Adversarial Distillation Based on Slack Matching and Attribution Region Alignment
Supplementary Material

Shenglin Yin[1], Zhen Xiao[1*], Mingxuan Song[1], Jieyi Long[2]
[1]School of Computer Science, Peking University
[2]Theta Labs, Inc.
{yinsl, songmingxuan}@stu.pku.edu.cn xiaozhen@pku.edu.cn jieyi@thetalabs.org

1. Further Analysis

1.1. Limitations of AD Methods Based on KL

In this section, we further analyse the effect of KL on AD. When using Kullback-Leibler (KL) as the loss function, we expect the output probability distribution of the student model to be close to the output probability distribution of the teacher model. KL is defined as follows:

$$KL(P \parallel Q) = \sum_i P(i) log(\frac{P(i)}{Q(i)}), \tag{1}$$

where P and Q are two probability distributions. Within the context of AD, P typically denotes the output probability distribution of the teacher model, whereas Q corresponds to that of the student model.

Below are a few key points that illustrate why the use of KL may harm the performance of the student model when the performance of the teacher model increases dramatically:

Deterministic teacher models:

The KL equation (Eq. 1) can be reformulated as:

$$KL(P \parallel Q) = H(P, Q) - H(P), \tag{2}$$

where $H(P, Q) = -\sum_i P(i) log(Q(i))$ represents the cross-entropy and $H(P) = -\sum_i P(i) log(P(i))$ represents the entropy of the teacher model output. The probability distributions of the teacher models with different abilities on clean and adversarial examples are presented in Figure 1. From this, we can observe that the higher the ability of the model, the higher its confidence level for a particular category, with the probability distribution of the corresponding category converging to 1, while the probabilities of the other categories remain close to 0. This indicates that as the confidence level of the teacher model increases, its output entropy $H(P)$ will gradually approach 0. At the same time, the probability of the non-target category in P also tends to 0, while the probability of the target category is close to 1. In this case, the KL dispersion essentially evolves into the cross entropy in the hard-labelled environment. This implies that the more capable the teacher model is, the more the knowledge distillation process may tend to degenerate into a traditional adversarial learning process based on hard labels, which would make it difficult for the student model to effectively learn from the teacher model. In particular, when the predictive confidence of the teacher model is extremely high (regardless of whether its predictions are accurate or not), this may provide misleading labelling information to the student model, which in turn reduces learning effectiveness.

Temperature adjustment may not be sufficient:

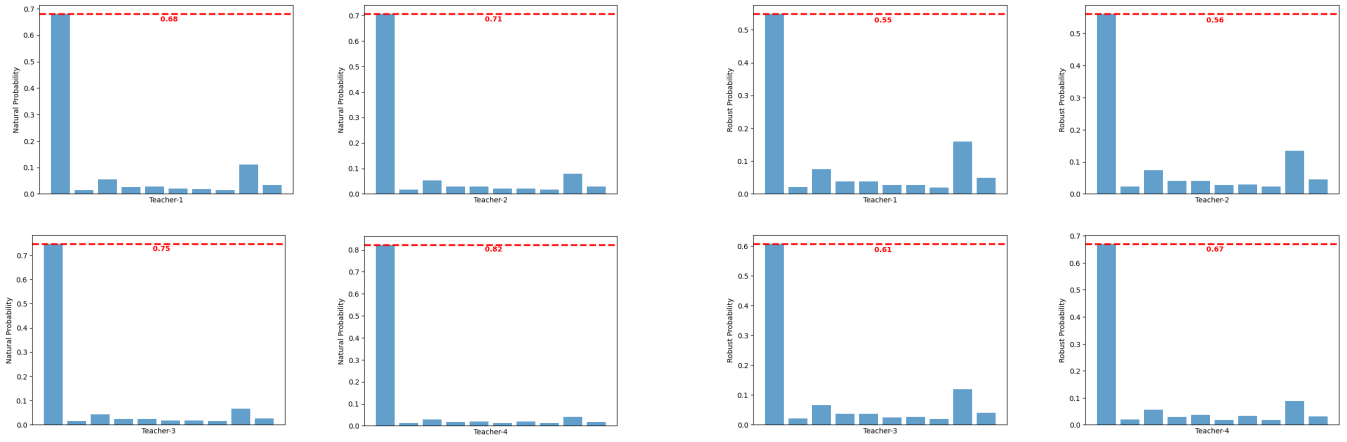
Modifying logits using temperature T redefines the softmax function as:

$$Softmax(z_i) = \frac{e^{\frac{z_i}{T}}}{\sum_j e^{\frac{z_j}{T}}}. \tag{3}$$

For $T > 1$, this adaptation yields a more uniform probability distribution. However, it isn't universally beneficial. An elevated T can diminish the teacher model's distinct "knowledge", depriving the student model of crucial learning cues. For categories in which the teacher model is otherwise confident, high temperatures can greatly reduce the probability of these categories, which may result in the student model not being sufficiently incentivised to mimic these deterministic predictions. Hence, augmenting the temperature T doesn't invariably enhance the student model's learning, especially when juxtaposed with a superior teacher model.

As previously mentioned, when the student model's capability is far inferior to the teacher model, emphasizing exact matching might lead to a decline in the performance of the student model. In contrast to this, as shown in Figure 2, we are not concerned about the exact probability val-

* Corresponding author.



Model	Teacher-1	Teacher-2	Teacher-3	Teacher-4
Clean	83.53%	87.33%	88.54%	92.23%

Model	Teacher-1	Teacher-2	Teacher-3	Teacher-4
AA	56.66%	60.73%	64.20%	66.56%

Figure 1. Probability distributions of different teacher models on clean and adversarial samples for the 'Airplane' category in the CIFAR-10 dataset. 'Clean' represents accuracy on clean samples. 'AA' indicates accuracy under AutoAttack. Teacher-1 employs ResNet-18, Teacher-2 employs WRN-28-10, Teacher-3 employs WRN-70-10, and Teacher-4 employs WRN-70-10 with additional training data.

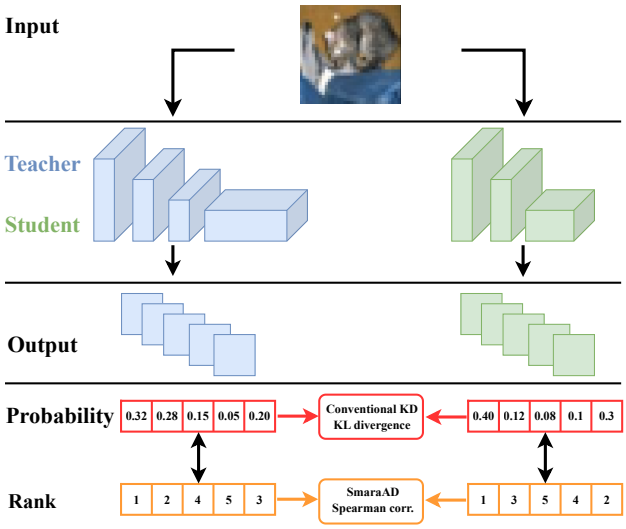


Figure 2. Difference between our SmaraAD and existing KD methods.

ues given by the two models; instead, we are interested in the similarity in the ranking of their predictions. The core idea of this method is: when the student model's ability is significantly below that of the teacher model, we shouldn't expect the student model to perfectly imitate the predictions of the teacher model. Instead, we hope the student model can capture the predictive trend of the teacher model.

This means our approach encourages the student model to learn more generalized and robust features, rather than overly focusing on mimicking every detail of the teacher model's output predictions. This might lead to better generalization performance in practical applications.

1.2. Analysis of γ

When we consider using the sigmoid function to simulate "soft ranking", the parameter γ plays a role in adjusting the slope of the sigmoid function. We can delve deeper into how this parameter affects SmaraAD.

Firstly, let's revisit the form of the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{4}$$

When z is very large, $\sigma(z)$ approaches 1; when z is very small (largely negative), $\sigma(z)$ approaches 0. Now, let's consider our "soft ranking" in terms of z:

$$z = \gamma(x - y) \tag{5}$$

where γ is a positive parameter. When γ is large:

- If $x > y$, then z becomes very large, thus $\sigma(z)$ is close to 1.
- If $x < y$, then z becomes very small (largely negative), thus $\sigma(z)$ is close to 0.

Hence, when γ is very large, the output of the sigmoid function approaches that of a traditional hard ranking. In Figure 3, we show the effect of r on Spearman based on soft ranking.

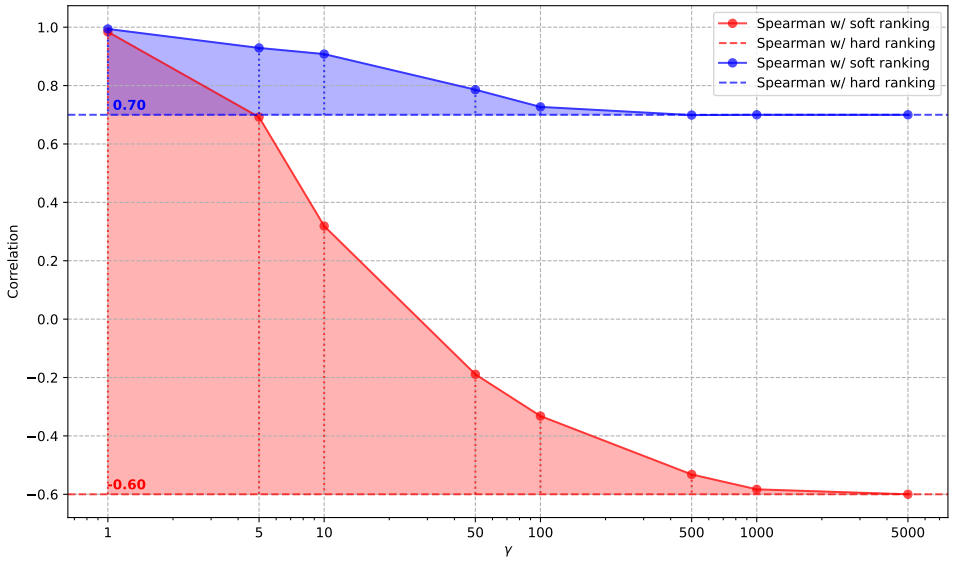


Figure 3. The figure illustrates the impact of the hyperparameter γ on two different Spearman correlation coefficient measurement methods. The red and blue colors represent two distinct sets of data. Solid lines indicate the Spearman correlation coefficients using soft ranking, while dashed lines represent the original Spearman correlation coefficients.

2. More Experiments

2.1. Details of Database

The CIFAR-10 and CIFAR-100 datasets are commonly employed for assessing adversarial robustness. CIFAR-10 comprises 50,000 training and 10,000 test images, spanning 10 distinct classes, each of size 32 × 32 pixels. CIFAR-100, while maintaining the same image counts and sizes as CIFAR-10, encompasses 100 different classes.

2.2. The Effect of Alleviating Robust Overfitting

Previous research [3] explored the relationship between robust overfitting and the weight loss landscape in Adversarial Training (AT) methods, revealing that a flatter weight loss landscape reduces the likelihood of robust overfitting during training. To further investigate the effectiveness of our proposed method, we plotted the weight loss landscapes for various AD and AT methods. As illustrated in Figure 4, our findings indicate that our method generates a flatter weight loss landscape, thereby mitigating robust overfitting. This aligns with the analysis of AD methods presented in [3].

2.3. Stronger Teacher

In this section, we employed a more powerful teacher model to validate the effectiveness of SmaraAD. The performance of the teacher model, as well as our method compared to other AD methods, is shown in Table 1. From the experimental results, it can be seen that when confronted with a stronger teacher, our approach further enhances the robustness of the student model, while other methods result in a decline in the performance of the student model.

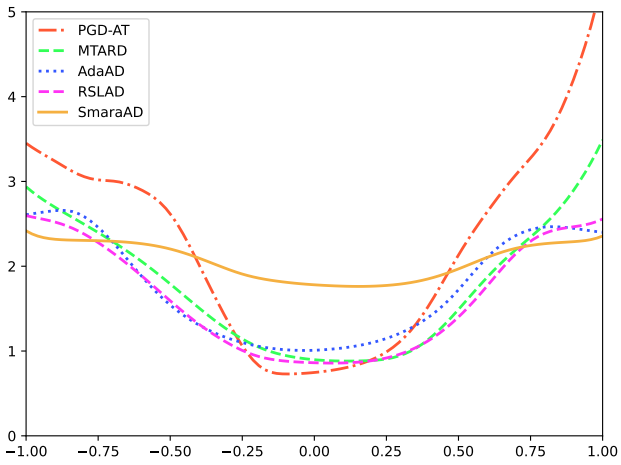


Figure 4. loss landscape for different methods.

Furthermore, we observed that, despite a significant improvement in the robustness of the teacher model, our method did not noticeably enhance the robustness of the student model. This is because the teacher model was trained with an additional dataset, while our method was trained on a standard dataset and did not utilize additional generated data. In real life, if the teaching materials used by the teacher and the student differ, it would similarly affect the effectiveness of the teacher's instruction. Therefore, during the adversarial distillation process, we also utilized the same generated data that was used during the training of the

Table 1. Performance of different AD methods under a more powerful teacher model. The maximum adversarial perturbation is $\epsilon = 8/255$. The best results are **boldfaced**, and the second best results are underlined.

Method	Clean	PGD	C&W	AA
Teacher	92.44%	70.06%	68.30%	67.12%
RSLAD	83.87%	51.18%	49.63%	47.68%
AdaAD	84.31%	51.88%	50.54%	48.63%
SmaraAD	87.37%	59.32%	58.89%	53.92%
SmaraAD (w/ generated data)	**89.35%**	**65.29%**	**64.63%**	**59.67%**

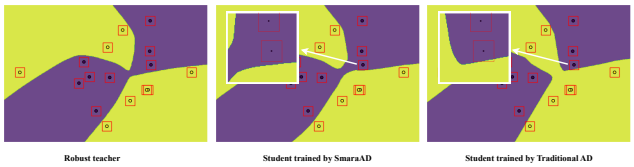


Figure 5. Decision boundaries for teacher models, student models trained by SmaraAD and student models trained by traditional AD.

teacher model. The experimental results show that when training with the same data as the teacher model, the robustness of the student model was further improved.

2.4. Decision Boundary of the Model

We explored whether the student model actually grasped the decision-making process of the teacher model during adversarial distillation. Figure 5 visualises the decision boundary for a toy problem. Here, the randomly generated training data are represented as coloured dots, while the boxes identify the desired robustness radius l_∞ . The background colour then represents the classification region of the network. If the box around a training point contains multiple colours, it indicates that the training point is vulnerable to attack. With these decision boundary maps, we observe that SmaraAD learns the decision-making process of teacher models more efficiently, whereas traditional AD methods are not as good at extracting robustness from robust teachers.

2.5. Comparison with the Method of Feature Distillation

In the field of knowledge distillation, feature alignment is a commonly used strategy [1, 2], aiming to transfer knowledge from the teacher model to the student model. Feature alignment focuses on matching the internal feature representations of the teacher and student models, typically achieved by minimizing the differences in their intermediate layer outputs. This method primarily targets the alignment of low to mid-level features, which may not fully capture the higher-level decision logic of the teacher model, and feature alignment might lead the student model to replicate some irrelevant or unnecessary features. In contrast, our proposed attribution region alignment focuses on the higher-level decision-making process, aiming to align the visual areas the model focuses on when making classification decisions. This not only facilitates the student model's absorption of knowledge at a higher level but also helps the model learn the attention distribution and decision patterns of the teacher model, thus more closely mirroring the high-level abstraction and decision logic of the teacher model. We compared our method with feature alignment, and the experimental results are shown in Table 2. The results demonstrate that our proposed attribution region alignment outperforms the feature alignment approach.

Table 2. Comparison with feature alignment.

Method	Clean	PGD	C&W	AA
Feature Alignment	85.53%	54.15%	53.72%	50.94%
Attribution Region Alignment	85.77%	55.70%	54.40%	52.36%

The advantage of attribution region alignment lies in its provision of a more direct and fine-grained approach to mimicking the decision-making process of the teacher model. By guiding the student model to focus on the same key areas as the teacher model, this method helps the student model not only replicate the outputs of the teacher model but also understand and mimic the mechanisms behind its decisions. This focus on high-level features and decision logic, especially in complex visual tasks, significantly enhances the generalization ability and performance of the student model. Moreover, by learning the attention mechanism of the teacher model, attribution region alignment also enhances the adaptability and robustness of the student model in dealing with novel and complex data. Therefore, although feature alignment remains effective in some scenarios, attribution region alignment demonstrates its unique superiority in simulating the advanced decision-making capabilities of the teacher model.

3. Pseudo-Code

The pseudo-code for our method is shown in this section.

Algorithm 1 Soft Rank

function SOFT_RANK(x, β)
  Initialize $soft\_ranks$ as an empty list
  for each element x_i in x **do**
    $rank \leftarrow \sum \text{sigmoid}(\beta \times (x - x_i))$
    Append $rank$ to $soft\_ranks$
  end for
  return $soft\_ranks$
end function

Algorithm 2 Slack Matching

function SLACK_MATCHING(x, y, β)
  $soft\_ranks\_x \leftarrow$ SOFT_RANK(x, β)
  $soft\_ranks\_y \leftarrow$ SOFT_RANK(y, β)
  $n \leftarrow$ length of x
  $d \leftarrow \sum$(square($soft\_ranks\_x - soft\_ranks\_y$))
  $\rho \leftarrow 1 - \frac{6 \times d}{n \times (n^2 - 1)}$
  return $-\rho$
end function

Algorithm 3 Inner Maximization

Input: Student model: $S(\cdot)$, Teacher model: $T(\cdot)$, Original input: x, True label: y, Perturbation size: ϵ, Step size :ξ, Number of iterations: $iter$
Output: Adversarial example: x_{adv}

$x_{adv} \leftarrow x +$ random noise in $[-\epsilon, \epsilon]$
for $i = 1$ **to** $iter$ **do**
  $loss_{kl} \leftarrow$ KL($S(x_{adv}), T(x_{adv})$)
  $loss_{mse} \leftarrow$ MSE(get_cam(S, x_{adv}), get_cam(T, x_{adv}))
  $loss \leftarrow loss_{kl} + loss_{mse}$
  $x_{adv} \leftarrow x_{adv} + \xi \cdot$ sign($\nabla_{x_{adv}} loss$)
  $x_{adv} \leftarrow$ clip($x_{adv}, x - \epsilon, x + \epsilon$) ▷ Ensure x_{adv} is within the ϵ-neighborhood of x
end for
return x_{adv}

Algorithm 4 Outer Minimization

Input: Training dataset: $\mathcal{D}$, student model with θ: $S(\cdot)$, teacher model: $T(\cdot)$, epochs E, learning rate: η, hyperparameter: α, constant: c
Output: Trained student model $S(\cdot)$

for $e = 1$ **to** E **do**
  for each batch (x, y) in $\mathcal{D}$ **do**
    $x' \leftarrow$ Inne_Maximization
    if weights are not used **then** ▷ SmaraAD
      $\mathcal{L}_{sm\&align}(S, T, x, x', c, \alpha) \leftarrow$ Calculated by Eq.9
    else ▷ SmaraAD++
      $\mathcal{L}_{outer}(S, T, x, x', y, c, \alpha) \leftarrow$ Calculated by Eq.10
    end if
    $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}$
  end for
end for
return $S(\cdot)$

References

[1] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7028–7036, 2021. 4

[2] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5008–5017, 2021. 4

[3] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020. 3