**Reply to Reviewer 1.** Most existing works put a prior on the worker accuracies too. Furthermore, variational mean field is used for the one-coin DS model by Liu *et al.* 2012. The novelty in our paper is how we optimise over the KL divergence in Eq. 5. In order to get an analytical form, we need both the Beta prior *and* a single-data update. Note that, while FBI can be *interpreted* as a form of stochastic GA, it does not correspond to its classic framework.

Other remarks: first, we will add a figure on the label reordering. Second, $F(\bullet)$ and $G(\bullet)$ have a probabilistic interpretation, but it only becomes clear after going through the proofs. Third, MC computes the true posterior. This is optimal when the model assumptions are met (synthetic data), but less robust when they are not (real data). References and implementation details are in Appendix C. Fourth, common practice in crowdsourcing is setting the uninformative prior $\alpha = 2, \beta = 1$. Fifth, the error bars are present but small (see Appendix C). Sixth, refer to our reply to reviewer 2 (point 4). Finally, we perform crowdsourcing when the workers, on average, are correct more often than not.

**Reply to Reviewer 2.** 1. A streaming algorithm allows the use of the US policy. This yields provably better accuracy than non-adaptive policies in crowdsourcing applications (Manino *et al.* 2018). We confirm this for FBI on synthetic data in Figure 1. To the best of our knowledge, there are no publicly available non-static datasets.

2. FBI relies on estimating the worker accuracies given the data observed so far (Eq. 7). If we change the order of the data, the sequence of estimates changes. This can affect the output of the algorithm. We will add a figure to illustrate this.

3. Our assumption holds when either the workers are very accurate or $R$ grows large. For example, in Figure 1b the theoretical error slope is attained for $R$ as small as 30. We will clarify this in the paper.

4. Our algorithm is designed for speed and theoretical guarantees. The empirical analysis shows that its accuracy is similar to the state of the art. We will move the running times to Section 5. Also, we will add the EM benchmark (see table below). On synthetic data, EM is slightly worse than AMF.

| Dataset | KOS | MAJ | MC | Sorted FBI | Fast FBI | AMF | TE | EM |
|---|---|---|---|---|---|---|---|---|
| Birds | 0.278 | 0.241 | 0.341 | 0.298 | 0.260 | 0.278 | **0.194** | 0.278 |
| Ducks | 0.396 | **0.306** | 0.412 | 0.405 | 0.400 | 0.412 | 0.408 | 0.412 |
| RTE | 0.491 | 0.100 | 0.079 | **0.072** | 0.075 | 0.075 | 0.257 | **0.072** |
| TEMP | 0.567 | **0.057** | 0.095 | 0.062 | 0.059 | 0.061 | 0.115 | 0.061 |
| TREC | 0.259 | 0.257 | 0.302 | 0.239 | 0.251 | 0.266 | 0.451 | **0.217** |

**Reply to Reviewer 3.** Referring to Equation 5, the value of $q$ sets the starting value of the log-odds $z_i^0$. If we have prior information, we can use $q$ to bias $z_i^0$. Otherwise, we can set an uninformative prior $q = \frac{1}{2}$ and thus $z_i^0 = 0$. We use the latter in all our experiments.

To work in a streaming setting, Algorithm 2 needs to be implemented differently (see below). There we maintain a separate *view* of the log-odds $\boldsymbol{v}^k$ for each task $k \in M$, and use it to run $|M|$ copies of Fast FBI in parallel. Note that in lines 9-13 we are computing the *real* log-odds by processing all data points we skipped in lines 2-8. This operation has to be performed for every $t$. Since the online version takes more space to explain, we will dedicate an appendix to it.

**Algorithm 1** Sorted FBI (online version)

**Input**: dataset $X$, availability $a$, policy $\pi$, prior $\theta$
**Output**: predictions $\hat{\boldsymbol{y}}$

1: $v_i^k = \log(q/(1-q)),\ \forall i, k \in M$
2: **for** $t = 1$ **to** $T$ **do**
3: $\quad i \leftarrow \pi(t)$
4: $\quad j \leftarrow a(t)$
5: $\quad$ **for all** $k \in M : k \neq i$ **do**
6: $\quad\quad \bar{p}_j^k \leftarrow \frac{\sum_{h \in M_j^t \setminus k} \mathrm{sig}(x_{hj} v_h^k) + \alpha}{|M_j^t \setminus k| + \alpha + \beta}$
7: $\quad\quad v_i^k \leftarrow v_i^k + x_{ij} \log(\bar{p}_j^k / (1 - \bar{p}_j^k))$
8: $\quad z_i^t = \log(q/(1-q)),\ \forall i \in M$
9: $\quad$ **for** $u = 1$ **to** $t$ **do**
10: $\quad\quad i \leftarrow \pi(u)$
11: $\quad\quad j \leftarrow a(u)$
12: $\quad\quad \bar{p}_j^i \leftarrow \frac{\sum_{h \in M_j^t \setminus i} \mathrm{sig}(x_{hj} v_h^i) + \alpha}{|M_j^t \setminus i| + \alpha + \beta}$
13: $\quad\quad z_i^t \leftarrow z_i^t + x_{ij} \log(\bar{p}_j^i / (1 - \bar{p}_j^i))$
14: **return** $\hat{y}_i = \mathrm{sign}(z_i^T),\ \forall i$