A. More results of Fig. 2

In Fig. 3, we provide additional empirical results of CIFAR-10 on ResNet-18, which show that our method can effectively flatten the loss landscape and find flat minima.

B. Details of the experiments

B.1. Network Architecture

For all of our experiments in Sec. 5, we use 4 network architectures as ResNet, WideResNet, VGG and MobileNetV2. We present the details in the following.

- ResNet/WideResNet: Architectures used are PreAct ResNet. All convolutional layers (except downsampling convolutional layers) have kernel size 3×3 with stride 1. Downsampling convolutions have stride 2. All the ResNets have five stages (0-4) where each stage has multiple residual/downsampling blocks. These stages are followed by a max-pooling layer and a final linear layer. We study the PreAct ResNet 18 and WideResNet-34-10.

- VGG: Architecture consists of multiple convolutional layers, followed by multiple fully connected layers and a final classifier layer (with output dimension 10 or 100). We study the VGG networks with 16 layers.

- MobileNetV2: Architecture is built on an inverted residual structure, with residual connections between bottleneck layers. As a source of non-linearity, the intermediate expansion layer filters features with lightweight depthwise convolutions. As a whole, the architecture of MobileNetV2 includes a fully convolutional layer with three filters, followed by 19 residual bottleneck layers.

B.2. Checkpoints

We set checkpoints for each epoch between $100 - 110$ and $150 - 160$, each 5 epoch between $110 - 150$ and $160 - 200$. All best performances are gotten from these checkpoints.

C. An un-rigorous theoretical perspective

Please note that we provide a ***potential, interesting but un-rigorous*** *theoretical perspective in the following. This section is* ***not claimed as the contribution*** *of this paper.*

This section explores theoretical implication of the use of randomized weights on robustness. Specifically, we provide a **shallow** theoretical perspective which discusses how randomized weights may affect the information-theoretic generalization bound of both clean and adversarial data.

Under the information-theoretic context, a learning algorithm can be taken as a randomized mapping, where training data set is input and hypothesis is output. With that, [75] considered a generalization bound based on the information contained in weights $I(\Lambda_{\mathbb{W}}(\mathcal{S});\mathbf{w})$, where $\Lambda_{\mathbb{W}}(\mathcal{S}) := \big(\mathcal{L}(f_{\mathbf{w}}(\mathcal{S}),\mathcal{Y})\big)_{\mathbf{w}\in\mathbb{W}}$ is the collection of empirical losses in hypotheses space $\mathbb{W}$. Let $\mathbb{P}(\mathcal{L}(f_{\mathbf{w}_1}(\mathcal{S}),\mathcal{Y}),\mathbf{w}_2) = 0$ where $\mathbf{w}_1 \neq \mathbf{w}_2$, we can use $I(\mathcal{L}(f_{\mathbf{w}}(\mathcal{S}),\mathcal{Y});\mathbf{w})$ to approximate $I(\Lambda_{\mathbb{W}}(\mathcal{S});\mathbf{w})$ where $\mathbf{w}$ is the randomized weight distributed in $\mathbb{W}$, then get the following perspective.

Suppose $\mathcal{L}(f_{\mathbf{w}}(\mathbf{s}),y)$ is σ_*-sub-Gaussian, $\mathcal{D}$ is the clean data distribution and $\mathcal{S}$ is the training data set with m samples, then

$$\begin{aligned}&\mathbb{E}_{\mathbf{w}}\Big(\mathcal{L}\big(f_{\mathbf{w}}(\mathcal{D}),\mathcal{Y}\big)-\mathcal{L}\big(f_{\mathbf{w}}(\mathcal{S}),\mathcal{Y}\big)\Big)\\ &\le \sqrt{\frac{2\sigma_*^2}{m} I\Big(\mathcal{L}\big(f_{\mathbf{w}}(\mathcal{S}),\mathcal{Y}\big);\mathbf{w}|\mathcal{S}\Big)}.\end{aligned} \tag{9}$$

The above inequation provides the upper bound on the expected generalization error of randomized weights. Building upon the above bound, we consider generalization errors of both clean and adversarial data based on discrete distribution, then obtain the following proposition.

Proposition C.1 *Let $\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}),y)$ and $\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}'),y)$ be σ_*-sub-Gaussian. We suppose the adversarial trained model contains information of $\mathcal{S} \vee \mathcal{S}'$, where $\mathcal{S} \vee \mathcal{S}'$ is the joint set and the training samples are chosen at random from $\mathcal{S}$ and $\mathcal{S}'$ with probabilities q and $1-q$, i.e., $\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S} \vee \mathcal{S}'),\mathcal{Y}) = q\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}) + (1-q)\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})$. We let $q \in (0, 0.5)$, since $\mathcal{S}'$ occupies a large proportion in adversarial training, then*

$$\begin{aligned}&\mathbb{E}_{\mathbf{w}+\mathbf{u}}\Big(\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{D}\vee\mathcal{D}'),\mathcal{Y}\big)-\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}\vee\mathcal{S}'),\mathcal{Y}\big)\Big)\\ &\le \sqrt{\frac{2\sigma_*^2}{m} I\Big(\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}\vee\mathcal{S}'),\mathcal{Y}\big);\mathbf{w}+\mathbf{u}|\mathcal{S}\vee\mathcal{S}'\Big)}\\ &\le \sqrt{\frac{2\sigma_*^2}{m} I\Big(\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big),\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y}\big);\mathbf{w}+\mathbf{u}|\mathcal{S}\vee\mathcal{S}'\Big)}.\end{aligned} \tag{10}$$

We now make several observations about above inequation. First, it is obvious that more training data (larger m) helps adversarial training to get a high-performance model. Second, take into account both generalization errors of clean and adversarial data with coefficient q, a lower mutual information between $\Big(\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big),\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y}\big)\Big)$ and $\mathbf{w}+\mathbf{u}$ is essential to get a better performance of robustness and clean accuracy.

The above mutual information is a statistic over high-dimensional space, thus we are almost impossible to directly estimate and optimize it during training. Nevertheless, we can reduce it implicitly through the following Lemma.

Lemma C.2 $I\Big(\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big),\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y}\big);\mathbf{w}+\mathbf{u}\Big)$ *is lower bounded by* $I\Big(\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big);\ \mathbf{w}+\mathbf{u}\Big)$

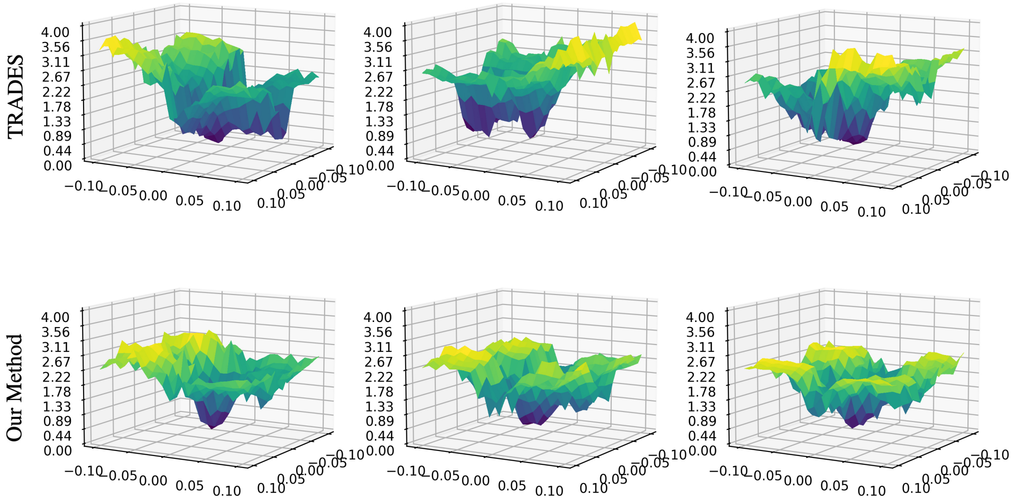

Figure 3. Comparison of loss landscapes of TRADES trained model (the first row) and TRADES+1$_{st}$+2$_{nd}$ (our method, the second row) trained model. Loss plots in each column are generated from the same original image randomly chosen from the CIFAR-10 test dataset. Following the settings in [20], the z axis represents the loss, the x and y axes represent the magnitude of the perturbation added in the directions of $\mathbf{sign}\nabla_{\mathbf{s}} f(\mathbf{s})$ and Rademacher(0.5) respectively.

and upper bounded by $I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathbf{w}+\mathbf{u}\big) + I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\big)$*, i.e.,*

$$\begin{aligned}
&I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathbf{w}+\mathbf{u}\big)\\
&\le I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}),\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\big)\\
&\le I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathbf{w}+\mathbf{u}\big) + I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\big).
\end{aligned} \tag{11}$$

Proof C.2 *We suppose the adversarial trained model contains information of* $\mathcal{S}\vee\mathcal{S}'$*, where* $\mathcal{S}\vee\mathcal{S}'$ *is the joint set with* $\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}\vee\mathcal{S}'),\mathcal{Y}) = q\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}) + (1-q)\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})$*. We let* $q\in(0,0.5)$ *because* $\mathcal{S}'$ *occupies a large proportion in adversarial training. Randomized weight* $\mathbf{w}$ *is generated by normal training with data set* $\mathcal{S}$ *and* $\mathbf{w}+\mathbf{u}$ *is generated by adversarial training with joint set* $\mathcal{S}\vee\mathcal{S}'$*, then*

$$\begin{aligned}
&I\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}),\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\Big)\\
&= H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}),\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})\Big)\\
&\quad - H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}),\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})\Big|\mathbf{w}+\mathbf{u}\Big)\\
&= H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y})\Big) + H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})\Big|\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y})\Big)\\
&\quad - H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y})\Big|\mathbf{w}+\mathbf{u}\Big)\\
&\quad - H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})\Big|\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}),\mathbf{w}+\mathbf{u}\Big)\\
&= H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y})\Big) + H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})\Big)\\
&\quad - I\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})\Big)\\
&\quad - H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y})\Big|\mathbf{w}+\mathbf{u}\Big) - H\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})\Big|\mathbf{w}+\mathbf{u}\Big)\\
&\quad + I\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y})\Big|\mathbf{w}+\mathbf{u}\Big)\\
&= I\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathbf{w}+\mathbf{u}\Big) + I\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\Big)\\
&\quad - I\Big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\Big).
\end{aligned} \tag{12}$$

Note that $I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathbf{w}+\mathbf{u}\big) \le I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\big)$ *holds, due to the data processing inequality and* $\mathcal{S}-\mathcal{S}\vee\mathcal{S}'-\mathbf{w}+\mathbf{u}$ *forms a Markov chain under adversarial training where* $\mathcal{S}'$ *occupies a larger proportion in* $\mathcal{S}\vee\mathcal{S}'$*. As* $I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\big) \ge I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\big)$*, we can easily get Lem. C.2.* □

Lem. C.2 gives us an upper bound and a lower bound. The upper bound represents the worst case of adversarial trained model where $\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big)$ and $\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y}\big)$ are radically different (uncorrelated). To some extent, it means the trained model is failed to extract common features of clean data and adversarial data, thus needs to use more parameters to recognize clean and adversarial examples respectively. Lem. C.2 also charts a realizable optimization direction for the model, the lower bound $I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y});\mathbf{w}+\mathbf{u}\big)$ is the optimal case of adversarial training for $I\big(\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}),\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y});\mathbf{w}+\mathbf{u}\big)$. In this situation, $\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big)$ and $\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y}\big)$ are completely correlated, that is, the optimal adversarial trained model is successful at extracting common features of clean data and adversarial data.

It is obvious that $I\big(\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big),\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y}\big);\mathbf{w}+\mathbf{u}\big)$ is still difficult to be estimated in a high-dimensional space. Fortunately, Lem. C.2 allows us to optimize it via narrowing the distance between $\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big)$ and $\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y}\big)$, which makes $I\big(\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big),\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}'),\mathcal{Y}\big);\mathbf{w}+\mathbf{u}\big)$ close to the lower bound $I\big(\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathcal{S}),\mathcal{Y}\big);\mathbf{w}+\mathbf{u}\big)$.

Lemma C.3 *In this lemma, we consider the case of binary response $y\in\{0,1\}$, then the gap between $\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}),y\big)$ and $\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}'),y\big)$ is positive correlated with the KL divergence between $f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})$ and $f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}')$, i.e.,*

$$\begin{aligned}&|\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}),y\big)-\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}'),y\big)|\\&\propto \mathrm{KL}\big(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})||f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}')\big),\end{aligned}\tag{13}$$

where we let $\propto$ represent positive correlation.

Proof C.3 *Let $f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})_{true}$ be the normalized output of $f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})$ for true label and we consider the case of binary response $y\in\{0,1\}$ in Lem. C.3. Then,*

$$\begin{aligned}&|\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}),y\big)-\mathcal{L}\big(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}'),y\big)|\\&=\Big|\log\frac{f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})_{true}}{f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}')_{true}}\Big|\\&\propto f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})_{true}\log\frac{f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})_{true}}{f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}')_{true}}\\&\quad+(1-f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})_{true})\log\frac{1-f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})_{true}}{1-f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}')_{true}}\\&=\mathrm{KL}\big(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})||f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}')\big),\end{aligned}\tag{14}$$

where we let $\propto$ represent positive correlation. □

Lem. C.3 demonstrates $|\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}),y)-\mathcal{L}(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}'),y)|$ is positive correlated with $\mathrm{KL}(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})||f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}'))$ in a binary case, this can also approximately hold in a multi-class case. Thus, it allows us to optimize the mutual information utilizing a simplified term of $\mathrm{KL}(f_{\mathbf{w}+\mathbf{u}}(\mathbf{s})||f_{\mathbf{w}+\mathbf{u}}(\mathbf{s}'))$. Although we are still difficult to directly deal with this KL term during training, it can be decomposed by our method in Sec. 4 with Taylor series.

D. Optimization

It is easy to see that minimizing $\mathbb{E}_{\mathbf{u}}(\mathcal{L}(g_{\mathbf{s}}'(\mathbf{w})^T\mathbf{u}, g_{\mathbf{s}'}'(\mathbf{w})^T\mathbf{u}))$, $\mathbb{E}_{\mathbf{u}}(\mathcal{L}(\mathbf{u}^T g_{\mathbf{s}}''(\mathbf{w})\mathbf{u},\mathbf{u}^T g_{\mathbf{s}'}''(\mathbf{w})\mathbf{u}))$ is equivalent to reducing the distance between $g_{\mathbf{s}}'(\mathbf{w})$ and $g_{\mathbf{s}'}'(\mathbf{w})$, $g_{\mathbf{s}}''(\mathbf{w})$ and $g_{\mathbf{s}'}''(\mathbf{w})$, respectively. Normally, the ℓ_2 distance between vectors $g_{\mathbf{s}}'(\mathbf{w})$ and $g_{\mathbf{s}'}'(\mathbf{w})$ can be defined as

$$\Big|\Big|\frac{\partial g_{\mathbf{s}}(\mathbf{w})}{\partial\mathbf{w}}-\frac{\partial g_{\mathbf{s}'}(\mathbf{w})}{\partial\mathbf{w}}\Big|\Big|_2^2=\sum_j\sum_i\Big[\frac{\partial(g_{\mathbf{s}}(\mathbf{w})-g_{\mathbf{s}'}(\mathbf{w}))}{\partial\mathbf{W}_{(j,i)}}\Big]^2.\tag{15}$$

We extend Eq. (15) by considering the sum of each row vector of $\frac{\partial(g_{\mathbf{s}}(\mathbf{w})-g_{\mathbf{s}'}(\mathbf{w}))}{\partial\mathbf{W}}$ and define the distance between $\frac{\partial g_{\mathbf{s}}(\mathbf{w})}{\partial\mathbf{w}}$ and $\frac{\partial g_{\mathbf{s}'}(\mathbf{w})}{\partial\mathbf{w}}$ as

$$\sum_j\Big[\sum_i\frac{\partial(g_{\mathbf{s}}(\mathbf{w})-g_{\mathbf{s}'}(\mathbf{w}))}{\partial\mathbf{W}_{(j,i)}}\Big]^2.\tag{16}$$

We notice that, according to chain rule, $\Big[\sum_i\frac{\partial(g_{\mathbf{s}}(\mathbf{w})-g_{\mathbf{s}'}(\mathbf{w}))}{\partial\mathbf{W}_{(j,i)}}\Big]^2$ corresponds to $g_{\mathbf{s}}(\mathbf{w})_j$ and $g_{\mathbf{s}'}(\mathbf{w})_j$, where $g_{\mathbf{s}}(\mathbf{w})_j$ and $g_{\mathbf{s}'}(\mathbf{w})_j$ are the j-th output of $g_{\mathbf{s}}(\mathbf{w})$ and $g_{\mathbf{s}'}(\mathbf{w})$ respectively. Thus we can easily increase $g_{\mathbf{s}'}(\mathbf{w})_j$ to reduce $\sum_i\frac{\partial(g_{\mathbf{s}}(\mathbf{w})-g_{\mathbf{s}'}(\mathbf{w}))}{\partial\mathbf{W}_{(j,i)}}$ when $\sum_i\frac{\partial(g_{\mathbf{s}}(\mathbf{w})-g_{\mathbf{s}'}(\mathbf{w}))}{\partial\mathbf{W}_{(j,i)}}$ is large. In contrast, we can increase $g_{\mathbf{s}}(\mathbf{w})_j$ to reduce $\sum_i\frac{\partial(g_{\mathbf{s}'}(\mathbf{w})-g_{\mathbf{s}}(\mathbf{w}))}{\partial\mathbf{W}_{(j,i)}}$ when $\sum_i\frac{\partial(g_{\mathbf{s}'}(\mathbf{w})-g_{\mathbf{s}}(\mathbf{w}))}{\partial\mathbf{W}_{(j,i)}}$ is large. Then, we can approximately optimize $\mathbb{E}_{\mathbf{u}}(\mathcal{L}(g_{\mathbf{s}}'(\mathbf{w})^T\mathbf{u},g_{\mathbf{s}'}'(\mathbf{w})^T\mathbf{u}))$ through minimizing

$$\begin{aligned}&\frac{1}{2}\mathcal{L}\Big(g_{\mathbf{s}'}(\mathbf{w}),\\&\quad\Big[\sum_i\frac{\partial(g_{\mathbf{s}}(\mathbf{w})-g_{\mathbf{s}'}(\mathbf{w}))}{\partial\mathbf{W}_{(1,i)}},...,\sum_i\frac{\partial(g_{\mathbf{s}}(\mathbf{w})-g_{\mathbf{s}'}(\mathbf{w}))}{\partial\mathbf{W}_{(N,i)}}\Big]^T\Big)\\&+\frac{1}{2}\mathcal{L}\Big(g_{\mathbf{s}}(\mathbf{w}),\\&\quad\Big[\sum_i\frac{\partial(g_{\mathbf{s}'}(\mathbf{w})-g_{\mathbf{s}}(\mathbf{w}))}{\partial\mathbf{W}_{(1,i)}},...,\sum_i\frac{\partial(g_{\mathbf{s}'}(\mathbf{w})-g_{\mathbf{s}}(\mathbf{w}))}{\partial\mathbf{W}_{(N,i)}}\Big]^T\Big),\end{aligned}\tag{17}$$

where $\mathbf{W}_{(j,k)}$ is the element of j-th row, k-th column of weight matrix $\mathbf{W}$, N is the number of neurons (units) on the layer.

Similarly, we define the distance between $\frac{\partial^2 g_{\mathbf{s}}(\mathbf{w})}{\partial\mathbf{w}\partial\mathbf{w}}$ and $\frac{\partial^2 g_{\mathbf{s}'}(\mathbf{w})}{\partial\mathbf{w}\partial\mathbf{w}}$ as

$$\sum_{l=1}^N\Big[\sum_i\sum_j\sum_k\frac{\partial^2 g_{\mathbf{s}}(\mathbf{w})-\partial^2 g_{\mathbf{s}'}(\mathbf{w})}{\partial\mathbf{W}_{(l,i)}\partial\mathbf{W}_{(j,k)}}\Big]^2.\tag{18}$$

We also notice that, according to chain rule, $\Big[\sum_i\sum_j\sum_k\frac{\partial^2 g_{\mathbf{s}}(\mathbf{w})-\partial^2 g_{\mathbf{s}'}(\mathbf{w})}{\partial\mathbf{W}_{(l,i)}\partial\mathbf{W}_{(j,k)}}\Big]^2$ corresponds to $g_{\mathbf{s}}(\mathbf{w})_l$ and $g_{\mathbf{s}'}(\mathbf{w})_l$. We can increase $g_{\mathbf{s}'}(\mathbf{w})_l$ to reduce $\sum_i\sum_j\sum_k\frac{\partial^2 g_{\mathbf{s}}(\mathbf{w})-\partial^2 g_{\mathbf{s}'}(\mathbf{w})}{\partial\mathbf{W}_{(l,i)}\partial\mathbf{W}_{(j,k)}}$ when $\sum_i\sum_j\sum_k\frac{\partial^2 g_{\mathbf{s}}(\mathbf{w})-\partial^2 g_{\mathbf{s}'}(\mathbf{w})}{\partial\mathbf{W}_{(l,i)}\partial\mathbf{W}_{(j,k)}}$ is large. In contrast, we can increase $g_{\mathbf{s}}(\mathbf{w})_l$ to reduce $\sum_i\sum_j\sum_k\frac{\partial^2 g_{\mathbf{s}'}(\mathbf{w})-\partial^2 g_{\mathbf{s}}(\mathbf{w})}{\partial\mathbf{W}_{(l,i)}\partial\mathbf{W}_{(j,k)}}$ when $\sum_i\sum_j\sum_k\frac{\partial^2 g_{\mathbf{s}'}(\mathbf{w})-\partial^2 g_{\mathbf{s}}(\mathbf{w})}{\partial\mathbf{W}_{(l,i)}\partial\mathbf{W}_{(j,k)}}$ is large. Thus, we can approximately optimize $\mathbb{E}_{\mathbf{u}}(\mathcal{L}(\mathbf{u}^T g_{\mathbf{s}}''(\mathbf{w})\mathbf{u},\mathbf{u}^T g_{\mathbf{s}'}''(\mathbf{w})\mathbf{u}))$ through minimizing

Table 7. CIFAR-10, PreAct ResNet 18, under RayS hard label attack (ℓ_∞,%).

Method	Clean Acc	ADBD	RayS Acc
TRADES	82.89	0.0412	56.06
TRADES+1_{st}+2_{nd}	84.13	0.0435	57.03

Table 8. CIFAR-10, ResNet 18, comparison of our method on AT with GAT and HAT (ℓ_∞,%).

Method	Clean	PGD-20	CW-20	AA
AT	82.41	52.77	50.43	47.1
AT+1_{st}+2_{nd}	83.56	54.23	52.19	48.7
GAT	80.49	53.13	-	47.3
TRADES+GAT	81.32	53.37	-	49.6
HAT	84.90	49.08	-	-
HAT(DDPM)	86.86	57.09	-	-

$$\frac{1}{2}\mathcal{L}\Big(g_{\mathbf{s}'}(\mathbf{w}), \big[\sum_i\sum_j\sum_k \frac{\partial^2 g_{\mathbf{s}}(\mathbf{w}) - \partial^2 g_{\mathbf{s}'}(\mathbf{w})}{\partial \mathbf{W}_{(1,i)} \partial \mathbf{W}_{(j,k)}}, ...,$$
$$\sum_i\sum_j\sum_k \frac{\partial^2 g_{\mathbf{s}}(\mathbf{w}) - \partial^2 g_{\mathbf{s}'}(\mathbf{w})}{\partial \mathbf{W}_{(N,i)} \partial \mathbf{W}_{(j,k)}}\big]^T\Big)$$
$$+ \frac{1}{2}\mathcal{L}\Big(g_{\mathbf{s}}(\mathbf{w}), \big[\sum_i\sum_j\sum_k \frac{\partial^2 g_{\mathbf{s}'}(\mathbf{w}) - \partial^2 g_{\mathbf{s}}(\mathbf{w})}{\partial \mathbf{W}_{(1,i)} \partial \mathbf{W}_{(j,k)}}, ...,$$
$$\sum_i\sum_j\sum_k \frac{\partial^2 g_{\mathbf{s}'}(\mathbf{w}) - \partial^2 g_{\mathbf{s}}(\mathbf{w})}{\partial \mathbf{W}_{(N,i)} \partial \mathbf{W}_{(j,k)}}\big]^T\Big). \tag{19}$$

E. More empirical results

More results of RayS hard-label attack [7], HAT [57], GAT [66] methods are given in Tabs. 7 and 8.