Thank you for yours detailed reviews. We would first like to address the **general concerns**. Firstly, we agree that the clarity and writing of the paper needs improvement, both in terms of notation and explanation of various terms, and especially in the section on adversarial attack detection. We have taken your comments on board and are currently addressing these issues in order to make the story of the paper far clearer and easier to follow. Secondly, we agree that the experimental evaluation needs improvement. This was mostly due to having legacy tensorflow code that was not suitable for extension. The last few months have been spent writing a cleaner PyTorch implementation which allows for the easier introduction of new architecures and integration with adversarial toolkits (FoolBox/Cleverhans) and allows for a head-to-head comparison of various methods. We are currently reproducing (classification accuracy results presented in table below) the OOD detection experiments using the WideResnet 28x10 architecture, where we are able to match SOTA classification performance on CIFAR-10, CIFAR-100 and TinyImageNet. Additionally, we will add experiments on RKL PNs trained on TinyImageNet in-domain and a 400-class subset of the other 800 imageNet classes, processed like TinyImageNet, as OOD training data. We will evaluate OOD detection on the remaining, heldout subset of 400 ImageNet classes. Third, we will update adversarial attack detection numbers to be on the CIFAR-10, CIFAR-100 and TinyImageNet datasets using the new architecture. Furthermore, having analyzed the Carlini&Wagner L2 attack, we believe that the current adaptive adversarial attack loss functions may be sufficient. An alternative adaptive attack loss function we will also consider is L1 loss between the predicted and permuted logits, which has the same fixed point as KL-divergence minimization, but potentially nicer gradients. This should yield an evaluation against a stronger adaptive whitebox adversary. However, we are unable to demonstrate updated Adversarial Attack Detection numbers in this rebuttal as we haven't integrated with Foolbox yet.

Model	CIFAR-10	CIFAR-100	TinyImageNet
VGG (Error %)	8.0 ± 0.4	30.4 ± 0.6	41.7 ± 0.4
WRN (Error %)	3.9 ± 0.1	19.3 ± 0.5	32.3 $\pm$ NA

Reviewer 2 The OOD results quoted in the papers use a range of classification network architectures. It is thus not appropriate to directly compare numbers. We are planning to do clean comparisons of the method using consistent architecture. Secondly, as presented in this paper, Prior Networks are *not* SOTA (but close) for OOD detection, as the current SOTA (Lee18) results make use of a set of *bespoke* post-processing techniques (such as ODIN) aimed at maximizing OOD detection performance. The same approaches can be applied on top of Prior Networks. The aim of this paper is to present a *general method* (training criterion) and analyze its properties in two different scenarios. Crucially, we show that adversarial training of Prior Networks using the proposed criterion gives a significant improvement over standard adversarial training *at not additional cost*, making it a drop-in replacement. Other techniques can be stacked on top of this. Third, a large perturbation size was selected in order to give more freedom to the adversarial attack to succeed against our detection scheme. In additional results to be completed we will conduct an analysis of the perturbation size on the success of adaptive whitebox attacks. Fourth, perhaps we have misunderstand your comment, but the joint success rate represents the success of *the attack*, rather than *the defense*. Specifically, the JSR represents the success of the attack at *both* successfully attaining the target class *and* avoiding detection. This will be made clearer in the text. Thus, it is unsurprising that an adaptive whitebox adversarial attack will have a very high success rate. Fifth, regarding the dropout attack. We need to make it clear in the text that attacks against dropout are completely undetectable, but have a slightly lower success rate since we generate the attack against the *mean network*, rather than against each of the 10 samples. The added stochasticity makes the attack less successful. This was difficult to address in the legacy code, but should be easier to do in the new implementation. Sixth, while we agree that the computational expense of an adversarial attack can be increased by gradient masking, we think that it is a significant result that we can do the same by essentially using an improved form of adversarial training at no additional expense. We stress that we analyse a general method. Other, more task-specific techniques can always be stacked on top of this. Finally, we use FGSM attack in training because we dynamically generate the attack on each minibatch during training. Using iterative attacks is possible, but expensive, as training would slow down.

Reviewer 3 Firstly, the Reverse KL is exactly equal to the variational criterion, where the reconstruction loss is weighted by β and where the KL-regularization loss is minimization the KL-divergence to a flat Dirichlet Prior. We will make this connection clearer in the paper. Secondly, this choice of OOD training data is quite standard in OOD detection experiments. The main requirement is that it is more diverse than the in-domain data. Third, the effect of varying β is not strong. The main goal is to make sure that the distribution of $\hat{\alpha}_0$ in-domain and OOD are clearly separable. It is necessary to set β to 0 for OOD training data (for OOD detection), as we have no a-priori knowledge about what the target class. However β can be set to 1 for adversarial data, as we do know what the target class should be. This allows the model to learn to *both* predict the correct target class *and* high uncertainty for adversarially perturbed training data. Finally, setting $\gamma = 10.0$ for the RKL loss was chosen based on results of the toy-data experiments. When training models on the other datasets we found that this choice of gamma was consistently better than setting gamma to 1.0 .