_{\bar{y}})$ leads to higher guarded adversarial attack area $\epsilon$. As result, the overall robustness to adversarial attack is improved by introducing the additional non-target entropy maximization loss.
Proof. We first write the theorem to prove as the following:

\[ H(\tilde{f}_y) \leq g_y(x) - \operatorname{avg}_{k \neq y} \{ g_k(x) \} + C \quad (14) \]

Before we provide the proof, we introduce the following two lemmas and make a mild assumption:

**Lemma A.3** LogSumExp is a smooth approximation to and upper bounded by the maximum function plus a constant.

\[ \log \sum_{k \neq y} \exp g_k \leq \max_{k \neq y} g_k + \log(K - 1) \quad (15) \]

**Proof.** We relax the summation with maximization and arrive at the following inequalities.

\[ \log \sum_{k \neq y} \exp g_k \leq \log((K - 1) \exp(\max_{k \neq y} g_k)) \]

\[ = \max_{k \neq y} g_k + \log(K - 1) \]

**Lemma A.4** The following inequality holds for real number vector \( g \) of length \( K \).

\[ \operatorname{avg}_k g_k \leq \sum_k \frac{g_k \exp g_k}{\sum_k \exp g_k} \quad (17) \]

**Proof.** W.l.o.g. we assume \( g_k \) is in descending order, i.e. \( \forall i < j, \quad g_i \geq g_j \). The proof is rewritten as,

\[ g_1(\exp g_1 + \cdots + \exp g_K) + \cdots + g_K(\exp g_1 + \cdots + \exp g_K) \leq Kg_1 \exp g_1 + \cdots + Kg_K \exp g_K \quad (18) \]

The difference between RHS and LHS is written as,

\[ \text{RHS} - \text{LHS} = (\exp g_1 - \exp g_2)(g_1 - g_2) + (\exp g_1 - \exp g_3)(g_1 - g_3) + \cdots + (\exp g_1 - \exp g_K)(g_1 - g_K) + \]

\[ (\exp g_2 - \exp g_3)(g_2 - g_3) + (\exp g_2 - \exp g_4)(g_2 - g_4) + \cdots + (\exp g_2 - \exp g_N)(g_2 - g_N) + \]

\[ \cdots + (\exp g_{K-1} - \exp g_K)(g_{K-1} - g_K) \tilde{F}_s \quad (23) \]

Obviously, \( \text{RHS} - \text{LHS} \) is non-negative, thus the inequality holds.

**Assumption A.5** Assume the clean samples are mostly correctly classified.

\[ \max_{k \neq y} g_k(x) \leq g_y(x) \quad (24) \]

Given the fact that we can achieve relatively high classification accuracy on clean samples, the assumption is realistic in most cases.

Now we prove the inequality for equation \( (14) \) holds.

\[ H(\tilde{f}_y) = - \sum_{k \neq y} \exp g_k \sum_{k \neq y} \exp g_k \log \sum_{k \neq y} \exp g_k \quad (25) \]

\[ = \sum_{k \neq y} \exp g_k \sum_{k \neq y} \exp g_k (\log \sum_{k \neq y} \exp g_k - g_k) \quad (26) \]

\[ \leq \sum_{k \neq y} \exp g_k \sum_{k \neq y} \exp g_k (\max_{k \neq y} g_k + \log(K - 1) - g_k) \quad (27) \]

\[ \leq \sum_{k \neq y} \exp g_k \sum_{k \neq y} \exp g_k (g_y + \log(K - 1) - g_k) \quad (28) \]

\[ \leq \sum_{k \neq y} \exp g_k \sum_{k \neq y} \exp g_k (g_y + \log(K - 1)) - \operatorname{avg}_{k \neq y} g_k \quad (29) \]

\[ \leq (g_y - \operatorname{avg}_{k \neq y} g_k + \log(K - 1)) \tilde{F}_s \quad (30) \]
A.4 Hyperparameters

We fine tune $\lambda_{DPP}$ and $\lambda_{NEM}$ with ARMOURED-I model trained on CIFAR-10-semi-4k. The tuning range are as follows: $\lambda_{DPP} \in [0.5, 1, 2, 5, 10]$ and $\lambda_{NEM} \in [0.5, 1, 1.5, 2, 2.5, 3, 5, 10]$. After tuning, we decided to apply $\lambda_{DPP} = 2$ and $\lambda_{NEM} = 0.5$ universally for all ARMOURED variants. For random augmentations $\eta(x)$, we applied translations and horizontal flips on CIFAR-10 images. For SVHN, we only applied random translations.

B Supplementary Results: Evaluation against Standard Attacks

Adversarial attacks: We apply three different standard attacks: (i) FGSM $\ell_\infty$ attack (Goodfellow et al. (2015)) with $\epsilon = 8/255$, (ii) PGD $\ell_\infty$ attack (Madry et al. (2018)) with $\epsilon = 8/255$ and (iii) PGD $\ell_2$ attack with $\epsilon = 0.5$. We perform 20-step PGD with a step size of $\epsilon/8$.

Results on CIFAR-10 (Table B.3): ARMOURED has performance comparable to MT when evaluated on clean samples (60%-70% on 1k, 80%-85% on 4k), while AR methods perform significantly worse (45%-55% on 1k, 60%-75% on 4k). We also observe significant drops (by 15%-20%) in clean accuracy when AT is applied to SSL methods; this is because AT objective is optimized on adversarial samples. Under FGSM $\ell_\infty$ attacks, AR methods start to outperform SSL methods, but the difference is small. MixMatch does exceptionally well against FGSM in the 4k setup; we suspect this is because FGSM is rather weak with only a 1-step gradient update. Under PGD attacks, SSL methods are heavily impacted, while SSAR methods are able to achieve better robustness against PGD attacks. Overall, ARMOURED variants are more robust than all other methods under PGD attacks, surpassing the second-best competitors by large margins (5%-25% on 1k, 20%-25% on 4k).

Table B.3: Benchmark Results on CIFAR-10

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR-10-semi-1k</th>
<th>CIFAR-10-semi-4k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Clean</td>
<td>FGSM</td>
</tr>
<tr>
<td>ADP</td>
<td>57.75 ± 1.49</td>
<td>13.20 ± 1.57</td>
</tr>
<tr>
<td>MT</td>
<td>65.12 ± 2.69</td>
<td>4.39 ± 0.16</td>
</tr>
<tr>
<td>MixMatch</td>
<td>75.70 ± 1.35</td>
<td>16.99 ± 1.96</td>
</tr>
<tr>
<td>MixMatch+AT</td>
<td>50.80 ± 0.37</td>
<td>20.39 ± 2.01</td>
</tr>
<tr>
<td>MixMatch+AT</td>
<td>66.03 ± 1.12</td>
<td>18.17 ± 1.65</td>
</tr>
<tr>
<td>ARG</td>
<td>51.20 ± 0.67</td>
<td>23.75 ± 0.55</td>
</tr>
<tr>
<td>RST</td>
<td>45.74 ± 2.07</td>
<td>21.31 ± 0.74</td>
</tr>
<tr>
<td>ARMOURED-I</td>
<td>67.42 ± 2.96</td>
<td>53.67 ± 0.45</td>
</tr>
<tr>
<td>ARMOURED-HI</td>
<td>64.11 ± 3.40</td>
<td>51.79 ± 1.44</td>
</tr>
<tr>
<td>ARMOURED-F</td>
<td>60.73 ± 2.29</td>
<td>40.60 ± 1.03</td>
</tr>
</tbody>
</table>

Table B.4: Benchmark Results on SVHN

<table>
<thead>
<tr>
<th>Method</th>
<th>SVHN-semi-1k</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Clean</td>
<td>FGSM</td>
<td>$\ell_\infty$</td>
</tr>
<tr>
<td>ADP</td>
<td>79.77 ± 1.08</td>
<td>11.12 ± 3.13</td>
<td>0.51 ± 0.26</td>
</tr>
<tr>
<td>MT</td>
<td>93.73 ± 0.96</td>
<td>24.42 ± 2.14</td>
<td>4.16 ± 0.82</td>
</tr>
<tr>
<td>MixMatch</td>
<td>91.21 ± 1.25</td>
<td>48.73 ± 15.06</td>
<td>0.06 ± 0.08</td>
</tr>
<tr>
<td>MixMatch+AT</td>
<td>84.39 ± 0.77</td>
<td>49.70 ± 5.50</td>
<td>41.28 ± 5.20</td>
</tr>
<tr>
<td>MixMatch+AT</td>
<td>83.19 ± 0.74</td>
<td>45.19 ± 1.09</td>
<td>36.99 ± 1.05</td>
</tr>
<tr>
<td>ARG</td>
<td>83.31 ± 4.12</td>
<td>53.72 ± 14.33</td>
<td>48.05 ± 22.19</td>
</tr>
<tr>
<td>RST</td>
<td>64.56 ± 3.75</td>
<td>27.87 ± 2.96</td>
<td>24.38 ± 2.46</td>
</tr>
<tr>
<td>ARMOURED-I</td>
<td>94.96 ± 0.16</td>
<td>78.68 ± 0.9</td>
<td>73.34 ± 2.60</td>
</tr>
<tr>
<td>ARMOURED-H</td>
<td>92.50 ± 0.60</td>
<td>60.66 ± 2.80</td>
<td>57.58 ± 4.33</td>
</tr>
<tr>
<td>ARMOURED-F</td>
<td>93.87 ± 0.77</td>
<td>76.67 ± 2.30</td>
<td>72.49 ± 3.83</td>
</tr>
</tbody>
</table>

Results on SVHN (Table B.4): We observe similar performance trends on SVHN as on CIFAR-10. One difference is that for SVHN, prior information no longer helps with generalization; we suspect that the 1k label budget is more than sufficient for this learning task. To summarize Table B.3 and Table B.4, both AT and SSL contribute to improved generalization. However, AT-based methods sacrifice clean accuracy, while SSL methods are easily attacked. Our proposed method outperforms existing methods by a large margin on robustness accuracy without compromising on clean accuracy.
Utilization of unlabeled data: To ensure unlabeled data are fully utilized, we compute the utilization rate, which is defined as the ratio between the number of stable samples and the total number of unlabeled samples in each minibatch. In the last 1000 iterations of training, the average utilization rates on CIFAR-10-semi-4k are 96.24%, 96.80% and 96.00% and on SVHN-semi-1k are 98.80%, 98.24% and 98.32% with ARMOURED-I, ARMOURED-H and ARMOURED-F respectively.

Regularization effect of DPP kernel (Figure B.2): We train semi-supervised models ARMOURED-B, ARMOURED-I and ARMOURED-H on CIFAR-10-semi-1k and plot the networks’ predictions on the test set. The results show how non-target predictions form interesting patterns based on the choice of kernel. Further results are provided in Appendix B.

Visualization of learned representations (Figure B.4): On CIFAR-10 test samples, we visualize the embeddings (representations) learned by three models: ARMOURED-B-Sup (similar to ARMOURED-B but without using unlabeled data for training), ADP and ARMOURED-F. Here, embeddings refer to the features extracted from the last layer of WRN-28-2 before the linear layer. It can be seen that ARMOURED-F produces better feature distributions with cleaner and more separable classes. Even under strong PGD attacks, the classes remain recognizable, which is not the case for the other methods shown (compare column 1 with column 3, and column 2 with column 4).

Ablation study (Table B.5): We perform an ablation study to investigate the contribution of each component to our method’s performance. We choose the ARMOURED-F model trained on CIFAR-10-semi-4k as the “complete” model. For w/o $\ell_{DPP}$, we remove the diversity regularizer from equation (1). This model performs marginally worse than the complete model (by 1%-2% under PGD attacks). With ARMOURED-B, both diversity and entropy regularization terms are removed from the total loss. This model consistently performs worse than the complete model for all $\epsilon$ values. For w/ $H(f)$, we replace $H(f_n)$ in $\ell_{NEM}$ by the entropy of the averaged prediction $f$ over all classes, similar to the term used by Pang et al. [2019]. This model is worse than ARMOURED-F for large $\epsilon$ (by about 1%). Finally, w/o $U$ is ARMOURED-F trained without unlabeled data. The huge performance gap between this model and the complete model shows the importance of unlabeled data towards improving adversarial robustness.

Custom kernels: The ARMOURED hand-crafted kernels and feature-based kernels are shown in Figure B.5 and Figure B.6 respectively. To keep different kernels in the same scale, we normalize each kernel by its largest eigenvalue. For feature kernel, we apply $\ell_2$ normalization to each individual feature vector in the feature matrix $B$ before computing the inner product. From Figure B.6a and Figure B.6b we can see a strong correlation within the groups of animals and vehicles. For SVHN, it can be seen from Figure B.6c that the digits 0, 5, 6, 8, 9 are more related, which is in alignment with our hand-crafted kernel in Figure B.5b.

Regularization effect of DPP kernel: Here, we report the full set of results for this experiment. In Figure B.7, we illustrate the average prediction of test samples generated by ARMOURED variants. From all the three plots, we can clearly see a high and low posterior value preference for each class between different networks. For example, in Figure B.7b, the left network tends to have high predictions for airplane, automobile, ship and truck, while the other network has higher predictions on the remaining six classes. This behaviour is promoted by the kernel. With the feature-based kernel, we observe a similar grouping of classes: the left network prefers bird, cat, deer, dog and horse; while the right network prefers airplane, automobile, frog, ship and truck. With the identity

<table>
<thead>
<tr>
<th>Attack</th>
<th>Budget $\epsilon$</th>
<th>ARMOURED-F w/o $\ell_{DPP}$</th>
<th>ARMOURED-B</th>
<th>w/ $H(f)$</th>
<th>w/o $U$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGD $\ell_{\infty}$</td>
<td>2/255</td>
<td>71.53 ± 0.87</td>
<td>69.89 ± 0.94</td>
<td>52.19 ± 0.67</td>
<td>71.87 ± 1.21</td>
</tr>
<tr>
<td></td>
<td>4/255</td>
<td>65.54 ± 1.71</td>
<td>63.97 ± 1.15</td>
<td>29.89 ± 0.42</td>
<td>66.11 ± 1.86</td>
</tr>
<tr>
<td></td>
<td>8/255</td>
<td>56.70 ± 2.60</td>
<td>54.38 ± 1.21</td>
<td>15.68 ± 0.66</td>
<td>56.65 ± 2.17</td>
</tr>
<tr>
<td></td>
<td>16/255</td>
<td>40.97 ± 3.62</td>
<td>38.61 ± 1.97</td>
<td>9.24 ± 0.50</td>
<td>40.33 ± 2.63</td>
</tr>
<tr>
<td>PGD $\ell_2$</td>
<td>0.25</td>
<td>70.47 ± 1.00</td>
<td>69.18 ± 0.93</td>
<td>16.07 ± 0.27</td>
<td>71.07 ± 1.09</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>64.52 ± 1.74</td>
<td>62.63 ± 1.11</td>
<td>12.44 ± 0.38</td>
<td>65.15 ± 1.54</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>55.34 ± 2.52</td>
<td>53.19 ± 1.17</td>
<td>13.13 ± 0.25</td>
<td>55.81 ± 2.39</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>40.20 ± 3.54</td>
<td>37.09 ± 1.97</td>
<td>13.47 ± 0.29</td>
<td>39.39 ± 2.65</td>
</tr>
</tbody>
</table>
Figure B.4: t-SNE plots of feature embeddings from CIFAR-10 test samples generated by the following models: (a) ARMOURED-B-Sup and (b) ADP are trained only with 4k labeled samples, while (c) ARMOURED-F is trained on CIFAR-10-semi-4k. (d) ARMOURED-F+AT applies adversarial training to ARMOURED-F. For each of the 8 network/method pairs, the clean and adversarial samples are processed together in a single t-SNE run. Adversarial samples are generated with PGD-$\ell_\infty$ ($\epsilon = 8/255$). We observe gradual improvements from no diversity regularization to supervised ADP diversity (better) and semi-supervised ARMOURED diversity (best).

Figure B.5: Hand-crafted kernels.

matrix as kernel, the predictions in Figure B.7a also form two groups, but the correlation among classes of the same group are less intuitive.
Table B.6: (Full) Benchmark Results against AutoAttack on CIFAR-10-semi-4k

<table>
<thead>
<tr>
<th>Attack (ε)</th>
<th>MT+AT</th>
<th>ARG</th>
<th>RST</th>
<th>ARMoured-F</th>
<th>ARMoured-F+AT</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Attack</td>
<td>64.77 ± 0.28</td>
<td>67.07 ± 0.22</td>
<td>58.82 ± 0.28</td>
<td>81.09 ± 0.69</td>
<td>75.27</td>
</tr>
<tr>
<td>APGD(8/255)</td>
<td>24.66 ± 3.07</td>
<td>30.97 ± 1.80</td>
<td>30.65 ± 0.35</td>
<td>23.18 ± 1.72</td>
<td>34.54</td>
</tr>
<tr>
<td>FAB(8/255)</td>
<td>24.86 ± 2.94</td>
<td>33.92 ± 4.83</td>
<td>25.83 ± 0.18</td>
<td>68.34 ± 1.38</td>
<td>66.83</td>
</tr>
<tr>
<td>Square(8/255)</td>
<td>29.76 ± 3.00</td>
<td>35.4 ± 1.55</td>
<td>28.98 ± 0.17</td>
<td>72.18 ± 0.82</td>
<td>68.19</td>
</tr>
<tr>
<td>AutoAttack(8/255)</td>
<td>24.12 ± 3.15</td>
<td>30.01 ± 1.85</td>
<td>25.37 ± 0.16</td>
<td>9.25 ± 1.32</td>
<td>24.69 ± 4.34</td>
</tr>
<tr>
<td>AutoAttack(4/255)</td>
<td>43.12 ± 1.99</td>
<td>48.09 ± 1.06</td>
<td>41.35 ± 0.14</td>
<td>30.83 ± 2.70</td>
<td>43.26 ± 4.36</td>
</tr>
<tr>
<td>AutoAttack(2/255)</td>
<td>53.94 ± 1.13</td>
<td>57.85 ± 0.36</td>
<td>50.08 ± 0.32</td>
<td>52.49 ± 1.76</td>
<td>55.51 ± 3.44</td>
</tr>
<tr>
<td>AutoAttack(1/255)</td>
<td>59.43 ± 0.72</td>
<td>62.57 ± 0.23</td>
<td>54.38 ± 0.30</td>
<td>64.70 ± 1.23</td>
<td>61.10 ± 2.63</td>
</tr>
<tr>
<td>APGD(0.5)</td>
<td>40.47 ± 1.71</td>
<td>41.73 ± 0.67</td>
<td>43.07 ± 0.38</td>
<td>40.21 ± 2.38</td>
<td>47.61</td>
</tr>
<tr>
<td>FAB(0.5)</td>
<td>40.23 ± 1.65</td>
<td>41.26 ± 0.67</td>
<td>39.97 ± 0.56</td>
<td>70.23 ± 1.11</td>
<td>67.45</td>
</tr>
<tr>
<td>Square(0.5)</td>
<td>51.8 ± 1.19</td>
<td>54.28 ± 0.56</td>
<td>48.78 ± 0.28</td>
<td>76.31 ± 0.78</td>
<td>70.38</td>
</tr>
<tr>
<td>AutoAttack(0.5)</td>
<td>39.42 ± 2.49</td>
<td>40.87 ± 0.66</td>
<td>39.79 ± 0.59</td>
<td>23.54 ± 2.10</td>
<td>39.55 ± 3.93</td>
</tr>
<tr>
<td>AutoAttack(0.3)</td>
<td>49.86 ± 1.21</td>
<td>51.83 ± 0.53</td>
<td>47.44 ± 0.39</td>
<td>41.79 ± 2.61</td>
<td>48.00 ± 3.53</td>
</tr>
<tr>
<td>AutoAttack(0.2)</td>
<td>54.91 ± 0.46</td>
<td>57.11 ± 0.21</td>
<td>51.33 ± 0.37</td>
<td>53.73 ± 1.83</td>
<td>53.46 ± 3.27</td>
</tr>
<tr>
<td>AutoAttack(0.1)</td>
<td>59.80 ± 0.46</td>
<td>62.27 ± 0.14</td>
<td>55.03 ± 0.25</td>
<td>65.07 ± 0.91</td>
<td>61.61 ± 2.44</td>
</tr>
</tbody>
</table>

Figure B.6: Feature-based kernels.
Figure B.7: Predictions of each individual network on CIFAR-10 test set. The left and right halves of each plot correspond to the two networks. Horizontal labels show the ground truth classes, vertical axis shows the predicted probabilities in percentage.