We thank all reviewers for the encouraging feedback and detailed comments which we’ll integrate into the next version.

**R1, R2, R3:** *FGSM + GradAlign is slower than FGSM.*

We admit that GradAlign leads to a slowdown: e.g. on CIFAR-10 the runtime of FGSM AT is 9.5 min while FGSM + GradAlign AT takes 30.9 min on an NVIDIA V100 GPU. However, we present GradAlign as a proof of concept motivated by our empirical and theoretical analysis. We hope that the same effect (stability of the gradients under random noise) can be achieved in future work with other regularization methods that avoid double backpropagation.

**R1:** *What makes the grad. alignment low by FGSM training? Why a noise-sensitive filter is learned by FGSM training?*

These are very interesting questions, and we believe they are connected to the finding of [25] that SGD for neural networks learns models of increasing complexity, e.g., measured in terms of local linearity.

**R2:** *Running stronger attacks, e.g., AutoAttack or MultiTargeted*

After running AutoAttack, we observe that it proportionally reduces the adversarial accuracy for all methods. E.g., for \( \varepsilon = \frac{9}{255} \), FGSM + GradAlign achieves 44.54\%\(\pm0.24\%\) adversarial accuracy while FGSM-RS achieves 42.80\%\(\pm0.58\%\). This is consistent with the results of the AutoAttack paper where they show an average reduction of 2\%–3\% adversarial accuracy compared to most of the evaluations that were originally done with variants of PGD.

**R2:** *Discussion on LLR [26] and CURE [24]. Their exact goal is to make the loss surface smoother.*

Indeed, the goals of LLR/CURE and GradAlign w.r.t smoothness are similar, however the important difference is that we were not looking for a replacement of adv. training, but rather for a complement that would prevent catastrophic overfitting. Related to this, in Table 7 we also provide the results for FGSM+CUR where we can see that CURE also stabilizes FGSM training, but performs worse than FGSM+GradAlign. Finally, GradAlign does not have any worst-case motivation unlike CURE (uses the FGSM point) or LLR (uses a point with the worst-case linear violation).

**R2:** *Line 118: How is the alpha step-size tuned for this experiments? (is it 1.25\varepsilon?)*

Yes, we used \( \alpha = 1.25\varepsilon \) since it was the recommended choice of Wong et al. [44] on all the datasets they considered.

**R2:** *Performance of FGSM + GradAlign in large \( \varepsilon \) settings (16/255) on ImageNet against random targeted attacks.*

First, we note that for FGSM+GradAlign on CIFAR-10 we did not encounter cat. overfitting even with \( \varepsilon = \frac{16}{255} \). We briefly tried targeted AT with \( \varepsilon = \frac{16}{255} \) on ImageNet but we did not succeed at training a sufficiently robust model. We think that it is likely that more epochs (e.g., LLR [26] used 110 epochs instead of 15 epochs as we did following [44]) and different hyperparameters are needed, but tuning them on ImageNet was too computationally expensive for us.

**R3:** *Learning curve with robust accuracy and the effect of the regularizer on grad. alignment, e.g., on CIFAR-10 with \( \varepsilon = 14 \) and 60 epochs.*

We present this experiment in Fig. A. The only two methods that do not fail at epoch 60 are PGD-10 and FGSM+GradAlign which is also reflected by their gradient alignment, i.e. cosine distances (highest for GradAlign).

**R3:** *I’d vote for early stop since it is not complicated and rather efficient.*

We’d like to clarify that using early stopping leads to worse PGD accuracy (particularly for high \( \varepsilon \)) and much worse clean accuracy as we comment in lines 307-310. Thus, early stopping alone is not a satisfying solution.

**R3:** *What happens when we use \( \varepsilon > 4 \) for ImageNet? Do all methods including the proposed one fail?*

We have the results for \( \varepsilon = 6 \) in Table 6 and catastrophic overfitting there occurs only for the FGSM-RS model. However, the results on ImageNet are not fully conclusive since we could not repeat the experiments over multiple random seeds due to the computational constraints.

**R3:** *Lemma 1: what if \( \eta \sim \mathcal{U}([-\beta, \beta]^d) \)? Will \( \beta \) appear in the bound?*

We were interested in \( \beta = \varepsilon \) since it was the setting of [44]. Indeed, \( \beta \) will appear in the bound, but this will not change the message of Lemma 1.

**R4:** *The empirical analysis is mainly done on CIFAR10 only.*

We’d like to emphasize that we have provided experiments on ImageNet to illustrate that GradAlign can be scaled to large datasets. But due to our limited computational resources, we could not do replications over random seeds and a thorough comparison to other methods (particularly, to PGD-10) for a range of \( \varepsilon \) as on CIFAR-10 (e.g., Fig. 1 required to train 480 different models).

**R4:** *Why does cat. overfitting not occur on ImageNet for \( \varepsilon \in \{2, 4\} \)? How does the gradient alignment evolve?*

The main reason is that these \( \varepsilon \) values are sufficiently small (we observed the same also on CIFAR/SVHN for \( \varepsilon \leq \frac{4}{255} \)). The gradient alignment decreases gradually over epochs, but without a sharp drop that would indicate cat. overfitting.