ON STATISTICAL BIAS IN ACTIVE LEARNING: HOW AND WHEN TO FIX IT

Sebastian Farquhar[†*], **Yarin Gal**[†], **Tom Rainforth**[‡*]
[†] OATML, Department of Computer Science; [‡] Department of Statistics
University of Oxford

ABSTRACT

Active learning is a powerful tool when labelling data is expensive, but it introduces a bias because the training data no longer follows the population distribution. We formalize this bias and investigate the situations in which it can be harmful and sometimes even helpful. We further introduce novel corrective weights to remove bias when doing so is beneficial. Through this, our work not only provides a useful mechanism that can improve the active learning approach, but also an explanation of the empirical successes of various existing approaches which ignore this bias. In particular, we show that this bias can be actively helpful when training overparameterized models—like neural networks—with relatively little data.

1 INTRODUCTION

In modern machine learning, unlabelled data can be plentiful while labelling requires scarce resources and expert attention, for example in medical imaging or scientific experimentation. A promising solution to this is active learning—picking the most informative datapoints to label that will hopefully let the model be trained in the most sample-efficient way possible (Atlas et al., 1990; Settles, 2010).

Active learning has a complication. By picking the most informative labels, the acquired dataset is *not* drawn from the population distribution. This sampling bias, noted by e.g., MacKay (1992) and Dasgupta & Hsu (2008), is worrying: key results in machine learning depend on the training data being identically and independently distributed (i.i.d.) samples from the population distribution. For example, we train neural networks by minimizing a Monte Carlo estimator of the population risk. If training data are actively sampled, that estimator is biased and we optimize the wrong objective. The possibility of bias in active learning has been considered by e.g., Beygelzimer et al. (2009); Chu et al. (2011); Ganti & Gray (2012), but the full problem is not well understood. Methods that remove active learning bias have been restricted to special cases, so it has been impossible to even establish whether removing active learning bias is helpful or harmful in typical situations.

To this end, we show how to remove the bias introduced by active learning with minimal changes to existing active learning methods. As a stepping stone, we build a Plain Unbiased Risk Estimator, $\tilde{R}_{\text{PURE}}$, which applies a corrective weighting to actively sampled datapoints in pool-based active learning. Our Levelled Unbiased Risk Estimator, $\tilde{R}_{\text{LURE}}$, builds on this and has lower variance and additional desirable finite-sample properties. We prove that both estimators are unbiased and consistent for arbitrary functions, and characterize the variance of the estimators.

Having shown how to remove the bias introduced by active learning, we show that this can sometimes lead to significant improvements. The benefits are most pronounced in underparameterized models where each datapoint affects the learned function globally. For example, in linear regression adopting our weighting allows better estimates of the parameters with less data. But in cases where the model is overparameterized and datapoints mostly affect the learned function locally—like neural networks—we find that correcting active learning bias can be ineffective or even harmful. This explains the existing success of active learning with neural networks, which almost entirely ignores the bias from active sampling (e.g., (Gal et al., 2017b; Shen et al., 2018) and others in Appendix D).

Risk estimator bias is especially important for model evaluation. Perhaps because of this, active learning has historically been used only for training. However, test labels cost as much as training

[*]Equal contribution. Corresponding author `sebastian.farquhar@cs.ox.ac.uk`.

labels, despite active learning works often relying on large test and validation sets (Haut et al., 2018; Sinha et al., 2019). Our estimators allow active sampling for *model selection* also and thus provide a means of avoiding this problem by labeling test data in an efficient manner without introducing bias.

Our main contributions are:

1. We offer a formalization of the problem of statistical bias in active learning.
2. We introduce active learning risk estimators, $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{PURE}}$, and prove both are unbiased, consistent, and with variance that can be less than the naive (biased) estimator.
3. Using these, we show that active learning bias can hurt in underparameterized cases like linear regression but help in overparameterized cases like neural networks and explain why.

2 Bias in Active Learning

We begin by characterizing the bias introduced by active learning. In supervised learning, generally, we aim to find a decision rule f_θ corresponding to inputs, $\mathbf{x}$, and outputs, y, drawn from a population data distribution $p_{\text{data}}(\mathbf{x}, \mathrm{y})$ which, given a loss function $\mathcal{L}(\mathrm{y}, f_\theta(\mathbf{x}))$, minimizes the population risk:

$$r = \mathbb{E}_{\mathbf{x},\mathrm{y}\sim p_{\text{data}}}\left[\mathcal{L}(\mathrm{y}, f_\theta(\mathbf{x}))\right].$$

The population risk cannot be found exactly, so instead we consider the empirical distribution for some dataset of N points drawn from the population. This gives the empirical risk: an unbiased and consistent estimator of r when the data are drawn i.i.d from p_{data} and are independent of θ,

$$\hat{R} = \frac{1}{N}\sum\nolimits_{n=1}^{N} \mathcal{L}(\mathrm{y}_n, f_\theta(\mathbf{x}_n)).$$

In pool-based active learning (Lewis & Gale, 1994; Settles, 2010), we begin with a large unlabelled dataset, known as the pool dataset $\mathcal{D}_{\text{pool}} \equiv \{\mathbf{x}_n | 1 \leq n \leq N\}$, and sequentially pick the most useful points for which to acquire labels. The lack of most labels means we cannot evaluate $\hat{R}$ directly, so we use the *sub-sample empirical risk* evaluated using the M actively sampled labelled points:

$$\tilde{R} = \frac{1}{M}\sum\nolimits_{m=1}^{M} \mathcal{L}(\mathrm{y}_m, f_\theta(\mathbf{x}_m)). \tag{1}$$

Though almost all active learning research uses this estimator (see Appendix D), it is not an unbiased estimator of either $\hat{R}$ or r when the M points are actively sampled. Under active—i.e. non–uniform—sampling the M datapoints are not drawn from the population distribution, resulting in a bias which we formally characterize in §5. See Appendix A for a more general overview of active learning.

Note an important distinction between what we will call "statistical bias" and "overfitting bias." The bias from active learning above is a statistical bias in the sense that using $\tilde{R}$ biases our estimation of r, regardless of θ. As such, optimizing θ with respect to $\tilde{R}$ induces bias into our optimization of θ. In turn, this breaks any consistency guarantees for our learning process: if we keep M/N fixed, take $M \to \infty$, and optimize for θ, we no longer get the optimal θ that minimizes r. Almost all work on active learning for neural networks currently ignores the issue of statistical bias.

However, even without this statistical bias, indeed even if we use $\hat{R}$ directly, the training process itself also creates an overfitting bias: evaluating the risk using training data induces a dependency between the data and θ. This is why we usually evaluate the risk on held-out test data when doing model selection. Dealing with overfitting bias is beyond the scope of our work as this would equate to solving the problem of generalization. The small amount of prior work which does consider statistical bias in active learning entirely ignores this overfitting bias without commenting on it.

In §3-7, we focus on statistical bias in active learning, so that we can produce estimators that are valid and consistent and let us optimize the intended objective, not so they can miraculously close the train–test gap. From a more formal perspective, our results all assume that θ is chosen independently of the training data; an assumption that is almost always (implicitly) made in the literature. This ensures our estimators form valid objectives, but also has important implications that are typically overlooked. We return to this in §8, examining the interaction between statistical and overfitting bias.

3 Unbiased Active Learning: $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$

We now show how to unbiasedly estimate the risk in the form of a weighted expectation over actively sampled data points. We denote the set of actively sampled points $\mathcal{D}_{\text{train}} \equiv \{(\mathbf{x}_m, \mathrm{y}_m) | 1 \leq m \leq M\}$,

where $\forall m : \mathbf{x}_m \in \mathcal{D}_{\text{pool}}$. We begin by building a "plain" unbiased risk estimator, $\tilde{R}_{\text{PURE}}$, as a stepping stone—its construction is quite natural in that each term is an individually unbiased estimator of the risk. We then use it to construct a "levelled" unbiased risk estimator, $\tilde{R}_{\text{LURE}}$, which is an unbiased and consistent estimator of the population risk just like $\tilde{R}_{\text{PURE}}$, but which reweights individual terms to produce lower variance and resolve some pathologies of the first approach. Both estimators are easy to implement and have trivial compute/memory requirements.

3.1 $\tilde{R}_{\text{PURE}}$: Plain Unbiased Risk Estimator

For our estimators, we introduce an active sampling proposal distribution over *indices* rather than the more typical distribution over datapoints. This simplifies our proofs, but the two are algorithmically equivalent for pool-based active learning because of the one–to–one relationship between datapoints and indices. We define the probability mass for each index being the next to be sampled, once $\mathcal{D}_{\text{train}}$ contains $M-1$ points, as $q(i_M; i_{1:M-1}, \mathcal{D}_{\text{pool}})$. Because we are learning actively, the proposal distribution depends on the indices sampled so far ($i_{1:M-1}$) and the available data ($\mathcal{D}_{\text{pool}}$). The only requirement on this proposal distribution for the proofs in this section is that it must place non-zero probability on all of the training data: anything else necessarily introduces bias. Considerations for the acquisition proposal distribution are discussed further in §4 We first present the estimator before proving the main results:

$$\tilde{R}_{\text{PURE}} \equiv \frac{1}{M} \sum\nolimits_{m=1}^{M} a_m; \quad \text{where} \quad a_m \equiv w_m \mathcal{L}_{i_m} + \frac{1}{N} \sum\nolimits_{t=1}^{m-1} \mathcal{L}_{i_t},$$

where the loss at a point $\mathcal{L}_{i_m} \equiv \mathcal{L}(\mathrm{y}_{i_m}, f_\theta(\mathbf{x}_{i_m}))$, the weights $w_m \equiv 1/Nq(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})$ and $i_m \sim q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})$. For practical implementation, $\tilde{R}_{\text{PURE}}$ can further be written in the following more computationally friendly form that avoids a double summation:

$$\tilde{R}_{\text{PURE}} = \frac{1}{M} \sum_{m=1}^{M} \left(\frac{1}{Nq(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})} + \frac{M-m}{N} \right) \mathcal{L}_{i_m},$$

but we focus on the first form for our analysis because a_m has some beneficial properties not shared by the above that we can exploit. In particular, in Appendix B.1 we prove that:

Lemma 1. *The individual terms a_m of $\tilde{R}_{PURE}$ are unbiased estimators of the risk: $\mathbb{E}[a_m] = r$.*

The motivation for the construction of a_m directly originates from constructing an estimator where Lemma 1 holds while only using the $i_1, \ldots, i_m$ datapoints, taking care with the fact that each new proposal distribution does not have support over points that have already been acquired. Except for trivial solutions, it is unique in this regard; naive importance sampling (i.e. $\frac{1}{M}\sum_{m=1}^{M} w_m \mathcal{L}_{i_m}$) does not lead to an unbiased, or even consistent, estimator. However, the overall estimator $\tilde{R}_{\text{PURE}}$ is not the only unbiased estimator of the risk, as we discuss in §3.2.

We can now characterize the behaviour of $\tilde{R}_{\text{PURE}}$ as follows (see Appendix B.2 for proof)

Theorem 1. *$\tilde{R}_{PURE}$ as defined above has the properties:*

$$\mathbb{E}\left[\tilde{R}_{PURE}\right] = r, \qquad \text{Var}[\tilde{R}_{PURE}] = \frac{\sigma^2}{M}, \qquad \textit{where}$$

$$\sigma^2 = \frac{1}{M} \sum_{m=1}^{M} \text{Var}[a_m] = \frac{\text{Var}\left[\mathcal{L}(\mathrm{y}, f_\theta(\mathbf{x}))\right]}{N} + \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}} \left[\text{Var}\left[w_m \mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right]. \tag{2}$$

Remark 1. *The first term of* (2) *is the variance of the loss on the whole pool, while the second term accounts for the interaction between sampled points and $\mathcal{D}_{\text{pool}}$. As such, the second term, which is $O(N)$ times larger, will always dominate in practice.*

Armed with Theorem 1, we can prove the consistency of $\tilde{R}_{\text{PURE}}$ under standard assumptions: $\tilde{R}_{\text{PURE}}$ converges in expectation (i.e. its mean squared error tends to zero) as $M \to \infty$ under the assumptions that $N > M$, $\mathcal{L}(\mathrm{y}, f_\theta(\mathbf{x}))$ is integrable, and $q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})$ is a valid proposal in the sense that it puts non-zero mass on each unlabelled datapoint. Formally, as proved in Appendix B.3,

Theorem 2. *Let $\alpha = N/M$ and assume that $\alpha > 1$. If $\mathbb{E}\left[\mathcal{L}(\mathrm{y}, f_\theta(\mathbf{x}))^2\right] < \infty$ and*

$$\exists \beta > 0 \ : \ \min_{n \in \{1:N \setminus i_{1:m-1}\}} q(i_m = n; i_{1:m-1}, \mathcal{D}_{\text{pool}}) \geq \beta/N \quad \forall N \in \mathbb{Z}^+, m \leq N,$$

then $\tilde{R}_{PURE}$ converges in its L^2 norm to r as $M \to \infty$, i.e., $\lim_{M\to\infty} \mathbb{E}\left[\left(\tilde{R}_{PURE} - r\right)^2\right] = 0$.

3.2 $\tilde{R}_{\text{LURE}}$: Levelled Unbiased Risk Estimator

$\tilde{R}_{\text{PURE}}$ is natural in that each term is an unbiased estimator of r. However, this creates surprising behaviour given the sequential structure of active learning. For example, with a uniform proposal distribution—equivalent to not actively learning—points sampled earlier are more highly weighted than later ones and $\tilde{R}_{\text{PURE}} \neq \tilde{R}$. Specifically, a uniform proposal, $q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}}) = \frac{1}{N-m+1}$, gives a weight on each sampled point of $1 + \frac{M-2m+1}{N} \neq 1$. Similarly, as $M \to N$ (such that we use the full pool) the weights also fail to become uniform: setting $M = N$ gives a weight for each point of $1 + \frac{M-2m+1}{N} \neq 1$. $\tilde{R}_{\text{LURE}}$ fixes this. We first quote the estimator before proving key results:

$$\tilde{R}_{\text{LURE}} \equiv \frac{1}{M} \sum_{m=1}^{M} v_m \mathcal{L}_{i_m} \ ; \quad v_m \equiv 1 + \frac{N-M}{N-m} \left(\frac{1}{(N-m+1)\, q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})} - 1 \right). \tag{3}$$

This estimator ensures that the expected value of the weight, v_m, does not depend on the *position* it was sampled in, but only on the probability with which it was sampled. That is, $\mathbb{E}\left[v_m\right] = 1$ for all m, M,N, and $q(i_m; i_{1:m-1}; \mathcal{D}_{\text{pool}})$. As a consequence the variance is generally lower. Moreover, we resolve the finite-sample behaviour shown by $\tilde{R}_{\text{PURE}}$. The weights become more even as M increases for a given N, and when $M = N$, each $v_m = 1$ such that $\tilde{R}_{\text{LURE}} = \tilde{R} = \hat{R}$. Additionally, if the proposal is uniform, all weights are always exactly 1 such that $\tilde{R}_{\text{LURE}} = \tilde{R}$.

To derive $\tilde{R}_{\text{LURE}}$ note that each a_m estimates r without bias so for any normalized linear combination:

$$\mathbb{E}\left[\frac{\sum_{m=1}^{M} c_m a_m}{\sum_{m=1}^{M} c_m}\right] = r,$$

provided that the c_m are constant with respect to the data and sampled indices (they can depend on M, N, and m). In Appendix B.4 we show that the choice of :

$$c_m = \frac{N(N-M)}{(N-m)(N-m+1)}$$

produces the v_m from (3) and in turn that these v_m have the desired property $\mathbb{E}\left[v_m\right] = 1$, $\forall m \in \{1, \ldots, M\}$. We note also that $\sum_{m=1}^{M} c_m = M$, such that $\tilde{R}_{\text{LURE}} = \frac{1}{M}\sum_{m=1}^{M} c_m a_m$. We further characterise the variance and unbiasedness of $\tilde{R}_{\text{LURE}}$ as follows (see Appendix B.5 for proof)

Theorem 3. *$\tilde{R}_{LURE}$ as defined above has the following properties:*

$$\mathbb{E}\left[\tilde{R}_{LURE}\right] = r, \qquad \text{Var}[\tilde{R}_{LURE}] = \frac{\sigma^2_{\text{LURE}}}{M}, \qquad \textit{where}$$

$$\sigma^2_{\text{LURE}} = \frac{1}{M} \sum_{m=1}^{M} c_m^2 \left(\frac{\text{Var}\left[\mathcal{L}(\mathrm{y}, f_\theta(\mathbf{x})\right]}{N} + \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}} \left[\text{Var}\left[w_m \mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right] \right). \tag{4}$$

The variance of $\tilde{R}_{\text{LURE}}$ is typically less than that of $\tilde{R}_{\text{PURE}}$. Remember that c_m were introduced to ensure that $\mathbb{E}\left[v_m\right]$ are all identically one. Therefore this weighting removes the tendency of $\tilde{R}_{\text{PURE}}$ to overemphasize the earlier samples; essentially increasing the effective sample size by correcting the imbalance. More precisely, note that (2) and (4) differ only in the scaling terms c_m^2. As c_m increases with m while their sum is fixed, namely $\frac{1}{M}\sum_{m=1}^{M} c_m = 1$, they upweight the lower variance later terms and downweight the higher variance earlier terms, such that the overall sum is typically less.

We finish by confirming that $\tilde{R}_{\text{LURE}}$ is a consistent estimator as $M \to \infty$ (proof in Appendix B.6):

Theorem 4. *Under the same assumptions as Theorem 2: $\lim_{M\to\infty} \mathbb{E}\left[\left(\tilde{R}_{LURE} - r\right)^2\right] = 0$.*

4 FROM ACTIVE LEARNING SCHEMES TO PROPOSALS

So far, we have introduced two separable elements of the active learning scheme. §3 introduced risk estimators $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$. The acquisition proposal distribution—$q(i_m|i_{1:m-1}, \mathcal{D}_{\text{pool}})$—has remained quite general. So long as the acquisition proposal puts non-zero mass on all the training data, $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$ are unbiased and consistent as proven above. This is in contrast to the naive risk estimator $\tilde{R}$, for which the choice of proposal distribution affects the bias of the estimator. Using our scheme lets us disentangle the choice of risk estimator from the choice of acquisition proposal. In §5, we will discuss how the properties of the acquisition proposal distribution can however affect the *variance* of the unbiased estimators, potentially resulting in lower variance than the naive biased estimator. Then in §8 we will turn to a third element of active learning schemes: the way in which optimization introduces a bias separately from the choice of risk estimator and proposal distribution. At that point, we will consider a final interaction: where the acquisition proposal distribution depends on a model which has been optimized using one of the risk estimators under consideration.

It is easy to statisfy the requirement for non-zero mass everywhere, even for prior work which selects points deterministically (e.g., Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011) or a geometric heuristic like coreset construction (Sener & Savarese, 2018)). Any scheme, like BALD, that selects the points to acquire by performing an `argmax` operation can instead return a distribution that supports the whole space by substituting a `softmax` operation. Alternatively, a distribution can be constructed analogous to epsilon-greedy exploration. With probability $1 - \epsilon$ we select the deterministically proposed datapoint from any arbitrary acquisition strategy, while with probability ϵ we uniformly select any datapoint. In this way, any existing deterministic active learning scheme can be adapted in a way that allows unbiased risk estimation.

It is possible to use $\tilde{R}_{\text{LURE}}$ and $\tilde{R}_{\text{PURE}}$ with data collected using a proposal distribution that does not fully support the training data, though they will not fully correct the bias here. Namely, if we have a set of points, I, that are ignored by the proposal (i.e. that are assigned zero mass), we can still use $\tilde{R}_{\text{LURE}}$ and $\tilde{R}_{\text{PURE}}$ in the same way but they both introduce the same following bias:

$$\begin{aligned}\mathbb{E}[\tilde{R}^I_{\text{LURE}}] = \mathbb{E}[\tilde{R}^I_{\text{PURE}}] &= \mathbb{E}\left[\mathbb{E}\left[\tilde{R}_{\text{LURE}}\middle|\mathcal{D}_{\text{pool}}\right] - \mathbb{E}\left[\frac{1}{N}\sum_{n\in I}\mathcal{L}_n\middle|\mathcal{D}_{pool}\right]\right] \\ &= r - \mathbb{E}\left[\frac{1}{N}\sum_{n\in I}\mathcal{L}_n\right].\end{aligned}$$

Sometimes this bias will be small and may be acceptable if it enables a desired acquisition scheme, but in general one of the stochastic adaptations described above is likely to be preferable. One can naturally also extend this result to cases where I varies at each iteration of the active learning (including deterministic acquisition strategies), for which we again have a non–zero bias.

5 UNDERSTANDING THE EFFECT OF $\tilde{R}_{\text{LURE}}$ AND $\tilde{R}_{\text{PURE}}$ ON VARIANCE

In order to show that the variance of our unbiased estimators can be *lower* than that of the biased $\tilde{R}$, with a well-chosen acquisition function, we first introduce an analogous result to Theorems 1 and 3 for $\tilde{R}$, the proof for which is given in Appendix B.7:

Theorem 5. *Let $\mu_m := \mathbb{E}\left[\mathcal{L}_{i_m}\right]$ and $\mu_{m|i,\mathcal{D}} := \mathbb{E}\left[\mathcal{L}_{i_m}|i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]$. For $\tilde{R}$ (defined in* (1)*):*

$$\begin{aligned}\mathbb{E}\left[\tilde{R}\right] &= \frac{1}{M}\sum\nolimits_{m=1}^{M}\mu_m \quad (\neq r \textit{ in general}) \\ \text{Var}[\tilde{R}] &= \frac{1}{M^2}\sum\nolimits_{m=1}^{M}\Bigg(\overbrace{\text{Var}\left[\mathcal{L}(\mathbf{y}, f_\theta(\mathbf{x}))\right]/N}^{1} + \overbrace{\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[\mathcal{L}_{i_m}|i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right]}^{2} \\ &\quad + \underbrace{\text{Var}[\mu_{m|i,\mathcal{D}} - \hat{R}]}_{3} + \underbrace{2\,\text{Cov}(\hat{R}, \mu_{m|i,\mathcal{D}} - \hat{R})}_{4} + 2\sum\nolimits_{k=1}^{m-1}\underbrace{\text{Cov}\left(\mathcal{L}_{i_m}, \mathcal{L}_{i_k}\right)}_{5}\Bigg).\end{aligned} \tag{5}$$

Examining this expression shows that the variance of $\tilde{R}_{\text{PURE}}$ can be either larger or smaller than $\tilde{R}$, depending on the the proposal distribution and dataset. Consider the terms of (5): 1 and 2 are

analagous to terms of the variance of $\tilde{R}_{\text{PURE}}$ in (2). Recall that ① is smaller than ② by $\mathcal{O}(N)$ in any case, so set it aside. ② is the expected conditional variance of $\mathcal{L}_{i_m}$, while the $\tilde{R}_{\text{PURE}}$ analogue has at the weighted loss $w_i \mathcal{L}_{i_m}$. An effective proposal distribution should select surprising/informative points—with above average losses—with high probability—low w_m—resulting in anticorrelation between $\mathcal{L}_{i_m}$ and w_m. This means the expected conditional variance of $\mathcal{L}_{i_m}$ should be greater than that of $w_i \mathcal{L}_{i_m}$ for a well-designed acquisition strategy.

Now consider the terms only in (5) with no analogue to the previous equations. ③ is always positive, so it contributes to $\tilde{R}$ having higher variance, and comes from the variability in the bias itself in index sampling. ④ expresses the correlation between the bias and the value of the pool estimate itself. It can be positive or negative, but is typically small and decreases as N grows because the variability of $\hat{R}$ scales as $O(1/N)$. ⑤ reflects correlations between the losses at different iterations. These correlations have been carefully eliminated in our approaches. For example, the analogous terms for $\tilde{R}_{\text{PURE}}$ is $\text{Cov}(a_m, a_k) = 0$ for $m \neq k$ as we show in the proof of Theorem 1; the analogous term for $\tilde{R}_{\text{LURE}}$, $\text{Cov}(c_m a_m, c_m a_k)$, is similarly zero. However, ⑤ is hard to quantify for $\tilde{R}$: it can be positive or negative depending on the acquisition function and dataset. For some acquistition strategies, it is conceivable that this term may not converge to zero even in the limit of large N. Nonetheless, we note that when this term is large, such that it might affect which estimator has higher variance, the correlation between losses in different acquired points would lead to high bias in $\tilde{R}$.

In contrast, we prove in Appendix B.8 that under an optimal proposal distribution both $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$ become exact estimators of the empirical risk for any number of samples M—such that they will inevitably have lower variance than $\tilde{R}$ in this case. A similar result holds when we are estimating gradients of the loss, though note that the optimal proposal is different in the two cases.

Theorem 6. *Given a non-negative loss, the optimal proposal distribution*

$$q^*(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}}) = \mathcal{L}_{i_m} / \Sigma_{n \notin i_{1:m-1}} \mathcal{L}_n$$

yields estimators exactly equal to the pool risk, that is $\tilde{R}_{PURE} = \tilde{R}_{LURE} = \hat{R}$ *almost surely* $\forall M$.

In practice, it is impossible to sample using the optimal proposal distribution. However, we make this point in order to prove that adopting our unbiased estimator is certainly capable of reducing variance relative to standard practice if appropriate acquisition strategies are used. It also provides interesting insights into what makes a good acquisition strategy from the perspective of the risk estimation itself.

6 Related Work

Pool-based active learning (Lewis & Gale, 1994) is useful in cases where input data are prevalent but labeling them is expensive (Atlas et al., 1990; Settles, 2010). The bias from selective sampling was noted by MacKay (1992), but dismissed from a Bayesian perspective based on the likelihood principle. Others have noted that the likelihood principle remains controversial (Rainforth, 2017), and in this case would assume a well-specified model. Moreover, from a discriminative learning perspective this bias is uncontentiously problematic. Lowell et al. (2019) observe that active learning algorithms and datasets become coupled by active sampling and that datasets often outlive algorithms.

Despite the potential for bias, in deep learning it is generally ignored. As an informal survey, we examined the 15 most-cited peer-reviewed papers citing Gal et al. (2017b), which introduced active learning to image data using neural networks. Of these, only two mentioned estimator bias but did not address it while the rest either ignored or were unaware of this problem (see Appendix D). Nguyen et al. (2018) propose active testing but ignore bias introduced by active sampling.

There have been some attempts to address active learning bias, but these have generally required fundamental changes to the active learning approach and only apply to particular setups. Beygelzimer et al. (2009), Chu et al. (2011), and (Cortes et al., 2019) apply importance-sampling corrections (Sugiyama, 2006; Bach, 2006) to *online* active learning. Unlike pool-based active learning, this involves deciding whether or not to sample a new point as it arrives from an infinite distribution. This makes importance-sampling estimators much easier to develop, but as Settles (2010) notes, "*the pool-based scenario appears to be much more common among application papers*". Ganti & Gray (2012) address unbiased active learning in a pool-based setting by sampling from the pool *with replacement*. This effectively converts pool-based learning into a stationary online learning setting, although it overweights data that happens to be sampled early. Sampling with replacement is

unwanted in active learning because it requires retraining the model on duplicate data which is either impossible or wasteful depending on details of the setting. Moreover, they only prove the consistency of their estimator under very strong assumptions (well-specified linear models with noiseless labels and a mean-squared-error loss). Imberg et al. (2020) consider optimal proposal distributions in an importance-sampling setting. Outside the context of active learning, Byrd & Lipton (2019) question the value of importance-weighting for deep learning, which aligns with our findings below.

7 Applying $\tilde{R}_{\text{LURE}}$ and $\tilde{R}_{\text{PURE}}$

We first verify that $\tilde{R}_{\text{LURE}}$ and $\tilde{R}_{\text{PURE}}$ remove the bias introduced by active learning and examine the variance of the estimators. We do this by taking a fixed function whose parameters are independent of $\mathcal{D}_{\text{train}}$ and estimating the risk using actively sampled points. We note that this is equivalent to the problem of estimating the risk of an already trained model in a sample-efficient way given unlabelled test data. We consider two settings: an inflexible model (linear regression) on toy but non-linear data and an overparameterized model (convolutional Bayesian neural network) on a modified version of MNIST with unbalanced classes and noisy labels.

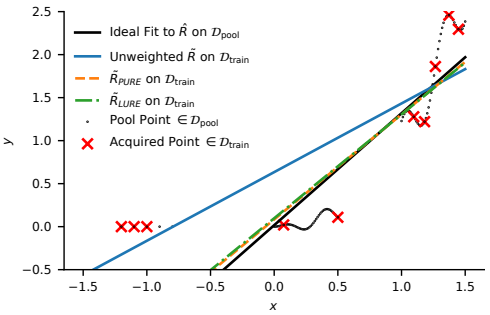
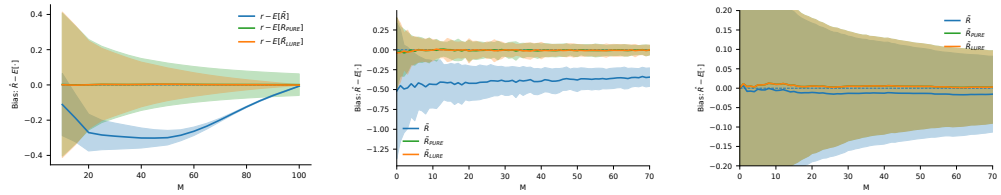


Figure 1: **Illustrative linear regression.** Active learning deliberately over-samples unusual points (red x's) which no longer match the population (black dots). Common practice uses the biased unweighted estimator $\tilde{R}$ which puts too much emphasis on unusual points. Our unbiased estimators $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$ fix this, learning a function using only $\mathcal{D}_{\text{train}}$ nearly equal to the ideal you would get if you had labels for the whole of $\mathcal{D}_{\text{pool}}$, despite only using a few points.

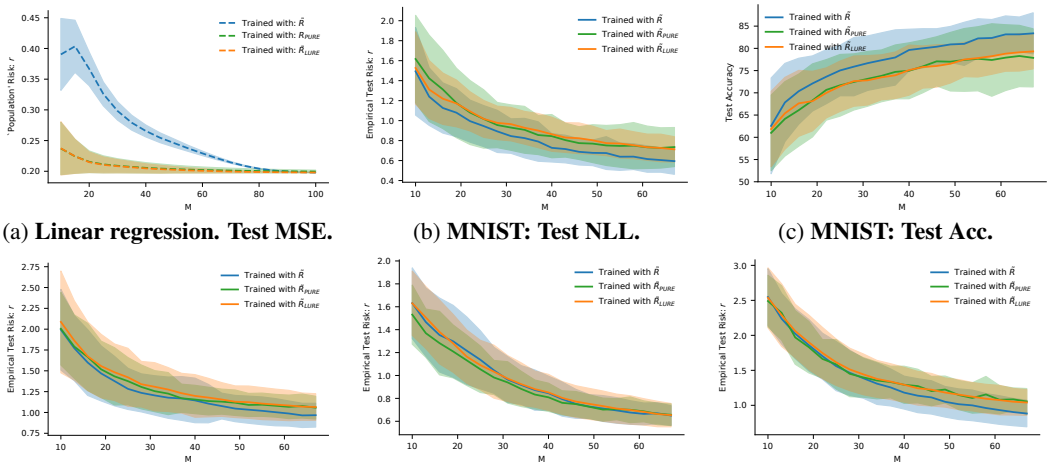
Linear regression. For linear functions, removing active learning bias (ALB), i.e., the statistical bias introduced by active learning, is critical. We illustrate this in Figure 1. Actively sampled points overrepresent unusual parts of the distribution, so a model learned using the unweighed $\mathcal{D}_{\text{train}}$ differs from the ideal function fit to the whole of $\mathcal{D}_{\text{pool}}$. Using our corrective weights more closely approximates the ideal line. The full details of the population distribution and geometric acquisition proposal distribution are in Appendix C.1, where we also show results using an alternative epsilon-greedy proposal. We inspect the ALB in Figure 2a by comparing the estimated risk (with squared error loss and a fixed function) to the corresponding true population risk $\hat{R}$. While $M < N$, the unweighted $\tilde{R}$ is biased (in practice we never have $M = N$ as then actively learning is unnecessary). $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$ are unbiased throughout. However, they have high variance because the proposal is rather poor. Shading represents the std. dev. of the bias over 1000 different acquisition trajectories.

Bayesian Neural Network. We actively classify MNIST and FashionMNIST images using a convolutional Bayesian neural network (BNN) with roughly 80,000 parameters. In Figure 2b and 2c we show that $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$ remove the ALB. Here the variance of $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$ is *lower*



(a) **Linear regression.** (b) **BNN: MNIST.** (c) **BNN: FashionMNIST.**

Figure 2: $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$ remove bias introduced by active learning, while unweighted $\tilde{R}$, which most active learning work uses, is biased. Note the sign: $\tilde{R}$ *overestimates* risk because active learning samples the hardest points. Variance for $\tilde{R}_{\text{PURE}}$ and $\tilde{R}_{\text{LURE}}$ depends on the acquisition distribution placing high weight on high-expected-loss points. In (b), the BALD-style distribution means that the variance of the unbiased estimators is *smaller*. For FashionMNIST, (c), active learning bias is small and high variance in all cases. Shading is ± 1 standard deviation.



(a) **Linear regression. Test MSE.** (b) **MNIST: Test NLL.** (c) **MNIST: Test Acc.**

(d) **FashionMNIST: Test NLL.** (e) **MNIST (MCDO): Test NLL.** (f) **MNIST (Balanced): Test NLL.**

Figure 3: For linear regression, the models trained with $\tilde{R}_{\text{PURE}}$ or $\tilde{R}_{\text{LURE}}$ have lower 'population' risk. In contrast, BNNs trained with $\tilde{R}_{\text{LURE}}$ or $\tilde{R}_{\text{PURE}}$ perform either similarly (e) or slightly worse (b,c,d,f), even though they remove bias and have lower variance. Shading is one standard deviation. For (a) 1000 samples and 'r' estimated on 10,100 points from distribution. For (b)/(c) 45 samples and 'r' estimated on the test dataset. (d)-(f) are alternative settings to validate consistency of result.

or similar to the biased estimator. This is because the acquisition proposal distribution, a stochastic relaxation of the Bayesian Active Learning by Disagreement (BALD) objective (Houlsby et al., 2011), is effective (c.f. §5). A full description of the dataset and procedure is provided in Appendix C.2. Our modified MNIST dataset is unbalanced and has noisy labels, which makes the bias more distorting.

Overall, Figure 2 shows that our estimators remove the bias introduced by active learning, as expected, and can do so with reduced variance given an acquisition proposal distribution that puts a high probability mass on more informative/surprising high-expected-loss points.

Next, we examine the overall effect of using the unbiased estimators to *learn* a model on *downstream performance*. Intuitively, removing bias in training while also reducing the variance ought to improve the downstream task objective: test loss and accuracy. To investigate this, we train models using $\tilde{R}$, $\tilde{R}_{\text{LURE}}$, and $\tilde{R}_{\text{PURE}}$ with actively sampled data and measure the population risk of each model.

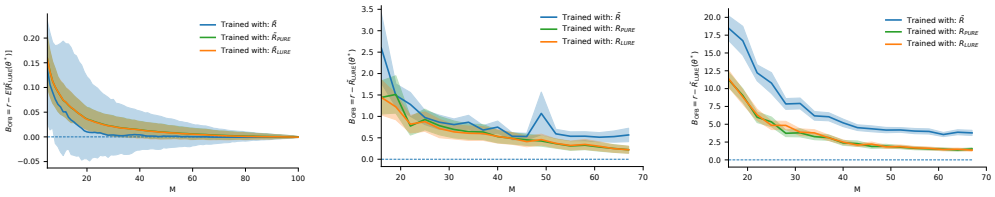
For linear regression (Figure 3a), the new estimators improve the test loss—even with small numbers of acquired points we have nearly optimal test loss (estimated with many samples). However, for the BNN, there is a small but significant negative impact on the full test dataset loss of training with $\tilde{R}_{\text{LURE}}$ or $\tilde{R}_{\text{PURE}}$ (Figure 3b) and a slightly larger negative impact on test accuracy (Figure 3c). That is, we get a better model by training using a biased estimator with higher variance!

To validate this further, we consider models trained instead on FashionMNIST (Fig. 3d), on MNIST but with Monte Carlo dropout (MCDO) (Gal & Ghahramani, 2015) (Fig. 3e), and on a balanced version of the MNIST data (Fig. 3f). In all cases we find similar patterns, suggesting the effects are not overly sensitive to the setting. Further experiments and ablations can be found in Appendix C.2.

8 Active Learning Bias in the Context of Overall Bias

In order to explain the finding that $\tilde{R}_{\text{LURE}}$ hurts training for the BNN, we return to the bias introduced by overfitting, allowing us to examine the effect of removing statistical bias in the context of *overall* bias. Namely, we need to consider the fact that training would induce am overfitting bias (OFB) even if we had not used active learning. If we optimize parameters θ according to $\hat{R}$, then $\mathbb{E}[\hat{R}(\theta^*)] \neq r$, because the optimized parameters θ^* tend to explain training data better than unseen data. Using $\tilde{R}_{\text{LURE}}$, which removes *statistical bias*, we can isolate OFB in an active learning setting. More formally, supposing we are optimizing any of the discussed risk estimators (which we will write using $\tilde{R}_{(\cdot)}$ as a placeholder to stand for any of them) we define the OFB as:

$$B_{\text{OFB}}(\tilde{R}_{(\cdot)}) = r - \tilde{R}_{\text{LURE}}(\theta^*) \quad \text{where} \quad \theta^* = \arg\min_\theta(\tilde{R}_{(\cdot)})$$



(a) **Linear regression.** (b) **BNN: MNIST.** (c) **BNN: FashionMNIST.**

Figure 4: Overfitting bias—B_{OFB}—for models trained using the three objectives. (a) Linear regression, B_{OFB} is small compared to ALB (c.f. Figure 2a). Shading IQR. 1000 trajectories. (b) BNN, B_{OFB} is similar scale and opposite magnitude to ALB (c.f. Figure 2b). (c) BNN on FashionMNIST, OFB is somewhat larger than with MNIST, particularly for $\tilde{R}$ (i.e. our approaches reduce overfitting) and dominates active learning bias (c.f. Figure 2c). Shading ± 1 standard error. 150 trajectories.

$B_{\text{OFB}}(\tilde{R}_{(\cdot)})$ depends on the details of the optimization algorithm and the dataset. Understanding it fully means understanding generalization in machine learning and is outside our scope. We can still gain insight into the interaction of active learning bias (ALB) and OFB. Consider the possible relationships between the magnitudes of ALB and OFB: [**ALB** $>>$ **OFB**] Removing ALB reduces overall bias and is most likely to occur when f_θ is not very expressive such that there is little chance of overfitting. [**ALB** $<<$ **OFB**] Removing ALB is irrelevant as model has massively overfit regardless. [**ALB** $\approx$ **OFB**] Here *sign* is critical. If ALB and OFB have opposite signs and similar scale, they will tend to cancel each other out. Indeed, they usually have opposite signs. B_{OFB} is usually positive: θ^* fits the training data better than unseen data. ALB is generally negative: we actively choose unusual/surprising/informative points which are harder to fit than typical points.

Therefore, when significant overfitting is possible, unless ALB is also large, removing ALB will have little effect and can *even be harmful*. This hypothesis would explain the observations in §7 if we were to show that B_{OFB} was small for linear regression but had a similar magnitude and opposite sign to ALB for the BNN. This is exactly what we show in Figure 4.

Specifically, we see that for linear regression, the B_{OFB} for models trained with $\tilde{R}$, $\tilde{R}_{\text{PURE}}$, and $\tilde{R}_{\text{LURE}}$ are all small (Figure 4a) when contrasted to the ALB shown in Figure 2a. Here ALB $>>$ OFB; removing ALB matters. For BNNs we instead see that the OFB has opposite sign to the ALB but is either similar in scale for MNIST (Figures 2b and 4b), or the OFB is much larger than ALB for Fashion MNIST (Figures 4c and 2c). The two sources of bias thus (partially) cancel out. Essentially, using active learning can be treated (quite instrumentally) as an ad hoc form of regularization. This explains why removing ALB can hurt active learning with neural networks.

9 Conclusions

Active learning is a powerful tool but raises potential problems with statistical bias. We offer a corrective weighting which removes that bias with negligible compute/memory costs and few requirements—it suits standard pool–based active learning without replacement. It requires a non–zero proposal distribution over all unlabelled points but existing acquisition functions can be easily transformed into sampling distributions. Indeed, estimates of scores like mutual information are so noisy that many applications already have an implicit proposal distribution.

We show that removing active learning bias (ALB) can be helpful in some settings, like linear regression, where the model is not sufficiently complex to perfectly match the data, such that the exact loss function and input data distribution are essential in discriminating between different possible (imperfect) model fits. We also find that removing ALB can be counter-productive for overparameterized models like neural networks, even if its removal also reduces the variance of the estimators, because here the ALB can help cancel out the bias originating from overfitting. This leads to the interesting conclusion that active learning can be helpful not only as a mechanism to reduce variance as it was originally designed, but also because it introduces a bias that can be *actively helpful* by regularizing the model. This helps explain why active learning with neural networks has shown success despite using a biased risk estimator.

We propose the following rules of thumb for deciding when to embrace or correct the bias, noting that we should always prefer $\tilde{R}_{\text{LURE}}$ to $\tilde{R}_{\text{PURE}}$. First, the more closely the acquisition proposal distribution approaches the optimal distribution (as per Theorem 6), the relatively better $\tilde{R}_{\text{LURE}}$ will be to $\tilde{R}$. Second, the less overfitting we expect, the more likely it is that $\tilde{R}_{\text{LURE}}$ will be useful as it reduces

the chance that the ALB will actually help. Third, $\tilde{R}_{\text{LURE}}$ will tend to have more of an effect for highly imbalanced datasets, as the biased estimator will over-represent actively selected but unlikely datapoints. Fourth, if the training data does not accurately represent the test data, using $\tilde{R}_{\text{LURE}}$ will likely be less important as the ALB will tend to be dwarfed by bias from the distribution shift. Fifth, at test–time, where optimization and overfitting bias are no-longer an issue, there is little cost to using $\tilde{R}_{\text{LURE}}$ to evaluate a model and it will usually be beneficial. This final application, of active learning for model evaluation, is a new research area that is opened up by our estimators.

REFERENCES

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *Eighth International Conference on Learning Representations*, April 2020.

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

Francis Bach. Active learning for mispecified generalized linear models. *Neural Information Processing Systems*, 19, 2006.

Sima Behpour, Anqi Liu, and Brian Ziebart. Active learning for probabilistic structured prediction of cuts and matchings. volume 97 of *Proceedings of Machine Learning Research*, pp. 563–572, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

William H Beluch, Tim Genewein, Andreas Nurnberger, and Jan M Kohler. The Power of Ensembles for Active Learning in Image Classification. *CVPR*, 2018.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langfor. Importance weighted active learning. *International Conference on Machine Learning*, 26, 2009.

Jonathon Byrd and Zachary C. Lipton. What is the Effect of Importance Weighting in Deep Learning? *ICML*, June 2019.

Jiaoyan Chen, Yan Zhou, Alexander Zipf, and Hongchao Fan. Deep Learning From Multiple Crowds: A Case Study of Humanitarian Mapping. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1713–1722, March 2019. ISSN 1558-0644. doi: 10.1109/TGRS.2018.2868748.

Wei Chu, Martin Zinkevich, Lihong Li, Achint Thomas, and Belle Tseng. Unbiased online active learning in data streams. KDD '11, pp. 195–203, 2011.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via Proxy: Efficient Data Selection for Deep Learning. In *Eighth International Conference on Learning Representations*, April 2020.

Corinna Cortes, Giulia Desalvo, Mehryar Mohri, Ningshan Zhang, and Claudio Gentile. Active learning with disagreement graphs. volume 97 of *Proceedings of Machine Learning Research*, pp. 1379–1387, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. *International Conference on Machine Learning*, 25, 2008.

Sebastian Farquhar, Michael Osborne, and Yarin Gal. Radial Bayesian Neural Networks: Robust Variational Inference In Big Models. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *Proceedings of the 33rd International Conference on Machine Learning*, 48:1050–1059, 2015.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete Dropout. *Neural Information Processing Systems*, pp. 10, 2017a.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian Active Learning with Image Data. *Proceedings of The 34th International Conference on Machine Learning*, 2017b.

Ravi Ganti and Alexander Gray. Upal: Unbiased pool based active learning. *Artificial Intelligence and Statistics*, 15, 2012.

Sambuddha Ghosal, Bangyou Zheng, Scott C. Chapman, Andries B. Potgieter, David R. Jordan, Xuemin Wang, Asheesh K. Singh, Arti Singh, Masayuki Hirafuji, Seishi Ninomiya, Baskar Ganapathysubramanian, Soumik Sarkar, and Wei Guo. A Weakly Supervised Deep Learning Framework for Sorghum Head Detection and Counting, 2019. Publisher: Science Partner Journal Volume: 2019.

Juan Mario Haut, Mercedes E. Paoletti, Javier Plaza, Jun Li, and Antonio Plaza. Active Learning With Convolutional Neural Networks for Hyperspectral Image Classification Using a New Bayesian Approach. *IEEE Transactions on Geoscience and Remote Sensing*, 56(11):6440–6461, November 2018. ISSN 1558-0644. doi: 10.1109/TGRS.2018.2838665. Conference Name: IEEE Transactions on Geoscience and Remote Sensing.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *CVPR*, 7(3):171–180, 2016.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian Active Learning for Classification and Preference Learning. *arXiv*, 2011.

Peiyun Hu, Zachary C. Lipton, Anima Anandkumar, and Deva Ramanan. Active Learning with Partial Feedback. In *International Conference on Learning Representations*, 2019.

Sheng-Jun Huang, Jia-Wei Zhao, and Zhao-Yang Liu. Cost-Effective Training of Deep CNNs with Active Model Adaptation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 1580–1588, New York, NY, USA, July 2018. Association for Computing Machinery. ISBN 978-1-4503-5552-0.

Henrik Imberg, Johan Jonasson, and Marina Axelson-Fisk. Optimal sampling in unbiased active learning. *Artificial Intelligence and Statistics*, 23, 2020.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

Benjamin Kellenberger, Diego Marcos, Sylvain Lobry, and Devis Tuia. Half a Percent of Labels is Enough: Efficient Animal Detection in UAV Imagery Using Deep CNNs and Active Learning. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):9524–9533, December 2019. ISSN 0196-2892. Publisher: Institute of Electrical and Electronics Engineers.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. *arXiv:1906.08158 [cs, stat]*, 2019.

Akshay Krishnamurthy, Alekh Agarwal, Tzu-Kuo Huang, Hal Daumé, III, and John Langford. Active learning for cost-sensitive classification. volume 70 of *Proceedings of Machine Learning Research*, pp. 1915–1924, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pp. 3–12, Berlin, Heidelberg, August 1994. Springer-Verlag. ISBN 978-0-387-19889-7.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. Practical Obstacles to Deploying Active Learning. *Empirical Methods in Natural Language Processing*, November 2019.

David J C MacKay. Information-Based Objective Functions for Active Data Selection. *Neural Computation*, 4(4):590–604, 1992.

Phuc Nguyen, Deva Ramanan, and Charless Fowlkes. Active Testing: An Efficient and Robust Framework for Estimating Accuracy. In *International Conference on Machine Learning*, pp. 3759–3768. PMLR, July 2018. ISSN: 2640-3498.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized Prior Functions for Deep Reinforcement Learning. *Neural Information Processing Systems*, 2018.

Tom Rainforth. *Automating Inference, Learning, and Design using Probabilistic Programming*. PhD thesis, University of Oxford, 2017.

Tom Rainforth, A Goliński, Frank Wood, and Sheheryar Zaidi. Target–aware bayesian inference: how to beat optimal conventional estimators. *Journal of Machine Learning Research*, 21(88), 2020.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the Convergence of Adam and Beyond. *International Conference on Learning Representations*, February 2018.

Ozan Sener and Silvio Savarese. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations*, February 2018.

Burr Settles. Active Learning Literature Survey. *Machine Learning*, 15(2):201–221, 2010.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep Active Learning for Named Entity Recognition. In *International Conference on Learning Representations*, 2018.

Weishi Shi and Qi Yu. Fast direct search in an optimally compressed continuous target space for efficient multi-label active learning. volume 97 of *Proceedings of Machine Learning Research*, pp. 5769–5778, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Aditya Siddhant and Zachary C. Lipton. Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2904–2909, Brussels, Belgium, 2018. Association for Computational Linguistics.

Samrath Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational Adversarial Active Learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5971–5980, Seoul, Korea (South), October 2019. IEEE. ISBN 978-1-72814-803-8. doi: 10.1109/ICCV.2019.00607.

Masashi Sugiyama. Active learning for misspecified models. *Neural Information Processing Systems*, 18:1305–1312, 2006.

Iiris Sundin, Peter Schulam, Eero Siivola, Aki Vehtari, Suchi Saria, and Samuel Kaski. Active learning for decision-making from imbalanced observational data. volume 97 of *Proceedings of Machine Learning Research*, pp. 6046–6055, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-Effective Active Learning for Deep Image Classification. *IEEE Transactions on Circuits and Systems for Video Technology*, January 2017. arXiv: 1701.03551.

Si Wen, Tahsin M. Kurc, Le Hou, Joel H. Saltz, Rajarsi R. Gupta, Rebecca Batiste, Tianhao Zhao, Vu Nguyen, Dimitris Samaras, and Wei Zhu. Comparison of Different Classifiers with Active Learning to Support Quality Control in Nucleus Segmentation in Pathology Images. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2017:227–236, 2018. ISSN 2153-4063.

Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. Active Learning with Logged Data. In *International Conference on Machine Learning*, pp. 5521–5530. PMLR, July 2018. ISSN: 2640-3498.

Jie Yang, Thomas Drake, Andreas Damianou, and Yoelle Maarek. Leveraging Crowdsourcing Data for Deep Active Learning An Application: Learning Intents in Alexa. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pp. 23–32, Republic and Canton of Geneva, CHE, April 2018. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-5639-8.

Donggeun Yoo and In So Kweon. Learning Loss for Active Learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 93–102, Long Beach, CA, USA, June 2019. IEEE. ISBN 978-1-72813-293-8.

Yao Zhang and Alpha A. Lee. Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chemical Science*, 10(35):8154–8163, September 2019. doi: 10.1039/C9SC00616H. Publisher: The Royal Society of Chemistry.

A Overview of Active Learning

Active learning selectively picks datapoints for which to acquire labels with the aim of more sample-efficient learning. For an excellent overview of the general active learning problem setting we refer the reader to Settles (2010).

Since that review was written, a number of significant advances have further developed active learning. Houlsby et al. (2011) develop an efficient way to estimate the mutual information between model parameters and the output distribution, which can be used for the Bayesian Active Learning by Disagreement (BALD) score.

Active learning has been applied to deep learning, especially for vision e.g., Gal et al. (2017b); Wang et al. (2017). In neural networks specifically, empirical work has suggested that simple geometric core-set style approaches can outperform uncertainty-based acquisition functions (Sener & Savarese, 2018).

Rather than trying to improve the quality of active learning training itself, recent work has aimed to speed up acquisition from a computational perspective (Coleman et al., 2020) or to allow batch acquisition in order to parallelize labelling (Kirsch et al., 2019; Ash et al., 2020). In some cases, work has also focused on applying active learning to specific settings with particular constraints (Krishnamurthy et al., 2017; Yan et al., 2018; Sundin et al., 2019; Behpour et al., 2019; Shi & Yu, 2019; Hu et al., 2019).

B Proofs

B.1 Proof of Lemma 1

Lemma 1. *The individual terms a_m of $\tilde{R}_{PURE}$ are unbiased estimators of the risk: $\mathbb{E}\left[a_m\right] = r$.*

Proof. We begin by applying the tower property of expectations:

$$\mathbb{E}\left[a_m\right] = \mathbb{E}\left[w_m \mathcal{L}_{i_m} + \frac{1}{N}\sum_{t=1}^{m-1} \mathcal{L}_{i_t}\right]$$

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\mathbb{E}_{i_m}\left[w_m \mathcal{L}_{i_m} + \frac{1}{N}\sum_{t=1}^{m-1} \mathcal{L}_{i_t} \,\middle|\, \mathcal{D}_{\text{pool}}, i_{1:m-1}\right]\right].$$

By further noting that $\mathbb{E}_{i_m}\left[w_m \mathcal{L}_{i_m} \,|\, \mathcal{D}_{\text{pool}}, i_{1:m-1}\right]$ can be written out analytically as a sum over all the possible values of i_m, while $\frac{1}{N}\sum_{t=1}^{m-1} \mathcal{L}_{i_t}$ is deterministic given $\mathcal{D}_{\text{pool}}$ and $i_{1:m-1}$ we have:

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\sum_{n \notin i_{1:m-1}} \left(\cancel{q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})} \frac{\mathcal{L}_{i_m}}{N\, \cancel{q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})}}\right) + \frac{1}{N}\sum_{t=1}^{m-1} \mathcal{L}_{i_t}\right]$$

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_n\right],$$

But $\mathcal{L}_n$ is now independent of the indices which have been sampled:

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}}\left[\frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_n\right] = \mathbb{E}_{\mathcal{D}_{\text{pool}}}\left[\hat{R}\right] = r.$$

□

B.2 PROOF OF THE UNBIASEDNESS OF $\tilde{R}_{\text{PURE}}$: THEOREM 1

Theorem 1. *$\tilde{R}_{PURE}$ as defined above has the properties:*

$$\mathbb{E}\left[\tilde{R}_{PURE}\right] = r, \qquad \text{Var}[\tilde{R}_{PURE}] = \frac{\sigma^2}{M}, \qquad \textit{where}$$

$$\sigma^2 = \frac{1}{M}\sum_{m=1}^{M}\text{Var}[a_m] = \frac{\text{Var}\left[\mathcal{L}(\mathbf{y}, f_\theta(\mathbf{x}))\right]}{N} + \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[w_m\mathcal{L}_{i_m}|i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right]. \tag{2}$$

Proof. Having established Lemma 1, unbiasedness follows quickly from the linearity of expectations:

$$\mathbb{E}\left[\tilde{R}_{\text{PURE}}\right] = \mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M}a_m\right] = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[a_m\right] = r.$$

For the variance we instead have (starting with the definition of $\tilde{R}_{\text{PURE}}$):

$$\text{Var}\left[\tilde{R}_{\text{PURE}}\right] = \mathbb{E}\left[\left(\frac{1}{M}\sum_{m=1}^{M}a_m - \mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M}a_m\right]\right)^2\right] = \mathbb{E}\left[\left(\frac{1}{M}\sum_{m=1}^{M}(a_m - r)\right)^2\right], \tag{6}$$

explicitly writing out the sum of the squares and moving expectations past the summation:

$$= \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}\left[\left(a_m - r\right)^2\right] + \frac{2}{M^2}\sum_{m=1}^{M}\sum_{k<m}\mathbb{E}\left[(a_m - r)(a_k - r)\right],$$

and noting the form of the first term:

$$= \frac{1}{M^2}\sum_{m=1}^{M}\text{Var}[a_m] + \frac{2}{M^2}\sum_{m=1}^{M}\sum_{k<m}\mathbb{E}\left[(a_m - r)(a_k - r)\right].$$

We attack this line in two parts, beginning with the summands of the second term—$\mathbb{E}\left[(a_m - r)(a_k - r)\right]$—showing that they are necessarily zero. Later, we will address the variance of a_m. Substitute in the definition of a_m from equation (3.1):

$$\mathbb{E}\left[(a_m - r)(a_k - r)\right] = \mathbb{E}\left[\left(w_m\mathcal{L}_{i_m} + \frac{1}{N}\sum_{t=1}^{m-1}\mathcal{L}_{i_t} - r\right)\left(w_k\mathcal{L}_{i_k} + \frac{1}{N}\sum_{s=1}^{k-1}\mathcal{L}_{i_s} - r\right)\right].$$

We introduce the notation $r_m^{rem} = r - \frac{1}{N}\sum_{t=1}^{m-1}\mathcal{L}_{i_t}$ to describe the remainder of the risk ascribable to the datapoint with index i_m. Then, by multiplying out the terms we have:

$$= \underbrace{\mathbb{E}\left[w_m\mathcal{L}_{i_m}w_k\mathcal{L}_{i_k}\right]}_{\text{a}} - \underbrace{\mathbb{E}\left[w_m\mathcal{L}_{i_m}r_k^{rem}\right]}_{\text{b}} - \underbrace{\mathbb{E}\left[w_k\mathcal{L}_{i_k}r_m^{rem}\right]}_{\text{c}} + \underbrace{\mathbb{E}\left[r_k^{rem}r_m^{rem}\right]}_{\text{d}}.$$

Now by the tower property:

$$\text{a} = \mathbb{E}\left[\mathbb{E}\left[w_m\mathcal{L}_{i_m}w_k\mathcal{L}_{i_k}\,|\,\mathcal{D}_{\text{pool}}, i_{1:m-1}\right]\right],$$

and noting that because $k < m$, $w_k\mathcal{L}_{i_k}$ is deterministic given $\mathcal{D}_{\text{pool}}$ and $i_{1:m-1}$:

$$= \mathbb{E}\left[w_k\mathcal{L}_{i_k}\mathbb{E}\left[w_m\mathcal{L}_{i_m}\,|\,\mathcal{D}_{\text{pool}}, i_{1:m-1}\right]\right]$$
$$= \mathbb{E}\left[w_k\mathcal{L}_{i_k}r_m^{rem}\right],$$

which thus cancels with c.

The b and d cancel similarly:

$$\text{b} = \mathbb{E}\left[\mathbb{E}\left[w_m\mathcal{L}_{i_m}r_k^{rem}\,|\,\mathcal{D}_{\text{pool}}, i_{1:m-1}\right]\right]$$
$$= \mathbb{E}\left[r_k^{rem}\mathbb{E}\left[w_m\mathcal{L}_{i_m}\,|\,\mathcal{D}_{\text{pool}}, i_{1:m-1}\right]\right]$$
$$= \mathbb{E}\left[r_k^{rem}r_m^{rem}\right] = \text{d}.$$

Putting this together, we have that:

$$\mathbb{E}\left[(a_m - r)(a_k - r)\right] = 0 \quad \forall k < m,$$

It follows that:

$$\frac{1}{M^2}\sum_{m=1}^{M}\text{Var}[a_m] + \frac{2}{M^2}\sum_{m=1}^{M}\sum_{k<m}\mathbb{E}\left[(a_m - r)(a_k - r)\right]$$
$$= \frac{1}{M^2}\sum_{m=1}^{M}\text{Var}[a_m].$$

We now prove the theorem by examining this variance. Note, similarly to (6), $\text{Var}[a_m] = \mathbb{E}\left[(a_m - r)^2\right]$ because a_m are unbiased. Then, applying the tower property of expectations:

$$\text{Var}[a_m] = \mathbb{E}\left[(a_m - r)^2\right]$$
$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\mathbb{E}_{i_m}\left[(a_m - r)^2 \,\middle|\, i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right],$$

and using the definition of the conditional variance of $a_m - r$:

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[a_m - r \mid i_{1:m-1}, \mathcal{D}_{\text{pool}}\right] + \left(\mathbb{E}_{i_m}\left[a_m - r \mid i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right)^2\right],$$

noting that r is not a random variable and that $\mathbb{E}_{i_m}\left[a_m | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right] = \hat{R}$:

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[a_m \mid i_{1:m-1}, \mathcal{D}_{\text{pool}}\right] + \left(\hat{R} - r\right)^2\right]$$

further noting that $\frac{1}{N}\sum_{t=1}^{m-1}\mathcal{L}_{i_t}$ is deteriministic conditioned on $i_{1:m-1}$, and using linearity of expectations:

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[w_m L_{i_m} \mid i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right] + \mathbb{E}\left[\left(\hat{R} - r\right)^2\right]$$

and noting again that r is not a random variable:

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[w_m \mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right] + \text{Var}\left[\hat{R}\right]$$
$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[w_m \mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right] + \frac{\text{Var}\left[\mathcal{L}(\text{y}, f_\theta(\mathbf{x})\right]}{N}$$

where $\mathbf{x}, \text{y} \sim p_{\text{data}}$, as required. □

B.3 Proof of the consistency of $\tilde{R}_{\text{PURE}}$: Theorem 2

Theorem 2. *Let $\alpha = N/M$ and assume that $\alpha > 1$. If $\mathbb{E}\left[\mathcal{L}(\text{y}, f_\theta(\mathbf{x}))^2\right] < \infty$ and*

$$\exists \beta > 0 \;:\; \min_{n \in \{1:N \setminus i_{1:m-1}\}} q(i_m = n; i_{1:m-1}, \mathcal{D}_{\text{pool}}) \geq \beta/N \quad \forall N \in \mathbb{Z}^+, m \leq N,$$

then $\tilde{R}_{PURE}$ converges in its L^2 norm to r as $M \to \infty$, i.e., $\lim_{M\to\infty}\mathbb{E}\left[\left(\tilde{R}_{PURE} - r\right)^2\right] = 0$.

Proof. Theorem 1 showed that $\tilde{R}_{\text{PURE}}$ is an unbiased estimator and so we first note that its MSE is simply its variance, which we found in (2). Substituting $N = \alpha M$:

$$\mathbb{E}\left[\left(\tilde{R}_{\text{PURE}} - r\right)^2\right] = \text{Var}\left[\tilde{R}_{\text{PURE}}\right]$$
$$= \frac{\text{Var}\left[\mathcal{L}(\text{y}, f_\theta(\mathbf{x})\right]}{\alpha M^2} + \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[w_m \mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right].$$

The first term tends to zero as $M \to \infty$ as our standard assumptions guarantee that $\frac{1}{\alpha}\text{Var}\left[\mathcal{L}(\text{y}, f_\theta(\mathbf{x})\right] < \infty$. For the second term we note that our assumptions about q guarantee

that $w_m \leq 1/\beta$ and thus:

$$\begin{aligned}\text{Var}\left[w_m \mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right] &= \mathbb{E}\left[w_m^2 \mathcal{L}_{i_m}^2 | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right] - \left(\hat{R} - \frac{1}{M}\sum_{t=1}^{m-1} \mathcal{L}_{i_t}\right)^2 \\ &\leq \frac{1}{\beta^2}\mathbb{E}\left[\mathcal{L}_{i_m}^2 | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right] - \left(\hat{R} - \frac{1}{M}\sum_{t=1}^{m-1} \mathcal{L}_{i_t}\right)^2 \\ &< \infty \quad \forall i_{1:m-1}, \mathcal{D}_{\text{pool}}\end{aligned}$$

as our assumptions guarantee that $\frac{1}{\beta^2} < \infty$,$\mathbb{E}_{i_m}\left[\mathcal{L}_{i_m}^2\right] < \infty$ and the empirical risk and losses are finite.

Given that each $\text{Var}\left[w_m \mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]$ is finite, it follows that:

$$s^2 := \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[w_m \mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right] < \infty,$$

and we thus have:

$$\lim_{M\to\infty} \mathbb{E}\left[\left(\tilde{R}_{\text{PURE}} - R\right)^2\right] = \lim_{M\to\infty}\left(\frac{\text{Var}\left[\mathcal{L}(\mathrm{y}, f_\theta(\mathbf{x})\right]}{\alpha M^2} + \frac{s^2}{M}\right) = 0$$

as desired. □

B.4 Derivation of the Constants of $\tilde{R}_{\text{LURE}}$

We note from before that because of the unbiasedness of a_m:

$$\mathbb{E}\left[C\sum_{m=1}^{M} c_m a_m\right] = r \quad \text{where} \quad C = \frac{1}{\sum_{m=1}^{M} c_m},$$

To construct our improved estimator $\tilde{R}_{\text{LURE}}$, we now need to find the right constants c_m, that in turn lead to overall weights v_m (as per (3)) such that $\mathbb{E}\left[v_m\right] = 1$. We start by substituting in the definition of a_m:

$$\tilde{R}_{\text{LURE}} := C\sum_{m=1}^{M} v_m \mathcal{L}_{i_m} := C\sum_{m=1}^{M} c_m a_m = C\sum_{m=1}^{M}\left(c_m w_m \mathcal{L}_{i_m} + \frac{c_m}{N}\sum_{t=1}^{m-1}\mathcal{L}_{i_t}\right),$$

and then redistributing the $\mathcal{L}_{i_t}$ from later terms where they match m:

$$v_m = c_m w_m + \frac{1}{N}\sum_{t=m+1}^{M} c_t.$$

Note that though $\tilde{R}_{\text{LURE}}$ remains an unbiased estimator of the risk, each individual term $v_m \mathcal{L}_{i_m}$ is not. Now we require: $\mathbb{E}\left[v_m\right] = 1 \quad \forall m \in \{1, ..., M\}$. Remembering that $w_m = 1/(N q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}}))$:

$$\begin{aligned}\mathbb{E}\left[v_m\right] &= \frac{c_m}{N}\mathbb{E}\left[\frac{1}{q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})}\right] + \frac{1}{N}\sum_{t=m+1}^{M} c_t \\ &= \frac{c_m}{N}\sum_{n \notin i_{1:m-1}} \frac{q(i_m = n; i_{1:m-1}, \mathcal{D}_{\text{pool}})}{q(i_m = n; i_{1:m-1}, \mathcal{D}_{\text{pool}})} + \frac{1}{N}\sum_{t=m+1}^{M} c_t \\ &= \frac{c_m(N - m + 1)}{N} + \frac{1}{N}\sum_{t=m+1}^{M} c_t.\end{aligned}$$

Imposing that each $\mathbb{E}\left[v_m\right] = 1$, we now have M equations for M unknowns $c_1, \ldots, c_M$, such that we can find the required values of c_m by solving the set of simultaneous equations:

$$(N - m + 1)\frac{c_m}{N} + \frac{1}{N}\sum_{t=m+1}^{M} c_t = 1 \quad \forall m \in \{1, \ldots, M\}. \tag{7}$$

We do this by induction. First, consider $\mathbb{E}[v_m] - \mathbb{E}[v_{m+1}] = 0$, for which can be rewritten:

$$(N-m+1)\frac{c_m}{N} - (N-m)\frac{c_{m+1}}{N} + \frac{c_{m+1}}{N} = 0$$

and thus:

$$c_m = \frac{N-m-1}{N-m+1}c_{m+1}. \tag{8}$$

By further noting that the solution for $m = M$ is trivial:

$$c_M = \frac{N}{N-M+1},$$

we have by induction

$$\begin{aligned} c_m &= \frac{N}{N-M+1}\prod_{t=m}^{M-1}\frac{N-t-1}{N-t+1} \\ &= \frac{N}{N-M+1}\exp\left(\sum_{t=m}^{M-1}\log(N-t-1) - \log(N-t+1)\right). \end{aligned}$$

Now we can exploit the fact that there is a canceling of most of the term in this sum. The exceptions are $-\log(N-m+1)$, $-\log(N-m)$, $\log(N-M+1)$, and $\log(N-M)$. We thus have:

$$\begin{aligned} c_m &= \frac{N}{N-M+1}\exp\left(\log(N-M) + \log(N-M+1) - \log(N-m) - \log(N-m+1)\right) \\ &= \frac{N(N-M)}{(N-m)(N-m+1)} \end{aligned}$$

which is our final simple form for c_m. We can now check that this satisfies the required recursive relationship (noting it trivially gives the correct value for c_M) as per (8):

$$\begin{aligned} c_m &= \left(\frac{N-m-1}{N-m+1}\right)c_{m+1} \\ &= \left(\frac{N-m-1}{N-m+1}\right)\frac{N(N-M)}{(N-m-1)(N-m)} \\ &= \frac{N(N-M)}{(N-m)(N-m+1)} \end{aligned}$$

as required. Similarly, we can straightforwardly show that this form of c_m satisfies (7) by substitution.

We then find the form of v_m given this expression for c_m. Remember that:

$$v_m = c_m w_m + \frac{1}{N}\sum_{t=m+1}^{M} c_t.$$

We can rearrange (7) to:

$$\frac{1}{N}\sum_{t=m+1}^{M} c_t = 1 - \frac{N-m+1}{N}c_m$$

from which it follows that:

$$v_m = 1 + c_m\left(w_m - \frac{N-m+1}{N}\right).$$

Substituting in our expressions for c_m and w_m we thus have:

$$\begin{aligned} v_m &= 1 + \frac{N-M}{(N-m)(N-m+1)}\left(\frac{1}{q(i_m; i_{1:m-1}, f(\theta_{m-1}), \mathcal{D}_{\text{pool}})} - (N-m+1)\right) \\ v_m &= 1 + \frac{N-M}{N-m}\left(\frac{1}{(N-m+1)\,q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})} - 1\right), \end{aligned}$$

which is the form given in the original expression.

To finish our definition, we simply need to derive C:

$$\begin{aligned} C &= \left(\sum_{m=1}^{M} c_m \right)^{-1} \\ &= \left(N(N-M) \sum_{m=1}^{M} \frac{1}{(N-m)(N-m+1)} \right)^{-1} \\ &= \left(N(N-M) \sum_{m=1}^{M} \frac{1}{N-m} - \frac{1}{N-m+1} \right)^{-1} \end{aligned}$$

where we now have a telescopic sum so

$$\begin{aligned} &= \left(N(N-M) \left(\frac{1}{N-M} - \frac{1}{N} \right) \right)^{-1} \\ &= \frac{1}{M}. \end{aligned} \tag{9}$$

We thus see that our v_m always sum to M, giving the quoted form for $\tilde{R}_{\text{LURE}}$ in the main paper.

B.5 Proof of Unbiasedness and Variance for $\tilde{R}_{\text{LURE}}$: Theorem 3

Theorem 3. *$\tilde{R}_{LURE}$ as defined above has the following properties:*

$$\mathbb{E}\left[\tilde{R}_{LURE}\right] = r, \qquad \text{Var}[\tilde{R}_{LURE}] = \frac{\sigma^2_{\text{LURE}}}{M}, \qquad \textit{where}$$

$$\sigma^2_{\text{LURE}} = \frac{1}{M} \sum_{m=1}^{M} c_m^2 \left(\frac{\text{Var}\left[\mathcal{L}(\mathbf{y}, f_\theta(\mathbf{x})\right]}{N} + \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}} \left[\text{Var}\left[w_m \mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}} \right] \right] \right). \tag{4}$$

Proof. $\tilde{R}_{\text{LURE}}$ is, by construction a linear combination of the weights a_m. By Lemma 1 each a_m is an unbiased estimator of r. So by the linearity of expectation, $\mathbb{E}\left[\tilde{R}_{\text{LURE}}\right] = r$.

The variance requires a degree of care because although we previously showed that $\text{Var}[\tilde{R}_{\text{PURE}}] = \frac{1}{M}\sum_{m=1}^{M} \text{Var}[a_m]$, we cannot directly apply this result to the weighted $c_m a_m$ because a_m are not independent.

Instead we employ an analogous proof to Theorem 1:

$$\begin{aligned} \text{Var}\big[\tilde{R}_{\text{LURE}}\big] &= \text{Var}\Big[\frac{1}{M}\sum_{m=1}^{M} c_m a_m\Big] \\ &= \mathbb{E}\left[\left(\frac{1}{M}\sum_{m=1}^{M} c_m a_m - \mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M} c_m a_m\right] \right)^2 \right], \end{aligned}$$

by the unbiasedness of $\tilde{R}_{\text{LURE}}$:

$$\begin{aligned} &= \mathbb{E}\left[\Big(\frac{1}{M}\sum_{m=1}^{M} \big(c_m a_m - r\big) \Big)^2 \right] \\ &= \frac{1}{M^2}\sum_{m=1}^{M} \mathbb{E}\left[\big(c_m a_m - r\big)^2 \right] + \frac{2}{M^2}\sum_{m=1}^{M}\sum_{k<m} \mathbb{E}\left[(c_m a_m - r)(c_k a_k - r) \right] \\ &= \underbrace{\frac{1}{M^2}\sum_{m=1}^{M} \mathbb{E}\left[\big(c_m a_m - r\big)^2 \right]}_{1} + \underbrace{\frac{2}{M^2}\sum_{m=1}^{M}\sum_{k<m} c_m c_k \mathbb{E}\left[\left(a_m - \frac{r}{c_m}\right)\left(a_k - \frac{r}{c_k}\right) \right]}_{2} \end{aligned} \tag{10}$$

Now considering the expectation inside term 2 we have:

$$\mathbb{E}\left[\left(a_m - \frac{r}{c_m}\right)\left(a_k - \frac{r}{c_k}\right)\right] = \mathbb{E}\left[\left(a_m - r + r - \frac{r}{c_m}\right)\left(a_k - r + r - \frac{r}{c_k}\right)\right]$$
$$= \left(r - \frac{r}{c_m}\right)\left(r - \frac{r}{c_k}\right) + \left(r - \frac{r}{c_k}\right)\mathbb{E}\left[(a_m - r)\right] + \left(r - \frac{r}{c_m}\right)\mathbb{E}\left[(a_k - r)\right]$$
$$= \left(r - \frac{r}{c_m}\right)\left(r - \frac{r}{c_k}\right).$$

Substituting this into 2, taking out r and multiplying in c_m and c_k we have (remembering $\sum_{m=1}^{M} c_m = M$ from (9)):

$$\begin{aligned} 2 &= \frac{2r^2}{M^2}\sum_{m=1}^{M}\sum_{k<m}(c_m - 1)(c_k - 1) \\ &= \frac{r^2}{M^2}\underbrace{\sum_{m=1}^{M}(c_m - 1)}_{=0}\underbrace{\sum_{k=1}^{M}(c_k - 1)}_{=0} - \frac{r^2}{M^2}\sum_{m=1}^{M}(c_m - 1)^2 \\ &= -\frac{r^2}{M^2}\sum_{m=1}^{M}(c_m - 1)^2, \end{aligned} \tag{11}$$

where the second term does not have an immediate simple form.

Now consider 1 and again similarly to Theorem 1 we have:

$$1 = \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\mathbb{E}_{i_m}\left[(c_m a_m - r)^2 \,\middle|\, i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right]$$
$$= \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[c_m a_m - r \mid i_{1:m-1}, \mathcal{D}_{\text{pool}}\right] + \left(\mathbb{E}_{i_m}\left[c_m a_m - r \mid i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right)^2\right]$$
$$= \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[c_m a_m \mid i_{1:m-1}, \mathcal{D}_{\text{pool}}\right] + \left(c_m \hat{R} - r\right)^2\right]$$
$$= \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[c_m^2 \text{Var}\left[w_m \mathcal{L}_{i_m} \mid i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right] + \mathbb{E}\left[\left(c_m \hat{R} - r\right)^2\right]$$
$$= \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[c_m^2 \text{Var}_{q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})}\left[w_m \mathcal{L}_{i_m}\right]\right] + c_m^2 \text{Var}\left[\hat{R}\right] + r^2(c_m - 1)^2$$
$$= \frac{1}{M^2}\sum_{m=1}^{M} c_m^2\left(\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}_{q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})}\left[w_m \mathcal{L}_{i_m}\right]\right] + \frac{\text{Var}\left[\mathcal{L}(\mathbf{y}, f_\theta(\mathbf{x}))\right]}{N}\right) + r^2(c_m - 1)^2$$

This final $r^2(c_m - 1)^2$ terms now cancel exactly with 2 as per (11), such that substituting the above and (11) back into (10) yields the desired result. □

B.6 PROOF OF THE CONSISTENCY OF $\tilde{R}_{\text{LURE}}$: THEOREM 4

Theorem 4. *Under the same assumptions as Theorem 2:* $\lim_{M\to\infty} \mathbb{E}\left[\left(\tilde{R}_{LURE} - r\right)^2\right] = 0.$

Proof. As before, since $\tilde{R}_{\text{LURE}}$ is unbiased the MSE is simply the variance so:

$$\mathbb{E}\left[\left(\tilde{R}_{\text{PURE}} - r\right)^2\right] = \text{Var}\left[\tilde{R}_{\text{PURE}}\right]$$
$$= \frac{1}{M^2}\sum_{m=1}^{M} c_m^2 \left(\frac{\text{Var}\left[\mathcal{L}(\text{y}, f_\theta(\mathbf{x})\right]}{N} + \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}_{q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})}\left[w_m \mathcal{L}_{i_m}\right]\right]\right).$$

We already showed in the proof of Theorem 2 that $\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}_{q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})}\left[w_m \mathcal{L}_{i_m}\right]\right]$ is finite given our assumptions, while our assumptions further guarantee that $\text{Var}\left[\mathcal{L}(\text{y}, f_\theta(\mathbf{x})\right] < \infty$. As such, there must be some finite constant d such that

$$\frac{\text{Var}\left[\mathcal{L}(\text{y}, f_\theta(\mathbf{x})\right]}{N} + \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}_{q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}})}\left[w_m \mathcal{L}_{i_m}\right]\right] < d, \quad \forall N \geq 1$$

and thus

$$\mathbb{E}\left[\left(\tilde{R}_{\text{PURE}} - r\right)^2\right] < \frac{d}{M^2}\sum_{m=1}^{M} c_m^2 = \frac{d}{M^2}\sum_{m=1}^{M}\left(\frac{N(N-M)}{(N-m)(N-m+1)}\right)^2$$

and by substituting $N = \alpha M$

$$= \frac{d}{M^2}\sum_{m=1}^{M}\left(\frac{\alpha M(\alpha M - M)}{(\alpha M - m)(\alpha M - m + 1)}\right)^2$$
$$< \frac{d}{M^2}\sum_{m=1}^{M}\left(\frac{\alpha M}{\alpha M - M + 1}\right)^2$$
$$= \frac{d\alpha^2 M}{\left((\alpha - 1)M + 1\right)^2}$$

which clearly tends to zero as $M \to \infty$ (remembering that $\alpha > 1$) and we are done. □

B.7 PROOF OF BIAS AND VARIANCE OF STANDARD ACTIVE LEARNING ESTIMATOR: THEOREM 5

Theorem 5. *Let* $\mu_m := \mathbb{E}\left[\mathcal{L}_{i_m}\right]$ *and* $\mu_{m|i,\mathcal{D}} := \mathbb{E}\left[\mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]$. *For* $\tilde{R}$ *(defined in* (1)*):*

$$\mathbb{E}\left[\tilde{R}\right] = \frac{1}{M}\sum\nolimits_{m=1}^{M} \mu_m \quad (\neq r \textit{ in general})$$

$$\text{Var}[\tilde{R}] = \frac{1}{M^2}\sum\nolimits_{m=1}^{M}\Bigg(\overbrace{\text{Var}\left[\mathcal{L}(\text{y}, f_\theta(\mathbf{x})\right]/N}^{1} + \overbrace{\mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[\mathcal{L}_{i_m} | i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right]}^{2}$$
$$+ \underbrace{\text{Var}[\mu_{m|i,\mathcal{D}} - \hat{R}]}_{3} + \underbrace{2\,\text{Cov}(\hat{R}, \mu_{m|i,\mathcal{D}} - \hat{R})}_{4} + 2\sum\nolimits_{k=1}^{m-1} \underbrace{\text{Cov}\left(\mathcal{L}_{i_m}, \mathcal{L}_{i_k}\right)}_{5}\Bigg). \tag{5}$$

Proof. The result of the bias follows immediately from definition of μ_m and the linearity of expectations.

For the variance we start by applying standard rules for the variance of the sum of correlated variables to give

$$\text{Var}[\tilde{R}] = \frac{1}{M^2}\sum_{m=1}^{M}\left(\text{Var}[\mathcal{L}_{i_m}] + 2\sum_{k=1}^{m-1}\text{Cov}\left(\mathcal{L}_{i_m}, \mathcal{L}_{i_k}\right)\right). \tag{12}$$

Focusing on the first term we have through the tower property of expectations

$$\text{Var}[\mathcal{L}_{i_m}] = \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\mathbb{E}_{i_m}\left[(\mathcal{L}_{i_m} - \mu_m)^2 \middle| i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right],$$

now applying a bias–variance decomposition to the inner conditional expectation

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[\mathcal{L}_{i_m} \middle| i_{1:m-1}, \mathcal{D}_{\text{pool}}\right] + \left(\mu_{m|i,\mathcal{D}} - \mu_m\right)^2\right],$$

and noting that $\mathbb{E}\left[\mu_{m|i,\mathcal{D}}\right] = \mu_m$

$$= \mathbb{E}_{\mathcal{D}_{\text{pool}}, i_{1:m-1}}\left[\text{Var}\left[\mathcal{L}_{i_m} \middle| i_{1:m-1}, \mathcal{D}_{\text{pool}}\right]\right] + \text{Var}[\mu_{m|i,\mathcal{D}}].. \tag{13}$$

We then further have

$$\begin{aligned}
\text{Var}[\mu_{m|i,\mathcal{D}}] &= \text{Var}\left[\hat{R} + \left(\mu_{m|i,\mathcal{D}} - \hat{R}\right)\right] \\
&= \text{Var}\left[\hat{R}\right] + \text{Var}\left[\mu_{m|i,\mathcal{D}} - \hat{R}\right] + 2\,\text{Cov}\left(\hat{R}, \mu_{m|i,\mathcal{D}} - \hat{R}\right) \\
&= \frac{\text{Var}\left[\mathcal{L}(\mathrm{y}, f_\theta(\mathbf{x})\right]}{N} + \text{Var}\left[\mu_{m|i,\mathcal{D}} - \hat{R}\right] + 2\,\text{Cov}\left(\hat{R}, \mu_{m|i,\mathcal{D}} - \hat{R}\right)
\end{aligned}$$

and the proof is completed but substituting the above into (13) and then (13) into (12). □

B.8 Proof of Optimal Proposal Distribution: Theorem 6

Theorem 6. *Given a non-negative loss, the optimal proposal distribution*

$$q^*(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}}) = \mathcal{L}_{i_m} / \Sigma_{n \notin i_{1:m-1}} \mathcal{L}_n$$

yields estimators exactly equal to the pool risk, that is $\tilde{R}_{PURE} = \tilde{R}_{LURE} = \hat{R}$ almost surely $\forall M$.

Proof. We start by proving the result for the simpler case of $\tilde{R}_{\text{PURE}}$ before considering $\tilde{R}_{\text{LURE}}$. To make the notation simpler, we will introduce hypothetical indices i_t for $t > M$, noting that their exact values will not change the proof provided that they are all distinct to each other and the real indices (i.e. that they are a possible realization of the active sampling process in the setting $M = N$).

For $\tilde{R}_{\text{PURE}}$, the proof follows straightforwardly from substituting the definition of the optimal proposal into the a_m form of the estimator:

$$\begin{aligned}
\tilde{R}_{\text{PURE}} &= \frac{1}{M}\sum_{m=1}^{M} a_m \\
&= \frac{1}{M}\sum_{m=1}^{M}\left(w_m \mathcal{L}_{i_m} + \frac{1}{N}\sum_{t=1}^{m-1}\mathcal{L}_{i_t}\right) \\
&= \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{N}\sum_{t=m}^{N}\mathcal{L}_{i_t}\right) + \left(\frac{1}{N}\sum_{t=1}^{m-1}\mathcal{L}_{i_t}\right) \\
&= \frac{1}{N}\sum_{t=1}^{N}\mathcal{L}_{i_t}
\end{aligned}$$

and because all possible indices are uniquely visited:

$$\begin{aligned}
&= \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}_n \\
&= \hat{R}.
\end{aligned}$$

The proof proceeds identically for the case of $\nabla\mathcal{L}_{i_m}$ because the gradient passes through the summations.

For $\tilde{R}_{\text{LURE}}$, we similarly substitute the optimal proposal into the definition of the estimator:

$$\begin{aligned}
\tilde{R}_{\text{LURE}} =& \frac{1}{M}\sum_{m=1}^{M} v_m \mathcal{L}_{i_m} \\
=& \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}_{i_m} + \frac{N-M}{N-m}\left(\frac{\mathcal{L}_{i_m}}{(N-m+1)q^*(i_m; i_{1:m-1}, f_{\theta_{m-1}}, \mathcal{D}_{\text{pool}})} - \mathcal{L}_{i_m}\right) \\
=& \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}_{i_m} + \frac{N-M}{N-m}\left(\frac{\sum_{t=m}^{N}\mathcal{L}_{i_t}}{(N-m+1)} - \mathcal{L}_{i_m}\right),
\end{aligned}$$

pulling out the loss:

$$\begin{aligned}
=& \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}_{i_m}\left(1 - \frac{N-M}{N-m} + (N-M)\underbrace{\sum_{k=1}^{m}\frac{1}{(N-k)(N-k+1)}}_{=m/(N(N-m))}\right) \\
&+ \frac{1}{M}\sum_{t=M+1}^{N}\mathcal{L}_{i_t}(N-M)\underbrace{\sum_{k=1}^{M}\frac{1}{(N-k)(N-k+1)}}_{=M/(N(N-M))} \\
=& \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}_{i_m}\left(1 - \frac{N-M}{N-m} + \frac{N-M}{N-m}\frac{m}{N}\right) + \frac{1}{N}\sum_{t=M+1}^{N}\mathcal{L}_{i_t} \\
=& \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}_{i_m}\left(1 - \frac{N-M}{N-m}\left(\frac{N-m}{N}\right)\right) + \frac{1}{N}\sum_{t=M+1}^{N}\mathcal{L}_{i_t} \\
=& \frac{1}{N}\sum_{m=1}^{M} \mathcal{L}_{i_m} + \frac{1}{N}\sum_{t=M+1}^{N}\mathcal{L}_{i_t} \\
=& \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}_n \\
=& \hat{R}
\end{aligned}$$

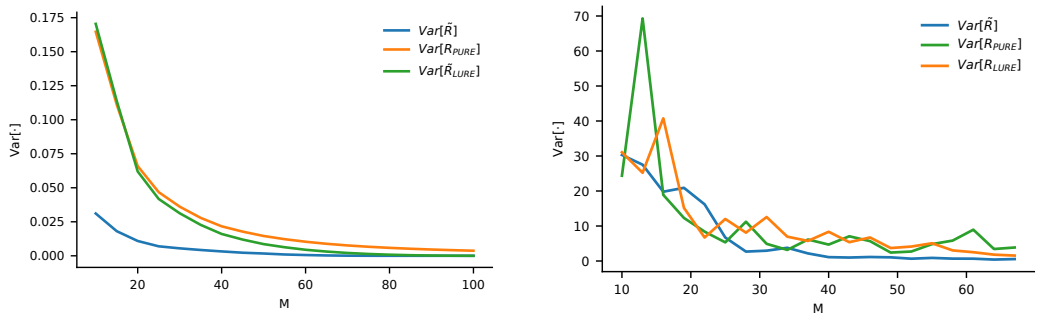
□

Remark 2. *The optimal proposal for estimating the gradient of the pool risk, $\nabla_\phi \hat{R}$, with respect to some scalar ϕ is instead*[1]

$$q^{**}(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}}) = \left|\nabla_\phi \mathcal{L}_{i_m}\right| / \sum\nolimits_{n \notin i_{1:m-1}} \left|\nabla_\phi \mathcal{L}_n\right|.$$

Note that when taking gradients with respect to multiple variables, the optimal proposal for each will be different for each.

[1] One can, in principle, actually construct an exact estimator in this scenario as well with the TABI approach of Rainforth et al. (2020) by employing two separate proposals that target $\max(\nabla_\theta \hat{R}, 0)$ and $-\min(\nabla_\theta \hat{R}, 0)$ respectively, then taking the difference between the two resultant estimators.



(a) **Linear Regression** (b) **Bayesian Neural Network**

Figure 5: For linear regression (a) the biased estimator has the lowest variance, and $\tilde{R}_{\text{LURE}}$ improves on $\tilde{R}_{\text{PURE}}$. (b) But for the BNN the variances are more comparable, with $\tilde{R}_{\text{LURE}}$ the lowest.

C Experimental Details

C.1 Linear Regression

Our training dataset contains a small cluster of points near $x = -1$ and two larger clusters at $0 \leq x \leq 0.5$ and $1 \leq x \leq 1.5$, sampled proportionately to the 'true' data distribution. The data distribution from which we select data in a Rao-Blackwellised manner has a probability density function over x equal to:

$$P(\text{x} = X) = \begin{cases} 0.12 & -1.2 \leq \text{x} \leq -0.8 \\ 0.95 & 0.0 \leq \text{x} \leq 0.5 \\ 0.95 & 1.0 \leq \text{x} \leq 1.5 \end{cases}$$

while the distribution over y is then induced by:

$$\text{y} = \max(0, x) \cdot \left(|x|^{\frac{3}{2}} + \frac{\sin(20x)}{4} \right).$$

We set $N = 101$, where there are 5 points in the small cluster and 96 points in each of the other two clusters, and consider $10 \leq M \leq 100$. We actively sample points without replacement using a geometric heuristic that scores the quadratic distance to previously sampled points and then selects points based on a Boltzman distribution with $\beta = 1$ using the normalized scores.

Here, we also show in Figure 6 results that are collected using an epsilon-greedy acquisition proposal. The results are aligned with those from the other acquisition distribution we consider in the main body of the paper. This proposal selects the point that is has the highest total distance to all previously selected points with probability 0.9 and uniformly at random with probability $\epsilon = 0.1$. That is, the acquistion proposal is given by:

$$P(i_m = j; i_{1:m-1}, \mathcal{D}_{\text{pool}}) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{D}_{\text{pool}}|} & \arg\max_{j \notin \mathcal{D}_{\text{train}}} \sum_{k \in \mathcal{D}_{\text{train}}} |x_k - x_j| \\ \frac{\epsilon}{|\mathcal{D}_{\text{pool}}|} & \text{otherwise} \end{cases}$$

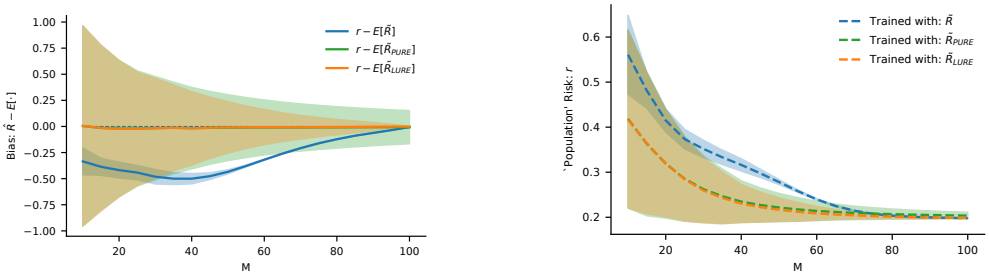
where of course $\mathcal{D}_{\text{train}}$ are the $i_{1:m-1}$ elements of $\mathcal{D}_{\text{pool}}$.

For all graphs we use 1000 trajectories with different random seeds to calculate error bars. Although, of course, each regression and scoring is deterministic, the acquistion distribution is stochastic.

Although the variance of the estimators can be inferred from Figure 2a, we also provide Figure 5a which displays the variance of the estimator directly.

C.2 Bayesian Neural Network

We train a Bayesian neural network using variational inference (Jordan et al., 1999). In particular, we use the radial Bayesian neural network approximating distribution (Farquhar et al., 2020). The details of the hyperparameters used for training are provided in Table 1.

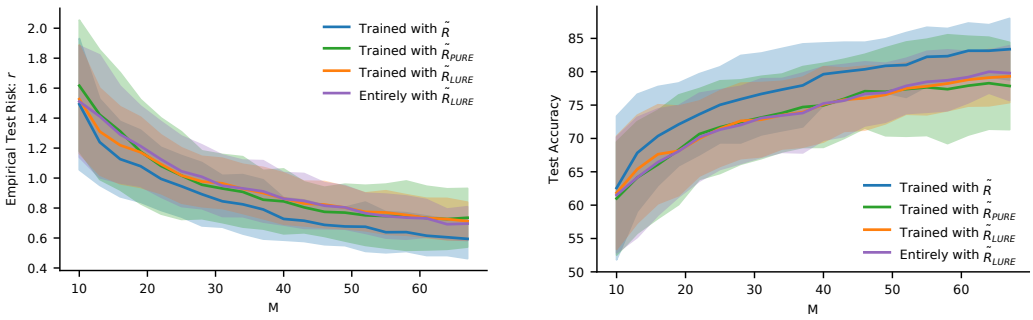


(a) **Bias (like Fig. 2a).** (b) **TestMSE (like Fig. 3a).**

Figure 6: Adopting an alternative proposal distribution—here an epsilon-greedy adaptation of a distance-based measure—does not change the overall picture for linear regression.

Hyperparameter	Setting description
Architecture	Convolutional Neural Network
Conv 1	1-16 channels, 5x5 kernel, 2x2 max pool
Conv 2	16-32 channels, 5x5 kernel, 2x2 max pool
Fully connected 1	128 hidden units
Fully connected 2	10 hidden units
Loss function	Negative log-likelihood
Activation	ReLU
Approximate Inference Algorithm	Radial BNN Variational Inference (Farquhar et al., 2020)
Optimization algorithm	Amsgrad (Reddi et al., 2018)
Learning rate	$5 \cdot 10^{-4}$
Batch size	64
Variational training samples	8
Variational test samples	8
Variational acquisition samples	100
Epochs per acquisition	up to 100 (early stopping patience=20), with 1000 copies of data
Starting points	10
Points per acquistion	1
Acquisition proposal distribution	$q(i_m; i_{1:m-1}, \mathcal{D}_{\text{pool}}) = \frac{e^{Ts_i}}{\sum e^{Ts_i}}$
Temperature: T	10,000
Scoring scheme: s	BALD (M.I. between θ and output distribution)
Variational Posterior Initial Mean	He et al. (2016)
Variational Posterior Initial Standard Deviation	$\log[1 + e^{-4}]$
Prior	$\mathcal{N}(0, 0.25^2)$
Dataset	Unbalanced MNIST
Preprocessing	Normalized mean and std of inputs.
Validation Split	1000 train points for validation
Runtime per result	2-4h
Computing Infrastructure	Nvidia RTX 2080 Ti

Table 1: Experimental Setting—Active MNIST.



(a) **Test loss** (b) **Test accuracy**

Figure 7: We contrast the effect of using $\tilde{R}_{\text{LURE}}$ throughout the entire acquisition procedure and training (rather than using the same acquisition procedure based on $\tilde{R}$ for all estimators). The purple test performance and orange are nearly identical, suggesting the result is not sensitive to this choice.

The unbalanced dataset is constructed by first noising 10% of the training labels, which are assigned random labels, and then selecting a subset of the training dataset such that the numbers of examples of each class is proportional to the ratio (1., 0.5, 0.5, 0.2, 0.2, 0.2, 0.1, 0.1, 0.01, 0.01)—that is, there are 100 times as many zeros as nines in the unbalanced dataset. (Figure 3f shows a version of this experiment which uses a balanced dataset instead, in order to make sure that any effects are not entirely caused by this design choice.) In fact, we took only a quarter of this dataset in order to speed up acquisition (since each model must be evaluated many times on each of the candidate datapoints to estimate the mutual information). 1000 validation points were then removed from this pool to allow early stopping. The remaining points were placed in $\mathcal{D}_{\text{pool}}$. We then uniformly selected 10 points from $\mathcal{D}_{\text{pool}}$ to place in $\mathcal{D}_{\text{train}}$. Adding noise to the labels and using an unbalanced dataset is designed to mimic the difficult situations that active learning systems are deployed on in practice, despite the relatively simple dataset. However, we used a simple dataset for a number of reasons. Active learning is very costly because it requires constant retraining, and accurately measuring the properties of estimators generally requires taking large numbers of samples. The combination makes using more complicated datasets expensive. In addition, because our work establishes a lower bound on architecture complexity for which correcting the active learning bias is no longer valuable, establishing that lower bound with MNIST is in fact a stronger result than showing a similar result with a more complex model.

The active learning loop then proceeds by:

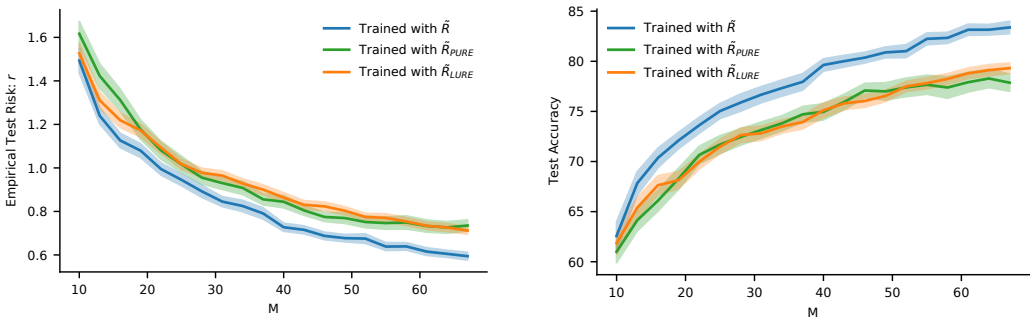
1. training the neural network on $\mathcal{D}_{\text{train}}$ using $\tilde{R}$;
2. scoring $\mathcal{D}_{\text{pool}}$;
3. sampling a point to be added to $\mathcal{D}_{\text{train}}$;
4. Every 3 points, we separately trained models on $\mathcal{D}_{\text{train}}$ using $\tilde{R}$, $\tilde{R}_{\text{PURE}}$, and $\tilde{R}_{\text{LURE}}$ and evaluate them.

This ensures that all of the estimators are on data collected under the same sampling distribution for fair comparison. As a sense-check, in Figures 7a and 7b we show an alternate version in which the first step trains with $\tilde{R}_{\text{LURE}}$ instead of $\tilde{R}$, and find that this does not have a significant effect on the results.

When we compute the bias of a fixed neural network in Figure 2b, we train a single neural network on 1000 points. We then sample evaluation points using the acquisition proposal distribution from the test dataset and evaluate the bias using those points.

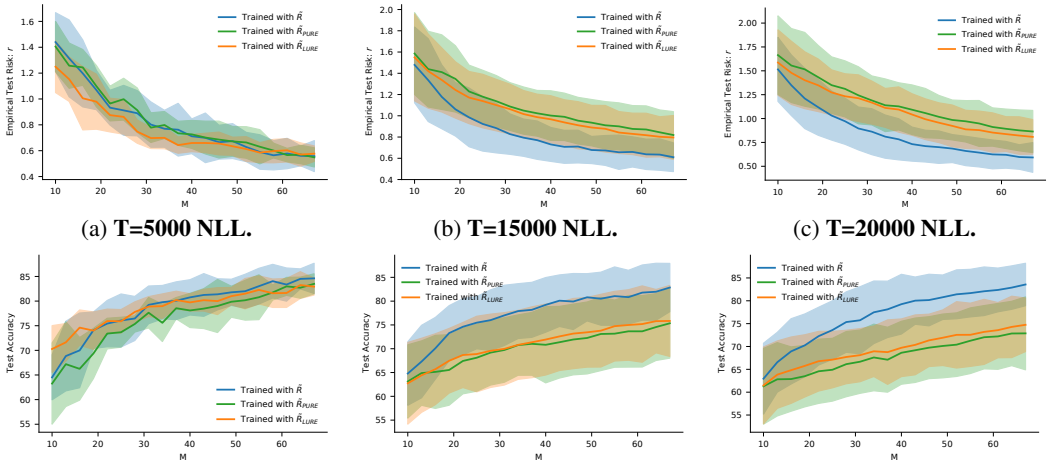
In Figures 8a and 8b we review the graphs shown in Figures 3b and 3c, this time showing standard errors in order to make clear that the biased $\tilde{R}$ estimator has better performance, while the earlier figures show that the performance is quite variable.

We considered a range of alternative proposal distributions. In addition to the Boltzman distribution which we used, we considered a temperature range between 1,000 and 20,000 finding it had relatively little effect. Higher temperatures correspond to more certainly picking the highest mutual information



(a) **Test loss** (b) **Test accuracy**

Figure 8: Versions of Figures 3b and 3c shown with standard errors (45 points) instead of standard deviations. This makes it clearer that the biased $\tilde{R}$ has better performance, even if only marginally so.



(a) **T=5000 NLL.** (b) **T=15000 NLL.** (c) **T=20000 NLL.**

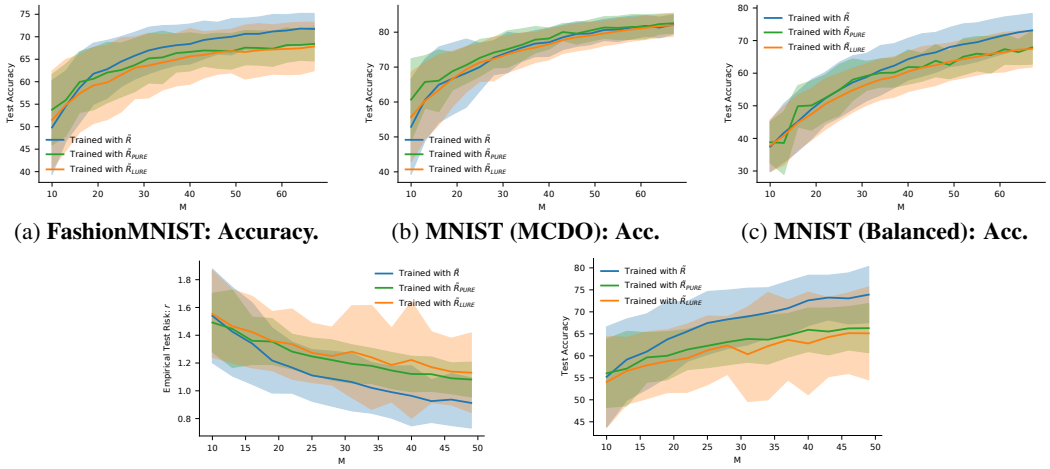
(d) **T=5000 Acc.** (e) **T=15000 Acc.** (f) **T=20000 Acc.**

Figure 9: Higher temperatures approach a deterministic acquisition function. These also tend to increase the variance of the risk estimator because the weight associated with unlikely points increases, when it happens to be selected. The overall pattern seems fairly consistent, however.

point, which approaches a deterministic proposal. We found that because the mutual information had to be estimated, and was itself a random variable, different trajectories still picked very different sets of points. However, for very high temperatures the estimators became higher variance, and for lower temperatures, the acquisition distribution became nearly uniform. In Figure 9 we show the results of networks trained with a variety of temperatures other than the 10,000 ultimately used. We also considered a proposal which was simply proportional to the scores, but found this was also too close to sampling uniformly for any of the costs or benefits of active learning to be visible.

We considered Monte Carlo dropout as an alternative approximating distribution (Gal & Ghahramani, 2015) (see Figures 3e and 10b). We found that the mutual information estimates were compressed in a fairly narrow range, consistent with the observation by Osband et al. (2018) that Monte Carlo dropout uncertainties do not necessarily converge unless the dropout probabilities are also optimized (Gal et al., 2017a). While this might be good enough when only the *relative* score is needed in order to calculate the argmax, for our proposal distribution we would ideally prefer to have good *absolute* scores as well. For this reason, we chose the richer approximate posterior distribution instead.

Last, we considered a different architecture, using a full-connected neural network with a single hidden layer with 50 units, also trained as a Radial BNN. This showed higher variance in downstream performance, but was broadly similar to the convolutional architecture (see Figures 10d and 10e).



(a) **FashionMNIST: Accuracy.** (b) **MNIST (MCDO): Acc.** (c) **MNIST (Balanced): Acc.**

(d) **MNIST (MLP): NLL** (e) **MNIST (MLP): Accuracy**

Figure 10: Further downstream performance experiments. (a)-(c) are partners to Figures 3d, 3e, and 3f. (d) and (e) show similar results for a smaller multi-layer perceptron (with one hidden layer of 50 units). In all cases the results broadly mirror the results in the main paper.

Reference	Application	Corrects Bias	Acknowledges Bias	Notes
(Sener & Savarese, 2018)				
(Shen et al., 2018)	✓			
(Beluch et al., 2018)				
(Haut et al., 2018)	✓			
(Sinha et al., 2019)				
(Siddhant & Lipton, 2018)	✓		✓	
(Ghosal et al., 2019)	✓			
(Yang et al., 2018)	✓			
(Yoo & Kweon, 2019)				
(Kirsch et al., 2019)				
(Huang et al., 2018)			✓	
(Wen et al., 2018)	✓			
(Chen et al., 2019)	✓			Discusses bias in $\mathcal{D}_{\text{pool}}$.
(Zhang & Lee, 2019)	✓			Discusses bias in $\mathcal{D}_{\text{pool}}$.
(Kellenberger et al., 2019)	✓			

Table 2: Existing applications of deep active learning rarely acknowledge the bias introduced by actively sampling points and do not, to the best of our knowledge, try to correct it.

D Deep Active Learning In Practice

In Table 2, we show an informal survey of highly cited papers citing Gal et al. (2017b), which introduced active learning to computer vision using deep convolutional neural networks. Across a range of papers including theory papers as well as applications ranging from agriculture to molecular science only two papers acknowledged the bias introduced by actively sampling and none of the papers took steps to address it. It is worth noting, though, that at least two papers motivated their use of active learning by observing that they expected their training data to already be unrepresentative of the population data and saw active learning as a way to address *that* bias. This does not quite work, unless you explicitly assume that the actively chosen distribution is more like the population distribution, but is an interesting phenomenon to observe in practical applications of active learning.