Appendix

A Threat Model and Adversarial Sample

In this section, we summarize essential terminologies of adversarial settings related to our work. We first define a threat model, which consists of a set of assumptions about the adversary. Then, we describe the generation mechanism of adversarial samples in AT frameworks for the threat model defending against adversarial attacks.

A.1 Threat Model

Adversarial perturbation was firstly discovered by [9], and it instantly strikes an array of studies in both adversarial attack and adversarial robustness. [7] specifies a threat model for evaluating a defense method including a set of assumptions about the adversary's goals, capabilities, and knowledge, which are briefly delineated as follows:

- Adversary's goals could be either simply deceiving a model to make the wrong prediction to any classes from a perturbed input or making the model misclassify a specific class to an intended class. They are known as *untargeted* and *targeted* modes, respectively.
- Adversary's capabilities define reasonable constraints imposed on the attackers. For instance, a L_p certified robust model is determined with the worst-case loss function $\mathcal{L}$ for a given perturbation budget ϵ:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\tilde{x}\in B_p(x,\epsilon)}\mathcal{L}(f(\tilde{x}),y)\right], \tag{13}$$

  where $B_p(x,\epsilon)=\{u\in\mathbb{R}^{\mathcal{I}}:||u-x||_p\leq\epsilon\}$.
- Adversary's knowledge indicates what knowledge of the threat model that an attacker is assumed to have. Typically, *white-box* and *black-box* attacks are two most popular scenarios studied. The white-box settings assume that attackers have full knowledge of the model's parameters and its defensive scheme. In contrast, the black-box settings have varying degrees of access to the model's parameter or the defense.

Bearing these assumptions about the adversary, we describe how a defense model generates adversarial samples for its training in the following section.

A.2 Adversarial Sample in AT

Among multiple attempts to defend against adversarial perturbed samples, adversarial training (AT) is known as the most successful defense method. In fact, AT is an data-augmenting training method that originates from the work of [20],

where crafted adversarial samples are created by the fast gradient sign method (FGSM), and mixed into the mini-batch training data. Subsequently, a wide range of studies focus on developing powerful attacks [8, 11, 16, 24]. Meanwhile, in the opposite direction to the adversarial attack, there are also several attempts to resist against adversarial examples [22, 35, 40]. In general, a defense model is optimized by solving a minimax problem:

$$\min_{\theta} \Big[\max_{\tilde{x} \in B_p(x,\epsilon)} \mathcal{L}_{\mathcal{XE}}(\tilde{x}, y; \theta) \Big], \tag{14}$$

where the inner maximization tries to successfully create perturbation samples subjected to an ϵ-radius ball B around the clean sample x in L_p space. The outer minimization tries to adjust the model's parameters to minimize the loss caused by the inner attacks. Among existing defensive AT, PGD-AT [26] becomes the most popular one, in which the inner maximization is approximated by the multi-step projected gradient (PGD) method:

$$\tilde{x}_{t+1} = \Pi_x^{\epsilon}(\tilde{x}_t + \eta \cdot \mathrm{sgn}(\nabla_{\tilde{x}_t} \mathcal{L}(\tilde{x}_t, y))), \tag{15}$$

where Π_x^{ϵ} is an operator that projects its input into the feasible region $B_\infty(x, \epsilon)$, and $\eta \in \mathbb{R}$ is called step size. The loss function in Eq. 15 can be modulated to derive different variants of generation mechanism for adversarial samples in AT. For example, Zhang *et al.* [41] utilizes the loss between the likelihood of clean and adversarial samples for updating the adversarial samples. In our work, we use Eq. 15 as our generation mechanism for our AT framework.

B Training Algorithm for OTJR

Our end-to-end algorithm for optimizing Eq. 12 is provided in Algorithm 1. As mentioned, in practice, deep learning libraries allow for the simultaneous sampling of K uniform vectors, denoted as $\hat{v}_k$. Consequently, the computation of random projections and the determination of optimal movement steps can be effectively vectorized and executed concurrently.

C Sanity Tests

The phenomenon of gradient obfuscation arises, when a defense method is tailored such that its gradients prove ineffective for generating adversarial samples [2]. However, the method designed in that manner can be an incomplete defense to adversarial examples [2]. Adhering to guidelines from [7], we evaluate our pre-trained model on CIFAR10 with WRN34 to affirm our proposed OTJR does not lean on gradient obfuscation. As detailed in Table 9, iterative attacks are strictly more powerful than single-step attacks, whereas when increasing perturbation budget ϵ can also raise attack successful rate. Finally, the PGD attack attains a 100% success rate when $\epsilon = 128/255$.

Algorithm 1 OTJR: AT with SW and optimal Jacobian regularization

Require: DNN f parameterized by θ, training dataset $\mathcal{D}$. Number of projection K. Maximum perturbation ϵ, step size η, number of adversarial iteration P. Loss' hyper-parameters λ_J and λ_{SW}. Learning rate α and a mini-batch size of $\mathcal{B}$.

while not converged **do**
  for $\{(x_i,\ y_i)\}_{\mathcal{B}} \in \mathcal{D}$ **do**
    $\nu := z_i = f_\theta(x_i)|_{i=1,..,\mathcal{B}}$ ▷ *forward a batch of clean samples through the model*
    for iteration $t \leftarrow 1$ to P **do**
      $\tilde{x_i} = \Pi_x^\epsilon(\tilde{x_i} + \eta \cdot \text{sgn}(\nabla_{\tilde{x_i}} \mathcal{L}(\tilde{x_i}, y_i)))|_{i=1,..,\mathcal{B}}$ ▷ *generate adv. samples by L_∞-PGD in P iterations*
    end for
    $\mu := \tilde{z_i} = f_\theta(\tilde{x_i})|_{i=1,..,\mathcal{B}}$ ▷ *forward a batch of adv. samples through the model*
    $SW \leftarrow 0$ ▷ *initialize SW loss*
    $\sigma_i \leftarrow 0|_{i=1,..,\mathcal{B}}$ ▷ *initialize $\mathcal{B}$ Jacobian projections*
    for iteration $k \leftarrow 1$ to K **do**
      $\hat{v}_k \leftarrow \mathcal{U}(\mathcal{S}^{C-1})$ ▷ *uniformly sample $\hat{v}_k$ from $\mathcal{S}^{C-1}$*
      $SW \leftarrow SW + \psi\left(\tau_1 \circ \mathcal{R}_{\hat{v}_k}\mu, \tau_2 \circ \mathcal{R}_{\hat{v}_k}\nu\right)$ ▷ *add SW under projection $\hat{v}_k$*
      $m_k \leftarrow \left(\tau_1^{-1} \circ \tau_2 \circ \mathcal{R}_{\hat{v}_k}\nu - \mathcal{R}_{\hat{v}_k}\mu\right) \otimes \hat{v}_k$ ▷ *calculate samples' movements under $\hat{v}_k$*
      $\sigma_i \leftarrow \sigma_i + m_{k,i}|_{i=1,..,\mathcal{B}}$
    end for
    $\sigma_i \leftarrow \sigma_i / ||\sigma_i||_2\ |_{i=1,..,\mathcal{B}}$
    $\mathcal{L} \leftarrow \sum^{\mathcal{B}} \left(\mathcal{L}_{\mathcal{XE}}(\tilde{x_i}, y_i) + \lambda_J ||J(x_i|\sigma_i)||_F^2\right) + \lambda_{SW} SW(\mu, \nu)$ ▷ *overall loss*
    $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta \mathcal{L}$ ▷ *update model's parameters θ*
  end for
end while

Table 9: Basic sanity tests for our OTJR method with *white-box* PGD attack.

Number of step					
Clean	1	10	20	40	50
84.01$_{.53}$	78.86$_{.69}$	56.26$_{.24}$	55.38$_{.29}$	55.1$_{.40}$	55.08$_{.40}$
Perturbation budget ϵ w/ PGD-20					
Clean	8/255	16/255	24/255	64/255	128/255
84.01$_{.53}$	55.38$_{.29}$	24.57$_{1.00}$	9.66$_{.54}$	0.57$_{.01}$	0.00$_{.01}$

D Hyper-parameter sensitivity

In Table 10, we present ablation studies focusing on hyper-parameter sensitivities, namely, λ_J, λ_{SW}, and K, using the CIFAR-10 dataset and WRN34 architecture. We observe that excessive λ_J values compromise accuracy and robustness, a result of the loss function gradients inducing adversarial perturbations during the AT step. While λ_{SW} offers flexibility in selection, models with minimal λ_{SW} values inadequately address adversarial samples, and high values risk eroding

clean accuracy. For the slice count K, a lower count fails to encapsulate transportation costs across latent space distributions; conversely, an overly large K brings marginal benefits at the expense of extended training times. We acknowledge potential gains from further hyper-parameter optimization.

E Training Time

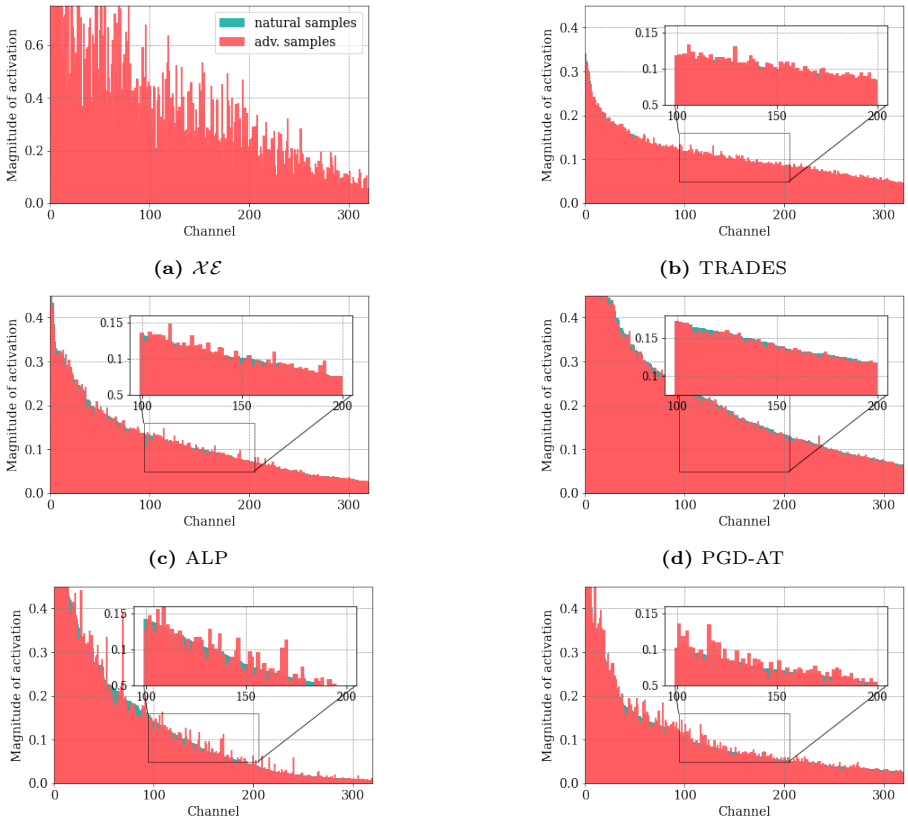
Table 11 indicates the average training time per epoch of all AT methods on our machine architecture using WRN34 model on CIFAR-100 dataset. Notably, although the SAT algorithm demonstrates a commendable per-epoch training duration, its convergence necessitates up to four times more epochs than alternative methods, especially on large scale datasets such as CIFAR-100. Despite our method delivering notable enhancements over prior state-of-the-art frameworks, its computational demand during training remains within acceptable bounds.

Table 10: Hyper-parameter tuning. The sensitivities of hyper-parameters: λ_J, λ_{SW} and K. Without Jacobian regularization, the model cannot achieve the best performance. Trade-off between model's accuracy vs. robustness is shown via λ_{SW}.

Hyper-parameters			Robustness			
λ_J	λ_{SW}	K	*Clean*	PGD^{20}	PGD^{100}	AutoAttack
0.002	64	32	*84.53*	55.07	54.69	52.41
0.01	64	32	*84.75*	54.37	54.06	52.13
0.05	64	32	*82.82*	54.98	54.72	52.00
0.002	32	32	*85.47*	54.85	54.46	52.23
0.002	72	32	*83.19*	55.70	55.40	53.04
0.002	64	16	*81.47*	55.10	54.98	51.82
0.002	64	64	*85.79*	53.80	53.36	51.83

Table 11: Training time per epoch of AT methods. Even though our method's training time/epoch is slightly slower than the SAT's as the additional Jacobian regularization, it can achieve faster convergence on large-scale datasets.

Method	Time (*mins*)	Method	Time (*mins*)
$\mathcal{XE}$	$1.63_{.00}$	PGD-AT	$12.32_{.03}$
ALP	$13.56_{.03}$	TRADES	$16.42_{.06}$
SAT	$14.68_{.01}$	OTJR (Ours)	$18.02_{.12}$



(a) $\mathcal{XE}$ **(b)** TRADES

(c) ALP **(d)** PGD-AT

(e) SAT **(f)** OTJR (*Ours*)

Fig. 8: Magnitude of activation at the penultimate layer for models trained with different defense methods. Our OTJR can regulate adversarial samples' magnitudes similar to clean samples' while well suppressing both of them.

F Activation Magnitude

Figure 8 depicts the activation magnitudes at the penultimate layer of WRN34 across various AT frameworks. Although AT methods manage to bring the adversarial magnitudes closer to their clean counterparts, the magnitudes generally remain elevated, with PGD-AT being especially prominent. Through a balanced integration of the input Jacobian matrix and output distributions, our proposed method effectively mitigates the model's susceptibility to perturbed samples.

G Broader Impact

Utilizing machine learning models in real-world systems necessitates not only high accuracy but also robustness across diverse environmental scenarios. The central motivation of this study is to devise a training framework that augments the robustness of Deep Neural Network (DNN) models in the face of various adversarial attacks, encompassing both white-box and black-box methodologies. To realize this objective, we introduce the OTJR framework, an innovative approach that refines traditional Jacobian regularization techniques and aligns output distributions. This research represents a significant stride in synergizing adversarial training with input-output Jacobian regularization—a combination hitherto underexplored—to construct a more resilient model.