We thank all the reviewers for the thoughtful comments and suggestions. We'll fix typos and make descriptions clear.

@R1, sharpness of the bounds and dimension dependence: Our work focuses on understanding the learner's privacy-efficiency trade-off. Our current results give the query complexity bounds that depend on the desired level of privacy and accuracy. Obtaining a dimension-dependent bound is an important, and ongoing research. *Lower bounds for secure binary search:* We'd like to first clarify that our setting is more towards a Bayesian setting. The bound cited by the reviewer is for a deterministic setting. To compare with [17] (we use the citation number in our submission in this response, [17] is Xu & Yang:2019) in the Bayesian setting, in our proof, we use a Proportional-Sampling (PS) estimator as in [18] (Xu 2018) to characterize the hardness results, while the authors in [17] use a different adversary (*truncated* PS estimator) and obtain sharper bounds. While ours are tight when the learner's error $\epsilon \to 0$ given a fixed $(\epsilon^{\text{adv}}, \delta^{\text{adv}})$. For a concise presentation, our results also hide logarithmic factors which could make a difference in these cases ($\epsilon \to 0$). We will add more discussions and make statements more rigorous.

@R1, terminology, notations and detailed comments: *"secure" vs "private":* We agree that "secure" would be a better choice to deliver our main message and will update accordingly. *ℓ_2-norm:* Yes, our Lipschitzness and uniform convexity properties are defined w.r.t ℓ_2-norm. We recognize a dimension-dependent bound as an imporant future direction. *"The convergence for point error would fail ...":* Yes, we meant to say that the lower bound cannot be bounded. We'll rephrase it. *Line 402, the proof of Lemma 1:* The proof of Lemma 1 should finish at Line 402, as Lemma 1 is stated about function error. The proof then continues (Line 402 – 408) for point error. *Line 429, conditional independency of M and Y_t:* Y_t and M are conditionally independent given the information (X^t, Y^{t-1}, ξ_k), in other words, $M \to (X^t, Y^{t-1}, \xi_k) \to Y_t$ is a Markov chain. We will make these statements/terminology clear.

@R2, applications: We focus on settings in which the learner cares about the accuracy of the their final estimate instead of the utility generated during learning. For example, in federated learning (the example used in [17]), companies may optimize the parameters of their learning models using gradient decent through sequentially broadcasting their models to data-holding users and obtain the gradient information. Adversaries can pretend to be users and estimate the final model through observing the broadcasted models (but not the gradient). As another example, companies may perform market research by sequentially inviting users in different population for interview. Competitors might free-ride the outcome by observing who attended the interview but not the responses.

@R2, (ϵ, δ) and $(\epsilon^{\text{adv}}, \delta^{\text{adv}})$: These two set of parameters cannot be the same and need to satisfy certain conditions. (as mentioned in Footnote 5). Intuitively, the learner has more information than adversary, therefore it would make more sense for adversary to have a weaker requirement. We will include the discussion in the revision.

@R2, difficult instances and adversary: To prove the lower bound, one only needs to find a hard problem instance (and a specific adversary) to prove that *any* algorithm has to use at least sufficient number of queries. While our current constructions of problem instance and adversary are specific, they serve the purpose of proving a lower bound. While one might wonder whether we can find constructions leading to tighter lower bound, since we also provide a matching upper bound, this means that this particular choice of constructions leads to a tight lower bound.

@R2, other comments: Randomization helps to ensure the adversary can't do better by guessing uniformly at random. Pre-defining a deterministic sampling would work as well, but we might need to account for the adversary's prior and belief. Line 195 – "...we adopt them ...", we mean that the two ways in defining PS estimator are essentially the same. Line 149 – "...accurate optimization algorithms.", we mean that an algorithm could efficiently optimize the function up to a small error. Line 187 – "While incorporating a weaker adversary ...", yes, we actually mean stronger adversary (sorry for the typo). *Notations*: We will carefully revise the notations.$\{\theta_1, \ldots, \theta_k\}$ is defined in Line 190: it represents a $2r$-packing set with radius r. $I(\cdot|)$ denotes the conditional mutual information and T in Lemma 1 should be $T_{\mathcal{P}}$.

@R4, organization and Sec 4: Thanks for suggestion! We will adjust the content in Sec 4 to make it more readable.

@R5, comparisons with Differential Privacy (DP): Thanks for the references. Our privacy notion builds on recent works [18, 15, 17] and as [18] points out, these two privacy notions are incompatible. At a high level, DP stands from a universal perspective, where the output distribution of a mechanism is supposed to be insensitive w.r.t any perturbation in the input, and thus the privacy of individuals is protected. In contrast, the current privacy measure is more specific and a goal-oriented one, which captures an adversary's (in)ability to perform a specific statistical inference task: Adversary's goal is to infer an optimizer from observing learner's queries.

@R5, results "to be expected": We respectfully disagree. Under our private learning setting, the adversary also has the complete knowledge of how learner's algorithm operates. Thus, even though the function value at any iterate during learning process is not close to the optimal value, as long as the learner uses these function values to derive a solution under the algorithm and when this is fully informed to the adversary, the adversary can formulate his own estimate for the optimizer via reverse-engineering the algorithm and the trace of queries. The key to prevent this privacy breach is that the learner should *obfuscate* her learning strategy.