[**Reviewer 1- Baselines [37] and [40]**] Thank you for bringing this discussion up. We would like to clarify that, on a high level, the discussion regarding sample complexity (in lines 119-123) of the second-order statistics and third-order statistics applies to both the work in [40] (Zhang et al, 2014) and [37] (Traganitis et al, 2018). However, since [37] uses the third order statistics directly (without grouping the data) like what we do, it is more fair to compare with [37]. Since [40] needs to group the data first and then estimates certain "group third-order statistics", it may need more samples to obtain accurate estimates. We will add one remark on this subtle point in the final version.

[**Reviewer 1 - MATLAB-based Runtime**] We fully agree with the reviewer that MATLAB-based implementations may not exactly reflect the runtime performance in real systems. On the other hand, we hope that the runtime performance in the paper can serve as a useful reference—in case one would like to gain some insights (instead of the exact runtime) on the computational complexities of the algorithms. Nevertheless, we do agree with the reviewer on this point, and will add a remark to notify the readers.

$\kappa(\boldsymbol{A}_m)$	MSE	K	MSE
3.15	0.006	2	0.002
6.33	0.012	3	0.013
10.14	0.033	4	0.021
60.32	0.074	5	0.024
100.82	0.086	6	0.025

Figure 1: Synthetic-data experiments. MSE against $\kappa(\boldsymbol{A}_m)$ and K, respectively. $N = 10^4, p = 1, K = 3$; $\kappa(\boldsymbol{A}_m)$ is controlled by assigning $\boldsymbol{A}_m = \boldsymbol{I}_K + \beta * \texttt{rand}(K, K)$, followed by column normalization and changing β; averaged over 10 random trials.

[**Reviewer 2 - More Insights on The Theorems**] Thank you for this nice suggestion. It is perhaps not easy to directly verify the theorems on real data since some of the problem parameters, such as ε, $\kappa(\boldsymbol{A}_m)$ and $\sigma_{\max}(\boldsymbol{A}_m)$, are hard to acquire. In our experiments, we change the parameter p that directly affects the number of available samples S for estimating the second order statistics; p also affects $T(m)$, i.e., the number of annotators who co-label data with m. To gain more insights, we will also add a number of synthetic-data and real-data experiments in the supplementary materials; see, e.g., Figures 1-2. Again, thanks for this constructive comment.

[**Reviewer 2 - Lines 180-183**] The reviewer is correct: sparser annotator responses yield lower values of S and thus the estimation error bound will get worse. We will re-write this part. In particular, "does not hurt" will be removed.

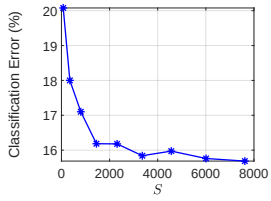

Figure 2: Real-data experiment. Classification error against the number of samples S on UCI 'Adult'.

[**Reviewer 3 - Label Estimation Accuracy**] Thank you for this good point. To analyze the label estimation accuracy, one way is to adopt and modify the results in [40]. To be specific, after model identification, we employ a MAP predictor (see [37,40]) for label estimation. Let y_n denote the true label of sample n. Assume that the conditions in Lemma 11 in [40] hold, and that $\boldsymbol{A}_m(k_m, k) \geq \nu$, for all m, k_m, k. In addition, assume that the MultiSPA-output estimates satisfy $\|\boldsymbol{A}_m - \widehat{\boldsymbol{A}}_m\|_\infty \leq \varphi = \min\left\{\frac{\nu}{2}, \frac{\nu\overline{D}}{16}\right\}$, for all m, where $\overline{D}$ is defined as in [40]. Then, if there exist at least $\widetilde{M} = \frac{4\log 2K}{\overline{D}}$ annotators, the MAP predictor yields $\widehat{y}_n = y_n$ for all n. Also notice that Theorem 2 in [37] can also be modified to characterize the label estimation accuracy using the models output by MultiSPA and MultiSPA-KL.

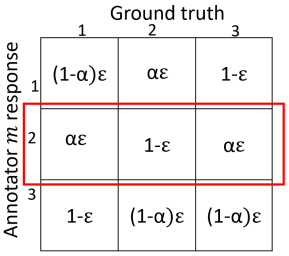

Figure 3: An example where the confusion matrix is specialized for class 2, but not diagonally dominant; $\alpha, \varepsilon \in [0, 1]$.

[**Reviewer 3 - Confusion Matrices without Diagonal Dominance**] Please note that MultiSPA and MultiSPA-KL do not need a particular $\boldsymbol{A}_m$ to be diagonal dominant. It only requires that, among annotators $m_1, \ldots, m_{T(m)}$, there exists at least one annotator who is specialized for class k (i.e, who does not confuse class k with other classes) for every $k = 1, \ldots, K$. Such annotators need not to have diagonal dominant confusion matrices; see Fig. 3. In our implementation, diagonal dominance was only used to fix the column permutation mismatches among the $\widehat{\boldsymbol{A}}_m$'s. But this is not the only way for fixing the mismatches. One can use the method as stated in Sec. D in the supplementary materials that does not need diagonal dominance. The method generally works; e.g., for MultiSPA on the Bluebird data, it outputs a classification error of 12.96% (while using diagonal dominance yields 13.88%); on the Web data, it gives 14.32% (15.22% using diagonal dominance). Nevertheless, we have observed that using diagonal dominance gives constantly good results over different datasets, while the method in Sec. D is not as stable (e.g., on the Dog data, 20.20% classification error v.s. 17.09% using diagonal dominance). Our understanding is that for real data, diagonal dominance is a reasonable assumption, and thus exploiting this structure may be beneficial. We will add these results.

[**Reviewer 3- Minimax-entropy Method**] We have observed that Minimax-entropy is also a strong candidate. However, the performance can be somewhat unstable especially when the annotator response data is very sparse. Our guess is that the objective function of the Minimax-entropy method involves some regularization parameters which are intended to prevent overfitting of the data as pointed out by the authors. For the TREC dataset that is very large but extremely sparse, the algorithm is somewhat sensitive to the regularization parameters—manually finding an 'optimal' regularization parameter is not easy and the results can be very far from being ideal from time to time.