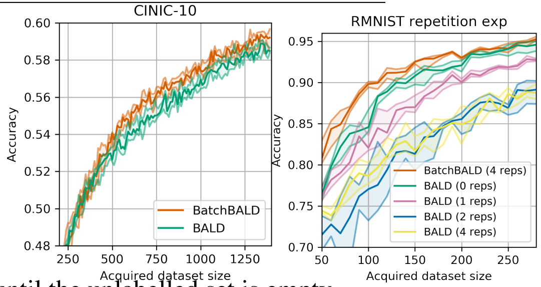
We thank the reviewers for taking the time to write these thorough reviews and their appreciation of BatchBALD as a theoretically grounded way to do batch active learning. We address reviewer 1, 2 and 3 as **R1**, **R2**, **R3**.



R1-(1): Indeed, any acquisition function that does not consider batch dependencies is sensitive to redundant acquisition. Var-Ratios and Mean-Std outperform random on RMNIST due to being more noisy methods than BALD, which helps them in this situation, see also lines 222-224 in the paper.

R1-(2): We agree that accuracy is the main metric. We want to highlight that batch acquisitions significantly reduce the number of times the oracle needs to be consulted and the model retrained. Figure 4 and 5 show that BatchBALD allows for batch acquisitions without losing accuracy or incurring extra computation.

R1-(3): We have updated the figure 8 with both sides sorted. The entropy can be found in figure 7. Training is performed until a suitable accuracy is reached, not until the unlabelled set is empty.

R1-(4): We will reword section 6. Using mutual information encourages acquiring a dataset where the model is informed equally about each class. For severely unbalanced test sets, the model does not have the right bias and will under-perform. **R1-(5)**: We use 25%, 75% quartiles for the shaded areas, see line 147 in the paper.

R1-improvements: No improvements mentioned.

R2 - Originality: Thank you for pointing us to additional relevant related work: we have added citations.

R2 - Significance - a: EMNIST, with its 47 classes and 112,800 data points, is substantially more difficult for active learning than regular MNIST. We provide additional results on CINIC-10 (top figure, left). We use 160k training samples and 20k validation samples, 90k test samples. We use an ImageNet pretrained VGG-16, with a dropout layer before a 512 HU (instead of 4096) fc layer. We use 50 MC dropout samples, acquisition size 10 and 6 trials. The results are in the figure above, with the 59% mark reached at 1170 for BatchBALD and 1330 for BALD (median).

R2 - Significance - b: We have considered the proposed methods: **CoreSet** uses very large acquisition sizes which is unrealistic for many applications; **Variational Adversarial Active Learning** is a semi-supervised learning method, and we consider using BatchBALD in that setting as future work; **DeepFool Active Learning (DFAL), Expected Gradient Length** reported for MNIST that they achieve 83% and 84% (DFAL) and 59% and 38% (EGL) accuracy after 100 samples with LeNet5 and VGG-8 respectively, while BatchBALD obtains 90% accuracy with only 90 samples with a smaller model, see table 1, line 196. We will add these baselines to our table.

R2 - Significance - c: We perform an ablation study on RepeatedMNIST (top figure, right): we vary repetitions from 0 to 4 (paper reports 2 repetitions, 0 still adds noise to MNIST), same setup as in section 4.1. BatchBALD performs the same on all repetition numbers (100 data points till 90%). BALD achieves 90% accuracy at 120 data points (0 repetitions), 160 data points (1 repetition), 280 data points (2 repetitions), 300 data points (4 repetitions). This shows that BALD and BatchBALD behave as expected. We will add the corresponding plot to the paper.

R2 - Quality: We understand the concern about the approximations. For any interesting dataset, we can't compute the joint MI exactly. Therefore, we rely on our mathematical result, with proof in Appendix A., which bounds the approximation. For Monte Carlo sampling, we have conducted further investigation into the variance of the estimator and concluded that, using consistent dropout masks, line 152 of the paper, and the used batch sizes, the variance doesn't affect the order of the top points. We will add the analysis and results to the appendix.

R2 - Clarity: We will restructure the methods section. We appreciate the time you took to look at the code. We attempt to make your point in line 126, where we state we can sample once at the beginning of the algorithm. We will emphasize that point in lines 127-131 for clarity.

R2 - Improvements: We address all requests: we add results for the CINIC-10 (a combination of CIFAR-10 and down-scaled ImageNet) dataset you proposed, see **R2 - Significance - a**. We added more comparisons in **R2 - Significance - b** and perform an ablation on the repeated MNIST setup in **R2 - Significance - c**.

R3 - cons 1: Indeed, the quantity in (10) becomes very large as n increases. In fact, the number of configurations $\hat{y}_{1:n}$ is C^n. In section 3.4.1 we discuss two methods to reduce the computational load: efficiently caching previous results (possible due to the greedy approach) and beyond the first 4 acquisitions, we use Monte Carlo sampling with m samples. In our setup, m is chosen according to our computational budget (10,000), while n is varied per problem.

R3 - cons 2: See **R2 - Significance - a** (eval on CINIC-10, a combination of CIFAR-10 and down-scaled ImageNet).

R3 - clarity: You are correct in saying the duplicated ("repeated") samples are generated by adding Gaussian noise: $x' = x + N(0, 0.1I)$. We don't explicitly ignore or remove duplicates. If we acquire a particular point, then its near duplicates (with added Gaussian noise), will not increase our information about the true parameters, and BatchBALD will prefer other points instead.

R3 - improvements We address all requests: 1. see **R3 - clarity**. 2. see **R3 - cons 1**. 3. Mean and stddev: BatchBALD - MNIST acq size 10: 90% acc at 91.7 with $\sigma = 16.7$, 95% acc at 211.7 with $\sigma = 27.3$; BALD - MNIST acq size 10: 90% at 148.33 with $\sigma = 29.1$, 95% acc at 273.33 with $\sigma = 26.9$. All results are significant with $\alpha = 0.05$. 4. See **R3 - cons 2**.