Stochastic Batch Acquisition:
A Simple Baseline for Deep Active Learning

Anonymous authors
Paper under double-blind review

Abstract

We examine a simple stochastic strategy for adapting well-known single-point acquisition functions to allow batch active learning. Unlike acquiring the top-K points from the pool set, score- or rank-based sampling takes into account that acquisition scores change as new data are acquired. This simple strategy for adapting standard single-sample acquisition strategies performs just as well as compute-intensive state-of-the-art batch acquisition functions, like BatchBALD or BADGE while using orders of magnitude less compute. In addition to providing a practical option for machine learning practitioners, the surprising success of the proposed method in a wide range of experimental settings raises a difficult question for the field: are expensive batch acquisition methods pulling their weight?

1 Introduction

Active learning is a widely used strategy for efficient learning in settings where unlabelled data are plentiful, but labels are expensive (Atlas et al., 1990; Settles, 2010). For example, labels for medical image data may require highly trained annotators, and when labels are the results of scientific experiments, each one can require months of work. Active learning uses information about unlabelled data and the current state of the model to acquire labels for those samples that are most likely to be informative.

While many acquisition schemes are designed to acquire labels one at a time (Houlsby et al., 2011; Gal et al., 2017), recent work has highlighted the importance of *batch acquisition* (Kirsch et al., 2019; Ash et al., 2020). Acquiring in a batch lets us parallelise labelling. For example, we could hire hundreds of annotators to work in parallel or run more than one experiment at once. Batch acquisition also saves compute as single-point selection also incurs the cost of retraining the model for every new data point.

Unfortunately, existing batch acquisition schemes are computationally expensive (Table 1). Intuitively, this is because batch acquisition schemes face combinatorial complexity when accounting for the interactions between possible acquisition points. Recent works (Ash et al., 2020; 2021) trade off a principled motivation with various approximations to remain tractable. A commonly used, though extreme, heuristic is to take the top-K highest scoring points from an acquisition scheme designed to select a single point.

This paper introduces a simple baseline for batch active learning that is competitive with methods that cost orders of magnitude more across a wide range of experimental contexts. Our method is motivated by noticing that single-acquisition score methods such as BALD (Houlsby et al., 2011) act as a noisy proxy for future acquisition scores, see also Figure 1. This observation leads us to stochastically acquire points following a distribution determined by the single-acquisition scores. Our method matches the prior state of the art for batch acquisition despite being very simple. Indeed, our acquisition scheme has a time complexity of only $\mathcal{O}(M \log K)$ in the pool size M and acquisition size K, just like top-K acquisition.

We show empirically that our stochastic strategy performs as well or better than top-K acquisition with almost identical computational cost on several commonly used acquisition scores, making it a strictly-better batch strategy. Strikingly, the empirical comparisons between our stochastic strategy and SotA methods cast doubt on whether they function as well as claimed.

Concretely, in this paper we:

Table 1: *Acquisition runtime (in seconds, 5 trials, ± s.d.).* Our stochastic acquisition methods are as fast as top-K, and **orders of magnitude** faster than BADGE or BatchBALD. Synthetic pool set with $M = 10,000$ pool points with 10 classes. BatchBALD and BALD with 20 parameter samples.

K	Top-K	**Ours**	BADGE	BatchBALD
10	0.2 ± 0.0	0.2 ± 0.0	9.2 ± 0.3	566.0 ± 17.4
100	0.2 ± 0.0	0.2 ± 0.0	82.1 ± 2.5	$5,363.6 \pm 95.4$
500	0.2 ± 0.0	0.2 ± 0.0	409.3 ± 3.7	$29,984.1 \pm 598.7$

- examine a computationally cheap stochastic batch acquisition strategy;
- demonstrate that this strategy is preferable to the commonly used top-K acquisition heuristic; and
- identify the failure of existing SoTA batch acquisition strategies to outperform this vastly cheaper and more heuristic strategy.

In §2, we present active learning notation and commonly used acquisition functions. We propose our stochastic extensions in §3, relate them to previous work in §4, and validate them empirically in §5 on various datasets, showing that our method is competitive with much more complex ones despite being orders of magnitude computationally cheaper. Finally, we validate some of the underlying theoretical motivation in §6 and discuss limitations in §7.

2 Problem setting

Our method applies to batch acquisition for active learning in a pool-based setting (Settles, 2010) where we have access to a large unlabelled *pool* set, but we can only label a small subset of the points. The challenge of active learning is to use what we already know to pick which points to label in the most efficient way. Generally, we want to avoid labelling points similar to those already labelled.

Notation. Following Farquhar et al. (2021), we formulate active learning over *indices* instead over datapoints. This simplifies the notation. The large, initially fully unlabelled, pool set containing M input points is

$$\mathcal{D}^{\text{pool}} = \{x_i\}_{i \in \mathcal{I}^{\text{pool}}}, \tag{1}$$

where $\mathcal{I}^{\text{pool}} = \{1, \ldots, M\}$ is the initial full index set. We initialise a training dataset with N_0 randomly selected points from $\mathcal{D}^{\text{pool}}$ by acquiring their labels, y_i,

$$\mathcal{D}^{\text{train}} = \{(x_i, y_i)\}_{i \in \mathcal{I}^{\text{train}}}, \tag{2}$$

where $\mathcal{I}^{\text{train}}$ is the index set of $\mathcal{D}^{\text{train}}$ containing N_0 indices between 1 and M. A model of the predictive distribution, $\mathrm{p}(y \mid x)$, can then be trained on $\mathcal{D}^{\text{train}}$.

Active Learning. At each acquisition step, we select additional points for which to acquire labels. Although many methods acquire one point at a time (Houlsby et al., 2011; Gal et al., 2017), one can alternatively acquire a whole batch of K examples. An acquisition function a takes $\mathcal{I}^{\text{train}}$ and $\mathcal{I}^{\text{pool}}$ and returns K indices from $\mathcal{I}^{\text{pool}}$ to be added to $\mathcal{I}^{\text{train}}$. We then label those K datapoints and add them to $\mathcal{I}^{\text{train}}$ while making them unavailable from the pool set. That is,

$$\mathcal{I}^{\text{train}} \leftarrow \mathcal{I}^{\text{train}} \cup a(\mathcal{I}^{\text{train}}, \mathcal{I}^{\text{pool}}), \tag{3}$$

$$\mathcal{I}^{\text{pool}} \leftarrow \mathcal{I}^{\text{pool}} \setminus \mathcal{I}^{\text{train}}. \tag{4}$$

A common way to construct the acquisition function is to define some scoring function, s, and then select the point(s) that score the highest.

---

[0] Work done while there.

Table 2: *Summary of stochastic acquisition variants.* Perturbing the scores s_i themselves with $\epsilon_i \sim \text{Gumbel}(0;\beta^{-1})$ i.i.d. yields a softmax distribution. Log-scores result in a power distribution, with assumptions that are reasonable for active learning. Using the score-ranking, r_i finally is a robustifying assumption. β is included for completeness; we use $\beta := 1$ in our experiments—except for the ablation in §6.1.

Perturbation	Distribution	Probability mass
$s_i + \epsilon_i$	Softmax	$\propto \exp \beta s_i$
$\log s_i + \epsilon_i$	Power	$\propto s_i^{\beta}$
$-\log r_i + \epsilon_i$	Soft-rank	$\propto r_i^{-\beta}$

Probabilistic Model. We assume classification with inputs X, labels Y, and a discriminative classifier $\mathrm{p}(y \mid x)$. In the case of Bayesian models, we further assume a subjective probability distribution over the parameters, $\mathrm{p}(\omega)$, and we have $\mathrm{p}(y \mid x) = \mathbb{E}_{\mathrm{p}(\omega)}[\mathrm{p}(y \mid x, \omega)]$.

BALD. One popular scoring function is *BALD* (Houlsby et al., 2011) which uses a Bayesian model and computes the expected information gain between the predictive distribution and the parameter distribution $\mathrm{p}(\omega \mid \mathcal{D}^{\text{train}})$. For each candidate pool index, i, with mutual information, I, and entropy, H, the score is

$$\begin{aligned} s_{\text{BALD}}(i;\mathcal{I}^{\text{train}}) &:= \mathrm{I}[Y;\Omega \mid X = x_i, \mathcal{D}^{\text{train}}] \\ &= \mathrm{H}[Y \mid X = x_i, \mathcal{D}^{\text{train}}] - \mathbb{E}_{\mathrm{p}(\omega \mid \mathcal{D}^{\text{train}})}[\mathrm{H}[Y \mid X = x_i, \omega]]. \end{aligned} \tag{5}$$

Entropy. Another popular scoring function is the *(predictive) entropy* (Gal et al., 2017). It does not require Bayesian models, unlike BALD, and performs worse for data with high observation noise. It is identical to the first term of the BALD score

$$s_{\text{entropy}}(i;\mathcal{I}^{\text{train}}) := \mathrm{H}[Y \mid X = x_i, \mathcal{D}^{\text{train}}]. \tag{6}$$

Acquisition Functions. These scoring functions were introduced for single-point acquisition:

$$a_s(\mathcal{I}^{\text{train}}) := \underset{i \in \mathcal{I}^{\text{pool}}}{\arg\max}\, s(i;\mathcal{I}^{\text{train}}). \tag{7}$$

For deep learning in particular, single-point acquisition is computationally expensive, and it was assumed deep learning models hardly change after adding a single new point to the training set (but c.f. Figure 1). Thus, single-point acquisition functions were trivially expanded to acquisition batches: the most commonly used batch acquisition function naively selects the highest K scoring points

$$a_s^{\text{batch}}(\mathcal{I}^{\text{train}};K) := \underset{I \subseteq \mathcal{I}^{\text{pool}}, |I|=K}{\arg\max} \sum_{i \in I} s(i;\mathcal{I}^{\text{train}}). \tag{8}$$

Some acquisition functions are explicitly designed for batch acquisition (Kirsch et al., 2019; Ash et al., 2020). They try to account for the interaction between points, which can improve performance relative to simply selecting the top-K scoring points. However, existing methods are computationally expensive. For example, BatchBALD rarely scales to acquisition sizes of more than 5–10 points (Kirsch et al., 2019); see Table 1.

3 Method

We observe that selecting the top-K points at acquisition step t amounts to the assumption that the informativeness of these points is independent of each other. Imagine adding the top-K points at a given acquisition step t to the training set one at a time. Each time, you retrain the model. Of course, the acquisition scores for the models trained with these additional points will be different from the first set of scores. After all, the purpose of active learning is to add the *most informative* points—those that will update the model the most. Yet selecting a top-K batch in one step implicitly assumes that the score ranking will not change due to these points. This is clearly wrong. We provide empirical confirmation that, in fact, the

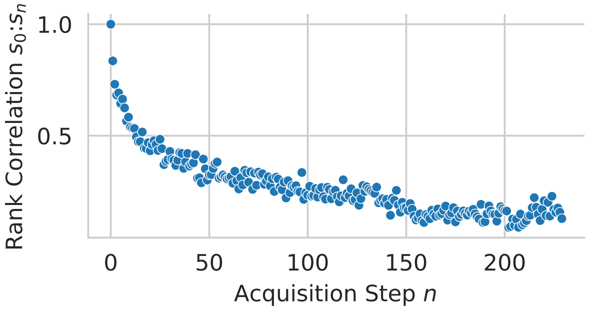


Figure 1: *Early acquisition scores are only a loose proxy for later scores.* Specifically, the Spearman rank-correlation between acquisition scores on the first and n'th time-step falls with n. While top-K acquisition incorrectly implicitly assumes their rank-correlation remains 1 our method does not. BNN trained on MNIST at initial 20 points and 73% initial accuracy, score ranks over test set.

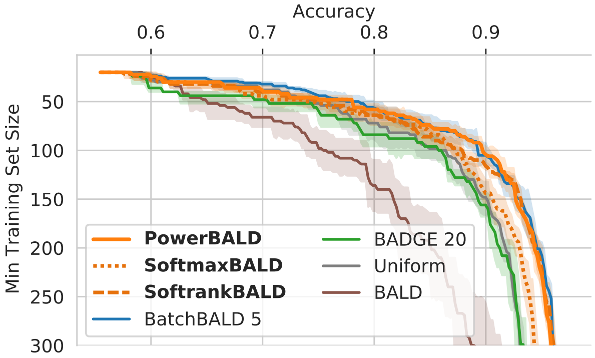


Figure 2: *Performance on Repeated-MNIST with 4 repetitions (5 trials).* **Up and to the right is better (↗).** Our PowerBALD outperforms (top-K) BALD and BADGE and is on par with BatchBALD. This is despite being orders of magnitude faster. Acquisition sizes: BatchBALD–5, BADGE–20, others–10. See Figure 8 in the appendix for an ablation study of BADGE's acquisition size.

ranking of acquisition scores at step t and $t+K$ is decreasingly correlated as K grows; see Figure 1. Moreover, this effect is the strongest for the most informative points; see §6 for more details.

Instead, our work uses stochastic sampling to acknowledge the uncertainty within the batch acquisition step using a simple noise process model governing how scores change. We examine three simple stochastic extensions of single-sample scoring functions $s(i;\mathcal{I}^{\text{train}})$ that make slightly different assumptions. These methods are compatible with conventional active learning frameworks that typically take the top-K highest scoring samples. For example, it is straightforward to adapt entropy, BALD, and other scoring functions for use with our proposed methods.

These stochastic acquisition distributions assume that future scores differ from the current score by a perturbation. We model the noise distribution of this perturbation as the addition of Gumbel-distributed noise $\epsilon_i \sim \text{Gumbel}(0;1)$, which is used frequently for modelling extrema.

The choice of a Gumbel distribution for the noise is one of mathematical convenience, in the spirit of providing a simple baseline. For example, the maximum of sets of many other standard distributions, such as the Gaussian distribution, is not analytically tractable.

Taking the highest-scoring points from this perturbed distribution is equivalent to sampling from a softmax distribution[1] without replacement with a 'coldness' parameter $\beta \geq 0$, which represents the expected rate at which the scores change as more data is acquired.

This follows from the Gumbel-Max trick (Gumbel, 1954; Maddison et al., 2014) and, more specifically, the Gumbel-Top-K trick (Kool et al., 2019). We provide a short proof in appendix B.2. Expanding on Maddison et al. (2014):

Proposition 3.1. *For scores s_i, $i \in \{1,\ldots,n\}$, and $k \leq n$ and $\beta > 0$, if we draw $\epsilon_i \sim$ Gumbel$(0;\beta^{-1})$ independently, then* $\arg\operatorname{top}_k\{s_i+\epsilon_i\}_i$ *is an (ordered) sample without replacement from the categorical distribution* $\text{Categorical}(\exp(\beta\, s_i)/\sum_j \exp(\beta\, s_j), i \in \{1,\ldots,n\})$.

In the spirit of providing a simple and surprisingly effective baseline without hyperparameters, we fix $\beta := 1$. For $\beta \to \infty$, this distribution will converge towards top-K acquisition. Whereas for $\beta \to 0$, it will converge towards uniform acquisition. We examine ablations of β in §6.1.

---

[1] Also known as Boltzmann/Gibbs distribution.

We apply the perturbation to three quantities in the three sampling schemes: the scores themselves, the log scores, and the rank of the scores. Perturbing the log scores assumes that scores are non-negative and uninformative points should be avoided. Perturbing the ranks can be seen as a robustifying assumption that requires the relative scores to be reliable but allows the absolute scores to be unreliable. We summarise the three versions with their associated sampling distributions are in Table 2.

Soft-Rank Acquisition. The first variant makes no assumptions on whether the acquisition scores are meaningful and relies on their rank order instead and thus uses uses the *least* amount of information from the acquisition scores. It is potentially valuable when the variance of the estimated scores is high or when the *relative score order* is reliable but the *absolute* scores is not. However, if the absolute scores are accurate, we would expect this method to perform worse than the other variants below as it throws away the values of the actual scores.

Ranking the scores $s(i; \mathcal{I}^{\text{train}})$ with descending ranks $\{r_i\}_{i \in \mathcal{I}^{\text{pool}}}$ such that $s(r_i; \mathcal{I}^{\text{train}}) \geq s(r_j; \mathcal{I}^{\text{train}})$ for $r_i \leq r_j$ and smallest rank being 1, we sample index i with probability $\mathrm{p}_{\text{softrank}}(i) \propto r_i^{-\beta}$ with coldness β. This is invariant to the actual scores. We can draw $\epsilon_i \sim \text{Gumbel}(0; \beta^{-1})$ and create a perturbed 'rank'

$$s^{\text{softrank}}(i; \mathcal{I}^{\text{train}}) := -\log r_i + \epsilon_i. \tag{9}$$

Taking the top-K samples is now equivalent to sampling without replacement from the rank distribution $\mathrm{p}_{\text{softrank}}(i)$.

Softmax Acquisition. The next simplest variant uses the actual scores instead of the ranks. Again, it perturbs the scores by a Gumbel-distributed random variable $\epsilon_i \sim \text{Gumbel}(0; \beta^{-1})$

$$s^{\text{softmax}}(i; \mathcal{I}^{\text{train}}) := s(i; \mathcal{I}^{\text{train}}) + \epsilon_i. \tag{10}$$

However, this makes no assumptions about the semantics of the absolute values of the scores: the softmax function is invariant to constants shifts. Hence, the sampling distribution will only depend on the scores relative to each other and not their absolute value.

Power Acquisition. For many scoring functions, the scores are non-negative, and a score close to zero means that the sample is not informative in the sense that we do not expect it will improve the model—we do not want to sample it. This is the case with commonly used score functions such as BALD and entropy. BALD measures the expected information gain. When it is zero for a sample, we do not expect anything to be gained from acquiring a label for that sample. Similarly, entropy is upper-bounding BALD, and the same consideration applies. This assumption also holds ideally for other scoring functions that are easily transformed to be non-negative; see appendix B.1. To take this into account, the last variant models the future log scores as perturbations of the current log score with Gumbel-distributed noise

$$s^{\text{power}}(i; \mathcal{I}^{\text{train}}) := \log s(i; \mathcal{I}^{\text{train}}) + \epsilon_i. \tag{11}$$

By Proposition 3.1, this is equivalent to sampling from a power distribution

$$\mathrm{p}_{power}(i) \propto \left(\frac{1}{s(i; \mathcal{I}^{\text{train}})} \right)^{-\beta}. \tag{12}$$

This may be seen by noting that $\exp(\beta \log s(i; \mathcal{I}^{\text{train}})) = s(i; \mathcal{I}^{\text{train}})^{\beta}$. Importantly, as scores $\to 0$, the (perturbed) log scores $\to -\infty$ and will have probability mass $\to 0$ assigned. This variant takes the absolute scores into account and avoids data points with score 0.

In Summary. Given the above considerations, when using BALD, entropy, and other appropriate scoring functions, power acquisition is the most sensible. Thus, we expect it to work best. Indeed, we find this to be the case in the toy experiment on Repeated-MNIST Kirsch et al. (2019) depicted in Figure 2. However, even soft-rank acquisition works well in practice, suggesting that the choice of score perturbation is not critical for its effectiveness; see also appendix §D for a more in-depth comparison. In the rest of the main paper, we focus on power acquisition, we include results for all methods in §C.

4 Related work

Researchers in active learning (Atlas et al., 1990; Settles, 2010) have identified the importance of *batch* acquisition as well as the failures of top-K acquisition using straightforward extensions of single-sample methods in a range of settings including support vector machines (Campbell et al., 2000; Schohn & Cohn, 2000; Brinker, 2003; Guo & Schuurmans, 2008), GMMs (Azimi et al., 2012), and neural networks (Sener & Savarese, 2018; Kirsch et al., 2019; Ash et al., 2020; Baykal et al., 2021). Many of these methods aim to introduce structured diversity to batch acquisition that accounts for the *interaction* of the points acquired in the learning process. In most cases, the computational complexity scales poorly with the acquisition size (K) or pool size (M), for example because of the estimation of joint mutual information (Kirsch et al., 2019), the $\mathcal{O}(KM)$ complexity of using a k-means++ initialisation scheme (Ash et al., 2020), or the $\mathcal{O}(M^2 \log M)$ complexity of methods based on K-centre coresets (Sener & Savarese, 2018) (although heuristics and continuous relaxations can improve this somewhat). In contrast, our method has the same complexity $\mathcal{O}(M \log K)$ as naive top-K batch acquisition, yet outperforms it and performs on par with above more complex methods. This is achieved because our method encourages diversity in the batch through stochastic sampling rather than via distances.

For multi-armed bandits, it has been shown that adding noise to the scores, specifically via Thompson sampling, is effective for choosing informative batches (Kalkanli & Ozgur, 2021). Similarly, in reinforcement learning, stochastic prioritisation has been employed as *prioritized replay* (Schaul et al., 2016) which may be effective for reasons analogous to those motivating our approach.

While stochastic sampling has not been extensively explored for acquisition in active learning, it is used as an auxiliary step in diversity-based active learning methods that rely on clustering as main mechanism (Ash et al., 2020; Citovsky et al., 2021). In a different vain, Farquhar et al. (2021) propose stochastic acquisition as part of de-biasing actively learned estimators. Kirsch et al. (2019) note empirically that additional noise in scores seems to benefit batch acquisition, without further investigation. Thus, while stochastic sampling is generally well-known within acquisition functions, to our knowledge, this work is the first to focus on simple stochastic sampling methods entirely as alternatives to naive top-K acquisition and to compare them to more complex approaches.

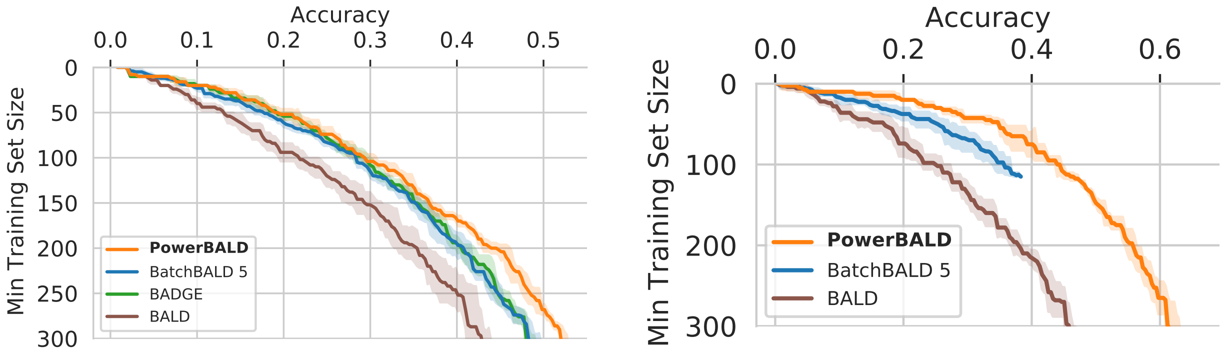
5 Experiments

In this section, we empirically verify that our stochastic acquisition methods (a) outperform top-K acquisition and (b) are competitive with specially designed batch acquisition schemes like BADGE (Ash et al., 2020) and BatchBALD (Kirsch et al., 2019); and are vastly cheaper than these more complicated methods.

To demonstrate the seriousness of the possible failings of current SoTA batch acquisition methods, we use a wide range of active learning data contexts. These experiments show that the performance of our method is not dependent on the specific characteristics of any particular dataset. Our experiments include computer vision, natural language processing (NLP), and causal inference (in §6.1). We show that stochastic acquisition helps avoid selecting redundant samples on Repeated-MNIST (Kirsch et al., 2019), examine performance in active learning for computer vision on EMNIST (Cohen et al., 2017), MIO-TCD (Luo et al., 2018), Synbols (Lacoste et al., 2020), and CLINC-150 (Larson et al., 2019) for intent classification in NLP. MIO-TCD is especially close to real-world datasets in size and quality. In appendix C.5, we further investigate edges cases using the Synbols dataset under different types of biases and aleatoric uncertainty.

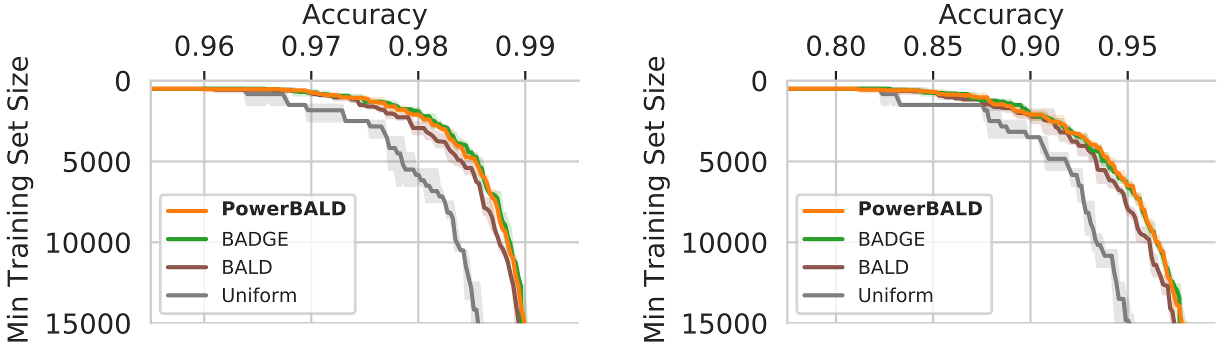
Here, we consider both BALD and predictive entropy as scoring functions. We examine other scoring functions on Repeated-MNIST in appendix C.2.1 and observe similar results. For the sake of legible figures, we focus on power acquisition in this section, as it fits BALD and entropy best: the scores are non-negative, and zero scores imply uninformative samples. We show that all three methods (power, softmax, softrank) perform similarly in appendix D.

We are not always able to compare to BADGE and BatchBALD because of computational limitations of those methods. BatchBALD is computationally infeasible for large acquisition sizes (> 10) because of time constraints, cf. Table 1. When possible, we use BatchBALD with acquisition size 5 as baseline. Similarly, BADGE runs out of memory for large dataset sizes, such as EMNIST ‘ByMerge’ with 814,255 examples.



(a) EMNIST (Balanced) (5 trials)

(b) EMNIST (ByMerge) (5 trials)



(c) MIO-TCD (3 trials)

(d) Synbols with minority groups (10 trials)

Figure 3: *Performance on various datasets.* BatchBALD took infeasibly long on these datasets & acquisition sizes. **(a)** *EMNIST 'Balanced':* On 132k samples, PowerBALD (acq. size 10) outperforms BatchBALD (acq. size 5) and BADGE (acq. size 40). **(b)** *EMNIST 'ByMerge':* On 814k samples, PowerBALD (acq. size 10) outperforms BatchBALD (acq. size 5). BADGE (not shown) OOM'ed, and BatchBALD took > 12 days for 115 acquisitions. **(c)** *MIO-TCD:* PowerBALD performs better than BALD and on par with BADGE (all acq. size 100). **(d)** *Synbols with minority groups:* PowerBALD performs on par with BADGE (all acq. size 100).

Figures interpolate linearly between available points, and we show 95% confidence intervals.

Experimental Setup & Compute. We document the experimental setup and model architectures in detail in appendix C.1. Our experiments used about 25,000 compute hours on Titan RTX GPUs.

Runtime Measurements. We emphasize that our method is computationally efficient compared to specialised batch-acquisition approaches like BADGE and BatchBALD. Runtimes, shown in Table 1, are essentially identical for top-K and our stochastic acquisition. Both are orders of magnitude faster than BADGE and BatchBALD even for small batches. Unlike those methods, ours scales *linearly* in pool size and *logarithmically* in acquisition size. Runtime numbers do not include the cost of retraining models (identical in each case). The runtimes for top-K and stochastic acquisition appear constant over K because the execution time is dominated by fixed-cost memory operations. The synthetic dataset used for benchmarking has 4,096 features, 10 classes, and 10,000 pool points.

Repeated-MNIST. Repeated-MNIST (Kirsch et al., 2019) duplicates MNIST a specified number of times and adds Gaussian noise to prevent perfect duplicates. Redundant data are incredibly common in industrial applications but are usually removed from standard benchmark datasets. The controlled redundancies in the dataset allow us to showcase pathologies in batch acquisition methods. We use an acquisition size of 10 and 4 dataset repetitions.

Figure 2 shows that PowerBALD outperforms top-K BALD. While much cheaper computationally, cf. Table 1, PowerBALD also outperforms BADGE and even performs on par with BatchBALD, which was SotA for small batch sizes. For BatchBALD, we use an acquisition size of 5, and for BADGE of 20. BatchBALD performs better for smaller acquisition sizes while BADGE counterintuitively performs better for larger ones; see Figure 8 in the appendix for an ablation.

Computer Vision: EMNIST. EMNIST (Cohen et al., 2017) contains handwritten digits and letters and comes with several splits: we examine the 'Balanced' split with 131,600 samples in Figure 3a[2] and the 'ByMerge' split with 814,255 samples in Figure 3b. Both have 47 classes. We use an acquisition size of 5 for BatchBALD, of 40 for BADGE, and of 10 otherwise.

We see that our methods outperform BatchBALD on it and both BADGE and BatchBALD on 'Balanced' (Figure 3a) and do not have any issues with huge pool sets on 'ByMerge' (Figure 3b). For 'ByMerge', BADGE ran out of memory on our machines, and BatchBALD took more than 12 days for 115 acquisitions when we halted execution.

Computer Vision: MIO-TCD. The Miovision Traffic Camera Dataset (MIO-TCD) (Luo et al., 2018) is a vehicle classification and localisation dataset with 648,959 images designed to exhibit realistic data characteristics like class imbalance, duplicate data, compression artefacts, varying resolution (between 100 and 2,000 pixels), and uninformative examples; see Figure 7 in the appendix. As depicted in Figure 3c, PowerBALD performs better than BALD and essentially matches BADGE despite being much cheaper to compute. We use an acquisition size of 100 for all methods.

Computer Vision: Synbols. Synbols (Lacoste et al., 2020) is a character dataset generator which can demonstrate the behaviour of batch active learning under various edge cases (Lacoste et al., 2020; Branchaud-Charron et al., 2021). In Figure 3d, we evaluate PowerBALD on a dataset with minority character types and colours. PowerBALD outperforms BALD and matches BADGE. Further details as well as an examination of the 'spurious correlation' and 'missing synbols' edge cases (Lacoste et al., 2020; Branchaud-Charron et al., 2021) can be found in appendix C.5.

Natural Language Processing: CLINC-150. We perform intent classification on CLINC-150 (Larson et al., 2019), which contains 150 intent classes plus an out-of-scope class. This setting captures data seen in production for chatbots. We fine-tune a pretrained DistilBERT model from HuggingFace (Wolf et al., 2020) on CLINC-150 for 5 epochs with Adam as optimiser. In appendix C.6, we see that PowerEntropy shows strong performance. This demonstrates that our technique is domain independent and can be easily reused for other tasks.

Summary. We have verified that stochastic acquisition functions outperform top-K batch acquisition in several different settings and perform on par with more complex methods such as BADGE or BatchBALD. Moreover, we refer the reader to Jesson et al. (2021), Murray et al. (2021) and Tigas et al. (2022) for additional new works that use our stochastic acquisition functions.

6 Further investigations

In this section, we validate our assumptions about the underlying score dynamics by examining the score rank correlations across acquisitions. We further hypothesise about when top-K acquisition is the most detrimental to active learning.

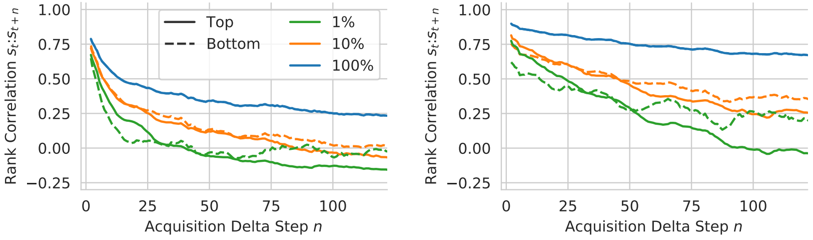
Rank Correlations Across Acquisitions. Our method is based on assuming: (1) the acquisition scores s_t at step t are a proxy for scores $s_{t'}$ at step $t' > t$; (2) the larger $t' - t$ is, the worse a proxy s_t is for s'_t; (3) this effect is the largest for the most informative points.

We demonstrate these empirically by examining the Spearman rank correlation between scores during acquisition. Specifically, we train a model for n steps using BALD as single-point acquisition function. We compare the rank order at each step to the starting rank order at step t.

Figure 1 shows that acquisition scores become less correlated as more points are acquired. Figure 4a shows this in more detail for the top and bottom 1%, 10% or 100% of scorers of the test set across acquisitions starting at step $t = 0$ for a model initialised with 20 points. The ranks of the top-10% scoring points (solid

---

[2] This result exactly reproduces BatchBALD's trajectory in Figure 7 from Kirsch et al. (2019).



(a) $t = 0$

(b) $t = 100$

Figure 4: *Rank correlations for BALD scores on MNIST between the initial scores and future scores of the top- or bottom-scoring 1%, 10% and 100% of test points (smoothed with a size-10 Parzen window).* Rank-orders decorrelate faster for the most informative samples and in the early stages of training. The top-1% scorers' ranks *anti-correlate* after roughly 40 (100) acquisitions unlike the bottom-1%. Later in training, the acquisition scores stay more strongly correlated. This suggests *the acquisition size could be increased later in training.*

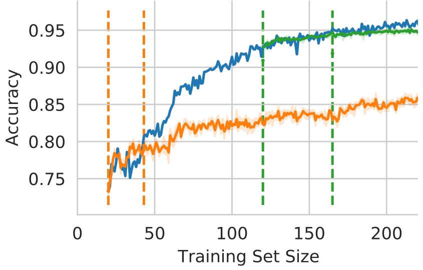


Figure 5: *Top-K acquisition hurts less later in training (BALD on MNIST).* At $t \in \{20, 100\}$ (blue), we keep acquiring samples using the BALD scores from those two steps. At $t = 20$ (orange), the model performs well for ≈ 20 acquisitions; at $t = 120$ (green), for ≈ 50; see §6.

green) become quickly uncorrelated with future scores and become *anti-correlated.* In contrast, the points overall (solid blue) correlate well over time (although they have a much weaker training signal on average). This result supports all three of our hypotheses.

At the same time, we see that as training progresses and we converge towards the best model, the order of scores becomes more stable across acquisitions. In Figure 4b the model begins with 120 points ($t = 100$), rather than 20 ($t = 0$). Here, the most informative points are less likely to change their rank—even the top-1% ranks do not become *anti-correlated*, only de-correlated. Thus, we hypothesise that further in training, we might be able to choose larger K.

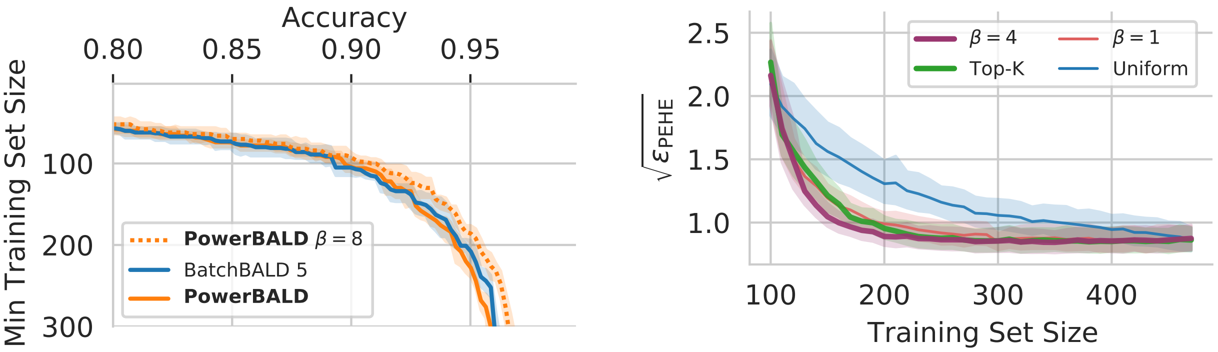
Increasing Top-K Analysis. Another way to investigate the effect of top-K selection is to freeze the acquisition scores during training and then continue single-point 'active learning' as if those were the correct scores. Comparing this to the performance of regular active learning with updated single-point scores allows us to examine how well earlier scores perform as proxies for later scores. We perform this toy experiment on MNIST, showing that freezing scores early on greatly harms performance while doing it later has only a small effect (Figure 5). For frozen scores at a training set size of 20 (73% accuracy, $t = 0$), the accuracy matches single-acquisition BALD up to a training set size of roughly 40 (dashed orange lines) before diverging to a lower level. But when freezing the scores of a more accurate model, at a training set size of 120 labels (93% accuracy, $t = 100$), selecting the next fifty points according to those frozen scores performs indistinguishably from step-by-step acquisition (dashed green lines). This result shows that top-K acquisition hurts less later in training but can negatively affect performance at the beginning of training.

These observations lead us to ask whether we could dynamically change the acquisition size: with smaller acquisition batches at the beginning and larger ones towards the end of active learning. We leave the exploration of this for future work.

6.1 Ablation: Changing β

So far, we have set $\beta = 1$ in the spirit of providing a simple baseline without additional hyperparameters. The results above show that this already works well and matches the performance of much more expensive methods, raising questions about their value. In addition, however, tuning β may be able to further improve performance. In the following, we show that other values of β can yield even higher performance on Repeated-MNIST and when estimating causal treatment effects; we provide additional results in appendix E.

Repeated-MNIST. In Figure 6a, we see that for PowerBALD the best-performing value, $\beta = 8$, outperforms BatchBALD.



(a) Repeated-MNISTx4

(b) IHDP

Figure 6: *Effect of changing β.* **(a)** *Repeated-MNISTx4 (5 trials):* PowerBALD outperforms SotA BatchBALD for $\beta = 8$. **(b)** *IHDP (400 trials):* At high temperature ($\beta = 0.1$), CausalBALD with power acquisition is like random acquisition. As the temperature decreases, the performance improves (lower $\sqrt{\epsilon_{\text{PEHE}}}$), surpassing top-$K$ acquisition.

Causal Treatment Effects: Infant Health Development Programme. Active learning for Conditional Average Treatment Effect (CATE) estimation Heckman et al. (1997; 1998); Hahn (1998); Abrevaya et al. (2015) on data from the Infant Health and Development Program (IHDP) estimates the causal effect of treatments on an infant's health from observational data. Statistical estimands of the CATE are obtainable from observational data under certain assumptions. Jesson et al. (2021) show how to use active learning to acquire data for label-efficient estimation. Among other subtleties, this prioritises the data for which matched treated/untreated pairs are available.

We follow the experiments of Jesson et al. (2021) on both synthetic data and the semi-synthetic IHDP dataset (Hill, 2011), a commonly used benchmark for causal effects estimation. In Figure 6b we show that power acquisition performs significantly better than both top-K and uniform acquisition, using an acquisition size of 10 in all cases with further. We provide additional results on semi-synthetic data in appendix E.2. Note that methods such as BADGE and BatchBALD are not well-defined for causal-effect estimation, while our approach remains applicable and is effective when fine-tuning β.

Performance on these tasks is measured using the expected *Precision in Estimation of Heterogeneous Effect (PEHE)* (Hill, 2011) such that $\sqrt{\epsilon_{\text{PEHE}}} = \sqrt{\mathbb{E}[(\widetilde{\tau}(\mathbf{X}) - \tau(\mathbf{X}))^2]}$ (Shalit et al., 2017) where $\widetilde{\tau}$ is the estimated CATE and τ is CATE (i.e. a form of RMSE).

Limitations. Although we highlight the possibility for future work to adapt β to specific datasets or score functions, our aim is not to offer a practical recipe for this to practitioners. Our focus is on showing how even the simplest form of stochastic acquisition already raises questions for existing SotA methods.

7 Discussion & Conclusion

We have demonstrated a surprisingly effective and efficient baseline for batch acquisition in active learning. Our stochastic method is orders of magnitude faster than sophisticated batch-acquisition strategies like BADGE and BatchBALD while retaining comparable performance in many settings. Compared to the flawed top-K batch acquisition heuristic, it is never worse: we see no reason to continue using top-K acquisition.

Importantly, our work raises serious questions about the current SotA methods. If they fail to outperform such a simple baseline in a wide range of settings, do they model the interaction between points sufficiently well? If so, are the scores themselves unreliable? We call on future work in batch active learning to at least demonstrate that it can outperform our simple strategy.

At the same time, our framework opens doors for improved methods. Although our stochastic model is put forward for its computational and mathematical simplicity, future work could explore more sophisticated modelling of the predicted score changes that take the current model and dataset into account. In its simplest form, this might mean adapting the temperature of the acquisition distribution to the dataset or estimating it online. Our experiments also highlight that the acquisition size could be dynamic, with larger batch sizes acceptable later in training.

References

Jason Abrevaya, Yu-Chin Hsu, and Robert P Lieli. Estimating conditional average treatment effects. *Journal of Business & Economic Statistics*, 33(4):485–505, 2015.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds, 2020.

Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. Gone fishing: Neural active learning with fisher embeddings, 2021.

Parmida Atighehchian, Frédéric Branchaud-Charron, and Alexandre Lacoste. Bayesian active learning for production, a systematic study and a reusable library. In *ICML Workshop on uncertainty and robustness in deep learning*, 2020.

Les Atlas, David Cohn, and Richard Ladner. Training Connectionist Networks with Queries and Selective Sampling. *Neural Information Processing Systems*, 1990.

Javad Azimi, Alan Fern, Xiaoli Zhang-Fern, Glencora Borradaile, and Brent Heeringa. Batch Active Learning via Coordinated Matching. *International Conference on Machine Learning*, 2012.

Cenk Baykal, Lucas Liebenwein, Dan Feldman, and Daniela Rus. Low-regret active learning, 2021.

Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in AI. Technical Report MSR-TR-2020-32, Microsoft, May 2020.

Frédéric Branchaud-Charron, Parmida Atighehchian, Pau Rodríguez, Grace Abuhamad, and Alexandre Lacoste. Can active learning preemptively mitigate fairness issues? *ICLR Workshop on Responsable AI*, 2021.

Klaus Brinker. Incorporating Diversity in Active Learning with Support Vector Machines. *International Conference on Machine Learning*, 2003.

Colin Campbell, Nello Cristianini, and Alex Smola. Query Learning with Large Margin Classifiers. *International Conference on Machine Learning*, 2000.

Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale, 2021.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017.

Sebastian Farquhar, Yarin Gal, and Tom Rainforth. On statistical bias in active learning: How and when to fix it. *International Conference on Learning Representations*, 2021.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pp. 1183–1192. PMLR, 2017.

Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954.

Yuhong Guo and Dale Schuurmans. Discriminative Batch Mode Active Learning. In *Advances in Neural Information Processing Systems*, 2008.

Jinyong Hahn. On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica*, pp. 315–331, 1998.

James J Heckman, Hidehiko Ichimura, and Petra E Todd. Matching as an econometric evaluation estimator: Evidence from evaluating a job training programme. *The review of economic studies*, 64(4):605–654, 1997.

James J Heckman, Hidehiko Ichimura, and Petra Todd. Matching as an econometric evaluation estimator. *The review of economic studies*, 65(2):261–294, 1998.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011.

Andrew Jesson, Panagiotis Tigas, Joost van Amersfoort, Andreas Kirsch, Uri Shalit, and Yarin Gal. Causal-bald: Deep bayesian active learning of outcomes to infer treatment-effects from observational data. *Advances in Neural Information Processing Systems*, 34, 2021.

Cem Kalkanli and Ayfer Ozgur. Batched thompson sampling, 2021.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pp. 7024–7035, 2019.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, pp. 3499–3508. PMLR, 2019.

Alexandre Lacoste, Pau Rodríguez, Frédéric Branchaud-Charron, Parmida Atighehchian, Massimo Caccia, Issam Laradji, Alexandre Drouin, Matt Craddock, Laurent Charlin, and David Vázquez. Synbols: Probing learning algorithms with synthetic datasets. *NeurIPS*, 2020.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. URL `https://www.aclweb.org/anthology/D19-1131`.

Zhiming Luo, Frederic Branchaud-Charron, Carl Lemaire, Janusz Konrad, Shaozi Li, Akshaya Mishra, Andrew Achkar, Justin Eichel, and Pierre-Marc Jodoin. Mio-tcd: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing*, 27(10):5129–5141, 2018.

Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *NIPS*, 2014.

Chelsea Murray, James U. Allingham, Javier Antorán, and José Miguel Hernández-Lobato. Depth uncertainty networks for active learning, 2021.

Jerzy Neyman. edited and translated by dorota m. dabrowska and terrence p. speed (1990). on the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, 5(4): 465–472, 1923.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, 75(371):591–593, 1980.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2016.

Greg Schohn and David Cohn. Less is more: Active learning with support vector machines. pp. 839–846. Morgan Kaufmann, 2000.

Jasjeet S Sekhon. The neyman-rubin model of causal inference and estimation via matching methods. *The Oxford handbook of political methodology*, 2:1–32, 2008.

Ozan Sener and Silvio Savarese. Active Learning for Convolutional Neural Networks: A Core-Set Approach. 2018.

Burr Settles. Active Learning Literature Survey. *Machine Learning*, 2010.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pp. 3076–3085. PMLR, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

Panagiotis Tigas, Yashas Annadani, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. Interventions, where and how? experimental design for causal models at scale, 2022.

Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. Improving deterministic uncertainty estimation in deep learning for classification and regression. *arXiv*, 2021.

Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*, pp. 1–7. IEEE, 2018.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

A Ethical impact

We do not foresee any ethical risks related to this work. Insofar as our sampling method reduces computational costs, applications might benefit from reduced resource consumption. Our method appears to be better than or as good as alternatives on evaluations examining the ability to learn from data with under-represented groups and on evaluations that measure the difference between performance for the most- and least-represented groups, which may aid algorithmic fairness (see C.5).

B Method

B.1 Other scoring functions

Following Gal et al. (2017), we also examine using variation ratios (least confidence) and standard deviation as scoring functions.

Variation Ratio. Also known as *least confidence*, the variation-ratios is the complement of the least-confindent class prediction:

$$s_{\text{variation-ratios}}(i; \mathcal{I}^{\text{train}}) := 1 - \max_{y} \mathrm{p}(y \mid X = x_i). \tag{13}$$

This scoring function is non-negative and a score of 0 means that the sample is uninformative: a score of 0 means that the respective prediction is one-hot, which means that the expected information gain is also 0 as can be easily verified. Thus, variation ratios matches the intuitions behind power acquisition.

Standard Deviation. The standard deviation score function measures the sum of the class probability deviations and is closely related to the BALD scores:

$$s_{\text{std-dev}}(i; \mathcal{I}^{\text{train}}) := \sum_{y} \sqrt{\mathrm{Var}_{\mathrm{p}(\omega)}[\mathrm{p}(y \mid X = x_i, \omega)]}. \tag{14}$$

This scoring function is also non-negative, and no variance for the predictions implies a zero expected information gain and thus an uninformative sample. Thus, the standard deviation should also perform well with power acquisition.

B.2 Proof of Proposition 3.1

First, we remind the reader that a random variable G is Gumble distributed $G \sim \mathrm{Gumbel}(\mu; \beta)$ when its cumulative distribution function follows $\mathrm{p}(G \le g) = \exp(-\exp(-\frac{g-\mu}{\beta}))$.

Furthermore, the Gumbel distribution is closed under translation and positive scaling:

Lemma B.1. *Let $G \sim Gumbel(\mu; \beta)$ be a Gumbel distributed random variable, then:*

$$\alpha G + d \sim \mathit{Gumbel}(d + \alpha\mu; \alpha\beta). \tag{15}$$

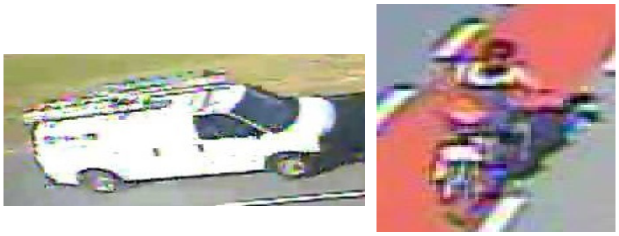
Proof. We have $\mathrm{p}(\alpha G + d \le x) = \mathrm{p}(G \le \frac{x-d}{\alpha})$. Thus, we have:

$$\begin{aligned} \mathrm{p}(\alpha G + d \le x) &= \exp(-\exp(-\frac{\frac{x-d}{\alpha} - \mu}{\beta})) && (16)\\ &= \exp(-\exp(-\frac{x - (d + \alpha\mu)}{\alpha\beta})) && (17)\\ \Leftrightarrow & \alpha G + d \sim \mathrm{Gumbel}(d + \alpha\mu; \alpha\beta). && (18) \end{aligned}$$

□

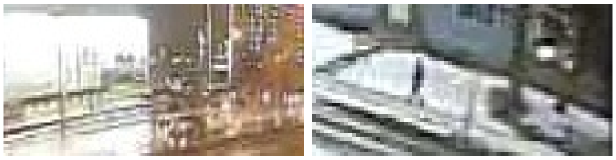
We can then easily prove Proposition 3.1 using Theorem 1 from Kool et al. (2019), which we present it here slightly reformulated to fit our notation:

(a) A good example in MIOTCD dataset.



(b) An example of duplicated samples in the dataset.

(c) An example of class confusion between motorcycle and bicycle.



(d) An example of heavy compression artefact.

(e) An example of low resolution samples.

Figure 7: *MIO-TCD Dataset* is designed to include common artifacts from production data. The size and quality of the images vary greatly between crops; from high-quality cameras on sunny days to low-quality cameras at night. (a) shows an example of clean samples that can be clearly assigned to a class. (b)(c)(d) and (e) show the different categories of noise. (b) shows an example of many near-duplicates that exist in the dataset. (c) is a good example where the assigned class is subject to interpretation (d) is a sample with heavy compression artefacts and (e) is an example of samples with low resolution which again is considered a hard example to learn for the model.

Lemma B.2. *For $k \leq n$, let $I_1^*, \ldots, I_k^* = \arg\operatorname{top}_k\{s_i + \epsilon_i\}_i$ with $\epsilon_i \sim$ Gumbel$(0;1)$, i.i.d.. Then $I_1^*, \ldots, I_k^*$ is an (ordered) sample without replacement from the Categorical$\left(\frac{\exp s_i}{\sum_{j \in n} \exp s_j}, i \in \{1, \ldots, n\}\right)$ distribution, e.g. for a realization $i_1^*, \ldots, i_k^*$ it holds that*

$$P\left(I_1^* = i_1^*, \ldots, I_k^* = i_k^*\right) = \prod_{j=1}^{k} \frac{\exp s_{i_j^*}}{\sum_{\ell \in N_j^*} \exp s_\ell}$$

where $N_j^* = N \backslash \left\{i_1^*, \ldots, i_{j-1}^*\right\}$ is the domain (without replacement) for the j-th sampled element.

Now, it is easy to prove the proposition:

Proposition 3.1. *For scores s_i, $i \in \{1, \ldots, n\}$, and $k \leq n$ and $\beta > 0$, if we draw $\epsilon_i \sim$ Gumbel$(0; \beta^{-1})$ independently, then $\arg\operatorname{top}_k\{s_i + \epsilon_i\}_i$ is an (ordered) sample without replacement from the categorical distribution* $\text{Categorical}({\exp(\beta\, s_i)}/{\sum_j \exp(\beta\, s_j)}, i \in \{1, \ldots, n\})$.

Proof. As $\epsilon_i \sim \text{Gumbel}(0; \beta^{-1})$, define $\epsilon_i' := \beta \epsilon_i \sim \text{Gumbel}(0;1)$. Further, let $s_i' := \beta s_i$. Applying Lemma B.2 on s_i' and ϵ_i', $\arg\operatorname{top}_k\{s_i' + \epsilon_i'\}_i$ yields (ordered) samples without replacement from the categorical distribution $\text{Categorical}(\frac{\exp(\beta\, s_i)}{\sum_j \exp(\beta\, s_j)}, i \in \{1, \ldots, n\})$. However, multiplication by β does not change the resulting indices of $\arg\operatorname{top}_k$:

$$\arg\operatorname{top}_k\{s_i' + \epsilon_i'\}_i = \arg\operatorname{top}_k\{s_i + \epsilon_i\}_i, \tag{19}$$

concluding the proof. □

C Experiments

C.1 Experimental setup & compute

Full code for all experiments will be available at `anonymized_github_repo`.

Frameworks. We use PyTorch. Repeated-MNIST and EMNIST experiments use PyTorch Ignite. Synbols and MIO-TCD experiments use the BaaL library `https://github.com/baal-org/baal` (Atighehchian et al., 2020). Predictive parity is calculated using FairLearn (Bird et al., 2020). The CausalBALD experiments use `https://github.com/anndvision/causal-bald` (Jesson et al., 2021).

Compute. Results shown in Table 1 were run inside Docker containers with 8 CPUs (2.2Ghz) and 32 Gb of RAM. Other experiments were run on similar machines with Titan RTX GPUs. The Repeated-MNIST and EMNIST experiments take about 5000 GPU hours. The MIO, Synbols and CLINC-150 experiments take about 19000 GPU hours. The CausalBALD experiments take about 1000 GPU hours.

Dataset Licenses. Repeated-MNIST is based on MNIST which is made available under the terms of the Creative Commons Attribution-Share Alike 3.0 license. The EMNIST dataset is made available as CC0 1.0 Universal Public Domain Dedication. Synbols is a dataset generator. MIO-TCD is made available under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. CLINC-150 is made available under the terms of Creative Commons Attribution 3.0 Unported License.

C.1.1 Runtime measurements

The synthetic dataset used for benchmarking has 4,096 features, 10 classes, and 10,000 pool points. VGG-16 models (Simonyan & Zisserman, 2014) were used to sample predictions and latent embeddings.

C.1.2 Repeated-MNIST

The Repeated-MNIST dataset is also constructed following Kirsch et al. (2019) with duplicated examples from MNIST with isotropic Gaussian noise added to the input images (standard deviation 0.1).

We use the same setup as Kirsch et al. (2019): a LeNet-5-like architecture with ReLU activations instead of tanh and added dropout. The model obtains 99% test accuracy when trained on the full MNIST dataset. Specifically, the model is made up of two blocks of a convolution, dropout, max-pooling, ReLU with 32 and 64 channels and 5x5 kernel size, respectively. As classifier head, a two-layer MLP with 128 hidden units (and 10 output units) is used that includes dropout between the layers. We use a dropout probability of 0.5 everywhere. The model is trained with early stopping using the Adam optimiser and a learning rate of 0.001. We sample predictions using 100 MC-Dropout samples for BALD. Weights are reinitialized after each acquisition step.

C.1.3 EMNIST

We follow the setup from (Kirsch et al., 2019) with 20 MC dropout samples. We use a similar model as for Repeated-MNIST but with three blocks instead of two. Specifically, we use 32, 64, and 128 channels and 3x3 kernel size. This is followed by a 2x2 max pooling layer before the classifier head. The classifier head is a two-layer MLP but with 512 hidden units instead of 128. Again, we use dropout probability 0.5 everywhere.

C.1.4 Synbols & MIO-TCD

The full list of hyperparameters for the Synbols and MIO-TCD experiments is presented in Table 3. Our experiments are built using the BaaL library (Atighehchian et al., 2020). We compute the predictive parity using FairLearn (Bird et al., 2020). We use VGG-16 model (Simonyan & Zisserman, 2014) trained for 10 epochs using Monte Carlo dropout for acquisition (Gal et al., 2017) with 20 dropout samples.

In Figure 7, we show a set of images with common problems that can be find in MIO-TCD.

Table 3: Hyper-parameters used in Section 5 and C.5

Hyperparameter	Value
Learning rate	0.001
Optimizer	SGD
Weight decay	0
Momentum	0.9
Loss function	Crossentropy
Training duration	10
Batch size	32
Dropout p	0.5
MC iterations	20
Query size	100
Initial set	500

C.1.5 CLINC-150

We fine-tune a pretrained DistilBERT model from HuggingFace (Wolf et al., 2020) on CLINC-150 for 5 epochs with Adam as optimiser. Estimating epistemic uncertainty in transformer models is an open research question, and hence, we do not report results using BALD and focus on entropy instead.

C.1.6 CausalBALD

Using the Neyman-Rubin framework (Neyman, 1923; Rubin, 1974; Sekhon, 2008), the CATE is formulated in terms of the potential outcomes, Y_{t}, of treatment levels $\mathrm{t} \in \{0, 1\}$. Given observable covariates, $\mathbf{X}$, the CATE is defined as the expected difference between the potential outcomes at the measured value $\mathbf{X} = \mathbf{x}$: $\tau(\mathbf{x}) = \mathbb{E}[\mathrm{Y}_1 - \mathrm{Y}_0 \mid \mathbf{X} = \mathbf{x}]$. This causal quantity is fundamentally unidentifiable from observational data without further assumptions because it is not possible to observe both Y_1 and Y_0 for a given unit. However, under the assumptions of consistency, non-interference, ignoreability, and positivity, the CATE is identifiable as the statistical quantity $\widetilde{\tau}(\mathbf{x}) = \mathbb{E}[\mathrm{Y} \mid \mathrm{T} = 1, \mathbf{X} = \mathbf{x}] - \mathbb{E}[\mathrm{Y} \mid \mathrm{T} = 0, \mathbf{X} = \mathbf{x}]$ (Rubin, 1980).

Jesson et al. (2021) define BALD acquisition functions for active learning CATE functions from observational data when the cost of acquiring an outcome, y, for a given covariate and treatment pair, $(\mathbf{x}, \mathrm{t})$, is high. Because we do not have labels for Y_1 and Y_0 for each $(\mathbf{x}, \mathrm{t})$ pair in the dataset, their acquisition function focusses on acquiring data points $(\mathbf{x}, \mathrm{t})$ for which it is likely that a matched pair $(\mathbf{x}, 1 - \mathrm{t})$ exists in the pool data or has already been acquired at a previous step. We follow their experiments on their synthetic dataset with limited positivity and the semi-synthetic IHDP dataset (Hill, 2011). Details of the experimental setup are given in (Jesson et al., 2021), we use their provided code, and implement the power acquisition function.

The settings for causal inference experiments are identical to those used in Jesson et al. (2021), using the IHDP dataset (Hill, 2011). Like them, we use a Deterministic Uncertainty Estimation Model (van Amersfoort et al., 2021), which is initialised with 100 datapoints and acquire 10 datapoints per acquisition batch for 38 steps. The dataset has 471 pool points and a 201 point validation set.

C.2 Repeated-MNIST

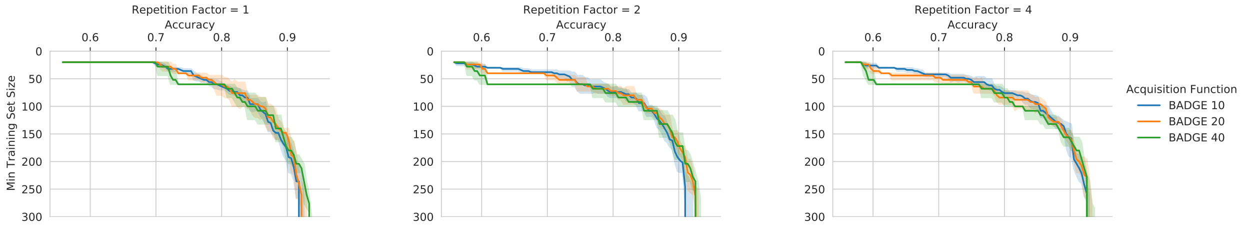


Figure 8: *Repeated-MNIST x4 (5 trials): acquisition size ablation for BADGE.* Acquisition size 20 performs best out of $\{10, 20, 40\}$. Hence, we use that for Figure 2.

BADGE Ablation. In Figure 8, we see that BADGE performs best with acquisition size 20 on Repeated-MNISTx4 overall. BADGE 40 and BADGE 20 have the highest final accuracy, cf. BADGE 10 while BADGE 20 performs better than BADGE 40 for small training set sizes.

C.2.1 Other scoring functions

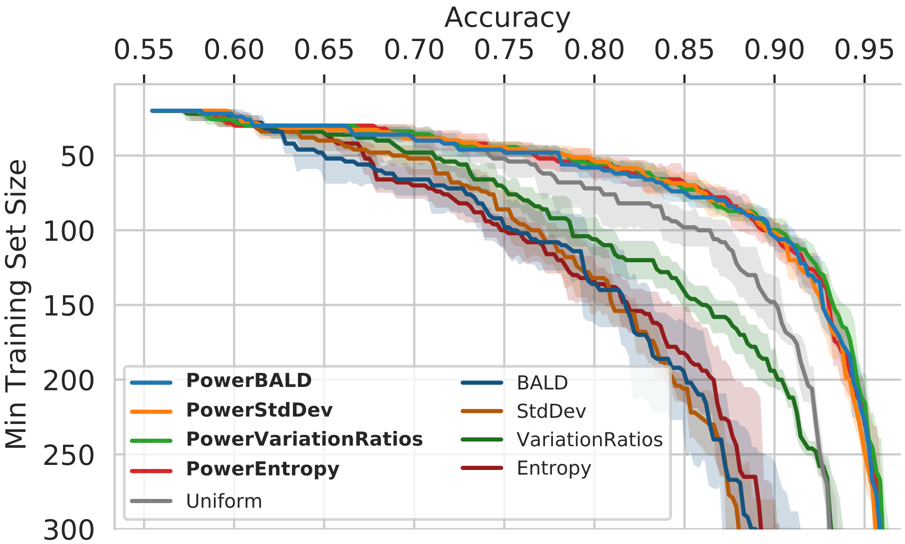


Figure 9: *Repeated-MNIST x4 (5 trials): Performance for other scoring functions.* Entropy, std dev, variation ratios behave like BALD when applying our stochastic sampling scheme.

In Figure 9 shows the performance of other scoring functions than BALD on RepeatedMNIST x4.

C.2.2 Redundancy ablation

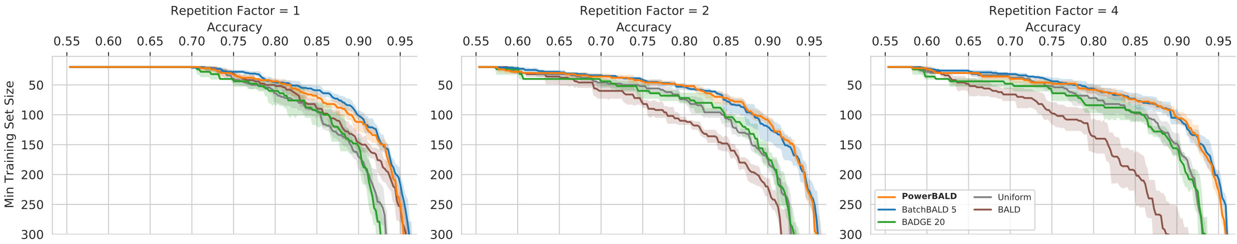
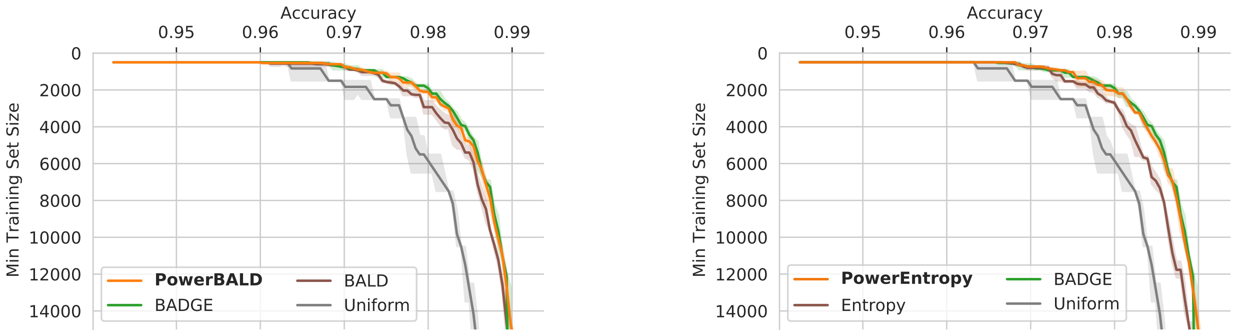


Figure 10: *Repeated-MNIST (5 trials): Performance ablation for different repetition counts.*

In Figure 10, we see the same behaviour in an ablation for different repetition sizes of Repeated-MNIST.

C.3 MIO-TCD



(a) BALD

(b) Entropy

Figure 11: *MIO-TCD (5 trials).*

In Figure 11, we see that power acquisition performs on par with BADGE with both BALD and entropy as underlying score functions.

C.4 EMNIST

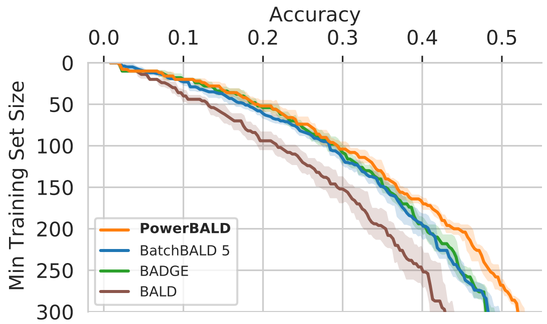


Figure 12: *EMNIST (Balanced) (5 trials): Performance with BALD.*

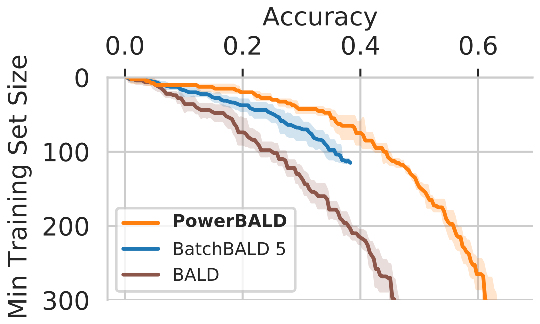


Figure 13: *EMNIST (ByMerge) (5 trials): Performance with BALD.*

In Figure 12 and 13, we see that PowerBALD outperforms BALD, BatchBALD, and BADGE.

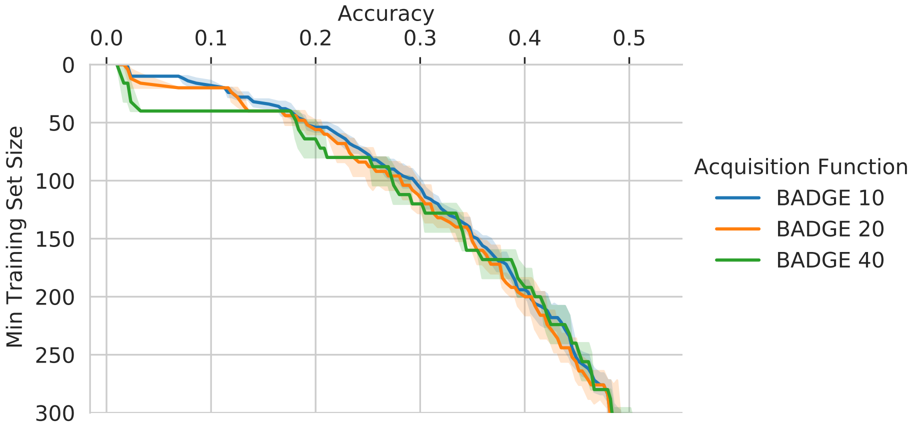


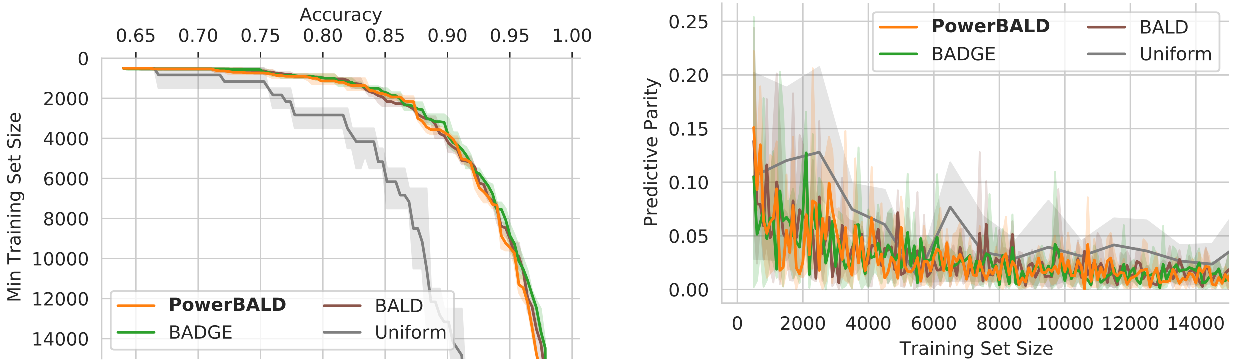
Figure 14: *EMNIST (Balanced) (5 trials): acquisition size ablation for BADGE.*

BADGE Ablation. In Figure 14, we see that BADGE performs similarly with all three acquisition sizes. Acquisition size 10 is the smoothest.

C.5 Edge cases in Synbols

We use Synbols (Lacoste et al., 2020) to demonstrate the behaviour of batch active learning in artificially constructed edge cases. Synbols is a character dataset generator for classification where a user can specify the type and proportion of bias and insert artefacts, backgrounds, masking shapes, and so on. We selected three datasets with strong biases supplied by Lacoste et al. (2020); Branchaud-Charron et al. (2021) to evaluate our method. The experimental settings are described in appendix C.1.

For these tasks, performance evaluation includes 'predictive parity', also known as 'accuracy difference', which is the maximum difference in accuracy between subgroups—which are, in this case, different coloured characters. This measure is used most widely in domain adaptation and ethics (Verma & Rubin, 2018). We want to maximise the accuracy while minimising the predictive parity.



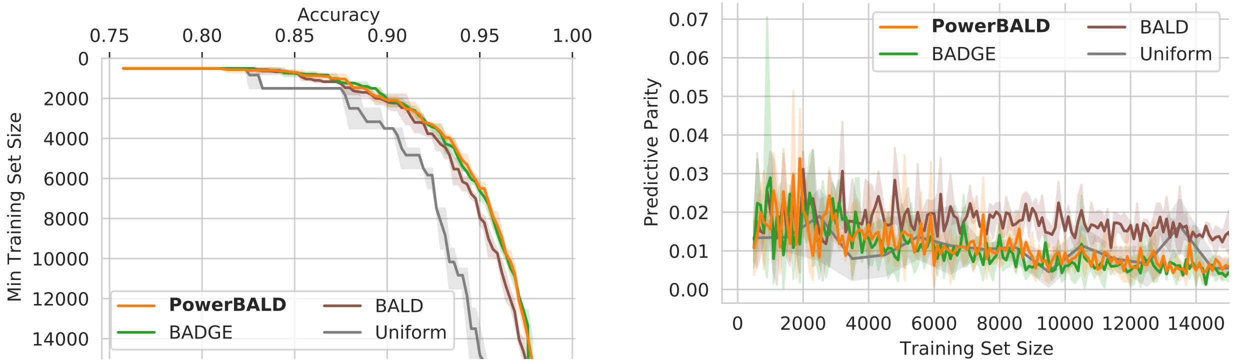
(a) Accuracy

(b) Predictive parity (**Down and left is better.**)

Figure 15: *Performance on Synbols Spurious Correlations (3 trials) with BALD.* Stochastic acquisition matches BADGE and BALD's predictive parity and performance, which is reassuring as stochastic acquisition functions might be affected by spurious correlations.

Spurious Correlations. This dataset includes spurious correlations between character colour and class. As shown in Branchaud-Charron et al. (2021), active learning is especially strong here as characters that do not follow the correlation will be informative and thus selected.

We compare the predictive parity between methods in Fig. 15b. We do not see any significant difference between our method and BADGE or BALD. This is encouraging, as stochastic approaches might select more examples following the spurious correlation and thus have higher predictive parity, but this is not the case.



(a) Accuracy

(b) Predictive parity

Figure 16: *Synbols Minority Groups (3 trials): Performance on BALD*. PowerBALD outperforms BALD and matches BADGE for both accuracy and predictive parity.

Minority Groups. This dataset includes a subgroup of the data that is under-represented; specifically, most characters are red while few are blue. As Branchaud-Charron et al. (2021) shows, active learning can improve the accuracy for these groups.

Our stochastic approach lets batch acquisition better capture under-represented subgroups. In Figure 16a, PowerBALD has an accuracy almost identical to that of BADGE, despite being much cheaper, and outperforms BALD. At the same time, we see in Figure 16b that PowerBALD has a lower predictive parity than BALD, demonstrating a fairer predictive distribution given the unbalanced dataset.

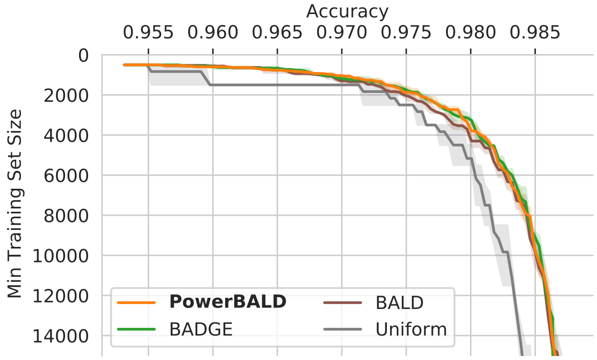


Figure 17: BALD

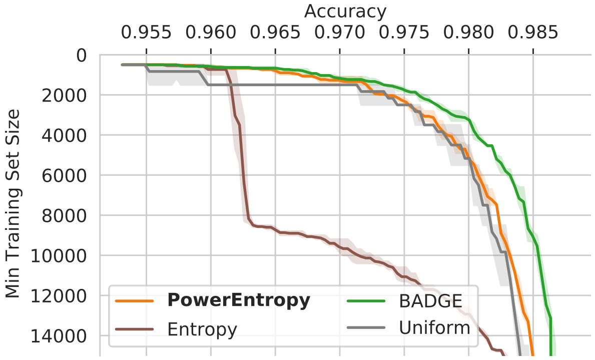


Figure 18: Entropy

Figure 19: *Performance on Synbols Missing Characters (3 trials).* In this dataset with high aleatoric uncertainty, PowerBALD matches BADGE and BALD performance. PowerEntropy significantly outperforms Entropy which confounds aleatoric and epistemic uncertainty.

Missing Synbols. This dataset has high aleatoric uncertainty. Some images are missing information required to make high-probability predictions—these images have shapes randomly occluding the character—so even a perfect model would remain uncertain. Lacoste et al. (2020) demonstrated that entropy is ineffective on this data as it cannot distinguish between aleatoric and epistemic uncertainty, while BALD can do so. As a consequence, entropy will unfortunately prefer samples with occluded characters, resulting in degraded active learning performance. For predictive entropy, stochastic acquisition largely corrects the failure of entropy acquisition to account for missing data (Figure 19) although PowerEntropy still underperforms BADGE here. For BALD, we show in Figure 17 in the appendix that, as before, our stochastic method performs on par with BADGE and marginally better than BALD.

C.6 CLINC-150

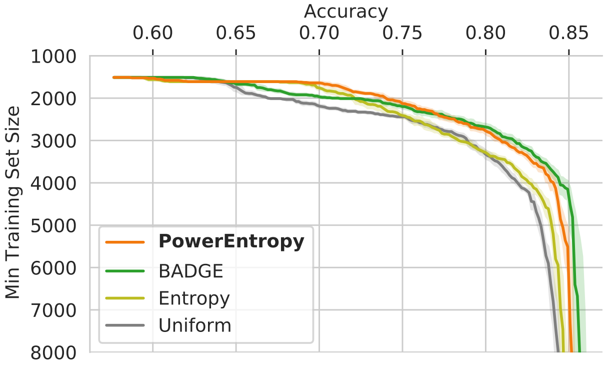
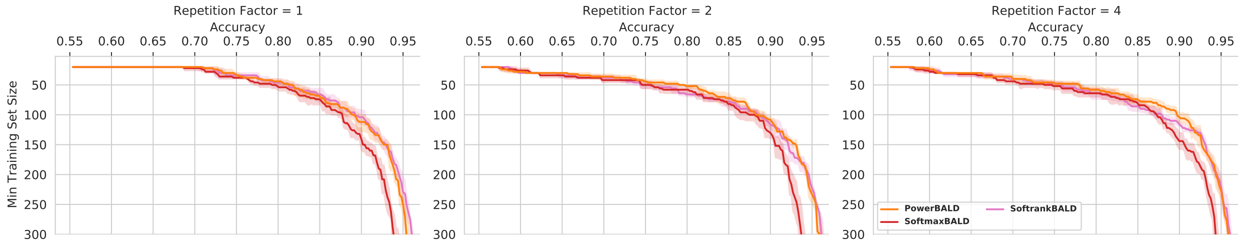


Figure 20: *Performance on CLINC-150 (10 trials).* PowerEntropy performs much better than entropy, which only performs marginally better than uniform, and almost on par with BADGE.

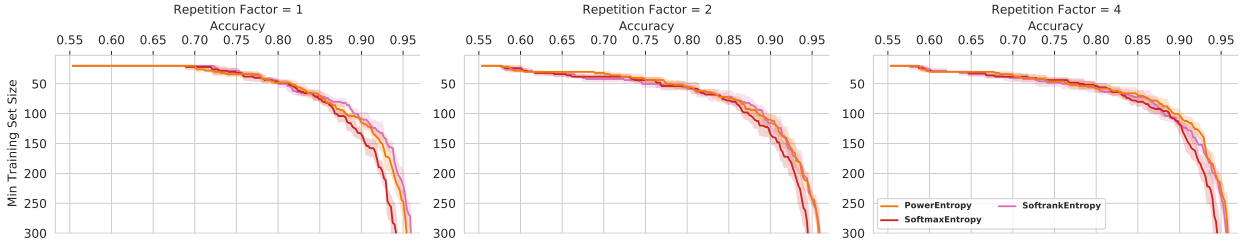
In Figure 20, we see that PowerEntropy performs much better than entropy which only performs marginally better than the uniform baseline. PowerEntropy also performs better than BADGE at low training set sizes, but BADGE performs better in the second half. Between ≈ 2300 and 4000 samples, BADGE and PowerEntropy perform the same.

D Comparing Power, Softmax and Soft-Rank

D.1 Empirical Evidence



(a) BALD



(b) Entropy

Figure 21: *Repeated-MNIST (5 trials): Performance with all three stochastic strategies.*

Repeated-MNIST. In Figure 21, power acquisition performs best overall, followed by soft-rank and then softmax.

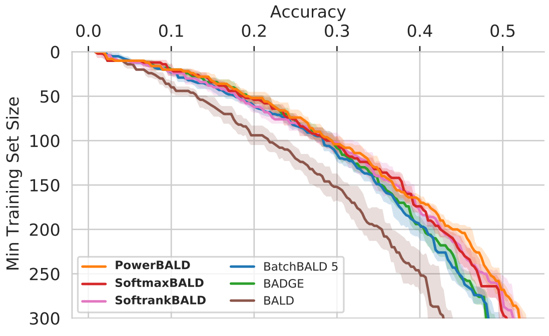


Figure 22: *EMNIST (Balanced) (5 trials): Performance with all three stochastic strategies with BALD.* PowerBALD performs best.

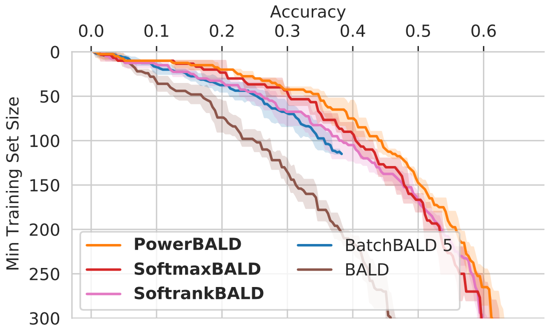
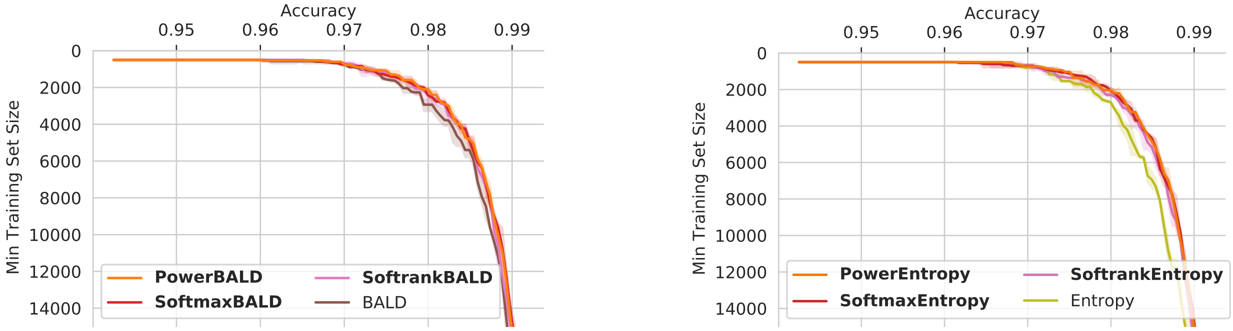


Figure 23: *EMNIST (ByMerge) (5 trials): Performance with all three stochastic strategies with BALD.* PowerBALD performs best.

EMNIST. In Figure 22 and 23, we see that PowerBALD performs best, but Softmax- and SoftrankBALD also outperform other methods. BADGE did not run on EMNIST (ByMerge) due to out-of-memory issues and BatchBALD took very long as EMNIST (ByMerge) has more than 800,000 samples.

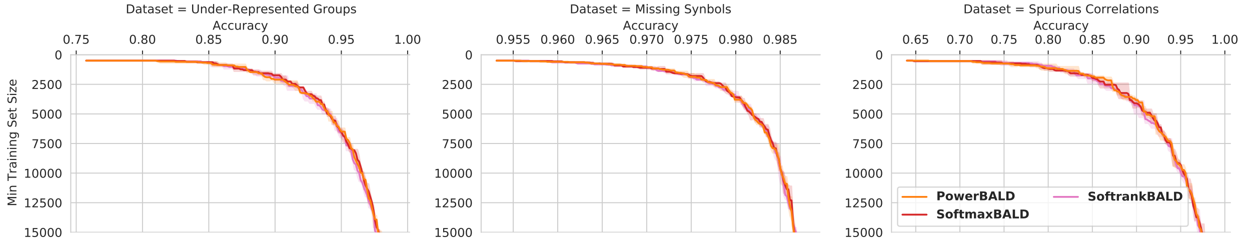


(a) BALD

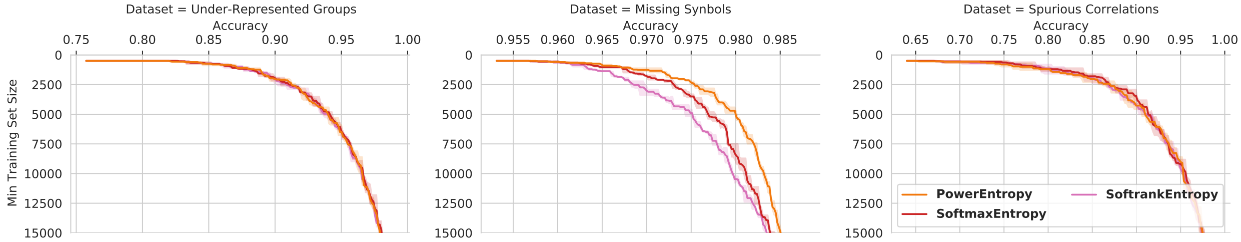
(b) Entropy

Figure 24: *MIO-TCD (3 trials): Performance with all three stochastic strategies.*

MIO-TCD. In Figure 24, we see that all three stochastic acquisition methods perform about equally well.



(a) BALD



(b) Entropy

Figure 25: *Synbols edge cases (3 trials): Performance with all three stochastic strategies.*

Synbols. In Figure 25, power acquisition seems to perform better overall—mainly due to the performance in Synbols Missing Characters.

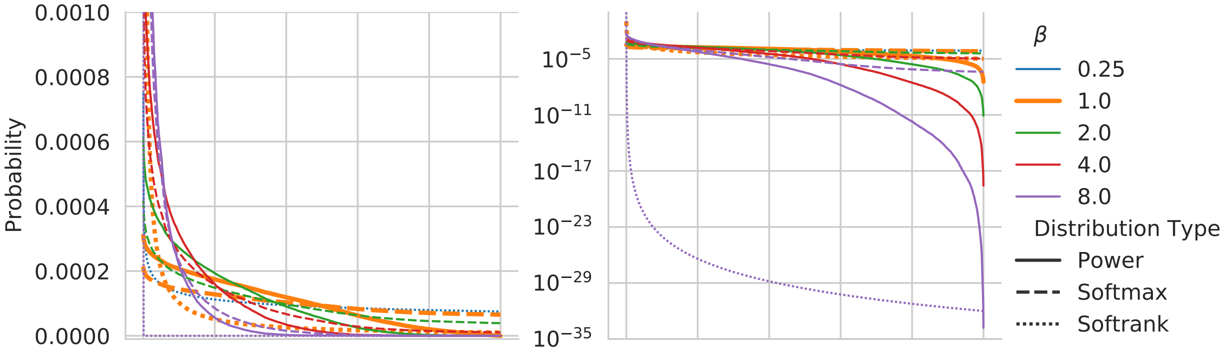


Figure 27: *Score distribution for power and softmax acquisition of BALD scores on MNIST for varying Coldness* β *at* $t = 0$. Linear and log plot over samples sorted by their BALD score. At $\beta = 8$ both softmax and power acquisition have essentially the same distribution for high scoring points (closely followed by the power distribution for $\beta = 4$). This might explain why the coldness ablation shows that these β to have very similar AL trajectories on MNIST. Yet, while softmax and power acquisition seem transfer to RMNIST, this is not the case for softrank which is much more sensitive to β. At the same time, power acquisition avoids low-scoring points more than softmax acquisition.

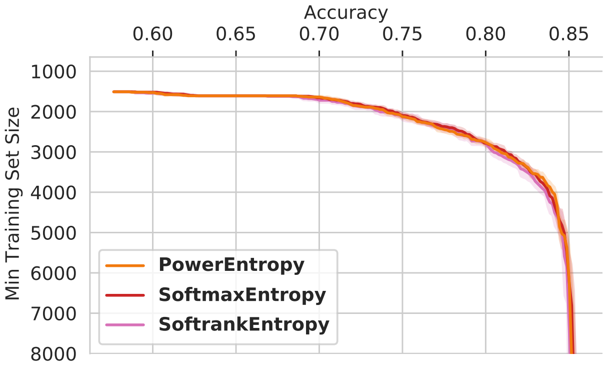


Figure 26: *CLINC-150 (10 trials): Performance with all three stochastic strategies.*

CLINC-150. In Figure 26, all three stochastic methods perform similarly.

D.2 Investigation

To further examine the three stochastic acquisition variants, we plot their score distributions, extracted from the same MNIST toy example, in Figure 27. Power and softmax acquisition distributions are similar for $\beta = 8$ (power, softmax) and $\beta = 4$ (softmax). This might explain why active learning with these β shows similar accuracy trajectories.

We find that power and softmax acquisition are quite insensitive to β and thus selecting $\beta = 1$ might generally work quite well.

E Effect of changing β

E.1 Repeated-MNIST

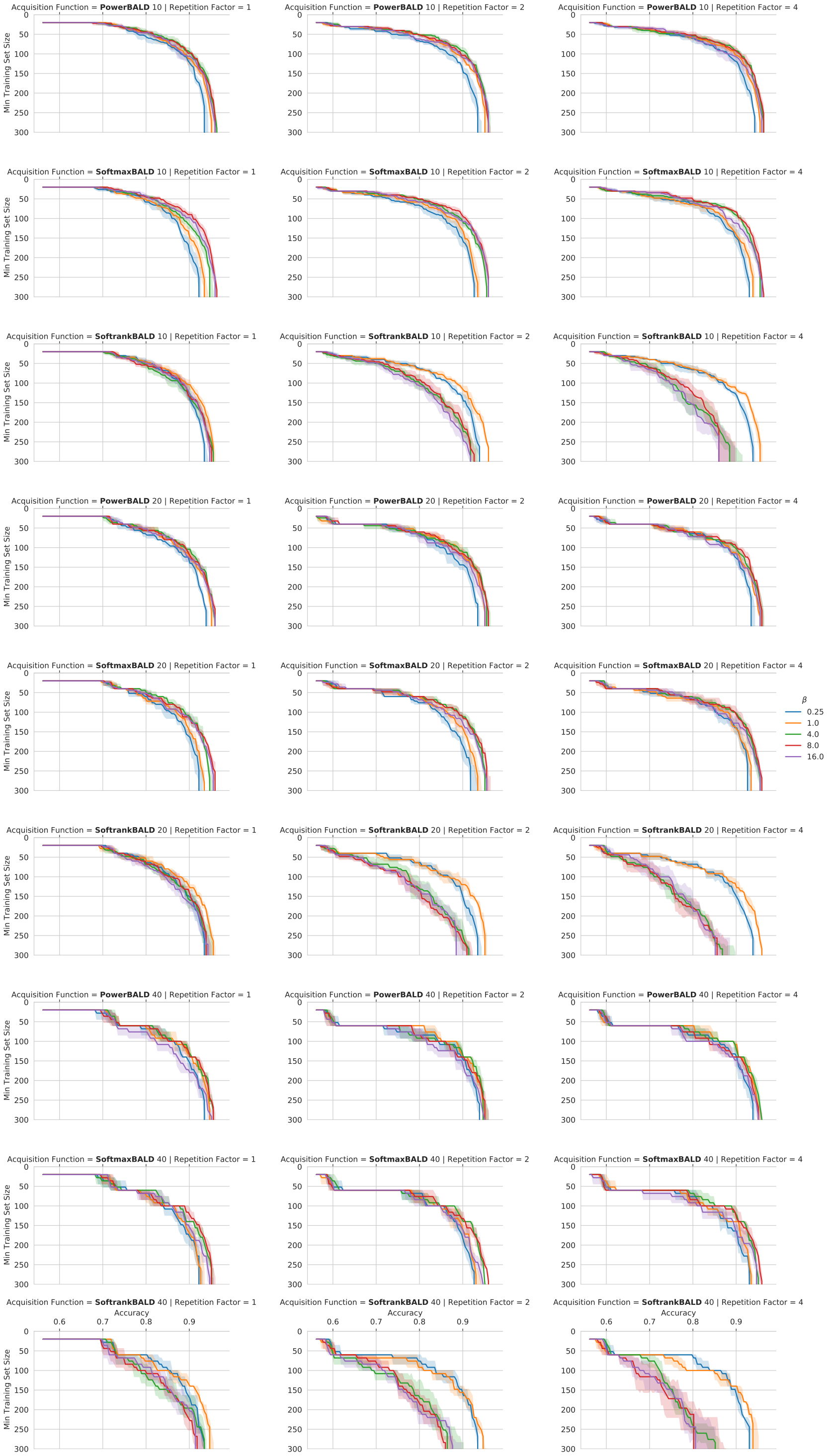


Figure 28: *Repeated-MNIST: β ablation for *BALD.*

E.1.1 MIO-TCD and Synbols

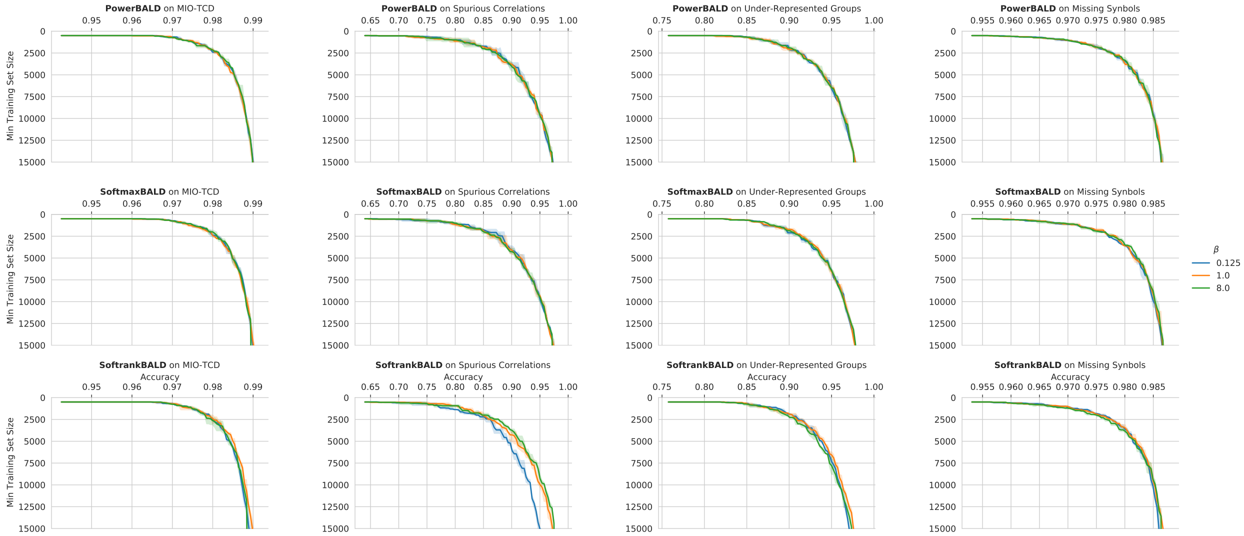


Figure 29: *MIO-TCD and Synbols: β ablation for *BALD.*

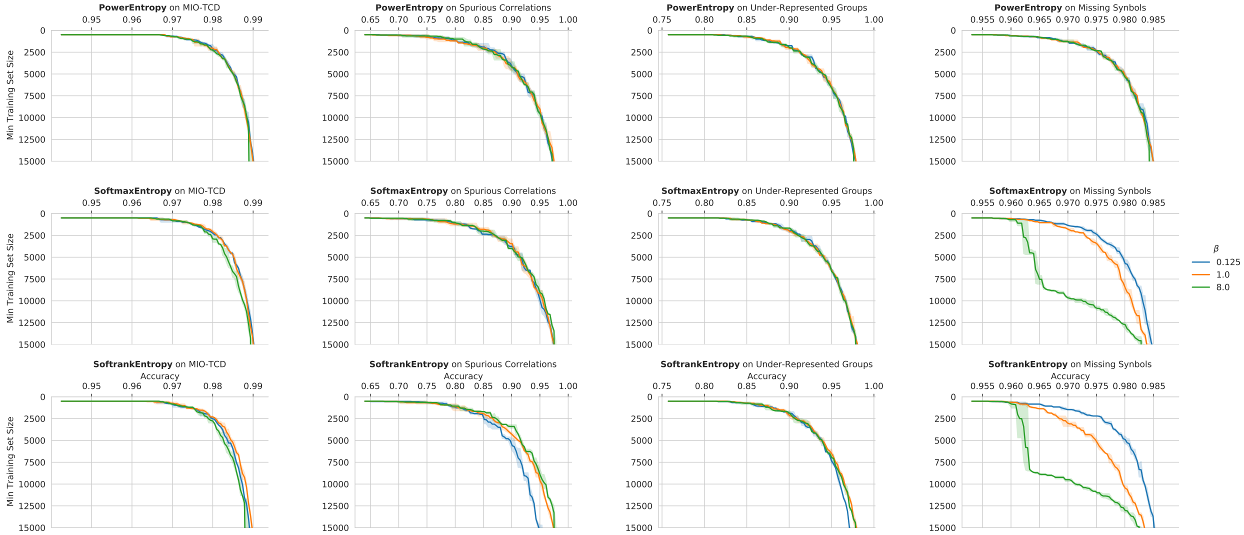
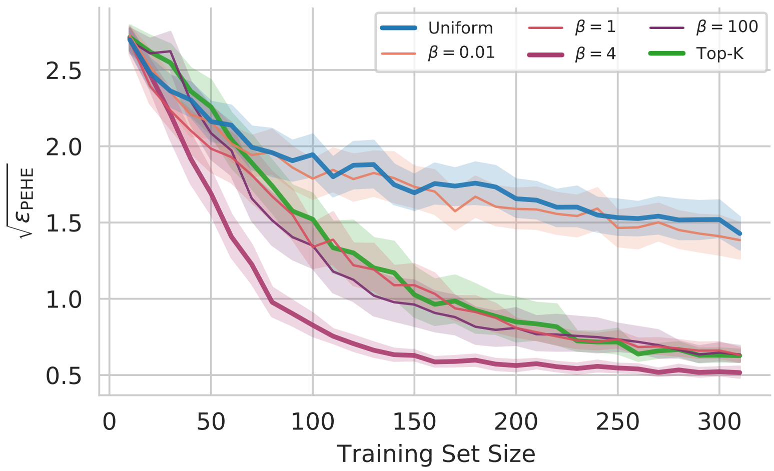
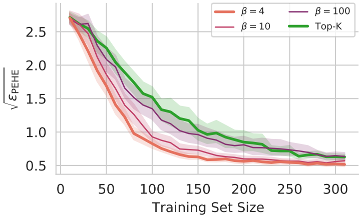


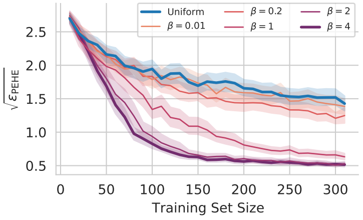
Figure 30: *MIO-TCD and Synbols: β ablation for *Entropy.*

E.2 CausalBALD: synthetic dataset





(b) Low Temperature Only



(a) Overall Ablation (Subset)

(c) High Temperature Only

Figure 31: *CausalBALD: Synthetic Dataset.* (a) At a very high temperature ($\beta = 0.1$), PowerBALD behaves very much like random acquisition, and as the temperature decreases the performance of the acquistion function improves (lower $\sqrt{\epsilon_{\text{PEHE}}}$). (b) Eventually, the performance reaches an inflection point ($\beta = 4.0$) and any further decrease in temperature results in the acquisition strategy performing more like top-K. We see that under the optimal temperature, power acquisition significantly outperforms both random acquisition and top-K over a wide range of temperature settings.

We provide further β ablations for CausalBALD on the entirely synthetic dataset which is used by Jesson et al. (2021). This demonstrates the ways in which β interpolates between uniform and top-K acquisition.

E.3 CLINC-150

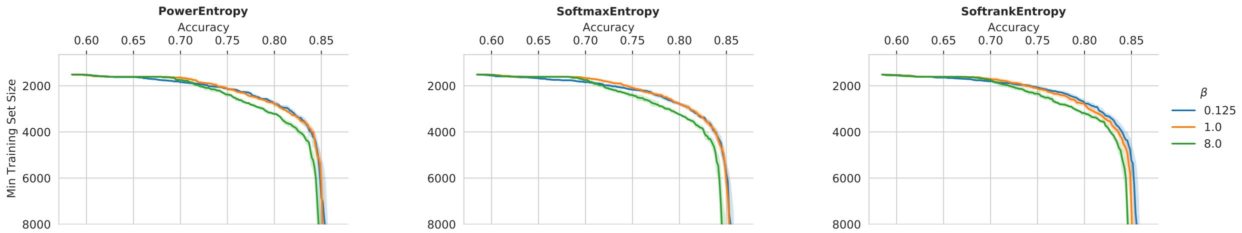


Figure 32: Performance CLINC-150: β ablation for *Entropy.