Adversarial Logit Separation

Zixi Chen [*1] **Jinli Xiao** [*1] **Yifei Zhu** [*1] **Jiawei Zhang** [*1]

Abstract

Model ensemble is a powerful tool to enhance classification accuracy and robustness. One key for ensemble is the model diversification. In this paper, we propose a sequential iterative method to optimize the cosine distance of output logits to maximize model diversity with a cosine annealing schedule to stabalize convergence. We call this strategy *logit separation*. This ensemble joint training technique needs to be added to a base method that acts on individual models. For this paper, we choose TRADES (Zhang et al., 2019) as our base model. We demonstrate that our method boosts adversarial robustness with theoretical justification and empirical evidence.

1. Introduction

Deep Learning has shown its progress in the past ten years. It has demonstrated its capability to solve various problems, ranging from computer visions (He et al., 2016), natural language processing (Vaswani et al., 2017), recommendation system (Covington et al., 2016), etc. However, since Szegedy et al. first proposed the existence of Adversarial Examples and the concept of Adversarial Attack (Szegedy et al., 2013), a lot of experiments emerge and reinforce that deep neural networks are not flawless. They are not actually as robust as we have previously assumed.

Moreover, notice that adversarial attack does not merely exist within the digital world. They can actually have physical consequences and bring thread to the security of machine learning models (Kurakin et al., 2016) (Eykholt et al., 2018).

Though abundant adversarial training methods are proposed, it remains an open problem to solve the problem of adversarial attacks and improve adversarial robustness. As pointed out in Athalye et al., many of the proposed ways to defend against adversarial attacks are just some heuristics, or tricks, that circumvent the adversarial attacks (Athalye et al., 2018).

This paper focuses on using model ensemble to increase model robustness. Tramer et al. is the first classical research in this field (Tramèr et al., 2017). It utilizes the transferability of adversarial examples to prevent the models from circumventing the generation process of adversarial examples by smoothing the loss function near the input data point. Though the models break down on following research by the author's own account (Tramèr et al., 2017), we believe the underlying idea is sound and has potential.

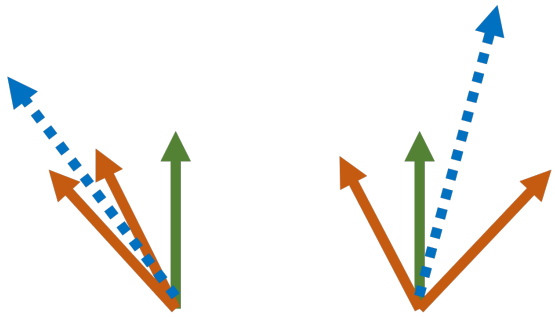

Figure 1. The left figure illustrates the training outcome if we conduct adversarial training separately. The green arrow represents the true label, and the orange arrow represents the logits of adversarial trained models. If they are perturbed to the same direction, their ensemble vector (blue dotted) has similarly bad performance. On the right, if models can give diversified output, their average is closer to the true label.

Our approach is to diversify models in an ensemble by enforcing an extra loss on the output logits. In particular, we want to push the logits to be separate. By diversifying the logits, we expect the output of models on adversarial examples also to diversify. So the ensemble after averaging would produce a better result (see Figure 1). Theoretically, diversifying models doesn't necessarily mean a compromise of classification accuracy because it is believed that there are multiple local minima that have similarly low natural loss values (Carlini & Wagner, 2017). We are just forcing them into distant and distinct local minima. In the following sections, we will present our detailed proposed algorithms, a more formal theoretical justification, and experimental results.

In summary, our contributions are

- **Algorithmic**: We propose a loss function based on the notion of *logit separation*, and use it to sequentially train several models to achieve diversity in parameters.

We then suggests preventing model to overfit on the logit separation error by using a simulated annealing scheduler.

- **Theoretical**: We give mathematical explanations on the robustness performance boost resulting from separation of masked logit vector, and analyze how such separation, expressed in our newly introduced loss term of cosine angle, is guaranteed through sequential negative correlation learning in an iterative manner.
- **Experimental**: We conduct studies to experimentally confirm several of our results in the theoretical section. Furthermore, using TRADES as a baseline and adapting our code on it, we show that our method indeed contributes to a boost in adversarial robustness through experiments using l_∞ AutoAttack (8/256) (Croce & Hein, 2020) on CIFAR-10.

2. Algorithms

One of the precursors in robustness training (Madry et al., 2017) talks about the folklore about the landscape of adversarial examples. Starting at various randomized perturbations from natural examples, they discovered that within the l_∞ norm ball, there exist many adversarial examples with similar local maxima on loss value. Some of these adversarial points even have negative dot products with the original PGD direction (Madry et al., 2017).

In light of the above folklore, we believe that the ability to distinguish all of such local maxima should be a characteristic of an optimal learner. Unfortunately manually identifying the complete set of adversarial examples is not computationally feasible at this time. A single PGD learner, absorbing adversarial examples only in a specific direction (since it generates adversary only based on the current model parameter), is likely to be biased. We seek to find a way to capture the whole landscape of adversaries into our model.

Robustness training is essentially guaranteeing that adversarial examples $\tilde{x}$, produced by adding a small ϵ perturbations on natural examples $x \in \mathbb{R}^d$, would not alter the adversarial prediction $h(\tilde{x})$ by much from the original prediction $h(x)$. Traditional models cannot accommodate this adversary well because they depend on a loss function $L(f(x), y)$.

To avoid the biased pitfall of a single model, we design an ensemble method of I models $f_1, f_2, \cdots, f_I$ and force them to each specialize on a different set of adversarial directions. This is done primarily based on the notion of logit separation. Then we combined these models using basic ensemble method to derive a stronger classifier $h = \sum_{i=1}^{I} \alpha_i f_i$.

Before giving the pseudo-code in Section 2.4, we first discuss the core techniques in the following subsections.

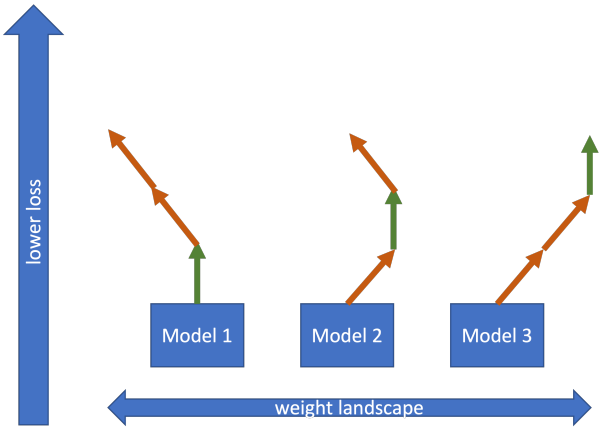

Figure 2. For the sake of illustration, only three models are presented. The green update is when the model is selected to be the reference. When the model is the reference model (which generates adversarial examples), it only optimize for better classification accuracy. Meanwhile, other models will be forced to have separate logit to the reference as well as increase classification accuracy. In real training, the loss and weight landscape are higher dimensional, so they have more flexibility to separate than in this 2D setting.

2.1. Logits Separation

Previous robustness models that utilize logits mainly use it as a measurement of the distance between adversarial prediction and natural prediction that serves as a term to minimize (Kannan et al., 2018). However, we notice that such logit output could also be an indicator of diversity between models trained using the same technique. To this purpose, we add a loss for the logits outputted by different models, in the form of cosine distance, and use backpropagation to diversify the model weights.

Intuitively, a set of models that yield larger distances between logits also indicate greater diversity, and thus would perform better when ensembled together (see Figure 1). We will theologically support this claim in Section 3.

The general idea for logit separation is to force diversity when increasing classification accuracy. By forcing the *non-true logits* (original logits excluding the true label y's component) to be different, we make the models themselves to be diverse after back propagation (see Figure 2). In particular, logit separation is forced on adversarial examples. We want the potential incorrect predictions for adversarial examples to be diverse. Hence, it has a better chance to predict the correct label at the adversarial examples.

Here we introduce a new loss component that seek to enforce diversity between models. The idea is to penalize the cosine distance of logits. In particular, the loss component

is framed as

$$m = \mathbf{1} - y = [1, \cdots, 0, \cdots, 1]^T$$

where the 0 is the y^{th} index

$$L_{\text{sim}}(f_r(\tilde{x}), f_o(\tilde{x}), y) = \lambda \cdot \frac{(f_r(\tilde{x}) * m)^T (f_o(\tilde{x}) * m)}{||f_r(\tilde{x}) * m|| \cdot ||f_o(\tilde{x}) * m||}$$

where f_r is the reference model and f_o is one of the the other models, or the current training model. As models take turns to be the reference model, the logit distance component is 1 when the reference model coincides with the current training model, so we left out L_{sim} for the total loss in this case.

We want to emphasize that our loss serves only as a secondary add-on component that need to ensure both that the original loss function should have the highest priority and that each model should not overfit on L_{sim}. This could either be achieved using a small λ, or simulated annealing, as discussed in section 2.3.

In this paper, we use TRADES (Zhang et al., 2019) as our primary loss function, but other loss functions could also suffice as long as they were trained based on adversarial examples. In our case, the total loss function for function f_o can be written as

$$L_{f_o} = \mathop{\mathbb{E}}_{(x,y)\sim D}[\phi(f_o(x), y) + \lambda_1 \cdot \phi(f_o(\tilde{x}) f_o(x)) + \lambda_2 \cdot L_{\text{sim}}(f_r(\tilde{x}), f_o(\tilde{x}), y)]$$

where ϕ is a surrogate 0-1 loss function.

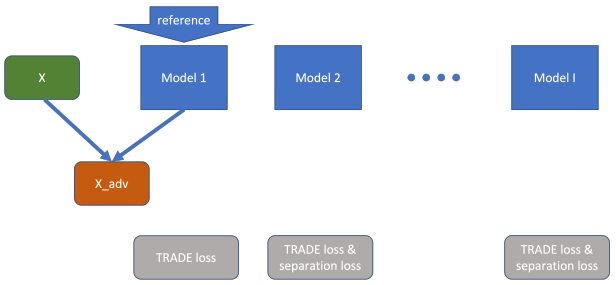

Figure 3. For each (X, y) pair, we assign the "reference" to each of the model in the ensemble. The adversarial example is generated by the reference model. The reference model is updated by TRADE loss while others are updated by both TRADE loss and separation loss. This graph can be read with the pseudo-code in Section 2.4.

2.2. Ensemble sequential iterative methods

Our logit separation specification requires the usage of different models. Due to the transferable nature of adversarial examples (Tramèr et al., 2017), we let models reinforce each other by generating adversarial examples for one another.

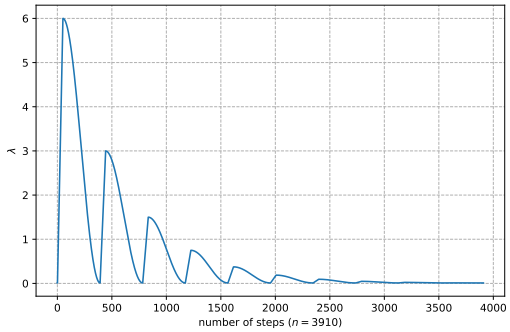

Figure 4. The trend of λ as step grows. λ first grows fast, signifying an energy growth that pushes models to explore different directions at first, and then gradually decreases and converges to 0, preventing models oscillate around optima.

Specifically, on each batch of input data, each model in the ensembles is selected to be the reference model in turn, which is used to generate adversarial examples given the input x. The reference model is updated by the TRADES loss and others are updated by the L_{f_o} defined before. See Figure 3 for visual illustration.

2.3. Cosine Annealing Scheduler

Note that the diversity may be conflicted with the accuracy since forcing the logits to be separate may not align with optimizing the accuracy for classifications, but at the same time, we want the model to explore the loss landscape and not be trapped in close local minima. So, we utilize the trick of cosine annealing to gradually decrease the weight for λ so that the algorithm can explore the landscape at first and then converge. This corresponds with our intuition that we would first want each models to grow in different directions by the influence of λ, and the gradually finds their respective local minima of TRADES loss. See Figure 4 for the trend.

2.4. Pseudocode

See Algorithm 1.

2.5. Comparison With Previous Models

TRADES (Zhang et al., 2019) serves as the baseline for our model. It is considered to be a classical adversarial training method which gives a robust guarantee against adversarial attacks. Some other state-of-the-art algorithms may have a better benchmark, but that don't necessarily indicate better robustness. There are a lot of cases where the "robustness" turns out to be fake as new attack are proposed (Athalye et al., 2018)(Engstrom et al., 2018). Still, at the time this paper is written, TRADES performs decently well on the

Algorithm 1 Logits Separation Joint Training

```
for i ← 1 to I do                          ▷ I is the number of models
    init f_i
end for
shr ← CosineAnnealingScheduler()
repeat
    for (x, y) ∈ S do
        for r ← 1 to I do ▷ f_r s be reference model in turn
            λ ← shr.step()
            x̃ ← PGD(f_r, x)
            for o ← 1 to I do
                L ← TRADES_loss(x, x̃, y)
                m ← 1 − y
                L ← L + 1_{r≠o} λ · (f_r(x̃)∗m)^T (f_o(x̃)∗m) / (‖f_r(x̃)∗m‖ · ‖f_o(x̃)∗m‖)
                Conduct gradient descent on loss L for f_o
            end for
        end for
    end for
until λ converges
return ensemble of models
```

adversarial benchmark (Croce et al., 2021). So, we believe the idea of TRADES is solid and insightful. TRADES adds a term for robustness regularization to the natural loss during training, encouraging the models to learn the good input while be aware of the adversarial ones. The loss for TRADES can be expressed as

$$L = \phi(f(X), Y) + \lambda\phi(f(X)f(\tilde{X})$$

ALP (Kannan et al., 2018), or Adversarial Logit Pairing, poses an alternative way to train a single model against adversarial examples. It requires the logit on adversarial examples to approach the logit on clean data. The loss used in ALP can be expressed as

$$L = t\phi(f(X), Y) + (1-t)\phi(f(\tilde{X}, Y) + \lambda\phi'(f(\tilde{X}), f(X))$$

However, the robustness stated in the paper is proved to be easily compromised (Engstrom et al., 2018). Nevertheless, we believe that the logits of the model output is a good indicator of the inherent property of the model. The reason it fails may be that the model is not necessarily the expert on adversarial examples, so the predictions on adversarial examples close to logits on clean samples doesn't necessarily lead to good predictions. We use logits in a totally opposite way. Our approach is agnostic and doesn't assume which model in the ensemble will perform well on the incoming sample. Instead, by forcing apart, we argue that the means of the prediction is more accurate and by spreading the weight, it is more likely one of the model in the ensemble performs well.

ADP (Pang et al., 2019), or Adaptive Diversity Promoting, uses the inner product between each pair of logits to denote the diversity of the model ensemble. During training, it adds the separation component to the loss to optimize both accuracy and diversity. The proposed regularization term is the weighted sum of the entropy of the ensemble and the sum of the inner products from two different models' logit in the ensemble. The loss can be expressed as

$$L = \sum \phi(f_k, X, y) + \alpha H(\mathcal{F}) + \beta \log\left(\sum_{i\neq j}(f_i(X))^T f_j(X)\right)$$

Though the author mentions the increasing robustness against adversarial attacks, they didn't add adversarial examples to the datasets and the AutoAttack (Croce & Hein, 2020) benchmark is unknown.

Although the metric we try to optimize is similar to what proposed by ADP, our algorithms are remarkably different. One significant difference is that instead of incorporating the distance of each pair of models be the diversity loss, we optimize the distance iteratively and sequentially (Section 2.2). In particular, when choosing one model as the reference, only the remaining models are forced to diversify. The reason behinds this is to avoid local smoothness around the data points, which result in producing weaker adversarial examples, rather than improving defense (Tramèr et al., 2017). In addition, we introduce the trick of cosine annealing so that the models can diverge at first and then converge to respective optima.

3. Theoretical Explanations and Intuitions

In this section, we are to closely examine (1) the theoretical foundations of our sequential learning method to minimize the cosine loss term and (2) how that terms adds to the model's robustness.

3.1. Sequential training implementation guarantees

There are three major strategies for training ensembles, i.e., independent training, simultaneous training, and sequential training (Islam et al., 2003). In our algorithm we are doing the sequential training, where individual neural networks take turns to be trained one after another. When the other models are trained, weights in the reference model are frozen.

The reason we select this training method is that it not only targets minimizing accuracy and robust accuracy, but also de-correlates the error of the model being trained from the previously trained models so that their output logits are as separated as possible. Moreover, since our goal is to improve adversarial robustness, we avoid the pitfall of the model circumventing the adversarial examples generated by itself(Athalye et al., 2018). we let the reference model

generate the adversarial examples for the other models. The algorithmic details are shown in the pseudo-code in Section 2.4.

From a statistical point of view, by minimizing the cosine angles among the non-maximal logits of the sub-models, we are actually separating them by reducing correlations among them, which is the central idea of this paper. Previous works with similar learning goal include (Liu & Yao, 1999) and (Rosen, 1996). However, their way of defining ensemble complexity relies on the outdated assumption that each sub-model is a weak learner itself. In our settings, they are actually strong learners. Rather, we claim that when minimizing the cosine angles, we optimize diversity and uncorrelation in a more direct way.

From our definition of L_{sim} in Section 2.1, it is safe to assume that, without the loss of generality, each component output of $f_r(\tilde{x}) * m$ and $f_o(\tilde{x}) * m$ as a random variable of zero expectation, since they represent the prediction confidence at non-maximal class label. Then the logits become random vectors. By connection between linear algebra and probability theory, notice that the expectation of the product of two zero-expectation random vectors satisfy:

1. bilinearity

$$\mathbb{E}[(X+Y)Z] = \mathbb{E}[XZ] + \mathbb{E}[YZ]$$
$$\mathbb{E}[kXY] = k \cdot \mathbb{E}[XY]$$

2. symmetry

$$\mathbb{E}[XY] = \mathbb{E}[YX]$$

3. non-degenerativity

$$\mathbb{E}[X^2] = 0 \iff X = 0 \text{ almost everywhere.}$$

Thus it fulfills the definition of a inner product for two random vectors. In addition, the length (norm) of random vector is its standard deviation. Therefore we have that the cosine of the angle between two vectors is their correlation.

$$\begin{aligned}
&\cos\left(f_r(\tilde{x}) * m, f_o(\tilde{x}) * m\right) \\
&= \frac{(f_r(\tilde{x}) * m)^T (f_o(\tilde{x}) * m)}{||f_r(\tilde{x}) * m|| \cdot ||f_o(\tilde{x}) * m||} \\
&= \frac{\text{Cov}(f_r(\tilde{x}) * m, f_o(\tilde{x}) * m)}{\sigma(f_r(\tilde{x}) * m) \cdot \sigma(f_o(\tilde{x}) * m)} \\
&= \text{Corr}(f_r(\tilde{x}) * m, f_o(\tilde{x}) * m)
\end{aligned}$$

3.2. Model robustness improved by separation

Intuitively, if we force logit output from each individual submodel to be more separate, since the separation is measured in the cosine angle of the non-maximal predictions in the labels, our ensemble model becomes more diverse and thus more robust. We express such intuition in mathematical language, with inspiration from proof of the non-maximal entropy loss model (Pang et al., 2018).

Theorem 1. *Given n normalized unit vectors in $\mathbb{R}^n$, say $v_1, \ldots, v_n$, and they form the columns of the matrix M_n. Then the determinant*

$$\det(M_n)$$

is maximized if and only if the sum

$$\sum_{\substack{i,j=1 \\ i \neq j}}^{n} (cos([v_i, v_j]))^2$$

is minimized, where $[v_i, v_j]$ denotes the angle between vectors v_i and v_j.

Proof. (based on geometric interpretation)
From basic linear algebra, the determinant of a $n \times n$ matrix is the signed volume of the parallelepided spanned by the column vectors of the matrix. The sign tells us whether the column vectors form a left-handed or a right-handed basis. And if they don't form a basis, the n vectors are linear dependent, so the determinant is zero. Since each vector v_i is a normalized unit vector, when the vectors form an orthogonormal basis for $\mathbb{R}^n$, meaning the square sum of their cosine values reaches the minimum of 0, the signed volume equals the maximum value, the product of the norm of all vectors. In this case, 1.

Instead, when the determinant attains maximum, WLOG, we consider the case when all other vectors remain unchanged and modify v_1. The volume is maximized if and only if the distance of end point of v_1 is as far from the hyperplane spanned by $\{v_2, \ldots, v_n\}$ as possible. Since $||v_1|| = 1$, by Pythagorean theorem, the distance is maximized when the projection is minimized, which is equivalent to that the square cosine value of the angle between v_1 and the hyperplane is minimized. □

Then, we can see that our loss function design is in fact equivalent to the ensemble diversity defined by (Pang et al., 2019) as:

$$\mathbb{ED} = det(\tilde{M}_{\setminus y}^T \tilde{M}_{\setminus y})$$

where $\tilde{M}_{\setminus y} = (\tilde{F}_{\setminus y}^1, \ldots, \tilde{F}_{\setminus y}^K)$, each column vector $\tilde{F}_y^i$ is the l-2 norm normalization of the non-maximal logit given by the i-th submodel (Pang et al., 2019). The authors, in another paper gives the theoretical guarantee that this model could converge to the solution space (Pang et al., 2018).

So in the decision space $A_{\hat{y}}$ of class $\hat{y}$, $\forall i, j \neq \hat{y}, z_{\hat{y}} \in A_{\hat{y}}$, we have a unique $a^0_{i,j} \in A_{i,j}$ such that

$$z_{\hat{y}} \in \underset{i,j\neq\hat{y}}{\cap} a^0_{i,j} = Q_0$$

And the solution set to the problem is

$$\arg\min_{z_{\hat{y}}} \left(\max_{z\in Q_0 \cap \bar{A}_{\hat{y}}} (f(z))_{\hat{y}} \right) = S_{\hat{y}}$$

where

$$\begin{aligned} a_{i,j} &= \{(f_r(\tilde{x}))_i = (f_o(\tilde{x}))_j\} \\ A_{i,j} &= \{(f_r(\tilde{x}))_i = (f_o(\tilde{x}))_j + c, c \in \mathbb{R}\} \\ a^+_{i,j} &= \{(f_r(\tilde{x}))_i \geq (f_o(\tilde{x}))_j\} \end{aligned}$$

are the set of hyperplanes, affine planes, and half-spaces divided by the hyperplanes, respectively. Our decision space for class $\hat{y}$ could be expressed by $A_{\hat{y}}$

$$A_{\hat{y}} = \underset{i\neq\hat{y}}{\cap} a^+_{\hat{y},i}$$

The boundary of a decision space is $\bar{A}_{\hat{y}}$. And our targeted solution space $S_{\hat{y}}$ is

$$S_{\hat{y}} = (\underset{i,j\neq\hat{y}}{\cap} a_{i,j}) \cap A_{\hat{y}}$$

This space is of dimension $l-1$ embedded in the decision space and is thus a lower-dimensional manifold in which our output $f_r(\tilde{x})$ and $f_o(\tilde{x})$ should have $l-1$ equal non-maximal components for arbitrary r and o.

We have proven that by forcing each sub-model to be separated, we could achieve a space where the ensemble model is more robust.

4. Experiments and Results

We adapted code from our chosen baseline model, TRADES, to conduct experiments. Our modified code could be found at https://github.com/CharlleChen/fml-final-project. We train our model on CIFAR-10 dataset, and using l_∞ ball perturbation in our robust training. We use $I = 3$ models in our ensemble. The batch size is 128. β for TRADES regularization term is 6 (default). The annealing schedule for λ (weight for logit separation) has largest value 1 and smallest value 0.01. The trained model weights can be found at https://drive.google.com/drive/folders/1duDKLafpdSINIg7Y_QpYeTXXSiBmXVBr?usp=sharing.

Due to time and resource limits, we trained the ensemble from scratch on RTX8000 (40G) for 10 epochs in 20 hours. Note that this is way less than the benchmark given by TRADES, which requires 100 epoch training (Zhang et al., 2019). Hence, instead of comparing to the benchmark given by the original paper, we compare our result with the baseline (naive ensemble of three randomly initialized models) trained with the same time and computation resources. It turns out that our methods alternate the models in the expected way and it has performance gains.

4.1. Vector Angle

	clean data	PGD adversarial data
Baseline	0.43087897	0.4294527
Ours	**0.4010905**	**0.3988733**

Table 1. Mean cosine distance of logit outputted by model pairs.

Table 1 shows the cosine distance of models in the ensemble. It is averaged through all test data and three pairs of models from the ensemble. By our algorithms, we see that the outputted logit on clean and adversarial examples both decrease. This means that it fulfills our wish to diversify the models in the ensemble.

4.2. Sub-model Performance

		model 1	model 2	model 3
Baseline	initial	73.50%	**74.30%**	73.90%
	adversarial	36.90%	35.30%	36.40%
Ours	initial	72.80%	72.90%	73.00%
	adversarial	37.00%	**37.50%**	37.20%

Table 2. Accuracy of individual models of the ensemble on clean and adversarial data (AutoAttack)

Table 2 shows the performance of individual models on initial and adversarial accuracy of AutoAttack. We can see that the initial accuracy of baselines are generally higher than our method, but our methods are better after the adversarial attack. This means that our model is more robust.

4.3. Ensemble Result

	Initial accuracy	AutoAttack accuracy
Baseline	**73.80%**	37.50%
Ours	73.00%	**38.6%**

Table 3. Ensemble accuracy before and after AutoAttack.

Table 3 shows the overall accuracy of the ensemble model. It has a consistent result with individual models: the initial accuracy is higher for baseline but the AutoAttack accuracy is higher for our method. This indicates that our model is indeed valid.

We haven't trained the model as long as the 100 epochs stated in the TRADES paper, but we have reasons to believe that it can gains even more performance as the model

converge. Also, we don't have time to conduct repeated research to demonstrate the statistical significance of the improvement, but our empirical results fit our theoretical expectations and has no obvious discrepancy. Further experiments with different hyperparameters (different λ, β, annealing schedule, etc) and ablation studies are required to improve the understanding on this field.

5. Conclusion

Through experimentation and theoretical analysis, we showed logical separation to be a valid and potentially very useful technique for add-ons to improve robustness training. Sequential iteration methods and cosine annealing schedule are proven to be useful for the model to converge to better optima. We showed the improved performance of TRADES and gave theoretical or intuitive analysis on many assumptions we made.

Further research on diversity training could also be made. In this paper, we only take TRADES as an example and uses CIFAR-10 as the only dataset. They are many potential improvements such as generalizing our method based on other primary models and training using different datasets.

References

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pp. 191–198, 2016.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://openreview.net/forum?id=SSKZPJCt7B.

Engstrom, L., Ilyas, A., and Athalye, A. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1625–1634, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Islam, M. M., Yao, X., and Murase, K. A constructive algorithm for training cooperative neural network ensembles. *IEEE Transactions on neural networks*, 14(4):820–834, 2003.

Kannan, H., Kurakin, A., and Goodfellow, I. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Kurakin, A., Goodfellow, I., Bengio, S., et al. Adversarial examples in the physical world, 2016.

Liu, Y. and Yao, X. Ensemble learning via negative correlation. *Neural networks*, 12(10):1399–1404, 1999.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks, 2017. URL https://arxiv.org/abs/1706.06083.

Pang, T., Du, C., Dong, Y., and Zhu, J. Towards robust detection of adversarial examples. *Advances in Neural Information Processing Systems*, 31, 2018.

Pang, T., Xu, K., Du, C., Chen, N., and Zhu, J. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pp. 4970–4979. PMLR, 2019.

Rosen, B. E. Ensemble learning using decorrelated neural networks. *Connection science*, 8(3-4):373–384, 1996.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.