We thank all the reviewers for their constructive feedback and address each one below.

Response to Reviewer 1: We emphasize the **novelty** and the **motivation** of our paper below.

1. Re: novelty of our technique: The most relevant (and state-of-the-art) previous work on detecting violation of DP is [11]. We improve over [11] as follows. (i) The method proposed in [11] involves heuristically enumerating over a large number of subsets $E \subseteq [S]$ that is extremely inefficient, whereas our estimator is efficient with complexity linear in $|S|$. (ii) Only we have a theoretical guarantee, and further achieve statistically optimality. (iii) We can estimate more general (ε, δ)-DP, whereas [11] can only check a special case of $(\varepsilon, 0)$-DP.

2. Re: motivating application: One major motivation of this paper is to **detect violation of DP**, whose importance is acknowledged by the best paper award given to [11] in 2018 ACM CCS conference, which study the same problem. We significantly improve the detection, by proposing a principled method as we discussed in the above paragraph. There are several points of failure to designing/implementing DP mechanisms, and a number of published algorithms are incorrect. In this paper, we propose a new approach to finding bugs that cause algorithms to violate differential privacy, and generating counterexamples that illustrate these violations. Such a counterexample generator would be useful in the development cycle in detecting errors and fixing them. This does not necessarily require checking all pairs of neighboring databases, which is infeasible.

Regarding running our estimator on one or a small number of paired neighboring databases, we would like to emphasize the following points. (i) If you have some side information, then this might significantly reduce the search space. For example, if your mechanism is noise adding, then you only need to check two data bases whose true query output is at maximum difference, i.e. the sensitivity. Heuristics on choosing those databases to check have been proposed, for example, in [11], and have been proven effective on real-world mechanisms (which we also demonstrate in Figure 1). Such data driven methods for checking DP guarantees were successfully used in reverse engineering the privacy loss in Apple's DP mechanisms in ["Privacy loss in Apple's implementation of differential privacy on MacOS 10.12," , J. Tang, A. Korolova, X. Bai, X. Wang, and X. Wang, 2017]. (ii) [12] showed that with a relaxed definition of approximate differential privacy called "random approximate differential privacy", we only need to test on randomly selected pairs of databases (non-adaptively) to guarantee privacy. Our estimator can be readily applied to such a scenario.

Response to Reviewer 2: There is a long line of research on estimating functionals of a single discrete distribution, which use similar techniques summarized in [25] for generic functionals. As we build upon similar polynomial approximation techniques, we will better acknowledge [25] in the final revision. However, we want to emphasize that our work diverges from [25] in the sense that we care about a divergence between two distributions, which requires a more careful design of the polynomial approximation. In that sense, our work is more closely related to [31], which we will better acknowledge in the revision as well.

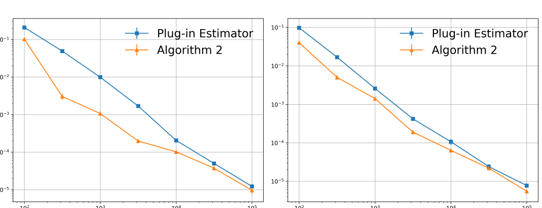
Response to Reviewer 3: We will fix the typos as suggested, and discuss major comments below.

(Re: Poissonization) The choice of Poissonization makes the analysis relatively simpler, and we will state this explicitly. Without Poissonization, the marginal distribution will change from Poisson to Binomial. The minimum variance unbiased estimator should be changed accordingly (for example see ["Bias reduction by taylor series", C.S.Withers, 1987]). With this modified approximation, we believe that the same guarantee might be achievable, but requires more careful analysis on the covariance, as we do not have independence. Getting more samples on symbol i implies getting less samples of other symbols, if we fix the sample size n (as opposed to choosing it from Poisson).

(Re: degree K) We will add the explanation that "Bias scales as $(1/K)\sqrt{(p_i \ln n)/n}$ and variance scales as $(B^K p_i \ln n)/n$, and the optimal trade-off is achieved for $K = c \ln n$ with an appropriate choice of the constant.".

(Re: experiments) We will add more results comparing the plug-in and proposed estimators. For example, the following show results for a different value of $\varepsilon = 0.2$ (left) and different distributions of Zipf and mixture of uniform (right).

MSE



sample size n sample size n

(Re: algorithm 2) We will move some proofs to the appendix and expand the explanation of the Algorithm 2. We will also explain that the division into case 1 and 2 in the range of $(x, e^{\varepsilon} y)$ is necessary. Case 2 is the standard approximation, but when $(p, q) \in [0, 2c_1 \ln n/n]^2$ this approximation fails to provide the desired bias, as in Eq. (149). This is because as both p and q get small, the desired level of bias also gets small, and the standard approximation is no longer sufficient.