We thank all the reviewers for their insightful comments and suggestions on improving the paper. Below we respond to specific technical and clarification questions.

On the Definition of Adversarial Example: The upper bounds in the paper consist of two parts, a) algorithms for finding adversarial examples (sections 3,6), and b) algorithms for adversarial PAC learning (section 4). Our algorithms for finding adversarial examples apply to both the definitions, namely when $f(x+z) \neq y$ or $f(x+z) \neq f(x)$. For simplicity we chose to present it for the latter. Then in section 4 we use our algorithms for finding adversarial examples for adversarial PAC learning. Here one needs to define what PAC learning means and in the realizable case we use the same definitions that have been used in [Cullina et al.'18] and [Montasser et al. 19]. Notice that similar to these works, we are in the realizable case and in this scenario the definition of [Bubeck et al'19] also becomes the same since $f^*(x+z) = f^*(x) = y$. Hence in the realizable case the comparison with the work of [Bubeck et al.'19] is indeed fair in our opinion. We do agree with the reviewer that in the non-realizable case these distinctions matter (see [11] for an in-depth discussion), but we make no claim about the non-realizable setting in our paper. In fact, studying under what assumptions efficient adversarial learning in the non-realizable is possible is a very interesting direction for future work and we are currently actively thinking about it. However, we do appreciate the reviewers' comments about this and will clarify more in the final version of the paper.

On the Realizability Assumption: Our goal was to make progress on adversarial PAC learnability which is very poorly understood currently. Hence, a natural starting point is the realizable setting. Notice that without the realizable setting, even standard PAC learning of LTFs is hard, unless one is willing to make strong assumptions about the distribution. Furthermore, in the realizable setting, as long as the classifier $f(x)$ is defined over all of $\mathbb{R}^d$, the assumption about δ-robust error being zero is a very natural one. In fact, for LTFs, this assumption simply boils down to asking for a large ℓ_∞/ℓ_1-margin, a well studied quantity in the learning theory literature. Furthermore, as in the standard margin assumptions, we only need the large margin (i.e., δ-robust error of zero) to hold only on the training set. So in fact our results also extend to the case where the δ-robust error is a suitably small inverse polynomial (since in that case it has no error on any of the training samples w.h.p). We chose to not focus on this extension for ease of presentation. If the reviewers' question concerns classifiers that might not be defined over the entire domain, then our model does not capture that case. In such a scenario, one would then want to model the adversarial perturbations as a general set rather than an ℓ_∞ or an ℓ_p ball. There have been works that have studied adversarial sample complexity for such cases (See Cullina et al.'18), but this model has not been studied from a computational complexity perspective. Finally, algorithms developed for PAC learning of PTFs in the realizable case, e.g., perceptron, SVMs, have found use beyond this particular realizable setting. So we believe our study may help identify robust algorithms that are more broadly applicable.

On the Assumption Needed for Depth-2 Networks: We would like to point that some assumption is necessary to make progress on the problem of finding adversarial examples for depth-2 networks and higher networks, as in the worst case the problem is hard (even to approximate to a reasonable factor). Our current assumption is justified through empirical observations (see Figure 4(a)). For our bounds to hold, we need the LHS of A1 to be roughly larger than log(n). As the figure shows the LHS is, in fact, overwhelmingly high. We have observed the same trend in other datasets. It is possible that one can replace the assumption with a more "natural" one, but our current result is a good starting point. Before our work, no non-trivial bounds were known for finding provable adversarial examples for depth-2 networks.

Other Clarifications: We thank the reviewers for the suggestion regarding Theorem 5.1 and remark 5.2. We went with the current choice due to balancing space constraints and stating a clean theorem statement. In the full version of the paper we will include a more detailed statement and discussion. We also thank the reviewers for the suggestion of exploring as future work the use of Burer-Monteiro style approaches for quickly solving our SDPs. This is especially promising given the effectiveness of our SDP over the projected gradient descent (PGD) method in the experiments.