We would like to thank reviewers **R1** , **R2** , and **R3** for their insightful comments. All will definitely help improving the quality of the paper. Below, we tried to address as many comments as possible given the space and time constraints.

R1 : <*Why "warm start" with the previous perturbation?*> Note that during the first replay step, with warm-start, we are computing the gradient at a point corresponding to the original image plus a partially-random perturbation. In contrast, without warm-start, the gradients are computed with respect to the clean image. For simplicity, consider $m = 2$ (*a.k.a.* Free-2) which is similar to FGSM (the approximate gradient is only computed and applied once before moving on to the next mini-batch). If we reset the perturbations to zero, the model doesn't become robust due to gradient masking (3.72% accuracy against a PGD attack on CIFAR-10) and a simple rand+FGSM attack can break it. We experimented with re-initializing the perturbation to be a random perturbation before the first replay step instead of re-using the perturbation from the previous mini-batch and got similar robustness to re-using the perturbation from the previous mini-batch. <$\epsilon = 2/255$ *used in ImageNet experiments is small.*> The results for other values (up to $\epsilon = 7$) are presented in Figure 4 and appendix. <*Exact parameters used when testing against the C&W attack (Table 1).*> We tested various values for the CW attack parameters based on the Madry's implementation: $\text{maximize} - RELU(z_y(x_i + \delta) - z_w(x_i + \delta) + S)$, where z_y and z_w are the correct and the largest incorrect logit, respectively. We use step-size=2, and $S = 0$ in the paper. <*Why CW and random restart only evaluated on CIFAR-10?*> Since CIFAR-10 is one of the main benchmark problems right now, we wanted to make sure of reporting all well-known attacks on this dataset. However per your request, we have run an expanded set of experiments for the rebuttal, including CIFAR-100: 7-PGD trained (CW-100: 23.0%/ 10-restart PGD-20: 22.6%) Free-8 (CW-100: 24.4%/ 10-restart PGD-20: 25.7%) and ImageNet: Free-8 (10-restart PGD-20 40.0%, CW: 38.6%). <*Paper mentions SPSA, but fails to compare, claiming it would not perform great.*> We started SPSA attacks, but results are still pending as SPSA is slow (over 24 hours/experiment). In preliminary results, PGD was consistently stronger for all scenarios observed (this was also observed in the original SPSA paper, which acknowledges the superiority of PGD when gradients are available to the attacker). <*How would one tune m?*> One advantage of our method is having only a single hyper-parameter. It is fairly simple to tune m using a coarse grid search and training for a few epochs starting from a pre-trained model. <*Make source code/models publicly available.*> Our official implementations based on both PyTorch and Tensorflow libraries are publicly available online. Moreover, our results are replicated by 2 independent unofficial implementations. To maintain anonymity, we cannot provide links here that reveal our Github IDs.

R2 : <*It is really impactful because the authors show empirically that it speeds up the training process while maintaining equal robustness...on the other hand, the idea itself isn't really outstanding.*> We feel that the simplicity of implementation is a key attribute that makes our method useful. To add depth to a paper about a simple idea, we tried our best to be thorough in our investigation. We observed the impact of minibatch replay on clean-trained nets, studied the generative properties of free trained models, and validated the absence of gradient masking using landscape visualizations. <*In the future, compare to "YOPO".*> YOPO appeared on arXiv after this paper was drafted, and used different datasets and wider networks (WRN-34-10) with larger batch-sizes (256) than we do. As shown in our supplementary, both of these factors increase robustness, so directly comparing to their arXiv results is difficult. To do a direct comparison, we used our "Free" code (available on GitHub) to train WRN-34-10 using $m = 10$, batch-size=256. We match their best reported result (48.03% against PGD-20 attacks for "Free" training *v.s.* 47.98% for YOPO 5-3).

R3 : <**great* to have error bars on your main tables/results.*> We have started to implement your great suggestion. For $m = 8$ CIFAR-10, the natural accuracy is 85.95±0.14 and robustness against a 20-random restart PGD-20 attack is 46.49±0.19. Producing the error bars takes some time; we will try to have all error bars by the camera-ready deadline. <*Figure 3 might be nice to contrast this with a non-robust model.*> The landscape of Non-robust models are not as smooth and also have large peaks where the classification loss is relatively huge (Fig. 1). <*Code?*> It is publicly available. See response to **R1** above. <*training adversarial on the *entire* Imagenet-11k*> Great suggestion! We will try to add that to the camera-ready version. <*It would be nice to have some intuition or some justification for *why* such a technique works so well.*> Our "free" updates are a coarse approximation of true adversarial training on the first replay step, but they behave more and more like the true adversarial training updates after a few replay steps. In Fig.2, we show that the gradients produced by "free" and "true" adversarial training become almost identical after 3 replay steps. This study will appear in the supplementary.

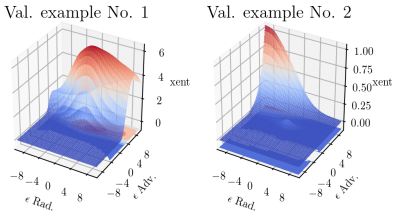

Figure 1: loss landscape of nat trained.

Figure 2: We compute avg(sign($\nabla_x l$)==sign($\hat{\nabla}_x l$)) where $\hat{\nabla}_x l$ is the grad carried over from the previous replay step, and $\nabla_x l$ is from the current step. The updates become almost identical after 3 replay steps.