

Supplementary Materials

Embracing Adaptation: An Effective Dynamic Defense Strategy Against Adversarial Examples

Anonymous Authors

1 ALGORITHM

Algorithm 1 Dynamic Defense Strategy

Input: Model $f_{\theta_0}(\cdot)$, input $\mathcal{X} = \{\mathbf{x}_b\}_{b=1}^B$, number of iterative steps N_{DIO} , step size ζ and number of optimization steps N_{DMO} ;
Output: Final prediction;

- 1: # Dynamic input optimization (DIO) algorithm
- 2: Obtain pseudo-label $\mathcal{Y} = \{y_p\}_{b=1}^B$ by Eq. 2
- 3: **for** $i = 1$ to N_{DIO} **do**
- 4: $\mathcal{X}^i = \text{Clip}_{\mathcal{X}, \xi}(\mathcal{X}^{i-1} + \zeta \text{sign}(\nabla_{\mathcal{X}} L(f_{\theta_0}(\mathcal{X}^{i-1}), \mathcal{Y}))$
- 5: **end for**
- 6: # Samples Filtering
- 7: $\mathcal{X}_{DIO} \leftarrow \mathcal{X}^i$
- 8: Calculate the entropy value for all $\mathbf{x} \in \mathcal{X}_{DIO}$ via $f_{\theta}(\cdot)$.
- 9: The filtered samples \mathcal{X}_{DIO-SF} are obtained through Eq. 12.
- 10: # Dynamic model optimization (DMO) algorithm
- 11: Initializing teacher and student model:
- 12: $f_{\theta_0}^T(\cdot) = f_{\theta}(\cdot)$
- 13: $f_{\theta_0}^S(\cdot) = f_{\theta}(\cdot)$
- 14: **for** $i = 1$ to N_{DMO} **do**
- 15: $\{\hat{y}^S\}_{b=1}^B = f_{\theta_t}^S(\mathcal{X}_{DIO-SF})$
- 16: $\{\hat{y}^T\}_{b=1}^B = \frac{1}{N} \sum_{i=1}^N f_{\theta_t}^T(\text{aug}_i(\mathcal{X}_{DIO-SF}))$
- 17: optimizing the student model by Eq. 8
- 18: optimizing the teacher model by Eq. 9
- 19: **end for**
- 20: $\{\hat{y}^S\}_{b=1}^B = f_{\theta_{N_{DMO}}}^T(\mathcal{X}_{DIO})$
- 21: Return $\{\hat{y}^S\}_{b=1}^B$

2 MORE EXPERIMENTAL RESULT

2.1 Results in the normally trained model.

We trained a WideResNet-28-10 model on the CIFAR-10 dataset using normal training. We then evaluated the model using different defense methods and various attacks with a perturbation size of 4/255. Table 1 shows the experimental results. We observed a significant decrease in the effectiveness of Dent and Anti under the normally trained model. Our method also experienced a decrease, but we still achieved favorable results. We analyze the reasons for the decrease in effectiveness in Section 2.10 (Further Analysis).

2.2 Impact of the optimized number of steps on results

In this section, we investigate the impact of the number of optimization steps on the generated results. We conducted experiments using the adversarially trained ResNet-18 model on the CIFAR-10 dataset, with PGD-20 as the attack. The experimental results are presented

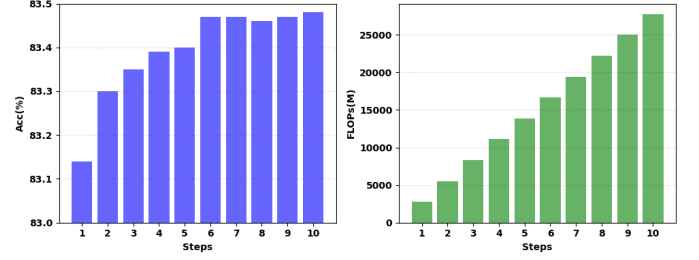


Figure 1: The effect of the optimization number of steps on the computational efficiency and accuracy.

in Figure 1. The figure illustrates that the number of steps has a significant effect on the defense’s effectiveness and the reasoning time. As the number of steps increases, the defense becomes stronger. However, our proposed method does not benefit significantly from a large number of optimization steps, and instead incurs a significant computational cost. This is because our method already significantly enhances the model’s robustness with a few optimization steps, resulting in the model’s performance on adversarial examples being close to that on clean samples. Therefore, it becomes difficult to further improve the model’s robustness by increasing the number of optimization steps. Although the dynamic defense strategy increases the computational cost of the inference process, it also imposes a higher cost on the attacker.

2.3 The effect of β on experimental results.

We verified the effect of β on the experimental results by conducting experiments using the PGD-20 attack under the CIFAR-10 dataset. The results are shown in Figure 2. When $\beta = 1$, it indicates that no samples were suppressed, while a smaller β value indicates greater suppression of the sample. As can be seen from the experimental results, the model’s accuracy decreases as β becomes smaller.

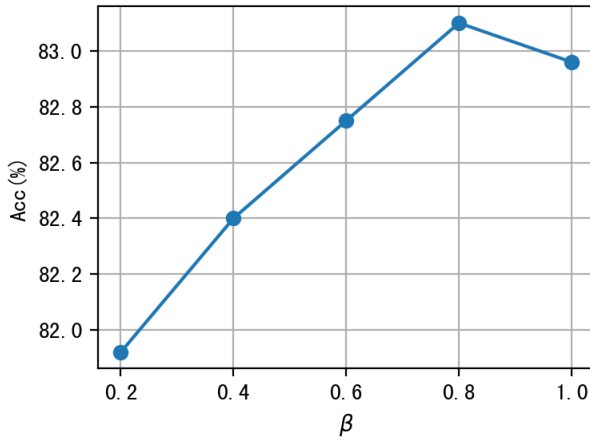
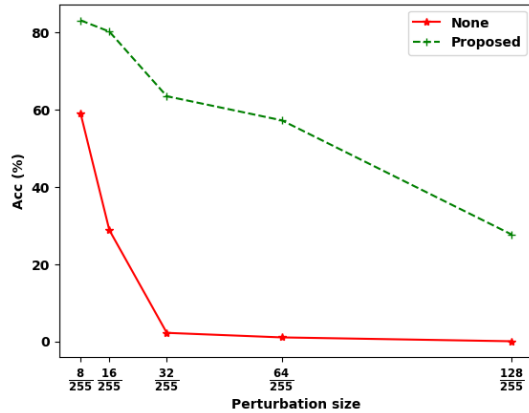
By suppressing uncertain pseudo-label samples, we can ensure the stability of the model training. However, if we suppress these samples too much, we may not fully utilize the knowledge provided by the target feature distribution. Therefore, we use a medium-sized value for β to suppress the uncertain samples, which allows us to ensure the stability of the model training and also enables the model to learn the knowledge provided by the feature distribution.

2.4 Attacks with higher norm bounds

We evaluated the performance of our proposed method against attacks using different sizes of adversarial perturbations and the experimental results are shown in Figure 3. A sufficiently large perturbation should allow the attack to achieve a high success rate [2]. The robustness of our proposed method decreases as we increase the size of the perturbations. Despite this, our method still enhances the

Table 1: Results under the normally trained model. The best results are boldfaced, and the second best results are underlined.

Method	PGD-20	PGD-50	C&W	APGD-CE	APGD-t	FAB-t	Square
None	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	13.95%
Dent	03.49%	02.45%	31.88%	05.59%	00.05%	51.15%	73.25%
Anti	<u>65.45%</u>	<u>63.04%</u>	<u>65.63%</u>	<u>36.95%</u>	<u>30.85%</u>	<u>73.90%</u>	46.20%
Proposed	80.58%	78.84%	80.78%	42.70%	39.70%	76.50%	<u>55.70%</u>

**Figure 2: The effect of different β on experimental results.****Figure 3: The effect of different sizes of perturbations on the results.**

model's ability to resist adversarial examples with large perturbations compared to the model without defense. Although the attack used to evaluate our method can find adversarial samples with large perturbations, our method improves the model's robustness against such attacks.

2.5 Evaluation of GMM.

We set up a series of comparative experiments, including the case without GMM clustering and the scenario with K-Means clustering, and the specific experimental results are shown in Table 2. From these results, it is obvious that incorporating the GMM clustering

Table 2: The impact of different pseudo label generation methods on the results.

Method	Clean	PGD-20	PGD-50	C&W	RayS	AVG
Model Prediction	82.43%	79.88%	79.49%	78.77%	70.15%	78.14%
K-Means	83.41%	81.11%	80.93%	79.66%	69.48%	78.92%
GMM	84.57%	83.10%	83.03%	81.30%	72.90%	80.98%

Table 3: Comparison of results using weighted method (w/) and not using weighted method (w/o).

Method	Clean	PGD-20	PGD-50	C&W	RayS	AVG
w/o	84.13%	81.69%	81.50%	80.17%	71.33%	79.76%
w/	84.57%	83.10%	83.03%	81.30%	72.90%	80.98%

information into the model significantly improves robustness in the face of adversarial attacks. GMM clustering greatly improves the robustness of the model by digging deeper into the internal structure of the data and providing exhaustive soft clustering information, which greatly enriches the structured representation of the data, which plays a key role in enhancing the accuracy and credibility of the pseudo-labeling. This not only helps the model to understand the complex and subtle data patterns more deeply, thus improving its performance on normal data, but also under counterattacks, this deep understanding and structured information helps to enhance the model's defense capability and reduce the risk of being attacked. In contrast, although the hard clustering method of K-Means also provides clustering information, the soft clustering method of GMM provides the model with a more detailed ability to determine the boundary regions due to its more comprehensive information on the probability of data points belonging to each cluster. This enables the model to show higher discriminative ability and robustness in the face of well-designed adversarial samples. Therefore, these experimental results fully demonstrate the importance and effectiveness of incorporating GMM clustering information to enhance the robustness of the model in an adversarial attack environment.

2.6 Evaluation of sample weighting.

If pseudo-labels are incorrect, model optimization will veer off course, impacting model performance. Past work overlooked this; we mitigate it by assigning weights to each sample via the method in Eq. 5, reducing the effect of incorrect pseudo-labels. The comparison in Table 3 illustrates the improvement over the unweighted approach. From the experimental results, we can see that our proposed sample weighting method can well alleviate the problem of error accumulation.

Table 4: Comparison of results from different optimization methods.

Method	Clean	PGD-20	PGD-50	C&W	RayS	AVG
Traditional optimization	83.29%	81.32%	80.93%	80.48%	72.34%	79.67%
hierarchical optimization	84.57%	83.10%	83.03%	81.30%	72.90%	80.98%

2.7 Evaluation of hierarchical optimization.

To assess the advantages of hierarchical optimization, we compared it with traditional optimization methods, and the experimental results are shown in Table 4.

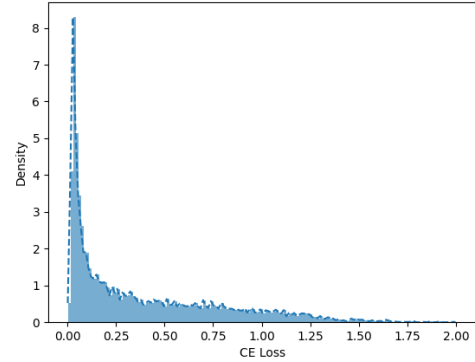
In deep learning, especially in the context of adversarial attacks, model robustness and stability are crucial. Traditional optimization methods that adjust the parameters of all layers simultaneously, while effective in general, may not be robust enough under adversarial attacks. This is because this approach may not be able to adequately take into account the differences in the sensitivity of different layers to perturbations, resulting in a model that is too sensitive to perturbations. In contrast, the hierarchical optimization approach adjusts the parameters of the model layer by layer in a staged manner, allowing each layer to first have a stable understanding of the underlying features before passing information to higher layers. In this approach, lower layers are optimized first and thus learn robust feature representations. This means that even if the upper layers are affected by adversarial perturbations, the base layer has already established a stable recognition of the key information, reducing the accumulation and spread of errors. Since the feature representations at lower levels are more robust, the model becomes less sensitive to small changes in the input data (e.g., adversarial perturbations) as training advances to higher levels. Thus, hierarchical optimization provides a more effective way to deal with adversarial attacks and enhances the robustness of the model. Experimental results show that this approach reduces the error accumulation caused by adversarial attacks, thus improving the overall model stability and performance.

2.8 Further Analysis

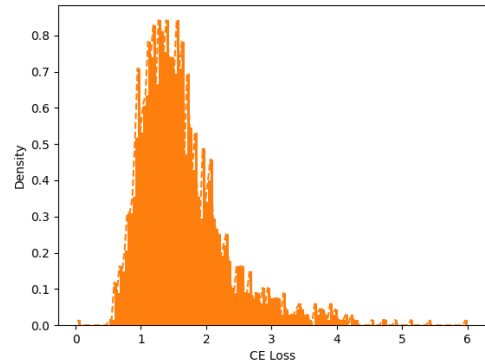
Our study employs Huang et al. [1]'s method and presents histograms of the loss values for both successful and unsuccessful samples using the proposed method, which is illustrated in Figure 4. In this context, a sample is considered successful if the proposed method can correctly predict the label of the adversarial example. Conversely, if the proposed method fails to correct the model's incorrect prediction, the sample is considered unsuccessful.

Our findings demonstrate that the proposed method is capable of handling adversarial examples with moderate loss values that are located near the decision boundary. However, for highly misclassified samples with large loss values, the proposed method may not be able to make significant changes in the model to accurately predict their labels. This highlights why we have chosen to use an adversarially trained model as a starting point for the proposed method.

The advantages of an adversarially trained model are also apparent in comparison to a normally trained model. An attacker can easily manipulate normal samples on a normally trained model, causing them to cross over and move away from the correct classification



(a) Successful



(b) Unsuccessful

Figure 4: Histogram of loss values for successful and unsuccessful proposed methods.

boundary. On the other hand, on an adversarially trained model, an attacker would need to pay a much higher cost to move normal samples away from the correct classification boundary.

3 REAL-WORLD USABILITY ANALYSIS

While we acknowledge that the proposed approach may incur additional overhead, we maintain that our methodology remains applicable in practical settings. In the field of medicine, for example, federated learning is commonly employed to safeguard data security during model training. In such scenarios, a complete data set may not be readily available, and techniques such as adversarial training may not be viable options for ensuring model robustness. Furthermore, the costs associated with retraining a model that is already online can be exorbitant. Our approach offers a means of enhancing model robustness without the need for training data. Additionally, in fields such as pathologic identification in medicine and UAV detection and identification in the military, accuracy is of paramount importance,

349 outweighing considerations of time consumption. Thus, the addi-
350 tional costs incurred by our approach are deemed acceptable in order
351 to attain superior accuracy in these domains.

REFERENCES

- [1] Zhichao Huang, Chen Liu, Mathieu Salzmann, Sabine Süsstrunk, and Tong Zhang. 2023. Test-time Adaptation for Better Adversarial Robustness. <https://openreview.net/forum?id=rUxKM6u8WER>
- [2] Dequan Wang, An Ju, Evan Shelhamer, David Wagner, and Trevor Darrell. 2021. Fighting Gradients with Gradients: Dynamic Defenses against Adversarial Attacks. *arXiv preprint arXiv:2105.08714* (2021).