We thank all the reviewers for their insightful comments and suggestions. We will update the final version accordingly.

R#1. We thank the reviewer for pointing us to the two relevant references (we will discuss them in the final version), that use low-rank representations to get robustness to specific empirical attacks like PGD. We remark that in our work low rank representations are used in a different way. To provide the improved *certified* robustness guarantees, we take advantage of good low-rank representations in extending the randomized smoothing approach appropriately.

(a) On using non-linear projections: We agree that introducing a non-linear dimensionality operation may lead to classifiers that are more robust against PGD/FGSM style attacks, as observed in Sanyal et al. Our setting is quite different since our aim is to produce classifiers with *certified* accuracy guarantees. Introducing a projection operation at an intermediate layer completely breaks down the theoretical analysis of certified accuracy. Hence, we are restricted to using linear projections in order to provide provable guarantees for our classifier. We believe that the reviewer's suggestion of using/analyzing non-linear projections is an excellent direction for future work.

(b) On training the linear projection vs. using a fixed projection: We agree that in certain cases such as text data where the input representation is not fixed, training the linear projection along with the network could be beneficial and in fact necessary. We are currently exploring this direction. For vision datasets that we used in the paper, we indeed had experimental results with simultaneously training the projection with the network parameters. The results we obtained were similar to using a fixed projection and we did not see any significant advantage. We chose to present the simpler approach to convey the core idea clearly. We will include these results in the final version.

(c) Choice of r: We plot the PCA reconstruction error as a function of r and choose a value of r in a certain range where the error is not too high (less than 3%). See Fig. 5. There are multiple choices of r that work equally well.

(d) Training complexity: The complexity of Algorithm 1 is comparable to the complexity of training a smoothed classifier as in the work of [SYL$^+$19]. The PCA step incurs a one time preprocessing cost and the projection step at the beginning simply corresponds to adding a linear layer to an existing ResNet architecture. As an example, on the CIFAR-10 dataset, for $\epsilon = 0.25$, training the classifier of [SYL$^+$19] takes on average 21.27 seconds per epoch, whereas Algorithm 1 takes 21.29 seconds per epoch on average. The same behavior holds across different parameter settings.

(e) On trading off certified accuracy vs. natural accuracy: Notice that for most values of ϵ, as Fig. 2 shows, we suffer almost no loss in accuracy at small radii. Additionally, as we state in Line 141 (Page 4), for a large range of values for the robustness radius, our method gets much better natural accuracy for a desired robust accuracy and radius. In practice, we may not know the radius of adversarial perturbation (and the ideal choice of ϵ) beforehand, hence sacrificing a small amount of accuracy at small radii for a significant gain at higher radii is a desirable tradeoff.

R#2. In Line 142, thanks for catching the typo: yes we meant blue instead of yellow. We now address the other points.

(a) Scaling PCA to large datasets: One can perform PCA on a smaller random sample to get the projection. Alternately, one can also train the network and the projection operator simultaneously. Formally, reparameterize $\Pi = UU^T$ and augment the loss function with two terms: 1) Reconstruction error on the **mini batch** namely, $\|x - Ux\|^2$; and 2) $\|U^TU - I\|_F^2$ to encourage U to be orthonormal. Our experiments with this approach on CIFAR-10/100 show results similar to those reported in the paper. Further, this approach naturally scales to large datasets.

(b) Certified accuracy vs. r & hyperparameters: Changing r by small amounts had negligible effect on the certified accuracy (we tried values of r close to 200). The smaller we can keep r, the more robustness we can achieve. As mentioned in our response to R#1 (1c), the reconstruction error (see Fig. 5) dictated our choice of $r = 200$. This choice made apriori to the training stage works well across different settings, suggesting that choice of hyperparameters can be decoupled. However, we did not run extensive experiments for end to end training for many different values of r. The performance of our algorithm is smooth in λ (see Fig. 1). Furthermore, in all our experiments $\lambda = 0.5$ worked very well pointing to the fact that there are generic settings of λ that one can often use. We will clarify more in the final version.

(c) On feature dimensionality: We do not know if linear projection forces the intermediate representations to be even lower dimensional. Forcing intermediate representations to be low dimensional may certainly lead to empirical benefits. However, it seems challenging to use this to get certified accuracy guarantees.

R#3. Please see responses to R#1 for the effect of PCA on natural accuracy (1a) and training cost (1d). Using autoencoders is an interesting idea. Currently we do not know how to use them to get certified accuracy guarantees.

R#4. Regarding *new SDP algorithm* we respectfully disagree with the opinion that the approach is straightforward. Please note two main contributions: (i) the algorithm is practical and novel within the broad MWU-framework (with an important change to the weight update) , (ii) the accompanied theoretical guarantee gives a significant improvement over the current SOTA, for an even broader class of general quadratic programs (as discussed in Secs. 3, A, and D). As experimentally shown in Sec. F, the algorithm is much faster than existing solvers, and may be of independent interest.

(a) MWU vs DCT: We are somewhat confused by the reviewer's question about comparing MWU and DCT. These are not two competing approaches. For certified ℓ_∞ guarantees in the DCT domain, these are used together. The MWU based algorithm is used as a subroutine to find a good robust low-rank representation in the DCT domain (the MWU based algorithm is crucial for obtaining certified guarantees).