We thank all reviewers for their feedback. We respond to the main concerns. We will also address all the minor points raised in the final version of the paper.

Motivation / Threat Model / Use Case (Reviewers 1, 2, 3) The main contribution of our paper is to provide a framework that cleanly models privacy against bounded adversaries, and allows **quantitative calculation of privacy loss** against particular adversarial classes. This was previously unknown (except for the special case of computationally bounded adversaries). Quantification of privacy loss is important because it allows a systems-designer to precisely determine the kind of privacy-accuracy tradeoff that is offered by a release.

One setting where quantifying privacy for different adversaries makes sense is when data sharing is coupled with data usage contracts (as mentioned in Section 1 of the paper). For example, an instance of the Laplace mechanism might offer $\epsilon = 1$ in general but $\epsilon = 1.25$ to a certain class of adversaries. Quantifying this tradeoff allows (a) better decisions in cases where we expect the adversaries to be bounded in what they can do – for example, automated adversaries or adversaries under a data-usage contract – and (b) better design of data-usage contracts – eg, if the loss against quadratic functions is much higher than linear, then we can only allow for linear functions.

R3 astutely observes that problems may arise if the adversary does not obey the data usage contract, or its output is viewed by someone else. In this case, we will sacrifice the improved privacy guarantee, but if we use a differentially private mechanism, then we can fall back to the original (weaker) differential privacy guarantee that holds for all adversaries.

We will add this discussion to the introduction in the final version of the paper.

Reviewer 1 "The only thing missing from the paper is a better sense of how adversary types can be linked to function classes, perhaps through an extensive example."

A concrete example is an excellent suggestion; we will do this in the final version.

"Why couldn't the adversary just use simulation and rely on $\mathcal{H}$ instead? Why wouldn't the adversary have access to the output of the cb-DP algorithm?"

To clarify, the adversary here can only compute certain functions (those in $\mathcal{H}$) on the output of the capacity bounded differential privacy algorithm. The above discussion about data usage agreements is one setting where adversaries of this form arise.

We will correct the error pointed out in the Appendix as well as add theorem numbers to our references where applicable.

Reviewer 2 "At this point it feels that the risks (in terms of possible privacy breach) of using this relaxation outweighs the potential benefits."

We emphasize that the main contribution of the paper is quantifying the risk for capacity bounded adversaries. As discussed above, under a mechanism (like a data usage agreement), one can get adversaries to only postprocess the outputs of a mechanism using restricted function classes to get a tighter privacy analysis. If the adversary deviates, one could still fallback to the general (and weaker) DP guarantee.

Reviewer 3 "The main problem here is that the definition - as the submission correctly observes - is not closed under post-processing. In other words, once the (ML or contractually bound) adversary does its computation, _its_ output can be observed and processed by someone else, without restrictions imposed by the definition."

Data use agreements usually restrict the outright release of data or its derivatives. This would ensure that the output of a capacity bounded DP algorithm is not released to an adversary of a different class. That said, if this does happen in violation of the data use agreement, we can still fall back on the (weaker) DP guarantee that applies to all adversaries.

We will qualify our statement on the Groce et. al. paper in Section 1, as was suggested. We will also add the suggested details to the definitions section.