Thank you to the reviewers for the constructive and positive comments. Many comments had to do with positioning relative to existing work, which we will clarify in a revision. The comments can be broken into two groups: point estimation methods and Bayesian methods.

Point Estimation

We chose to investigate the ability of methods to produce calibrated posteriors, but agree that evaluation of point estimates is an interesting and important question. Our initial hypothesis was that noise-aware methods will in general perform similarly to their noise-naive counterparts for point estimation. We observed this in preliminary experiments for simple estimation problems (not regression). However, in ongoing applications of our method from this paper, we observe that it produces better point estimates than noise-naive SSP. Furthermore, noise-naive SSP is a competitive and often state-of-the-art baseline for point estimation (Foulds 2016, Wang 2018). We therefore plan on exploring this more deeply in the future.

In general, a Bayesian method is not expected to produce better point estimates than a non-Bayesian counterpart unless the prior is informative.

We also conjecture that noise-aware inference will almost never hurt when it comes to point estimation, when compared to a similar noise-naive algorithm. We also conjecture that it will not help *in general*, which is confirmed by experiments in simple estimation problems. However, there are two possible mechanisms by which noise-aware inference *can* improve point estimation. One is by automatically respecting any constraints on parameters, if present. Another is by avoiding certain pathologies of noise-naive methods. For example, for linear regression, noisy sufficient statistics introduce bias into the traditional least-squares estimator, which is unbiased in the absence of noise introduced for privacy. Our noise-naive Bayesian method may avoid this pathology.

Other Bayesian Methods

As for other Bayesian methods, all are noise-naive. We compared to naive SSP, which we consider to be the most competitive baseline in this setting. The posterior sampling MCMC method due to Wang (ICML 2015) allows public release of many posterior samples (unlike OPS), but still suffers from per-sample privacy loss due to the noise injected in each iteration. Private variational inference (VI) is most relevant for problems where the original posterior inference problem requires VI, i.e., when naive SSP is not an option. Bayesian linear regression permits conjugate updates and therefore we can use naive SSP. In fact, VIPS due to Park et al. uses naive SSP on a per iteration basis within the variational inference framework. It is clearly better to privatize the full sufficient statistics only once as in naive SSP. We will clarify these points in the related work section.

Runtime and Data Size

There are two parameters that may affect runtime: number of individuals and number of dimensions. The plot in Figure 3f addresses the former. Its purpose is to emphasize that the runtime of `MCMC-Ind` scales with the number of individuals, but the runtime of `Gibbs-SS` methods do not, due to their usage of fixed-dimension sufficient statistics. We will clarify that point in the text. The main purpose of the plot was not to compare against `Naive` or to make a statement about runtime for specific practical settings, though that is a question of interest.

While we have not run in-depth experiments on running time relative to covariate dimensionality, we can understand its effect analytically. Let d be the covariate dimension. The most expensive operation used by `Gibbs-SS` will be the inversion of the covariance matrix (defined in Equation 3) in the NormProduct subroutine on Line 4 of Algorithm 1. This matrix has dimension $(d^2 + d + 1) \times (d^2 + d + 1)$, where $d^2 + d + 1$ are the total number of components in the feature function vector $t(\mathbf{x}, y) = [\mathbf{x}\mathbf{x}^T, \mathbf{x}y, y^2]$. The cubic matrix inversion would then be $O((d^2 + d + 1)^3) = O(d^6)$. Modern computing platforms can reasonably invert matrices of size 10K or more, corresponding to linear regression with $d \approx 100$ features.

Misc.

We agree with R1's suggestion to cite McSherry & Mironov's KDD'09 paper as an early work on SSP.

R1 correctly notes that the normalization of the X and y data in the Section 4.3 case study is in fact not a private operation and the normalization constant would have to be publicly released in order to make predictions on new instances. We found this step to be a preprocessing assumption prevalent in many existing works, e.g. the Jalko et al. paper previously mentioned by R3, and one could assume these bounds come from domain knowledge instead of sensitive private data.