Making Better Use of Unlabelled Data in Bayesian Active Learning

Freddie Bickford Smith
University of Oxford

Adam Foster
Microsoft Research

Tom Rainforth
University of Oxford

Abstract

Fully supervised models are predominant in Bayesian active learning. We argue that their neglect of the information present in unlabelled data harms not just predictive performance but also decisions about what data to acquire. Our proposed solution is a simple framework for semi-supervised Bayesian active learning. We find it produces better-performing models than either conventional Bayesian active learning or semi-supervised learning with randomly acquired data. It is also easier to scale up than the conventional approach. As well as supporting a shift towards semi-supervised models, our findings highlight the importance of studying models and acquisition methods in conjunction.

1 Introduction

Bayesian active learning (Gal et al, 2017; Houlsby et al, 2011; Kirsch, 2023a; MacKay, 1992a,b) involves seeking the most informative labels for training a given model. The model is the basis not only for prediction but also for deciding what data to acquire, with the latter requiring a notion of how the model's uncertainty will change after updating on new data. Choosing the right model is therefore critical to success.

We make a case against the fully supervised models conventionally used in Bayesian active learning (Atighehchian et al, 2020; Beluch et al, 2018; Bickford Smith et al, 2023; Chitta et al, 2018; Gal et al, 2017; Houlsby et al, 2011; Jeon, 2020; Kirsch, 2023a,b,c; Kirsch et al, 2023, 2019; Murray et al, 2021; Pinsler et al, 2019; Pop & Fulop, 2018; Tran et al, 2019). Crucially these models do not account for all the information at hand: they neglect the rich information often conveyed by unlabelled data, which is usually assumed to be cheaply available (Lewis & Gale, 1994; Settles, 2012). This neglected information can be critical to both making predictions and estimating the reducible uncertainty present in the model. As a result the conventional approach can fail to appropriately target the novel information that new labels can provide.

This problem is exacerbated by the common practice of using big models and assuming all the model parameters will be updated after each data-acquisition step. The consequences of this include significant redundancy in parameter uncertainty and inconsistencies in reducible-uncertainty estimation. While the former might be addressed by focusing on predictions instead of parameters (Bickford Smith et al, 2023), the latter remains a thorny problem. The (implicit) priors used in practice can poorly reflect our beliefs about the reducible uncertainty present in the model, and it is usually not possible to accurately approximate the Bayesian updates that formal notions of reducible uncertainty assume (Sharma et al, 2023). Difficulties like these can pose a barrier to capturing how a model's uncertainty will change as new data is acquired.

Targeting updates in all the model parameters can additionally be computationally demanding. Even when resorting to crude inference approximations or heuristics, updating all the model parameters after each data-acquisition step tends to be costly, and estimating the model's reducible uncertainty can also be challenging. These high costs limit the applicability of conventional Bayesian active learning and often necessitate additional approximations (Kirsch et al, 2019).

We argue that all these problems can be addressed by rethinking the conventional model setup with an emphasis on the specific needs of Bayesian active learning. In particular we suggest the field should shift towards using semi-supervised models that can capitalise on unlabelled data while also producing useful estimates of their reducible uncertainty.

To this end we propose a simple framework: pretrain a deterministic encoder on unlabelled data, then train a stochastic prediction head using Bayesian active learn-
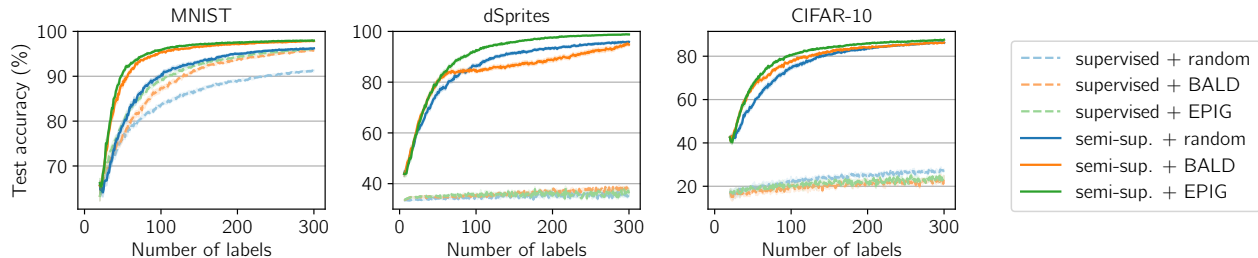
Figure 1 Semi-supervised models produce much better predictions than fully supervised ones. At the same time, using the right acquisition method is critical for effective active learning. BALD, which targets direct reductions in parameter uncertainty, does not consistently outperform random acquisition here; EPIG, a prediction-oriented alternative, does.

ing. The encoder (which we choose to keep fixed) allows us to incorporate relevant information from the unlabelled data into the model, and contains much of the model's processing power. The prediction head is then a more lightweight component that captures label information and that, relative to the conventional model setup, supports more faithful estimation of the model's reducible uncertainty by allowing cheaper and more accurate updates as new data is acquired.

Our approach can produce notably better predictive performance than the conventional alternative of using fully supervised models (see Figure 1). It also significantly reduces the computational cost of training and acquisition. As a result it opens up new practical use cases, including scaling up to high-dimensional input spaces and large pools of unlabelled data.

Given its simplicity and benefits, why is this approach not already the default in Bayesian active learning? One explanation is an implicit assumption in past work that acquisition methods can be studied in isolation from considerations about the model. We suggest this assumption is misguided: accounting for unlabelled data has a fundamental effect on what labels should be acquired. Another explanation is a degree of ambiguity in the literature about the benefit of active learning, Bayesian or otherwise, when using semi-supervised models (see Section 5). We show that using the right acquisition method is key to the success of semi-supervised Bayesian active learning. In particular we find that EPIG (Bickford Smith et al, 2023), a prediction-oriented method, produces more consistent gains over random acquisition than BALD (Houlsby et al, 2011), a parameter-oriented method, does.

Our contribution is thus threefold. First, we explain why the fully supervised models predominantly used in Bayesian active learning can undermine data acquisition. Second, we set out a framework for using semi-supervised models instead, and show empirically that this outperforms training the same models on randomly acquired labels. Third, we provide some clarity

on how the field can move forward: it should embrace semi-supervised models and carefully study the interplay between models and acquisition methods.

2 Background

We focus on learning a probabilistic predictive model, $p_\phi(y|x)$, where x is an input and y is a label. We assume there are some underlying stochastic model parameters, $\theta \sim p_\phi(\theta)$, defined such that

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

Active learning (Atlas et al, 1989; Settles, 2012) is the process of training a model on data selected by an algorithm. In the context of supervised learning the algorithm decides which labels to acquire, commonly over a series of steps. Each step, t, comprises three substeps. First, the algorithm selects a query input, x_t, or a batch of inputs, often by optimising an acquisition function. In pool-based active learning (Lewis & Gale, 1994), the setting we focus on in this work, the set of candidate inputs is a fixed pool, $\mathcal{D}_{\text{pool}} = \{x_i\}_{i=1}^N$. Second, the algorithm obtains a label, y_t, from the true conditional label distribution, $p(y|x = x_t)$, and adds (x_t, y_t) to the training dataset, $\mathcal{D}_{\text{train}}$. Third, the model, $p_\phi(y|x)$, is updated on $\mathcal{D}_{\text{train}}$.

Bayesian experimental design (Chaloner & Verdinelli, 1995; Lindley, 1956; Rainforth et al, 2024) is a rigorous framework for identifying what data we expect to yield the most information gain in a quantity of interest, ψ. Given a joint distribution, $p(y, \psi|x)$, the information gain in ψ due to new data, (x, y), is the reduction in Shannon entropy in ψ that results from updating on (x, y). For a given x the expected information gain over possible realisations of y is

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

where $p_\psi(y|x) = \mathbb{E}_{p(\psi)}[p(y|x,\psi)]$ is the marginal predictive distribution and $p(\psi|x,y) \propto p(y,\psi|x)$ is the

posterior that results from a Bayesian update on (x, y). This quantity can be recognised as $I(\psi; y|x)$, the mutual information between ψ and y given x.

Bayesian active learning (Gal et al, 2017; Houlsby et al, 2011; Kirsch, 2023a; MacKay, 1992a,b) is a form of active learning that uses acquisition functions derived from the framework of Bayesian experimental design. Traditionally it has targeted information gain in the model parameters, θ, by using the BALD score (Houlsby et al, 2011), $\text{BALD}(x) = \text{EIG}_\theta(x)$, as the acquisition function. Recent work (Bickford Smith et al, 2023) noted the typically prediction-oriented nature of machine learning and showed the benefit of instead targeting information gain in y_*, the model's prediction on a random target input, x_*. This yields the expected predictive information gain, $\text{EPIG}(x) =$

$$\mathbb{E}_{p_*(x_*)p_\phi(y|x)}[\text{H}[p_\phi(y_*|x_*)] - \text{H}[p_\phi(y_*|x_*, x, y)]] .$$

This is equivalent to $I((x_*, y_*); y|x)$, the mutual information between (x_*, y_*) and y given x. Whereas BALD targets a global reduction in parameter uncertainty, EPIG targets only reductions in parameter uncertainty that correspond to reduced predictive uncertainty on x_*. As an expected value over target inputs, EPIG explicitly accounts for the input distribution.

Unsupervised pretraining (Bengio et al, 2006; Bommasani et al, 2021; Hinton et al, 2006; Ranzato et al, 2006) is a way to distil information from unlabelled data into a model, with the aim of improving the model's performance on downstream tasks. It is common practice to construct a model composed of two parts: an encoder that maps a given input to a latent representation, and a prediction head that maps the latent representation to a predictive distribution. The encoder is trained on unlabelled data (pretraining), then either the whole model or just the prediction head is trained on labelled data (finetuning). This two-step training scheme produces a semi-supervised model, incorporating both labelled and unlabelled data.

3 Problems with current models

Unlabelled data can provide a significant amount of information helpful for downstream predictive tasks, as demonstrated by cases like few-shot learning in vision models (Chen et al, 2020c) and in-context learning in language models (Brown et al, 2020). A key aspect of this information relates to the underlying input manifold, knowledge of which lets the model make better similarity judgements between inputs and thus produce more appropriate predictive correlations. Such correlations are crucial to downstream uncertainty estimation (Osband et al, 2022a,b; Wang et al, 2021).

The fully supervised models widely used in Bayesian active learning—examples include supervised deep ensembles (Beluch et al, 2018; Chitta et al, 2018) and supervised dropout networks (Atighehchian et al, 2020; Gal et al, 2017; Kirsch, 2023c; Kirsch et al, 2019)—cannot directly capture the information present in unlabelled data, as inputs without labels do not convey information under such models. A consequence of this is well documented in multiple studies of other active-learning approaches (see Section 5): the model suffers a direct reduction in predictive performance.

Another consequence, one specific to Bayesian active learning, has been overlooked in past work: not accounting for unlabelled data undermines our estimation of the model's reducible uncertainty and therefore poses a challenge to acquiring informative labels. One perspective on this is that the model's predictive correlations are much worse than they would be if unlabelled data were incorporated. An alternative (and related) perspective is that failing to incorporate known relevant information into the model from the start leads to reducible uncertainty in the model that need not exist, which in turn leads to spending precious labels gathering the very same information.

The problem here is particularly pertinent when using big models and assuming all the model parameters will be updated after each data-acquisition step. This is partly because big models complicate prior specification. In principle we can incorporate information from unlabelled data into a fully supervised model by analysing the data and encoding our insights into the prior beliefs that we specify (Gelman et al, 2013; Kruschke, 2014). But working with big models and lots of unlabelled data often makes this infeasible (Fortuin, 2021; Tran et al, 2022; Wenzel et al, 2020).

On top of the issue of prior specification is that of non-Bayesian updating on new data. Whenever we use a form of reducible uncertainty, such as BALD or EPIG, we implicitly assume new data will be incorporated using an exact Bayesian update (Rainforth et al, 2024). Yet the high parameter counts seen in practice typically necessitate updating schemes that do not match the Bayesian ideal, including those based on inaccurate inference approximations (Farquhar, 2022; Sharma et al, 2023). This introduces a mismatch between the assumed and actual updates, reducing the correspondence between how informative new data is estimated to be and how informative it actually is. Targeting the best data can thus become challenging.

A further problem is the computational cost of active-learning steps. Updates on new data can be expensive even if they are not Bayesian. Estimating the model's reducible uncertainty for data-acquisition decisions is also potentially costly. While the extra overall com-
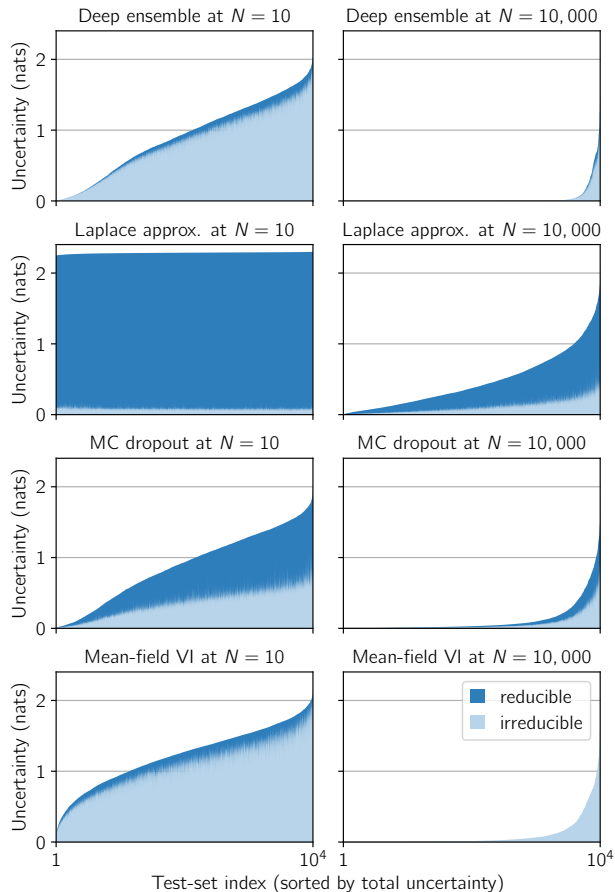
Figure 2 Bayesian deep learning can struggle to appropriately decompose a model's estimated uncertainty. Most of the uncertainty of the model here is ultimately reducible, as demonstrated by the decrease in total uncertainty between training on a small amount of data ($N = 10$) and training on more data ($N = 10,000$). This suggests the irreducible-uncertainty estimates produced by these methods are inconsistent. Using deep ensembles, Monte Carlo dropout and mean-field variational inference, much of the uncertainty deemed to be irreducible at $N = 10$ is resolved at $N = 10,000$. Using Laplace approximation, the irreducible uncertainty increases on many of the test inputs after training on more data. See Appendix B for details.

putational cost is often justified if it allows us to more wisely allocate our labelling budget, it might be too great for many practical applications, especially when performing batch acquisition (Kirsch et al, 2019).

In Figure 2 we demonstrate the kind of problem that can occur with commonly used models and inference approximations. Given this relatively simple MNIST classification problem (LeCun et al, 1998), we can reasonably believe that with enough training data the neural network of interest will be able to make correct, confident predictions on test inputs (Belkin et al, 2018; Zhang et al, 2017). In other words, most of the model's uncertainty can be reduced. Our results reflect this:

across all the methods considered, the model has high total uncertainty on many test inputs after training on a small dataset and then has much lower total uncertainty after training on a bigger one. This reveals the uncertainty estimates to often be overly pessimistic, with a significant fraction of the model's uncertainty deemed to be irreducible. Failing to appropriately decompose the model's uncertainty can have a critical effect on Bayesian active learning: it directly affects which inputs are prioritised for labelling.

4 A semi-supervised approach to Bayesian active learning

We now address how to effectively incorporate unlabelled data into Bayesian active learning. Our proposal is a simple semi-supervised framework: pretrain a deterministic encoder on unlabelled inputs, then train a stochastic prediction head through Bayesian active learning. The idea is that the encoder (which we keep fixed) will capture much of the information present in the unlabelled data, while the prediction head will encapsulate the information we gather from labelled data and help guide acquisition by capturing a notion of the model's reducible uncertainty.

4.1 Semi-supervised model

The basis of our framework is to split the predictive model into an encoder, g, and a prediction head, h:

$$p_\phi(y|x) = \mathbb{E}_{p_\phi(\theta_h)}[p_\phi(y|g(x), \theta_h)]$$

where $\theta_h \sim p_\phi(\theta_h)$ represents the parameters of the prediction head and $p_\phi(y|g(x), \theta_h)$ is the predictive distribution for a given configuration of θ_h.

The encoder is pretrained on unlabelled data and then treated as a fixed function with no stochastic parameters (see Section 4.2). Any pretraining method can be used, with a preference for methods that give strong downstream predictive performance. Because the encoder is deterministic and only needs to be trained once, it can be big and have a lot of processing power while still avoiding the issues discussed in Section 3.

The prediction head, in contrast, is updated as new labels are acquired and conforms to the setup described in Section 2. Typically it can be significantly more lightweight than either the encoder or the fully supervised models conventionally used in Bayesian active learning. This is partly because the amount of unlabelled data typically dwarfs the label budget, and in such settings this model setup supports good predictions (Assran et al, 2022; Chen et al, 2020c; Hénaff et al, 2020) and uncertainty estimation (Daxberger et al, 2021; Kristiadi et al, 2020; Sharma et al, 2023;

Snoek et al, 2015). It is also because using a smaller prediction head allows cheaper and more consistent parameter updates on new data (see Section 4.2).

Importantly the model's reducible uncertainty is now dictated entirely by $p_\phi(\theta_h)$. For example, BALD, which measures the total reducible uncertainty in the model parameters, can be written as $\mathrm{BALD}(x) = \mathrm{I}((\theta_g, \theta_h); y|x) = \mathrm{I}(\theta_h; y|x)$, due to the fact that the encoder parameters, θ_g, are not random.

4.2 Fixing the encoder

A key decision when using a pretrained encoder is whether to keep it fixed or finetune it along with the prediction head. We suggest that fixing the encoder is a sensible default in our framework (though occasionally updating it using semi-supervised representation learning might sometimes be helpful). Our reasons for this draw on insights from Bayesian experimental design along with more practical considerations.

4.2.1 Update consistency

In Section 3 we explained that data-acquisition decisions can be undermined by a mismatch between the exact Bayesian update assumed by formal notions of reducible uncertainty and the actual update made in practice. Finetuning the encoder thus introduces a difficult technical issue: regardless of how we do it, it substantially increases the scope for a mismatch between assumed and actual updates. On the one hand we could use a deterministic encoder and update some or all of its parameters in each active-learning step. Here the update mismatch would be because only distributional updates on the model's stochastic parameters are accounted for when estimating the model's reducible uncertainty. On the other hand we could use a stochastic encoder and update it through approximate inference. Here the update mismatch would be due to inaccuracies in the approximation scheme.

The model setup we propose can help mitigate this issue. It makes consistent updating more viable by focusing updates on the lightweight prediction head.

4.2.2 Computational cost

Bayesian active learning becomes considerably cheaper computationally when we use a powerful fixed encoder. One reason for this is that we only need to make updates to $p_\phi(\theta_h)$ at each active-learning step, with any associated inference much easier for this small subset of the model parameters than for all of them. Another key reason is that we reduce the number of encoder forward passes we have to compute: we only need to pass the inputs through the encoder once at the start

of active learning to precompute $\{g(x_i)\}_{i=1}^N$, whereas in the conventional setup we need to evaluate the full model each time we use $p_\phi(y|x, \theta)$.

This reduction in computational cost can matter a lot in practice. Training a powerful predictive model tends to require a lot of computation. The pretraining step in our framework amortises this computation, shifting it outside the active-learning loop. We can thus get the benefit of active learning—the ability to spend extra computation to deal with high labelling costs—without introducing big delays as the model's parameter distribution updates or as its uncertainty is estimated. This opens up use cases that otherwise might be impractical, such as human-in-the-loop labelling.

4.3 Acquisition function

While it is well established that using semi-supervised models leads to better predictive performance, some past empirical studies have raised doubts about the benefit of active learning in these models (see Section 5). This appears to contradict our argument that incorporating information from unlabelled data should lead to better decisions about what data to acquire. A partial explanation for this apparent contradiction is the data used in these studies. The need for active learning is clearest when there is a big disparity between the most and least relevant inputs to label. Past studies have commonly used carefully curated pools of unlabelled data, which provide an unrepresentative boost to the performance of random acquisition.

An important additional explanation is that previous evaluations of Bayesian active learning have overwhelmingly focused on BALD acquisition (Atighehchian et al, 2020; Beluch et al, 2018; Burkhardt et al, 2018; Chitta et al, 2018; Gal et al, 2017; Houlsby et al, 2011; Jeon, 2020; Kirsch, 2023b,c; Kirsch et al, 2023, 2019; Lee & Kim, 2019; Murray et al, 2021; Osband et al, 2023; Pop & Fulop, 2018; Tran et al, 2019). Recent work showed that BALD can exhibit serious pathologies when our aim is to maximise predictive performance, motivating the use of EPIG as an alternative (Bickford Smith et al, 2023). In our own experiments we often find EPIG to be a key contributor to the success of semi-supervised Bayesian active learning: BALD can perform worse than random acquisition, while EPIG produces a reliable improvement.

Crucial to this success, we believe, is that EPIG directly targets information about downstream predictions. Adapting a result from Bernardo (1979) clarifies the relationship between the two acquisition functions.

Proposition 1 $\mathrm{EPIG}(x) \leq \mathrm{BALD}(x)$, *with equality if y_* is a one-to-one function of θ for all x_*.*

Proof Using the chain rule for mutual information, we can equate two ways of writing $I(y; y_*, \theta | x, x_*)$:

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

Now we note that $I(y; y_* | x, x_*, \theta) = 0$ because the model's predictions are independent if we condition on θ. We also note that $I(y; \theta | x, x_*, y_*) \geq 0$, so

$$I(y; \theta | x, x_*) - I(y; y_* | x, x_*) \geq 0.$$

Taking an expectation over $p(x_*)$ and noting that $I(y; \theta | x, x_*) = I(y; \theta | x) = \text{BALD}(x)$ now gives the desired inequality. This in turn gives an equality if y_* is a one-to-one function of θ for all x_* because that implies $I(y; \theta | x, x_*) = I(y; y_* | x, x_*)$, since the mutual information between two variables is invariant to invertible transformations (Cover & Thomas, 2005). □

With this we can interpret EPIG as a "filtered" version of BALD that only retains information relevant to the model's predictions on the task of interest. As a result it avoids BALD's pathology of acquiring data that is informative with respect to the model's parameters but not its downstream predictions. EPIG's role in our framework can therefore be seen as complementary to that of unsupervised pretraining: pretraining provides breadth, capturing much of the information that might be useful, while EPIG provides the ability to hone in on the task of interest.

5 Related work

Early work on active learning included a demonstration by McCallum & Nigam (1998) of using a semi-supervised model, specifically a naive-Bayes classifier, with query-by-committee data acquisition (Seung et al, 1992). Muslea et al (2002) likewise proposed using a semi-supervised naive-Bayes model along with disagreement-based acquisition. Riccardi & Hakkani-Tür (2003), Tur et al (2005) and Tomanek & Hahn (2009) considered an approach based on model confidence: select low-confidence inputs for labelling; assign pseudolabels (using the model) to high-confidence inputs and include them in the training data. Zhu et al (2003) studied a Gaussian-random-field binary classifer, using an estimate of the expected loss reduction as their acquisition function. Nguyen & Smeulders (2004) used a cluster-based binary classifier and decided which labels to acquire based on candidate inputs' estimated contribution to the model's error.

The recent work most directly related to ours is that which includes empirical evaluations of BALD acquisition in combination with semi-supervised models.

(EPIG has not previously been tested with such models.) The findings of the evaluations have been mixed: some show BALD outperforming random acquisition while others show it performing the same or worse.

Some of the most positive results have been found on language datasets. Burkhardt et al (2018) used a semi-supervised variational autoencoder and BALD acquisition in the context of classifying social-media posts, and demonstrated benefits both from incorporating unlabelled data and from active data acquisition. Seo et al (2022) combined unsupervised pretraining with active learning and found BALD gave consistent gains over random acquisition on relation-extraction and sentence-classification tasks. Osband et al (2023) studied a similar method on a range of text-classification tasks and saw BALD perform strongly.

BALD has performed less convincingly in computer-vision evaluations. Mittal et al (2023) reported positive results on semantic-segmentation datasets. But Sener & Savarese (2018), Hacohen et al (2022), Yehuda et al (2022) and Lüth et al (2023) found BALD sometimes performed better and other times worse than random acquisition on image-classification datasets.

In the broader literature there has also been a degree of ambiguity about the benefit of active learning (relative to training on randomly acquired data) when using semi-supervised models. A number of methods for semi-supervised active learning have been found to beat non-active counterparts in evaluations focusing on computer vision (Bhatnagar et al, 2020; Chu et al, 2022; Ebrahimi et al, 2020; Gao et al, 2020; Gudovskiy et al, 2020; Guo et al, 2016, 2021a,b; Hacohen et al, 2022; Huang et al, 2021; Lüth et al, 2023; Margatina et al, 2022; Mittal et al, 2023; Mohamadi et al, 2022; Mottaghi & Yeung, 2019; Pourahmadi et al, 2021; Rhee et al, 2017; Rottmann et al, 2018; Sener & Savarese, 2018; Shui et al, 2020; Tamkin et al, 2022; Wang et al, 2020; Yang et al, 2015; Yehuda et al, 2022; Yi et al, 2022; Zhang et al, 2024, 2020), natural-language processing (Ein-Dor et al, 2020; Maekawa et al, 2022; Seo et al, 2022; Tamkin et al, 2022; Yuan et al, 2020), molecular-property prediction (Hao et al, 2020; Zhang & Lee, 2019) and speech recognition (Drugman et al, 2016). Methods notable for their simplicity and their strong performance on standard problems include three that use unsupervised pretraining and that base data acquisition on different approaches to covering the input space: ProbCover (Yehuda et al, 2022), TypiClust (Hacohen et al, 2022) and a method based on k-means clustering (Pourahmadi et al, 2021).

Alongside the studies reporting positive results have been several casting doubt on how helpful active learning is when using a semi-supervised model. Bengar
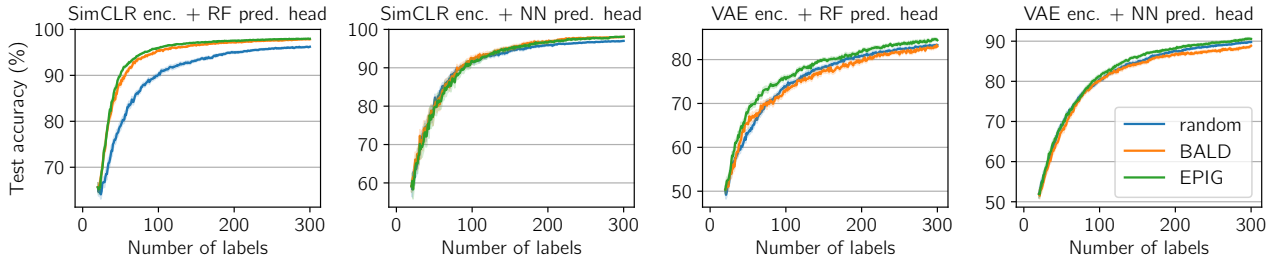
Figure 3 EPIG's boost over random acquisition is relatively robust to the choice of encoder and prediction head in semi-supervised models applied to MNIST data. BALD only convincingly beats random for two out of four configurations.

et al (2021), Chan et al (2021), Mittal et al (2019) and Siméoni et al (2020) all evaluated multiple acquisition methods on multiple image-classification datasets and found no acquisition method reliably outperformed random acquisition once unlabelled data had been incorporated into the model. Likewise none of the acquisition methods tried by Gleave & Irving (2022) improved over random acquisition in the context of text classification using a pretrained model.

6 Experiments

To help resolve some of the ambiguity in the literature and to validate our proposed framework, we evaluated variants of Bayesian active learning on a range of tasks. We focused on image classification, in line with much of the literature on foundational active-learning methodology (see Section 5). Details about the data, models and active-learning setup we used are given in Appendix B. Code for reproducing our results is available at github.com/fbickfordsmith/epig.

6.1 Semi-supervised models outperform fully supervised ones but BALD is unreliable

We first set out to understand the interplay between semi-supervised models and BALD-based active learning, since past work leaves it unclear what to expect from the combination (see Section 5). Using the MNIST (LeCun et al, 1998), dSprites (Matthey et al, 2017) and CIFAR-10 (Krizhevsky, 2009) datasets, we evaluated BALD-based active learning with fully supervised and semi-supervised models.

We find that using a semi-supervised model instead of a fully supervised one leads to better predictive performance, but also that BALD does not reliably help on top of this (see Figure 1). This finding is in line with results from multiple past studies. A plausible explanation is that, despite the model incorporating information from unlabelled data and supporting better uncertainty estimation, the type of uncertainty reduction that BALD targets causes irrelevant data to

be acquired. The results might thus indicate a need for more targeted information-seeking.

6.2 EPIG produces a reliable boost over both BALD and random acquisition

We next investigated whether acquiring data using EPIG addresses the problems associated with using BALD. Returning to the datasets and semi-supervised models from before, we ran active learning with EPIG.

Our results show EPIG convincingly outperforming random acquisition in cases where BALD fails to (see Figure 1). Referring to our discussion in Section 4, we can understand this to be a result of EPIG being more specific in the information it favours, with the pretrained encoder supporting effective judgements of similarity between inputs and thus helping to target information relevant to downstream prediction.

EPIG's advantage over BALD is also clear when we consider their sensitivity to changes in the semi-supervised model. Revisiting MNIST with changes to the encoder and prediction head, we see in Figure 3 that BALD's performance is even shakier than the results in Figure 1 let on: it does not convincingly outperform random acquisition when we switch from the SimCLR encoder to the VAE encoder. EPIG shows greater robustness to changes in the model.

6.3 The semi-supervised approach works with messy data and at scale

We then evaluated our approach in scenarios that capture aspects of working with messy data in practical applications, contrasting with the relatively well-curated and small-scale datasets from before. To this end we first looked at "redundant" variants of MNIST and CIFAR-10, in which the pool of unlabelled data contains many more classes than the ones we want to discriminate between (Bickford Smith et al, 2023).

The strong performance we see on Redundant MNIST and Redundant CIFAR-10 (see Figure 4) has impor-
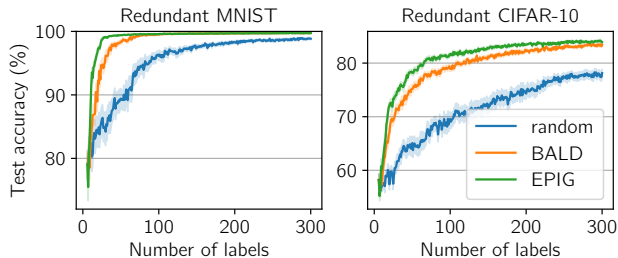
Figure 4 BALD and EPIG provide especially big gains in predictive performance relative to random acquisition when the pool of unlabelled data is not carefully curated.
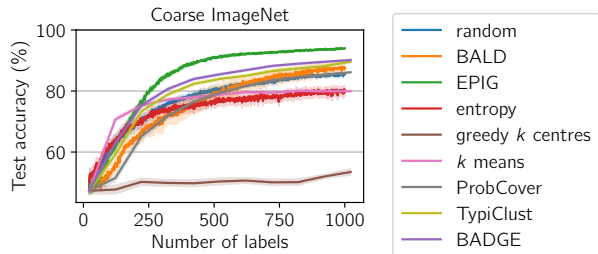
Figure 5 Semi-supervised Bayesian active learning performs strongly on ImageNet data, with EPIG producing a particularly notable boost over random acquisition.

tant practical implications. Modern machine learning often entails using messy pools of unlabelled data—for example, huge collections of audio, images and text (Ardila et al, 2020; Gemmeke et al, 2017; Mahajan et al, 2018; Radford et al, 2021; Raffel et al, 2020; Sun et al, 2017)—where many of the inputs have little relevance to the task we are ultimately interested in. Our framework provides a way to make the most of these resources while overcoming the challenges they pose. Through unsupervised pretraining we can capitalise on the information encoded in unlabelled inputs, and through EPIG-based data acquisition we can avoid the trap of gathering labels for obscure inputs that have limited relevance to the task of interest.

We finally focused on scaling up to higher-dimensional input spaces. In particular we looked at classifying ImageNet images into eleven superclasses based on the WordNet semantic hierarchy (Engstrom et al, 2019; Miller, 1995): bird, boat, car, cat, dog, fruit, fungus, insect, monkey, truck and other. For a better sense of context we evaluated six extra acquisition methods: maximum entropy (Settles & Craven, 2008), BADGE (Ash et al, 2020), greedy k centres (Sener & Savarese, 2018), k means (Pourahmadi et al, 2021), ProbCover (Yehuda et al, 2022) and TypiClust (Hacohen et al, 2022). As noted in Section 5, the last three of these methods have been found to perform well in past work. For example, ProbCover's reported performance on

Dataset	Time (sup.)	Time (semi)	Speedup	Enc. time
MNIST	32 sec	17 sec	2×	2 msec
dSprites	4 min 17 sec	16 sec	16×	6 msec
CIFAR-10	42 min 14 sec	29 sec	89×	75 msec

Table 1 Semi-supervised Bayesian active learning is much faster than the conventional fully supervised approach, with the speedup increasing as we scale to harder datasets and use more powerful encoders. We report the mean duration of an active-learning step with EPIG acquisition, and the speedup from using the semi-supervised approach. To give an indication of encoder power, we report the time taken to encode a batch of 256 inputs (mean across seven runs on an Nvidia GeForce RTX 2080 Ti).

ImageNet data is the best we are aware of.

Figure 5 shows our approach providing outstanding performance here, to our knowledge the strongest yet observed for Bayesian active learning. Even when compared with state-of-the-art baselines, it provides a substantial boost in test accuracy. This demonstrates the possibility of scaling up information-theoretic data acquisition to challenging, high-dimensional inputs. Notably this is possible provided an appropriate model and acquisition strategy are used: as before, our use of EPIG is key in achieving such strong results. Helping to explain these gains is a notable difference in class distributions in the labelled datasets produced by random, BALD and EPIG acquisition (see Figure 6).

6.4 Semi-supervised models allow much faster Bayesian active learning

A key practical finding is how much faster Bayesian active learning becomes when we use the semi-supervised approach (see Table 1). This validates the argument we presented in Section 4.2: unsupervised pretraining gives us a powerful encoder that we can adapt to a task of interest using a lightweight prediction head, keeping the encoder fixed. This setup allows a drastic speedup, which could unlock new practical use cases.

7 Conclusion

We have shown that the fully supervised models predominantly used in Bayesian active learning are problematic. In particular, by not accounting for the information present in unlabelled data, they undermine the pursuit of useful novel information. We have identified a simple semi-supervised framework as an appealing solution. While previous work has seen mixed results on the benefits of active learning with semi-supervised models, we have shown that using EPIG, a prediction-oriented acquisition function, can consistently provide improvements over random acquisition.

Freddie Bickford Smith, Adam Foster, Tom Rainforth
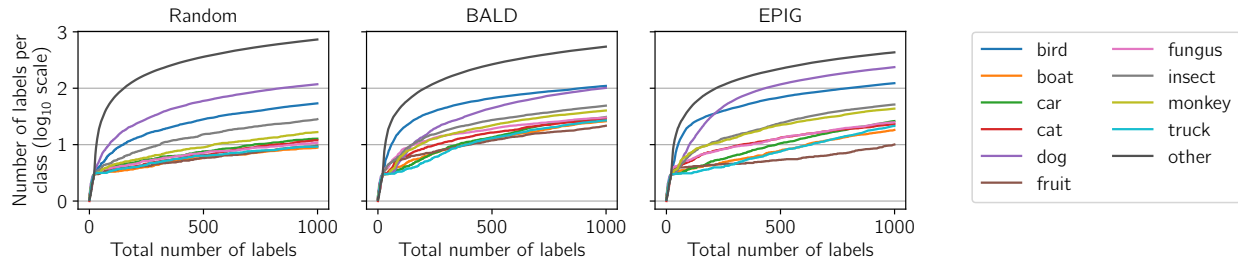
Figure 6 The labelled datasets generated by random, BALD and EPIG acquisition are notably distinct in their class distributions. Random matches the class distribution of the pool, while both BALD and EPIG differentially upweight the non-"other" classes. We report the acquisitions corresponding to Figure 5 (mean label counts across random seeds).

Our findings suggest a crucial update for the field. Work on Bayesian active learning has largely been disconnected from that on semi-supervised learning. This implicitly assumes that results from studying fully supervised models cleanly translate to semi-supervised models. Empirical results from both prior work and our own show this assumption to be misguided. At the same time, obtaining useful methodological insights requires studying the tools practitioners actually use, and few practitioners would neglect to make good use of unlabelled data. Research on Bayesian active learning should embrace a shift to semi-supervised models.

Acknowledgements

References

Aitchison (2021). A statistical theory of cold posteriors in deep neural networks. *International Conference on Learning Representations.*

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

Asano, Rupprecht, Zisserman, & Vedaldi (2021). PASS: an ImageNet replacement for self-supervised pretraining without humans. *Track on Datasets and Benchmarks, Conference on Neural Information Processing Systems.*

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

Assran, Caron, Misra, Bojanowski, Bordes, Vincent, Joulin, Rabbat, & Ballas (2022). Masked Siamese networks for label-efficient learning. *European Conference on Computer Vision.*

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.*

Belkin, Hsu, & Mitra (2018). Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate. *Conference on Neural Information Processing Systems.*

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

Bengar, van de Weijer, Twardowski, & Raducanu (2021). Reducing label effort: self-supervised meets active learning. *International Conference on Computer Vision Workshops.*

Bengio, Lamblin, Popovici, & Larochelle (2006). Greedy layer-wise training of deep networks. *Conference on Neural Information Processing Systems.*

Bernardo (1979). Expected information as expected utility. *Annals of Statistics.*

Bhatnagar, Tank, Goyal, & Sethi (2020). PAL: pretext-based active learning. *arXiv.*

Bickford Smith, Kirsch, Farquhar, Gal, Foster, & Rainforth (2023). Prediction-oriented Bayesian active learning. *International Conference on Artificial Intelligence and Statistics.*

Bommasani, Hudson, Adeli, Altman, Arora, von Arx, Bernstein, Bohg, Bosselut, Brunskill, Brynjolfsson, Buch, Card, Castellon, Chatterji, Chen, Creel, Davis, Demszky, Donahue, Doumbouya, Durmus, Ermon, Etchemendy, Ethayarajh, Fei-Fei, Finn, Gale, Gillespie, Goel, Goodman, Grossman, Guha, Hashimoto, Henderson, Hewitt, Ho, Hong, Hsu, Huang, Icard, Jain, Jurafsky, Kalluri, Karamcheti, Keeling, Khani, Khattab, Koh, Krass, Krishna, Kuditipudi, Kumar, Ladhak, Lee, Lee, Leskovec, Levent, Li, Li, Ma, Malik, Manning, Mirchandani, Mitchell, Munyikwa, Nair, Narayan, Narayanan, Newman, Nie, Niebles, Nilforoshan, Nyarko, Ogut, Orr, Papadimitriou, Park, Piech, Portelance, Potts, Raghunathan, Reich, Ren, Rong, Roohani, Ruiz, Ryan, Ré, Sadigh, Sagawa, Santhanam, Shih, Srinivasan, Tamkin, Taori, Thomas, Tramèr, Wang, Wang, Wu, Wu, Wu, Xie, Yasunaga, You, Zaharia, Zhang, Zhang, Zhang, Zhang, Zheng, Zhou, & Liang (2021). On the opportunities and risks of foundation models. *arXiv*.

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

Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, Askell, Agarwal, Herbert-Voss, Krueger, Henighan, Child, Ramesh, Ziegler, Wu, Winter, Hesse, Chen, Sigler, Litwin, Gray, Chess, Clark, Berner, McCandlish, Radford, Sutskever, & Amodei (2020). Language models are few-shot learners. *Conference on Neural Information Processing Systems*.

Burgess, Higgins, Pal, Matthey, Watters, Desjardins, & Lerchner (2017). Understanding disentangling in β-VAE. *Workshop on "Learning Disentangled Representations", Conference on Neural Information Processing Systems*.

Burkhardt, Siekiera, & Kramer (2018). Semi-supervised Bayesian active learning for text classification. *Workshop on "Bayesian Deep Learning", Conference on Neural Information Processing Systems*.

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

Chan, Li, & Oymak (2021). On the marginal benefit of active learning: does self-supervision eat its cake? *International Conference on Acoustics, Speech and Signal Processing*.

Chen, Fan, Girshick, & He (2020a). Improved baselines with momentum contrastive learning. *arXiv*.

Chen, Kornblith, Norouzi, & Hinton (2020b). A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning*.

Chen, Kornblith, Swersky, Norouzi, & Hinton (2020c). Big self-supervised models are strong semi-supervised learners. *Conference on Neural Information Processing Systems*.

Chen, Li, Grosse, & Duvenaud (2018). Isolating sources of disentanglement in variational autoencoders. *Conference on Neural Information Processing Systems*.

Chitta, Álvarez, & Lesnikowski (2018). Large-scale visual active learning with deep probabilistic ensembles. *arXiv*.

Chu, Chiang, Emam, Czaja, Leapman, Goldblum, & Goldstein (2022). Active learning at the ImageNet scale. *OpenReview*.

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

da Costa, Fini, Nabi, Sebe, & Ricci (2022). solo-learn: a library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research*.

da Costa-Luis (2019). tqdm: a fast, extensible progress meter for Python and CLI. *Journal of Open Source Software*.

Daxberger, Kristiadi, Immer, Eschenhagen, Bauer, & Hennig (2021). Laplace redux—effortless Bayesian deep learning. *Conference on Neural Information Processing Systems*.

Denker & LeCun (1990). Transforming neural-net output levels to probability distributions. *Conference on Neural Information Processing Systems*.

Dosovitskiy, Beyer, Kolesnikov, Weissenborn, Zhai, Unterthiner, Dehghani, Minderer, Heigold, Gelly, Uszkoreit, & Houlsby (2021). An image is worth 16x16 words: transformers for image recognition at scale. *International Conference on Learning Representations*.

Drugman, Pylkkönen, & Kneser (2016). Active and semi-supervised learning in ASR: benefits on the acoustic and language models. *Interspeech*.

Dubois, Kastanos, Lines, & Melman (2019). Disentangling VAE. github.com/yanndubs/disentangling-vae.

Ebrahimi, Gan, Salahi, & Darrell (2020). Minimax active learning. *arXiv*.

Ein-Dor, Halfon, Gera, Shnarch, Dankin, Choshen, Danilevsky, Aharonov, Katz, & Slonim (2020). Active learning for BERT: an empirical study. *Conference on Empirical Methods in Natural Language Processing*.

Engstrom, Ilyas, Salman, Santurkar, & Tsipras (2019). Robustness (Python library). github.com/madrylab/robustness.

Farquhar (2022). *Understanding approximation for Bayesian inference in neural networks.* PhD thesis, University of Oxford.

Fortuin (2021). Priors in Bayesian deep learning: a review. *International Statistical Review.*

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.*

Gao, Zhang, Yu, Arik, Davis, & Pfister (2020). Consistency-based semi-supervised active learning: towards minimizing labeling cost. *European Conference on Computer Vision.*

Gelman, Carlin, Stern, Dunson, Vehtari, & Rubin (2013). *Bayesian Data Analysis.* CRC Press.

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.*

Gleave & Irving (2022). Uncertainty estimation for language reward models. *arXiv.*

Gudovskiy, Hodgkinson, Yamaguchi, & Tsukizawa (2020). Deep active learning for biased datasets via Fisher kernel self-supervision. *Conference on Computer Vision and Pattern Recognition.*

Guo, Ding, Gao, & Wang (2016). Semi-supervised active learning with cross-class sample transfer. *International Joint Conference on Artificial Intelligence.*

Guo, Pang, Sun, Li, & Chen (2021a). Redundancy removal adversarial active learning based on norm online uncertainty indicator. *Computational Intelligence and Neuroscience.*

Guo, Shi, Kang, Kuang, Tang, Jiang, Sun, Wu, & Zhuang (2021b). Semi-supervised active learning for semi-supervised models: exploit adversarial examples with graph-based virtual labels. *International Conference on Computer Vision.*

Hacohen, Dekel, & Weinshall (2022). Active learning on a budget: opposite strategies suit high and low budgets. *International Conference on Machine Learning.*

Hao, Lu, Huang, Wang, Hu, Liu, Chen, & Lee (2020). ASGN: an active semi-supervised graph neural network for molecular property prediction. *International Conference on Knowledge Discovery and Data Mining.*

Harris, Millman, van der Walt, Gommers, Virtanen, Cournapeau, Wieser, Taylor, Berg, Smith, Kern, Picus, Hoyer, van Kerkwijk, Brett, Haldane, Fernandez del Rio, Wiebe, Peterson, Gerard-Marchant, Sheppard, Reddy, Weckesser, Abbasi, Gohlke, & Oliphant (2020). Array programming with NumPy. *Nature.*

He, Zhang, Ren, & Sun (2015). Deep residual learning for image recognition. *Conference on Computer Vision and Pattern Recognition.*

Hénaff, Srinivas, De Fauw, Razavi, Doersch, Eslami, & van den Oord (2020). Data-efficient image recognition with contrastive predictive coding. *International Conference on Machine Learning.*

Higgins, Matthey, Pal, Burgess, Glorot, Botvinick, Mohamed, & Lerchner (2017). β-VAE: learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations.*

Hinton, Osindero, & Teh (2006). A fast learning algorithm for deep belief nets. *Neural Computation.*

Hinton & van Camp (1993). Keeping neural networks simple by minimizing the description length of the weights. *Conference on Computational Learning Theory.*

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

Huang, Wang, Xiong, Huan, & Dou (2021). Semi-supervised active learning with temporal output discrepancy. *International Conference on Computer Vision.*

Hunter (2007). Matplotlib: a 2D graphics environment. *Computing in Science & Engineering.*

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

Johnson, Douze, & Jégou (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data.*

Kim & Mnih (2018). Disentangling by factorising. *International Conference on Machine Learning.*

Kingma, Salimans, & Welling (2015). Variational dropout and the local reparameterization trick. *Conference on Neural Information Processing Systems.*

Kingma & Welling (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations.*

Kirsch (2023a). *Advancing deep active learning and data subset selection: unifying principles with information-theory intuitions*. PhD thesis, University of Oxford.

Kirsch (2023b). Black-box batch active learning for regression. *Transactions on Machine Learning Research*.

Kirsch (2023c). Speeding up BatchBALD: a k-BALD family of approximations for active learning. *arXiv*.

Kirsch, Farquhar, Atighehchian, Jesson, Branchaud-Charron, & Gal (2023). Stochastic batch acquisition: a simple baseline for deep active learning. *Transactions on Machine Learning Research*.

Kirsch, Rainforth, & Gal (2021). Test distribution-aware active learning: a principled approach against distribution shift and outliers. *arXiv*.

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

Kluyver, Ragan-Kelley, Perez, Granger, Bussonnier, Frederic, Kelley, Hamrick, Grout, Corlay, Ivanov, Avila, Abdalla, Willing, & Jupyter development team (2016). Jupyter notebooks—a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*.

Kristiadi, Hein, & Hennig (2020). Being Bayesian, even just a bit, fixes overconfidence in relu networks. *International Conference on Machine Learning*.

Krizhevsky (2009). Learning multiple layers of features from tiny images. Master's thesis, University of Toronto.

Kruschke (2014). *Doing Bayesian Data Analysis: a Tutorial with R, JAGS, and Stan*. Academic Press.

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*.

Li (2020). Bayesianize: a Bayesian neural network wrapper in PyTorch. github.com/microsoft/bayesianize.

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

Lüth, Bungert, Klein, & Jaeger (2023). Navigating the pitfalls of active learning evaluation: a systematic framework for meaningful performance assessment. *Conference on Neural Information Processing Systems*.

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

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

MacKay (1992c). A practical Bayesian framework for backpropagation networks. *Neural Computation*.

Maekawa, Zhang, Kim, Rahman, & Hruschka (2022). Low-resource interactive active labeling for fine-tuning language models. *Conference on Empirical Methods in Natural Language Processing*.

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

Margatina, Barrault, & Aletras (2022). On the importance of effectively adapting pretrained language models for active learning. *Annual Meeting of the Association for Computational Linguistics*.

Matthey, Higgins, Hassabis, & Lerchner (2017). dSprites: disentanglement testing sprites dataset. github.com/deepmind/dsprites-dataset.

McCallum & Nigam (1998). Employing EM and pool-based active learning for text classification. *International Conference on Machine Learning*.

McKinney (2010). Data structures for statistical computing in Python. *Python in Science Conference*.

Miller (1995). WordNet: a lexical database for English. *Communications of the ACM*.

Mittal, Niemeijer, Schäfer, & Brox (2023). Best practices in active learning for semantic segmentation. *arXiv*.

Mittal, Tatarchenko, Çiçek, & Brox (2019). Parting with illusions about deep active learning. *arXiv*.

Mohamadi, Doretto, & Adjeroh (2022). Deep active ensemble sampling for image classification. *Asian Conference on Computer Vision*.

Mottaghi & Yeung (2019). Adversarial representation active learning. *arXiv*.

Murray, Allingham, Antorán, & Hernández-Lobato (2021). Depth uncertainty networks for active learning. *Workshop on "Bayesian Deep Learning", Conference on Neural Information Processing Systems*.

Muslea, Minton, & Knoblock (2002). Active + semi-supervised learning = robust multi-view learning. *International Conference on Machine Learning.*

Nguyen & Smeulders (2004). Active learning using pre-clustering. *International Conference on Machine learning.*

Osband, Asghari, Van Roy, McAleese, Aslanides, & Irving (2023). Fine-tuning language models via epistemic neural networks. *International Conference on Machine Learning.*

Osband, Wen, Asghari, Dwaracherla, Lu, Ibrahimi, Lawson, Hao, O'Donoghue, & Van Roy (2022a). The neural testbed: evaluating joint predictions. *Conference on Neural Information Processing Systems.*

Osband, Wen, Asghari, Dwaracherla, Lu, & Van Roy (2022b). Evaluating high-order predictive distributions in deep learning. *Conference on Uncertainty in Artificial Intelligence.*

Paszke, Gross, Massa, Lerer, Bradbury, Chanan, Killeen, Lin, Gimelshein, Antiga, Desmaison, Kopf, Yang, DeVito, Raison, Tejani, Chilamkurthy, Steiner, Fang, Bai, & Chintala (2019). PyTorch: an imperative style, high-performance deep learning library. *Conference on Neural Information Processing Systems.*

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.*

Pop & Fulop (2018). Deep ensemble Bayesian active learning. *Workshop on "Bayesian Deep Learning", Conference on Neural Information Processing Systems.*

Pourahmadi, Nooralinejad, & Pirsiavash (2021). A simple baseline for low-budget active learning. *arXiv.*

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, Foster, Ivanova, & Bickford Smith (2024). Modern Bayesian experimental design. *Statistical Science.*

Ranzato, Poultney, Chopra, & LeCun (2006). Efficient learning of sparse representations with an energy-based model. *Conference on Neural Information Processing Systems.*

Rezende, Mohamed, & Wierstra (2014). Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning.*

Rhee, Erdenee, Kyun, Ahmed, & Jin (2017). Active and semi-supervised learning for object detection with imperfect data. *Cognitive Systems Research.*

Riccardi & Hakkani-Tür (2003). Active and unsupervised learning for automatic speech recognition. *European Conference on Speech Communication and Technology.*

Rottmann, Kahl, & Gottschalk (2018). Deep Bayesian active semi-supervised learning. *International Conference on Machine Learning and Applications.*

Russakovsky, Deng, Su, Krause, Satheesh, Ma, Huang, Karpathy, Khosla, Bernstein, Berg, & Fei-Fei (2014). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision.*

Sener & Savarese (2018). Active learning for convolutional neural networks: a core-set approach. *International Conference on Learning Representations.*

Seo, Kim, Ahn, & Lee (2022). Active learning on pre-trained language model with task-independent triplet loss. *AAAI Conference on Artificial Intelligence.*

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.*

Seung, Opper, & Sompolinsky (1992). Query by committee. *Workshop on Computational Learning Theory.*

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

Shui, Zhou, Gagné, & Wang (2020). Deep active learning: unified and principled method for query and training. *International Conference on Artificial Intelligence and Statistics.*

Siméoni, Budnik, Avrithis, & Gravier (2020). Rethinking deep active learning: using unlabeled data at model training. *International Conference on Pattern Recognition.*

Smith & Gal (2018). Understanding measures of uncertainty for adversarial example detection. *Conference on Uncertainty in Artificial Intelligence.*

Snoek, Rippel, Swersky, Kiros, Satish, Sundaram, Patwary, Prabhat, & Adams (2015). Scalable Bayesian optimization using deep neural networks. *International Conference on Machine Learning.*

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

Susmelj, Heller, Wirth, Prescott, & Ebner (2020). Lightly. github.com/lightly-ai/lightly.

Tamkin, Nguyen, Deshpande, Mu, & Goodman (2022). Active learning helps pretrained models learn the intended task. *Conference on Neural Information Processing Systems.*

Tomanek & Hahn (2009). Semi-supervised active learning for sequence labeling. *Annual Meeting of the Association for Computational Linguistics.*

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

Tran, Rossi, Milios, & Filippone (2022). All you need is a good functional prior for Bayesian deep learning. *Journal of Machine Learning Research.*

Tur, Hakkani-Tür, & Schapire (2005). Combining active and semi-supervised learning for spoken language understanding. *Speech Communication.*

Virtanen, Gommers, Oliphant, Haberland, Reddy, Cournapeau, Burovski, Peterson, Weckesser, Bright, van der Walt, Brett, Wilson, Millman, Mayorov, Nelson, Jones, Kern, Larson, Carey, Polat, Feng, Moore, VanderPlas, Laxalde, Perktold, Cimrman, Henriksen, Quintero, Harris, Archibald, Ribeiro, Pedregosa, van Mulbregt, & SciPy 1.0 contributors (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods.*

Wang, Li, Ma, Ma, Guan, & Zheng (2020). Dual adversarial network for deep active learning. *European Conference on Computer Vision.*

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

Wenzel, Roth, Veeling, Swiatkowski, Tran, Mandt, Snoek, Salimans, Jenatton, & Nowozin (2020). How good is the Bayes posterior in deep neural networks really? *International Conference on Machine Learning.*

Yadan (2019). Hydra—a framework for elegantly configuring complex applications. github.com/facebookresearch/hydra.

Yang, Ma, Nie, Chang, & Hauptmann (2015). Multiclass active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision.*

Yang, Yau, Fei-Fei, Deng, & Russakovsky (2021). A study of face obfuscation in ImageNet. *International Conference on Machine Learning.*

Yehuda, Dekel, Hacohen, & Weinshall (2022). Active learning through a covering lens. *Conference on Neural Information Processing Systems.*

Yi, Seo, Park, & Choi (2022). Using self-supervised pretext tasks for active learning. *European Conference on Computer Vision.*

Yuan, Lin, & Boyd-Graber (2020). Cold-start active learning through self-supervised language modeling. *Conference on Empirical Methods in Natural Language Processing.*

Zhang, Bengio, Hardt, Recht, & Vinyals (2017). Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations.*

Zhang, Chen, Canal, Mussmann, Das, Bhatt, Zhu, Bilmes, Du, Jamieson, & Nowak (2024). LabelBench: a comprehensive framework for benchmarking adaptive label-efficient learning. *arXiv.*

Zhang & Lee (2019). Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chemical Science.*

Zhang, Li, Yang, Wang, Zha, & Huang (2020). State-relabeling adversarial active learning. *Conference on Computer Vision and Pattern Recognition.*

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

Checklist

1. For all models and algorithms presented, check if you include:

    (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. *Yes. See Sections 2 and 4.*

    (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. *Yes. See Table 1.*

    (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. *Yes. See link to code in Section 6.*

2. For any theoretical claim, check if you include:

    (a) Statements of the full set of assumptions of all theoretical results. *Yes. See Proposition 1.*

    (b) Complete proofs of all theoretical results. *Yes. See Proposition 1.*

    (c) Clear explanations of any assumptions. *Yes. See Proposition 1.*

3. For all figures and tables that present empirical results, check if you include:

    (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). *Yes. See link to code in Section 6.*

    (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). *Yes. See Appendix B.*

    (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). *Yes. See Appendix B.*

    (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). *Yes. See Appendix C.*

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

    (a) Citations of the creator, if your work uses existing assets. *Yes. See Appendix C.*

    (b) The license information of the assets, if applicable. *Yes. See Appendix C.*

    (c) New assets either in the supplemental material or as a URL, if applicable. *Yes. See link to code in Section 6.*

    (d) Information about consent from data providers/curators. *Not applicable.*

    (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. *One of our experiments is based on ImageNet data. Some images in this dataset depict people, raising concerns with respect to issues such as demographic representation and privacy. We support work that aims to address these and other ethical concerns, for example the work of Asano et al (2021) and Yang et al (2021).*

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

    (a) The full text of instructions given to participants and screenshots. *Not applicable.*

    (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. *Not applicable.*

    (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. *Not applicable.*

A   Extra results
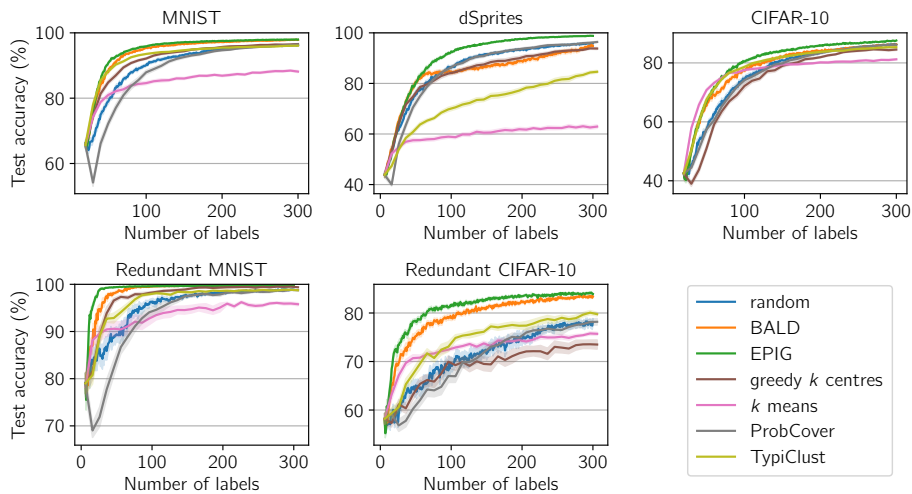
Figure 7 EPIG outperforms non-Bayesian acquisition methods in the settings previously covered in Figures 1 and 4. Unlike in Figure 5, BADGE and maximum entropy are not shown here because the former does not by default work with random forests (Kirsch, 2023b) and the latter is equivalent to BALD when using random forests.
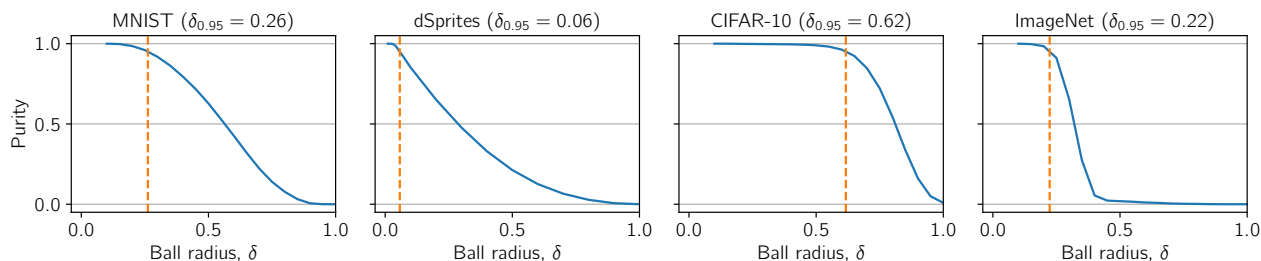
Figure 8 ProbCover has a "ball radius" parameter that needs to be tuned for the particular representation being used. Shown here are the results we used to choose the ball radius to ensure a "purity" of 0.95, in line with Yehuda et al (2022).

B   Experiment details

B.1   Experiment in Section 3

Using MNIST data (LeCun et al, 1998) and the two-convolutional-layer model architecture from Kirsch et al (2019), we investigated the uncertainty estimates produced by four approaches to Bayesian deep learning:

1. Deep ensembles (Lakshminarayanan et al, 2017) using an ensemble size of 10.

2. Laplace approximation (Denker & LeCun, 1990; MacKay, 1992c) using a diagonal posterior covariance matrix and a standard Gaussian prior, $\mathcal{N}(0, I)$, when determining the posterior covariance.

3. Monte Carlo dropout (Gal & Ghahramani, 2016) using a dropout rate of 0.5.

4. Mean-field variational inference (Hinton & van Camp, 1993) using a Gaussian variational posterior, a standard Gaussian prior, $\mathcal{N}(0, I)$, and the local reparameterisation trick (Kingma et al, 2015) during training.

With each approach we trained models on labelled datasets (randomly sampled and balanced by class) of two different sizes, $N = 10$ and $N = 10,000$, then we estimated the models' uncertainty on the MNIST test set. For training we used the same setup as described in Appendix B.2.4, except with a validation set of 1,000 examples.

For uncertainty estimation we considered the decomposition discussed by Smith & Gal (2018):

$$\underbrace{\text{BALD}(x)}_{\text{reducible}} = \underbrace{\text{H}\big[\mathbb{E}_{p_\phi(\theta)}[p_\phi(y|x,\theta)]\big]}_{\text{total}} - \underbrace{\mathbb{E}_{p_\phi(\theta)}[\text{H}[p_\phi(y|x,\theta)]]}_{\text{irreducible}}.$$

While deep ensembles permit exact computation of the required expectations over $p_\phi(\theta)$, the other approaches require Monte Carlo estimation of those expectations, for which we used 1,000 parameter samples.

B.2 Experiments in Section 6

B.2.1 Data

We focused on four datasets: MNIST (LeCun et al, 1998), dSprites (Matthey et al, 2017), CIFAR-10 (Krizhevsky, 2009) and ImageNet (ILSVRC2012) (Russakovsky et al, 2014). dSprites is less commonly studied than the others in the context of active learning but appeals because it is simple—the inputs are black-and-white images of symbols with known factors of variation, including the symbol shape (ellipse, heart or square), which we use as the class—yet harder than MNIST for fully supervised models, and a number of pretraining methods have been developed with it in mind (Burgess et al, 2017; Chen et al, 2018; Higgins et al, 2017; Kim & Mnih, 2018).

In addition to studying the standard versions of MNIST and CIFAR-10, we looked at "redundant" versions in which the unlabelled pool remains as before but the task of interest involves classifying inputs from just two classes (1 vs 7 for MNIST; cat vs dog for CIFAR-10). All the remaining classes were grouped into a third "other" class during acquisition and training. At test time we treated the model as a binary classifier between the two classes of interest. This test-time setup differs from that used by Bickford Smith et al (2023) in the original Redundant MNIST experiment; we used it because it more faithfully reflects the assumptions of the task setup.

Our ImageNet experiment focuses on classifying images into eleven superclasses based on the WordNet semantic hierarchy (Engstrom et al, 2019; Miller, 1995): bird (52 classes), boat (6 classes), car (10 classes), cat (8 classes), dog (116 classes), fruit (7 classes), fungus (7 classes), insect (27 classes), monkey (13 classes), truck (7 classes) and other (747 classes). Contrasting with our experiment based on Redundant MNIST and Redundant CIFAR-10, in this experiment we aimed to simulate scenarios where we cannot be sure that test inputs will belong to the main classes of interest. So here we defined the test set to include all the examples from the main superclasses but additionally a small number of examples from the "other" superclass. Inducing a mismatch between the pool set and the test set in terms of their class distributions, we set up the pool of candidate inputs for labelling to have an equal number of inputs from each of the 1,000 ImageNet classes.

B.2.2 Models

Main semi-supervised models Our proposed model setup comprises an encoder and a prediction head. We tailored our choice of encoder architecture and pretraining technique to each dataset, mostly based on the performance of publicly available model checkpoints in evaluations using randomly acquired data. For MNIST we used the two-convolutional-layer architecture from Kirsch et al (2019), with dropout disabled, and SimCLR pretraining (Chen et al, 2020b); for dSprites, the architecture from Burgess et al (2017) and β-TCVAE pretraining (Chen et al, 2018); for CIFAR-10, the ResNet-18 architecture from He et al (2015) and MoCo v2 pretraining (Chen et al, 2020a); for ImageNet, the ViT-B/4 architecture from Dosovitskiy et al (2021) and MSN pretraining (Assran et al, 2022). In our choice of prediction heads we sought a simple off-the-shelf option. For MNIST, dSprites and CIFAR-10 we therefore used a random forest (Breiman, 2001) configured using the default settings in Scikit-learn (Pedregosa et al, 2011). This prediction head performed poorly on ImageNet, so there we used a dropout-enabled fully connected neural network with a dropout rate of 0.1 and three hidden layers of 128 units.

Fully supervised models We chose fully supervised models to match the size of the semi-supervised models. For MNIST we used the two-convolutional-layer model from Kirsch et al (2019) with their dropout rate of 0.5; for dSprites, the encoder from Burgess et al (2017) combined with a stack of dropout-enabled fully connected layers (with a dropout rate of 0.05); for CIFAR-10, a ResNet-18 modified to incorporate two additional dropout-enabled fully connected layers (with a dropout rate of 0.5) before the final layer, as used by Kirsch et al (2021).

Alternative semi-supervised models In order to investigate the sensitivity of BALD and EPIG acquisition to changes in the semi-supervised model, we looked at alternative model setups for MNIST. For the encoder we

tried using the same architecture as we used for dSprites but trained as part of a standard variational autoencoder (Kingma & Welling, 2014; Rezende et al, 2014). For the prediction head we tried a fully connected neural network with one hidden layer of 128 units, using Laplace approximation to infer a parameter distribution (for this we used a standard Gaussian prior, $\mathcal{N}(0, I)$, and a diagonal, tempered posterior (Aitchison, 2021), with tempering implemented by raising the likelihood term to a power of $\dim(\theta_h)$, the parameter count of the prediction head).

B.2.3 Active learning

We initialised the training dataset by sampling two examples from each class, then we ran the loop described in Section 2 until the training-label budget (1,000 for ImageNet; 300 for the other datasets) was used up. We evaluated a number of acquisition methods: random, BALD (Houlsby et al, 2011), EPIG (Bickford Smith et al, 2023), maximum entropy (Settles & Craven, 2008), BADGE (Ash et al, 2020), greedy k centres (Sener & Savarese, 2018), k means (Pourahmadi et al, 2021), ProbCover (Yehuda et al, 2022) and TypiClust (Hacohen et al, 2022). To estimate BALD and EPIG we used the same setup as Bickford Smith et al (2023): we used 100 realisations of θ_h (for random forests this meant using the individual trees; otherwise this meant sampling from the parameter distribution) and 100 samples of x_* from a finite set of unlabelled inputs representative of the downstream task. After each time the model was trained we evaluated its predictive accuracy on the test dataset. We ran the entire active-learning process with each acquisition function 20 times with different random-number-generator seeds, then plotted the test accuracy (mean $\pm$ standard error) as a function of the size of the training dataset.

B.2.4 Training neural-network prediction heads

We used a loss function comprising the negative log likelihood (NLL) of the model parameters on the training data along with an l_2 regulariser on the parameters. We ran up to 50,000 steps of gradient-based optimisation with a learning rate of 0.01. To mitigate overfitting we recorded the NLL on a small validation set (roughly a fifth of the size of the training-label budget), stopped training if this NLL did not decrease for more than 5,000 consecutive steps, then restored the model parameters to the setting that gave the lowest validation-set NLL.

C Resources

C.1 Software

Project	Citation	License	URL
BatchBALD Redux	Kirsch et al (2019)	Apache 2.0	github.com/blackhc/batchbald_redux
Bayesianize	Li (2020)	MIT	github.com/microsoft/bayesianize
Disentangling VAE	Dubois et al (2019)	MIT	github.com/yanndubs/disentangling-vae
Faiss	Johnson et al (2019)	MIT	github.com/facebookresearch/faiss
Hydra	Yadan (2019)	MIT	github.com/facebookresearch/hydra
Jupyter	Kluyver et al (2016)	BSD (3-clause)	jupyter.org
Laplace	Daxberger et al (2021)	MIT	github.com/aleximmer/laplace
Lightly	Susmelj et al (2020)	MIT	github.com/lightly-ai/lightly
Matplotlib	Hunter (2007)	PSF (modified)	matplotlib.org
MSN	Assran et al (2022)	CC BY-NC 4.0	github.com/facebookresearch/msn
NumPy	Harris et al (2020)	BSD (3-clause)	numpy.org
Pandas	McKinney (2010)	BSD (3-clause)	pandas.pydata.org
PyTorch	Paszke et al (2019)	BSD-style	pytorch.org
Robustness	Engstrom et al (2019)	MIT	github.com/madrylab/robustness
Scikit-learn	Pedregosa et al (2011)	BSD (3-clause)	scikit-learn.org
SciPy	Virtanen et al (2020)	BSD (3-clause)	scipy.org
solo-learn	da Costa et al (2022)	MIT	github.com/vturrisi/solo-learn
tqdm	da Costa-Luis (2019)	MPL	github.com/tqdm/tqdm

Table 2 This work builds on a number of software projects.

C.2 Compute

We used an internal cluster equipped with Nvidia GeForce RTX 2080 Ti (12GB) and Nvidia Titan RTX (24GB) chips. Our cheapest run (dSprites data, semi-supervised model, random acquisition) took 15 minutes on an Nvidia GeForce RTX 2080 Ti. Our most expensive run (CIFAR-10 data, fully supervised model, EPIG acquisition) took 200 hours on an Nvidia Titan RTX.