We will fix all minor comments and typos without explicitly addressing them in the rebuttal.

Response to Reviewer 1:

Practical Impact: Our primary aim in this work is indeed theoretical. There has been substantial interest in the theoretical understanding of adversarial robustness recently. Our work highlights the deficiencies in some of these theoretical formulations (see also response to Reviewer 2 below), which we hope will lead to better theoretical models, which in turn may lead to practical advances. Regarding an algorithm for monotone conjunctions in Theorem 10's setting, the standard PAC learning algorithm for conjunctions suffices. An outline of this already appears in the Appendix, but we will add a reference to it in the main paper.

PAC Terminology: We have assumed that readers will be familiar with standard terminology from PAC learning. Given that many NeurIPS attendees may be unfamiliar with this terminology, we will add an appendix giving definitions that we require and point readers to standard texts for further details.

Non-trivial Class: The definition of non-trivial class appears just before the statement of Theorem 5 (in lines 182-183).

Undefined Algorithm: The algorithm for *exact learning* monotone conjunctions using membership queries would be considered folklore in the computational learning theory world; the key idea is that starting from the instance where all bits are 1 (which is always a positive example), we can test whether each variable is in the target conjunction by setting the corresponding bit to 0 and requesting the label. We will add this in the aforementioned new appendix.

Finite Concept Classes: Since (Thm. 11/Prop. 12) are primarily concerned with showing hardness of robust learning, we don't think finite concept classes is a restriction. Please also see lines 70-90 for discussion regarding concept classes defined over $\mathbb{R}^n$.

Experiments: We do not believe that *artificial* experiments will add to the value of the paper; that's not the main point of the submission.

Comparison to Prior Work/Contributions: We will expand on the section in the paper, but we also refer to the review by R3, which we believe very clearly summarizes our contributions.

Response to Reviewer 2: *Right Model*: We obviously disagree with the reviewer about this being a bad paper, but to a great extent do agree with the reviewer about these being *unsuitable models* or *inadequate definitions* for adversarial robustness. The point is that we *weren't* the inventors of these definitions (cf. [4, 5, 7, 8] for theory papers and others more applied papers [A, B, C]). Our aim was precisely to show that once these definitions are *accepted*, even the most elementary classes prove to be hard to robustly learn—and that proving computational hardness is much easier and straightforward compared to the proofs that appeared in prior work.

Having criticized the definitions, we should acknowledge the contributions of prior work. Indeed, our initial aim was to show positive results for at least some non-trivial classes under these definitions. It is clear that these definitions are in many ways *natural* and *reasonable*, but when one looks at them under the lens of computational learning theory their inadequacies surface immediately. We hope our work will highlight these issues and lead to future work (including hopefully by us) that comes up with definitions that still (somewhat) retain the *simplicity* and *naturalness* of the current definitions, while allowing one to separate non-trivial classes that are easy to robustly learn from those that are not!

Computability/Halting Problem: There is no connection to the halting problem, which is only one of the reasons why uncomputability arises; there can be several others. The difficulty in this case is *enumeration* over (uncountably) infinite sets. How would one compute the function, $\mathbf{1}(\exists y \in B_\rho(x).h(y) \neq c(y))$ in *finite time* even if one had black-box access to evaluate h and c (the latter is not possible without membership queries)? Even under the assumption that the Turing machine has the power to perform arithmetic over reals in unit time, the existential quantifier makes evaluating the robust loss impossible! Even if the instance space is $\mathbb{Q}^n$, the decision problem for detecting an adversarial example would be *recursively enumerable*, but not *recursive*. This problem disappears for finite instance spaces, but even there it is not obvious how to evaluate this loss *without* membership queries. This is why one gets the separation for monotone conjunctions depending on whether or not the learning algorithm has access to membership queries. In the case of infinite instance spaces, we can't see a way to avoid the enumeration question without a strong inductive bias on h and c; in that case, properties of these functions, e.g. Lipschitzness, could be used to compute the loss in finite time.

Response to Reviewer 3:

We thank the reviewer for the comments and will obviously fix the typos they observed.

A. Cisse et al. Parseval Networks. ICML 2017.

B. Madry et al. Towards deep learning models that are resistant to adversarial attacks. ICLR 2018

C. Tramer et al. Ensemble Adversarial Training. ICLR 2018