# Enhancing White-Box Attacks: Dynamic and Adaptive Threshold PGD for more effective Adversarial Examples

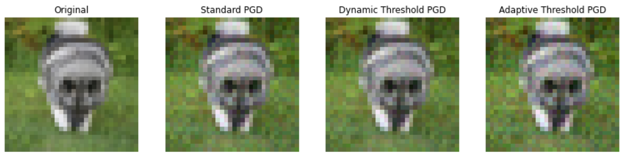
**Krishna Pranay Angara** [*] **Govind Vardhan Polnati** [*] **Vishnu Vardhan Koppera** [*]

## Abstract

The traditional Projected Gradient Descent (PGD) attack applies fixed-size perturbation and uses the gradient sign method to update the loss regarding the target label. This approach is ineffective in exploring input space thoroughly, getting trapped in local maxima. Our work discusses some alternate techniques for white-box attacks and addresses these limitations. We propose dynamic threshold PGD and adaptive threshold PGD through the addition of momentum and threshold to control perturbations and the inclusion of adapting threshold, potentially leading to more effective adversarial examples by exploiting a broader search space and adapting to the model's vulnerabilities during the attack process. We evaluate our attack's efficacy through state-of-the-art defense technique Margin aware instance reweighting learning (MAIL) mechanisms.

## 1. Introduction

Machine learning algorithms have developed significantly over the past decade transforming many areas such as computer vision, Natural language processing, malware detection, and robotics. Many applications now include machine learning to power their analysis and recognition systems. These algorithms must be robust and reliable. Many SOTA (state–of–the–art) models perform extremely well for standard data. However, recent studies have shown that ML algorithms can be easily fooled through simple adversarial examples. As stated by the authors (Szegedy et al., 2014), Machine learning algorithms are weak when it comes to classifying examples that are only slightly different than the examples the model trained on. Our SOTA training algorithms with varying architecture when trained on different subsets of training data sometimes misclassify the same adversarial example. (Goodfellow et al., 2015) Many researchers feel that adversarial example exposes fundamental blind spots in our training algorithms. The development of many defense techniques gave rise to newer and stronger attacks. These attacks typically are untargeted adversarial attacks aimed at maximizing the loss value. This is done through alteration within a small area surrounding the image space, usually by using $L_p$ norm (such as $L_\infty$ ). These small changes are imperceptible to the naked eye. However, the model fails to classify the image correctly by sometimes misclassifying the image with a wrong label or failing to identify any label at all. The attacks were further refined to enable the attacker to control and steer the direction of the attack. These are targeted attacks. Further attacks can be classified into two different types, White-box setting, where the attacker is aware of the network parameters and can easily tune the attack accordingly, and, Black-box setting – where the attacker relies on the knowledge of inputs and outputs and uses query-based methods or transfer attacks.


Original

Standard PGD

Dynamic Threshold PGD

Adaptive Threshold PGD

*Figure 1.* The sample image represents an example image attacked by our technique.The subtle noise induced by our technique makes the perturbation harder to notice with a naked eye.

Neural network, to be successfully fooled by an adversarial attack, small and meticulously crafted perturbations are to be applied to the input. When dealing with high dimensional images, this task of adding small perturbations becomes complex and this involves a non-convex optimization, I.e., the function to be minimized or maximized has multiple local minima or maxima which makes it challenging to find the global optimum.

Projected Gradient Descent (PGD) emerges as a universal adversary method and the standard approach for large-scale constrained optimization (Madry et al., 2019). PGD extends the basic gradient descent algorithm by incorporating a projection step that forcefully pulls the solution back into the feasible set (the set of points satisfying the constraints) whenever it steps out due to a gradient descent update. The methodological design to handle a non-convex optimization landscape makes the algorithm suitable in the adversarial setting operating in high dimensional spaces, PGD does not attempt to find global optimum, it only finds ways to effective local optima to meet the adversarial criteria by misleading the network while staying within plausible modifications of the original input, contained within epsilon-ball around the input.

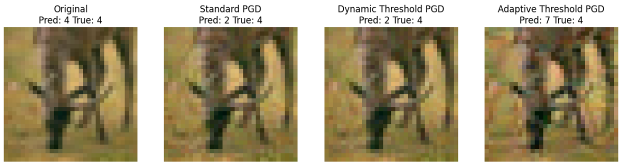


*Figure 2.* The misclassification of the image by the model after the attack can be seen in the figure. The image also represents the true label against predicted label.

The paper aims at improving the Projected Gradient descent technique through inclusion of momentum, threshold, and adaptive threshold to address the aforementioned limitation.The initial ideation of Dynamic threshold PGD where we incorporate adaptive step sizes based on the magnitude of the gradient introduces randomness in the perturbation process by applying the gradient update only when the threshold is met. This allows for more flexibility in adjusting the perturbation intensity based on the model's response, potentially leading to more effective adversarial examples. This is further refined to enhance adaptability by introducing Adaptive threshold PGD based on the loss magnitude and temperature scaling. It dynamically adjusts the threshold based on the loss, allowing for more aggressive exploration of the perturbation space when the loss is high and more conservative perturbations when the loss is low. Additionally, it introduces random noise based on scaled temperature, enhancing the unpredictability and robustness of the attack.Figure 1. demonstrates the perturbation when compared with the original image to understand the extent of noisification being applied. The changes are barely noticeable to the human eye. however, it has a prominent impact on the model accuracy.

We compare our methods with each other and standard PGD to understand the effect. We find that the novel attack has significantly higher efficacy when compared to the initial idea and standard PGD. We have used CIFAR-10 data along with ResNet18. The defense technique against these attacks is the MAIL method (Wang et al., 2022), which incorporates the probabilistic margin(PM) approach. PM approach focuses on boosting adversarial robustness by reweighting adversarial data during training, where data closer to the current decision boundaries are more critical and thus require larger weights. MAIL revealed its superiority against state-of-the-art methods, independent of adopted (basic) learning objectives.Figure 2. helps us understand how the model misclassified the deer image after the attacks, the changes are very minute when compared with the original image but the accuracy of the model has been compromised significantly.

Incorporating more sophisticated attacks such as dynamic threshold PGD and Adaptive threshold PGD, MAIL systems would be equipped to develop more robust models. Dynamic threshold PGD and Adaptive threshold PGD attack techniques dynamically adapt their attack parameters based on real-time feedback from the model, ensuring that generated adversarial examples are specifically tailored to exploit the most current vulnerabilities of the model. Employing these methods within MAIL also aids in benchmarking model robustness against complex and realistically varied adversarial attacks, thereby enhancing the credibility of security claims in practical AI deployments, and indicating the model's overall robustness.

## 2. Related works

Most attack algorithms concentrate on inducing misclassification in target classifiers. They do this by identifying adversarial perturbation and utilizing it to construct an example from clean data. This perturbation causes the clean example to be misclassified from its original class into another.

Given a fixed classifier with parameters $\theta$, a clean example $x_{\text{cln}}$ with its true label y, and a classification loss function $L()$, the constrained non-targeted adversarial perturbation $\delta_I$ is determined by optimizing the expression:

$$\max_{\delta_I} L(x_{\text{cln}} + \delta_I, y; \theta), \text{ subject to } \|\delta_I\|_p \leq \epsilon$$

where $\|\delta_I\|_p$ represents a specified $l_p$-norm distance metric, and $\epsilon$ denotes the adversarial perturbation constraint. In this work, we delve into the powerful PGD attack technique, known for its effectiveness in adversarial scenarios.

Projected Gradient Descent (PGD) adversarial attack was originally proposed by (Madry et al., 2019) as the strongest "first-order adversarial" attack. This approach initializes a perturbation $\delta$ from a predefined set S, which is then iteratively updated at each step size as:

$$\delta \leftarrow \Pi_S \left[\delta + \alpha \text{ sign} \left(\nabla_x L(x + \delta, y)\right)\right]$$

where $L()$ is a loss function, $x$ is the input to the model whose parameters are represented by $\theta$, $y$ is the target related to $x$, $\Pi_S$ is a projection operator with perturbation set $x + \delta$, and $\alpha$ is a gradient step size. The PGD attack relies on the sign of gradients because it aims to find an adversarial perturbation that maximizes the loss function concerning the input while constraining the perturbation within a specified range. By taking the sign of the gradients of the loss function concerning the input, the attack moves the input in the direction that increases the loss the most, thus generating an adversarial example. The sign of the gradients provides information about the direction of the steepest ascent in the loss landscape, guiding the attack towards finding perturbations that fool the model into making incorrect predictions.

In the paper (Chiang et al., 2020), they enhance the Projected Gradient Descent (PGD) attack by introducing a novel strategy of randomly chosen coordinate-wise step sizes. Unlike traditional methods that rely on fixed step sizes for each pixel in the gradient update, their approach independently selects random step sizes for each pixel. This randomization, coupled with initial random initialization, bolsters the exploration capacity of the optimization process, preventing it from becoming trapped in local minima or oscillating between fixed points. Termination of the algorithm occurs once the image classifier is successfully fooled. In this case, the applied adversarial perturbation is given as:

$$\delta \leftarrow \Pi_S \left[\delta + \tau \odot \text{sign}\left(\nabla_x L(x + \delta, class)\right)\right]$$

where $\tau$ is the step size array sampled with entries independent and uniformly distributed on the interval $U(0, 2a)$, $a$ is the expected step size, $L()$ is a loss function, $x$ is the input to the model whose parameters are represented by $\theta$, $class$ is the target class for a given $x$, $\Pi_S$ is a projection operator with perturbation set $x + \delta$. The step size array $\tau$ is multiplied into the gradient update using a Hadamard (i.e., coordinate-wise) product, denoted as $\odot$.

We have referenced the paper (Wang et al., 2022) to assess our attacks against robust defense methods. This paper introduces a novel approach to enhance adversarial robustness by adjusting the weights of adversarial data during training, giving greater importance to data near decision boundaries. The proposed method utilizes a probabilistic margin (PM) to measure this proximity, which is based on the multi-class margin in the probability space of model outputs. The paper explores various types of PMs with different geometric properties and presents a novel framework called Margin-based Adversarial Instance Reweighting Learning (MAIL). MAIL focuses on boosting adversarial robustness by reweighting adversarial data during training based on probabilistic margins. The proposed MAIL method (Wang et al., 2022) provides a versatile approach to reweighting adversarial data, allowing flexibility to integrate with existing methodologies. To illustrate, they offer three distinctive approaches:

**MAIL-AT:** Built upon the vanilla AT approach, MAIL-AT is determined by the learning objective:

$$-\sum_i \omega_i \log p_{y_i}(x_i + \delta_i^{(T)}; \theta)$$

This loss function minimizes the negative log-likelihood of the predicted labels given the perturbed inputs, penalizing deviations between the predicted labels and the ground truth labels.

**MAIL-TRADES:** MAIL-TRADES employs the Kullback-Leibler (KL) divergence to compare natural and adversarial predictions, incorporating a regularization term on natural prediction. The overall loss function is given as:

$$\beta \sum_i \omega_i \text{KL}(p(x_i + \delta_i^{(T)}; \theta)||p(x_i; \theta)) - \sum_i \log p_{y_i}(x_i; \theta)$$

where $\beta > 0$ represents the trade-off parameter, and $\text{KL}(p||q) = \sum_k p_k \log \frac{p_k}{q_k}$ denotes the KL divergence.

**MAIL-MART:** In MAIL-MART, the loss function consists of two components. The first component is the boosted cross-entropy (BCE) loss, which includes a common cross-entropy loss term and a regularization term to enhance prediction confidence. The second component is the misclassification aware KL (MKL) term, which reweights the KL divergence based on the estimated probability of correct prediction. The MAIL-MART loss function is given by:

$$\text{BCE}(x_i + \delta(T)_i, y_i; \theta) + \text{MKL}(x_i, \delta(T)_i; \theta)$$

where

$$\text{BCE} = -\log p_{y_i}(x_i + \delta(T)_i) - \log(1 - \max_{k \neq y_i} p_k(x_i + \delta(T)_i))$$

and

$$\text{MKL} = \text{KL}(p(x_i + \delta(T)_i; \theta)||p(x_i; \theta)) \times (1 - p_{y_i}(x_i; \theta))$$

The paper only assesses the performance of MAIL-AT and MAIL-TRADES against the Projected Gradient Descent (PGD) attack. However, we plan to evaluate our proposed attacks against all three defense mechanisms discussed above and analyze the outcomes for comparison.

## 3. Our Algorithms

### 3.1. Dynamic Threshold PGD

The Dynamic threshold PGD Attack is an advancement over the standard PGD attack, by introducing momentum to the perturbation updates. This allows the attack to maintain momentum in directions that lead to a reduction of the loss values, thereby enhancing its ability to overcome local minima and generate more effective adversarial attack. Initially, the momentum term $\mu$ is initialized as a zero tensor with shape matching the actual image. At each iteration, a random number is generated and if this random number exceeds a threshold, a perturbation based on the sign of the accumulated gradient is introduced to the actual image. The threshold serves as a hyper parameter governing the probability of adding perturbation into actual image. By randomly deciding whether to apply perturbations to each element of the input data based on a threshold, the attack introduces unpredictability and increases uncertainty for defense mechanisms.

**Algorithm 1** Dynamic Threshold PGD Attack

```
Require: Network $f$, input $x$, true labels $y$, permissible perturbation $\epsilon$, number of steps $n$, step size $\alpha$, loss function $L$, and momentum factor $\mu$.
  $g = 0$
  Generate $r \sim U(-\epsilon, \epsilon)$
  $x' = x + r$
  for $step = 1$ to $n$ do
    Reset optimizer gradients.
    $g = \mu \times g + \nabla_x L(f(x'; \theta), y)$
    if rand$(0, 1) > t$ then
      $\eta = \alpha \times \text{sign}(g)$
      $x' = x' + \eta$
      $\eta \leftarrow \Pi_\epsilon \left[x' - x\right]$
      $x' \leftarrow \Pi_{[0,1]} \left[x + \eta\right]$
    end if
  end for
```

### 3.2. Adaptive Threshold PGD:

The Adaptive threshold PGD attack takes a new approach to induce perturbation into an actual image. In addition to all the hyper parameters from the Dynamic threshold PGD attack, this algorithm includes a temperature and noise standard deviation to control the amount of perturbation added to the actual image. The threshold is calculated based on the magnitude of the loss function i.e., when the loss is high, the threshold decreases, allowing for more aggressive perturbations. The threshold is further scaled by a temperature factor to control the smoothness of the decision boundary between applying and not applying perturbations. A higher temperature reduces the threshold, intensifying the attack, while a lower temperature increases the threshold. The algorithm introduces a random tensor, ensuring it has the same shape as the input images. The scaled threshold is then compared element-wise with the random tensor to create a mask of 0 and 1. This step effectively determines which elements in the tensor should undergo random modification. Subsequently, a random noise tensor sampled from a normal distribution with mean 0 and standard deviation 1, is created and scaled by the noise standard deviation. The noise standard deviation hyper parameter controls the magnitude of the random noise added to perturbation. Finally, a perturbation using gradient and random noise is introduced to an actual image. The mask tensor containing binary values indicates where random modification should occur and ensures that noise is only added to specific elements of the actual image. This approach of perturbation makes it difficult for defense mechanisms to distinguish between actual and adversarial images. While this approach generates more potent adversarial images or examples, it also introduce additional computational overhead due to the increased complexity of the attack. However, the potential increase in the effectiveness of the generated adversarial examples could justify this additional computational cost.

**Algorithm 2** Adaptive Threshold PGD Attack

```
Require: Network $f$, input $x$, true labels $y$, permissible perturbation $\epsilon$, number of steps $n$, step size $\alpha$, loss function $L$, initial threshold $t_{\text{init}}$, temperature factor $T$, noise standard deviation $\sigma$, momentum factor $\mu$, random tensor r, and mask m.
  $x' = x$
  $g = 0$
  $t = t_{\text{init}}$
  for $step = 1$ to $n$ do
    Reset model gradients.
    $g = \mu \times g + \nabla_x L(f(x'; \theta), y)$
    Reset input gradients.
    $t = \max(t_{\text{init}} \cdot (1.0/(1.0 + \exp(-L))), 0.1)/T$
    $r \sim U(-\epsilon, \epsilon)$
    $m = \nVdash(r > t)$
    $\eta = \alpha \times \text{sign}(g) + m \times N(0, 1) \times \sigma$
    $x' = x' + \eta$
    $\eta \leftarrow \Pi_\epsilon \left[x' - x\right]$
    $x' \leftarrow \Pi_{[0,1]} \left[x + \eta\right]$
  end for
```

## 4. Experimentation

The attack methodologies are tested using the CIFAR-10 dataset against ResNet18 architecture of the authors (Wang et al., 2022) . The experimental setup along with the hyper parameters for attacks and defense methods are discussed below:

**Attack Setup** : Dynamic Threshold PGD is instantiated with an epsilon ($\epsilon$) of 0.03 , step_size ($\alpha$) of 0.003, and momentum of 0.7. Adaptive Threshold PGD is tuned using epsilon ($\epsilon$) of 0.03, step_size ($\alpha$) of 0.003, temperature of 0.5, initial threshold of 0.2 and standard noise of 0.0001. Initially, we have run the attacks with a iterations( number of steps) of 40 and 100 for dynamic PGD and 40 iterations for adaptive threshold PGD.

**Defense Setup:** The attacks were tested against MAIL-MART , MAIL-AT and MAIL-TRADES methods. For these defense methods, ResNet-18 network was trained using mini-batch gradient descent with momentum 0.9, batch size 128, and initial learning rate 0.01. Due to the time constraint we tested our attacks against the model trained for 30 epochs.

We have included Cross entropy loss(CE) and Carlini-

Wagner(CW) loss by the authors(Wang et al., 2022) for the adversarial attack assessment. Employing CE loss helps target the probability distribution of model output, while CW loss manipulates model prediction towards predetermined incorrect outputs. CW loss inclusion helps test model robustness against sophisticated and more nuanced adversarial inputs.

*Table 1.* Accuracies $\mathbf{A_{adv}}$ against attacks for the MAIL-MART method with the ResNet18 model on the CIFAR-10 dataset.

| Attack | $\mathbf{A_{adv}}$**CE** | $\mathbf{A_{adv}}$**CW** |
|---|---|---|
| 40-Step PGD | 49.92 % | 41.53 % |
| 40-Step Dynamic Threshold | 49.15 % | 40.93 % |
| 40-Step Adaptive Threshold | 42.22 % | 35.02 % |
| 100-Step PGD | 47.68 % | 39.49 % |
| 100-Step Dynamic Threshold | 47.52 % | 39.52 % |

*Table 2.* Accuracies $\mathbf{A_{adv}}$ against attacks for the MAIL-AT method with the ResNet18 model on the CIFAR-10 dataset.

| Attack | $\mathbf{A_{adv}}$**CE** | $\mathbf{A_{adv}}$**CW** |
|---|---|---|
| 40-Step PGD | 44.91 % | 42.11 % |
| 40-Step Dynamic Threshold | 44.19 % | 41.39 % |
| 40-Step Adaptive Threshold | 37.47 % | 35.32 % |
| 100-Step PGD | 43.73 % | 41.02 % |
| 100-Step Dynamic Threshold | 43.71 % | 40.93 % |

*Table 3.* Accuracies $\mathbf{A_{adv}}$ against attacks for the MAIL-TRADES method with the ResNet18 model on the CIFAR-10 dataset.

| Attack | $\mathbf{A_{adv}}$**CE** | $\mathbf{A_{adv}}$**CW** |
|---|---|---|
| 40-Step PGD | 43.45 % | 40.64 % |
| 40-Step Dynamic Threshold | 42.8 % | 40.26 % |
| 40-Step Adaptive Threshold | 35.2 % | 32.7 % |
| 100-Step PGD | 42.5 % | 39.93 % |
| 100-Step Dynamic Threshold | 42.49 % | 39.89 % |

The consistency in the reduced accuracy for the dynamic threshold PGD from Table1., Table2. and Table3. is evidence for their attack potencies. The momentum-based gradient update and the gradient-based perturbation direction help overcome local minima and more effective exploration of the adversarial landscape.

The Adaptive Threshold PGD significantly reduces the model accuracy through a more strategic perturbation application focusing on randomization and more realistic noise inclusion. Adaptive threshold PGD offers a more nuanced adversarial exploration is made possible by adding temperature and noise adjustments modulating the intensity of perturbations.

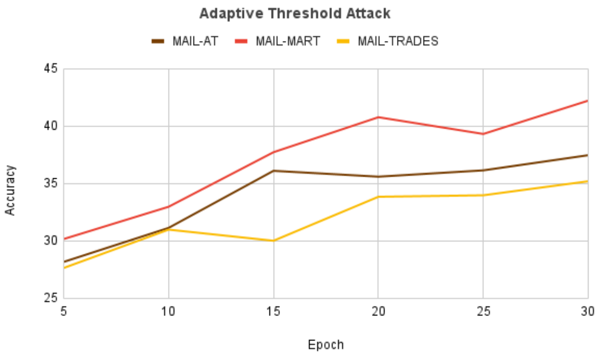


*Figure 3.* Performance (Model Accuracy after the attack) of 40-Step Adaptive Threshold Attack against the defense methods for different epochs.

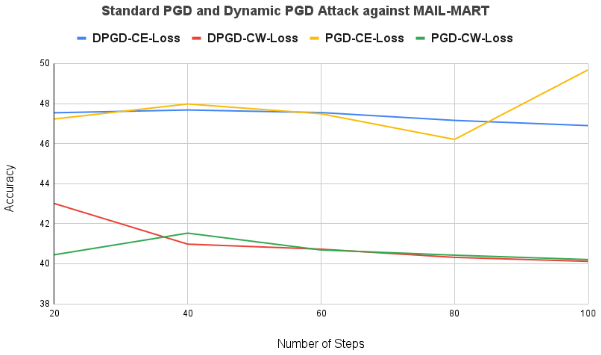


*Figure 4.* Loss trends of Standard PGD w.r.t Dynamic PGD attack over 100 iterations for 30 epochs.

The plot in Figure 3. help showcase the reduction of model's accuracy across epochs for the three defence methods. Adaptive threshold has performed well on all the three attacks, but MAIL-AT and MAIL-TRADES are more vulnerable to this attack compared to MAIL-MART.

The plot in Figure 4. helps us understand the trends in CW loss and cross entropy loss for both standard PGD and dynamic threshold PGD for 30 epochs. Over 100 iterations we observe that dynamic PGD is performing better when compared with standard PGD.

The results show a significant improvement in the attack accuracy, despite the inclusion of state-of-the-art defense techniques like Probabilistic margin(PM) that weigh adversarial data during AT. Margin-aware instance reweighting learning (MAIL) employs various existing reliable AT methods like MART and TRADES, incorporating their loss functions and perturbation control. Our methodologies present a formidable challenge to the adversarially trained models of MAIL-AT, MAIL-TRADES and MAIL-MART. Incorporating the MAIL helped us understand the attack performance in more reliable settings. Analyzing the metrics of robust accuracy and convergence helped us understand the computational limitations associated with the attack techniques.

## 5. Conclusion

Our approach offers a unique way of creating adversarial examples. The metrics showcase substantial improvements in fooling the model under the MAIL-AT, MAIL-MART, and MAIL-TRADES methodologies. The project emphasizes identifying a technique to perturbate the image by including randomness in the existing PGD model. We have employed two ways of attacks, The first is through the inclusion of momentum and threshold to control the perturbation quantity and also to remove the limitation of local maxima of PGD. The second is to employ randomness through various hyper parameters such as temperature, noise standard deviation, and scaled threshold, where, the threshold is constantly adapting for every iteration. Our algorithms have been tested against the reliable existing adversarial methodologies and empirical results indicate that our attacks yield promising outcomes compared to the standard PGD.

## 6. Future work

Future work includes optimizing the techniques for computational efficiency and comprehensive testing scenarios to improve the attack robustness. We intend to test the adaptive threshold for 100 step iterations as well. In addition, we want to check how well the Adversarial examples crafted using Dynamic threshold PGD and Adaptive dynamic threshold PGD on one model performs against another model (transferability), with different architectures and training data.

## References

Chiang, P.-Y., Geiping, J., Goldblum, M., Goldstein, T., Ni, R., Reich, S., and Shafahi, A. Witchcraft: Efficient pgd attacks with random step size. 2020. URL https://arxiv.org/pdf/1911.07989.pdf.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. 2015. URL https://arxiv.org/pdf/1412.6572.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. 2019. URL https://arxiv.org/pdf/1706.06083.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. 2014. URL https://arxiv.org/pdf/1312.6199.

Wang, Q., Liu, F., Han, B., Liu, T., Gong, C., Niu, G., Zhou, M., and Sugiyama, M. Probabilistic margins for instance reweighting in adversarial training. 2022. URL https://arxiv.org/pdf/2106.07904.pdf.