We thank all the reviewers for their valuable comments. Below, we address the detailed comments of each reviewer.

To Reviewer #1. **Perturbations crafted with FN:** The transfer accuracy under PGD-10 for TRADES+HE $\rightarrow$ TRADES is 63.20%, and for TRADES $\rightarrow$ TRADES+HE is 65.90%. We also apply PGD-1 and PGD-2 w/wo FN to attack a standard WRN-34-10 model, and report the accuracy in Table A. As seen, the attacks are more efficient with FN. **Choice of (s, m):** Heuristically, the value range of s is based on the averaged logit norms of standard training, which is around 10. The margin range m is chosen to be around $\cos 30^\circ \approx 0.15$.

Table A: PGD w/wo FN.

Attack	FN	Acc (%)
PGD-1	✖	67.09
	✔	**62.89**
PGD-2	✖	50.37
	✔	**33.75**

FN on learning hard adversarial examples: Indeed, larger $\nabla_{\boldsymbol{\omega}} \cos(\theta)$ could have larger contribution in the mini-batch. However, since $\cos(\theta)$ is bounded, its gradient will be smaller when the sample is gradually well-learned (i.e., $\cos(\theta) \rightarrow 1$). Then other samples that are not well-learned will dynamically contribute more. In contrast, if we do not apply FN, the unbounded $\|z\|$ will cause vicious circles (i.e., large $\nabla_{\boldsymbol{\omega}}\|z\|$ leads to larger $\|z\|$), and the easy examples will keep dominating the training. As we show in Table B, our mechanism can help the model to achieve SOTA performance under the stronger attacks. **Fig. 1, Sec. 3.5, and other comments:** Thank you for the suggestions. We will construct synthetic demos and better organize Sec. 3.5; We will involve more related work and re-check the references into published versions; We will include complete results on adaptive attacks and polish our Tables.

To Reviewer #2. **Evaluation under stronger attacks:** We evaluate under two stronger attacks including RayS[1] and AutoAttack[2] on CIFAR-10. We train WRN models via PGD-AT+HE, with weight decay of 5×10^{-4}. For RayS, we evaluate on $1,000$ test samples due to the high computation. The results are shown in Table B, where the trained WRN-34-20 model achieves SOTA performance (no additional data) according to the reported benchmarks.

Table B: Acc. (%) of PGD-AT+HE.

Model	Clean	RayS	AA
WRN-34-10	86.25	57.8	53.16
WRN-34-20	85.14	59.0	53.74

First-order adversary: We cited Simon-Gabriel et al. [55] in line 100 when we introducing first-order adversaries, and we never claimed it as one of our contributions. Thank you for pointing out other related work and we'll discuss on them in the revision. **Training objective of TRADES:** In the 4-th line of Sec. 5.2 in TRADES paper [78], the authors clarify that they choose $\mathcal{L}$ as the cross-entropy loss, so we provide the formula under the cross-entropy loss in Table 1. In the TRADES code[3], they apply KL loss, and we also use KL loss to keep consistency. **Why use multiplicative scalar:** First, the softmax function is invariant to any additive offset, i.e., $\mathbb{S}(x + s) = \mathbb{S}(x)$. After executing FN and WN, the logit values will be constrained to $[-1, 1]$, which will make the training loss be trapped at a very high value and vanish the gradients. Then a multiplicative scalar s can enlarge the value interval and promote the training process.

To Reviewer #3. **High-level intuition of line 36-40:** In the binary classification, the CE objective equals to maximizing $\mathcal{L}(x) = (W_0 - W_1)^\top z = \|W_{01}\|\|z\| \cos(\theta)$ on an input x with label $y = 0$. (**i**) If x is correctly classified, there is $\mathcal{L}(x) > 0$, and adversaries aim to craft x' such that $\mathcal{L}(x') < 0$. Since $\|W_{01}\|$ and $\|z\|$ are always positive, they cannot alter the sign of $\mathcal{L}$. Thus FN and WN encourage the adversaries to attack the crucial component $\cos(\theta)$; (**ii**) In a data batch, points with larger $\|z\|$ will dominate (vicious circle on increasing $\|z\|$), which makes the model ignore the critical component $\cos(\theta)$. FN alleviates this problem, and well-learned hard examples will dynamically have smaller weights during training since $\cos(\theta)$ is bounded; (**iii**) When there are much more samples of label 0, the CE objective will tend to have $\|W_0\| \gg \|W_1\|$ to minimize the loss. WN can relieve this trend and encourage W_0 and W_1 to diversify in directions; (**iv**) The role of margin is analogous to it in SVM. We will better reorganize Sec. 3 in the revision.

Relation to Cosface: The HE mechanism in Eq. (6) has the same form as Cosface [65], and we contribute to applying it in the adversarial training with both theoretical and empirical analyses. We will detail the relation in the revision. **Hard examples:** We define the hardness w.r.t. $\nabla_{\boldsymbol{\omega}}\mathcal{L}(x)$, where hard (adversarial) examples usually correspond to the worst cases for a model. As you suggest, bias towards them may be unreasonable in the sense of human perception. However, under a threat model (e.g., $8/255$, ℓ_∞) in which we evaluate our defenses, the ground-truth labels are assumed to be invariant and always well-defined. **Union of attacks:** The improvement on PGD-AT is more significant (SOTA as in Table B) since its framework is better aligned with our analyses. We will provide more comparisons in the revision.

To Reviewer #5. Thank you for your kind words. **The role of margin:** The margin m encourages a larger gap between the logit of the true label and other logits. This makes the learned features more aligned with the corresponding softmax weights, as well as more distinguished weight directions. We will detail on the margin with more empirical results. **Results in Table 2:** We can observe that the HE mechanism is better suitable for PGD-AT, since its framework is more consistent with our analyses. Our newest experiment results in Table B demonstrate the SOTA performance of PGD-AT+HE. In contrast, encoding HE into ALP and TRADES is less formally justified, and we will try to elaborate on them with fine-tuned formulas in the revision. **Training time:** We show the training time in Table C on CIFAR-10, and we can see that HE only introduces little extra computation.

Table C: Average training time (minutes) per epoch.

Method	Time
PGD-AT	19.22
PGD-AT+HE	19.23
ALP	20.89
ALP+HE	20.95
TRADES	25.83
TRADES+HE	25.84

Reference: [1]RayS benchmark: github.com/uclaml/RayS. [2]AutoAttack benchmark: github.com/fra31/auto-attack. [3]TRADES code: github.com/yaodongyu/TRADES.