A Proof of Theorem 1

Considering $||\epsilon||_2$ or $||\epsilon||_\infty$ is usually very small for adversarial examples, we utilize Taylor Expansion for x as the approximation for the adversarial loss $\mathcal{L}(x+\epsilon;\theta)$, such that:

$$\mathcal{L}(x+\epsilon;\theta) = \mathcal{L}(x;\theta) + (\frac{\partial \mathcal{L}(x;\theta)}{\partial x})^T \epsilon + \mathcal{O}(\|\epsilon\|^2) \tag{5}$$

To derive an upper bound on the gradient conflict in the regime that $\|\epsilon\|$ gets small, we will only consider the first-order term above. We then take the derivative of both sides of the equation with respect to θ to obtain:

$$g_{\text{a}} = g_{\text{c}} + \frac{\partial^2 \mathcal{L}(x;\theta)}{\partial x \partial \theta}\epsilon = g_{\text{c}} + \frac{\partial g_{\text{c}}}{\partial x}\epsilon = g_{\text{c}} + H\epsilon \tag{6}$$

where $H = \frac{\partial g_{\text{c}}}{\partial x} \in \mathbb{R}^{d_\theta \times d_x}$. d_θ/d_x denotes the dimension of parameter θ and input data x. By multiplying g_{a}^T and g_{c}^T on the two sides of Eq. (6), respectively, we can obtain Eq. (7) and Eq. (8) as follows.

$$g_{\text{c}}^T g_{\text{a}} = ||g_{\text{c}}||_2^2 + g_{\text{c}}^T H\epsilon \tag{7}$$

$$||g_{\text{a}}||_2^2 = g_{\text{a}}^T g_{\text{c}} + g_{\text{a}}^T H\epsilon \tag{8}$$

Eq. (7) minus Eq. (8):

$$g_{\text{c}}^T g_{\text{a}} = \frac{||g_{\text{a}}||_2^2 + ||g_{\text{c}}||_2^2 + \epsilon^T H^T (g_{\text{c}} - g_{\text{a}})}{2} \tag{9}$$

Based on Eq. (6), we can replace $(g_{\text{c}} - g_{\text{a}})$ as $H\epsilon$:

$$g_{\text{c}}^T g_{\text{a}} = \frac{||g_{\text{a}}||_2^2 + ||g_{\text{c}}||_2^2 - \epsilon^T H^T H\epsilon}{2} \tag{10}$$

Recall the definition of μ as $\mu = ||g_{\text{c}}||_2 \cdot ||g_{\text{a}}||_2 \cdot (1 - \cos(g_{\text{c}}, g_{\text{a}}))$

$$\begin{aligned}
\mu &= ||g_{\text{c}}||_2 \cdot ||g_{\text{a}}||_2 \cdot (1 - \cos(g_{\text{c}}, g_{\text{a}})) \\
&= ||g_{\text{c}}||_2 \cdot ||g_{\text{a}}||_2 - g_{\text{c}}^T g_{\text{a}} \\
&= \frac{2||g_{\text{c}}||_2 \cdot ||g_{\text{a}}||_2 - ||g_{\text{a}}||_2^2 - ||g_{\text{c}}||_2^2 + \epsilon^T H^T H\epsilon}{2} \quad \text{(Use Eq. (10))} \\
&= \frac{\epsilon^T \mathcal{K}(\theta, x)\epsilon - (||g_{\text{c}}||_2 - ||g_{\text{a}}||_2)^2}{2} \leq \frac{\epsilon^T \mathcal{K}(\theta, x)\epsilon}{2} \leq \frac{\lambda_{max}\epsilon^T\epsilon}{2}
\end{aligned} \tag{11}$$

where $\mathcal{K}(\theta, x) = H^T H$ is a symmetric and positive semi-definite matrix, and λ_{max} is the largest eigenvalue of K, where $\lambda_{max} \geq 0$.

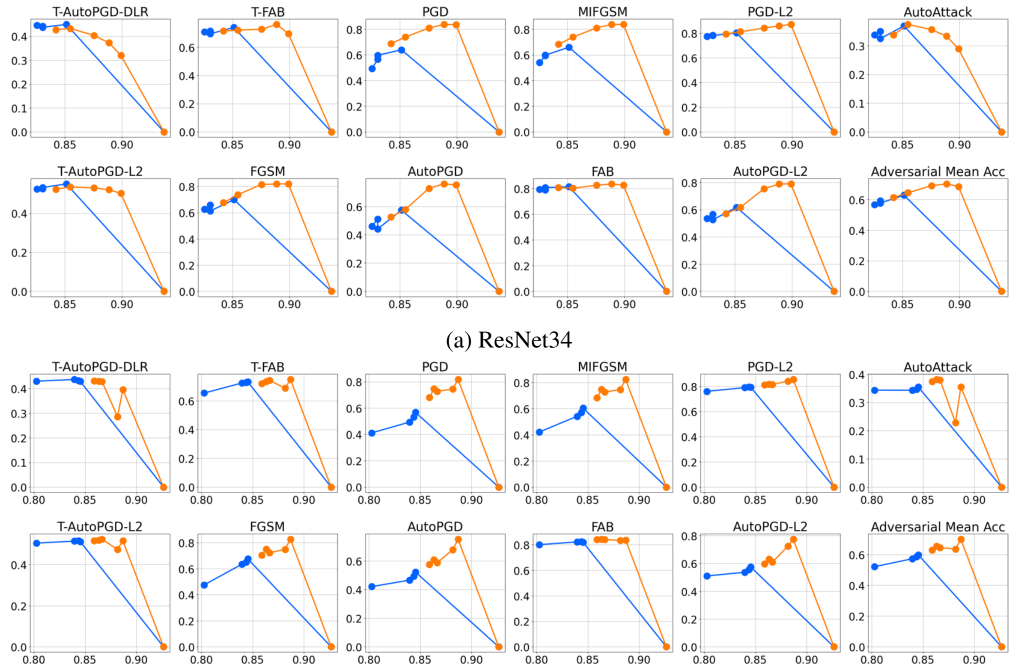
Considering two widely-used restrictions for perturbation ϵ applied in adversarial examples as l_2 and l_∞ norm, we have:

- For $||\epsilon||_2 \leq \delta$, where $\mu \leq \frac{1}{2}\lambda_{max}\delta^2$. The upper bound of μ is $\mathcal{O}(\delta^2)$.
- For $||\epsilon||_\infty \leq \delta$, it implies that the absolute value of each element of ϵ is bounded by δ, where $\epsilon^T\epsilon = \sum_{i=0}^d \epsilon_i^2 \leq d^2\delta^2$. The upper bound of μ is $\mathcal{O}(d^2\delta^2)$.

B Analytical Solution for the Inner Maximization

We introduce the details about how to get the analytical inner-max solution (Eq. (3)) for our synthetic experiment presented in Section 3. As we introduced in Section 3, consider a linear model as $f(x) = w^T x + b$ under a binary classification task where $y \in \{+1, -1\}$. The predicted probability of sample x with respect to its ground truth y can be defined as:

$$p(y|x) = \frac{1}{1 + \exp(-y \cdot f(x))} \tag{12}$$



(a) ResNet34

(b) WRN-28-10

Figure 9: SA-AA Fronts for Adversarial Training from Scratch on CIFAR10 with ResNet34 and WRN-28-10.

	Standard	T-AutoPGD-DLR	T-AutoPGD-L2	T-FAB	FGSM	PGD	AutoPGD	MIFGSM	FAB	PGD-L2	AutoPGD-L2	AutoAttack	Adversarial Mean Acc
$\gamma = 0.8$, PGD	**0.8659**	0.4004	0.5356	0.6861	**0.7649**	**0.7442**	0.6301	**0.7419**	**0.8177**	**0.8211**	0.67	**0.3517**	**0.6512**
$\gamma = 0.8$, PGD-DLR	0.8646	**0.4147**	**0.539**	**0.7168**	0.7452	0.672	**0.6429**	0.6864	0.8001	0.8196	**0.6919**	0.3260	0.6413
$\gamma = 0.9$, PGD	**0.9009**	0.2844	0.5075	**0.6986**	0.7781	0.7021	**0.6624**	0.7251	**0.8371**	**0.8588**	**0.7267**	0.2472	0.6389
$\gamma = 0.9$, PGD-DLR	0.8923	**0.3794**	**0.5353**	0.6874	**0.779**	**0.7207**	0.6428	**0.7315**	0.8229	0.8488	0.7038	**0.2992**	**0.6501**

Table 2: Evaluation results for CA-AT for using different inner maximization solver (PGD/PGD-DLR) during the process of AT.

Then, the BCE loss function for sample x can be formulated as:

$$\mathcal{L}(f(x), y) = -\log(p(y|x)) = \log(1 + \exp(-y \cdot f(x))) \tag{13}$$

Consider the perturbation ϵ under the restriction of L_∞ norm, the adversarial attack for such a linear model can be formulated as an inner maximization problem as Eq. (14).

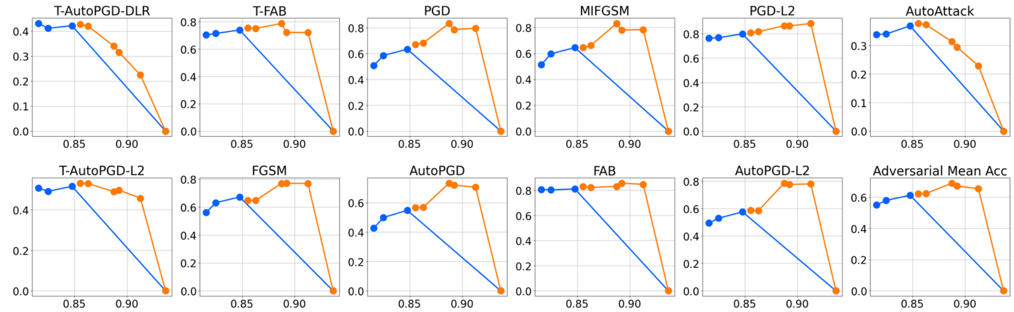
$$\max_{\|\epsilon\|_\infty \leq \delta} \log(1 + \exp(-y \cdot f(x + \epsilon))) \equiv \min_{\|\epsilon\|_\infty \leq \delta} y \cdot w^T \epsilon \tag{14}$$

Consider the case that $y = +1$, where the L_∞ norm says that each element in ϵ must have magnitude less than or equal δ, we clearly minimize this quantity when we set $\epsilon_i = -\delta$ for $w_i \geq 0$ and $\epsilon_i = \delta$ for $w_i < 0$. For $y = -1$, we would just flip these quantities. That is, the optimal solution ϵ^* to the above optimization problem for the L_∞ norm is expressed as Eq. (15).

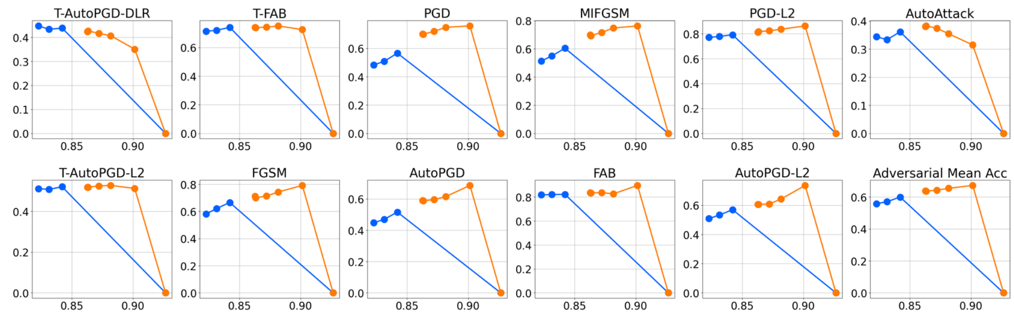
$$\epsilon^* = -y \cdot \delta \odot \text{sign}(w) \tag{15}$$

where $\odot$ is the element-wise multiplication. Based on Eq. (15), we can formulate the adversarial loss as follows, which is as same as the adversarial loss presented in Eq. (3).

$$\begin{aligned} \mathcal{L}(f(x + \epsilon^*), y) &= \log(1 + \exp(-y \cdot w^T x - y \cdot b - y \cdot w^T \epsilon^*)) \\ &= \log(1 + \exp(-y \cdot f(x) + \delta \|w\|_1)) \end{aligned} \tag{16}$$

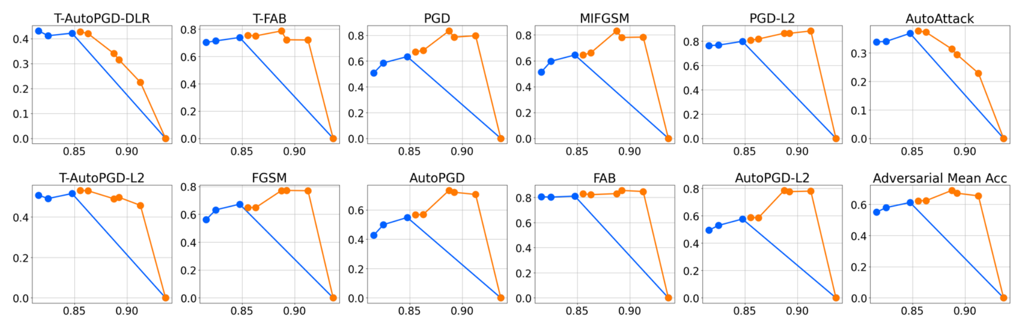


(a) ResNet34

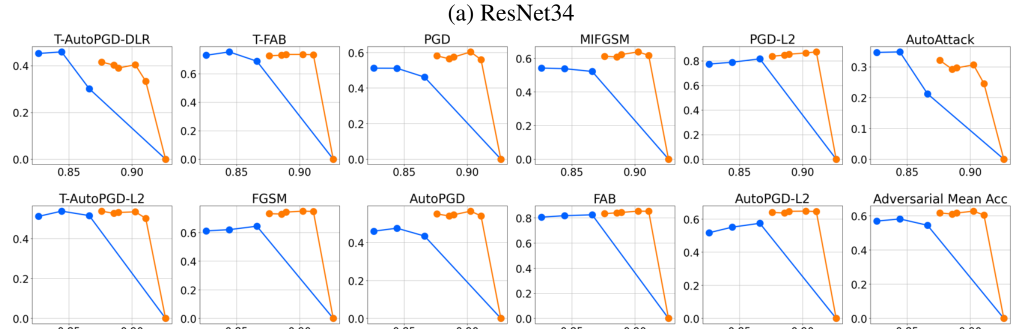


(b) WRN-28-10

Figure 10: SA-AA Fronts on CIFAR10 for Adversarial Training from Scratch using TRADES with ResNet34 and WRN-28-10.



(a) ResNet34



(b) WRN-28-10

Figure 11: SA-AA Fronts for Adversarial Training from Scratch on CIFAR10 using CLP with ResNet34 and WRN-28-10.

		Standard Accuracy	DDN Attack	C&W Attack	Square Attack
ResNet18	Standard Training	0.9392	0.133	0.1171	0.6795
	Vanilla AT, $\lambda = 0.5$	0.8239	0.4585	0.4565	0.7329
	CA-AT, $\gamma = 0.8$	**0.8659**	**0.5991**	**0.5089**	**0.7656**
ResNet 34	Standard Training	0.9363	0.1036	0.088	0.6771
	Vanilla AT, $\lambda = 0.5$	0.8305	0.5411	0.4747	0.7429
	CA-AT, $\gamma = 0.8$	**0.8753**	**0.7207**	**0.4885**	**0.7934**

Table 3: The comparison Results against Black-box Attack (Square Attack) and Optimization-based Attack (C&W Attack and DDN Attack) between Vanilla AT and CA-AT on CIFAR10.

C Additional Experimental Results

Experimental setup on adversarial PEFT. For the experiments on adversarial PEFT, we leverage the adversarially pretrained Swin-T and ViT downloaded from ARES[1]. For adapter, we implement it as (Pfeiffer et al., 2020) by inserting an adapter module subsequent to the MLP block at each layer with a reduction factor of 8.

The effect of inner maximization solver in AT. In Table 2, we conduct the ablation study for using different attack methods to generate adversarial samples during adversarial training from scratch. We find that PGD-DLR can achieve higher adversarial accuracies when $\gamma = 0.9$ but lead them worse when $\gamma = 0.8$ but not significant. We conclude that the effect of the inner maximization solver, as well as the adversarial attack method during AT, does not dominate the performance of CA-AT.

Results for different model architectures. For different model architectures such as ResNet34 and WRN-28-10, their SA-AA front on CIFAR10 and and CIFAR100 with different adversarial loss functions are shown in Fig. 9, Fig. 11, and Fig. 10. All of those figures demonstrate CA-AT can consistently surpass Vanilla AT across different model architectures.

Results for L_2-based adversarial attacks with different budgets. Besides evaluating the adversarial accuray on L_∞-based attacks with different budgets (Table 1), we also evaluate the adversarial robustness against L_2-based adversarial attacks with different budgets ($||\epsilon||_2 = [0.5, 1, 1.5, 2]$), which is shown in Table 4.

Results for Black-box Attack & Optimization-based Attack. To further evaluate the robustness of CA-AT against optimization-based attacks, and also demonstrate that the performance gain of adversarial accuracy is not brought by obfuscated gradients (Athalye et al., 2018), we evaluate the adversarial robustness via black-box attack (Square Andriushchenko et al. (2020)) and optimization-based attack (C&W Carlini & Wagner (2017), DDN Rony et al. (2019)). Table 3 shows that CA-AT ($\gamma = 0.8$) outperforms Vanilla AT ($\lambda = 0.5$) on defending against the both black-box attack and optimization-based attack, while achieving higher standard accuracy.

The Degraded Version of CA-AT. To rigorously demonstrate that projecting adversarial graodient g_a into the cone of g_c can boost both standard and adversarial accuracy, we conduct an ablation study by using traditional λ-weighted mean of g_a and g_c when $\phi \leq \gamma$ and only g_c when $\phi > \gamma$. As shown in Algorithm 2, we named such an ablated version of CA-AT as **CA-AT-AV**. The comparsion results shown in Figure 12 for Vanilla AT, CA-AT and CA-AT-DV demonstrate that the boost of tradeoff between standard accuracy and adversarial accuracy.

Ablation Study on Learning Rate and Batch Size. We conducted ablation study on different training parameters such as learning rate and batch size in Fig. 13. The observation is, although batch size and learning rate effect the standard accuracy and adversarial accuracies aganist various attacks, CA-AT can consistently lead to better standard performance and adversarial robustnessn accorss different batch size and learning rate.

Experimental Results for Larger Dataset. We also evaluated our method on Tiny-ImageNet. Results presented in Table 5 demonstrate the superiority of CA-AT on large-scale dataset.

More Advanced Adversarial Loss.,Besides TRADES and CLP, we conducted more experiments on MART shown in Table 6.

[1]https://github.com/thu-ml/ares

Algorithm 2 CA-AT (Ablated Version)

Input: Training dataset D, Loss function $\mathcal{L}$, Perturbation budget δ, Training epochs N, Initial model parameter θ_1, Projection margin threshold γ, learning rate lr, Trade-off Factor λ
Output: Trained model parameter θ_{N+1}

for $t = 1$ to N **do**
  for each batch B in D **do**
    $\mathcal{L}_\text{c} = \frac{1}{|B|}\sum_{(x,y)\in B}\mathcal{L}(x, y; \theta_t)$
    $\mathcal{L}_\text{a} = \frac{1}{|B|}\sum_{(x,y)\in B}\max_{||\epsilon||_\infty\leq\delta}\mathcal{L}(x+\epsilon, y; \theta_t)$
    $g_\text{c}, g_\text{a} = \nabla_{\theta_t}\mathcal{L}_\text{c}, \nabla_{\theta_t}\mathcal{L}_\text{a}$
    $\phi = \cos(g_\text{c}, g_\text{a})$
    if $\phi < \gamma$ **then**
      $g_* = \lambda g_\text{a} + (1-\lambda) g_\text{c}$ $\triangleright$ ϵ Averaging g_a and g_c instead of projection
    else
      $g_* = g_\text{c}$
    end if
    $\theta_t = \theta_t - lr * g_*$
  end for
  $\theta_{t+1} = \theta_t$
end for

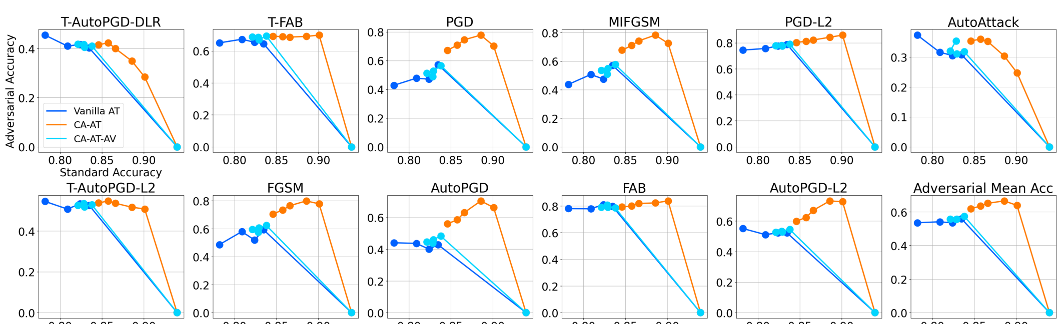
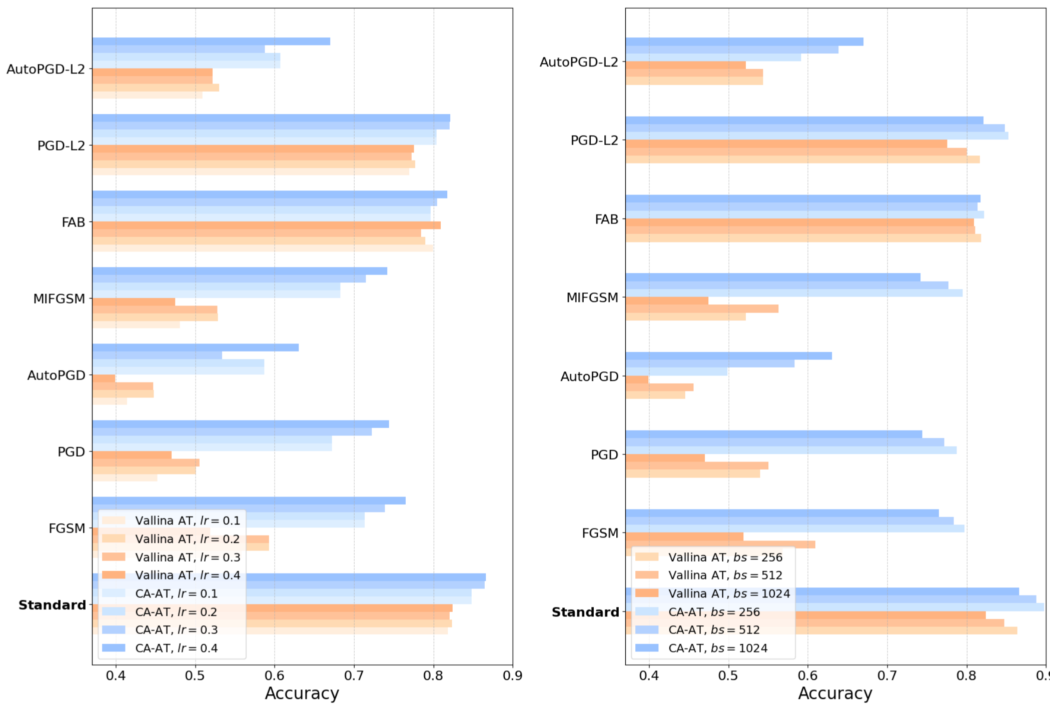


Figure 12: Results for Ablation Study on CA-AT for ResNet18 on CIFAR10.

		PGD-L2		AutoPGD-L2		T-AutoPGD-L2	
	$p = 2$	$\gamma = 0.8$	$\lambda = 0.5$	$\gamma = 0.8$	$\lambda = 0.5$	$\gamma = 0.8$	$\lambda = 0.5$
ResNet18	0.5	**0.8211**	0.7759	**0.67**	0.5222	**0.5356**	0.5327
	1	**0.8207**	0.7748	**0.603**	0.3036	0.261	**0.2762**
	1.5	**0.8194**	0.7738	**0.5652**	0.2405	**0.1483**	0.1428
	2	**0.8187**	0.7734	**0.5331**	0.2115	**0.0904**	0.088
		PGD-L2		AutoPGD-L2		T-AutoPGD-L2	
	$p = 2$	$\gamma = 0.8$	$\lambda = 0.5$	$\gamma = 0.8$	$\lambda = 0.5$	$\gamma = 0.8$	$\lambda = 0.5$
ResNet34	0.5	**0.8411**	0.78	**0.7534**	0.5255	**0.5301**	**0.5325**
	1	**0.8412**	0.7791	**0.7249**	0.3806	0.2571	**0.2683**
	1.5	**0.8403**	0.7784	**0.7022**	0.3462	**0.1446**	0.1438
	2	**0.8386**	0.7781	**0.679**	0.3196	0.0899	**0.0905**

Table 4: Evaluation Results for CA-AT ($\gamma = 0.8$) and vanilla AT ($\lambda = 0.5$) across different L_2-based attacks with various restriction θ.



(a) Different Learning Rate

(b) Different Batch Size

Figure 13: Ablation Study for Different Training Hyperparameters including Learning Rate and Batch Size.

	Standard	FGSM	PGD	AutoPGD	MIFGSM	T-FAB
Vanilla AT, $\lambda = 0.5$	0.4881	0.1872	0.152	0.1615	0.16	0.347
CA-AT, $\gamma = 0.8$	**0.4989**	**0.254**	**0.1867**	**0.1753**	**0.2044**	**0.3584**

Table 5: Results for Training PreActResNet18 on TinyImageNet

		Standard	T-AutoPGD	T-FAB	FGSM	PGD	AutoPGD	MIFGSM	FAB
ResNet 18	Vanilla AT, $\lambda = 0.5$	0.83	0.5344	0.657	0.6017	0.5343	0.2666	0.4483	0.55
	CA-AT, $\gamma = 0.8$	**0.8848**	**0.5381**	**0.6826**	**0.7953**	**0.782**	**0.669**	**0.7859**	**0.8151**
ResNet 34	Vanilla AT, $\lambda = 0.5$	0.82	0.3624	0.503	0.4832	0.3466	0.2479	0.3359	0.6284
	CA-AT, $\gamma = 0.8$	**0.8857**	**0.4922**	**0.7357**	**0.8297**	**0.8466**	**0.7687**	**0.8466**	**0.8299**

Table 6: Results for Training with MART loss on CIFAR10 with ResNet18