Active Surrogate Estimators: An Active Learning Approach to Label–Efficient Model Evaluation

Jannik Kossen[1]* **Sebastian Farquhar**[1,3] **Yarin Gal**[1] **Tom Rainforth**[2]

[1] OATML, Department of Computer Science, University of Oxford
[2] Department of Statistics, University of Oxford
[3] DeepMind

Abstract

We propose *Active Surrogate Estimators (ASEs)*, a new method for label-efficient model evaluation. Evaluating model performance is a challenging and important problem when labels are expensive. ASEs address this active testing problem using a surrogate-based estimation approach that interpolates the errors of points with unknown labels, rather than forming a Monte Carlo estimator. ASEs actively learn the underlying surrogate, and we propose a novel acquisition strategy, *XWED*, that tailors this learning to the final estimation task. We find that ASEs offer greater label-efficiency than the current state-of-the-art when applied to challenging model evaluation problems for deep neural networks.

1 Introduction

Machine learning research routinely assumes access to large, labeled datasets for supervised learning. Practitioners often do not have that luxury. Instead, no labeled data might be available initially, and label acquisition often involves significant cost: it might require medical experts to analyze x-ray images, physicists to perform experiments in particle accelerators, or chemists to synthesize molecules. In these situations, we should carefully acquire only those labels that are most informative for the task at hand, maximizing performance while minimizing labeling cost [1, 62, 27, 43].

While a wide array of active learning methods [1, 62] have been developed to address the problem of *training* models when labels are expensive, much less attention has been given to the problem of label-efficient model *evaluation*. Such model evaluation is important, e.g. for performance analysis, model selection, or hyperparameter optimization. Indeed, Lowell et al. [45] argue that a lack of efficient approaches to model evaluation holds back adoption of active learning methods in practice.

Prior work in label-efficient model evaluation has focused on using importance sampling (IS) techniques to reduce the variance of Monte Carlo (MC) estimates [38, 59, 70]. In particular, recent approaches [38, 18] have iteratively refined the proposal using information from previous evaluations, producing adaptive IS (AIS) estimators with state-of-the-art performance.

Though these methods provide substantial label-efficiency gains over naive sampling, they have some clear weaknesses that suggest improvements should be possible. Firstly, the AIS weights assigned to samples depend on the acquisition proposal used *at that iteration*. Thus, early samples are often inappropriately weighted, because we are yet to learn an effective proposal. This can significantly increase the variance of the resulting estimator. Secondly, MC estimates can be inefficient at dealing with label noise: constraining the estimate to be a weighted sum of observed losses is limiting when the losses themselves are noisy. Finally, IS methods can be restrictive on the acquisition strategies used to choose points to label: acquisition must be stochastic with known probabilities.

*Correspondence to jannik.kossen@cs.ox.ac.uk.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

To address these limitations, we introduce **Active Surrogate Estimators (ASEs)**, a new approach to label-efficient model evaluation that is based on interpolation rather than MC estimation. Specifically, ASEs actively learn a surrogate model to directly predict losses at all observed *and unobserved* points in the pool. Here, the surrogate can take the form of any appropriate machine learning model, and we introduce a general framework for learning effective surrogates. The interpolation-based nature of ASEs leads to a variety of advantages over MC estimates: ASEs better accommodate noisy label observations, adapt quicker to new information, and provide complete flexibility in how new datapoints are acquired, allowing for careful customizations that are not appropriate in the AIS framework. To exploit this, we propose a novel acquisition strategy, **XWED**, that tailors the acquisition towards points that will be most beneficial to the final estimation task.

To summarize, our main contributions are: **(1)** We introduce Active Surrogate Estimators; a new method for label-efficient model evaluation based on actively learning a surrogate model (§3). **(2)** We introduce XWED, a novel acquisition strategy that targets a weighted disagreement that is highly relevant to our final surrogate-based prediction (§4). **(3)** We provide theoretical insights into ASEs' errors (§5). **(4)** We apply ASEs to the evaluation of deep neural networks trained for image classification, where we consistently outperform the current state-of-the-art (§7).

2 Label-Efficient Model Evaluation

We first describe the general setup of label-efficient model evaluation, also known as *active testing* [38]. We wish to evaluate the predictive model $f : \mathcal{X} \to \mathcal{Y}$, mapping inputs $x \in \mathcal{X}$ to labels $y \in \mathcal{Y}$. Here, model evaluation refers to estimating the expected loss of the model predictions, that is, the risk

$$r = \mathbb{E}\left[\mathcal{L}(f(X), Y)\right], \tag{1}$$

where we use capital letters for random variables, $\mathcal{L}$ is a loss function, and the expectation is over the true test distribution, which can be different to the distribution of the training data for f. We make no assumptions on f or the way in which it has been trained. We only require that we can query its predictions at arbitrary x. Because we wish to estimate the risk of f, we keep f fixed and it does not change during evaluation. Everything here applies to both classification and regression tasks.

Following existing approaches, we study active testing in the pool-based setting. This is the typical scenario experienced in practice [62]: samples x_i can often be cheaply collected in the form of unlabeled data, but the density for the data generating process itself is typically unknown. Concretely, one assumes the existence of a pool of samples, with indices $\mathcal{P} = \{1, \dots, N\}$, drawn from the test distribution, and for which we know input features x_i but not labels y_i, $i \in \mathcal{P}$.

If we knew the labels at all test pool indices $\mathcal{P}$, we could compute the empirical test risk $\hat{r} = {}^1\!/\!_N \sum_{i \in \mathcal{P}} \mathcal{L}(f(x_i), y_i)$. When N is large, this provides an accurate estimate of r. However, one often cannot afford to evaluate $\hat{r}$: the cost associated with label acquisition is too high to acquire labels for all points in $\mathcal{P}$. Instead, active testing methods must decide on a subset $\mathcal{O} \subseteq \mathcal{P}$ of points to label, where typically we have that $M = |\mathcal{O}| \ll |\mathcal{P}| = N$.

The simplest approach would be to sample indices uniformly at random. The resulting naive MC estimate, $\hat{r}_{\text{iid}} = {}^1\!/\!_M \sum_{i \in \mathcal{O}} \mathcal{L}(f(x_i), y_i)$, represents the predominant practice of using randomly constructed test set instead of actively evaluating models. However, the variance of $\hat{r}_{\text{iid}}$ is often unacceptably high when M is small. Methods for active testing seek to provide risk estimates with lower estimation error than $\hat{r}_{\text{iid}}$, without increasing the number of label acquisitions M.

Similar to active learning, it is advantageous in active testing to not select all points $\mathcal{O}$ upfront, but instead to select them adaptively by using information from previous evaluations. The mechanism by which points are chosen is known as an *acquisition strategy*. As with active learning, it is typical to assume that the cost of labeling is dominant, and all other costs incurred during model evaluation are typically ignored. Therefore, it can be worth (repeatedly) training surrogate models, despite the resulting expense, if this helps minimize the estimation error for a given label budget M.

2.1 Importance Sampling–Based Active Testing

Importance sampling (IS) [30, 51] is a classic strategy for forming MC estimates with lower variance than naive MC. IS samples points from a proposal $X \sim q$ instead of the original reference distribution, which we will denote p. Adaptive importance sampling (AIS) [32, 48, 5] improves IS by refining the proposal q using information from previous iterations. However, traditional IS and AIS are only applicable if we can evaluate the density of the reference distribution, at least up to normalization. This is not the case in many relevant active testing scenarios, e.g. when x_i are high-dimensional images.

IS cannot be applied directly in the pool-based setting because of the need to sample *without* replacement. However, in the context of a training objective for active learning, Farquhar et al. [18] introduced LURE, which converts IS to this pool-based setting. Specifically, they compute

$$\hat{r}_{\text{LURE}} = \frac{1}{M} \sum\nolimits_{m=1}^{M} v_m \, \mathcal{L}\left(f(x_{i_m}), y_{i_m}\right), \; v_m = 1 + \frac{N-M}{N-m}\left(\frac{1}{(N-m+1)q(i_m)} - 1\right), \quad (2)$$

where $q(i_m)$ is the probability of choosing to label sample i_m at iteration m. Kossen et al. [38] showed that LURE can also be applied to active testing, for which the optimal proposal is

$$q^*(i_m) \propto \mathbb{E}[\mathcal{L}(f(X), Y)|X = x_{i_m}]. \quad (3)$$

In practice, q^* is intractable as the distribution $Y|X = x_{i_m}$ is unknown. However, using ideas from AIS, one can adaptively improve q with information gathered from previous labels. In particular, Kossen et al. [38] introduce strategies to directly approximate (3) (see §6). When q correctly identifies regions in the input where the loss is large, $\hat{r}_{\text{LURE}}$ reduces variance compared to $\hat{r}_{\text{iid}}$.

Unlike standard AIS, adaptive LURE conveniently does not require knowledge of an explicit density $p(x)$—a pool of samples suffices. However, most techniques for reducing the variance of the weights in the AIS literature (see e.g. Bugallo et al. [5]) cannot currently be transferred over to adaptive LURE. In particular, we cannot employ methods like deterministic mixture AIS [50] to update historic weights using future proposal adaptations. This means that poorly-calibrated early weights can persist and significantly increase the variance of the estimator.

3 Active Surrogate Estimators

We now introduce Active Surrogate Estimators (ASEs), our novel method for label-efficient model evaluation. Following the setup in §2, we assume a pool of unlabeled samples from which we iteratively choose points to label. Unlike existing methods that rely on MC estimates, ASEs use an interpolation-based approach to estimate model risk: a surrogate model directly predicts losses for *all* points in the test pool. ASEs are further based around an adaptive and iterative procedure: we use an active learning approach to carefully select the labels used to train our surrogate. We first discuss the form that the ASE and surrogate take, and later cover our novel acquisition strategy in §4.

3.1 Deriving the ASE

The fundamental idea of ASEs is that of interpolation-based estimation: we wish to build a risk estimator of the form $\hat{r}_{\text{ASE}}(\phi) = {}^1\!/_N \sum_{i\in\mathcal{P}} g(x_i;\phi)$, where $g(x_i;\phi)$ is a flexible surrogate model with parameters ϕ that approximates the conditional expected loss, $\mathbb{E}\left[\mathcal{L}(f(X), Y)|X = x\right]$.

Though algorithmically quite different, the motivation behind this interpolation-based approach is quite similar to that of variational inference (VI) [74]: rather than relying on a sample-based approximation for $\hat{r}$, we are converting our estimation into the optimization of a functional approximation. Though, as with VI, this comes at the cost of weaker theoretical asymptotic guarantees compared with MC estimation (see §5), it allows us to generalize more effectively when only a small number of labels are available. This trade-off is typically worthwhile in the small sample size regime of active testing.

To decide what form g should take, we first note that the theoretically optimal g—in terms of mean squared error (MSE) for r—is simply the expected loss above, analogously to the optimal proposal for LURE. While this is itself obviously intractable, we can learn an effective g by trying to approximate this optimal g^*. Though one could directly approximate this using a generic regressor, such as a neural network, this would disregard that we already know the loss function $\mathcal{L}$, can evaluate f, and know how both are combined to give the optimal g^*. In fact, the only unknown is the conditional expectation $Y|X = x$ for the test distribution.

We therefore introduce an auxiliary model $\pi(y|x;\phi)$ of only this conditional. Combining π, f, and $\mathcal{L}$, we construct g as $g(x;\phi) = \mathbb{E}_{Y\sim\pi(\cdot|x;\phi)}\left[\mathcal{L}(f(x), Y)\right]$, where the only learnable components of g are the parameters of π, ϕ. With π introduced, we can finally write the ASE as

$$\hat{r}_{\text{ASE}}(\phi) = \frac{1}{N} \sum\nolimits_{i\in\mathcal{P}} \mathbb{E}_{Y\sim\pi(\cdot|x_i;\phi)}\left[\mathcal{L}(f(x_i), Y)\right], \quad (4)$$

where we emphasize that the expectation over $Y|X = x_i$ is now with respect to π and not the unknown true label conditional. Note that the fact that the ASE includes an explicit model of the label noise is a decisive advantage over prior work (see §8). For classification, we can now directly evaluate the expectation over the labels as a weighted sum over the predictive probabilities $\sum_y \pi(y|x_i;\phi)\mathcal{L}(f(x_i), y)$. For regression, closed-form solutions of the expectation exist for some losses, and we can always compute unbiased and accurate MC estimates using cheap samples from π.

Assembling g from $\mathcal{L}$, f, and π will typically be advantageous to modeling g directly: a direct model would need to learn not only about predictions of f, but also how they relate to the test distribution. In contrast, learning models $\pi(y|x;\phi)$ of conditional outcomes is standard procedure. This also makes it easy to include other prior information. For example, initializing π on the *training* data can sometimes give the ASE a useful estimate of the risk before observing any test data at all. Lastly, the modular construction of g allows us to quickly adapt the ASE to different models or loss functions.

Often, f, like π, will be a probabilistic model of outcomes. However, it will generally be a highly inappropriate choice for π, such that it is important to always learn a separate auxiliary model. This is because we are essentially using π to predict the errors of f, and we should therefore aim to decorrelate the errors of π from those of f by training a separate π. Further, we may want to choose π from a different model class than f, e.g. to allow for meaningful incorporation of uncertainties. Lastly, while f is by construction fixed, we can adapt π during evaluation to reflect our updated knowledge of the test label distribution, as we explain next.

3.2 Adaptive Refinement of ASEs

ASEs are adaptive: they iteratively improve their estimate by updating the surrogate as new labels are acquired. A crucial component to them doing this effectively is that they actively select the labels to acquire at each iteration using an acquisition strategy. As such, they can be thought of as performing estimation via active learning of the auxiliary model π, noting that the estimate they produce is, in turn, fully defined by this auxiliary model and the pool set.

Algorithm 1 Adaptive Refinement of ASEs

```
Inputs: Test pool $\{x_i\}_{i\in\mathcal{P}}$, acquisition strategy $a$, target model $f$, initial auxiliary model $\pi_0$
Initialize $\mathcal{O} = \emptyset$ and $\mathcal{U} = \mathcal{P}$
for $m = 1$ to $M$ do
    Select $i_m = a(\pi_{m-1}, f, \{x_i\}_{i\in\mathcal{P}}, \{y_i\}_{i\in\mathcal{O}})$
    Acquire label $y_{i_m} \sim Y \mid x_{i_m}$
    Update $\mathcal{O} \leftarrow \mathcal{O} \cup i_m$ and $\mathcal{U} \leftarrow \mathcal{U} \setminus i_m$
    Update auxiliary model $\pi_m$ (e.g. by retraining)
end for
Return $\hat{r}_{\text{ASE}}$ as per (4) with $\pi = \pi_M$
```

To define this adaptive refinement process more formally, we denote with $\mathcal{U}$ and $\mathcal{O}$ the sets of unobserved and observed points in the test pool. These are initialized as $\mathcal{U} = \mathcal{P}$ and $\mathcal{O} = \emptyset$. At each subsequent iteration m of our adaptive refinement process, an acquisition strategy a selects an index $i_m \in \mathcal{U}$ of an unlabeled datapoint, for which we then acquire a label y_{i_m}. We remove i_m from $\mathcal{U}$, and add it to $\mathcal{O}$. Then, we obtain an improved ASE by retraining π, e.g. on the total observed test data $\{(x_i, y_i) : i \in \mathcal{O}\}$; this retraining does not necessarily need to be performed at every iteration if doing so would be computationally infeasible. A summary of this approach is given in Algorithm 1.

In §E, we show that the computational complexity of ASEs is given by $\mathcal{O}\left(M \cdot (N + |\mathcal{D}_{\text{train}}|)\right)$, assuming an initial training set of size $|\mathcal{D}_{\text{train}}|$ for the surrogate. This is identical to the computational complexity of LURE-based active testing but more expensive than a naive MC estimate.

In ASEs, our acquisition strategy takes the current auxiliary model π_m, the original model f, the pool of samples, and all previous evaluations as input, i.e. we have $i_m = a(\pi_{m-1}, f, \{x_i\}_{i\in\mathcal{P}}, \{y_i\}_{i\in\mathcal{O}})$. A good acquisition strategy selects points such that the expected ASE error after M iterations is minimal [28]. While ASEs can be used with arbitrary acquisition strategies, we next introduce a novel acquisition strategy that we specifically design for use with ASEs.

4 The XWED Acquisition Function

It is important we tailor our acquisition strategy to the problem of active surrogate estimation, rather than apply more generic active learning approaches. We need to acquire labels such that we produce the best ASE estimate given a limited labeling budget. To do this, we propose a novel acquisition function called the E**x**pected **We**ighted **D**isagreement, or *XWED* for short, that is designed specifically to target improvements of the ASE estimate itself.

In line with acquisition strategies in active learning, we now require that π takes the form $\pi(y|x;\phi) = \mathbb{E}_{\Theta\sim\pi(\cdot;\phi)}\left[\pi(y|x,\Theta;\phi)\right]$, where Θ represents some model parameters we wish to learn about. In a BNN, Θ is the weights and biases of the network, while ϕ corresponds to the parameters of the posterior approximation given by previous observations (such that it contains all information from previous data). To avoid clutter, we omit ϕ in the following presentation of acquisition functions.

For active learning, BALD [27] is an established acquisition strategies that acquires points with high epistemic uncertainty [13, 34]. For a predictive model π as above, BALD can be calculated as the disagreement between the total predictive entropy and the expected entropies over posterior draws,

$$\mathrm{BALD}(x) = \mathbb{E}_{Y\sim\pi(\cdot|x)}\left[-\log\pi(Y|x)\right] - \mathbb{E}_{\Theta\sim\pi(\cdot)}\left[\mathbb{E}_{Y\sim\pi(\cdot|x,\Theta)}\left[-\log\pi(Y|x,\Theta)\right]\right]. \tag{5}$$

However, BALD, and other acquisition functions from active learning, are ill-suited to ASEs because they do not consider the effect the uncertainties in π have on the ASE estimate. In particular, ASEs depend on the loss $\mathcal{L}(y, f(x))$ of the original model f at the pool samples, but neither the form of the loss function $\mathcal{L}$ nor the original model f are considered in BALD acquisition.

Acquisition for ASEs should instead target points that are both informative about Θ *and* which we expect will contribute significantly to our final risk estimate. The basis of XWED is therefore to *weight* the disagreement terms for particular inputs x using their corresponding loss $\mathcal{L}(Y, f(x))$:

$$\begin{aligned}\mathrm{XWED}(x) = &\,\mathbb{E}_{Y\sim\pi(\cdot|x)}\left[-\mathcal{L}(Y,f(x))\log\pi(Y|x)\right] \\ &- \mathbb{E}_{\Theta\sim\pi(\cdot)}\left[\mathbb{E}_{Y\sim\pi(\cdot|x,\Theta)}\left[-\mathcal{L}(Y,f(x))\log\pi(Y|x,\Theta)\right]\right].\end{aligned} \tag{6}$$

With XWED, we acquire points with large epistemic uncertainty [13, 34] in the auxiliary π about the outcomes *only if they are relevant to the estimate* $\hat{r}_{\text{ASE}}$. To convert this acquisition function to an acquisition strategy, we simply choose the point for which the XWED score is maximal: $a(\pi_{m-1}, f, \{x_i\}_{i\in\mathcal{P}}, \{y_i\}_{i\in\mathcal{O}}) = \arg\max_{i\in\mathcal{U}} \mathrm{XWED}(x_i)$. As we show in §7, acquisition from XWED allows for significant improvements compared to BALD and other strategies.

XWED is compatible with a variety of model classes for π, such as Bayesian Neural Networks (BNNs) [46, 21] and deep ensembles [41]. In §D.2, we give a worked through example for the practical computation of XWED when the main model f is a neural network and the surrogate π a deep ensemble, both trained for classification. We emphasize that XWED is an acquisition strategy explicitly designed for active testing; it is not even defined in a traditional active learning context, where the distinction between a fixed model f and the adaptive auxiliary model π does not exist.

5 Theoretical Analysis of the ASE Error

We next provide a theoretical analysis of ASEs to understand individual contributions to their error. Specifically, we will break down the errors of ASEs into (I) error from using a finite evaluation pool and (II) error from imperfection of the surrogate, further breaking the latter down as (IIA) error from any limited expressivity in our surrogate class and (IIB) error from imperfect training.

To start, let Φ be the final parameters obtained from the stochastic training procedure of the surrogate g, such that our learned surrogate is $g(\cdot;\Phi)$. Here, Φ is a random variable with the stochasticity originating from (i) sampling the test pool itself, (ii) any stochasticity in the acquisition strategy, (iii) randomness in the acquired labels, and (iv) stochasticity in the optimization process of the surrogate.

Next, let $X_0 \sim p(x)$ be a hypothetical additional input sample that is independent from the test pool. We can use this to define $G(\phi) = \mathbb{E}\left[g(X_0;\phi)\right]$ as the true expectation with respect to the input distribution of a surrogate with parameters ϕ. In turn, we can define $\phi^* = \mathrm{argmin}_\phi\, |G(\phi) - r|$ as the parameters which give the best approximation of the true expected loss $\mathbb{E}\left[\mathcal{L}(f(X), Y)|X = x\right]$. Note here that ϕ^* is not random as it depends only on the fixed function class of g and the fixed true risk r.

Denoting our test pool as $P = \{X_1, \dots, X_N\}$, where $X_i \sim p(x)$ are independently sampled, we can write $\hat{r}_{\text{ASE}}(\Phi, P)$ to make the dependency of the ASE estimator (4) on P and Φ explicit. By using the triangle inequality, we can then break down the error $\|\hat{r}_{\text{ASE}}(\Phi, P) - r\|$ to the true model risk r as

$$\begin{aligned}\|\hat{r}_{\text{ASE}}(\Phi,P) - r\| &= \|\hat{r}_{\text{ASE}}(\Phi,P) - G(\Phi) + G(\Phi) - G(\phi^*) + G(\phi^*) - r\| \\ &\leq \|\hat{r}_{\text{ASE}}(\Phi,P) - G(\Phi)\| + \|G(\Phi) - r\| \\ &\leq \|\hat{r}_{\text{ASE}}(\Phi,P) - G(\Phi)\| + |G(\phi^*) - r| + \|G(\Phi) - G(\phi^*)\|,\end{aligned} \tag{7}$$

where $\|\cdot\|$ is any valid norm, and we note that $G(\phi)$ is independent of X_0 and $G(\phi^*)$ is deterministic.

We now consider each of the terms in (7):

$(I) = \|\hat{r}_{\text{ASE}}(\Phi, P) - G(\Phi)\|$ is the error that originates from only evaluating the surrogate on a finite set of pool of points. As per standard results for MC estimation, this will generally decrease as $\mathcal{O}(1/\sqrt{N})$, such that it will be small for large test pools. Note here that $\hat{r}_{\text{ASE}}(\Phi, P)$ is typically not an unbiased estimator for $G(\Phi)$ as the set of input points available in the pool can influence the distribution of Φ. However, this effect will typically be very small and the bias will diminish as N increases. We can also, if desired, eliminate it completely by evaluating g over a separate test pool than that used during the acquisition of test labels and updating of g. This variation on the standard ASE estimator can also be useful when the pool is impractically large for performing label acquisition, as we can evaluate the final learned g on a larger pool than used during its training.

$(II) = \|G(\Phi) - r\|$ represents the error from the imperfection of the surrogate. For the squared norm, it equals $\mathbb{E}_{\Phi}[(\mathbb{E}_{X_0}[g(X_0;\Phi)] - r)^2]$, such that it is the expected squared difference (over surrogate parameters) between the surrogate's expectation over inputs and the true risk. It can itself be further broken down into terms (IIA) and (IIB) as described below. Note that this term demonstrates the critical dependency of ASE's performance on the regressional performance of the surrogate itself.

$(IIA) = |G(\phi^*) - r|$ gives the error from limits in the choice of the surrogate function class. It measures the difference between g^* and $g(\cdot;\phi^*)$. Whenever our surrogate function class contains the true expected loss, this term will be exactly zero. Even when this is not the case, it should generally be negligible for sensible choices of the surrogate function class. For example, if π is a deep neural network, the limiting factor will almost always be the training error from using finite data (i.e. term (IIB) below), rather than the theoretical expressivity of the network.

$(IIB) = \|G(\Phi) - G(\phi^*)\|$ can be thought of as the error originating from the training of the surrogate: it is the norm of the difference between the true expectation of the surrogate we have learned and the true expectation of the best possible surrogate we could have learned.

5.1 Discussion and Empirical Analysis

In practice, (IIB) will typically dominate the ASE error. If term (I) is significant, the pool size is a limiting factor, which can only be the case when our ASE estimation has been successful (assuming $M \ll N$), as its implies we have a highly accurate estimate of the full empirical test risk $\hat{r}$. If term (IIA) is significant, this means we are using an inappropriately weak model class for our surrogate.

To check these assertions hold in practice, we empirically estimate (I) and (IIA) for the experiments in §7.1 at $M = 1$: we find that they respectively make up only 0.00013% and 0.031% of the full error; thus (IIB) is completely dominant here.

Further characterizing the dominating term (IIB) in the fully general setting is unfortunately not feasible: it equates to characterizing the error of a regression, which necessarily can only be done by considering a particular class of surrogate. However, for a particular choice of surrogate, existing results from learning theory could be applied to further categorize its convergence. Here work on the convergence of models in active *learning* [23, 24, 58, 7, 54] will be particularly appropriate, noting that these all make specific assumptions about the model and acquisition strategy. More specifically, for Gaussian process surrogates and acquisition strategies from Bayesian quadrature, the results of Kanagawa & Hennig [31] could be applied. On the other hand, for surrogates based on deep learning architectures, one could use results from the convergence of stochastic gradient descent methods [47], which typically give rates of $\mathcal{O}(1/\sqrt{M})$ under appropriate assumptions. These will require unbiased gradients in the optimization of the objective function, which we can achieve for ASEs with stochastic acquisition and the de-biasing scheme of Farquhar et al. [18].

In summary, for large pool sizes and appropriate surrogate classes, (IIB) will be the dominant contributor to the error, noting that the inequality in (7) becomes tight as one of the errors dominates. As such, the convergence of ASEs essentially reduces to the convergence of the surrogate, and thus, by extension, the convergence of the auxiliary model. Moreover, if we can ensure that this auxiliary model convergences to the true conditional output distribution as the number of label acquisitions M increases (this itself implies that our surrogate function class is sufficient powerful and so $(IIA) = 0$), we can thus ensure that the ASE estimator itself converges, noting that $M \to \infty$ also predicates $N \to \infty$ (as $N > M$ by construction), such that $(I) \to 0$.

6 Related Work

Active Learning. Active learning (AL) [1, 62, 27, 21, 61] allows for the training of supervised models in a label-efficient manner, and it is a particular instance of optimal experimental design (OED) [44, 6, 60, 20]. OED formalizes the idea of targeted acquisition to maximize the information gained with respect to a target objective. Relatedly, transductive AL [71, 68, 36] performs AL of model predictions with respect to a known set of points. Filstroff et al. [19] propose AL for decision making.

Active Testing. Sawade et al. [59] explore IS for active risk estimation. They rely on vanilla IS which does not respect the pool-based setting [38]. Yilmaz et al. [70] have recently proposed active testing based on Poisson sampling. Both [70] and [59] outperform simple baselines; however, they use *non-adaptive* acquisitions that cannot adjust to test data.

Of particular relevance is the current state-of-the-art approach by Kossen et al. [38]. Building on the LURE estimator, they propose an AIS-style estimate of the model test risk (§2.1). To approximate the optimal acquisition (3), they, like us, introduce an auxiliary model $\pi(y|x;\phi)$ that is (usually) adapted as test labels are acquired. Given restrictions of the LURE estimator, they always sample from

$$q(i_m) \propto \mathbb{E}_{Y \sim \pi(\cdot|x_i;\phi)} \left[\mathcal{L}(f(x_{i_m}), Y) \right] \tag{8}$$

for label acquisition and do not actively improve the proposal, e.g. through active learning.

[2, 33, 40, 29] explore *stratification* for efficient evaluation of classifiers: a simple, non-adaptive criterion is used to divide the test pool into strata, from which labels are selected uniformly at random. These approaches can be improved by applying the method considered here within the strata.

Surrogates. The use of surrogate models to replace expensive black-box functions is pervasive throughout much of the sciences [53, 37, 10]. Bayesian optimisation (BO) [63, 4] and Bayesian quadrature (BQ) [49, 56, 3] are prominent examples of surrogate-based approaches that actively select samples. BO iteratively finds the maximum of an unknown function, using a surrogate model and ideas similar to active learning to select sample locations. Unlike active learning, BO does not focus on a globally accurate surrogate. BQ obtains a Bayesian posterior for the integration of a Gaussian process (GP) [55] surrogate against a known data density $p(x)$. Despite biased estimates, BQ can drastically outperform MC in low dimensions. However, BQ does not scale well to high dimensions, where, often, GP surrogates are not appropriate and explicit densities $p(x)$ are unknown and hard to approximate.

Model Evaluation without Labels. Similar to our experiments in §7.3, [22, 57, 9, 11, 12, 64, 65] study the evaluation of model test performance without test labels or even without test sets. However, unlike ASEs, these works are based around performing meta analysis and require access to multiple related datasets (for which test labels are available). They rely on regressions from dataset/model summaries to performance scores and cannot be used in the setting considered in this paper, where only one dataset is available. Similarly, meta-learning for hyperparameter optimization, e.g. [15, 67, 16, 72, 26], interpolates test performance across a variety of datasets *and* model hyperparameters.

7 Experimental Results

We next study the performance of ASEs for active testing in comparison to relevant baselines. Concretely, we compare to naive MC and the current state-of-the-art LURE-based active testing approach by Kossen et al. [38]. As a general and common application, we consider the important problem of actively evaluating deep neural networks trained on high-dimensional image datasets.

We give full details on the experiments in the appendix. In particular, §B and §C contain additional results and figures, and §D gives further details on the computation of XWED and the baselines.

7.1 Evaluation with Distribution Shift

We first explore model evaluation in a challenging distribution shift scenario. Concretely, we are given a fixed model trained on 2000 digits of MNIST [42], where all sevens have been removed from the training set. We then estimate the model risk on a test set with all classes present in their normal proportions. The test set has size $N = 60000$ but we can only afford to acquire labels for a small subset. Such distribution shift settings are relevant in practice *and* challenging for active testing: both ASE- and LURE-based active testing needs to approximate the expected loss, and this is particularly hard under distribution shift, where the initial π has yet to learn about the test distribution.

Following [38], we use radial Bayesian Neural Networks [17] for f and π, with π initialized on the training set and regularly retrained during evaluation. For LURE, we acquire labels by sampling from (8). For the ASE, we use the XWED acquisition strategy (6). Both ASE and LURE use the same auxiliary model class π, but arrive at different network weights, ϕ_{ASE} and ϕ_{LURE}, because their distinct acquisition strategies lead to different test observations used for the re-training of π.

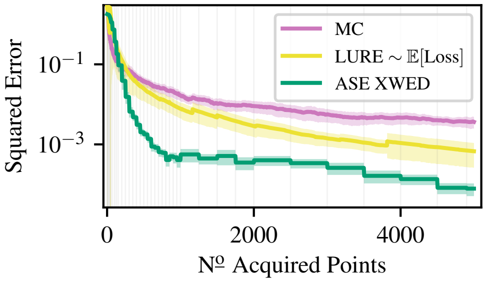


Figure 1: ASE significantly improves over LURE and MC in terms of squared error for the distribution shift experiment. Shown are mean errors and the shading is two std. errors over 100 runs. We retrain π at steps marked with grey lines.

Main Result. Figure 1 shows the MSE for ASE and baselines when estimating the unknown test pool risk. Note here that we are careful to use the best possible setup of the LURE baseline to construct a challenging benchmark, as shown in the next section. We see that ASE improves on all baselines when used with the XWED acquisition strategy. Further, LURE can struggle with variance, with errors of individual runs occasionally spiking. This occurs when there is a significant mismatch between observed loss and proposal, e.g. when an acquired point is ambiguous or mislabeled. This does not happen for ASEs, as we do not weight true observations against predictions in the estimate.

7.2 Acquisition Strategies for ASE and LURE

We next perform an ablation to investigate different acquisition strategies besides XWED in the distribution shift scenario: We select points with maximal BALD score, Eq. (5). Further, we acquire points by sampling from the expected loss; this is the exact same acquisition strategy used for LURE-based active testing, cf. (8). See §D.1 for further details on how we compute these baselines.

Figure 2 visualizes how often the acquisition functions acquire the missing class as testing progresses, and Figure 3 (a) shows that ASE is able to outperform naive MC for *all* acquisition strategies, but there are noticeable variations in their performance. The XWED acquisition, that we have proposed specifically with ASEs in mind, provides the best performance. BALD acquisition does not target those points that lead to the largest improvements in the estimate, which leads to a much slower decrease in ASE error early on, though it does catch up with XWED later. Expected loss acquisition—which is optimal for LURE—will sample the missing class (where loss is high) almost exclusively, which

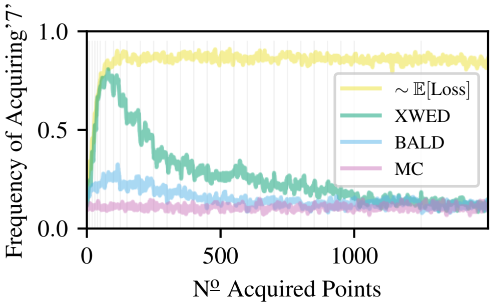


Figure 2: Proportion of acquired points which have class '7'. Results averaged over 100 runs.

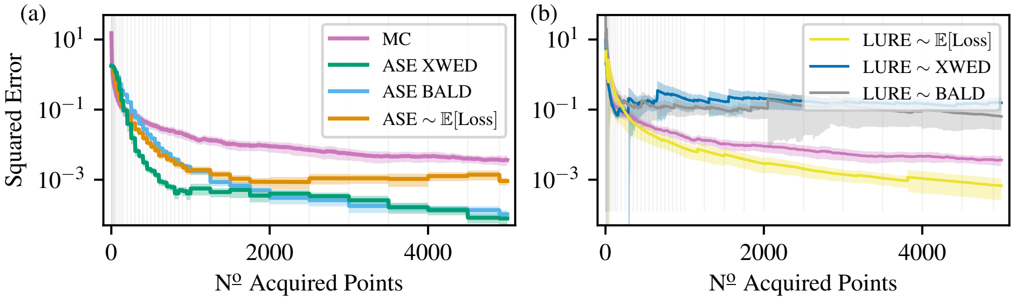


Figure 3: Different acquisition strategies for ASEs and LURE. (a) While XWED performs best, ASE outperforms MC for all investigated acquisition functions. (b) LURE suffers high variance for all acquisition strategies except expected loss. Shown are mean errors, shading is two std. errors over 100 runs. We retrain π at steps marked with grey lines. The '~'-symbol indicates stochastic acquisition.

initially works well but later leads to π failing to improve on the other classes. In contrast, XWED focuses on the missing class early on but—once the loss is well characterized for the missing class—explores other areas as well. This behavior is preferable over the baselines, and we continue this discussion in §B.1.

In Figure 3 (b), we explore how LURE behaves under different acquisition strategies, such as sampling from XWED or BALD scores. We find that LURE is noticeably more sensitive to the acquisition strategy than ASE: both XWED and BALD acquisition lead to high-variance estimates that no longer outperform the naive MC baseline. It seems one cannot expect acquisition strategies other than expected loss to perform well with LURE. In contrast, ASEs can directly apply any acquisition scheme without introducing such sensitivity because the acquisition does not affect which points are included in the estimate or how they are weighted.

We provide additional results in the appendix: §B.2 shows that adaptive improvement of the acquisition function is necessary for label-efficient active testing in the distribution shift scenario, and, in §B.3, we study additional choices for ASE acquisition strategies (which do not outperform XWED).

7.3 Evaluation of CNNs for Classification

Next, we study active testing of ResNets on more complex image classification datasets. Concretely, we study the efficient evaluation of a Resnet-18 [25] on Fashion-MNIST [69] and CIFAR-10 [39], and of a WideResNet [73] on CIFAR-100. In each case, we train the model to be evaluated, f, on a training set containing $40\,000$ points, and then use an evaluation pool of size $N = 2000$. As a ground truth, we compare to the empirical risk on a held out set of $20\,000$ points. For π, we use deep ensembles [41] of the original ResNet models. This experiment design is also used by [38].

In this experiment, there is no distribution shift between the training data for f and the data on which we evaluate it. At the same time, the training set is large and, therefore, a lot of prior information is available to train the initial π. Further, the number of test acquisitions is comparatively small ($M = 50$) and unlikely to affect π meaningfully, given the models have already seen $40\,000$ points from the training data. Indeed, we find that updating π with this additional test data makes no noticeable difference to performance. To minimize computational costs, we therefore report results with π fixed throughout this experiment, such that, here, π is equal for ASE and LURE. A constant π leads to a constant LURE proposal (making the method more akin to IS than AIS) and constant prediction for ASE.

Figure 4 shows that ASEs perform effectively in this setting, significantly outperforming LURE and naive MC for all datasets. We also display the error from the finite pool size, by comparing the empirical risk of all N unobserved test points against our additional ground truth set and see that ASE consistently performs similarly to this pool limit. This suggests that the pool size N may be the limit of the performance for ASEs here rather than the quality of the surrogate.

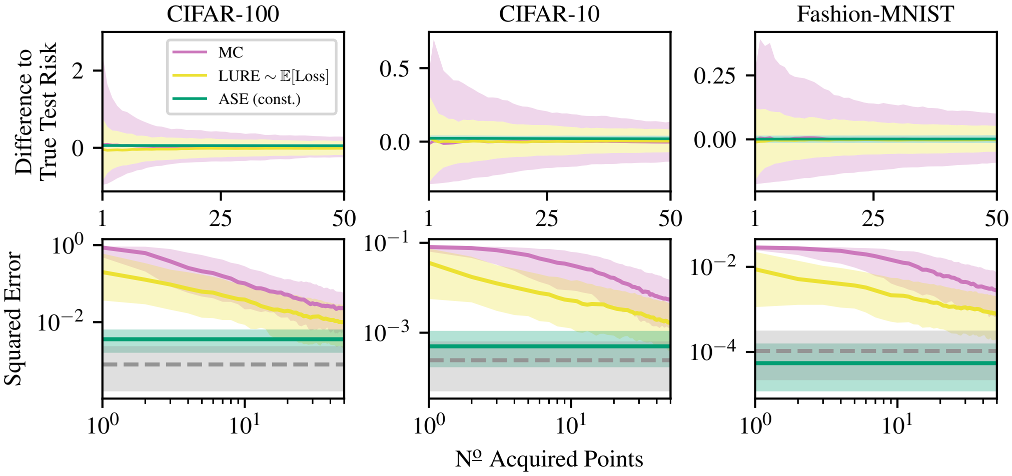


Figure 4: Active Testing of ResNets on CIFAR-100, CIFAR-10, and FashionMNIST without retraining the auxiliary model π. Despite not observing any test labels, ASE again improves upon LURE. We display medians over 1000 random test sets and 10/90 % (top) and 25/75 % (bottom) quantiles (top: too small to be visible for ASE). In grey, we display a lower bound on the error from the finite pool.

This experiment produces a rather unusual result: as π is fixed throughout, the ASE actually does not make use of *any* information from the test labels! Its accurate estimates of the test risk are entirely due to information from the training data. Note that LURE here has access to the same π as ASE in addition to up to 50 test labels. Nevertheless, the surrogate prediction of the ASE at all pool points leads to much lower error than a LURE estimate. This 'zero-shot' setting opens up interesting avenues for model evaluation when there are no test labels available at all.

However, in practice, one may not know if the test data follow the same distribution as the training data. Figure B.1 (a) shows that both LURE and ASE perform poorly when not retraining π in the distribution shift setting for any labeling budget M. We therefore generally suggest regular adaptation of π for both ASE- and LURE-based active testing approaches. And even without distribution shift, the information from test labels will always eventually become relevant as M increases.

ASEs are compatible with arbitrary loss functions, cf. (4). For example, for 0-1-Loss, ASEs can be used to estimate the accuracy of the main model. In §B.4, we demonstrate that ASEs continue to outperform all other baselines for the task of accuracy estimation.

In our theoretical analysis in §5, we argue that, in most practical settings, the error originating from the surrogate will dominate the ASE error. In §B.5, we reduce the size of the training set to $10\,000$. This makes the problem more challenging for ASEs because it reduces the quality of the surrogate model. Nevertheless, we find that ASEs maintain their significant advantage over the baselines.

8 Discussion

In the previous section, we have seen ASEs outperform MC-based alternatives in practice. Here, we discuss advantages of ASEs that are important to their practical success.

Actively Learning Estimators. ASEs are flexible about which active learning strategy can be used. In contrast, AIS-based strategies must balance competing goals: the acquisition strategy must both find informative data to improve the proposal in future iterations and provide importance-sampling weights that reduce the estimator variance. For AIS, acquiring the most informative data with high probability can make the estimator *worse* because the resulting acquisition probabilities are a bad approximation for the optimal q^*. Prior work using LURE does not address this trade-off and always acquires with the best approximation of q^* available. ASE acquisitions can also be deterministic, while AIS requires full-support stochastic acquisition to avoid bias.

Surrogate Updates. For AIS, improving the proposal distribution only improves future acquisitions, but does not retrospectively improve weights attached to earlier observations. Bad initial acquisition weights have a lasting influence on the variance of AIS. In contrast, ASE makes better retrospective use of information: the iterative refinement of g leads to updated predictions at all pool points.

Noisy Labels. ASEs have a further advantage when label distributions are *noisy*, i.e. observations y are stochastic even for the true label conditional $p(y|x)$. This is common in practice, because of ambiguous inputs near class-boundaries, mislabeled examples, or measurement error. ASEs directly model the noisy outcome conditionals using π. If π is accurate, the ASE error does not increase due to noisy labels, and the ASE can interpolate beyond the noise. In contrast, AIS suffers badly: noisy observations form a direct part of the estimate, leading to increased variance; while the acquisition (8) models the noise, this only affects weights, and noisy loss observations directly enter the estimate (2). Chen & Choe [8] provide a detailed analysis of how stochastic observations affect IS estimators negatively.

9 Conclusions

We have introduced Active Surrogate Estimators (ASEs), a surrogate-based method for label-efficient model evaluation. The surrogate is actively learned, and we introduced the XWED acquisition function to directly account for the risk prediction task. Our experiments show that ASEs and XWED acquisition give state-of-the-art performance for efficient evaluation of deep image classifiers.

Acknowledgments and Disclosure of Funding

We thank Andreas Kirsch and the anonymous reviewers for helpful feedback and interesting discussions that have led to numerous improvements of the paper. We acknowledge funding from the New College Yeotown Scholarship (JK) and the Oxford CDT in Cyber Security (SF).

References

[1] Atlas, L., Cohn, D., and Ladner, R. Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems*, volume 2, pp. 566–573, 1990.

[2] Bennett, P. N. and Carvalho, V. R. Online stratified sampling: evaluating classifiers at web-scale. In *International conference on Information and knowledge management*, volume 19, pp. 1581–1584, 2010.

[3] Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., and Sejdinovic, D. Probabilistic integration: A role in statistical computation? *Statistical Science*, 34(1):1–22, 2019.

[4] Brochu, E., Cora, V. M., and De Freitas, N. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv:1012.2599*, 2010.

[5] Bugallo, M. F., Elvira, V., Martino, L., Luengo, D., Miguez, J., and Djuric, P. M. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

[6] Chaloner, K. and Verdinelli, I. Bayesian experimental design: A review. *Statistical Science*, pp. 273–304, 1995.

[7] Chaudhuri, K., Kakade, S. M., Netrapalli, P., and Sanghavi, S. Convergence rates of active learning for maximum likelihood estimation. *Advances in Neural Information Processing Systems*, 28, 2015.

[8] Chen, Y.-C. and Choe, Y. Importance sampling and its optimality for stochastic simulation models. *Electronic Journal of Statistics*, 13(2):3386–3423, 2019.

[9] Corneanu, C. A., Escalera, S., and Martinez, A. M. Computing the testing error without a testing set. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[10] Cozad, A., Sahinidis, N. V., and Miller, D. C. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014.

[11] Deng, W. and Zheng, L. Are labels always necessary for classifier accuracy evaluation? In *Conference on Computer Vision and Pattern Recognition*, 2021.

[12] Deng, W., Gould, S., and Zheng, L. What does rotation prediction tell us about classifier accuracy under varying testing environments? In *International Conference on Machine Learning*, 2021.

[13] Der Kiureghian, A. and Ditlevsen, O. Aleatory or epistemic? does it matter? *Structural safety*, 31(2): 105–112, 2009.

[14] DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552*, 2017.

[15] Eggensperger, K., Hutter, F., Hoos, H., and Leyton-Brown, K. Efficient benchmarking of hyperparameter optimizers via surrogates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

[16] Eggensperger, K., Lindauer, M., Hoos, H. H., Hutter, F., and Leyton-Brown, K. Efficient benchmarking of algorithm configurators via model-based surrogates. *Machine Learning*, 2018.

[17] Farquhar, S., Osborne, M. A., and Gal, Y. Radial Bayesian neural networks: Beyond discrete support in large-scale Bayesian deep learning. In *International Conference on Artificial Intelligence and Statistics*, volume 23, pp. 1352–1362, 2020.

[18] Farquhar, S., Gal, Y., and Rainforth, T. On statistical bias in active learning: How and when to fix it. In *International Conference on Learning Representations*, 2021.

[19] Filstroff, L., Sundin, I., Mikkola, P., Tiulpin, A., Kylmäoja, J., and Kaski, S. Targeted active learning for bayesian decision-making. *arXiv:2106.04193*, 2021.

[20] Foster, A., Ivanova, D. R., Malik, I., and Rainforth, T. Deep adaptive design: Amortizing sequential bayesian experimental design. In *International Conference on Machine Learning*, 2021.

[21] Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pp. 1183–1192, 2017.

[22] Guerra, S. B., Prudêncio, R. B., and Ludermir, T. B. Predicting the performance of learning algorithms using support vector machines as meta-regressors. In *International Conference on Artificial Neural Networks*, 2008.

[23] Hanneke, S. Adaptive rates of convergence in active learning. In *COLT*, 2009.

[24] Hanneke, S. Rates of convergence in active learning. *The Annals of Statistics*, pp. 333–361, 2011.

[25] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[26] He, X., Zhao, K., and Chu, X. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 2021.

[27] Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011.

[28] Imberg, H., Jonasson, J., and Axelson-Fisk, M. Optimal sampling in unbiased active learning. *Artificial Intelligence and Statistics*, 23, 2020.

[29] Ji, D., Logan, R. L., Smyth, P., and Steyvers, M. Active bayesian assessment of black-box classifiers. In *AAAI Conference on Artificial Intelligence*, volume 35, pp. 7935–7944, 2021.

[30] Kahn, H. and Marshall, A. W. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1:263–278, 1953.

[31] Kanagawa, M. and Hennig, P. Convergence guarantees for adaptive bayesian quadrature methods. In *Advances in neural information processing systems*, 2019.

[32] Karamchandani, A., Bjerager, P., and Cornell, C. Adaptive importance sampling. In *Structural Safety and Reliability*, pp. 855–862. ASCE, 1989.

[33] Katariya, N., Iyer, A., and Sarawagi, S. Active evaluation of classifiers on large datasets. In *International Conference on Data Mining*, volume 12, pp. 329–338, 2012.

[34] Kendall, A. and Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Advances In Neural Information Processing Systems*, 2017.

[35] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[36] Kirsch, A., Rainforth, T., and Gal, Y. Active learning under pool set distribution shift and noisy data. *arXiv:2106.11719*, 2021.

[37] Kleijnen, J. P. Kriging metamodeling in simulation: A review. *European journal of operational research*, 192(3):707–716, 2009.

[38] Kossen, J., Farquhar, S., Gal, Y., and Rainforth, T. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*, 2021.

[39] Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images, 2009.

[40] Kumar, A. and Raj, B. Classifier risk estimation under limited labeling resources. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 3–15, 2018.

[41] Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6402–6413, 2017.

[42] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[43] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

[44] Lindley, D. V. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pp. 986–1005, 1956.

[45] Lowell, D., Lipton, Z. C., and Wallace, B. C. Practical Obstacles to Deploying Active Learning. *Empirical Methods in Natural Language Processing*, November 2019.

[46] MacKay, D. J. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3): 448–472, 1992.

[47] Moulines, E. and Bach, F. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 24, 2011.

[48] Oh, M.-S. and Berger, J. O. Adaptive importance sampling in monte carlo integration. *Journal of Statistical Computation and Simulation*, 41(3-4):143–168, 1992.

[49] O'Hagan, A. Bayes–hermite quadrature. *Journal of statistical planning and inference*, 29(3):245–260, 1991.

[50] Owen, A. and Zhou, Y. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95:135–143, 2000.

[51] Owen, A. B. Monte carlo theory, methods and examples, 2013.

[52] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, 2019.

[53] Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K., and Jeff Wu, C. Building surrogate models based on detailed and approximate simulations. *Journal of Mechanical Design*, 128(4):668–677, 2005.

[54] Raj, A. and Bach, F. Convergence of uncertainty sampling for active learning. *arXiv:2110.15784*, 2021.

[55] Rasmussen, C. E. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71, 2003.

[56] Rasmussen, C. E. and Ghahramani, Z. Bayesian monte carlo. In *Advances in neural information processing systems*, pp. 505–512, 2003.

[57] Redyuk, S., Schelter, S., Rukat, T., Markl, V., and Biessmann, F. Learning to validate the predictions of black box machine learning models on unseen data. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 2019.

[58] Sabato, S. and Munos, R. Active regression by stratification. *Advances in Neural Information Processing Systems*, 27, 2014.

[59] Sawade, C., Landwehr, N., Bickel, S., and Scheffer, T. Active risk estimation. In *International Conference on Machine Learning*, volume 27, pp. 951–958, 2010.

[60] Sebastiani, P. and Wynn, H. P. Maximum entropy sampling and optimal Bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157, 2000.

[61] Sener, O. and Savarese, S. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[62] Settles, B. Active Learning Literature Survey. *Machine Learning*, 2010.

[63] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104, 2015.

[64] Sun, X., Hou, Y., Li, H., and Zheng, L. Label-free model evaluation with semi-structured dataset representations. *arXiv:2112.00694*, 2021.

[65] Talagala, T. S., Li, F., and Kang, Y. Fformpp: Feature-based forecast model performance prediction. *International Journal of Forecasting*, 2022.

[66] Van Rossum, G. and Drake, F. L. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

[67] Vanschoren, J. Meta-learning: A survey. *arXiv:1810.03548*, 2018.

[68] Wang, C., Sun, S., and Grosse, R. Beyond marginal uncertainty: How accurately can bayesian regression models estimate posterior predictive correlations? In *International Conference on Artificial Intelligence and Statistics*, pp. 2476–2484, 2021.

[69] Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017.

[70] Yilmaz, E., Hayes, P., Habib, R., Burgess, J., and Barber, D. Sample efficient model evaluation. *arXiv:2109.12043*, 2021.

[71] Yu, K., Bi, J., and Tresp, V. Active learning via transductive experimental design. In *International Conference on Machine learning*, volume 23, 2006.

[72] Yu, T. and Zhu, H. Hyper-parameter optimization: A review of algorithms and applications. *arXiv:2003.05689*, 2020.

[73] Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference*, pp. 87.1–87.12, 2016.

[74] Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

Checklist

1. For all authors...
   - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   - (b) Did you describe the limitations of your work? [Yes] See §5. The ASE error is mainly bounded by the error of the surrogate regressor.
   - (c) Did you discuss any potential negative societal impacts of your work? [Yes] §F discusses this in detail.
   - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

   w
2. If you are including theoretical results...
   - (a) Did you state the full set of assumptions of all theoretical results? [Yes] Cf. §5.
   - (b) Did you include complete proofs of all theoretical results? [Yes] Cf. §5.
3. If you ran experiments...
   - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Full code for reproducing the experiments is linked to in the appendix.
   - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We give all training details in §7 and §D.
   - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We repeat experiments and report error bars for all experiments.
   - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We report this in §G.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   - (a) If your work uses existing assets, did you cite the creators? [Yes] Yes, we cite the creators of the datasets and code libraries we use in §G.
   - (b) Did you mention the license of the assets? [Yes] Yes, we mention licenses in §G.
   - (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We release code for ASEs in the supplement.
   - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Does not apply.
   - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Does not apply.
5. If you used crowdsourcing or conducted research with human subjects...
   - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Does not apply.
   - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Does not apply.
   - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Does not apply.

Appendix

Active Surrogate Estimators: An Active Learning Approach to Label–Efficient Model Evaluation

A Code

We release the full code for reproducing the experiments at https://github.com/jlko/active-surrogate-estimators.

B Additional Results

B.1 Comparison of Acquisition Behavior

We next investigate how XWED outperforms all competing acquisition strategies in the experiments of §7.1. As a reminder, in the distribution shift setup the model f is not exposed to any samples of class '7' during training, while the test set contains '7's in their normal proportion. Hence, predictions at samples with class '7' should contribute significantly to the risk of f on the test distribution. However, the overall number of training points for f is relatively small such that contributions from other classes to the risk are not negligible.

In Figure 2, we examine the probability of acquiring a '7' as a function of the number of acquired points for XWED, BALD, expected loss acquisition, and a uniformly at random sampling strategy (MC). We see that XWED initially focuses on 7s but then diversifies. The baselines always prioritize 7s or never do. The XWED behavior is preferable: we are initially unsure about the loss of these points, but once the loss is well characterized for the 7s we should explore other areas as well. Note also that this highlights the benefit of ASEs over LURE: LURE, which requires expected loss acquisition, inefficiently needs to keep giving preferences to 7s even once the loss for these is well characterized.

B.2 Constant π Fails for Distribution Shift.

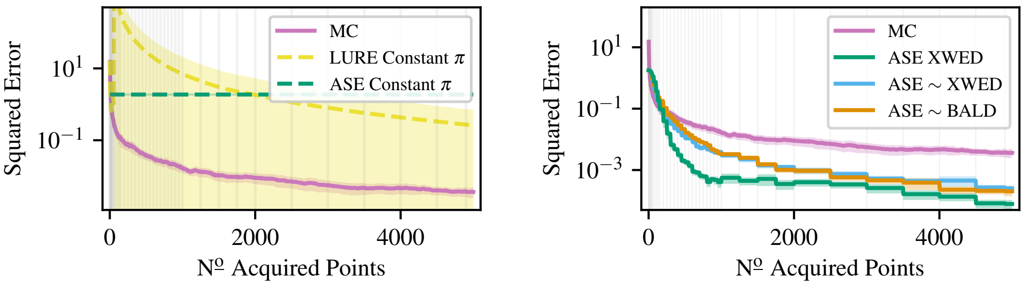
We investigate if retraining of the auxiliary model π is necessary to achieve competitive active testing performance in the distribution shift scenario. Here, we train π once on the original training data and then keep it constant throughout testing, not updating π with any information from the test data. ASE and LURE thus use the exact same π. When π is constant, the ASE *estimate* and the LURE *proposal* are constant. For LURE, we acquire by sampling from (8) as usual. Figure B.1 (a) shows that, for both ASE- and LURE-based active testing, naive MC cannot be outperformed when π is fixed. For this distribution shift scenario, it therefore seems that a fixed π is neither informative enough as a proposal for LURE, nor does it allow good ASE estimates. This justifies our focus on *adaptive* approaches for active testing.

B.3 Additional Acquisition Strategies for ASEs

For completeness, we here also investigate the performance for ASE when performing *stochastic* acquisition from the XWED and BALD scores. These are the exact acquisition strategies for which LURE suffered high variance in Figure 3. In Figure B.1 (b), we observe that ASE continues to perform well for both these acquisition strategies, although they do not outperform our default strategy that uses deterministic acquisition of the maximal XWED scores.

B.4 Estimating Accuracy with ASEs

ASEs are compatible with arbitrary loss functions $\mathcal{L}$. For 0-1-Loss, $\mathcal{L}(f(x), y) = \mathbf{1}[f(x) = y]$, ASEs can be used to estimate the accuracy of the main model. We perform these experiments in the setup of §7.3. Figure B.2 demonstrates that ASEs continue to outperform all other baselines for the task of accuracy estimation. The accuracies of the main model in these experiments are $(73.31 \pm 0.93)\%$ for CIFAR-100, $(91.07 \pm 0.62)\%$ for CIFAR-10, and $(93.10 \pm 0.53)\%$ for Fashion-MNIST.



(a) (b)

Figure B.1: (a) If π is constant, both ASE and LURE do not outperform naive MC in the distribution shift scenario. This result highlights the importance of the *adaptive* nature of both ASE- and LURE-based active testing. (b) Additional acquisition functions for ASE in the distribution shift scenario: while deterministic acquisition from XWED performs best, ASEs also give good performance when acquiring *stochastically* from XWED or BALD scores. This is unlike LURE, which is highly sensitive to the correct choice of acquisition function. We display means over 100 runs and shading indicates two standard errors (too small to be visible for some of the lines). The vertical grey lines mark iterations at which we retrain the secondary model.

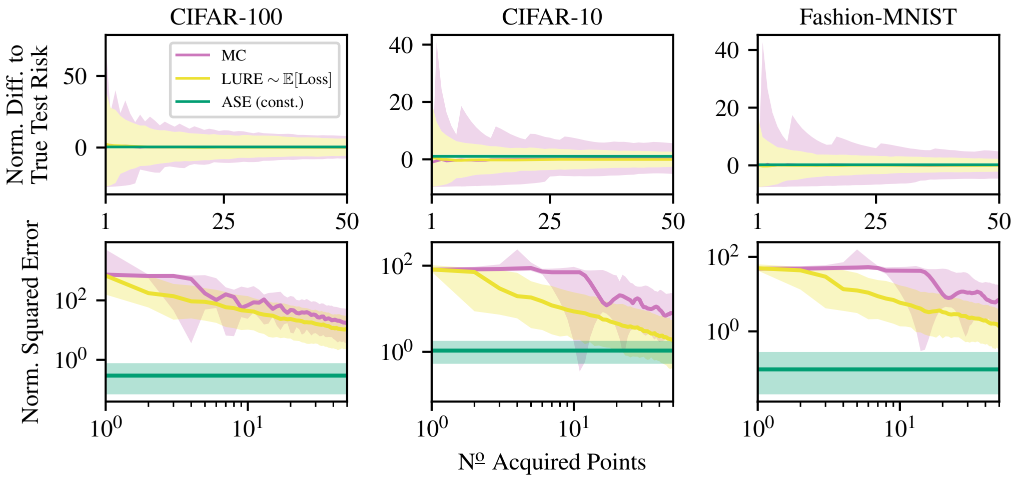


Figure B.2: Variant of the experiments of §7.3 where we estimate the *accuracy* of the main model. We display medians over 1000 random test sets and 10/90 % (top) and 25/75 % (bottom) quantiles (top: too small to be visible for ASE).

B.5 Reduced Training Set Size for Experiments of §7.3

We here investigate a variation of the experiments in §7.3: reducing the size of the training set to 10 000 samples. This reduces the quality of the surrogate model and thus makes the problem more challenging for ASEs. Despite this, Figure B.3 demonstrates that ASEs continue to outperform all baselines. Again, note that the ASE does not use any test labels here and is therefore constant. Retraining of the ASE surrogate on the newly observed test data (with labels) will be necessary for ASEs to be able to continue to improve over the baselines at larger number of acquired test labels.

C Additional Figures

In the main body of the paper, we display both median and mean squared errors across the different figures to provide a variety of perspectives on ASEs. Here, we provide complimentary versions: for figures that show median behavior in the main paper, we here show means, and vice versa.

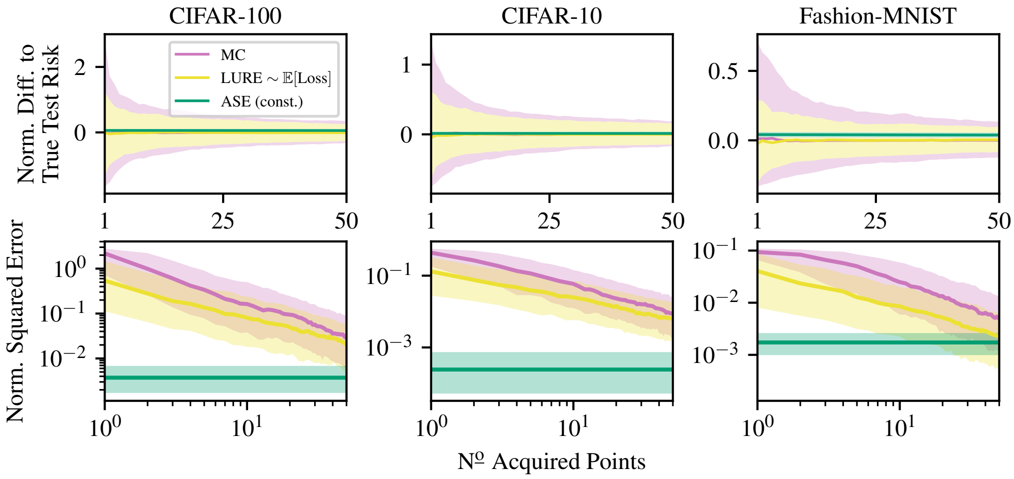


Figure B.3: Variant of the experiments of §7.3 where the training set size is reduced to 10 000 samples. We display medians over 1000 random test sets and 10/90 % (top) and 25/75 % (bottom) quantiles (top: too small to be visible for ASE).

See Figure C.1, Figure C.2, Figure C.3, and Figure C.4 for complimentary versions of the plots from the experimental evaluations in §7 and §B.

For ASE, there are no significant differences between the mean and median behavior. In contrast, LURE occasionally does suffer from high variance (mean behavior), e.g. for CIFAR-10 in Figure C.3. As discussed in the main body of the paper, this happens when the weight assigned by the proposal diverges strongly from the observed outcome for individual observations. This does not happen for ASE because, unlike LURE, we do not weight observations against predictions from the surrogate.

We also include versions of the main figures of the paper that normalize the squared errors of the estimators by the squared true test loss of the main model f on the full test set. See Figure C.5, Figure C.6, and Figure C.7 for these plots. For the experiments in §7.1 and §7.2, the cross-entropy loss of the main model is 1.51 ± 0.18 (average over 100 runs) and the accuracy is $(85.63 \pm 0.25)\%$. For the experiments of §7.3, the cross-entropy losses are 0.961 ± 0.039 for CIFAR-100, 0.289 ± 0.021 for CIFAR-10, and 0.172 ± 0.015 for Fashion-MNIST. The accuracies are 72.54% for CIFAR-100, 90.92% for CIFAR-10, and 92.05% for Fashion-MNIST. (We provide standard deviations on accuracies for the experiments of §B.4.)

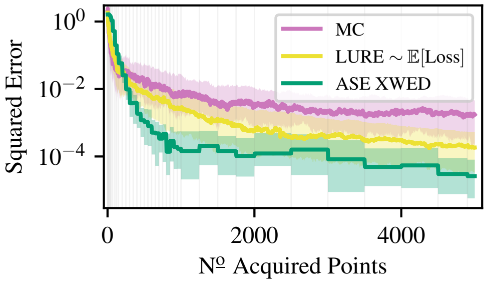


Figure C.1: Version of Figure 1 with medians and quantiles instead of means and standard errors. Convergence of the squared error for the distribution shift experiment. Shown are median squared errors and the shading indicates the 25/75 % quantiles over 100 runs. We see that ASE provides significant improvements over both LURE and MC. The vertical grey lines mark where we retrain the secondary model.

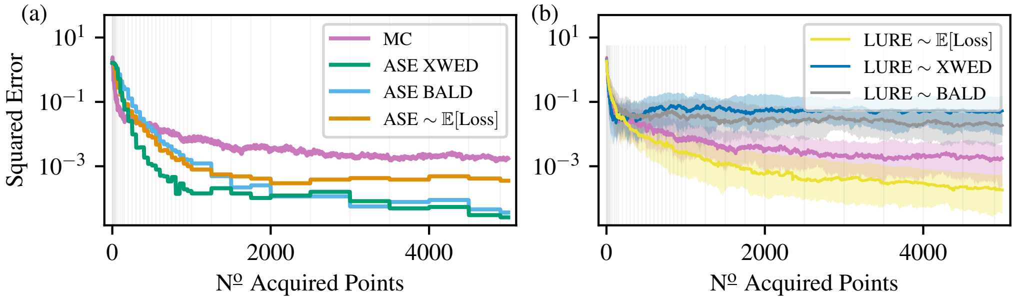


Figure C.2: Version of Figure 3 with median squared errors. (a) While XWED performs best, ASE outperforms MC for all investigated choices of acquisition functions. Shown are median squared errors over 100 runs, and we omit quantile shading to aid legibility. (b) LURE suffers high error when not acquiring from expected loss and instead using a different acquisition strategy. Shown are median squared errors and the shading indicates 25/75 % quantiles over 100 runs. We retrain π at steps marked with grey lines. The '$\sim$'-symbol indicates stochastic acquisition.

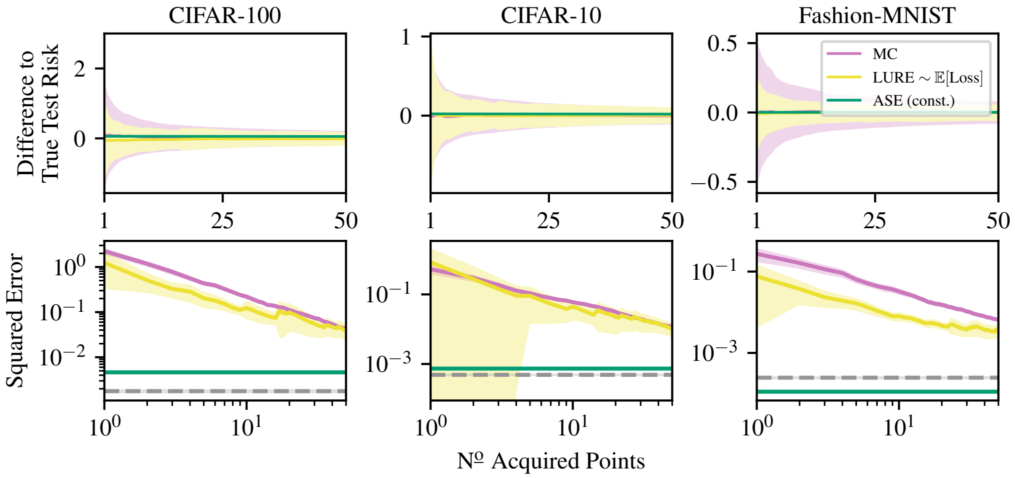


Figure C.3: Version of Figure 4 with means and standard errors. We display means over 1000 random test sets and shading indicates standard deviations for the differences (top, too small to be visible for ASE) and two standard errors for the squared error (bottom, too small to be visible for ASE and naive MC). Active Testing of ResNets on CIFAR-100, CIFAR-10, and FashionMNIST without retraining the auxiliary model π. Despite not observing any test labels, ASE again improves upon LURE. The grey line indicates the lower bound on the error from the finite size of the pool.

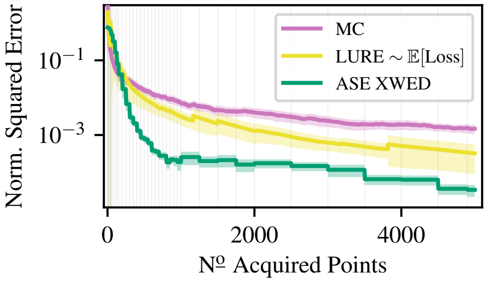
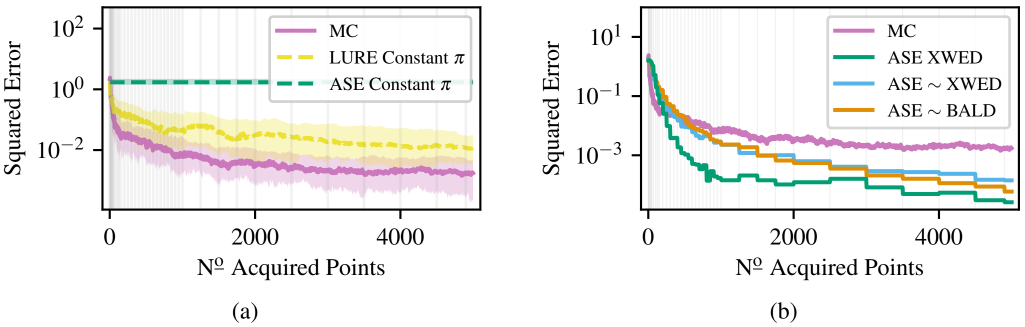


Figure C.5: Version of Figure 1 with squared error of the estimates normalized by the true value of the test error.



(a)

(b)

Figure C.4: (a) Version of Figure B.1 (a) with medians and quantiles instead of means and standard errors. Convergence of the squared error for the distribution shift experiment with constant secondary model π. Shown are median squared errors and the shading indicates the 25/75 % quantiles over 100 runs. Again, we see naive MC is not outperformed if π is constant. (b) Version of Figure B.1 (b) with medians instead of means. Comparison of *stochastic* acquisition strategies for ASE on the distribution shift experiment. Shown are median squared errors over 100 runs and we omit quantiles for reasons of legibility. The vertical grey lines mark where we retrain the secondary model.

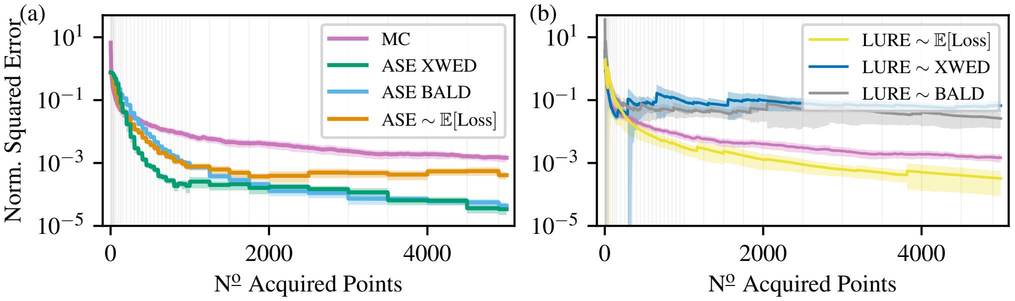


Figure C.6: Version of Figure 3 with squared error of the estimates normalized by the true value of the test error.

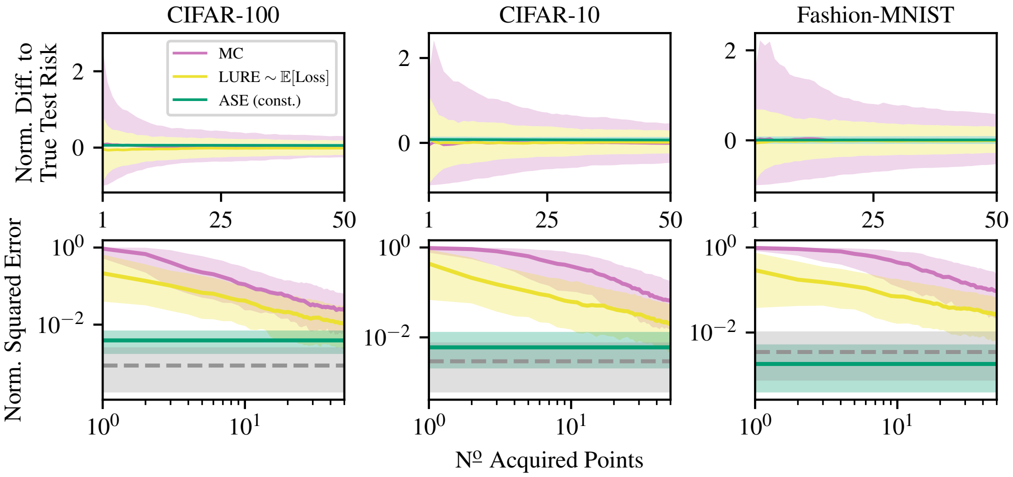


Figure C.7: Version of Figure 4 with differences and squared errors of the estimates normalized by the true value of the test error.

D Experimental Details

D.1 Experiment Definition

In this section, we provide a more detailed explanation of how we combine different estimators and acquisition functions in the experimental evaluation in §7. For our purposes, there are two ingredients to a method for model evaluation: the estimator and the acquisition strategy.

In terms of estimators, this paper compares the naive MC estimator, $\hat{r}_{\text{iid}}$, introduced in §2, and the LURE estimator, $\hat{r}_{\text{LURE}}$, Eq. (2), against the ASE estimator, $\hat{r}_{\text{ASE}}$ Eq. (4).

For the MC estimator, the acquisition of labels from the remaining samples in the test pool is always uniformly at random, so in the figures in §7, we simply write "MC" to denote this naive baseline. For any other acquisition scheme, $\hat{r}_{\text{iid}}$ would no longer be unbiased. However, for ASE and LURE, we have the option to use a variety of acquisition functions a. In this paper, we consider XWED, Eq. (6), BALD, Eq. (5), and $\mathbb{E}[\text{Loss}]$, Eq. (3) as acquisition functions. For all of the these acquisition functions, at any iteration m, we evaluate the acquisition function on all remaining samples $\mathcal{U}$ in the test pool, obtaining a set of acquisition scores $\{a(x_i)\}_{i\in\mathcal{U}}$.

Given these scores, there are two ways in which we can determine the next acquisition: deterministic and stochastic acquisition. In deterministic acquisition, we acquire a label for the sample x_{i_m} remaining in the test pool with the largest acquisition score $i_m = \arg\max_i a(x_i)$. For stochastic acquisition, we choose the next point to acquire by sampling from the distribution over acquisition scores: we define a probability mass function over the indices as $p(i) = a(x_i)/\sum_{i\in\mathcal{U}} a_i$ and then sample an index $i_m \sim p(i)$ for acquisition. For the LURE estimator, acquisition needs to be stochastic for the importance sampling estimator to remain unbiased, while ASEs are compatible with both stochastic and deterministic acquisition.

In the experiments in §7, we add a "$\sim$" prefix for experiments with stochastic acquisition, and we add no prefix for deterministic acquisition. For example, "LURE $\sim$ BALD" is an experiment that uses the LURE estimator and stochastically samples acquisitions from the BALD scores, while "ASE XWED" denotes a run that uses the ASE estimator and deterministically acquires labels for samples with the highest XWED score in each round.

D.2 Computing XWED and Related Quantities

To give a practical example of how to compute some of the quantities related to the ASE and active testing, we now present a worked through example for the case in which the main model f is a neural network and the surrogate π a deep ensemble, both trained for a classification task.

Here, both $f(x)$ and $\pi(y|x)$ are classifiers whose outputs are vectors of dimensionality C, where C denotes the number of classes. Each element i of these vectors, represents the predicted probability that the observed class y is equal to i. The quantity we wish to estimate is the average negative log-likelihood on the test set, also known as the cross-entropy loss of the test set. Our loss function is thus $\mathcal{L}(y, f(x)) = -\log f(x)_y$, i.e. the negative log probability the model f assigns to the true class.

For classification tasks, we can evaluate expectations with respect to outcomes Y exactly by enumerating all possibilities. Further, for the deep ensemble surrogate π, we model the posterior distribution of the deep ensemble surrogate parameters as $p(\theta) = \frac{1}{E}\sum_{e=1}^{E} \delta(\theta - \theta_e)$, where δ is the Dirac delta and $e \in \{1, \dots, E\}$ enumerates the ensemble components. Together, this simplifies the XWED acquisition function to simple sums over class labels and ensemble components:

$$\text{XWED}(x) = \sum_{y=1}^{C} -\pi(y|x)\mathcal{L}(y, f(x)) \log \pi(y|x) + \frac{1}{E}\sum_{e=1}^{E} \pi(y|x, \theta_e)\mathcal{L}(y, f(x)) \log \pi(y|x, \theta_e)$$

where, lastly, we note that a posterior predictive distribution for deep ensembles can be obtained as the average prediction of the ensemble components,

$$\pi(y|x) = \int \pi(y|x,\theta)\pi(\theta)\,\mathrm{d}\theta = \frac{1}{E}\sum_{e=1}^{E}\int \pi(y|x,\theta)\delta(\theta-\theta_e)\,\mathrm{d}\theta = \frac{1}{E}\sum_{e=1}^{E}\pi(y|x,\theta_e).$$

D.3 Training Setup

Evaluation with Distribution Shift. We follow Kossen et al. [38] in the setup of the radial BNN, using code and default hyperparameters provided by Farquhar et al. [18]: we employ a learning rate of 1×10^{-4} and set the weight decay to 1×10^{-4}, in combination with the ADAM optimizer [35] and a batch size of 64. We use 8 variational samples during training and 100 variational samples during testing of the BNN. We use convolutional layers with 16 channels. We train for a maximum of 500 epochs with early stopping patience of 10, and we use validation sets of size 500. We further use the validation set to calibrate predictions via temperature scaling.

Evaluation of CNNs for Classification. We follow Kossen et al. [38], who follow DeVries & Taylor [14], for the setup of the ResNet-18 for the CIFAR-10 and Fashion-MNIST datasets. Concretely, we set the learning rate to 0.1, the weight decay 5×10^{-4}, and use momentum of 0.9 with an SGD optimizer and batch size of 128. We anneal the learning rate using a cosine schedule. We use early stopping on a validation set of size 5000, with patience of 20 epochs, and we set the maximum number of epochs to 160. We use the validation set to calibrate predictions via temperature scaling.

Similarly, we follow Kossen et al. [38] for the setup of the WideResNet on CIFAR-100: we set the depth to 40 and again follow DeVries & Taylor [14] to set hyperparameters. Specifically, we train for 200 epochs, setting the learning rate to 0.1, the weight decay to 5×10^{-4}, and the momentum to 0.9. We use the SGD optimizer with a batch size of 128. We decrease the initial learning rate by a factor of 0.2 after 60, 120, and 160 epochs. Again, we use the validation set to calibrate predictions via temperature scaling.

E Computational Complexity

The computational complexity of ASEs can be split into two components: the computational complexity of the ASE estimate and the computational complexity of the acquisition process.

The ASE estimate requires us to predict with the surrogate model on each of the N points in the test pool, cf. Eq. (4), leading to a computational complexity of $\mathcal{O}(N)$.

During acquisition, at each iteration m, we need evaluate the acquisition function on all $N - m$ remaining samples in the pool. This is repeated for all M iterations leading to an overall computational complexity of $\mathcal{O}(MN)$. Additionally, we have to consider the computational complexity of retraining the surrogate model between iterations. We assume, for worst case complexity, that the surrogate is retrained in each of the M iterations. Further, we assume that the complexity of training the surrogate is proportional to the size of the dataset. If we assume an initial training set of size $|\mathcal{D}_{\text{train}}|$ for the surrogate, then in each testing iteration m, we incur additional computational complexity of $\mathcal{O}(m + |\mathcal{D}_{\text{train}}|)$, or $\mathcal{O}(M \cdot (M + |\mathcal{D}_{\text{train}}|))$ over all iterations.

In total, this gives the computational complexity of ASEs as $\mathcal{O}\left(N + M \cdot (N + M + |\mathcal{D}_{\text{train}}|)\right)$. Given that $M \leq N$ always, this can be simplified to $\mathcal{O}\left(M \cdot (N + |\mathcal{D}_{\text{train}}|)\right)$.

Similarly, Kossen et al. [38] derive the computational complexity of active testing with LURE as $\mathcal{O}(M \cdot (N + |\mathcal{D}_{\text{train}}|)))$. Therefore both ASE- and LURE-based active testing have identical big-$\mathcal{O}$ complexity. Trivially, the naive Monte Carlo baseline simply has a much lower complexity of $\mathcal{O}(M)$ than either active testing approaches. We note that, while naive Monte Carlo is cheaper for given number of acquisitions M, the expense of ASEs and LURE is justified in the active testing scenario by the assumption that these labels are expensive to acquire. Note here that the *labelling* cost of ASE and LURE is exactly the same as MC (and thus $\mathcal{O}(M)$).

F Societal Impact

With ASEs, we present a general framework for the evaluation of machine learning models. It is therefore hard to evaluate the societal impact of ASEs without restricting the discussion to one of many possible areas of application. As with any application of machine learning in critical contexts, we advocate that the alignment of model predictions and societal values is thoroughly evaluated. Below, we present some more specific thoughts we hope may be useful when considering the societal impact of ASEs in a specific application.

Biases in Surrogate Predictions. Estimates from ASE are inherently subjective as they rely directly on the predictions of the surrogate model. Because of this, decisions based on estimates from ASE should be more carefully monitored than those based on MC. In particular, one usually would only have to consider potential unfairness/societal bias in the original model, but for ASE these also need to be considered as part of the estimation itself.

The Definition of 'Expensive'. ASEs assume that label acquisition is more expensive than the computational cost required to iteratively train and retrain the surrogate model. Depending on the application, the goals of efficient evaluation and the goals of society may or may not be aligned. We could, for example, assume that the ASE user cares about saving money, while society cares about the CO2 footprint of the approach. In some applications, labels might be really expensive in terms of their monetary price but might not require much CO2, e.g. expert annotations in medical applications. In this case, the goals of the ASE user and society might not be aligned because society would prefer more expert queries rather than CO2-intensive retraining of large surrogate models. On the other hand, labels might be expensive *and* CO2-intensive, e.g. when the function is a computationally demanding numerical simulation. In this case, the goals of society and the ASE user may be aligned because being efficient in labels benefits both. Note that this discussion similarly applies to both active testing and active learning more generally.

G Compute Details, Licenses, and Resource Citations

Compute Details. All experiments were run on desktop machines in an internal cluster with either a NVIDIA GeForce RTX 2080 Ti (12 GB memory) or a NVIDIA Titan RTX (24 GB memory) graphics card.

A single run for the experiment of §7.1 and §7.2 takes about 8 hours on a single machine. For the §7.3, the experiments with the Resnet-18 ensemble on Fashion-MNIST take about 9 hours to compute, and 14 hours for the Resnet-18 ensemble on CIFAR-10. The results for the WideResNet ensemble on CIFAR-100 can be computed in 24 hours, if ensemble members of π are trained in parallel across different GPUs.

License Agreements and Resource Citations.
License agreement of the MNIST dataset [42]: License: Yann LeCun and Corinna Cortes hold the copyright of MNIST dataset, which is a derivative work from original NIST datasets. MNIST dataset is made available under the terms of the Creative Commons Attribution-Share Alike 3.0 license.

License agreement of the Fashion-MNIST dataset [69]: Fashion-Mnist published under MIT license.

License agreement of the CIFAR-10 and CIFAR-100 dataset [39]: CIFAR-10 and CIFAR-100 are published under MIT license.

License agreement of Python [66]: Python is dual licensed under the PSF License Agreement and the Zero-Clause BSD license.

License agreement of PyTorch [52]: PyTorch is licensed under a modified BSD license, see https://github.com/pytorch/pytorch/blob/master/LICENSE.