# Prediction-Oriented Bayesian Active Learning

**Freddie Bickford Smith*** University of Oxford

**Andreas Kirsch*** University of Oxford

**Sebastian Farquhar** University of Oxford

**Yarin Gal** University of Oxford

**Adam Foster** Microsoft Research

**Tom Rainforth** University of Oxford

## Abstract

Information-theoretic approaches to active learning, such as BALD, typically focus on maximising the information gathered about the model parameters. We highlight that this can be suboptimal from the perspective of predictive performance. In particular, BALD fails to account for the input distribution and thus is prone to prioritise data that is of low relevance to prediction. Addressing this shortfall, we propose the expected predictive information gain (EPIG), an acquisition function that measures information gain in the space of predictions rather than parameters. We find that using EPIG leads to stronger predictive performance compared with BALD across a range of datasets and models, and thus provides an appealing drop-in replacement.

## 1 Introduction

Active learning (Atlas et al, 1989; Settles, 2012) allows for data-efficient learning by carefully selecting which inputs to acquire labels for when training a model. A principled basis for acquisition is to formalise a label's utility through the information it provides. Doing this requires a probabilistic generative model for possible future labels, leading to an approach known as Bayesian active learning (Gal et al, 2017; Houlsby et al, 2011; MacKay, 1992a,b).

Historically the literature has focused on trying to maximise the expected information gain (EIG) in the model parameters. This yields an acquisition function typically known as BALD, having been popularised by a method called Bayesian active learning by disagreement (Houlsby et al, 2011). It has been successfully applied in settings including computer vision (Gal et al, 2017) and natural-language processing (Shen et al, 2018).

In this work we highlight that BALD can be misaligned with our typical overarching goal of making accurate predictions on unseen inputs. In particular, it neglects a crucial fact: not all information about the model parameters is equally useful when it comes to making predictions. With a nonparametric model, for instance, we can gain an infinite amount of information about the model parameters without any of it being relevant to prediction on inputs of interest. In short, BALD lacks a notion of how the model will be used and so fails to ensure that the data acquired is relevant to our particular predictive task.

This has considerable practical implications. Real-world datasets are often messy, with inputs that vary widely in their relevance to a given task. Large pools of audio, images and text commonly fit this description (Ardila et al, 2020; Gemmeke et al, 2017; Mahajan et al, 2018; Radford et al, 2021; Raffel et al, 2020; Sun et al, 2017). We show that BALD can be actively counterproductive in cases like these, picking out the most obscure, least relevant inputs.

To address BALD's shortcomings we propose the expected predictive information gain (EPIG), an alternative acquisition function. We derive EPIG by returning to the foundational framework of Bayesian experimental design (Lindley, 1956), from which BALD itself is derived. Whereas BALD is the EIG in the model parameters, EPIG is the EIG in the model's predictions: it measures how much information the label of a candidate input is expected to provide about the label of a random target input. While BALD favours global reductions in parameter uncertainty, EPIG favours only information that reduces downstream predictive uncertainty (Figure 1). Thus EPIG allows us to directly seek improvements in predictive performance.

The randomness of the target input in EPIG is critical. We do not aim for predictive information gain on a particular input or set of inputs. Instead the gain is in expectation with respect to a target input distribution. This can be chosen to

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. 

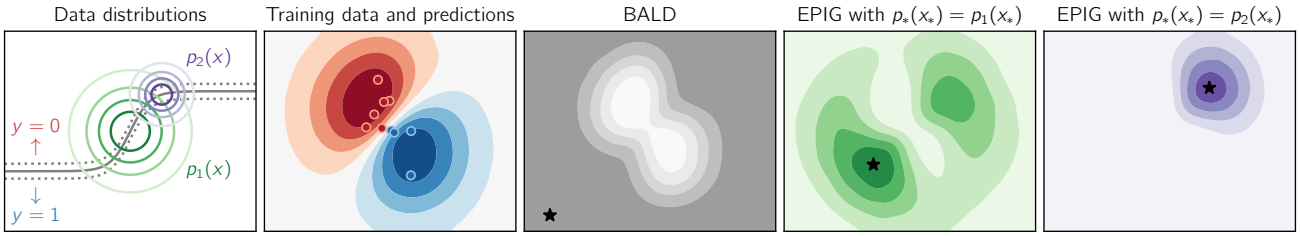


**Figure 1** The expected predictive information gain (EPIG) can differ dramatically from the expected information gain in the model parameters (BALD). BALD increases (darker shading) as we move away from the existing data, yielding a distant acquisition (star) when maximised. It seeks a global reduction in parameter uncertainty, regardless of any input distribution. In contrast EPIG is maximised only in regions of relatively high density under the target input distribution, $p_*(x_*)$. It seeks a reduction in parameter uncertainty only insofar as it reduces predictive uncertainty on samples from $p_*(x_*)$. See Section 5.1 for details.

be the same distribution that the pool of unlabelled inputs is drawn from, or it can be a distinct distribution that reflects a downstream task of interest.

We find that EPIG often produces notable gains in final predictive performance over BALD across a range of datasets and models. EPIG's gains are largest when the pool of unlabelled inputs contains a high proportion of irrelevant inputs with respect to the target input distribution. But its advantage still holds when the pool is directly drawn from this distribution. As such, it can provide a simple and effective drop-in replacement for BALD in many settings.

## 2 Background

We consider supervised learning of a probabilistic predictive model, $p_\phi(y|x)$, where $x$ is an input, $y$ is a label and $\phi$ indexes the set of models we can learn. We assume the model has some underlying stochastic parameters, $\theta$, such that we can write

$$p_\phi(y|x) = \mathbb{E}_{p_\phi(\theta)}[p_\phi(y|x,\theta)] \tag{1}$$

$$p_\phi(y_1, y_2|x_1, x_2) = \mathbb{E}_{p_\phi(\theta)}[p_\phi(y_1, y_2|x_1, x_2, \theta)]\,. \tag{2}$$

We also assume predictions are independent given $\theta$, which gives $p_\phi(y_1, y_2|x_1, x_2, \theta) = p_\phi(y_1|x_1, \theta)p_\phi(y_2|x_2, \theta)$.

The class of models satisfying our assumptions is broad. It includes effectively all Bayesian models, for which $p_\phi(y|x, \theta)$ is a fixed likelihood function and $p_\phi(\theta) = p(\theta|\mathcal{D})$ is a posterior given observed data $\mathcal{D}$. Also included are ensembles (Dietterich, 2000) and neural networks with stochasticity in a subset of parameters (Sharma et al, 2023).

### 2.1 Active learning

In the supervised setting, active learning involves having an algorithm select which labels to acquire when training a model (Settles, 2012). Typically acquisition takes place across a number of steps. Each step, $t$, consists of three parts (Figure 2). First, the algorithm selects a query input, $x_t$, to acquire a label for—or sometimes a batch of inputs (Kirsch et al, 2019). It does this by maximising an acquisition function, which is intended to capture the expected utility of acquiring the label for a given input. Often the set of candidate inputs is a fixed, finite pool, $\mathcal{D}_{\text{pool}} = \{x_i\}_{i=1}^N$. We focus on such settings, known as pool-based active learning (Lewis & Gale, 1994). Second, the algorithm samples a label, $y_t$, from the true conditional label distribution, $p(y|x = x_t)$, and incorporates $(x_t, y_t)$ into the training dataset. Third, the predictive model, $p_\phi(y|x)$, is updated.

### 2.2 Bayesian experimental design

Bayesian experimental design (Chaloner & Verdinelli, 1995; Lindley, 1956; Rainforth et al, 2023) is a formal framework for quantifying the information gain from an experiment. In the context of active learning we can view the input, $x$, as the design of the experiment and the acquired label, $y$, as the outcome of the experiment.

Let $\psi$ be a quantity we are aiming to learn about. Given a prior, $p(\psi)$, and a likelihood function, $p(y|x, \psi)$, both of which could be implicit, we can quantify the information gain in $\psi$ due to an experiment, $(x, y)$, as the reduction in Shannon entropy in $\psi$ that results from observing $(x, y)$:

$$\begin{aligned}\mathrm{IG}_\psi(x, y) &= \mathrm{H}[p(\psi)] - \mathrm{H}[p(\psi|x, y)] \\ &= \mathbb{E}_{p(\psi)}[-\log p(\psi)] - \mathbb{E}_{p(\psi)}[-\log p(\psi|x, y)]\,,\end{aligned}$$

where $p(\psi|x, y) \propto p(\psi)p(y|x, \psi)$ is the posterior that results from a Bayesian update on observing $(x, y)$.

Since $y$ is a random variable, we consider the expected information gain (EIG) across possible realisations of $y$, where this expectation is defined by simulating outcomes using the marginal predictive distribution, $p_\psi(y|x) = \mathbb{E}_{p(\psi)}[p(y|x, \psi)]$:

$$\mathrm{EIG}_\psi(x) = \mathbb{E}_{p_\psi(y|x)}[\mathrm{H}[p(\psi)] - \mathrm{H}[p(\psi|x, y)]]\,.$$

This is the expected reduction in uncertainty in $\psi$ after conditioning on $(x, y)$. Equivalently it is the mutual information between $\psi$ and $y$ given $x$ (Cavagnaro et al, 2010).

### 2.3 Bayesian active learning by disagreement (BALD)

Current approaches to Bayesian active learning generally target information gain in the model parameters by setting $\psi$ to $\theta$. This yields what is often referred to in the active-learning literature as BALD (Houlsby et al, 2011):

$$\begin{aligned}\mathrm{BALD}(x) &= \mathbb{E}_{p_\phi(y|x)}[\mathrm{H}[p_\phi(\theta)] - \mathrm{H}[p_\phi(\theta|x,y)]] \\ &= \mathbb{E}_{p_\phi(\theta)}[\mathrm{H}[p_\phi(y|x)] - \mathrm{H}[p_\phi(y|x,\theta)]]\,.\end{aligned}$$

Notably BALD is often used even when updating is not Bayesian, for example when using Monte Carlo dropout in a neural network (Gal et al, 2017).

## 3 The shortfalls of BALD

To establish the need for a new approach to Bayesian active learning, we highlight that BALD can be poorly suited to the prediction-oriented settings that constitute much of machine learning. We explain that this stems from the mismatch that can exist between parameter uncertainty and predictive uncertainty. We also highlight that targeting predictive uncertainty requires reasoning about what inputs we want to make predictions on, which BALD does not do.

### 3.1 Focusing on prediction

In statistics it is common for the model parameters to be valued in their own right (Beck & Arnold, 1977; Blei et al, 2003; Fisher, 1925). But in many machine-learning contexts, particularly the supervised settings where BALD is typically applied, the parameters are only valued insofar as they serve a prediction-oriented goal. We often, for example, seek the parameters that maximise the model's predictive performance on a test data distribution (Hastie et al, 2009). This frequentist notion of success often remains our motivation even if we use a Bayesian approach to data acquisition and/or learning (Komaki, 1996; Snelson & Ghahramani, 2005).

### 3.2 Not all information is equal

In some models, such as linear models, parameters and predictions are tightly coupled. This means that a reduction in parameter uncertainty typically yields a wholesale reduction in predictive uncertainty (Chaloner & Verdinelli, 1995). But more generally the coupling can be loose. Deep neural networks, for instance, can have substantial redundancy in their parameters (Belkin et al, 2019), while Bayesian nonparametric models can be thought of as having an infinite number of parameters (Hjort et al, 2010). When the coupling is loose, parameter uncertainty can be reduced without a corresponding reduction in predictive uncertainty on inputs of interest. In fact it is possible to gain an infinite amount of parameter information while seeing an arbitrarily small reduction in predictive uncertainty.

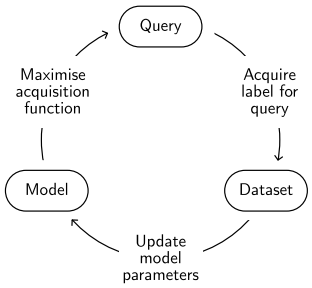


**Figure 2** Active learning typically loops over selecting a query, acquiring a label and updating the model parameters. In this work we focus on the acquisition function used to select queries. We consider pool-based settings, where the acquisition function is maximised across a fixed, finite set of unlabelled inputs.

**Example 1** *Consider a supervised-learning problem where $x \in \mathbb{R}$ is an input, $y \in \mathbb{R}$ is a label, and we use a model consisting of a Gaussian likelihood function, $p(y|x,\theta) = \mathcal{N}(\theta(x), 1)$, and a zero-mean Gaussian-process prior, $\theta \sim \mathrm{GP}(0, k)$, with covariance function $k(x,x') = \exp(-(x-x')^2)$. Suppose we are interested in making predictions in the interval $x_* \in [0,1]$. Now consider observing $y_1, y_2, \ldots, y_M$ at the input locations $M, 2M, \ldots, M^2$ for some $M \in \mathbb{N}^+$. In the limit $M \to \infty$, BALD converges to infinity while the EIG in the prediction of interest, $\theta(x_*)$, converges to zero:*

$$\lim_{M\to\infty} \mathrm{BALD}((M, 2M, \ldots, M^2)) = \infty$$
$$\lim_{M\to\infty} \mathrm{EIG}_{\theta(x_*)}((M, 2M, \ldots, M^2)) = 0.$$

See Appendix A for the proof. This example is a concrete demonstration that a high BALD score need not coincide with any reduction in the predictive uncertainty of interest, $\mathrm{EIG}_{\theta(x_*)}$. If the aim is to predict, then maximising BALD is not guaranteed to help to any extent whatsoever.

### 3.3 BALD has no notion of an input distribution

In order to reason about what information is relevant to prediction, we need some notion of the inputs on which we want to make predictions. Without this we have no mechanism to ensure the model we learn is well-suited to the task we care about. Our model could be highly effective on inputs from one region of input space but useless for typical samples from an input distribution of interest.

Appreciating the need to account for which inputs might arise at test time, it becomes clear why BALD can be problematic. BALD focuses on the model parameters in isolation, with no explicit connection to prediction. As such, it does not account for the distribution over inputs.

### 3.4 Real-world data can exacerbate this flaw

BALD can be particularly problematic in the very settings that often motivate active learning: those where we have access to a large pool of unlabelled inputs whose relevance to some task of interest varies widely. In contrast with the carefully curated datasets often used in basic research, real-world data is often drawn from many sources of varying fidelity and relation to the task. Pools of web-scraped audio, images and text are canonical examples of this. Active learning ought to help deal with the mess by identifying only the most useful inputs to label. But BALD can in fact be worse than random acquisition in these settings, targeting obscure data that is not helpful for prediction.

The experiment presented in Figure 3 highlights this flaw. As we increase the size of the pool that BALD is maximised over, inputs of greater obscurity become more likely to be included in the pool, and BALD produces worse and worse predictive accuracy. This result is corroborated by the work of Karamcheti et al (2021). Focusing on visual-question-answering tasks, they found that BALD failed to outperform random acquisition when using uncurated pools, and that a substantial amount of curation was required before this shortfall could be overturned.

### 3.5 Failure can occur without distribution shift

It might be tempting to just think of this problem with BALD as being analogous to the issues caused by train-test input-distribution shifts elsewhere in machine learning. But the problem is more deep-rooted than this: BALD has no notion of any input distribution in the first place. This is why increasing the size of the pool can induce failures as in Figure 3, without any distribution shift or changes to the distribution that the pool inputs are drawn from. Distribution shift can cause additional problems for BALD, as some of the results in Section 5 show. But it is by no means a necessary condition for failure to occcur.

### 3.6 Filtering heuristics are not a general solution

We might suppose we could just discard irrelevant data before deploying BALD. But this filtering process would require us to be able to determine each input's relevance at the outset of training, which is impractical in many cases. Even if we have access to a target input distribution, this on its own can be insufficient for judging relevance to a task of interest. A candidate input could have relatively low density under the target distribution but nevertheless share high-level features with a target input, such that the two inputs' labels are highly mutually informative. With high-dimensional inputs, it can also be surprisingly difficult to identify unrepresentative inputs purely through their density (Nalisnick et al, 2018). Rather than trying to design an auxiliary process to mitigate BALD's problematic behaviour, we seek an acquisition function that can automatically determine what is relevant.

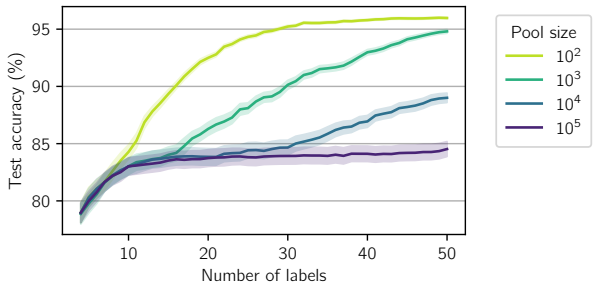


**Figure 3** BALD can fail catastrophically on big pools. A bigger pool typically contains more inputs with low density under the data-generating distribution. Often these inputs are of low relevance if the aim is to maximise expected predictive performance. BALD can nevertheless favour these inputs. See Figure 1 for intuition and Section 5.1 for details.

## 4 Expected predictive information gain

Motivated by BALD's weakness in prediction-oriented settings, we return to the framework of Bayesian experimental design that underlies BALD, and derive an acquisition function that we call the expected predictive information gain (EPIG). Whereas BALD targets a reduction in parameter uncertainty, EPIG directly targets a reduction in predictive uncertainty on inputs of interest.

To reason about reducing predictive uncertainty, we need an explicit notion of the predictions we want to make with our model. We therefore introduce a random target input, $x_* \sim p_*(x_*)$, and define our goal to be confident prediction of $y_*|x_*$ for samples from the target input distribution.

To derive EPIG we first consider the information gain in $y_*$ that results from conditioning on new data, $(x, y)$:

$$\mathrm{IG}_{y_*}(x, y, x_*) = \mathrm{H}[p_\phi(y_*|x_*)] - \mathrm{H}[p_\phi(y_*|x_*, x, y)]\,,$$

where $p_\phi(y_*|x_*, x, y) = \mathbb{E}_{p_\phi(\theta|x,y)}[p_\phi(y_*|x_*, \theta)]$. Note that this is a function of $x_*$ as well as $x$ and $y$. Next we take an expectation over both the random target input, $x_*$, and the unknown label, $y$:

$$\mathrm{EPIG}(x) = \mathbb{E}_{p_*(x_*)p_\phi(y|x)}[\mathrm{IG}_{y_*}(x, y, x_*)]\,.$$

Thus we see that EPIG is the expected reduction in predictive uncertainty at a randomly sampled target input, $x_*$.

There are other interpretations too. EPIG is the mutual information between $(x_*, y_*)$ and $y$ given $x$, $\mathrm{I}((x_*, y_*); y|x)$:

$$\mathrm{EPIG}(x) = \mathbb{E}_{p_*(x_*)p_\phi(y,y_*|x,x_*)}\left[\log \frac{p_\phi(y, y_*|x, x_*)}{p_\phi(y|x)p_\phi(y_*|x_*)}\right]. \tag{3}$$

This is equivalent to $\mathbb{E}_{p_*(x_*)}[\mathrm{I}(y; y_*|x, x_*)]$, the expected mutual information between $y$ and $y_*$ given $x$ and $x_*$, which can be written as an expected KL divergence between $p_\phi(y, y_*|x, x_*)$ and $p_\phi(y|x)p_\phi(y_*|x_*)$: $\mathrm{EPIG}(x) =$

$$\mathbb{E}_{p_*(x_*)}[\mathrm{KL}[p_\phi(y, y_*|x, x_*) \,\|\, p_\phi(y|x)p_\phi(y_*|x_*)]]\,. \quad (4)$$

We can also take a frequentist perspective. In classification settings EPIG is equal (up to a constant) to the negative expected generalisation error under a cross-entropy loss:

$$\mathrm{EPIG}(x) = \mathbb{E}_{p_*(x_*)p_\phi(y,y_*|x,x_*)}[\log p_\phi(y_*|x_*, x, y)] + c, \quad (5)$$

where $c$ is a constant and we have used the fact that $\mathrm{H}[p_\phi(y_*|x_*)]$ is constant with respect to $x$. Maximising EPIG can therefore be thought of as seeking to minimise the expected generalisation error after acquisition.

### 4.1 Sampling target inputs

EPIG involves an expectation with respect to a target input distribution, $p_*(x_*)$. In practice we estimate this expectation by Monte Carlo and so require a sampling mechanism.

In many active-learning settings an input distribution is implied by the existence of a pool of unlabelled inputs. There are cases where we know (or are happy to assume) the pool has been sampled from $p_*(x_*)$. Alternatively we might be forced to assume this is the case: perhaps we know the pool is not sampled from $p_*(x_*)$ but lack access to anything better. In these cases we can simply subsample from the pool to obtain samples of $x_*$. Empirically we find that this can work well relative to acquisition with BALD (Section 5).

Another important case is where we have access to samples from $p_*(x_*)$ but we cannot label them. Limits on the ability to acquire labels might arise due to privacy-related and other ethical concerns, geographical restrictions, the complexity of the labelling process for some inputs, or the presence of commercially sensitive information in some inputs. At the same time there might be a pool of inputs for which we have no labelling restrictions. In a case like this we can estimate EPIG using samples from $p_*(x_*)$ while using only the pool as a source of candidates for labelling. Thus we can target information gain in predictions on samples from $p_*(x_*)$ without labelling those samples themselves.

A further scenario that we might encounter is a classification problem where the pool is representative of the target class-conditional input distribution but not the target marginal class distribution: that is, $p_{\mathrm{pool}}(x_*|y_*) = p_*(x_*|y_*)$ but $p_{\mathrm{pool}}(y_*) \neq p_*(y_*)$. The pool might, for example, consist of uncurated web-scraped inputs from many more classes than those we care about. In this scenario it can often be the case that we know or can reasonably approximate the distribution over classes that we are targeting, $p_*(y_*)$. With this we can approximately sample from $p_*(x_*)$ using a combination of our model, $p_\phi(y|x)$, and the pool. We first note that

$$p_{\mathrm{pool}}(x_*|y_*) = \frac{p_{\mathrm{pool}}(x_*)p_{\mathrm{pool}}(y_*|x_*)}{\int p_{\mathrm{pool}}(x)p_{\mathrm{pool}}(y = y_*|x)dx}.$$

Then, using the fact that $p_{\mathrm{pool}}(x_*|y_*) = p_*(x_*|y_*)$, we get

$$\begin{aligned} p_*(x_*) &= \sum_{y_*} p_*(y_*)p_*(x_*|y_*) \\ &= p_{\mathrm{pool}}(x_*) \sum_{y_*} \frac{p_*(y_*)p_{\mathrm{pool}}(y_*|x_*)}{\int p_{\mathrm{pool}}(x)p_{\mathrm{pool}}(y = y_*|x)dx} \\ &\approx p_{\mathrm{pool}}(x_*) \sum_{y_*} \frac{p_*(y_*)p_\phi(y_*|x_*)}{\frac{1}{N}\sum_{x\in\mathcal{D}_{\mathrm{pool}}} p_\phi(y = y_*|x)} \\ &= p_{\mathrm{pool}}(x_*)w(x_*), \end{aligned}$$

where we have approximated $p_{\mathrm{pool}}(y_*|x_*)$ with our model. Now we can approximately sample from $p_*(x_*)$ by subsampling inputs from the pool using a categorical distribution with probabilities $w(x_*)/N$.

### 4.2 Estimation

The best way to estimate EPIG depends on the task and model of interest. In the empirical evaluations in this paper we focus on classification problems and use models whose marginal and joint predictive distributions are not known in closed form. This leads us to use $\mathrm{EPIG}(x) \approx$

$$\frac{1}{M}\sum_{j=1}^{M} \mathrm{KL}\big[\hat{p}_\phi(y, y_*|x, x_*^j) \,\|\, \hat{p}_\phi(y|x)\hat{p}_\phi(y_*|x_*^j)\big]\,, \quad (6)$$

where $x_*^j \sim p_*(x_*)$, $\hat{p}$ denote Monte Carlo approximations of the predictive distributions in Equations 1 and 2. Classification is an instance of where the required expectation over $y$ and $y_*$ can be computed analytically, such that our only required estimation is from marginalisations over $\theta$.

If we cannot integrate over $y$ and $y_*$ analytically, we can revert to nested Monte Carlo estimation (Rainforth et al, 2018). For this we first note that, using Equation 2, we can sample $y, y_* \sim p_\phi(y, y_*|x, x_*)$ exactly by drawing a $\theta$ and then a $y$ and $y_*$ conditioned on this $\theta$. By also drawing samples for $\theta$, we can then construct the estimator $\mathrm{EPIG}(x) \approx$

$$\frac{1}{M}\sum_{j=1}^{M} \log \frac{K\sum_{i=1}^{K} p_\phi(y^j|x, \theta_i)p_\phi(y_*^j|x_*^j, \theta_i)}{\sum_{k=1}^{K} p_\phi(y^j|x, \theta_k)\sum_{k=1}^{K} p_\phi(y_*^j|x_*^j, \theta_k)}, \quad (7)$$

where $y^j, y_*^j \sim p_\phi(y, y_*|x, x_*^j)$, $\theta_i \sim p_\phi(\theta)$, and $x_*^j \sim p_*(x_*)$. Subject to some weak assumptions on $p_\phi$, this estimator converges as $K, M \to \infty$ (Rainforth et al, 2018).

The EPIG estimators in Equations 6 and 7 each have a total computational cost of $O(MK)$. This is comparable to BALD estimation for regression problems. But it can be

more expensive than BALD estimation for classification problems: BALD can be collapsed to a non-nested Monte Carlo estimation for an $O(K)$ cost, but EPIG cannot.

Other possible estimation schemes include a variational approach inspired by Foster et al (2019). This is too expensive to be practically applicable in the settings we consider but could be useful elsewhere. See Appendix E for details.

## 5 Experiments

For consistency with existing work on active learning for prediction, our empirical evaluation of EPIG focuses on classification problems. Code for reproducing our results is available at `github.com/fbickfordsmith/epig`.

### 5.1 Synthetic data (Figures 1, 3 and 4)

First we demonstrate the difference between BALD and EPIG in a setting that is easy to visualise and understand: binary classification with two-dimensional inputs.

**Data** The first input distribution of interest, denoted $p_1(x)$ in Figure 1, is a bivariate Student's $t$ distribution with $\nu = 5$ degrees of freedom, location $\mu = [0, 0]$ and scale matrix $\Sigma = 0.8I$. The second distribution, denoted $p_2(x)$ in Figure 1, is a scaled and shifted version of the first, with parameters $\nu = 5$, $\mu \approx [0.8, 0.9]$ and $\Sigma = 0.4I$. This serves to illustrate in Figure 1 how EPIG's value changes with the target input distribution; it is not used elsewhere. The conditional label distribution is defined as $p(y = 1|x) = \Phi(20(\tanh(2x_{[1]}) - x_{[2]}))$, where $x_{[i]}$ denotes the component of input $x$ in dimension $i$, and $\Phi$ is the cumulative distribution function of the standard normal distribution. For the training data in Figure 1, we sample ten input-label pairs, $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{10}$, where $x_i, y_i \sim p(y|x)p_1(x)$. Likewise we sample $\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^{10,000}$ for evaluating the model's performance in active learning.

**Model and training** We use a model with a probit likelihood function, $p_\phi(y = 1|x, \theta) = \Phi(\theta(x))$, where $\Phi$ is defined as above, and a Gaussian-process prior, $\theta \sim \text{GP}(0, k)$, where $k(x, x') = 10 \cdot \exp\left(-\|x - x'\|^2/2\right)$. The posterior over latent-function values cannot be computed exactly so we optimise an approximation to it using variational inference (Hensman et al, 2015). To do this we run 10,000 steps of full-batch gradient descent using a learning rate of 0.005 and a momentum factor of 0.95.

**Active learning** We initialise the training dataset, $\mathcal{D}_{\text{train}}$, with two randomly sampled inputs from each class. Thereafter we run the active-learning loop described in Section 2.1 until a budget of 50 labels is used up. We acquire data using three acquisition functions: random, BALD and EPIG. Random acquisition involves sampling uniformly from the pool without replacement. We estimate BALD using Equation 11 and EPIG using Equation 6, in both cases drawing 5,000 samples from the model's approximate posterior. For EPIG we sample $x_* \sim p_1(x_*)$. After each time the model is trained, we evaluate its predictive accuracy on $\mathcal{D}_{\text{test}}$ as defined above. Using a different random-number-generator seed each time, we run active learning with each acquisition function 100 times. We report the test accuracy (mean $\pm$ standard error) as a function of the size of $\mathcal{D}_{\text{train}}$.

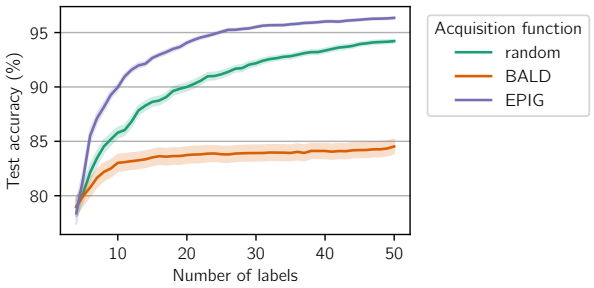


**Figure 4** In contrast with BALD, EPIG deals effectively with a big pool ($10^5$ unlabelled inputs). BALD is overwhelming counterproductive relative to random acquisition. See Figures 1 and 3 for intuition and Section 5.1 for details.

**Discussion** Figure 4 shows a striking gap between BALD and EPIG in active learning. Figures 1 and 3 provide some intuition about the underlying cause of this disparity: BALD has a tendency to acquire labels at the extrema of the input space, regardless of their relevance to the predictive task of interest.

### 5.2 UCI data (Figure 5)

Next we compare BALD and EPIG in a broader range of settings. We use problems drawn from a repository maintained at UC Irvine (UCI; Dua & Graff, 2017), which has been widely used as a data source in past work on Bayesian methods (Gal & Ghahramani, 2016; Lakshminarayanan et al, 2017; Sun et al, 2018; Zhang et al, 2018). The problems we use vary in terms of the number of classes, the input dimension and any divergence between the pool and target data distributions. We assume knowledge of $p_*(x_*)$ when estimating EPIG but note that this assumption has little significance if $p_{\text{pool}}(x)$ and $p_*(x_*)$ match, which is true for two out of the three problems.

**Data** We use three classification datasets from the UCI repository, each with a different number of classes, $C$, and input dimension, $D$: Magic ($C = 2$, $D = 11$), Satellite ($C = 6$, $D = 36$) and Vowels ($C = 11$, $D = 10$). The inputs are telescope readings in Magic, satellite images in Satellite and speech recordings in Vowels. Magic is interesting because it serves as a natural instance of a mismatch between pool and target distributions (see Appendix F.1).

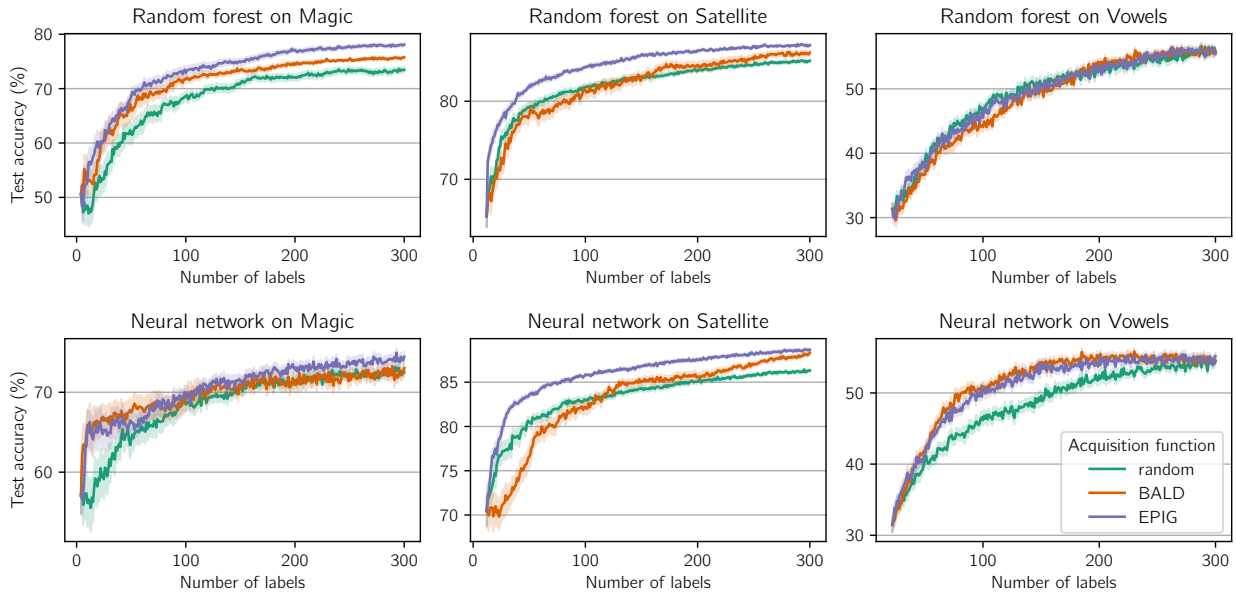


**Figure 5** EPIG outperforms or matches BALD across three standard classification problems from the UCI machine-learning repository (Magic, Satellite and Vowels) and two models (random forest and neural network). See Section 5.2 for details.

**Models and training** We use two different models. The first is a random forest (Breiman, 2001). To emphasise that EPIG works with an off-the-shelf setup, we use the Scikit-learn (Pedregosa et al, 2011) implementation with its default parameters. The second model is a dropout-enabled fully connected neural network with three hidden layers and a softmax output layer. A dropout rate of 0.1 is used during both training and testing. Training the neural network consists of running up to 50,000 steps of full-batch gradient descent using a learning rate of 0.1. We use a loss function consisting of the negative log likelihood (NLL) of the training data combined with an $l_2$ regulariser (with coefficient 0.0001) on the model parameters. To mitigate overfitting we use early stopping: we track the model's NLL on a small validation set (approximately 20% of the size of the training-label budget) and stop training if this NLL does not decrease for more than 10,000 consecutive steps. We then restore the model parameters to the configuration that achieved the lowest validation-set NLL.

**Active learning** We use largely the same setup as described in Section 5.1. Here the label budget is 300 and we run active learning 20 times with different seeds. We use the same BALD and EPIG estimators as before, treating each tree in the random forest as a different $\theta$ value, and treating each stochastic forward pass through the neural network (we compute 100 of them) as corresponding to a different $\theta$ value. To estimate EPIG we sample $x_*$ from a set of inputs designed to be representative of $p_*(x_*)$.

**Discussion** Figure 5 shows EPIG performing convincingly better than BALD in some cases while matching it in others. These results provide broader validation of EPIG, complementing the results in Figure 4.

### 5.3 MNIST data (Figures 6, 7 and 8)

Finally we evaluate BALD and EPIG in settings intended to capture challenges that occur when applying deep neural networks to high-dimensional inputs. Our starting point is the MNIST dataset (LeCun et al, 1998), in which each input is an image of a handwritten number between 0 and 9. This dataset has been widely used in related work on Bayesian active learning with deep neural networks (Beluch et al, 2018; Gal et al, 2017; Jeon, 2020; Kirsch et al, 2019, 2022; Lee & Kim, 2019; Tran et al, 2019). We construct three settings based on this dataset, each corresponding to a different practical scenario: Curated MNIST, Unbalanced MNIST and Redundant MNIST.

As well as investigating how BALD and EPIG perform across these settings, we seek to understand the effect on EPIG of varying the amount of knowledge we have of the target data distribution, $p_*(x_*)$. To this end we assume we know this for one set of runs (Figure 6) and then relax this assumption for another set (Figure 7).

**Data** Curated MNIST is intended to reflect the data often used in academic machine-learning research. The pool and target class distributions, $p_{\text{pool}}(y)$ and $p_*(y_*)$, are both uniform over all 10 classes. In terms of class distributions, this effectively represents a worst-case scenario for active learning relative to random acquisition. Given matching class-conditional input distributions, namely $p_{\text{pool}}(x_*|y_*) = p_*(x_*|y_*)$, uniformly sampling from the pool input distri-

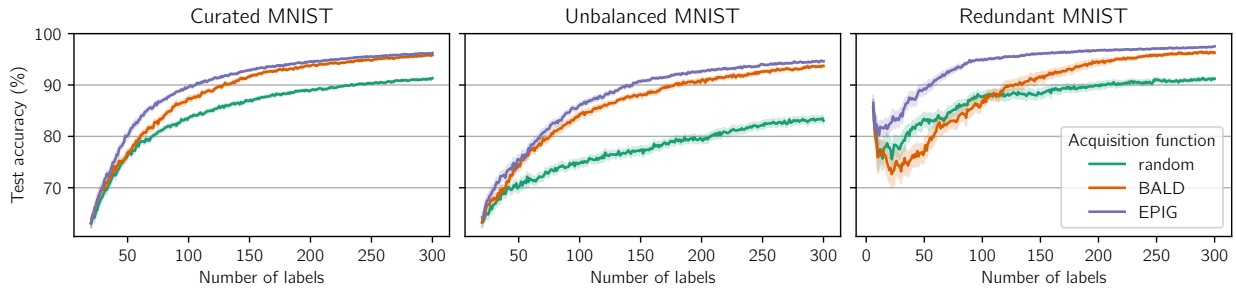


**Figure 6** EPIG outperforms BALD across three image-classification settings. Curated MNIST reflects the data often used in academic research. The pool and target input distributions, $p_{\text{pool}}(x)$ and $p_*(x_*)$ match; the marginal class distributions, $p_{\text{pool}}(y)$ and $p_*(y_*)$, are uniform. Unbalanced MNIST is a step closer to real-world data. While $p_*(y_*)$ remains uniform, $p_{\text{pool}}(y)$ is non-uniform: the pool contains more inputs from some classes than others. Redundant MNIST simulates a separate practical problem. Whereas $p_*(y_*)$ only has nonzero mass on two classes of interest, $p_{\text{pool}}(y)$ has substantial mass across all classes. See Section 5.3 for details.

bution, $p_{\text{pool}}(x)$, is equivalent to uniformly sampling from the target input distribution, $p_*(x_*)$. Thus random acquisition is a strong baseline in this setting.

Unbalanced MNIST is a step closer to real-world data. We might expect $p_*(y_*)$ to be uniform—that is, the task of interest might involve classifying examples in equal proportion from each class—but it could be difficult to curate a pool that is similarly uniform in its class distribution. To reflect this we consider a case with a non-uniform $p_{\text{pool}}(y)$: classes 0 to 4 each have probability $1/55$ and classes 5 to 9 each have probability $10/55$.

Redundant MNIST captures a separate practical problem that occurs, for instance, when using web-scraped data. The pool might contain inputs from many more classes than we want to focus on in the predictive task of interest. To simulate this we suppose that the task involves classifying just images of 1s and 7s, occurring in equal proportion—that is, $p_*(y_*)$ places probability mass of $1/2$ on class 1, $1/2$ on class 7, and 0 on all other classes—while $p_{\text{pool}}(y)$ is uniform over all 10 classes. If the acquisition function selects an input from a class other than 1 and 7, the labelling function produces a "neither" label. Thus we have three-way classification during training: 1 vs 7 vs neither.

**Model and training** For both runs we use the same dropout-enabled convolutional neural network as used by Kirsch et al (2019). The dropout rate here is 0.5. Training is similar to as described in Section 5.2, except that the learning rate is 0.01 and early stopping triggers after 5,000 consecutive steps of non-decreasing validation-set NLL.

**Active learning** Initially we retain the setup described in Section 5.2, with $p_*(x_*)$ known (Figure 6). Then we investigate the sensitivity of EPIG to removing full access to $p_*(x_*)$, focusing on two different settings (Figure 7). In one we assume knowledge of $p_*(y_*)$ and use the resampling technique discussed in Section 4.1. In the other we simply sample target inputs from the pool: $x_* \sim p_{\text{pool}}(x_*)$.

**Discussion** Figure 6 shows EPIG again outperforming BALD and random across all three dataset variants when given access to $p_*(x_*)$. (EPIG additionally beats predictive entropy (Settles & Craven, 2008) and BADGE (Ash et al, 2020), acquisition functions commonly studied in the active-learning literature, as shown in Appendix G.) EPIG's advantage over BALD is appreciable on Curated MNIST and Unbalanced MNIST. But it is emphatic on Redundant MNIST. This suggests EPIG is particularly useful when working with highly diverse pools.

Figure 7 shows the even more impressive result that EPIG retains its strong performance even when no access to $p_*(x_*)$ is assumed. We thus see that EPIG provides a good degree of robustness in its performance to the level of knowledge about the target data distribution.

## 6 Related work

The idea of using the EIG to quantify the utility of data was introduced by Lindley (1956) and has a long history of use in experimental design (Chaloner & Verdinelli, 1995; Rainforth et al, 2023). The framework of Bayesian experimental design has many applications outside active learning, and in these applications the model parameters are commonly the quantity of interest—Bayesian optimisation (Hennig & Schuler, 2012; Hernández-Lobato et al, 2014; Villemonteix et al, 2009) being a notable exception. The EIG in the parameters is thus often a natural acquisition function.

The EIG in the parameters was suggested as an acquisition function for active learning by MacKay (1992a,b), who called it the total information gain. It was popularised as BALD by Houlsby et al (2011), while its use with deep neural networks was demonstrated by Gal et al (2017). Despite its shortfalls, BALD has been widely used in cases where the model's predictions, not the model parameters, are the true objects of interest (Atighehchian et al, 2020; Beluch et al, 2018; Gal et al, 2017; Houlsby et al, 2011; Jeon, 2020; Kirsch et al, 2019, 2022; Lee & Kim, 2019;

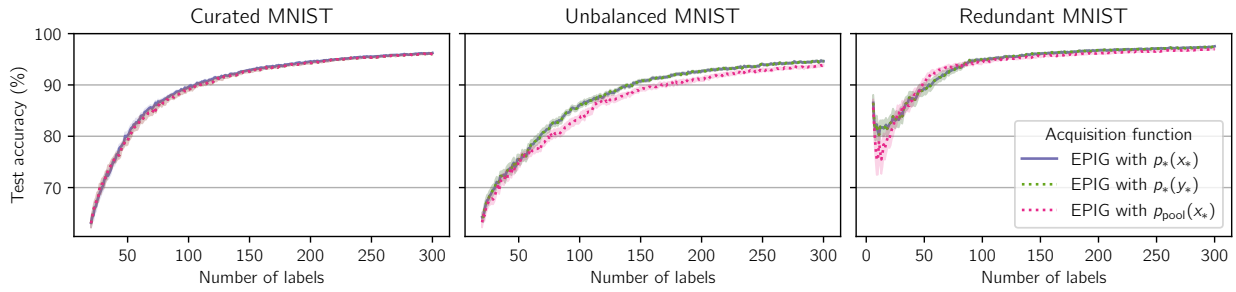


**Figure 7** Even without knowledge of the target input distribution, $p_*(x_*)$, EPIG retains its strong performance on Curated MNIST, Unbalanced MNIST and Redundant MNIST. "EPIG with $p_*(x_*)$" assumes exact samples from $p_*(x_*)$, as in Figure 6. "EPIG with $p_*(y_*)$" corresponds to using the approximate-sampling scheme outlined in Section 4.1, using knowledge of $p_*(y_*)$. "EPIG with $p_{\text{pool}}(x_*)$" corresponds to using samples from the pool as a proxy for $p_*(x_*)$. See Section 5.3 for details.

Munjal et al, 2022; Pinsler et al, 2019; Shen et al, 2018; Siddhant & Lipton, 2018; Tran et al, 2019).

Maximising the information gathered about a quantity other than the model parameters has been proposed a number of times as an approach to active learning. Perhaps most relevant to our work, MacKay (1992a,b) introduced an acquisition function called the mean marginal information gain. Based on a Gaussian approximation of the posterior over the model parameters, it measures the average information gain in the predictions made on a fixed set of inputs. Though it has since received surprisingly little attention in the literature, it was discussed by Huszár (2013) and later used by Wang et al (2021) to evaluate the quality of predictive-posterior correlations. Seeking information gain on a fixed set of inputs—in contrast with the input distribution considered by EPIG—is a transductive approach to active learning (Vapnik, 1982; Yu et al, 2006).

Aside from the work of MacKay (1992a,b), there are numerous prediction-oriented methods (Afrabandpey et al, 2019; Chapelle, 2005; Cohn, 1993; Cohn et al, 1996; Daee et al, 2016; Donmez & Carbonell, 2008; Evans et al, 2015; Filstroff et al, 2021; Krause et al, 2008; Seo et al, 2000; Sundin et al, 2018, 2019; Tan et al, 2021; Yu et al, 2006; Zhao et al, 2021a,b,c; Zhu et al, 2003). Many of these, with notable examples including the work of Cohn et al (1996) and Krause et al (2008), are tied to a particular model class or approximation scheme and so lack EPIG's generality.

There is an additional limitation associated with techniques based on the idea, due to Roy & McCallum (2001), of measuring the expected loss reduction that would result from updating the model on a given input-label pair. These techniques often require updating the model within the computation of the acquisition function, which can be extremely expensive. Despite a strong conceptual connection to the acquisition function proposed by Roy & McCallum (2001), EPIG allows a significantly lower computational cost: its information-theoretic formulation allows us to derive an estimator that does not require nested model updating.

## 7 Conclusion

We have demonstrated that BALD, a widely used acquisition function for Bayesian active learning, can be suboptimal. While much of machine learning focuses on prediction, BALD targets information gain in a model's parameters in isolation and so can seek labels that have limited relevance to the predictions of interest. Motivated by this, we have proposed EPIG, an acquisition function that targets information gain in terms of predictions. Our results show EPIG outperforming BALD across a number of data settings (low- and high-dimensional inputs, varying degrees of divergence between the pool and target data distributions, and varying degrees of knowledge of the target distribution) and across multiple different models. This suggests EPIG can serve as a compelling drop-in replacement for BALD, with particular scope for performance gains when using large, diverse pools of unlabelled data.

## Acknowledgements

We thank Arnaud Doucet, Mike Osborne, Jannik Kossen, Jan Brauner, Joost van Amersfoort, Florentin Coeurdoux and the anonymous reviewers of this paper for their feedback. Freddie Bickford Smith and Andreas Kirsch are supported by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (EP/L015897/1, EP/S024050/1).

## References

Afrabandpey, Peltola, & Kaski (2019). Human-in-the-loop active covariance learning for improving prediction in small data sets. *International Joint Conference on Artificial Intelligence*.

Ardila, Branson, Davis, Kohler, Meyer, Henretty, Morais, Saunders, Tyers, & Weber (2020). Common Voice: a massively-multilingual speech corpus. *Language Resources and Evaluation Conference*.

Arthur & Vassilvitskii (2007). k-means++: the advantages of careful seeding. *ACM-SIAM Symposium on Discrete Algorithms*.

Ash, Zhang, Krishnamurthy, Langford, & Agarwal (2020). Deep batch active learning by diverse, uncertain gradient lower bounds. *International Conference on Learning Representations*.

Atighehchian, Branchaud-Charron, & Lacoste (2020). Bayesian active learning for production, a systematic study and a reusable library. *Workshop on "Uncertainty and Robustness in Deep Learning", International Conference on Machine Learning*.

Atlas, Cohn, & Ladner (1989). Training connectionist networks with queries and selective sampling. *Conference on Neural Information Processing Systems*.

Barber & Agakov (2003). The IM algorithm: a variational approach to information maximization. *Conference on Neural Information Processing Systems*.

Beck & Arnold (1977). *Parameter Estimation in Engineering and Science*. Wiley.

Belkin, Hsu, Ma, & Mandal (2019). Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proceedings of the National Academy of Sciences*.

Beluch, Genewin, Nürnberger, & Kohler (2018). The power of ensembles for active learning in image classification. *Conference on Computer Vision and Pattern Recognition*.

Blei, Ng, & Jordan (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*.

Breiman (2001). Random forests. *Machine Learning*.

Cavagnaro, Myung, Pitt, & Kujala (2010). Adaptive design optimization: a mutual information-based approach to model discrimination in cognitive science. *Neural Computation*.

Chaloner & Verdinelli (1995). Bayesian experimental design: a review. *Statistical Science*.

Chapelle (2005). Active learning for Parzen window classifier. *International Conference on Artificial Intelligence and Statistics*.

Cohn (1993). Neural network exploration using optimal experiment design. *Conference on Neural Information Processing Systems*.

Cohn, Ghahramani, & Jordan (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*.

Cover & Thomas (2005). *Elements of Information Theory*. John Wiley and Sons.

Daee, Peltola, Soare, & Kaski (2016). Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction. *Machine Learning*.

Dietterich (2000). Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*.

Donmez & Carbonell (2008). Optimizing estimated loss reduction for active sampling in rank learning. *International Conference on Machine Learning*.

Dua & Graff (2017). UCI Machine Learning Repository. archive.ics.uci.edu/ml.

Evans, Adams, & Anagnostopoulos (2015). Estimating optimal active learning via model retraining improvement. *arXiv*.

Filstroff, Sundin, Mikkola, Tiulpin, Kylmäoja, & Kaski (2021). Targeted active learning for Bayesian decision-making. *arXiv*.

Fisher (1925). *Statistical Methods for Research Workers*. Oliver and Boyd.

Foster, Jankowiak, Bingham, Horsfall, Teh, Rainforth, & Goodman (2019). Variational Bayesian optimal experimental design. *Conference on Neural Information Processing Systems*.

Gal & Ghahramani (2016). Dropout as a Bayesian approximation: representing model uncertainty in deep learning. *International Conference on Machine Learning*.

Gal, Islam, & Ghahramani (2017). Deep Bayesian active learning with image data. *International Conference on Machine Learning*.

Gemmeke, Ellis, Freedman, Jansen, Lawrence, Moore, Plakal, & Ritter (2017). Audio Set: an ontology and human-labeled dataset for audio events. *International Conference on Acoustics, Speech and Signal Processing*.

Hastie, Tibshirani, Friedman, & Friedman (2009). *The Elements of Statistical Learning*. Springer.

Hennig & Schuler (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*.

Hensman, Matthews, & Ghahramani (2015). Scalable variational Gaussian process classification. *International Conference on Artificial Intelligence and Statistics*.

Hernández-Lobato, Hoffman, & Ghahramani (2014). Predictive entropy search for efficient global optimization of black-box functions. *Conference on Neural Information Processing Systems*.

Hjort, Holmes, Müller, & Walker (2010). *Bayesian Nonparametrics*. Cambridge University Press.

Horn & Johnson (2012). *Matrix Analysis*. Cambridge University Press.

Houlsby (2014). *Efficient Bayesian active learning and matrix modelling*. PhD thesis, University of Cambridge.

Houlsby, Huszár, Ghahramani, & Lengyel (2011). Bayesian active learning for classification and preference learning. *arXiv*.

Huszár (2013). *Scoring rules, divergences and information in Bayesian machine learning*. PhD thesis, University of Cambridge.

Jeon (2020). ThompsonBALD: a new approach to Bayesian batch active learning for deep learning via Thompson sampling. Master's thesis, University College London.

Karamcheti, Krishna, Fei-Fei, & Manning (2021). Mind your outliers! Investigating the negative impact of outliers on active learning for visual question answering. *International Joint Conference on Natural Language Processing*.

Kirsch, Farquhar, Atighehchian, Jesson, Branchaud-Charron, & Gal (2022). Stochastic batch acquisition for deep active learning. *arXiv*.

Kirsch, van Amersfoort, & Gal (2019). BatchBALD: efficient and diverse batch acquisition for deep Bayesian active learning. *Conference on Neural Information Processing Systems*.

Komaki (1996). On asymptotic properties of predictive distributions. *Biometrika*.

Krause, Singh, & Guestrin (2008). Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*.

Lakshminarayanan, Pritzel, & Blundell (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Conference on Neural Information Processing Systems*.

LeCun, Bottou, Bengio, & Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Lee & Kim (2019). BALD-VAE: generative active learning based on the uncertainties of both labeled and unlabeled data. *International Conference on Robot Intelligence Technology and Applications*.

Lewis & Gale (1994). A sequential algorithm for training text classifiers. *ACM-SIGIR Conference on Research and Development in Information Retrieval*.

Lindley (1956). On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*.

MacKay (1992a). The evidence framework applied to classification networks. *Neural Computation*.

MacKay (1992b). Information-based objective functions for active data selection. *Neural Computation*.

Mahajan, Girshick, Ramanathan, He, Paluri, Li, Bharambe, & van der Maaten (2018). Exploring the limits of weakly supervised pretraining. *European Conference on Computer Vision*.

Munjal, Hayat, Hayat, Sourati, & Khan (2022). Towards robust and reproducible active learning using neural networks. *Conference on Computer Vision and Pattern Recognition*.

Nalisnick, Matsukawa, Teh, Gorur, & Lakshminarayanan (2018). Do deep generative models know what they don't know? *arXiv*.

Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, VanderPlas, Passos, Cournapeau, Brucher, Perrot, & Duchesnay (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*.

Pinsler, Gordon, Nalisnick, & Hernández-Lobato (2019). Bayesian batch active learning as sparse subset approximation. *Conference on Neural Information Processing Systems*.

Radford, Kim, Hallacy, Ramesh, Goh, Agarwal, Sastry, Askell, Mishkin, Clark, Krueger, & Sutskever (2021). Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*.

Raffel, Shazeer, Roberts, Lee, Narang, Matena, Zhou, Li, & Liu (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Rainforth, Cornish, Yang, Warrington, & Wood (2018). On nesting Monte Carlo estimators. *International Conference on Machine Learning*.

Rainforth, Foster, Ivanova, & Bickford Smith (2023). Modern Bayesian experimental design. *arXiv*.

Roy & McCallum (2001). Toward optimal active learning through sampling estimation of error reduction. *International Conference on Machine Learning*.

Seo, Wallat, Graepel, & Obermayer (2000). Gaussian process regression: active data selection and test point rejection. *International Joint Conference on Neural Networks*.

Settles (2012). *Active Learning*. Morgan and Claypool.

Settles & Craven (2008). An analysis of active learning strategies for sequence labeling tasks. *Conference on Empirical Methods in Natural Language Processing*.

Sharma, Farquhar, Nalisnick, & Rainforth (2023). Do Bayesian neural networks need to be fully stochastic? *International Conference on Artificial Intelligence and Statistics*.

Shen, Yun, Lipton, Kronrod, & Anandkumar (2018). Deep active learning for named entity recognition. *International Conference on Learning Representations*.

Siddhant & Lipton (2018). Deep Bayesian active learning for natural language processing: results of a large-scale empirical study. *arXiv*.

Snelson & Ghahramani (2005). Sparse Gaussian processes using pseudo-inputs. *Conference on Neural Information Processing Systems*.

Sun, Shrivastava, Singh, & Gupta (2017). Revisiting unreasonable effectiveness of data in deep learning era. *International Conference on Computer Vision*.

Sun, Zhang, Wang, Zeng, Li, & Grosse (2018). Differentiable compositional kernel learning for Gaussian processes. *International Conference on Machine Learning*.

Sundin, Peltola, Micallef, Afrabandpey, Soare, Majumder, Daee, He, Serim, Havulinna, Heckman, Jacucci, Marttinen, & Kaski (2018). Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge. *Bioinformatics*.

Sundin, Schulam, Siivola, Vehtari, Saria, & Kaski (2019). Active learning for decision-making from imbalanced observational data. *International Conference on Machine Learning*.

Tan, Du, & Buntine (2021). Diversity enhanced active learning with strictly proper scoring rules. *Conference on Neural Information Processing Systems*.

Tran, Do, Reid, & Carneiro (2019). Bayesian generative active deep learning. *International Conference on Machine Learning*.

Vapnik (1982). *Estimation of Dependences Based on Empirical Data*. Springer.

Villemonteix, Vazquez, & Walter (2009). An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*.

Wang, Sun, & Grosse (2021). Beyond marginal uncertainty: how accurately can Bayesian regression models estimate posterior predictive correlations? *International Conference on Artificial Intelligence and Statistics*.

Yu, Bi, & Tresp (2006). Active learning via transductive experimental design. *International Conference on Machine Learning*.

Zhang, Sun, Duvenaud, & Grosse (2018). Noisy natural gradient as variational inference. *International Conference on Machine Learning*.

Zhao, Dougherty, Yoon, Alexander, & Qian (2021a). Bayesian active learning by soft mean objective cost of uncertainty. *International Conference on Artificial Intelligence and Statistics*.

Zhao, Dougherty, Yoon, Alexander, & Qian (2021b). Efficient active learning for Gaussian process classification by error reduction. *Conference on Neural Information Processing Systems*.

Zhao, Dougherty, Yoon, Alexander, & Qian (2021c). Uncertainty-aware active learning for optimal Bayesian classifier. *International Conference on Learning Representations*.

Zhu, Lafferty, & Ghahramani (2003). Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. *International Conference on Machine Learning*.

## A Proof for Example 1

Let $\mathbf{X} = (M, 2M, \dots, M^2)$ denote a collection of inputs, $\mathbf{y} = (y_1, y_2, \dots, y_M)$ denote their labels, and $\mathbf{f} = (\theta(M), \theta(2M), \dots, \theta(M^2))$ denote the values of the Gaussian process at $\mathbf{X}$. The conditional distribution of $\mathbf{y}$ given $\mathbf{f}$ and the marginal distribution of $\mathbf{y}$ are both multivariate Gaussian. Their covariance matrices are

$$\mathrm{Cov}(\mathbf{y}|\mathbf{f}, \mathbf{X}) = I_M$$

$$\mathrm{Cov}(\mathbf{y}|\mathbf{X}) = \begin{pmatrix} 2 & e^{-M^2} & e^{-4M^2} & \dots \\ e^{-M^2} & 2 & e^{-M^2} & \dots \\ e^{-4M^2} & e^{-M^2} & 2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

We now use two facts: that $\mathrm{BALD}(\mathbf{X}) = \mathbb{E}_{\mathbf{f}}[\mathrm{H}[\mathbf{y}] - \mathrm{H}[\mathbf{y}|\mathbf{f}]]$; and that for a multivariate Gaussian random variable, $Z \sim \mathcal{N}(\mu, \Sigma)$, the entropy is $\mathrm{H}[Z] = \frac{1}{2} \log \det 2\pi e \Sigma$ (Cover & Thomas, 2005). Combining these with the simple form of $\mathrm{Cov}(\mathbf{y}|\mathbf{f}, \mathbf{X})$, in particular its independence from $\mathbf{f}$, gives

$$\mathrm{BALD}(\mathbf{X}) = \frac{1}{2} \log \begin{vmatrix} 2 & e^{-M^2} & e^{-4M^2} & \dots \\ e^{-M^2} & 2 & e^{-M^2} & \dots \\ e^{-4M^2} & e^{-M^2} & 2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{vmatrix} := \frac{1}{2} \log \det \Omega_M. \tag{8}$$

To control this determinant we apply the Gershgorin circle theorem (Horn & Johnson, 2012), which states that the eigenvalues of $\Omega_M$ in Equation 8 lie within the interval $[2 - \varepsilon_M, 2 + \varepsilon_M]$, where

$$\varepsilon_M = \sum_{i \neq j}^{M} e^{-M^2 |i-j|^2} \leq M^2 e^{-M^2} \to 0 \text{ as } M \to \infty.$$

We therefore have

$$(1 - \varepsilon_M/2)^M \leq \frac{\det \Omega_M}{2^M} \leq (1 + \varepsilon_M/2)^M.$$

Taking the limit as $M \to \infty$ we have

$$\log\left((1 - \varepsilon_M/2)^M\right) = M \log(1 - \varepsilon_M/2) = -\frac{1}{2} M^3 e^{-M^2} + O\left(M^6 e^{-2M^2}\right) \to 0 \text{ as } M \to \infty,$$

with a similar result for $(1 + \varepsilon_M/2)$. From this we deduce

$$\mathrm{BALD}(\mathbf{X}) = \frac{1}{2} \log \det \Omega_M \to \frac{1}{2} \log 2^M \to \infty \text{ as } M \to \infty.$$

Next we turn to $\mathrm{EIG}_{\theta(x_*)}(\mathbf{X})$. We have the covariance matrix

$$\mathrm{Cov}(\theta(x_*), \mathbf{y}|\mathbf{X}) = \begin{pmatrix} 1 & e^{-|x_* - M|^2} & e^{-|x_* - 2M|^2} & \dots \\ e^{-|x_* - M|^2} & 2 & e^{-M^2} & \dots \\ e^{-|x_* - 2M|^2} & e^{-M^2} & 2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Now consider a set of random variables, $\theta(x_*)'$ and $\mathbf{y}'$, that have the same marginal distributions as $\theta(x_*)$ and $\mathbf{y}$ respectively but are independent of each other. Thus $\theta(x_*)'$ and $\mathbf{y}'$ are jointly Gaussian with covariance matrix

$$\mathrm{Cov}(\theta(x_*)', \mathbf{y}'|\mathbf{X}) = \begin{pmatrix} 1 & 0 & 0 & \dots \\ 0 & 2 & e^{-M^2} & \dots \\ 0 & e^{-M^2} & 2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Using the fact that $\mathrm{EIG}_{\theta(x_*)}(\mathbf{X}) = \mathrm{I}(\theta(x_*); \mathbf{y}|\mathbf{X}) = \mathrm{H}[(\theta(x_*)', \mathbf{y}')] - \mathrm{H}[(\theta(x_*), \mathbf{y})]$, along with the formula used above for the entropy of a multivariate Gaussian random variable, we can write

$$\mathrm{EIG}_{\theta(x_*)}(\mathbf{X}) = \frac{1}{2}\log\det\mathrm{Cov}(\theta(x_*)', \mathbf{y}'|\mathbf{X}) - \frac{1}{2}\log\det\mathrm{Cov}(\theta(x_*), \mathbf{y}|\mathbf{X}).$$

Noting that you can remove a factor from any row of a matrix as a prefactor on the determinant, for both matrices we remove the factors of 2 from each row except the first:

$$\begin{aligned}\mathrm{EIG}_{\theta(x_*)}(\mathbf{X}) =& \frac{1}{2}\log 2^{M-1}\begin{vmatrix} 1 & 0 & 0 & \dots \\ 0 & 1 & e^{-M^2}/2 & \dots \\ 0 & e^{-M^2}/2 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{vmatrix} \\ & - \frac{1}{2}\log 2^{M-1}\begin{vmatrix} 1 & e^{-|x_*-M|^2} & e^{-|x_*-2M|^2} & \dots \\ e^{-|x_*-M|^2}/2 & 1 & e^{-M^2}/2 & \dots \\ e^{-|x_*-2M|^2}/2 & e^{-M^2}/2 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{vmatrix}.\end{aligned}$$

As the $\log 2^{M-1}$ terms then cancel out,

$$\mathrm{EIG}_{\theta(x_*)}(\mathbf{X}) = \frac{1}{2}\log\begin{vmatrix} 1 & 0 & 0 & \dots \\ 0 & 1 & e^{-M^2}/2 & \dots \\ 0 & e^{-M^2}/2 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{vmatrix} - \frac{1}{2}\log\begin{vmatrix} 1 & e^{-|x_*-M|^2} & e^{-|x_*-2M|^2} & \dots \\ e^{-|x_*-M|^2}/2 & 1 & e^{-M^2}/2 & \dots \\ e^{-|x_*-2M|^2}/2 & e^{-M^2}/2 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{vmatrix}. \tag{9}$$

Finally we apply the Gershgorin circle theorem again, along with the fact that $|x_* - iM| \geq |M-1|$ for $i = 1, 2, \dots$. We conclude that the eigenvalues of both matrices in Equation 9 lie within the interval $[1-\varepsilon_M, 1+\varepsilon_M]$, where

$$\varepsilon_M = \sum_{i\neq j}^{M} e^{-(x_*-|i-j|M)^2} \leq M^2 e^{-|M-1|^2} \to 0 \text{ as } M \to \infty.$$

Therefore both determinants are bounded below by $(1-\varepsilon_M)^M$ and above by $(1+\varepsilon_M)^M$. Taking the limit as $M \to \infty$ we have

$$\log\left((1-\varepsilon_M)^M\right) = M\log(1-\varepsilon_M) = -M^3 e^{-|M-1|^2} + O\left(M^6 e^{-2|M-1|^2}\right) \to 0 \text{ as } M \to \infty,$$

with a similar result for $(1+\varepsilon_M)$. Thus both determinants in Equation 9 converge to 1 as $M \to \infty$. From this we conclude

$$\mathrm{EIG}_{\theta(x_*)}(\mathbf{X}) \to 0 \text{ as } M \to \infty.$$

## B BALD derivation

The information gain in $\theta$ due to $(x, y)$ is the reduction in Shannon entropy in $\theta$ that results from observing $(x, y)$:

$$\mathrm{IG}_\theta(x, y) = \mathrm{H}[p_\phi(\theta)] - \mathrm{H}[p_\phi(\theta|x, y)]\,,$$

where $p_\phi(\theta|x, y) \propto p_\phi(y|x, \theta)p_\phi(\theta)$ is the posterior after updating on $(x, y)$.

Since $y$ is a random variable, we compute the expected information gain in $\theta$, known as BALD. To do this we use the model's marginal predictive distribution, $p_\phi(y|x) = \mathbb{E}_{p_\phi(\theta)}[p_\phi(y|x,\theta)]$, to simulate the labels we might observe:

$$\begin{aligned}
\mathrm{BALD}(x) &= \mathbb{E}_{p_\phi(y|x)}[\mathrm{IG}_\theta(x,y)] \\
&= \mathbb{E}_{p_\phi(y|x)}[\mathrm{H}[p_\phi(\theta)] - \mathrm{H}[p_\phi(\theta|x,y)]] \\
&= \mathbb{E}_{p_\phi(y|x)}\left[-\mathbb{E}_{p_\phi(\theta)}[\log p_\phi(\theta)] + \mathbb{E}_{p_\phi(\theta|x,y)}[\log p_\phi(\theta|x,y)]\right] \\
&= \mathbb{E}_{p_\phi(\theta)p_\phi(y|x,\theta)}\left[\log \frac{p_\phi(\theta|x,y)}{p_\phi(\theta)}\right] \\
&= \mathbb{E}_{p_\phi(\theta)p_\phi(y|x,\theta)}\left[\log \frac{p_\phi(y|x,\theta)}{p_\phi(y|x)}\right] \\
&= \mathbb{E}_{p_\phi(\theta)p_\phi(y|x,\theta)}[-\log p_\phi(y|x) + \log p_\phi(y|x,\theta)] \\
&= \mathbb{E}_{p_\phi(\theta)}\left[-\mathbb{E}_{p_\phi(y|x)}[\log p_\phi(y|x)] + \mathbb{E}_{p_\phi(y|x,\theta)}[\log p_\phi(y|x,\theta)]\right] \\
&= \mathbb{E}_{p_\phi(\theta)}[\mathrm{H}[p_\phi(y|x)] - \mathrm{H}[p_\phi(y|x,\theta)]]\,. \qquad (10)
\end{aligned}$$

# C BALD estimation

In general we can estimate BALD using nested Monte Carlo (Rainforth et al, 2018):

$$\begin{aligned}
\mathrm{BALD}(x) &= \mathbb{E}_{p_\phi(\theta)}\left[-\mathbb{E}_{p_\phi(y|x)}[\log p_\phi(y|x)] + \mathbb{E}_{p_\phi(y|x,\theta)}[\log p_\phi(y|x,\theta)]\right] \\
&\approx \frac{1}{M}\sum_{j=1}^{M} -\log\left(\frac{1}{K}\sum_{i=1}^{K} p_\phi(y_j|x,\theta_i)\right) + \log p_\phi(y_j|x,\theta_j),
\end{aligned}$$

where $\theta_i \sim p_\phi(\theta)$, $(\theta_j, y_j) \sim p_\phi(\theta)p_\phi(y|x,\theta)$. Special cases allow us to use computationally cheaper estimators.

## C.1 Categorical predictive distribution

When $y$ and $y_*$ are discrete we can write

$$\begin{aligned}
\mathrm{BALD}(x) &= \mathbb{E}_{p_\phi(\theta)}\left[-\mathbb{E}_{p_\phi(y|x)}[\log p_\phi(y|x)] + \mathbb{E}_{p_\phi(y|x,\theta)}[\log p_\phi(y|x,\theta)]\right] \\
&= -\mathbb{E}_{p_\phi(y|x)}[\log p_\phi(y|x)] + \mathbb{E}_{p_\phi(\theta)p_\phi(y|x,\theta)}[\log p_\phi(y|x,\theta)] \\
&= -\sum_{y\in\mathcal{Y}} p_\phi(y|x)\log p_\phi(y|x) + \mathbb{E}_{p_\phi(\theta)}\left[\sum_{y\in\mathcal{Y}} p_\phi(y|x,\theta)\log p_\phi(y|x,\theta)\right].
\end{aligned}$$

This can be estimated using samples, $\theta_i \sim p_\phi(\theta)$ (Houlsby, 2014):

$$\mathrm{BALD}(x) \approx -\sum_{y\in\mathcal{Y}} \hat{p}_\phi(y|x)\log \hat{p}_\phi(y|x) + \frac{1}{K}\sum_{i=1}^{K}\sum_{y\in\mathcal{Y}} p_\phi(y|x,\theta_i)\log p_\phi(y|x,\theta_i), \qquad (11)$$

where

$$\hat{p}_\phi(y|x) = \frac{1}{K}\sum_{i=1}^{K} p_\phi(y|x,\theta_i).$$

## C.2 Gaussian predictive distribution

Suppose we have a model whose likelihood function, $p_\phi(y|x,\theta)$, and predictive distribution, $p_\phi(y|x)$, are Gaussian. Then, using Equation 10 along with knowledge of the entropy of a Gaussian (Cover & Thomas, 2005), we have

$$\mathrm{BALD}(x) = \frac{1}{2}\log 2\pi e\mathbb{V}[p_\phi(y|x)] - \frac{1}{2}\log 2\pi e\mathbb{V}[p_\phi(y|x,\theta)] = \frac{1}{2}\left(\log \mathbb{V}[p_\phi(y|x)] - \log \mathbb{V}[p_\phi(y|x,\theta)]\right).$$

Relatedly Houlsby et al (2011) identified a closed-form approximation of BALD for the particular case of using a probit likelihood function, a Gaussian-process prior and a Gaussian approximation to the predictive distribution.

# D EPIG derivation

The information gain in $y_*$ due to $(x, y)$ is the reduction in Shannon entropy in $y_*$ that results from observing $(x, y)$:

$$\mathrm{IG}_{y_*}(x, y, x_*) = \mathrm{H}[p_\phi(y_*|x_*)] - \mathrm{H}[p_\phi(y_*|x_*, x, y)]\,,$$

where $p_\phi(y_*|x_*, x, y) = \mathbb{E}_{p_\phi(\theta|x,y)}[p_\phi(y_*|x_*, \theta)]$.

Computing an expectation over both $y$ and $x_*$ gives the expected predictive information gain:

$$\begin{aligned}
\mathrm{EPIG}(x) &= \mathbb{E}_{p_*(x_*)p_\phi(y|x)}[\mathrm{IG}_{y_*}(x, y, x_*)] \\
&= \mathbb{E}_{p_*(x_*)p_\phi(y|x)}[\mathrm{H}[p_\phi(y_*|x_*)] - \mathrm{H}[p_\phi(y_*|x, y, x_*)]] \\
&= \mathbb{E}_{p_*(x_*)p_\phi(y|x)}\left[-\mathbb{E}_{p_\phi(y_*|x_*)}[\log p_\phi(y_*|x_*)] + \mathbb{E}_{p_\phi(y_*|x,y,x_*)}[\log p_\phi(y_*|x, y, x_*)]\right] \\
&= \mathbb{E}_{p_*(x_*)p_\phi(y,y_*|x,x_*)}\left[\log \frac{p_\phi(y_*|x, y, x_*)}{p_\phi(y_*|x_*)}\right] \\
&= \mathbb{E}_{p_*(x_*)p_\phi(y,y_*|x,x_*)}\left[\log \frac{p_\phi(y|x)p_\phi(y_*|x, y, x_*)}{p_\phi(y|x)p_\phi(y_*|x_*)}\right] \\
&= \mathbb{E}_{p_*(x_*)p_\phi(y,y_*|x,x_*)}\left[\log \frac{p_\phi(y, y_*|x, x_*)}{p_\phi(y|x)p_\phi(y_*|x_*)}\right] \\
&= \mathbb{E}_{p_*(x_*)}[\mathrm{I}(y; y_*|x, x_*)] \\
&= \mathbb{E}_{p_*(x_*)}[\mathrm{KL}[p_\phi(y, y_*|x, x_*) \,\|\, p_\phi(y|x)p_\phi(y_*|x_*)]]\,.
\end{aligned}$$

# E EPIG estimation

While in general we can use Equation 7 to estimate EPIG, special cases allow computationally cheaper estimators.

## E.1 Categorical predictive distribution

When $y$ and $y_*$ are discrete we can write

$$\begin{aligned}
\mathrm{EPIG}(x) &= \mathbb{E}_{p_*(x_*)}[\mathrm{KL}[p_\phi(y, y_*|x, x_*) \,\|\, p_\phi(y|x)p_\phi(y_*|x_*)]] \\
&= \mathbb{E}_{p_*(x_*)}\left[\sum_{y\in\mathcal{Y}}\sum_{y_*\in\mathcal{Y}} p_\phi(y, y_*|x, x_*) \log \frac{p_\phi(y, y_*|x, x_*)}{p_\phi(y|x)p_\phi(y_*|x_*)}\right].
\end{aligned}$$

This can be estimated using samples, $\theta_i \sim p_\phi(\theta)$ and $x_*^j \sim p_*(x_*)$:

$$\mathrm{EPIG}(x) \approx \frac{1}{M}\sum_{j=1}^{M}\sum_{y\in\mathcal{Y}}\sum_{y_*\in\mathcal{Y}} \hat{p}_\phi(y, y_*|x, x_*^j) \log \frac{\hat{p}_\phi(y, y_*|x, x_*^j)}{\hat{p}_\phi(y|x)\hat{p}_\phi(y_*|x_*^j)},$$

where

$$\begin{aligned}
\hat{p}_\phi(y, y_*|x, x_*^j) &= \frac{1}{K}\sum_{i=1}^{K} p_\phi(y|x, \theta_i)p_\phi(y_*|x_*^j, \theta_i) \\
\hat{p}_\phi(y|x) &= \frac{1}{K}\sum_{i=1}^{K} p_\phi(y|x, \theta_i) \\
\hat{p}_\phi(y_*|x_*^j) &= \frac{1}{K}\sum_{i=1}^{K} p_\phi(y_*|x_*^j, \theta_i).
\end{aligned}$$

## E.2 Gaussian predictive distribution

Consider a joint predictive distribution that is multivariate Gaussian with mean vector $\mu$ and covariance matrix $\Sigma$:

$$p_\phi(y, y_*|x, x_*) = \mathcal{N}(\mu, \Sigma) = \mathcal{N}\left(\mu, \begin{bmatrix} \mathrm{cov}(x, x) & \mathrm{cov}(x, x_*) \\ \mathrm{cov}(x, x_*) & \mathrm{cov}(x_*, x_*) \end{bmatrix}\right).$$

In this setting the mutual information between $y$ and $y_*$ given $x$ and $x_*$ is a closed-form function of $\Sigma$:

$$\begin{aligned}
\mathrm{I}(y; y_*|x, x_*) &= \mathrm{H}[p_\phi(y|x)] + \mathrm{H}[p_\phi(y_*|x_*)] - \mathrm{H}[p_\phi(y, y_*|x, x_*)] \\
&= \frac{1}{2}\log 2\pi e\mathbb{V}[p_\phi(y|x)] + \frac{1}{2}\log 2\pi e\mathbb{V}[p_\phi(y_*|x_*)] - \frac{1}{2}\log\det 2\pi e\Sigma \\
&= \frac{1}{2}\log\frac{\mathbb{V}[p_\phi(y|x)]\mathbb{V}[p_\phi(y_*|x_*)]}{\det\Sigma} \\
&= \frac{1}{2}\log\frac{\mathrm{cov}(x,x)\mathrm{cov}(x_*,x_*)}{\det\Sigma} \\
&= \frac{1}{2}\log\frac{\mathrm{cov}(x,x)\mathrm{cov}(x_*,x_*)}{\mathrm{cov}(x,x)\mathrm{cov}(x_*,x_*) - \mathrm{cov}(x,x_*)^2}.
\end{aligned}$$

We can estimate EPIG using samples, $x_*^j \sim p_*(x_*)$:

$$\mathrm{EPIG}(x) = \mathbb{E}_{p_*(x_*)}[\mathrm{I}(y; y_*|x, x_*)] \approx \frac{1}{M}\sum_{j=1}^{M}\mathrm{I}\big(y; y_*|x, x_*^j\big) = \frac{1}{2M}\sum_{j=1}^{M}\log\frac{\mathrm{cov}(x,x)\mathrm{cov}(x_*^j,x_*^j)}{\mathrm{cov}(x,x)\mathrm{cov}(x_*^j,x_*^j) - \mathrm{cov}(x,x_*^j)^2}.$$

### E.3 Connection with Foster et al (2019)

Foster et al (2019) primarily considered variational estimation of the expected information gain. Since the joint density, $p_\phi(y, y_*|x, x_*)$, that appears in EPIG is often not known in closed form, EPIG estimation broadly falls under the "implicit likelihood" category of methods considered in that paper. Here we focus on showing how the "posterior" or Barber-Agakov bound (Barber & Agakov, 2003) from this earlier work applies to EPIG estimation. We first recall Equation 5,

$$\mathrm{EPIG}(x) = \mathbb{E}_{p_*(x_*)p_\phi(y,y_*|x,x_*)}[\log p_\phi(y_*|x_*, x, y)] + \mathrm{H}[p_\phi(y_*|x_*)]\,,$$

and the observation that $c = \mathrm{H}[p_\phi(y_*|x_*)]$ does not depend upon $x$ and hence can be neglected when choosing between designs. By Gibbs's inequality, we must have

$$\mathrm{EPIG}(x) \geq \mathbb{E}_{p_*(x_*)p_\phi(y,y_*|x,x_*)}[\log q(y_*|x_*, x, y)] + \mathrm{H}[p_\phi(y_*|x_*)]$$

for any distribution $q$. We can now consider a variational family, $q_\psi(y_*|x_*, x, y)$, and a maximisation over the variational parameter, $\psi$:

$$\mathrm{EPIG}(x) \geq \sup_\psi \mathbb{E}_{p_*(x_*)p_\phi(y,y_*|x,x_*)}[\log q_\psi(y_*|x_*, x, y)] + \mathrm{H}[p_\phi(y_*|x_*)]\,.$$

A practical implication of this bound is that we could estimate EPIG by learning an auxiliary network, $q_\psi(y_*|x_*, x, y)$, using data simulated from the model to make one-step-ahead predictions. That is, $q_\psi$ is trained to make predictions at $x_*$, incorporating the knowledge of the hypothetical acquisition $(x, y)$. For our purposes, training such an auxiliary network at each acquisition is prohibitively expensive. But this approach might be valuable in other applications of EPIG.

## F Dataset construction

### F.1 UCI data

For each dataset we start by taking the base dataset, $\mathcal{D}_{\mathrm{base}}$, from the UCI repository. Satellite and Vowels have predefined test datasets, $\mathcal{D}_{\mathrm{test}}$. In contrast, Magic does not have a predefined train-test split. It is stated in Magic's documentation that one of the classes is underrepresented in the dataset relative to real-world data (Magic is a simulated dataset). Whereas classes 0 and 1 respectively constitute 65% and 35% of the dataset, it is stated that class 1 constitutes the majority of cases in reality (the exact split is not stated; we assume 75% for class 1). We therefore uniformly sample 30% of $\mathcal{D}_{\mathrm{base}}$ to form a test base dataset, $\mathcal{D}'_{\mathrm{base}}$; then we set $\mathcal{D}_{\mathrm{base}} \leftarrow \mathcal{D}_{\mathrm{base}} \setminus \mathcal{D}'_{\mathrm{base}}$; then we make $\mathcal{D}_{\mathrm{test}}$ by removing input-label pairs from $\mathcal{D}'_{\mathrm{base}}$ such that class 1 constitutes 75% of the subset. With the test set defined, we proceed to sample two disjoint subsets of $\mathcal{D}_{\mathrm{base}}$ such that their class proportions match those of $\mathcal{D}_{\mathrm{base}}$: a pool set, $\mathcal{D}_{\mathrm{pool}}$, whose size varies between datasets, and a validation set, $\mathcal{D}_{\mathrm{val}}$, of 60 input-label pairs. Regardless of the class proportions of $\mathcal{D}_{\mathrm{base}}$, we always use an initial training dataset, $\mathcal{D}_{\mathrm{init}}$, of 2 input-label pairs per class, sampled from $\mathcal{D}_{\mathrm{base}}$. Finally we sample a representative set of inputs, $\mathcal{D}_*$, whose class proportions match those of $\mathcal{D}_{\mathrm{test}}$.

### F.2 MNIST data

Implementing each setting starts by using the standard MNIST training data (60,000 input-label pairs) as the base dataset, $\mathcal{D}_{\text{base}}$, and the standard MNIST testing data (10,000 input-label pairs) as the test base dataset, $\mathcal{D}'_{\text{base}}$. For Redundant MNIST we make $\mathcal{D}_{\text{test}}$ by removing input-label pairs from $\mathcal{D}'_{\text{base}}$ such that only classes 1 and 7 remain. Otherwise we set $\mathcal{D}_{\text{test}} = \mathcal{D}'_{\text{base}}$. Next we construct the pool set, $\mathcal{D}_{\text{pool}}$. For Curated MNIST and Redundant MNIST we sample 4,000 inputs per class from $\mathcal{D}'_{\text{base}}$. For Unbalanced MNIST we sample 400 inputs per class for classes 0-4 and 4,000 inputs per class for classes 5-9. After this we make the initial training dataset, $\mathcal{D}_{\text{init}}$. For Curated MNIST and Unbalanced MNIST we sample 2 input-label pairs per class from $\mathcal{D}_{\text{base}}$. For Redundant MNIST we sample 2 input-label pairs from class 1, 2 input-label pairs from class 7 and 1 input-label pair per class from 2 randomly selected classes other than 1 and 7. Next, the validation set, $\mathcal{D}_{\text{val}}$. For all settings this comprises 60 input-label pairs such that the class proportions match those used to form $\mathcal{D}_{\text{pool}}$. Finally we sample a representative set of inputs, $\mathcal{D}_*$, whose class proportions match those of $\mathcal{D}_{\text{test}}$.

## G Extra results

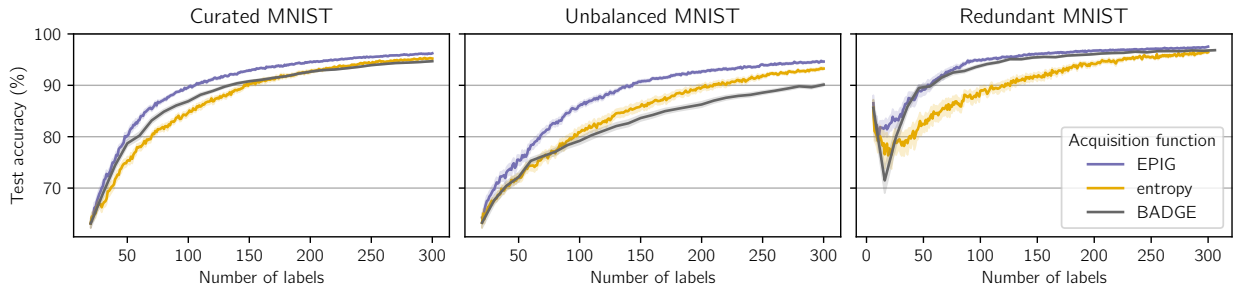


**Figure 8** EPIG outperforms two acquisition functions popularly used as baselines in the active-learning literature. The first is the model's predictive entropy, $\mathrm{H}[p_\phi(y|x)]$ (Settles & Craven, 2008). The second is BADGE (Ash et al, 2020). Calculating BADGE involves computing a gradient-based embedding for each candidate input in the pool and then applying $k$-means++ initialisation (Arthur & Vassilvitskii, 2007) in embedding space to select a diverse batch of inputs for labelling. We acquire 10 labels at a time with BADGE.