Active Learning under Pool Set Distribution Shift and Noisy Data

Andreas Kirsch [1] Tom Rainforth [2] Yarin Gal [1]

Abstract

Active Learning is essential for more label-efficient deep learning. Bayesian Active Learning has focused on BALD, which reduces model parameter uncertainty. However, we show that BALD gets stuck on out-of-distribution or junk data that is not relevant for the task. We examine a novel *Expected Predictive Information Gain (EPIG)* to deal with distribution shifts of the pool set. EPIG reduces the uncertainty of *predictions* on an unlabelled *evaluation set* sampled from the test data distribution whose distribution might be different to the pool set distribution. Based on this, our new EPIG-BALD acquisition function for Bayesian Neural Networks selects samples to improve the performance on the test data distribution instead of selecting samples that reduce model uncertainty everywhere, including for out-of-distribution regions with low density in the test data distribution. Our method outperforms state-of-the-art Bayesian active learning methods on high-dimensional datasets and avoids out-of-distribution junk data in cases where current state-of-the-art methods fail.

1. Introduction

Active learning is essential for increasing label- and thus cost-efficiency in real-world machine learning applications, especially when they use deep learning. Similarly, quantifying uncertainty using Bayesian methods is important for safety-critical systems. Combining active learning with Bayesian methods for deep neural networks has been an important research avenue for this reason.

In *active learning* (Atlas et al., 1990; Settles, 2009), we have access to a huge reservoir of unlabelled data $\{x_i^{\text{pool}}\}_{i \in \{1,\dots,|\mathcal{D}^{\text{pool}}|\}}$ in a *pool set* $\mathcal{D}^{\text{pool}}$. We iteratively use an *acquisition function* to score and select samples from this pool set to be labeled by an oracle (e.g. human experts) and added to the training set. Ideally, the selected samples are informative and increase the performance of the machine learning model faster than a *random acquisition* of samples would.

Bayesian neural networks treat the model parameters Ω as a random variable with a distribution $\mathrm{p}(\omega)$. Using training data $\mathcal{D}^{\text{train}}$, a posterior distribution $\mathrm{p}(\omega \mid \mathcal{D}^{\text{train}})$ is inferred. This contrasts with regular deep learning, which only learns a maximum-likelihood point estimate of the model parameters.

Conventionally, Bayesian Active Learning selects samples x which maximize the *expected information gain* $\mathrm{I}[\Omega; Y \mid x, \mathcal{D}^{\text{train}}]$ between the model parameters Ω and the prediction Y of the model for x, which is also referred to as *BALD (Bayesian Active Learning by Disagreement)* acquisition function (Houlsby et al., 2011). Using this acquisition function ensures that the *model posterior uncertainty* $\mathrm{H}[\Omega \mid \mathcal{D}^{\text{train}}]$ is reduced as quickly as possible with the aim of converging to the true model parameters.

This approach suffers from several issues: firstly, unlike in experiment design and statistics, where the model parameters are sought as the main goal, in active learning, the task is to minimize an empirical risk objective in a supervised setting. We are interested in making correct predictions for samples form the test distribution, not necessarily getting a good estimate of the posterior distribution. Secondly, the models are of limited capacity and do not contain the data-generating model, which means that there will be trade-offs in the performance of the model depending on what we focus (Cobb et al., 2018).

Moreover, the pool set might not follow the distribution of the test set the model should generalize to: it is easy to collect unlabelled data in a pool set, but it is hard to clean, filter and resample the pool set to make it follow the test set distribution one is interested in. For example, for big text datasets that are created by crawling the internet, the crawled data might not follow the distribution which we want to predict on. This is not covered by the usual active learning setting which assumes no distribution shift.

We introduce a novel acquisition function which selects samples that improve the performance on an *evaluation*

[1] OATML, Department of Computer Science, [2] Department of Statistics, Oxford. Correspondence to: Andreas Kirsch <andreas.kirsch@cs.ox.ac.uk>.

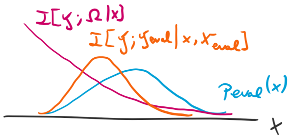


Figure 1. Expected Information Gain (BALD) vs Expected Predictive Information Gain. EPIG acquires samples near the evaluation set.

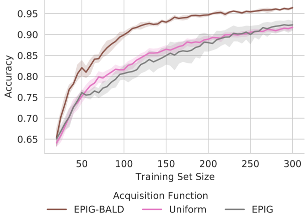


Figure 2. Ablation: EPIG vs EPIG-BALD with Bayesian Neural Networks on MNIST. EPIG does not perform better than uniform acquisition.

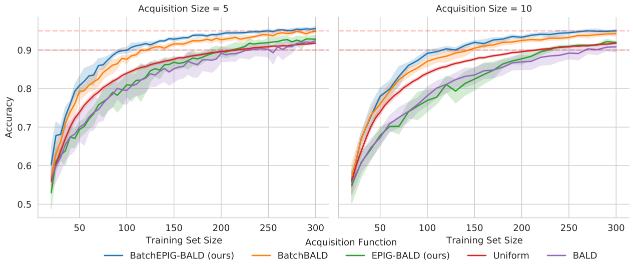


Figure 3. (Batch)BALD vs (Batch)EPIG-BALD on RepeatedMNIST (MNISTx2) with batch acquisition size 5. EPIG-BALD outperforms BALD. However, EPIG does not perform well with BNNs as explained in §4.2.

set $\boldsymbol{\mathcal{D}}^{\text{eval}}$, which is unlabeled[1] and whose samples $\{x_i^{\text{eval}}\}_{i \in \{1,\dots,|\boldsymbol{\mathcal{D}}^{\text{eval}}|\}}$ follow the test data distribution: the *Expected Predictive Information Gain* (EPIG) $\mathrm{I}[\{Y_i^{\text{eval}}\}_i; Y \mid x, \{x_i^{\text{eval}}\}_i, \boldsymbol{\mathcal{D}}^{\text{train}}]$ between the evaluation set $\boldsymbol{\mathcal{D}}^{\text{eval}}$ and a candidate sample x. We show that EPIG avoids acquiring samples that only reduce the model uncertainty in out-of-distribution regions with low density in the test data distribution and that EPIG is directly connected to selecting samples that help minimze the generalization loss.

In fig. 1, we show a toy example which visualizes the difference between the Expected Information Gain and our new Predictive Information Gain. Whereas BALD acquires samples outside the evaluation set, EPIG correctly focuses on acquiring samples that meaningfully reduce the uncertainty for the evaluation set.

The expected predictive information gain can be viewed as a version of the expected information gain $\mathrm{I}[\Omega; Y \mid x, \boldsymbol{\mathcal{D}}^{\text{train}}]$ (BALD) where the predictions on the evaluation set take the place of the model parameters. The model parameters become nuisance variables, as we do not care about learning the true model parameters in this setting: we only care about learning the model parameters insofar as they allow us to make correct future predictions on the data distribution of the unlabelled evaluation set $\boldsymbol{\mathcal{D}}^{\text{eval}}$.

To expand EPIG to high-dimensional datasets and deep learning, we show that

$$
\begin{aligned}
&\mathrm{I}[\{Y_i^{\text{eval}}\}_i; Y \mid x, \{x_i^{\text{eval}}\}_i, \boldsymbol{\mathcal{D}}^{\text{train}}] = \\
&\quad = \mathrm{I}[\{Y_i^{\text{eval}}\}_i; Y; \Omega \mid x, \{x_i^{\text{eval}}\}_i, \boldsymbol{\mathcal{D}}^{\text{train}}],
\end{aligned}
$$

and use the triple mutual information $\mathrm{I}[\{Y_i^{\text{eval}}\}_i; Y; \Omega \mid x, \{x_i^{\text{eval}}\}_i, \boldsymbol{\mathcal{D}}^{\text{train}}]$ (EPIG-BALD) to evaluate EPIG in a tractable way for Bayesian parametric models with approximate posteriors. We decompose $\mathrm{I}[\{Y_i^{\text{eval}}\}_i; Y; \Omega \mid x, \{x_i^{\text{eval}}\}_i, \boldsymbol{\mathcal{D}}^{\text{train}}]$ into two tractable BALD terms:

$$
\begin{aligned}
&\mathrm{I}[\{Y_i^{\text{eval}}\}_i; Y; \Omega \mid x, \{x_i^{\text{eval}}\}_i, \boldsymbol{\mathcal{D}}^{\text{train}}] \\
&\quad = \mathrm{I}[\Omega; Y \mid x, \boldsymbol{\mathcal{D}}^{\text{train}}] - \mathrm{I}[\Omega; Y \mid x, \{Y_i^{\text{eval}}\}_i, \{x_i^{\text{eval}}\}_i, \boldsymbol{\mathcal{D}}^{\text{train}}].
\end{aligned}
$$

The first BALD term measures the epistemic uncertainty of the model at x. The second BALD term is a conditional mutual information over pseudo-label variables $\{Y_i^{\text{eval}}\}_i$ for the evaluation set $\{x_i^{\text{eval}}\}_i$. We will show that we can approximate this term tractably by training a separate model using self-distillation[2]. As BALD captures epistemic uncertainty, intuitively, the first term is large when the model has high epistemic uncertainty about its prediction at x, and learning the true label would thus be informative for the model, while the second term captures the epistemic uncertainty about the model's prediction at x assuming we had obtained labels for the evaluation set. This second term is small when x is similar to the evaluation set and the model can explain it well given the pseudo-labels. Together EPIG-BALD (and thus EPIG) is large when the first term is large and the second term is small, so the sample x is both informative for the model and similar to an element in the evaluation set.

Beyond introducing a novel and intuitive acquisition function, we show that we outperform BALD and other acquisition functions both in standard experimental settings with high-dimensional data as well as new experimental settings in which there is a distribution shift between the pool set and test set distribution. When the pool set is imbalanced or contains junk or noisy data, state-of-the-art acquisition functions repeatedly select junk samples and waste acquisitions on OoD or noisy data, while our method performs well and follows the distribution of the evaluation set. This allows EPIG-BALD to work directly from unfiltered pool set data as long as an unlabelled curated evaluation set of smaller size can be provided. This is of particular interest for practical applications as data cleaning and filtering is expensive.

[1] We distinguish the evaluation set $\boldsymbol{\mathcal{D}}^{\text{eval}}$ from a *validation set* as the latter is commonly understood to be labelled.

[2] If we had labels for the evaluation set, we could train with those. However, this would reduce the label efficiency of our active learning algorithm.

2. Related Work

The Expected Information Gain was introduced in Bayesian optimal experiment design by Lindley (1956). While Expected Information Gain focuses on the formulation of the mutual information term as a reduction in model posterior uncertainty given a potential sample, in active learning, BALD was introduced as a tractable formulation of the same term, with the focus on measuring the level of model disagreement for a given sample as a proxy of epistemic uncertainty (Houlsby et al., 2011). For high-dimensional data, BALD has been extended to Bayesian deep learning models using Monte-Carlo dropout (Gal et al., 2017). BALD was further extended to BatchBALD to correctly capture redundancies in the batch acquisition setting (Kirsch et al., 2019).

The predictive information as mutual information between the past and future was introduced by Bialek and Tishby (1999) and has been used in reinforcement learning to increase sample efficienc (Lee et al., 2020). In Bayesian optimization with Gaussian Processes, an information gain is usually computed between potential query candidates and the already acquired points (Srinivas et al., 2009).

To our knowledge, within (Bayesian) Active Learning, EPIG and EPIG-BALD are novel and have not been examined, previously.

3. Background

In this section, we revisit Bayesian deep learning and active learning.

Bayesian Neural Networks. The model parameters are treated as a random variable Ω with prior distribution $\mathrm{p}(\omega)$. We denote the training set $\boldsymbol{\mathcal{D}}^{\text{train}} = \{(x_i^{\text{train}}, y_i^{\text{train}})\}_{1,\ldots,i\in|\boldsymbol{\mathcal{D}}^{\text{train}}|}$, where $\{x_i^{\text{train}}\}_{i\in\{1,\ldots,|\boldsymbol{\mathcal{D}}^{\text{train}}|\}}$ are the input samples and $\{y_i^{\text{train}}\}_{i\in\{1,\ldots,|\boldsymbol{\mathcal{D}}^{\text{train}}|\}}$ the labels or targets. The probabilistic model is $\mathrm{p}(y, x, \omega) = \mathrm{p}(y \mid x, \omega)\,\mathrm{p}(\omega)\,\mathrm{p}(x)$, where x and y are outcomes for the random variables X and Y denoting the input and label, respectively. We are only interested in discriminative models and do not explicitly model $\mathrm{p}(x)$ but use empirical sample distributions instead.

To include multiple labels and inputs, we expand the model to joints of random variables $\{x_i\}_{i\in I}$ and $\{y_i\}_{i\in I}$ obtaining

$$\mathrm{p}(\{y_i\}_i, \{x_i\}_i, \omega) = \prod_{i\in I} \mathrm{p}(y_i \mid x_i, \omega)\,\mathrm{p}(x_i)\,\mathrm{p}(\omega).$$

The posterior parameter distribution $\mathrm{p}(\omega \mid \boldsymbol{\mathcal{D}}^{\text{train}})$ is determined via Bayesian inference. We obtain $\mathrm{p}(\omega \mid \boldsymbol{\mathcal{D}}^{\text{train}})$ using Bayes' theorem:

$$\mathrm{p}(\omega \mid \boldsymbol{\mathcal{D}}^{\text{train}}) \propto \mathrm{p}(\{y_i^{\text{train}}\}_i \mid \{x_i^{\text{train}}\}_i, \omega)\,\mathrm{p}(\omega).$$

which allows for predictions by marginalizing over Ω:

$$\mathrm{p}(y \mid x, \boldsymbol{\mathcal{D}}^{\text{train}}) = \mathbb{E}_{\omega\sim\mathrm{p}(\omega\mid\boldsymbol{\mathcal{D}}^{\text{train}})}\,\mathrm{p}(y \mid x, \omega).$$

Exact Bayesian inference is intractable in deep learning, and we use variational inference for approximate inference using a variational distribution $\mathrm{q}(\omega)$. We determine $\mathrm{q}(\omega)$ by minimizing the following KL divergence:

$$\begin{aligned}
&\mathrm{D_{KL}}(\mathrm{q}(\omega) \parallel \mathrm{p}(\omega \mid \boldsymbol{\mathcal{D}}^{\text{train}})) = \\
&\quad = \mathbb{E}_{\mathrm{q}(\omega)}[\underbrace{-\log \mathrm{p}((\{y_i^{\text{train}}\}_i) \mid (\{x_i^{\text{train}}\}_i), \omega)}_{\text{likelihood}}] \\
&\qquad + \underbrace{\mathrm{D_{KL}}(\mathrm{q}(\omega) \parallel \mathrm{p}(\omega))}_{\text{prior regularization}} + \underbrace{\log \mathrm{p}(\boldsymbol{\mathcal{D}}^{\text{train}})}_{\text{model evidence}} \geq 0.
\end{aligned}$$

We can use the local reparameterization trick and Monte-Carlo dropout for $\mathrm{q}(\omega)$ in a deep learning context.

Batch Active Learning. Generally, samples are acquired in batches to avoid retraining the model all the time. We score possible candidate batches x^{batch} of *acquisition batch size* b using an acquisition function $a(\{x_i^{\text{batch}}\}_i, \mathrm{p}(\omega \mid \boldsymbol{\mathcal{D}}^{\text{train}}))$ and pick the highest scoring batch:

$$\underset{\{x_i^{\text{batch}}\}_{i\in\{1,\ldots,b\}}\subseteq\boldsymbol{\mathcal{D}}^{\text{pool}}}{\arg\max} \quad a(\{x_i^{\text{batch}}\}_i, \mathrm{p}(\omega \mid \boldsymbol{\mathcal{D}}^{\text{train}}))$$

BALD was introduced as a one-sample acquisition function of the expected information gain between the prediction Y^{batch} for an input x^{batch} and the stochastic model parameters Ω: $\mathrm{I}[Y^{\text{batch}}; \Omega \mid x^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}]$. This was trivially extended to batch acquisition by selecting the top-k highest scorers as a batch (Gal et al., 2017). In Kirsch et al. (2019), this approach was shown to lead to the selection of redundant samples, and instead the one-sample case was canonically extended to the batch acquisition case using the expected information gain between the *joint* of the predictions $\{Y_i^{\text{batch}}\}_i$ for the batch candidates $\{x_i^{\text{batch}}\}_i$ and the model parameters Ω (*BatchBALD*):

$$\begin{aligned}
&a_{\text{BatchBALD}}(\{x_i^{\text{batch}}\}_i, \mathrm{p}(\Omega \mid \boldsymbol{\mathcal{D}}^{\text{train}})) := \\
&\quad = \mathrm{I}[\{Y_i^{\text{batch}}\}_i; \Omega \mid \{x_i^{\text{batch}}\}_i, \boldsymbol{\mathcal{D}}^{\text{train}}] \\
&\quad = \mathrm{I}[Y_1^{\text{batch}}, \ldots, Y_b^{\text{batch}}; \Omega \mid x_1^{\text{batch}}, \ldots, x_b^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}].
\end{aligned}$$

In practice, samples and batches are selected greedily because the information gain is submodular, and greedy selection is $1 - \frac{1}{e}$ optimal (Krause and Golovin, 2014).

Notation. Instead of $\{Y_i^{\text{eval}}\}_i$, $\{x_i^{\text{eval}}\}_i$, we will write $\boldsymbol{Y}^{\text{eval}}$, $\boldsymbol{x}^{\text{eval}}$ and so on to to cut down on notation. Like above, all terms can be canonically extended to sets by substituting the joint. We provide the full derivations in the appendix. Also note again that lower-case variables like y^{eval} are outcomes and upper-case variables like Y^{eval} are random variables, with the exception of the datasets $\boldsymbol{\mathcal{D}}^{\text{pool}}$, $\boldsymbol{\mathcal{D}}^{\text{train}}$, etc, which are sets of outcomes.

4. Method

We introduce the novel *Expected Predictive Information Gain (EPIG)* acquisition function, provide intuitions, and show how it can be computed. However, we also argue that is not feasible to compute EPIG with approximate Bayesian posteriors in practice, and hence introduce EPIG-BALD, which works for approximate Bayesian posteriors. Finally, we present an approximation that allows us to evaluate EPIG-BALD on Bayesian neural networks efficiently.

4.1. Expected Predictive Information Gain

In supervised learning, we want to minimize the generalization loss: $\mathrm{H}(\mathrm{p}_{\text{data}}(X, Y) \parallel \mathrm{p}(Y \mid X))$, the cross-entropy between the true test data distribution and our model's predictive distribution:

$$\mathrm{H}(\mathrm{p}_{\text{data}}(Y, X) \| \mathrm{p}(Y|X)) := \mathbb{E}_{\mathrm{p}_{\text{data}}(y,x)} \left[-\log \mathrm{p}(y|x) \right].$$

The data distribution is available to us as the empirical sample distribution $\hat{\mathrm{p}}_{\text{test}}(x)$, the test set.

The main idea of EPIG is to reduce the uncertainty of our model's predictions for samples that follow the test data distribution. Hence, we introduce an unlabeled *evaluation set* $\boldsymbol{\mathcal{D}}^{\text{eval}} = \{x_i^{\text{eval}}\}_{i \in \{1,\ldots,|\boldsymbol{\mathcal{D}}^{\text{eval}}|\}}$ which follows the distribution of the test set and provides us with an empirical sample distribution $\hat{\mathrm{p}}_{\text{eval}}(x)$. We use this distribution to determine our active learning acquisitions.

EPIG is a natural objective that is connected to the generalization loss on the test distribution. Assuming we had a label y^{eval} for every x^{eval}, we could write:

$$\mathrm{H}(\mathrm{p}_{\text{test}}(Y, X) \parallel \mathrm{p}(Y \mid X)) \approx \mathrm{H}(\hat{\mathrm{p}}_{\text{eval}}(Y, X) \parallel \mathrm{p}(Y \mid X)),$$

for any model distribution $\mathrm{p}(\omega)$. In active learning, we want to acquire a set of new training samples $\boldsymbol{x}^{\text{batch}}$ from the pool set which minimize the generalization loss, i.e. the cross-entropy on the test distribution:

$$\underset{\boldsymbol{x}^{\text{batch}} \subseteq x^{\text{pool}}}{\arg\min} \underbrace{\mathrm{H}(\mathrm{p}_{\text{test}}(X, Y) \parallel \mathrm{p}(Y \mid X, \boldsymbol{y}^{\text{batch}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}))}_{\approx \mathrm{H}(\hat{\mathrm{p}}_{\text{eval}}(X, Y) \parallel \mathrm{p}(Y \mid X, \boldsymbol{y}^{\text{batch}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}))},$$

where we have assumed that we know the labels $\boldsymbol{y}^{\text{batch}}$. However, we do not know the labels $\boldsymbol{y}^{\text{batch}}$ a-priori. Instead we approximate this by taking the expectation over the predicted labels for $\boldsymbol{x}^{\text{batch}}$ using the current model $\mathrm{p}(\boldsymbol{y}^{\text{batch}} \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}})$. Similarly, we also do not know the labels for $\boldsymbol{y}^{\text{eval}}$ and can compute the expectation over their predictions instead.

$$\begin{aligned}
&\mathrm{H}(\hat{\mathrm{p}}_{\text{eval}}(X, Y) \parallel \mathrm{p}(Y \mid X, \boldsymbol{y}^{\text{batch}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}})) \\
&= \mathbb{E}_{y^{\text{eval}}, x^{\text{eval}} \sim \hat{\mathrm{p}}_{\text{eval}}(y^{\text{eval}}, x^{\text{eval}})} \mathrm{H}[y^{\text{eval}} \mid x^{\text{eval}}, \boldsymbol{y}^{\text{batch}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \\
&\approx \mathbb{E}_{x^{\text{eval}} \sim \hat{\mathrm{p}}_{\text{eval}}(x^{\text{eval}})} \mathrm{H}[Y^{\text{eval}} \mid x^{\text{eval}}, \boldsymbol{Y}^{\text{batch}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \\
&\geq \mathrm{H}[\boldsymbol{Y}^{\text{eval}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{Y}^{\text{batch}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}].
\end{aligned}$$

Minimizing this conditional entropy is equivalent to maximizing the expected predictive information gain $\mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}]$. The bound is not tight when there is redundancy between the samples according to the model: $\mathrm{I}[Y_i^{\text{eval}}; Y_j^{\text{eval}} \mid x_i^{\text{eval}}, x_j^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \neq 0$. In the infinite data limit, the bound is tight, however.

Definition 4.1. *The* Expected Predictive Information Gain (EPIG) *is the expected information gain between the predictions $\boldsymbol{Y}^{batch}$ for the batch candidate samples and the predictions $\boldsymbol{Y}^{eval}$ for the unlabeled evaluation set:*

$$\mathrm{I}[\boldsymbol{Y}^{eval}; \boldsymbol{Y}^{batch} \mid \boldsymbol{x}^{eval}, \boldsymbol{x}^{batch}, \boldsymbol{\mathcal{D}}^{train}]. \tag{1}$$

Lemma 4.2. *EPIG is submodular.*

Proof. This follows analogously to the proof for the submodularity of (Batch)BALD in Kirsch et al. (2019). □

This mutual information does not directly depend on the model parameters: the model parameters are marginalized out as a nuisance variables. Because EPIG is submodular, greedy sample acquisition is $1 - \frac{1}{e}$ optimal. Note that while the unlabeled *evaluation set* $\boldsymbol{\mathcal{D}}^{\text{eval}} = \{x_i^{\text{eval}}\}_{i \in \{1,\ldots,|\boldsymbol{\mathcal{D}}^{\text{eval}}|\}}$ follows the distribution of the test set, the pool set might follow a different distribution.

Expected Reduction of Predictive Uncertainty for the Evaluation Set. We can rewrite EPIG as

$$\begin{aligned}
&\mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}] = \\
&\quad = \mathrm{H}[\boldsymbol{Y}^{\text{eval}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \quad \textcircled{1} \\
&\quad - \mathrm{H}[\boldsymbol{Y}^{\text{eval}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{Y}^{\text{batch}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \quad \textcircled{2},
\end{aligned} \tag{2}$$

where $\textcircled{2}$ is the conditional entropy over the (joint) R.V.s Y^{eval} and Y^{batch}. The difference between the two terms then measures the reduction in prediction uncertainty for the evaluation set when taking into account the batch set: when it is high, it means that the batch set tells us a lot about the evaluation set. Intuitively, this reduction will be larger when batch candidate samples are similar to the samples in the evaluation set. Conversely, it is $= 0$ exactly when $Y^{\text{eval}} \perp\!\!\!\perp Y^{\text{batch}} \mid x^{\text{eval}}, x^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}$.

Evaluating the Predictive Information Gain. Even though $\textcircled{1}$ in eq. (2) above is constant for fixed $\boldsymbol{\mathcal{D}}^{\text{train}}$ and $\boldsymbol{x}^{\text{eval}}$, it is impractical to reevaluate $\textcircled{2}$ and perform a Bayesian inference step for each potential batch. Luckily, the mutual information is symmetric in its arguments, and we can expand it the other way, which yields:

$$\begin{aligned}
&\mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}] = \\
&\quad = \mathrm{H}[\boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \quad \textcircled{1'} \\
&\quad - \mathrm{H}[\boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \quad \textcircled{2'},
\end{aligned} \tag{3}$$

$\textcircled{1'}$ is simply the conditional entropy with the Bayesian model $\mathrm{p}(\omega \mid \boldsymbol{\mathcal{D}}^{\text{train}})$. To see how we can evaluate $\textcircled{2'}$, we expand using the definition of the conditional entropy:

$$\mathrm{H}[\boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] = \\ = \mathbb{E}_{\mathrm{p}(\boldsymbol{y}^{\text{eval}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}})} \mathrm{H}[\boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}].$$

We take an expectation over labels $\boldsymbol{y}^{\text{eval}}$ and the conditional entropy using the Bayesian model $\mathrm{p}(\omega \mid \boldsymbol{y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}})$.

In other words, we train a model on $\boldsymbol{\mathcal{D}}^{\text{train}}$, then sample possible joint predictions $\boldsymbol{y}^{\text{eval}}$ for the evaluation set $\boldsymbol{x}^{\text{eval}}$ and then for each such joint prediction, we evaluate the conditional entropy $\mathrm{H}[\boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{batch}}, \Omega]$ using $\omega \sim \mathrm{p}(\omega \mid \boldsymbol{y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}})$, which we can approximate using one variational distribution per sampled joint prediction for $\textcircled{2'}$ instead of one per potential batch and sampled joint prediction in the case of $\textcircled{2}$.

4.2. Triple Mutual Information: EPIG-BALD

When trying to evaluate EPIG with approximate Bayesian models like commonly used Bayesian neural network approximations, we empirically find that for variational approximations $\mathrm{q}_1(\omega) \approx \mathrm{p}(\omega \mid \boldsymbol{\mathcal{D}}^{\text{train}})$ and $\mathrm{q}_2(\omega) \approx \mathrm{p}(\omega \mid \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}})$:

$$\mathrm{H}(\mathrm{q}_1(Y \mid x, \Omega)) \neq \mathrm{H}(\mathrm{q}_2(Y \mid x, \Omega)),$$

even though

$$\mathrm{H}[Y|x, \Omega, \boldsymbol{\mathcal{D}}^{\text{train}}] = \mathrm{H}[Y|x, \Omega, \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}]$$

as $Y \perp\!\!\!\perp \boldsymbol{Y}^{\text{eval}} \mid x, \boldsymbol{x}^{\text{eval}}, \omega$. We can take this into account with the following triple mutual information which includes the model parameters explicitly:

Definition 4.3. *We define EPIG-BALD as the triple mutual information between predictions on the evaluation set, predictions on the batch set, and model parameters:*

$$\mathrm{I}[\boldsymbol{Y}^{eval}; \boldsymbol{Y}^{batch}; \Omega \mid \boldsymbol{x}^{batch}, \boldsymbol{x}^{eval}, \boldsymbol{\mathcal{D}}^{train}]. \tag{4}$$

Lemma 4.4. *We have:*

$$\mathrm{I}[\boldsymbol{Y}^{eval}; \boldsymbol{Y}^{batch}; \Omega \mid \boldsymbol{x}^{batch}, \boldsymbol{x}^{eval}, \boldsymbol{\mathcal{D}}^{train}] = \\ = \mathrm{I}[\boldsymbol{Y}^{eval}; \boldsymbol{Y}^{batch} \mid \boldsymbol{x}^{batch}, \boldsymbol{x}^{eval}, \boldsymbol{\mathcal{D}}^{train}],$$

so EPIG-BALD = EPIG, and EPIG-BALD is submodular.

Proof. We can expand this term using the properties of a triple mutual information to:

$$\mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \boldsymbol{Y}^{\text{batch}}; \Omega \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] = \\ = \mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \\ - \underbrace{\mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{x}^{\text{eval}}, \Omega, \boldsymbol{\mathcal{D}}^{\text{train}}]}_{=0} \\ = \mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}],$$

where we have used the independence $\boldsymbol{Y}^{\text{eval}} \perp\!\!\!\perp \boldsymbol{Y}^{\text{batch}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{x}^{\text{batch}}, \Omega$. □

Evaluating EPIG-BALD. We obtain a more tractable version by expanding while conditioning on $\boldsymbol{Y}^{\text{eval}}$:

$$\mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \boldsymbol{Y}^{\text{batch}}; \Omega \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \\ = \mathrm{I}[\boldsymbol{Y}^{\text{batch}}; \Omega \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \\ - \mathrm{I}[\boldsymbol{Y}^{\text{batch}}; \Omega \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}].$$

Both terms are BALD terms now, except that the second one is a conditional mutual information over $\boldsymbol{Y}^{\text{eval}}$. Again, using the definition of the conditional entropy, we expand this term to clarify how to evaluate it:

$$\mathrm{I}[\boldsymbol{Y}^{\text{batch}}; \Omega \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \\ = \mathbb{E}_{\mathrm{p}(\boldsymbol{y}^{\text{eval}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}})} \mathrm{I}[\boldsymbol{Y}^{\text{batch}}; \Omega \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}].$$

This an expectation of BALD scores for models $\mathrm{p}(\omega \mid \boldsymbol{y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}})$ over $\boldsymbol{y}^{\text{eval}} \sim \mathrm{p}(\boldsymbol{y}^{\text{eval}} \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}})$.

Expected Reduction of Epistemic Uncertainty of the Evaluation set. In addition to the intuition provided in §1, we see that by conditioning on $\boldsymbol{Y}^{\text{batch}}$ instead of $\boldsymbol{Y}^{\text{eval}}$ we obtain:

$$\mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \boldsymbol{Y}^{\text{batch}}; \Omega \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] = \\ = \mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \Omega \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \\ - \mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \Omega \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{Y}^{\text{batch}}, \boldsymbol{x}^{\text{batch}}, \boldsymbol{\mathcal{D}}^{\text{train}}].$$

Since BALD is known to measure epistemic uncertainty (Smith and Gal, 2018), maximizing EPIG-BALD (and thus EPIG) selects a batch which reduces the epistemic uncertainty for predictions on the evaluation set the most (as $\mathrm{I}[\boldsymbol{Y}^{\text{eval}}; \Omega \mid \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] = \text{const}$ independently of x^{batch}).

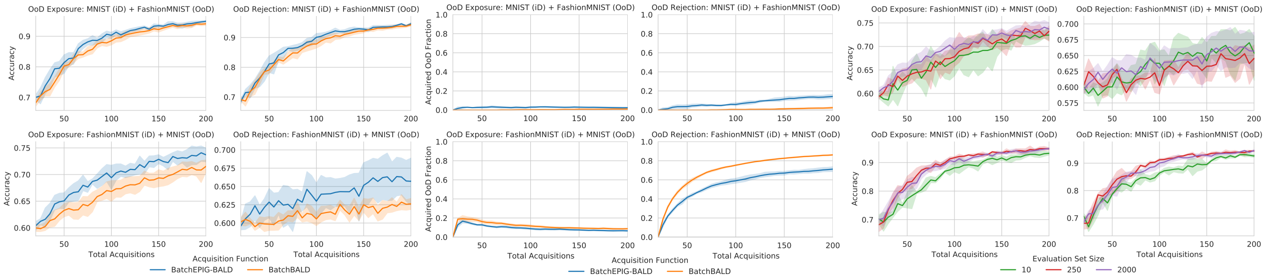
4.3. Estimating $\mathrm{I}[\boldsymbol{Y}^{\text{batch}}; \Omega \mid \boldsymbol{x}^{\text{batch}}, \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}]$

We want to find a model approximation $\hat{\Omega}$ with distribution $\mathrm{q}(\hat{\omega})$, such that for all possible batch sets, we have:

$$\mathrm{I}[\boldsymbol{Y}^{\text{batch}}; \Omega | \boldsymbol{x}^{\text{batch}}, \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \approx \mathrm{I}[\boldsymbol{Y}^{\text{batch}}; \hat{\Omega} | \boldsymbol{x}^{\text{batch}}].$$

We note two properties of this conditional mutual information and the underlying models $\mathrm{p}(\Omega \mid \boldsymbol{y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}})$ for different $\boldsymbol{y}^{\text{eval}} \sim \mathrm{p}(\boldsymbol{y}^{\text{eval}} \mid \boldsymbol{\mathcal{D}}^{\text{train}})$:

1. marginalizing $\mathrm{p}(\Omega \mid \boldsymbol{y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}})$ over all possible $\boldsymbol{y}^{\text{eval}}$ yields the predictions of the original posterior $\mathrm{p}(\Omega \mid \boldsymbol{\mathcal{D}}^{\text{train}})$, so we would like $\mathbb{E}_{\mathrm{q}(\hat{\Omega})} \mathrm{p}(y \mid x, \hat{\omega}) = \mathrm{p}(y \mid x, \boldsymbol{\mathcal{D}}^{\text{train}})$; and
2. $\mathrm{I}[Y; \Omega \mid x, \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \leq \mathrm{I}[Y; \Omega \mid x, \boldsymbol{\mathcal{D}}^{\text{train}}]$, and when $x \in \boldsymbol{x}^{\text{eval}}$, we expect $\mathrm{I}[Y; \Omega \mid x, \boldsymbol{Y}^{\text{eval}}, \boldsymbol{x}^{\text{eval}}, \boldsymbol{\mathcal{D}}^{\text{train}}] \ll \mathrm{I}[Y; \Omega \mid x, \boldsymbol{\mathcal{D}}^{\text{train}}]$. In other words, the epistemic uncertainty of evaluation



(a) Accuracy (b) Aquired OoD Ratio (c) Evaluation Set Size Ablation

Figure 4. MNIST and FashionMNIST pairings with OoD rejection or exposure. EPIG-BALD performs better than BALD. 5 trials.

samples x^{eval} ought to decrease when we also train on the evaluation set (using pseudo-labels $\boldsymbol{y}^{\text{eval}}$), and we would expect the same for $\hat{\Omega}$.

Hence, as a tractable approximation for Ω, we choose to use *self-distillation*, where we train a model with $\boldsymbol{\mathcal{D}}^{\text{train}}$ and the predictions of the original model $\mathrm{p}(\Omega \mid \boldsymbol{\mathcal{D}}^{\text{train}})$ on $\boldsymbol{x}^{\text{eval}}$ using a KL-divergence loss. The resulting model posterior $\hat{\Omega}$ fulfills both properties described above.

5. Empirical Validation

We evaluate the performance of EPIG and EPIG-BALD using self-distillation in a regular active learning setting and under distribution shift. We also provide an ablation with different evaluation set sizes.

Setup. We compare various Bayesian acquisition functions under batch acquisition with acquisition size 5, both using the top-k individual scores (Gal et al., 2017) and using the joint density (Kirsch et al., 2019). On MNIST and MNISTx2 (RepeatedMNIST), we use a LeNet-5 model (LeCun et al., 1998), which we train as described in Kirsch et al. (2019). We use MC dropout models with 100 dropout samples when computing the acquisition scores.

Performance in Regular Active Learning. We compare BALD and EPIG-BALD in fig. 3 on MNISTx2 and EPIG and EPIG-BALD in fig. 2 on MNIST. In the regular active learning setup, there is no distribution shift between the pool set and test set, so we use all of the unlabeled pool set as evaluation set.

Both in the top-k and the batch variant, EPIG-BALD outperforms BALD on MNISTx2 (and also MNIST, not shown). Yet, EPIG does not perform better than uniform acquisition even on MNIST. This is because the two approximate Bayesian models are separately trained and not compatible. EPIG-BALD which is based on a difference of epistemic uncertainties works, however. We conclude that the difference in informativeness and epistemic uncertainty (via mutual information) in EPIG-BALD is more meaningful than the difference in overall uncertainty (via conditional entropy) in EPIG.

Performance in Active Learning under Distribution Shift with MNIST and FashionMNIST. We compare BALD and EPIG-BALD under distribution shift. For this, we add junk out-of-distribution data to the pool set. In this experiment, the pool set contains MNIST and FashionMNIST (Xiao et al., 2017), while the test set contains one or the other. We deal with an acquisition function attempting to acquire OoD data in two different modes: *OoD rejection* rejects OoD data from the batch and does not acquire it; while *OoD exposure* acquires OoD data with uniform targets, similar to outlier exposure methods in OoD detection (Hendrycks et al., 2018). We use an evaluation set with 2000 unlabeled samples.

EPIG-BALD outperforms BALD on in all combinations, see fig. 4a. In all cases but one, EPIG-BALD acquires fewer junk/OoD samples, see fig. 4b. The ablation in fig. 4c shows that larger evaluation sets are beneficial. Note that the evaluation set is unlabeled and thus does not count towards sample acquisitions.

We provide results for active learning on CIFAR-10 in §A.1 and for active learning under distribution shift with CIFAR-10 and SVHN in §A.2.

6. Conclusion & Limitations

We have introduced a new Bayesian acquisition function which is grounded in information theory and shown that it outperforms BALD in both the regular active learning setup as well as active learning under distribution shift of the pool set. While EPIG-BALD selects fewer OoD junk samples than BALD, we would like it to select even fewer. Moreover, we need to examine additional dataset combinations and other sources of distribution shift, for example class imbalances and label noise.

References

Anonymous. Batch active learning with stochastic acquisition functions. *ICML Workshop Submission*, 2021.

Les E Atlas, David A Cohn, and Richard E Ladner. Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems*, pages 566–573. Citeseer, 1990.

William Bialek and Naftali Tishby. Predictive information. *arXiv preprint cond-mat/9902341*, 1999.

Adam D Cobb, Stephen J Roberts, and Yarin Gal. Loss-calibrated approximate inference in bayesian neural networks. *arXiv preprint arXiv:1805.03901*, 2018.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

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

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*, pages 7024–7035, 2019.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3:71–104, 2014.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Kuang-Huei Lee, Ian Fischer, Anthony Liu, Yijie Guo, Honglak Lee, John Canny, and Sergio Guadarrama. Predictive information accelerates learning in rl. *arXiv preprint arXiv:2007.12401*, 2020.

Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005, 1956.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Burr Settles. Active learning literature survey. 2009.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

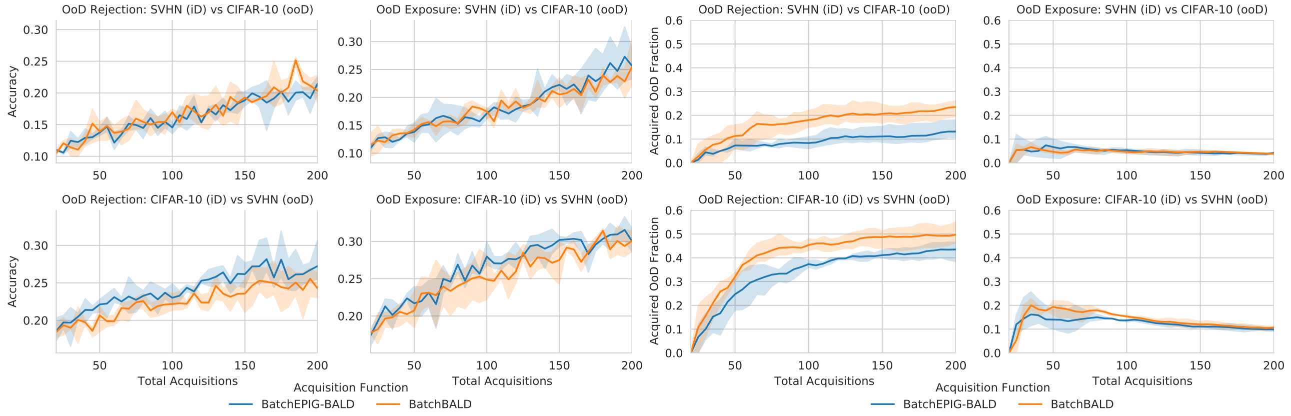
A. Additional Experiments

A.1. Performance in Regular Active Learning.

CIFAR-10. EPIG-BALD outperforms SoftmaxBALD, as depicted in fig. 6. For CIFAR-10 (Krizhevsky et al., 2009), we use a ResNet18 model (He et al., 2016) which was modified as described in Kirsch et al. (2019) to add MC dropout to the classifier head and also follows the described training regime. We train with an acquisition batch size of 250 and an intial training set size of 1000. We use stochastic acquisition functions (Softmax*) instead of BatchBALD and variants, which samples without replacement from the pool set using the Softmax of the acquisition scores with temperature 8 (Anonymous, 2021).

A.2. Performance in Active Learning under Distribution Shift.

CIFAR-10 and SVHN. EPIG-BALD outperforms BALD under distribution shift with CIFAR-10 and SVHN (Netzer et al., 2011) in all but one combination and selects fewer OoD samples, see fig. 5. We use an evaluation set with 1000 unlabeled samples.



(a) Accuracy

(b) Aquired OoD Ratio

Figure 5. CIFAR-10 and SVHN pairings with OoD rejection or exposure. EPIG-BALD performs better than BALD. 5 trials. Acquisition size 5. Initial training size 5.

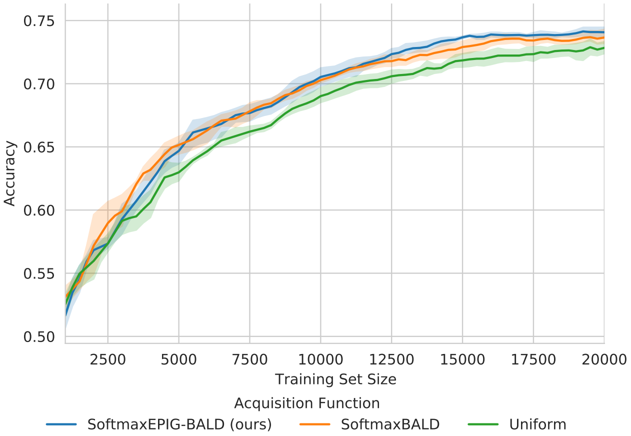


Figure 6. BALD vs EPIG-BALD on CIFAR-10. EPIG-BALD outperforms BALD. 5 trials each. With batch acquisition size 250, and initial training size 1000. Median accuracy after smoothing with a Parzen window filter over 30 acquisition steps to denoise.