Loss-Driven Bayesian Active Learning

Zhuoyue Huang
University of Oxford

Freddie Bickford Smith
University of Oxford

Tom Rainforth
University of Oxford

Abstract

The central goal of active learning is to gather data that maximises downstream predictive performance, but popular approaches have limited flexibility in customising this data acquisition to different downstream problems and losses. We propose a rigorous loss-driven approach to Bayesian active learning that allows data acquisition to directly target the loss associated with a given decision problem. In particular, we show how any loss can be used to derive a unique objective for optimal data acquisition. Critically, we then show that any loss taking the form of a weighted Bregman divergence permits analytic computation of a central component of its corresponding objective, making the approach applicable in practice. In regression and classification experiments with a range of different losses, we find our approach reduces test losses relative to existing techniques.

1 Introduction

In machine learning we commonly use a loss function (or equivalently a reward or utility function) to quantify a model's performance (Murphy, 2022). Formally a loss function encodes preferences over the outcomes of a given decision (von Neumann & Morgenstern, 1947). It could for instance measure the accuracy of a prediction, the impact of a medical treatment on a patient's health, or the return on a financial investment.

The diversity of losses in use across the field reflects the variety of decision problems we want to solve. For example, predictive decision problems vary in the significance assigned to different prediction errors. In protein design (Notin et al, 2024) we might care more about accurately predicting a desirable property (eg, binding affinity) when it has a high value, and in safety-critical settings (Roy, 1952) we might want to penalise optimistic predictions more than pessimistic ones.

We argue that rigorously tailoring active learning to this breadth of possible decision problems is not possible with the approaches most popularly used in the field. In particular, while a range of data-acquisition objectives have been proposed (Li et al, 2024; Settles, 2012), in general these either have limited flexibility in targeting different downstream losses or rely on nested training of the downstream model within the acquisition function, meaning they can only be used with particular models or substantial approximations. For example, information-theoretic objectives (Bickford Smith et al, 2023; MacKay, 1992b) and Bayesian variants of popular variance-based objectives (Cohn, 1993; Cohn et al, 1994) are fixed in the losses they target (log loss and quadratic loss respectively), while popular error-based objectives (Roy & McCallum, 2001) allow for different losses but are heavily restricted in the models they can be practically used with. There is thus a pressing need for flexible, principled methods that directly optimise downstream loss in a range of decision problems without model restrictions.

We address this by proposing an explicitly loss-driven approach to active learning based on Bayesian decision theory (Berger, 1985; Ramsey, 1926; Savage, 1954). Specifically, we revisit the decision-theoretic foundations of Bayesian experimental design (DeGroot, 1962; Lindley, 1972; Raiffa & Schlaifer, 1961) to show how data utility can be formalised as the Bayesian expected loss of a Bayes-optimal downstream action informed by the acquired data. This then reveals that any model-loss pairing defines a unique objective whose minimiser achieves theoretically optimal data acquisition.

Objectives derived from our framework are not practically usable in the fully general case: they comprise an expectation of a minimisation of an inner expectation, which is rarely computationally viable to directly optimise. However, we show that any loss that can be written as a weighted Bregman divergence (Bregman, 1967) allows us to compute the inner minimisation an-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

alytically, such that the overall objective can be simplified to a form amenable to practical estimation and optimisation. Given that this broad class of losses includes many losses used in practice, this then yields a powerful and highly applicable class of loss-driven Bayesian active learning methods.

To help demonstrate the performance benefits of our approach, we introduce two concrete data-acquisition objectives with prediction-space weighting: one analogous to information-based objectives (Bickford Smith et al, 2023, 2024; MacKay, 1992a,b) and one analogous to Bayesian variants of variance-based objectives (Cohn, 1993; Cohn et al, 1994). On classification and regression problems these loss-matched objectives improve the corresponding (weighted) test losses, and a same-task comparison across downstream losses (quadratic vs Linex) highlights the cost of mismatch.

2 Background

Bayesian decision theory (Berger, 1985; Ramsey, 1926; Savage, 1954) provides a rigorous framework for choosing an action, $a \in \mathcal{A}$, under imperfect knowledge of a world state, $z \in \mathcal{Z}$. A decision-maker's loss function, $\ell : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$, encodes their preferences over outcomes, and their model, $p(z)$, encodes their beliefs about the world state. Any given action can be judged in terms of its Bayesian expected loss,

$$L(a) = \mathbb{E}_{p(z)}[\ell(z, a)] \,. \tag{1}$$

A Bayes-optimal action minimises this expected loss:

$$a^* = \arg\min_{a \in \mathcal{A}} L(a). \tag{2}$$

Bayesian experimental design (BED) (DeGroot, 1962, 1970; Lindley, 1956, 1972; Raiffa & Schlaifer, 1961; Raiffa, 1968) applies Bayesian decision theory to the problem of designing experiments. The design, ξ, of an abstractly defined experiment represents controllable aspects of the data generation, and our choice of ξ affects the data, d, we will observe. In the original formulation of BED (Lindley, 1956), designs are judged in terms of the expected information gain (EIG) in some quantity of interest, ψ, from observing d:

$$\mathrm{EIG}_\psi(\xi) = \mathbb{E}_{p(d;\xi)}[\mathrm{H}[p(\psi)] - \mathrm{H}[p(\psi|d;\xi)]] \,, \tag{3}$$

where $\mathrm{H}[\cdot]$ denotes Shannon entropy (Shannon, 1948). Here and elsewhere we use $p(a; b)$ to denote a distribution over a that depends on b without necessarily being the conditional distribution, $p(a|b)$, associated with an explicit joint distribution, $p(a, b)$.

BED more generally involves choosing a data policy, π_d, that minimises a Bayesian expected loss of the form

$$L_{\mathrm{BED}}(\pi_d) = \mathbb{E}_{p(d,\psi;\pi_d)}[\ell_d(\pi_d, d, \psi)] \,.$$

Here π_d could represent a fixed sequence of designs or a more complex decision-making procedure that adaptively selects designs based on the experiment history (Foster et al, 2021; Huan & Marzouk, 2016). Meanwhile ℓ_d is defined not in terms of downstream actions but in terms of the data policy and the data it yields, along with the quantity of interest. Determining what ℓ_d should be can be challenging; the predominant approach is to use $\ell_d(\pi_d, d, \psi) = \log p(\psi) - \log p(\psi|d; \pi_d)$, recovering the EIG (Rainforth et al, 2024).

Bayesian active learning (Bickford Smith et al, 2023, 2024; Gal et al, 2017; Houlsby et al, 2011; Kirsch et al, 2019; MacKay, 1992a,b) is the application of BED to sequentially selecting data to use in training a predictive model for outputs, $\tilde{y} \in \mathcal{Y}$ given inputs $\tilde{x} \in \mathcal{X}$. Typically each design, ξ, corresponds to choosing one or more inputs, $\xi \in \mathcal{X}^n$, and the data corresponds to the corresponding observed output label(s). This process is then iterated, updating the model with the new data before again choosing new inputs to label. Typically the inputs for labelling are selected from some large *pool* of unlabelled examples.

The most popular approaches use myopic information-theoretic data acquisition, each step of which involves choosing ξ to maximise $\mathrm{EIG}_\psi(\xi)$, namely the expected uncertainty reduction in ψ from performing a Bayesian update on ψ given the observed d, with uncertainty measured by Shannon entropy (Equation 3). If $\psi = \theta$ represents a set of stochastic model parameters then the EIG is known as the BALD score (Houlsby et al, 2011); if instead $\psi = (\tilde{x}, \tilde{y})$ represents a downstream input-output pair then we get the expected predictive information gain (EPIG; Bickford Smith et al, 2023).

The **variance-based objective** introduced by Cohn (1993) similarly corresponds to an expected reduction in uncertainty, but with uncertainty measured by variance rather than Shannon entropy and with generic belief updating on new data rather than a Bayesian update. With $p(\tilde{x})$ denoting beliefs over downstream inputs, the expected variance reduction (EVR) is

$$\mathrm{EVR}(\xi) = \mathbb{E}_{p(d;\xi)p(\tilde{x})}\left[\mathbb{V}_{p(\tilde{y};\tilde{x})}[\tilde{y}] - \mathbb{V}_{p(\tilde{y};\tilde{x},d)}[\tilde{y}]\right] .$$

The **expected future error** (EFE) framework introduced by Roy & McCallum (2001) is based around measuring the future expected classification loss of predictions on the unlabelled pool data, $\mathcal{X}_p$. Specifically, they consider retraining the model on a hypothetical new datapoint, (x, y), for each $x \in \mathcal{X}_p$ and all possible class labels, $y \in \mathcal{Y}$, evaluating some loss function for the updated model's predictions on all other points in the pool, and then choosing the input that gives the lowest average loss in expectation over possible labels.

That is, they choose the $x \in \mathcal{X}_p$ that minimises

$$\mathrm{EFE}(x) = \mathbb{E}_{p(y;x)}\left[\sum_{\tilde{x}\in\mathcal{X}_p} \frac{\mathbb{E}_{p(\tilde{y};\tilde{x},x,y)}[\ell(p(\tilde{y};\tilde{x},x,y),\tilde{y})]}{|\mathcal{X}_p|}\right],$$

where $p(y;x)$ is based on the current model, each $p(\tilde{y};\tilde{x},x,y)$ is a retrained model with the new hypothetical data, and ℓ is the assumed classification loss.

In principle, their framework is applicable to any downstream loss defined on the same space as their training data. However, it requires a nested retraining of the model for each possible (x, y), meaning it is not generally applicable in practice. Indeed, their specific implementations, which focus on log loss and zero-one loss, rely on using models with cheap incremental updates, along with additional approximations.

A **Bregman divergence** (Bregman, 1967) between elements in a convex set, $x, y \in \Omega \subseteq \mathbb{R}^K$, is defined as

$$D_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y\rangle,$$

where $\phi : \Omega \to \mathbb{R}$ is a differentiable and strictly convex potential function, and $\langle\cdot,\cdot\rangle$ denotes an inner product. It is nonnegative, typically asymmetric in x and y (and thus is not a metric), and vanishes if and only if $x = y$.

Two important examples of Bregman divergences are Mahalanobis distance and KL divergence (Kullback & Leibler, 1951). Specifically, setting $\Omega = \mathbb{R}^K$ and $\phi(x) = x^\top Ax$ for $A \succ 0$, yields $D_\phi(x, y) = (x - y)^\top A(x - y)$ (Banerjee et al, 2005b), which reduces to the squared error when $A = I_K$ is the K-dimensional identity matrix. Meanwhile, setting $\Omega = \Delta^{K-1}$ and $\phi(x) = \sum_{j=1}^K x_j \log x_j$ gives the KL divergence, $D_\phi(x, y) = \sum_{j=1}^K x_j(\log x_j - \log y_j)$, which reduces to the negative log likelihood (NLL), $-\log y_i$, when $x = e_i$ is the ith standard basis vector.

Notably the results we present apply not just to Bregman divergences between finite-dimensional vectors, as above, but also to Bregman divergences between functions (Frigyik et al, 2008). This is practically relevant in many machine-learning settings. For example, if we are working with probability densities then we require a functional Bregman divergence to extend the KL divergence from above to this setting (Csiszár & Matúš, 2012). We provide details in Appendix A.

In probabilistic prediction, the regret of a differentiable strictly proper scoring rule is exactly a Bregman divergence of the associated generalised entropy, so Bregman geometry is the canonical regret geometry of proper probabilistic evaluation. See Appendix B for further discussion on the links between proper scoring rules and Bregman divergences.

3 Loss-driven data acquisition

The approaches to Bayesian experimental design and active learning presented in Section 2 have a fundamental shortfall: they are difficult to customise to different decision problems that we might want to solve using the data we are acquiring. In particular, it is not immediately clear how to align the loss on acquired data with our ultimate goal of minimizing loss on "terminal" (downstream) actions.

To address this, we now revisit BED from the perspective of decision-making and prediction. Specifically, by revisiting the Bayesian decision theory that underlies BED, we identify a loss-driven approach to data acquisition that in the special case of a negative-log-likelihood loss is equivalent to EIG maximisation.

3.1 Bayes-optimal data acquisition

We consider the end-to-end problem of acquiring data and then using that data to take a terminal action. To this end, we first consider the optimal downstream action if data d is acquired, then work backwards to figure out the best way to gather data from this.

Assume we have some terminal loss function, $\ell : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$, that depends on our terminal action, $a \in \mathcal{A}$, and true world state, $z \in \mathcal{Z}$ (in principle the loss can also directly depend on (d, π_d), but we will omit such dependency for simplicity). Further, let our model's belief over z after it has observed some hypothetical data, d, be given by $p(z|d;\pi_d) \propto p(z)p(d|z;\pi_d)$, where $p(z)$ encodes our model's current beliefs and $p(d|z;\pi_d)$ is our predictive model for new data. The downstream Bayes-optimal action is now given by

$$a^* = \arg\min_{a\in\mathcal{A}} \mathbb{E}_{p(z|d;\pi_d)}[\ell(z, a)]\,. \tag{4}$$

The Bayesian expected loss associated with this Bayes-optimal action corresponds to a *generalised entropy* (Bickford Smith et al, 2025; Dawid, 1998):

$$h_\ell[p(z|d;\pi_d)] = \min_{a\in\mathcal{A}} \mathbb{E}_{p(z|d;\pi_d)}[\ell(z, a)]\,. \tag{5}$$

This provides a *loss-driven* measure of uncertainty on our posterior beliefs $p(z|d;\pi_d)$. It represents the best expected performance we can achieve given the data we have managed to acquire and the data policy we used. Better data provides more information about the state of the world, z, reducing our uncertainty and, in turn, allowing for better downstream actions.

Moreover, sequential application of Bayesian decision theory dictates that we should assume downstream actions are made Bayes-optimally when making earlier decisions to preserve coherence (Lindley, 1972). Thus,

from a Bayesian decision theory perspective, the generalised entropy in Equation 5 is *the* canonical measure we should use to assess how effective our data acquisition policy has been once the data has been observed.

From here it is easy to see that the Bayes-optimal data gathering policy for a given model-loss pairing and allowed space of policies Π_d is

$$\pi_d^* = \arg\min_{\pi_d \in \Pi_d} \mathrm{EPU}^{\ell}_{p(d,z;\pi_d)}$$
$$\text{where} \quad \mathrm{EPU}^{\ell}_{p(d,z;\pi_d)} = \mathbb{E}_{p(d;\pi_d)}[h_\ell[p(z|d;\pi_d)]] \tag{6}$$

with $p(d;\pi_d) = \mathbb{E}_{p(z)}[p(d|z;\pi_d)]$. Here $\mathrm{EPU}^{\ell}_{p(d,z;\pi_d)}$ denotes the *expected posterior uncertainty* (EPU) in z for a given loss function, ℓ, and a given joint model over d and z, $p(d,z;\pi_d)$. If our loss depends only on (z,a) and not directly on (π_d, d), then this is equivalent to the expected uncertainty reduction (EUR),

$$\mathrm{EUR}^{\ell}_{p(d,z;\pi_d)} = h_\ell[p(z)] - \mathrm{EPU}^{\ell}_{p(d,z;\pi_d)}.$$

Equivalent notions to the EPU and EUR have been considered in past work (Bickford Smith et al, 2025; Dawid, 1998; Huang et al, 2024; Neiswanger et al, 2022); the most recent of these relaxed the form of the future beliefs to be potentially non-Bayesian. While principled, the definition of the generalised entropy involves a minimisation, creating a nested problem where we must minimise the expectation of a minimisation of an inner expectation that is itself with respect to an intractable posterior. Using it as a practical active learning objective thus further requires a mechanism to get around this nested minimisation.

3.2 Using Bregman-divergence losses

Our key insight is now to show that the EPU defined in Equation 6 is in fact a viable objective for practical active learning approaches for a wide range of losses. Specifically, if our loss takes the form

$$\ell(z,a) = w(z) D_\phi(T(z), a), \tag{7}$$

where $w : \mathcal{Z} \to \mathbb{R}_+$ is a weighting function, D_ϕ is the Bregman divergence associated with some strictly convex differentiable potential function, $\phi : \Omega \to \mathbb{R}$, and $T : \mathcal{Z} \to \mathrm{dom}(\phi)$ is measurable, then the required inner minimisation can be performed analytically, leading to the following core result.[1]

Theorem 1. *Assume the terminal loss takes the form of a weighted Bregman divergence on measurable transformations of the world state as per Equation 7, and that* $\mathbb{E}_{q(z)}[w(z)]$, $\mathbb{E}_{q(z)}[w(z)T(z)]$ *and* $\mathbb{E}_{q(z)}[w(z)\phi(T(z))]$ *are finite for some generic distribution* $q(z)$. *If* $\mathcal{A} \subseteq \mathrm{ri}(\mathrm{dom}(\phi))$ *is convex and contains* $\mathbb{E}_{q_w(z)}[T(z)]$ *where* $q_w(z) = w(z)q(z)/\bar{w}_q$ *and* $\bar{w}_q = \mathbb{E}_{q(z)}[w(z)]$, *then the generalised entropy associated with this* q *is given by*

$$h^w_{\phi,T}[q(z)] := \min_{a \in \mathcal{A}} \mathbb{E}_{q(z)}[w(z) D_\phi(T(z), a)]$$
$$= \bar{w}_q \left(\mathbb{E}_{q_w(z)}[\phi(T(z))] - \phi(\mathbb{E}_{q_w(z)}[T(z)]) \right). \tag{8}$$

Assuming that $p(z|d;\pi_d)$ *satisfies the above restrictions of* q *for all* (d,π_d), *then the Bayes-optimal data gathering policy for our model-loss pairing is given by*

$$\pi_d^* = \arg\min_{\pi_d \in \Pi_d} \bar{w}\, \mathbb{E}_{p_w(d;\pi_d)} \Big[\mathbb{E}_{p_w(z|d;\pi_d)}[\phi(T(z))] - \phi(\mathbb{E}_{p_w(z|d;\pi_d)}[T(z)]) \Big], \tag{9}$$

where we define beliefs $p_w(d;\pi_d) = \bar{w}(d,\pi_d) p(d;\pi_d)/\bar{w}$, $p_w(z|d;\pi_d) = w(z)p(z|d;\pi_d)/\bar{w}(d,\pi_d)$, *and weights* $\bar{w}(d,\pi_d) = \mathbb{E}_{p(z|d;\pi_d)}[w(z)]$, $\bar{w} = \mathbb{E}_{p(z)}[w(z)]$.

This result (see Appendix C.1 for proof) means Equation 9 is now something that we can realistically target for data acquisition, as it no longer requires us to perform a nested optimisation within the acquisition function (though it is still in general a nested expectation (Rainforth et al, 2018)). The implications of this are considerable because Equation 7 generalises a number of standard predictive losses. In addition to the KL divergence (reducing to NLL) and Mahalanobis distance (reducing to squared error) mentioned in Section 2, some further examples are presented in Table 1.

Remark. *The Bayes-optimal action when using a loss of the form Equation 7 with constant weight function* w *is given by the posterior mean of* $T(z)$. *Moreover, this result goes both ways: under mild regularity conditions, if our Bayes act is a posterior mean, this itself implies that our loss function must be a Bregman divergence loss of the form of Equation 7 up to additive constant terms (Banerjee et al, 2005a).*

The above remark explains why Bregman-type losses are not only numerically convenient, but also a natural fit for constructing a loss-driven active learning framework: they align with mean-oriented decision semantics and provide the analytic structure needed for tractable acquisition. The inclusion of the weighting factor, $w(z)$, increases the flexibility of the framework, allowing us to target scenarios where the optimal act is not to use the mean under our beliefs because of asymmetry in losses incurred by different errors (e.g. penalising under-prediction more than over-prediction).

The transformed-space formulation also gives a unified treatment of finite categorical targets: taking $T(z) = e_z$ for a true class label, z, maps labels to the

[1] Our results generally extend to cases where the loss depends directly on (π_d, d), provided each possible (π_d, d) realisation produces a loss that takes the form in Equation 7 (potentially with different w, ϕ, and/or T).

Name	Example	$\mathcal{Z}$	$T(z)$	$\phi(z)$	$D_\phi(T(z), T(b))$
Squared error	Berger (1985)	$\mathbb{R}$	z	z^2	$(z-b)^2$
Box-Cox squared error	Box & Cox (1964)	$\mathbb{R}^+$	$(z^\lambda - 1)/\lambda, \lambda \neq 0$	z^2	$(z^\lambda - b^\lambda)^2/\lambda^2$
Linex	Zellner (1986)	$\mathbb{R}^+$	$\exp(-\alpha z), \alpha > 0$	$-\log z$	$\exp(\alpha(b-z)) - \alpha(b-z) - 1$

Table 1 Standard predictive losses can be expressed as a Bregman divergence in a transformed space, $D_\phi(T(z), T(b))$, where $z \in \mathcal{Z}$ is a world state, $b \in \mathcal{Z}$ is an in-space action, ϕ is a convex function and $T : \mathcal{Z} \to \text{dom}(\phi)$ is measurable.

simplex, so the Bayesian act is the class-probability vector, $\mathbb{E}_{p(z)}[e_z] = p \in \mathbb{R}^K$, and the one-hot case follows from the same computation rather than requiring a separate construction. For example, using $\phi(x) = \sum_{i=1}^K x_i \log x_i$ and $T(z) = e_z$ allows us to recover Shannon entropy as our h_ℓ and thus an EUR that corresponds to the EIG in z.

More broadly, the transformation enlarges the class of tractable Bregman-type losses because, for data acquisition, we only need the corresponding uncertainty, $h_\phi[\cdot]$, to evaluate the acquisition function, not necessarily a minimiser that lies in the original world space $\mathcal{Z}$. When such an in-space minimiser is required, i.e., $z, b \in \mathcal{Z}$, one can instead use the pullback loss $D_\phi(T(z), T(b))$, and, assuming T is injective and $T(\mathcal{Z})$ is convex, recover the explicit Bayesian act $b^* = T^{-1}(\mathbb{E}_{p(z)}[T(z)])$ (see Table 1 for examples).

3.3 Focusing on active learning

Now we consider an active-learning setting in which z represents an output variable of interest, $\pi_d = x$ represents an input for labelling and $d = y$ represents the label of that input. We introduce the notion of a context variable, c, that relates to the output we want to predict; it could for example represent a test input.

The setup we have considered so far implicitly allows for transductive learning (Vapnik, 1982), in which c is known upfront: c can be used to inform our joint beliefs over y and z, and it does not need to be explicitly denoted. Our focus is instead going to be on settings where c is unknown at the time of data acquisition but known at the time of choosing a terminal action.

Extending to this new setting simply requires us to define our joint model to incorporate the unknown c, leading to a new form of EPU. We define our joint model to be $p(c, y, z|x) = p(c)p(z|c)p(y|c, x, z)$ and, assuming $p(y|x, c) = p(y|x)$ for simplicity, this gives us

$$\text{EPU}^{\phi,T,w}_{p(c,y,z|x)} = \mathbb{E}_{p(c)p(y|x)}\left[h^w_{\phi,T}[p(z|c, x, y)]\right],$$

which is a function of the x that we want to choose. As noted in Section 3.1, we can equivalently minimise this EPU or maximise a corresponding EUR, given by

$$\text{EUR}^{\phi,T,w}_{p(c,y,z|x)} = \mathbb{E}_{p(c)}\left[h^w_{\phi,T}[p(z|c)]\right] - \text{EPU}^{\phi,T,w}_{p(c,y,z|x)},$$

which is the negative EPU plus a term constant in x.

It is worth noting a caveat around using these forms of EPU and EUR defined in terms of a single step of data acquisition. This follows the standard practice in Bayesian active learning to repeatedly use a single-step acquisition objective to choose a sequence of inputs for labelling (Section 2). However, it means it does not conform to true Bayesian optimality as per Equation 9, which would require us to go through the expensive process of fully planning ahead through all future data acquisitions as would be required, e.g. using deep adaptive design (Foster et al, 2021). Our results show it can nevertheless perform well and provide a practical loss-driven Bayesian active learning approach.

3.4 Two concrete objectives

We next use our setup to identify two new example objectives from our framework that are related to existing information-based and variance-based objectives. We obtain the first objective using a weighted NLL loss, produced by $T(z) = e_z$ and $\phi(x) = \sum_{i=1}^K x_i \log x_i$:

$$\ell_{\text{WNLL}}(z, a) = -w(z) \log a_z$$

where a_z denotes the probability mass assigned to z by the distribution that a represents. The resulting uncertainty for a generic $q(z|c)$ is $h_{\text{WNLL}}[q(z|c)] = \bar{w}(c)\text{H}[q_w(z|c)]$ where we define $\bar{w}(c)$ and $q_w(z|c)$ as in Equation 8 except with $q(z)$ replaced by $q(z|c)$. The corresponding EUR is a "weighted EPIG",

$$\text{EPIG}_w(x) = \mathbb{E}_{p(c)p(y|x)}[\bar{w}(c)\text{H}[p_w(z|c)] - \bar{w}(c, x, y)\text{H}[p_w(z|c, x, y)]]$$

which is equal to EPIG if $w(z) = 1$ for all z.

Next we use a weighted squared error:

$$\ell_{\text{WSE}}(z, a) = w(z)(z - a)^2.$$

Similar to before, our uncertainty is $h_{\text{WSE}}[q(z|c)] = \bar{w}(c)\mathbb{V}_{q_w(z|c)}[z]$, which leads to a "weighted EVR",

$$\text{EVR}_w(x) = \mathbb{E}_{p(c)p(y|x)}\left[\bar{w}(c)\mathbb{V}_{p_w(z|c)}[z] - \bar{w}(c, x, y)\mathbb{V}_{p_w(z|c,x,y)}[z]\right].$$

If, as above, we consider $w(z) = 1$ for all z then this objective is a Bayesian variant of the expected variance reduction proposed by Cohn (1993).

Choosing w in practice Our framework is agnostic to how the weight function w is obtained: w is simply a user-specified encoding of which values of z matter more for downstream accuracy. In practice, w may be hand-specified from domain knowledge, derived from an explicit cost model, or elicited from user preferences. When the downstream evaluation loss is already fixed, w should be chosen to match that loss.

3.5 Insights on prediction-space weighting

There are important links between the EPU produced by our weighted-Bregman losses and the EPU of corresponding unweighted losses under an alternative model, as explained in the following result.

Corollary 1. *Assume the generalised entropy terms are well defined. Then, with* $\bar{w}(c) = \mathbb{E}_{p(z|c)}[w(z)]$ *and* $p_w(z|c) = w(z)\,p(z|c)/\bar{w}(c)$,

$$\mathrm{EPU}^{\phi,T,w}_{p(z,y|c,x)} = \bar{w}(c)\,\mathrm{EPU}^{\phi,T}_{p_w(z|c)\,p(y|x,z,c)},$$

where the lack of the w *superscript in the second* EPU *is used to imply* $w(z) = 1, \forall z$. *Moreover, averaging over contexts gives*

$$\mathrm{EPU}^{\phi,T,w}_{p(c,y,z|x)} = \bar{w}\,\mathrm{EPU}^{\phi,T}_{p_w(c)\,p_w(z|c)\,p(y|x,z,c)}, \tag{10}$$

$$= \bar{w}\,\mathrm{EPU}^{\phi,T}_{p_w(z)\,p(c,y|x,z)}, \tag{11}$$

where $\bar{w} = \mathbb{E}_{p(c)}[\bar{w}(c)]$ *and* $p_w(c) = \bar{w}(c)\,p(c)/\bar{w}$. *The same identities hold with* EPU *replaced by* EUR.

This result (see Appendix C.2 for proof) makes precise what prediction-space weighting does to the active-learning objective: weighting the downstream loss by $w(z)$ is equivalent to evaluating the *unweighted* EPU under an alternative model where we have replaced the *marginal* prior on z, $p(z)$, with the weighted prior, $p_w(z) \propto p(z)w(z)$, while keeping the conditional distribution on all other variables, $p(c, y|x, z)$, fixed. Thus, if our original model was based on directly placing a prior on z and then a likelihood on data and contexts given z (e.g. for BALD, z corresponds to the model parameters and $c = \emptyset$), we can easily just adjust our prior instead of imposing any kind of weighting.

However, in many cases our prior on z is not specified directly but is the pushforward of some other distribution. For example, when using EPIG then our model is constructed by first defining $p(c)$, a prior on underlying parameters $p(\theta)$, and a shared predictive distribution for outputs given inputs parameterised by θ, $p_{\mathrm{pred}}(\mathrm{out}|\theta, \mathrm{in})$. The required model terms in Corollary 1 are then derived indirectly from these, namely $p(z|c) = \mathbb{E}_{p(\theta)}[p_{\mathrm{pred}}(\mathrm{out} = z|\theta, \mathrm{in} = c)]$, $p(z) = \mathbb{E}_{p(c)}[p(z|c)]$, $p(\theta|z, c) = p(\theta)p_{\mathrm{pred}}(\mathrm{out} = z|\theta, \mathrm{in} = c)/p(z|c)$, and $p(y|x, z, c) = \mathbb{E}_{p(\theta|z,c)}[p_{\mathrm{pred}}(\mathrm{out} = y|\theta, \mathrm{in} = x)]$. Here it is not clear in general (at least without solving an inverse problem) how to adjust $p(\theta)$ and $p(c)$ to induce a desired change from $p(z)$ to $p_w(z)$, particularly given we need to also leave $p(y|x, z, c)$ unchanged. The weighted EPU form is thus essential in allowing us to formulate our model–loss pairing in practice.

Equation 10 further shows that when we average over contexts, then the weighting on z induces a change in the *effective* context distribution to $p_w(c) \propto \bar{w}(c)\,p(c)$. For example, with EPIG, then it adjusts our original test-time input distribution $p(c)$ to increase the emphasis on inputs that have high expected weight under the context dependent prior $p(z|c)$. Given one will generally construct the model by defining $p(c)$ and then $p(z|c)$, this highlights an important pitfall: simply adjusting the distribution $p(z|c)$ fails to account for the impact the weights have on contexts as well.

An important further nuance to these results occurs when the model is *learned* from previous data using empirical risk minimisation, rather than being directly specified by the user. Here one might consider incorporating the weight into the loss on which the model is trained instead of into the acquisition function. That is, when training the model parameters, θ, we could incorporate a $w(z)$ factor into the loss to induce the required change from $p(z)$ to $p_w(z)$. While this will indeed actually have the desired effect on $p(z)$, it will also have the unwanted consequence of changing the distribution on $p(y|x, z, c)$. Namely, by changing $p(\theta)$ to incorporate our desired weighting, this in turn changes $p(\theta|z, c)$, and in turn $p(y|x, z, c)$ (noting that in general $p_{\mathrm{pred}}(\mathrm{out}|\theta, \mathrm{in})$ will itself be fixed). Thus, though this approach is in principle possible, it still requires the application of a weighting in the EPU (along with appropriately adjusting $p(c)$), this time to y to account for the fact that our data distribution no longer represents our beliefs for what data we will actually see.

4 Related work

Early decision-oriented work in the active-learning literature includes that of Margineantu (2005), who augmented standard predictive losses with input-dependent label-acquisition costs. Similar considerations of variable labelling cost—as well as other practical factors, such as choosing between multiple possible label sources—were explored by Donmez & Carbonell (2008), Kapoor et al (2007), Nguyen et al (2015) and Werling et al (2015). A bigger departure from standard practice was proposed by Saar-Tsechansky & Provost (2007), who shifted from predictive decision problems to more general problems. Later work by Javdani et al (2014), Sundin et al (2019) and Filstroff

et al (2024) was motivated by a similar emphasis on non-predictive decisions. Krishnamurthy et al (2019) focused on predictive decision problems but considered non-standard losses that place more weight on some label values than others. More recently, Hino & Eguchi (2023) used Bregman divergences to define new measures of committee disagreement in the context of query-by-committee data acquisition, while Tan et al (2023) used them to extend the "mean cost of uncertainty" framework (Yoon et al, 2013) in the particular case of classification. All of the aforementioned work on active learning is similar to ours in its emphasis on non-standard losses, and some of it shares technical elements with our work, but (to our knowledge) our framework enables the derivation of practical acquisition objectives distinct from those in past work.

Relevant foundational work in the experimental-design literature includes that of Dawid (1998), DeGroot (1962) and Lindley (1972), who used Bayesian decision theory to derive acquisition objectives. While principled, these approaches are generally computationally impractical. More recent work by Huang et al (2024) demonstrated practical ideas in modern contexts by instead using amortisation to try and approximate the inner minimisation. By contrast, our work avoids the need for amortisation or nested updating entirely by analytically minimising the Bregman divergence.

A complementary literature studies loss mismatches with an emphasis on learning from fixed data rather than acquiring new data. For example, "decision-focused learning" trains predictors end-to-end through downstream (including combinatorial) optimisers to improve decision quality (Wilder et al, 2019), and "predict then optimise" develops decision-calibrated surrogate losses with statistical guarantees for downstream optimisation (Elmachtoub & Grigas, 2022).

5 Experiments

Now we empirically assess the benefit of explicitly targeting the loss in a given predictive decision problem. Specifically we run active learning on a range of regression and classification problems, in each case evaluating the performance of an acquisition objective that targets the test loss and comparing this against an alternative objective. Code for reproducing our results is available at `github.com/Zhuoyue-Huang/loss-driven-bayesian-active-learning`. Additional practical details are in Appendices D and E.

5.1 Regression

We begin with four scenarios in which $z \in \mathbb{R}$ represents a real-valued output. For each scenario, we consider two losses and their corresponding acquisition objectives (Section 3.4): standard squared error (standard EVR) and weighted squared error (weighted EVR). Since one of these scenarios aligns with the original motivation for the Linex loss (Zellner, 1986), we additionally explore the use of Linex on that problem.

Method	SEL ↓	SEL_w ↓
Random	0.9238 ± 0.0940	343.4 ± 109.5
EVR	**0.3449** ± 0.0025	107.1 ± 1.971
EVR_w	1.5872 ± 0.1158	**72.06** ± 1.508

Table 2 Test losses in 1d regression. Here SEL denotes squared-error loss and SEL_w denotes a weighted variant. For both test losses, the corresponding acquisition objective (EVR for the former; EVR_w for the latter) achieves the lowest loss. We report mean ± SEM over 25 runs. Bold indicates the best mean for each loss.

5.1.1 Synthetic data

We begin with a scenario that can be easily visualised. Specifically we consider one-dimensional inputs, $x \in \mathbb{R}$, and a true function, $f(x)$, defined as $f^* = 2f_0 + 8f_{(2.5,0.5)} + 10f_{(7.5,0.25)} - 6f_{(-4.5,0.5)}$ where $f_0(x) = \sin(2x)$ is a sine wave and $f_{(\mu,\sigma)}$ is a Gaussian density function parametrised by mean μ and standard deviation σ. Observations follow $y = f(x) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.04)$. We consider $w(z) = \exp(z)$ as the weighting function of interest. This corresponds to predictive errors being more consequential for higher values of the output, z, being predicted.

Model & training We use exact Gaussian-process (GP) regression (Williams & Rasmussen, 2006) with zero mean, covariance $k(x, x') = \exp\left(-\frac{1}{2}(x - x')^2\right)$ and a Gaussian likelihood function that matches the true input-conditional output distribution.

Data acquisition In each run we start with 3 input-label pairs and acquire 25 additional pairs. Candidates inputs, context inputs (for calculating EVR and weighted EVR) and test inputs are evenly spaced in $[-8, 8]$, with 65, 49 and 97 points respectively. With fixed hyperparameters, the GP one-step posterior update is exact, which we exploit to compute our acquisition objectives without refitting.

Results In Figure 1 we see how our choice of loss shapes data acquisition: the standard EVR just prioritises output regions in which the model's predictive variance is highest, while the weighted EVR additionally incorporates a prioritisation of high-output regions (in Appendix F.1 we show how changing the weighting function leads to prioritising low-output regions). In Table 2 we see the importance of incorporating the test loss into data acquisition. For both standard and weighted squared error, the acquisition

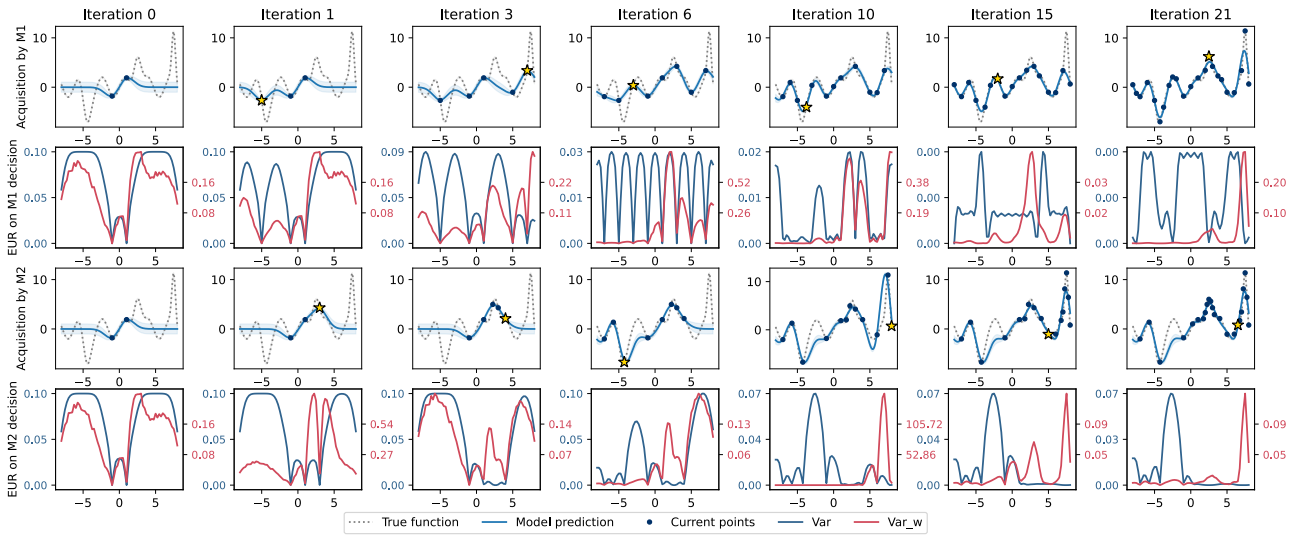

Figure 1 Expected variance reduction (EVR) vs. a variant with weighting $w(z) = \exp(z)$ (EVR_w). Row 1: prediction with data acquired by EVR. Row 2: values of EVR and EVR_w given the Row 1 training set (the maximiser is labelled next). Row 3: prediction with data acquired by EVR_w. Row 4: values of EVR and EVR_w given the Row 3 training set.

objective that targets the loss achieves much better average loss across test data.

5.1.2 UCI data

Next we shift to three scenarios based on datasets from the UCI repository (Dua & Graff, 2017): SLUMP ($N = 103$, $D = 7$; mix proportions $\to$ slump; Yeh, 2007), YACHT ($N = 308$, $D = 6$; hull descriptors $\to$ residuary resistance; Gerritsma et al, 1981) and ESTATE ($N = 414$, $D = 6$; location/time features $\to$ price per unit area; Yeh, 2018); N denotes the number of input-label pairs and D the input dimensionality. We consider $w(z) = \exp(-z)$ as the weighting function of interest, favouring accuracy in low-output regions; this is especially meaningful for YACHT, where low residuary resistance corresponds to lower required propulsive power and therefore improved performance.

Model & training Again we use a GP model. Here we use a constant prior mean, $m(x) = \text{const}$. The kernel is a sum of a linear (dot-product) term and a Matérn term with i.i.d. white noise. Additional modelling details are in Appendix E.2.

Data acquisition In each run we start with 10 input-label pairs and acquire 20 additional pairs. For reproducibility across methods, contexts and test inputs are drawn per dataset and resampled at each trial: we use fixed test sizes of 20, 60, and 80 for SLUMP, YACHT, and ESTATE, respectively, with the rest being pool candidates. We use the same acquisition methods as in Section 5.1.1. Estimation details are in Appendix D.2 and are independent of the kernel choice.

Results In Figure 2 we again see that the objective matched to the given test loss performs better than the unmatched objective. In all data scenarios, EVR_w attains the best SEL_w and EVR attains the best SEL.

5.1.3 Linex loss

So far we have compared EVR to its weighted variant. Here we compare EVR to an alternative objective that targets a Linex loss, as presented in Table 1. Linex is asymmetric, penalising large overestimation errors much more strongly than underestimation for $\alpha > 0$. We reuse our setup for ESTATE from before but here acquire 50 new input-label pairs rather than 20.

Results In Figure 3 we see that the previously observed benefit of targeted data acquisition holds convincingly when the comparison is between standard squared error and the Linex loss. Each acquisition objective is best under its own downstream loss, with the difference often constituting an order of magnitude.

5.2 Classification

Now we study three scenarios in which $z \in \{1, 2, \ldots, C\}$ is a class label using the additional UCI datasets VEHICLE ($C = 4$, $N = 946$, $D = 18$; silhouettes; Mowforth & Shepherd, 1993), LANDSAT ($C = 6$, $N = 6435$, $D = 36$; satellite imagery; Srinivasan, 1993) and VOWEL ($C = 11$, $N = 528$, $D = 10$; speech features; Deterding et al, 1988). Two classes per dataset are designated high-priority with $w(i) = 50$; the remaining classes use $w(i) = 1$. For each scenario, we consider two losses and their corresponding acquisition objectives (Section 3.4): NLL (EPIG) and weighted NLL (weighted EPIG).

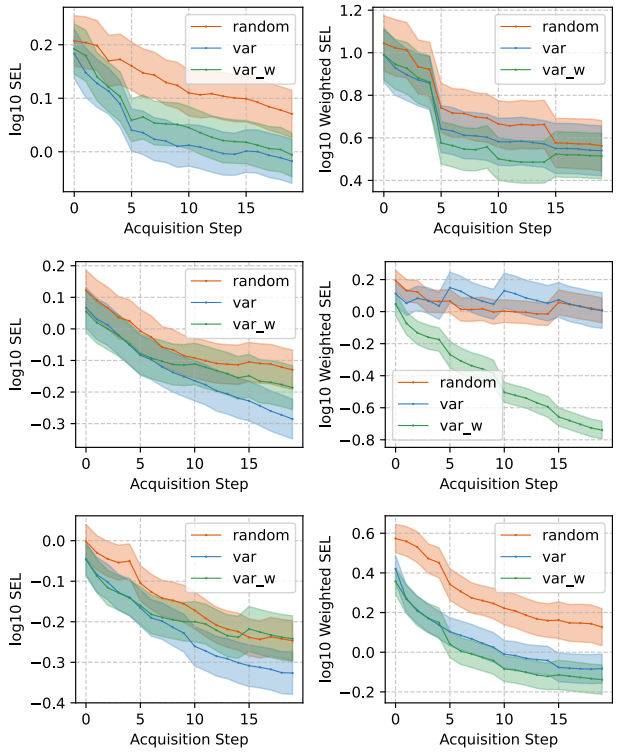

Figure 2 Top to bottom: performance (mean ± SEM) on SLUMP, YACHT and ESTATE with $w(z) = \exp(-z)$.

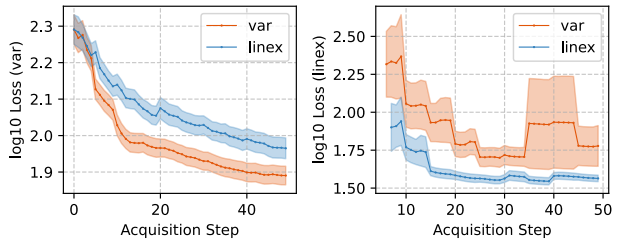

Figure 3 Evolution of performance (mean ± SEM) under downstream losses (SEL and Linex) on ESTATE.

Model & training We use random forests with 1,000 trees. Given a forest θ, each tree j yields class probabilities $\theta_{j,i}(x)$ for class i, so that $p(z = i \mid x, \theta_j) = \theta_{j,i}(x)$ and the forest defines $p_w(z \mid x) \propto w(z)p(z \mid x)$ by averaging over trees then weighting with class importance. The random forests are updated with standard (non-Bayesian) training; the acquisition computations use the Bayesian predictive posterior.

Data acquisition In each run we start with 5 input-label pairs per class, after which we acquire 100 additional pairs. For both acquisition-objective estimation and construction of the test set, we perform stratified subsampling with a fixed number per class: 45 examples per class for VEHICLE ($4 \times 45 = 180$ total), 200 per class for LANDSAT ($6 \times 200 = 1{,}200$), and 15 per class for VOWEL ($11 \times 15 = 165$).

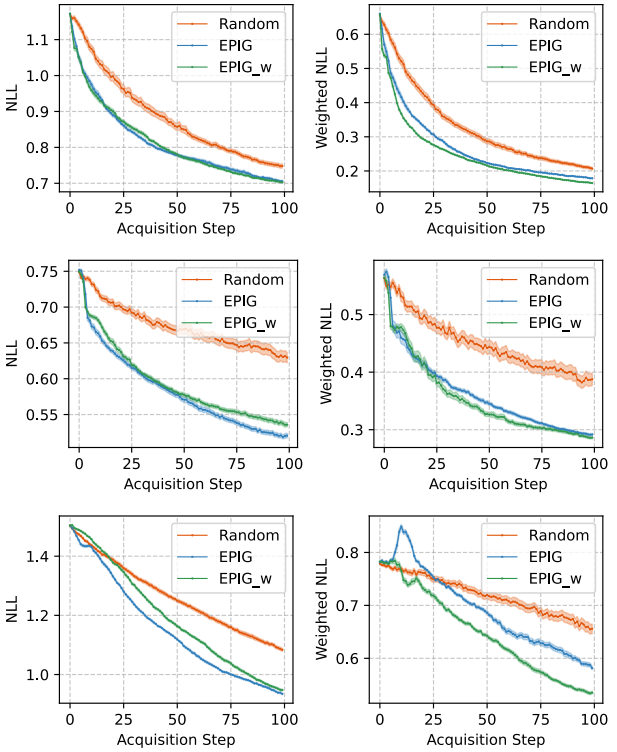

Figure 4 Top to bottom: mean ± SEM on VEHICLE ($w = [50, 1, 1, 50]$), LANDSAT ($w = [1, 1, 1, 1, 50, 50]$) and VOWEL ($w = [1, 1, 1, 1, 1, 1, 50, 50, 1, 1, 1]$).

Results With reference to the test losses in Figure 4 and class proportions in Figure 6, we can see that targeting the weighted loss leads us to prioritise high-weight classes, helping improve the test loss. The performance benefit we saw in regression problems therefore carries over to classification problems.

6 Conclusion

We have argued that popular active learning approaches do not generally account for the full range of decision problems that we want to target in practice. To address this we have revisited the decision-theoretic foundations of Bayesian experimental design and identified a principled, general approach that directly targets the loss of interest. We make this general approach practically applicable by showing that it can be analytically simplified when using weighted Bregman-divergence losses based on transformations of the world state. These losses are very general, corresponding to the scenario where the Bayes act is a (weighted) posterior mean, which is true in many practical prediction and estimation problems. We have further provided two example realisations of our framework, in the form of weighted variants of the EPIG and EVR acquisition functions that allow us to prioritise predictive performance over specific classes or regions.

Acknowledgements

ZH is supported by the EPSRC CDT in Statistics and Machine Learning (EP/Y034813/1). FBS is supported by the EPSRC Probabilistic AI Hub (EP/Y028783/1). TR is supported by the EPSRC grant EP/Y037200/1.

References

Banerjee, Guo, & Wang (2005a). On the optimality of conditional expectation as a Bregman predictor. *Transactions on Information Theory*.

Banerjee, Merugu, Dhillon, & Ghosh (2005b). Clustering with Bregman divergences. *Journal of Machine Learning Research*.

Berger (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer.

Bickford Smith, Foster, & Rainforth (2024). Making better use of unlabelled data in Bayesian active learning. *International Conference on Artificial Intelligence and Statistics*.

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

Bickford Smith, Kossen, Trollope, van der Wilk, Foster, & Rainforth (2025). Rethinking aleatoric and epistemic uncertainty. *International Conference on Machine Learning*.

Box & Cox (1964). An analysis of transformations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*.

Bregman (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*.

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

Cohn, Ghahramani, & Jordan (1994). Active learning with statistical models. *Conference on Neural Information Processing Systems*.

Cressie (1993). *Statistics for Spatial Data (Revised Edition)*. Wiley.

Csiszár & Matúš (2012). Generalized minimizers of convex integral functionals, Bregman distance, Pythagorean identities. *Kybernetika*.

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

Dawid (1998). Coherent measures of discrepancy, uncertainty and dependence, with applications to Bayesian predictive experimental design. Technical report, University College London.

Dawid (2007). The geometry of proper scoring rules. *Annals of the Institute of Statistical Mathematics*.

Dawid & Musio (2014). Theory and applications of proper scoring rules. *Metron*.

DeGroot (1962). Uncertainty, information, and sequential experiments. *Annals of Mathematical Statistics*.

DeGroot (1970). *Optimal Statistical Decisions*. John Wiley and Sons.

Deterding, Niranjan, & Robinson (1988). Connectionist Bench (Vowel Recognition - Deterding Data). UCI Machine Learning Repository. https://doi.org/10.24432/C58P4S.

Diggle & Ribeiro (2007). *Model-based Geostatistics*. Springer.

Donmez & Carbonell (2008). Proactive learning: cost-sensitive active learning with multiple imperfect oracles. *ACM Conference on Information and Knowledge Management*.

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

Elmachtoub & Grigas (2022). Smart "predict, then optimize". *Management Science*.

Filstroff, Sundin, Mikkola, Tiulpin, Kylmäoja, & Kaski (2024). Targeted active learning for Bayesian decision-making. *Transactions on Machine Learning Research*.

Foster, Ivanova, Malik, & Rainforth (2021). Deep adaptive design: amortizing sequential Bayesian experimental design. *International Conference on Machine Learning*.

Foster, Jankowiak, O'Meara, Teh, & Rainforth (2020). A unified stochastic gradient approach to designing Bayesian-optimal experiments. *International Conference on Artificial Intelligence and Statistics*.

Frigyik, Srivastava, & Gupta (2008). Functional Bregman divergence and Bayesian estimation of distributions. *IEEE Transactions on Information Theory*.

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

Gardner, Pleiss, Bindel, Weinberger, & Wilson (2018). GPyTorch: blackbox matrix-matrix Gaussian process inference with GPU acceleration. *Conference on Neural Information Processing Systems*.

Gerritsma, Onnink, & Versluis (1981). Yacht Hydrodynamics. UCI Machine Learning Repository. https://doi.org/10.24432/C5XG7R.

Gneiting & Raftery (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*.

h5py developers (2025). HDF5 for Python — h5py 3.14.0 documentation. https://docs.h5py.org/. Accessed: 2025-10-07.

Hall, Kay, & Titterington (1990). Asymptotically optimal difference-based estimation of variance in nonparametric regression. *Biometrika*.

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

Hino & Eguchi (2023). Active learning by query by committee with robust divergences. *Information Geometry*.

Hiriart-Urruty & Lemaréchal (2004). *Fundamentals of Convex Analysis*. Springer Science and Business Media.

Hoerl & Kennard (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*.

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

Huan & Marzouk (2016). Sequential Bayesian optimal experimental design via approximate dynamic programming. *arXiv*.

Huang, Guo, Acerbi, & Kaski (2024). Amortized Bayesian experimental design for decision-making. *Conference on Neural Information Processing Systems*.

Huber & Ronchetti (2009). *Robust Statistics*. Wiley.

Hübotter, Sukhija, Treven, As, & Krause (2024). Transductive active learning: theory and applications. *Conference on Neural Information Processing Systems*.

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

Javdani, Chen, Karbasi, Krause, Bagnell, & Srinivasa (2014). Near optimal Bayesian active learning for decision making. *International Conference on Artificial Intelligence and Statistics*.

Kapoor, Horvitz, & Basu (2007). Selective supervision: guiding supervised learning with decision-theoretic active learning. *International Joint Conference on Artificial Intelligence*.

Kelly, Longjohn, & Nottingham (2025). The UCI machine learning repository. https://archive.ics.uci.edu. General repository citation; accessed 2025-10-07.

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

Krishnamurthy, Agarwal, Huang, Daumé III, & Langford (2019). Active learning for cost-sensitive classification. *Journal of Machine Learning Research*.

Kullback & Leibler (1951). On information and sufficient statistics. *Annals of Mathematical Statistics*.

Li, Wang, Chen, Jiang, Ding, & Okumura (2024). A survey on deep active learning: recent advances and new frontiers. *IEEE Transactions on Neural Networks and Learning Systems*.

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

Lindley (1972). *Bayesian Statistics: a Review*. Society for Industrial and Applied Mathematics.

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

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

Margineantu (2005). Active cost-sensitive learning. *International Joint Conference on Artificial Intelligence*.

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

Mowforth & Shepherd (1993). Statlog (Vehicle Silhouettes). UCI Machine Learning Repository. https://doi.org/10.24432/C5HG6N.

Murphy (2022). *Probabilistic Machine Learning: an Introduction*. MIT Press.

Neiswanger, Yu, Zhao, Meng, & Ermon (2022). Generalizing Bayesian optimization with decision-theoretic entropies. *Conference on Neural Information Processing Systems*.

Nguyen, Wallace, & Lease (2015). Combining crowd and expert labels using decision theoretic active learning. *AAAI Conference on Human Computation and Crowdsourcing*.

Notin, Rollins, Gal, Sander, & Marks (2024). Machine learning for functional protein design. *Nature Biotechnology*.

Olson, La Cava, Orzechowski, Urbanowicz, & Moore (2017). PMLB: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*.

Ovcharov (2018). Proper scoring rules and Bregman divergence. *Bernoulli*.

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

Pedregosa, Varoquaux, Gramfort, et al (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*.

Pfau, D. (2013). A generalized bias-variance decomposition for bregman divergences. *Unpublished manuscript*.

Raiffa (1968). *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*. Addison-Wesley.

Raiffa & Schlaifer (1961). *Applied Statistical Decision Theory*. Division of Research, Harvard Business School.

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

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

Ramsey (1926). Truth and probability. *Studies in Subjective Probability*.

Rice (1984). Bandwidth choice for nonparametric regression. *The Annals of Statistics*.

Romano, Le, La Cava, et al (2021). PMLB v1.0: an open source dataset collection for benchmarking machine learning methods. *arXiv preprint*.

Roy (1952). Safety first and the holding of assets. *Econometrica*.

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

Saar-Tsechansky & Provost (2007). Decision-centric active learning of binary-outcome models. *Information Systems Research*.

Savage (1954). *The Foundations of Statistics*. John Wiley and Sons.

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

Shannon (1948). A mathematical theory of communication. *The Bell System Technical Journal*.

Srinivasan (1993). Statlog (Landsat Satellite). UCI Machine Learning Repository. https://doi.org/10.24432/C55887.

Stein (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer.

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

Tan, Du, & Buntine (2023). Bayesian estimate of mean proper scores for diversity-enhanced active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Tyler (2011). A short course on robust statistics. Lecture notes.

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

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

von Neumann & Morgenstern (1947). *Theory of Games and Economic Behavior*. Princeton University Press.

Werling, Chaganty, Liang, & Manning (2015). On-the-job learning with Bayesian decision theory. *Conference on Neural Information Processing Systems*.

Wilder, Dilkina, & Tambe (2019). Melding the data-decisions pipeline: decision-focused learning for combinatorial optimization. *AAAI Conference on Artificial Intelligence*.

Williams & Rasmussen (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Yeh (2007). Concrete Slump Test. UCI Machine Learning Repository. https://doi.org/10.24432/C5FG7D.

Yeh (2018). Real Estate Valuation. UCI Machine Learning Repository. https://doi.org/10.24432/C5J30W.

Yoon, Qian, & Dougherty (2013). Quantifying the objective cost of uncertainty in complex dynamical systems. *IEEE Transactions on Signal Processing*.

Zellner (1986). Bayesian estimation and prediction using asymmetric loss functions. *Journal of the American Statistical Association*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Functional Bregman divergences

For readability, the main text presents Bregman divergences and associated results in finite-dimensional vector notation. Many targets of interest in Bayesian active learning are more naturally viewed as elements in an infinite-dimensional function space (e.g., probability densities or latent functions). This appendix summarises *functional* Bregman divergences and records the functional analogue of the key optimality and uncertainty identities used in the paper.

Let $(\mathcal{X}, \Sigma, \mu)$ be a σ-finite measure space and let $L^p := L^p(\mathcal{X}, \mu)$ for some $1 \le p < \infty$. Let $\mathcal{K} \subset L^p$ be an open convex set and let $\Phi : \mathcal{K} \to \mathbb{R}$ be strictly convex and Gâteaux differentiable on $\mathcal{K}$. For $g \in \mathcal{K}$ and direction $h \in L^p$, write the (Gâteaux) derivative as

$$\delta\Phi(g; h) := \lim_{\epsilon \to 0} \frac{\Phi(g + \epsilon h) - \Phi(g)}{\epsilon},$$

whenever the limit exists (Hiriart-Urruty & Lemaréchal, 2004).

Definition 1 (Functional Bregman divergence (Frigyik et al, 2008))**.** For $f, g \in \mathcal{K}$, the functional Bregman divergence induced by Φ is

$$D_\Phi(f, g) := \Phi(f) - \Phi(g) - \delta\Phi(g; f - g).$$

A mild regularity condition is that for each fixed $g \in \mathcal{K}$ the map $h \mapsto \delta\Phi(g; h)$ defines a bounded linear functional on L^p (Frigyik et al, 2008). By L^p–L^q duality, there exists $\phi_g \in L^q$ (with $1/p + 1/q = 1$, and $q = \infty$ if $p = 1$) such that

$$\delta\Phi(g; h) = \int_{\mathcal{X}} \varphi_g(x)\, h(x)\, d\mu(x) \quad \text{for all } h \in L^p.$$

In this case,

$$D_\Phi(f, g) \;=\; \Phi(f) - \Phi(g) - \langle \varphi_g, f - g \rangle, \qquad \langle \varphi_g, f - g \rangle := \int_{\mathcal{X}} \varphi_g(x)\, (f(x) - g(x))\, d\mu(x). \tag{12}$$

Throughout we assume this mild regularity so that $\delta\Phi(g; \cdot) \in (L^p)^*$ and therefore admits the integral representation above, allowing us to identify φ_g with the (Gâteaux/Fréchet) gradient of Φ at g; for a convex Φ this coincides with the unique subgradient at g when differentiability holds. Readers may safely think of φ_g as the "gradient of Φ at g" for intuition.

All results in the paper stated for D_ϕ can be extended to functional Bregman divergence D_Φ under the replacements $\phi \mapsto \Phi$, $\nabla\phi(\cdot) \mapsto \varphi_{(\cdot)}$ (or $\delta\Phi(\cdot; \cdot)$), and $\langle \cdot, \cdot \rangle \mapsto \int \cdot \cdot \, d\mu$. In particular, the Bayes act under $D_\Phi(F, g)$ for a random function F is still the mean function: under mild integrability and assuming $\mathbb{E}[F] \in \mathcal{K}$,

$$\arg\min_{g \in \mathcal{K}} \mathbb{E}\big[D_\Phi(F, g)\big] = \mathbb{E}[F], \qquad \min_{g \in \mathcal{K}} \mathbb{E}\big[D_\Phi(F, g)\big] = \mathbb{E}[\Phi(F)] - \Phi(\mathbb{E}[F]),$$

a nonnegative Jensen gap (Frigyik et al, 2008).

Example: KL divergence between densities. Let $\mathcal{K}$ be a convex set of strictly positive probability densities on $\mathcal{X}$ (w.r.t. μ) and define $\Phi(p) := \int_{\mathcal{X}} p(x) \log p(x)\, d\mu(x)$. Then the induced functional Bregman divergence equals

$$D_\Phi(p, q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}\, d\mu(x) = \mathrm{KL}(p \| q),$$

where the normalisation $\int p\, d\mu = \int q\, d\mu = 1$ ensures cancellation of additive terms (Frigyik et al, 2008).

B Proper scoring rules and divergence representation

When the prediction target is itself a probability distribution (e.g., in probabilistic learning tasks), losses are naturally expressed as *scoring rules*. This appendix records the standard link between (strictly) proper scoring rules and (functional) Bregman divergences. This connection motivates our focus on Bregman-type losses: many

commonly used probabilistic losses, including log loss, arise as proper scoring rules whose regret is a Bregman divergence.

Let $(Z, \mathcal{Z})$ be a measurable space, $\mathcal{P}$ a convex class of probability laws on Z, and $S : \mathcal{P} \times Z \to \mathbb{R}$ a (loss-oriented) scoring rule. S is *(strictly) proper* if for all $p, q \in \mathcal{P}$,

$$\mathbb{E}_{p(z)}[S(p, z)] \leq \mathbb{E}_{p(z)}[S(q, z)]$$

with equality if and only if $q = p$. A differentiable scoring rule is strictly proper if and only if its *score regret* is a (functional) Bregman divergence (Gneiting & Raftery, 2007; Ovcharov, 2018):

$$\mathbb{E}_{p(z)}[S(q, z)] - \mathbb{E}_{p(z)}[S(p, z)] = \begin{cases} D_\phi(p, q), & p, q \in \Delta^{K-1} \\ D_\Phi(p, q), & p, q \in \mathcal{P}. \end{cases} \tag{13}$$

In finite-outcome problems this yields the *divergence representation* (Gneiting & Raftery, 2007):

$$\begin{aligned} S(q, z) &= D_\phi(e_z, q) + c(z) \\ &= -\phi(q) - \langle \nabla\phi(q),\, e_z - q \rangle + c(z), \end{aligned} \tag{14}$$

where scores are equivalent up to an additive $c(z)$. On general outcome spaces, the *Savage representation* (Gneiting & Raftery, 2007; Dawid & Musio, 2014) states that there exists a concave entropy H on $\mathcal{P}$ and a subgradient $h_q \in \partial H(q)$ such that

$$S(q, z) = H(q) + h_q(z) - \int h_q \, dq + c(z).$$

Writing $\Phi := -H$ (so Φ is convex) and choosing any $\varphi_q \in \partial\Phi(q)$ with $\varphi_q = -h_q$, we obtain the continuous analogue[2] of the discrete formula:

$$S(q, z) = -\Phi(q) - \varphi_q(z) + \mathbb{E}_{q(z)}[\varphi_q(z)] + c(z), \tag{15}$$

and consequently the score regret equals the functional Bregman divergence D_Φ (Ovcharov, 2018). Equation 13 makes clear that, when the inputs are *distributions*, one uses the appropriate Bregman geometry for the input type: vector-input D_ϕ on the simplex (discrete) and functional D_Φ on densities (continuous). Minimising expected score under the truth p is therefore equivalent to minimising (functional) Bregman divergence to p, uniquely solved by $q = p$.

Example: KL divergence between densities. The log score $S(q, z) = -\log q(z)$ is strictly proper and its regret is $\mathrm{KL}(p\|q)$. Equivalently, it corresponds to the convex functional $\Phi(p) = \int p \log p$ in Appendix A.

Note that we restrict the discussion to cases where the decision rule outputs a predictive distribution q, rather than a point prediction. The primitive loss is then a scoring rule $S(q, z)$. In this setting, the Bregman divergence appears at the level of expected score regret. It should therefore be distinguished from the pointwise Bregman loss $D_\phi(T(z), a)$ used in the main text for point or embedded-point prediction.

In the case of discrete probability distribution spaces, one can bridge between Bregman divergence for point prediction and distribution directly. If $z \in \{1, \ldots, K\}$ and $e_z \in \Delta^{K-1}$ is the one-hot encoding, then $\mathbb{E}_{p_n(z)}[e_z] = p_n(z)$, the full class-probability vector. By the same proof of evaluation-discrepancy identity in Proposition 2,

$$\mathbb{E}_{p_n(z)}[D_\phi(e_z, q)] - \mathbb{E}_{p_n(z)}[D_\phi(e_z, p_n(z))] = D_\phi(p_n, q)$$

so the expected pointwise Bregman loss on one-hot outcomes induces the Bregman divergence on the simplex under the same potential. Therefore, when the targets are in a discrete probability space, the optimal data-acquisition policy is equivalent to the minimiser of the expected proper scoring rule S (defined previously in Equations 14 and 15):

$$\pi_d^* = \arg\min_{\pi_d \in \Pi_d} \mathbb{E}_{p_n(z, d; \pi_d)}[S(p_n(\cdot|d; \pi_d), z)] \,.$$

[2] To match with the Bregman divergence formulation in Equation 12, intuitively $\langle \varphi_q, \delta_z - q \rangle := \varphi_q(z) - \int \varphi_q \, dq$. This extension sits slightly outside the L^p–L^q pairing used for functional Bregman divergences but is consistent with the general function–measure duality (Dawid, 2007).

C Proofs and derivations

C.1 Generalised entropy and expected predictive uncertainty

Proposition 1 (Banerjee et al, 2005a)**.** *Let $\mathcal{Z} \subseteq \mathbb{R}^K$ be an open convex set, $\phi : \mathcal{Z} \to \mathbb{R}$ be a strictly convex differentiable potential function, and $z \sim p(z)$ a random variable taking values in $\mathcal{Z}$ for which both $\mathbb{E}_{p(z)}[z]$ and $\mathbb{E}_{p(z)}[\phi(z)]$ are finite. With the assumption $\mathbb{E}_{p(z)}[z] \in \mathcal{Z}$, the minimiser of the expected Bregman divergence is the expectation of z:*

$$\arg\min_{a \in \mathcal{Z}} \mathbb{E}_{p(z)}[D_\phi(z, a)] = \mathbb{E}_{p(z)}[z]\,.$$

Theorem. *Assume the terminal loss takes the form of a weighted Bregman divergence on measurable transformations of the world state as per Equation 7, and that $\mathbb{E}_{q(z)}[w(z)]$, $\mathbb{E}_{q(z)}[w(z)T(z)]$ and $\mathbb{E}_{q(z)}[w(z)\phi(T(z))]$ are finite for some generic distribution $q(z)$. If $\mathcal{A} \subseteq \mathrm{ri}(\mathrm{dom}(\phi))$ is convex and contains $\mathbb{E}_{q_w(z)}[T(z)]$ where $q_w(z) = w(z)q(z)/\bar{w}_q$ and $\bar{w}_q = \mathbb{E}_{q(z)}[w(z)]$, then the generalised entropy associated with this q is given by*

$$h^w_{\phi,T}[q(z)] := \min_{a \in \mathcal{A}} \mathbb{E}_{q(z)}[w(z) D_\phi(T(z), a)] = \bar{w}_q \left(\mathbb{E}_{q_w(z)}[\phi(T(z))] - \phi(\mathbb{E}_{q_w(z)}[T(z)])\right).$$

Assuming that $p(z|d;\pi_d)$ satisfies the above restrictions of q for all (d, π_d), then the Bayes-optimal data gathering policy for our model-loss pairing is given by

$$\pi^*_d = \arg\min_{\pi_d \in \Pi_d} \; \bar{w}\, \mathbb{E}_{p_w(d;\pi_d)} \Big[\mathbb{E}_{p_w(z|d;\pi_d)}[\phi(T(z))] - \phi(\mathbb{E}_{p_w(z|d;\pi_d)}[T(z)]),$$

where we define beliefs $p_w(d;\pi_d) = \bar{w}(d,\pi_d)p(d;\pi_d)/\bar{w}$ and $p_w(z|d;\pi_d) = w(z)p(z|d;\pi_d)/\bar{w}(d,\pi_d)$, and weights $\bar{w}(d,\pi_d) = \mathbb{E}_{p(z|d;\pi_d)}[w(z)]$ and $\bar{w} = \mathbb{E}_{p(z)}[w(z)]$.

Proof. With a change of belief, we have

$$\mathbb{E}_{q(z)}[w(z) D_\phi(T(z), a)] = \bar{w}_q\, \mathbb{E}_{q_w(z)}[D_\phi(T(z), a)]\,.$$

As $\bar{w}$ is a constant, by Proposition 1, the minimiser of $\bar{w}_q\, \mathbb{E}_{q_w(z)}[D_\phi(T(z), a)]$ is $a^* = \mathbb{E}_{q_w(z)}[T(z)]$. Therefore, the corresponding minimal value is

$$\begin{aligned}
h^w_\phi[q(z)] &= \bar{w}_q \mathbb{E}_{q_w(z)}[D_\phi(T(z), a^*)] \\
&= \bar{w}_q \left(\mathbb{E}_{q_w(z)}[\phi(T(z)) - \phi(a^*) - \langle \nabla\phi(a^*), T(z) - a^*\rangle]\right) \\
&= \bar{w}_q \left(\mathbb{E}_{q_w(z)}[\phi(T(z))] - \phi(a^*) - \mathbb{E}_{q_w(z)}[\langle \nabla\phi(a^*), T(z)\rangle] + \langle \nabla\phi(a^*), a^*\rangle\right) \\
&= \bar{w}_q \left(\mathbb{E}_{q_w(z)}[\phi(T(z))] - \phi(a^*)\right).
\end{aligned}$$

Under the predictive distribution $p(z|d;\pi_d)$, the expected predictive uncertainty (6) can be expressed as

$$\begin{aligned}
\mathrm{EPU}^{\phi,T,w}_{p(d,z;\pi_d)} &= \mathbb{E}_{p(d;\pi_d)}\left[h^w_\phi[p(z|d;\pi_d)]\right] \\
&= \mathbb{E}_{p(d;\pi_d)}\left[\bar{w}(d,\pi_d)\left(\mathbb{E}_{p_w(z|d;\pi_d)}[\phi(T(z))] - \phi(\mathbb{E}_{p_w(z|d;\pi_d)}[T(z)])\right)\right] \\
&= \bar{w}\, \mathbb{E}_{p_w(d;\pi_d)}\Big[\mathbb{E}_{p_w(z|d;\pi_d)}[\phi(T(z))] - \phi(\mathbb{E}_{p_w(z|d;\pi_d)}[T(z)])\Big],
\end{aligned}$$

which leads to the optimal policy $\pi^*_d = \arg\min_{\pi_d \in \Pi_d} \mathrm{EPU}^{\phi,T,w}_{p(d,z;\pi_d)}$ as the form required. □

C.2 Expected posterior uncertainty with weighted divergence

Corollary. *Assume the generalised entropy terms are well defined. Then, with $\bar{w}(c) = \mathbb{E}_{p(z|c)}[w(z)]$ and $p_w(z|c) = w(z)\, p(z|c)/\bar{w}(c)$,*

$$\mathrm{EPU}^{\phi,T,w}_{p(z,y|c,x)} = \bar{w}(c)\, \mathrm{EPU}^{\phi,T}_{p_w(z|c)\, p(y|x,z,c)},$$

where the lack of the w superscript in the second EPU *is used to imply $w(z) = 1, \forall z$. Moreover, averaging over contexts gives*

$$\mathrm{EPU}^{\phi,T,w}_{p(c,y,z|x)} = \bar{w}\,\mathrm{EPU}^{\phi,T}_{p_w(c)\,p_w(z|c)\,p(y|x,z,c)}, \tag{16}$$

$$= \bar{w}\,\mathrm{EPU}^{\phi,T}_{p_w(z)\,p(c,y|x,z)}, \tag{17}$$

where $\bar{w} = \mathbb{E}_{p(c)}[\bar{w}(c)]$ and $p_w(c) = \bar{w}(c)\,p(c)/\bar{w}$. The same identities hold with EPU *replaced by* EUR.

Proof. The expected posterior uncertainty with weighted divergence can be expanded as

$$\begin{aligned}
\mathrm{EPU}^{\phi,T,w}_{p(z,y|c,x)} &= \mathbb{E}_{p(y|x,c)}\left[h^{w}_{\phi,T}\left[p(z \mid c,y,x)\right]\right] \\
&= \mathbb{E}_{p(y|x,c)\,p(z|c,y,x)}\left[w(z) D_\phi\left(T(z), \mathbb{E}_{p_w(z|c,y,x)}[T(z)]\right)\right] \\
&= \mathbb{E}_{p(z|c)\,p(y|x,z,c)}\left[w(z) D_\phi\left(T(z), \mathbb{E}_{p_w(z|c,y,x)}[T(z)]\right)\right] \\
&= \mathbb{E}_{p(z|c)}[w(z)] \cdot \mathbb{E}_{p_w(z|c)\,p(y|x,z,c)}\left[D_\phi\left(T(z), \mathbb{E}_{p_w(z|c,y,x)}[T(z)]\right)\right].
\end{aligned}$$

Note that $p_w(z \mid c,y,x)$ can also be expressed by the Bayes rule,

$$\begin{aligned}
p_w(z \mid c,y,x) &= \frac{w(z)}{\mathbb{E}_{p(z|c,y,x)}\left[w(z)\right]} \cdot p(z \mid c,y,x) \\
&= \frac{w(z)}{\mathbb{E}_{p(z|c)}\left[\frac{p(y|x,z,c)}{\mathbb{E}_{p(z'|c)}\left[p(y|x,z',c)\right]}\,w(z)\right]} \cdot \frac{p(z \mid c)\,p(y \mid x,z,c)}{\mathbb{E}_{p(z'|c)}\left[p(y \mid x,z',c)\right]} \\
&= \frac{w(z)}{\mathbb{E}_{p(z|c)}\Big[p(y \mid x,z,c)\,w(z)\Big]} \cdot p(z \mid c)\,p(y \mid x,z,c) \\
&= \frac{w(z)\,p(z \mid c)}{\mathbb{E}_{p(z|c)}\left[w(z)\right]} \cdot \frac{p(y \mid x,z,c)}{\mathbb{E}_{p_w(z|c)}\left[p(y \mid x,z,c)\right]} \\
&= p_w(z \mid c) \cdot \frac{p(y \mid x,z,c)}{\mathbb{E}_{p_w(z|c)}\left[p(y \mid x,z,c)\right]},
\end{aligned} \tag{18}$$

where the reweighted marginal likelihood over y is shown in the form of

$$q(y \mid x,c) \coloneqq \mathbb{E}_{p_w(z|c)}[p(y \mid x,z,c)].$$

In this case, we can further simplify the EPU as

$$\begin{aligned}
\mathrm{EPU}^{\phi,T,w}_{p(z,y|c,x)} &= \mathbb{E}_{p(z|c)}[w(z)] \cdot \mathbb{E}_{q(y|x,c)\,p_w(z|c,y,x)}\left[D_\phi\left(T(z), \mathbb{E}_{p_w(z|c,y,x)}[T(z)]\right)\right] \\
&= \mathbb{E}_{p(z|c)}[w(z)] \cdot \mathbb{E}_{q(y|x,c)}\left[h_{\phi,T}\left[p_w(z \mid c,y,x)\right]\right] \\
&= \mathbb{E}_{p(z|c)}[w(z)] \cdot \mathrm{EPU}^{\phi,T}_{q(y|x,c)\,p_w(z|c,y,x)} \\
&= \bar{w}(c)\,\mathrm{EPU}^{\phi,T}_{p_w(z|c)\,p(y|x,z,c)}.
\end{aligned}$$

Averaging over contexts gives

$$\begin{aligned}
\mathrm{EPU}^{\phi,T,w}_{p(z,y,c|x)} &= \mathbb{E}_{p(c)}\left[\bar{w}(c) \cdot \mathrm{EPU}^{\phi,T}_{p_w(z|c)\,(y|x,z,c)}\right] \\
&= \mathbb{E}_{p(c)}[\bar{w}(c)] \cdot \mathbb{E}_{p(c)}\left[\frac{\bar{w}(c)}{\mathbb{E}_{p(c)}[\bar{w}(c)]} \cdot \mathrm{EPU}^{\phi,T}_{p_w(z|c)\,p(y|x,z,c)}\right] \\
&= \mathbb{E}_{p(c)}[\bar{w}(c)] \cdot \mathbb{E}_{p_w(c)}\left[\mathrm{EPU}^{\phi,T}_{p_w(z|c)\,p(y|x,z,c)}\right] \\
&= \bar{w}\,\mathrm{EPU}^{\phi,T}_{p_w(c)\,p_w(z|c)\,p(y|x,z,c)},
\end{aligned}$$

where the joint weighted distribution can also be expressed as

$$\begin{aligned} p_w(c)\, p_w(z|c)\, p(y|x,z,c) &= \frac{\bar{w}(c)}{\bar{w}}\, p(c) \cdot \frac{w(z)}{\bar{w}(c)}\, p(z|c) \cdot p(y|x,z,c) \\ &= \frac{w(z)}{\mathbb{E}_{p(c)\,p(z|c)}[w(z)]}\, p(z)\, p(c|z)\, p(y|x,z,c) \\ &= p_w(z)\, p(y,c|x,z). \end{aligned}$$

As $h^w_{\phi,T}[p(z|c)] = \bar{w}(c) h_{\phi,T}[p_w(z|c)]$ and $\mathbb{E}_{p(c)}\left[h^w_{\phi,T}[p(z|c)]\right] = \bar{w}\,\mathbb{E}_{p_w(c)}\left[h_{\phi,T}[p_w(z|c)]\right]$, we have that

$$\begin{aligned} \mathrm{EUR}^{\phi,T,w}_{p(y,z|x,c)} &= h^w_{\phi,T}[p(z|c)] - \mathrm{EPU}^{\phi,T,w}_{p(y,z|x,c)} \\ &= \bar{w}(c)\left(h_{\phi,T}[p_w(z|c)] - \mathrm{EPU}^{\phi,T}_{p_w(z|c)\,p(y|x,z,c)}\right) \\ &= \bar{w}(c)\,\mathrm{EUR}^{\phi,T}_{p_w(z|c)\,p(y|x,z,c)} \\ \mathrm{EUR}^{\phi,T,w}_{p(c,y,z|x)} &= \mathbb{E}_{p(c)}\left[h^w_{\phi,T}[p(z|c)]\right] - \mathrm{EPU}^{\phi,T,w}_{p(c,y,z|x)} \\ &= \bar{w}\left(\mathbb{E}_{p_w(c)}\left[h_{\phi,T}[p_w(z|c)]\right] - \mathrm{EPU}^{\phi,T}_{p_w(c)\,p_w(z|c)\,p(y|x,z,c)}\right) \\ &= \bar{w}\,\mathrm{EUR}^{\phi,T}_{p_w(c)\,p_w(z|c)\,p(y|x,z,c)} \\ &= \bar{w}\,\mathrm{EUR}^{\phi,T}_{p_w(z)\,p(y,c|x,z)}. \end{aligned}$$

□

C.3 Decomposition of evaluation discrepancy

In practice we often evaluate a model against an *external* system $p_{\mathrm{eval}}(z)$ (a data source, simulator, sensor, or benchmark), which need not coincide with the model's predictive $p(z)$. Following the decision-theoretic view of externally grounded evaluation, the relevant score is the expected loss of the Bayes action under p, taken with respect to p_{eval} (Bickford Smith et al, 2025, Sec. 5.4). For Bregman losses, this score admits a clean two-term decomposition.

Proposition 2 (Evaluation-discrepancy decomposition under Bregman loss)**.** *Let D_ϕ be the Bregman divergence induced by a strictly convex differentiable ϕ. Then*

$$\mathbb{E}_{p_{\mathrm{eval}}}[D_\phi(z,\, \mathbb{E}_p[z])] = \underbrace{D_\phi(\mathbb{E}_{p_{\mathrm{eval}}}[z],\, \mathbb{E}_p[z])}_{\text{estimation error}} + \underbrace{\mathbb{E}_{p_{\mathrm{eval}}}[D_\phi(z,\, \mathbb{E}_{p_{\mathrm{eval}}}[z])]}_{\text{irreducible dispersion}}.$$

The left-hand side of Proposition 2 (see Appendix C.3 for a proof) is the *expected Bregman loss* of acting with the model's Bayes action when reality is generated by p_{eval}. The right-hand side separates: (i) a *reducible* term—how far the model's Bayes act $\mathbb{E}_p[z]$ is from the evaluation Bayes act $\mathbb{E}_{p_{\mathrm{eval}}}[z]$ in the geometry of ϕ; and (ii) an *irreducible* term—the Jensen gap of p_{eval} under ϕ, i.e., the data dispersion intrinsic to the evaluation source. Thus, improvement under an external metric comes *only* by shrinking the first term; the second is fixed by p_{eval} and the chosen loss. This generalises the evaluation identities in Bickford Smith et al (2025) from squared and log loss to *any* Bregman divergence, unifying bias-variance and cross-entropy decompositions within Bregman-geometric framework within one decision-theoretic statement.

Proof. Following Bickford Smith et al (2025), define the evaluation discrepancy

$$d(p, p_{\mathrm{eval}}) = \mathbb{E}_{p_{\mathrm{eval}}}[\ell(z, a^*) - \ell(z, a_{\mathrm{eval}})],$$

where a^* and a_{eval} are the minimisers of expected loss over p and p_{eval}, respectively. When the loss is in the form of a Bregman divergence, we can further reduce the expression as

$$\begin{aligned} d(p, p_{\mathrm{eval}}) &= \mathbb{E}_{p_{\mathrm{eval}}}\left[D_\phi\left(z, \mathbb{E}_p(z)\right) - D_\phi\left(z, \mathbb{E}_{p_{\mathrm{eval}}}(z)\right)\right] \\ &= \mathbb{E}_{p_{\mathrm{eval}}}\left[\phi(z) - \phi\left(\mathbb{E}_{p(z)}[z]\right) - \left\langle \nabla\phi\left(\mathbb{E}_{p(z)}[z]\right), z - \mathbb{E}_{p(z)}[z]\right\rangle\right] - \left(\mathbb{E}_{p_{\mathrm{eval}}}\left[\phi(z)\right] - \phi\left(\mathbb{E}_{p_{\mathrm{eval}}}[z]\right)\right) \\ &= \mathbb{E}_{p_{\mathrm{eval}}}\left[\phi(z)\right] - \phi\left(\mathbb{E}_{p(z)}[z]\right) - \left\langle \nabla\phi\left(\mathbb{E}_{p(z)}[z]\right), \mathbb{E}_{p_{\mathrm{eval}}}[z] - \mathbb{E}_{p(z)}[z]\right\rangle - \left(\mathbb{E}_{p_{\mathrm{eval}}}\left[\phi(z)\right] - \phi\left(\mathbb{E}_{p_{\mathrm{eval}}}[z]\right)\right) \\ &= D_\phi\left(\mathbb{E}_{p_{\mathrm{eval}}}[z], \mathbb{E}_{p(z)}[z]\right). \end{aligned}$$

Rearrange the first and the last equation, we have that

$$\mathbb{E}_{p_{\text{eval}}}\left[D_\phi\left(z, \mathbb{E}_p(z)\right)\right] = D_\phi\left(\mathbb{E}_{p_{\text{eval}}}[z], \mathbb{E}_{p(z)}[z]\right) + \underbrace{\mathbb{E}_{p_{\text{eval}}}\left[D_\phi\left(z, \mathbb{E}_{p_{\text{eval}}}(z)\right)\right]}_{\text{irreducible}},$$

which aligns with the bias-variance decomposition in Bregman divergence (Equation 19). For any (possibly random) predictor A, define the *central prediction* $\mathcal{E}_\phi[A] := \arg\min_u \mathbb{E}_A[D_\phi(u, A)]$. Then (e.g., Pfau, 2013)

$$\begin{aligned}\mathbb{E}_{Z,A}[D_\phi(Z, A)] &= \mathbb{E}_Z[D_\phi(Z, \mathbb{E}[Z])] + \mathbb{E}_A[D_\phi(\mathbb{E}[Z], A)] \\ &= \underbrace{\mathbb{E}_Z[D_\phi(Z, \mathbb{E}[Z])]}_{\text{Bayes error}} + \underbrace{D_\phi(\mathbb{E}[Z], \mathcal{E}_\phi[A])}_{\text{bias}} + \underbrace{\mathbb{E}_A[D_\phi(\mathcal{E}_\phi[A], A)]}_{\text{model variance}}. \end{aligned} \tag{19}$$

□

C.4 Uncertainty-based acquisition via Bregman divergences recovers predictive-variance reduction, BALD, and EPIG

Predictive posterior variance (Cohn et al, 1994) Set the Bregman potential to the quadratic $\phi(u) = u^2$, take the quantity of interest to be the latent response $z = f(c)$ at a context c, and let the context distribution be the input distribution $p(c)$. Then the Bregman uncertainty is the predictive variance, $h_\phi[p(z \mid c)] = \mathrm{Var}_n\big(z \mid c\big)$, and the pool-based EPU at candidate x is

$$\mathrm{EPU}(x) = \mathbb{E}_{p(c)\, p(y|x,c)}\big[\mathrm{Var}_{p(z|c,y,x)}[z]\big],$$

which retrieves a Bayesian variant of the objective in Cohn et al (1994).

BALD (Houlsby et al, 2011) Choose the log score (entropy) potential ϕ and set the target to the model parameters $z = \theta$. The pool-based EUR becomes the expected drop in entropy of θ,

$$\mathrm{EUR}(x) = \mathbb{E}_{p(y|x)}\big[\,\mathrm{H}\big[p(\theta)\big] - \mathrm{H}\big[p(\theta \mid x, y)\big]\big] = \mathrm{I}(\theta; y \mid x),$$

which is the BALD mutual-information objective Bickford Smith et al (2023).

EPIG (Bickford Smith et al, 2023) With the same entropy potential, take the target-context pair $(z, c) = (y_*, x_*)$ and average the context-aware EUR over the target-input distribution $p(x_*)$ (cf. Section 3):

$$\mathbb{E}_{p(x_*)}[\mathrm{EUR}(x; x_*)] = \mathbb{E}_{p(x_*)\, p(y|x)}\big[\mathrm{H}\big[p(y_* \mid x_*)\big] - \mathrm{H}\big[p(y_* \mid x_*, x, y)\big]\big],$$

which is precisely the Expected Predictive Information Gain (EPIG).

D Estimation

D.1 Computing EUR by mutual information for classification tasks

We start by deriving the weighted version of EPIG:

$$\begin{aligned}\mathrm{EPIG}_w &= \mathbb{E}_{p(c)}\Big[\mathrm{EUR}^{\phi,T,w}_{p(z,y|c,x)}(x; c)\Big] \\ &= \mathbb{E}_{p(c)}\big[h^w_\phi[p(z|c)] - \mathbb{E}_{p(y|x,c)}\big[h^w_\phi[p(z|c,x,y)]\big]\big] \\ &= \mathbb{E}_{p(c)}\big[\mathbb{E}_{p(z|c)}[-w(z)\log p_w(z|c)] - \mathbb{E}_{p(z,y|c,x)}[-w(z)\log p_w(z|c,y,x)]\big] \\ &= \mathbb{E}_{p(c)p(z,y|c,x)}\left[w(z)\log\frac{p_w(z|c,y,x)}{p_w(z|c)}\right] \\ &= \mathbb{E}_{p(c)\,p(z|c)\,p(y|x,z,c)}\Big[w(z)\,\log\frac{p(y \mid x, z, c)}{q(y \mid x, c)}\Big],\end{aligned}$$

where the last equation is derived by Equation 18. In terms of implementation, we use the formula for mutual information directly to reduce computational cost.

$$
\begin{aligned}
\mathrm{EPIG}_w &= \mathbb{E}_{p(c)\,p(z|c)\,p(y|x,z,c)}\Big[w(z)\,\log\frac{p(y\mid x,z,c)}{q(y\mid x,c)}\Big]\\
&= \mathbb{E}_{p(c)\,p(z|c)\,p(y|x,z,c)}\left[w(z)\,\log\frac{p(y\mid x,z,c)}{\mathbb{E}_{p_w(z|c)}\big[p(y\mid x,z,c)\big]}\right],\\
&= \mathbb{E}_{p(c)\,p(z,y|c,x)}\left[w(z)\,\log\frac{p(y\mid x,z,c)}{\mathbb{E}_{p(z|c)}\big[\frac{w(z)}{\mathbb{E}_{p(z|c)}[w(z)]}\,p(y\mid x,z,c)\big]}\right]\\
&= \mathbb{E}_{p(c)\,p(z,y|c,x)}\left[w(z)\,\log\frac{w(z)p(z,y\mid c,x)}{\frac{w(z)}{\mathbb{E}_{p(z|c)}[w(z)]}p(z\mid c)\mathbb{E}_{p(z|c)}[w(z)\,p(y\mid x,z,c)]}\right]\\
&= \mathbb{E}_{p(c)}\left[\sum_{z,y} w(z)\,p(z,y\mid c,x)\,\log\frac{w(z)p(z,y\mid c,x)}{p_w(z\mid c)\sum_z w(z)\,p(z,y\mid c,x)}\right].
\end{aligned}
$$

Following the notation and strategy in Bickford Smith et al (2023), suppose we have samples from $\theta_i \sim p(\theta)$ and $c^j \sim p(c)$. The predictive distribution needed can be estimated as

$$
\begin{aligned}
\hat{p}_n\left(z, y \mid c^j, x\right) &= \frac{1}{K}\sum_{i=1}^{K} p\left(z \mid c^j, \theta_i\right)\; p\left(y \mid x, \theta_i\right)\\
\hat{p}_n\left(z \mid c^j\right) &= \frac{1}{K}\sum_{i=1}^{K} p\left(z \mid c^j, \theta_i\right)\\
\hat{p}_w(z \mid c^j) &= \frac{w(z)}{\sum_z \hat{p}(z \mid c^j) w(z)} \cdot \hat{p}(z \mid c^j).
\end{aligned}
$$

D.2 Exact one-step posterior update for variance-based acquisition

We consider the GP regression setting from Section 5.1. Let $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ with inputs $X_n = [x_1, \ldots, x_n]^\top$ and responses $\boldsymbol{y} = [y_1, \ldots, y_n]^\top$. The prior model is defined as

$$
f \sim \mathcal{GP}(0, k), \qquad y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2),
$$

with fixed hyperparameters. Denote by

$$
A_n := K_{X_n X_n} + \sigma^2 I, \qquad \boldsymbol{k}(x, X_n) := \big[k(x, x_1), \ldots, k(x, x_n)\big]^\top,
$$

and write the posterior predictive mean and (co)variance under $\mathcal{D}_n$ as

$$
\begin{aligned}
m_n(x) &= \boldsymbol{k}(x, X_n)^\top A_n^{-1} \boldsymbol{y},\\
v_n(x, x') &= k(x, x') - \boldsymbol{k}(x, X_n)^\top A_n^{-1} \boldsymbol{k}(x', X_n), \qquad v_n(x) := v_n(x, x).
\end{aligned}
$$

Let x^+ be a candidate pooled input and y^+ the (as yet unobserved) label at x^+. The one–step posterior after hypothetically observing (x^+, y^+) is Gaussian with (Williams & Rasmussen, 2006, Sec. 2.4)

$$
m_{n+1}(x) = m_n(x) + \beta_n(x; x^+)\,\big(y^+ - m_n(x^+)\big), \tag{20}
$$

$$
v_{n+1}(x, x') = v_n(x, x') - \beta_n(x; x^+)\,\beta_n(x'; x^+)\,\tau_n^2(x^+), \tag{21}
$$

where

$$
\beta_n(x; x^+) := \frac{\mathrm{cov}_n\big(f(x), y^+\big)}{\mathrm{var}_n(y^+)} = \frac{v_n(x, x^+)}{v_n(x^+) + \sigma^2}, \qquad \tau_n^2(x^+) := v_n(x^+) + \sigma^2.
$$

Expected variance reduction (EVR) Let $c \sim p(c)$ denote the distribution over contexts (test inputs) for which we evaluate uncertainty (Section 3). Since $v_{n+1}(x, x)$ in (21) does not depend on the realised y^+, $\mathbb{E}_{p(y^+|x^+)}[v_{n+1}(c, c)] = v_{n+1}(c, c)$. Thus the squared–loss EUR (EVR) for candidate x^+ can be approximated by samples of c:

$$\mathrm{EVR}(x^+) \approx \frac{1}{M}\sum_{j=1}^{M}\Big(v_n(c_j) - \mathbb{E}_{p(y^+|x^+)}\big[v_{n+1}(c_j, c_j)\big]\Big) = \frac{1}{M}\sum_{j=1}^{M}\frac{v_n(c_j, x^+)^2}{v_n(x^+) + \sigma^2}. \tag{22}$$

This is exactly the expected reduction in the Jensen gap h_ϕ for $\phi(x) = x^2$, averaged over contexts, and corresponds to the unweighted EUR in Section 3. This score is also equivalent (up to constants) to minimising $\mathrm{tr}\left(\mathrm{Var}[f_A \mid D, y_x]\right)$ in variance-based transductive active learning (VTL), with A corresponding to the context set (Hübotter et al, 2024).

Weighted expected variance reduction (EVR_w) To encode region-specific preferences, fix $w : \mathbb{R} \to (0, \infty)$ and, for a context $c \sim p(c)$, define the weighted Bregman uncertainty (squared-error case) under the current predictive $z \sim \mathcal{N}\big(m_n(c), v_n(c)\big)$:

$$U_w^{(n)}(c) := \mathbb{E}\Big[w(z)\big(z - \mu_w^{(n)}(c)\big)^2\Big], \qquad \mu_w^{(n)}(c) := \frac{\mathbb{E}[w(z)\, z]}{\mathbb{E}[w(z)]}.$$

By the reweighting identity in Section 3, $U_w^{(n)}(c) = \bar{w}_n(c) \cdot \mathrm{Var}_{p_w}(z)$ with $p_w(\cdot) \propto w(\cdot)\mathcal{N}(m_n(c), v_n(c))$ and $\bar{w}_n(c) = \mathbb{E}_p[w(z)]$; for general w the moments are not closed form, so we estimate them by Monte Carlo.

After updating with (x^+, y^+), $z \mid y^+ \sim \mathcal{N}\big(m_{n+1}(c),\, v_{n+1}(c)\big)$ with mean/variance from Equations 20 and 21. We approximate the expectation over the GP predictive law $p(y^+ \mid x^+) = \mathcal{N}\big(m_n(x^+), \tau_n^2(x^+)\big)$ using common random numbers:

$$\begin{aligned} y^{+,(j)} &= m_n(x^+) + \sqrt{\tau_n^2(x^+)}\,\eta^{(j)}, \qquad \eta^{(j)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \\ z_{\cdot|+}^{(i,j)} &= m_n(c) + \beta_n(c_j; x^+)\sqrt{\tau_n^2(x^+)}\,\eta^{(j)} + \sqrt{v_{n+1}(c_j)}\,\varepsilon^{(i)}, \qquad \varepsilon^{(i)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \end{aligned}$$

reusing the same $\{\eta^{(j)}\}$ and $\{\varepsilon^{(i)}\}$ across candidates x^+ to reduce estimator variance. By jointly sampling from (y^+, z, c), the expected posterior weighted uncertainty is approximated by

$$\mathbb{E}_{p(y^+|x^+)}\Big[U_w^{(n+1)}(c_j)\Big] \approx \frac{1}{S}\sum_{i=1}^{S} w\big(z_{\cdot|+}^{(i,j)}\big)\left(z_{\cdot|+}^{(i,j)} - \mu_w^{(j)}(c)\right)^2, \quad \mu_w^{(j)}(c) := \frac{\sum_{i=1}^{S} w\big(z_{\cdot|+}^{(i,j)}\big)\, z_{\cdot|+}^{(i,j)}}{\sum_{i=1}^{S} w\big(z_{\cdot|+}^{(i,j)}\big)}.$$

The weighted EUR score averages the reduction across contexts:

$$\mathrm{EVR}_w(x^+) \approx \frac{1}{M}\sum_{j=1}^{M}\Big(U_w^{(n)}(c_j) - \mathbb{E}_{p(y^+|x^+)}\big[U_w^{(n+1)}(c_j)\big]\Big).$$

Remark. For exponential weights $w(z) = \exp(\alpha z)$, the reweighting is analytic: $p_w = \mathcal{N}\big(m_n(c) + \alpha v_n(c),\, v_n(c)\big)$, $\bar{w}_n(c) = \exp\big(\alpha m_n(c) + \frac{1}{2}\alpha^2 v_n(c)\big)$, and $U_w^{(n)}(c) = \bar{w}_n(c)\, v_n(c)$, yielding a closed–form pre–update term.

Implementation notes Compute and cache the Cholesky $A_n = LL^\top$. For any candidate x^+, obtain

$$s_n(\mathcal{C}, x^+) := v_n(\mathcal{C}, x^+) = k(\mathcal{C}, x^+) - K_{\mathcal{C}X_n} A_n^{-1} \boldsymbol{k}(X_n, x^+), \qquad \tau_n^2(x^+) = v_n(x^+) + \sigma^2,$$

then evaluate (22) via $\mathrm{EVR}(x^+) = \frac{1}{M}\sum_{j=1}^{M} s_n(c_j, x^+)^2/\tau_n^2(x^+)$. The same cached solves provide $\beta_n(c; x^+) = s_n(c, x^+)/\tau_n^2(x^+)$ and $v_{n+1}(c) = v_n(c) - s_n(c, x^+)^2/\tau_n^2(x^+)$ for the weighted estimator above; only the nonlinear weighting $w(\cdot)$ requires Monte Carlo. Note that we compute EVR_w to empirically demonstrate its non-negativity. To reduce computation, one could also minimise expected predictive posterior score over contexts.

D.3 General techniques for variance-based uncertainty estimation

We briefly simplify the objectives used to score a candidate query x and provide low-variance Monte Carlo estimators.

Setup. Let $c \sim p(c)$ denote a context (e.g. a test input), $z \in \mathbb{R}$ a scalar predictive target, and $p(\cdot)$ the predictive distribution after n observations. We consider the weighted squared-loss geometry with a strictly positive weight function $w : \mathbb{R} \to (0, \infty)$ and the reweighted belief

$$p_w(z \mid c) \;\propto\; w(z)\, p(z \mid c), \qquad \bar{w}(c) \;:=\; \mathbb{E}_{p(z|c)}[w(z)].$$

In this geometry the uncertainty is the non-negative Jensen gap $h_\varphi[p(z \mid c)] = \mathrm{Var}_{p(z|c)}[z]$ (and its weighted counterpart $\bar{w}(c)\,\mathrm{Var}_{p_w(z|c)}[z]$), so EUR reduces to (weighted) expected predictive-variance reduction; see Secs. 3.3–3.4 and Appendix C.4.

EVR. The unweighted expected variance reduction is

$$\begin{aligned}\mathrm{EVR}(x) &= \mathbb{E}_{p(c)}\, \mathbb{E}_{p(y|x)}[\mathrm{Var}(z \mid c) - \mathrm{Var}(z \mid c, y, x)] \\ &= \text{constant} + \mathbb{E}_{p(c)}\, \mathbb{E}_{p(y|x)}\left[\left(\mathbb{E}_{p(z|c,y,x)}[z]\right)^2\right].\end{aligned}$$

Using two i.i.d. replicates $z, z' \overset{\text{i.i.d.}}{\sim} p(z \mid c, y, x)$ as per Rainforth et al (2018) yields $\left(\mathbb{E}[z]\right)^2 = \mathbb{E}[zz']$ and thus

$$\begin{aligned}\mathrm{EVR}(x) &= \text{constant} + \mathbb{E}_{p(c)}\, \mathbb{E}_{p(y|x)}\, \mathbb{E}_{p(z|c,y,x)p(z'|c,y,x)}[zz'], \\ &= \text{constant} + \mathbb{E}_{p(c)}\, \mathbb{E}_{p(z|c)}\, \mathbb{E}_{p(y|x,z,c)p(z'|c,y,x)}[zz'], \\ &= \text{constant} + \mathbb{E}_{p(c)}\, \mathbb{E}_{p(\theta)p(z|\theta,c)p(y|x,\theta,c)}\, \mathbb{E}_{p(\theta'|y,x)p(z'|\theta',c)}[zz'], \\ &= \text{constant} + \mathbb{E}_{p(c)}\, \mathbb{E}_{p(\theta)p(y|x,\theta,c)p(z|\theta,c)}\, \mathbb{E}_{p(\theta')p(z'|\theta',c)}\left[\frac{p(y \mid x, \theta', c)}{\mathbb{E}_{p(\theta'')}\left[p(y \mid x, \theta', c)\right]} zz'\right], \\ &= \text{constant} + \mathbb{E}_{p(c)}\, \mathbb{E}_{p(\theta)p(y|x,\theta,c)}\, \mathbb{E}_{p(\theta')}\left[\frac{p(y \mid x, \theta', c)}{\mathbb{E}_{p(\theta'')}\left[p(y \mid x, \theta'', c)\right]} \mathbb{E}_{p(z|\theta,c)}\left[z\right] \mathbb{E}_{p(z'|\theta',c)}\left[z'\right]\right].\end{aligned}$$

Either the final or penultimate line can now be estimated using nested Monte Carlo (Rainforth et al, 2018) or adapting a contrastive estimator like PCE (Foster et al, 2020), while if we have a mechanism for drawing (approximate) samples from $p(z' \mid c, y, x)$, then we can use the second line directly instead. One can also replace the sampling from any of the prior terms, e.g. $p(\theta''|c)$, with an appropriate importance sampler instead if desired.

Weighted EVR. With prediction-space weighting, the objective becomes

$$\begin{aligned}\mathrm{EVR}_w(x) &= \mathbb{E}_{p(c)}\, \mathbb{E}_{p(y|x)}\left[\bar{w}(c)\,\mathrm{Var}_{p_w(z|c)}[z] \;-\; \bar{w}(c, y, x)\,\mathrm{Var}_{p_w(z|c,y,x)}[z]\right] \\ &= \text{constant} - \mathbb{E}_{p(c)}\, \mathbb{E}_{p(y|x)}\left[\bar{w}(c, y, x)\,\mathrm{Var}_{p_w(z|c,y,x)}[z]\right].\end{aligned}$$

Using $\bar{w}\,\mathrm{Var}_q[z] = \mathbb{E}_p[w(z)z^2] - \frac{\left(\mathbb{E}_p[w(z)z]\right)^2}{\mathbb{E}_p[w(z)]}$ (where expectations are under the indicated $p(z \mid \cdot)$), and since $\mathbb{E}_{p(y|x)}\left[\mathbb{E}_{p(z|c,y,x)}[w(z)z^2]\right]$ is constant in x by marginalisation, we then have

$$\begin{aligned}&\mathrm{EVR}_w(x) - \text{constant} = \; + \mathbb{E}_{p(c)}\, \mathbb{E}_{p(y|x)}\left[\frac{\left(\mathbb{E}_{p(z|c,y,x)}[w(z)z]\right)^2}{\mathbb{E}_{p(z|c,y,x)}[w(z)]}\right] \\ &= \mathbb{E}_{p(c)}\, \mathbb{E}_{p(y|x)p(z|c,y,x)p(z'|c,y,x)}\left[\frac{zz'w(z)w(z')}{\mathbb{E}_{p(\theta''|y,x,c)p(z|\theta'',c)}[w(z)]}\right] \\ &= \mathbb{E}_{p(c)}\, \mathbb{E}_{p(\theta)p(y|x,\theta,c)p(z|\theta,c)}\mathbb{E}_{p(\theta'|c,y,x)p(z'|\theta',c)}\left[\frac{zz'w(z)w(z')\mathbb{E}_{p(\theta'')}\left[p(y|x,\theta'',c)\right]}{\mathbb{E}_{p(\theta'')p(z|\theta'',c)}[w(z)p(y|x,\theta'',c)]}\right] \\ &= \mathbb{E}_{p(c)}\, \mathbb{E}_{p(\theta)p(y|x,\theta,c)p(z|\theta,c)}\mathbb{E}_{p(\theta')p(z'|\theta',c)}\left[\frac{p(y|x,\theta',c)}{\mathbb{E}_{p(\theta'')p(z|\theta'',c)}[w(z)p(y|x,\theta'',c)]} zz'w(z)w(z')\right] \\ &= \mathbb{E}_{p(c)}\, \mathbb{E}_{p(\theta)p(y|x,\theta,c)}\mathbb{E}_{p(\theta')}\left[\frac{p(y|x,\theta',c)}{\mathbb{E}_{p(\theta'')p(z|\theta'',c)}[w(z)p(y|x,\theta'',c)]}\mathbb{E}_{p(z|\theta,c)}[zw(z)]\mathbb{E}_{p(z'|\theta',c)}[z'w(z')]\right].\end{aligned}$$

We can now again do a nested Monte Carlo or PCE-style estimator for the nested estimations given in the last and penultimate lines in the same way as for the unweighted case with the same underlying computational complexity.

D.4 Method complexity

Classification. The weighted EPIG estimator, EPIG_w (see Appendix D.1), has per–candidate-input complexity $\mathcal{O}(MK)$, where M is the number of Monte Carlo draws over contexts and joint predictive outputs, and K is the number of samples of model parameters. This matches the asymptotic cost of standard (unweighted) EPIG estimators Bickford Smith et al (2023).

Regression. For regression we use expected variance reduction (EVR) and its weighted counterpart (EVR_w). In general, both have complexity $\mathcal{O}(MS)$, with S inner Monte Carlo draws used to estimate weighted posterior means and/or variances. In models that admit an exact one-step posterior update—e.g., exact Gaussian processes where the predictive variance after a single observation is available in closed form—the per–candidate complexity reduces to $\mathcal{O}(M)$ because no parameter sampling is required.

E Experiment details

E.1 Metrics

Our primary metrics match the losses assumed in the acquisition objective. For classification we report negative log likelihood (NLL) and its weighted counterpart,

$$\text{NLL}_w = \frac{\mathbb{E}_{p_{\text{eval}}(y,x)}\big[w(y)\,\big(-\log q_{n+m}(y \mid x)\big)\big]}{\mathbb{E}_{p_{\text{eval}}(y)}[w(y)]}.$$

For regression we report squared-error loss (SEL) and its weighted counterpart,

$$\text{SEL}_w = \frac{\mathbb{E}_{p_{\text{eval}}(y,x)}\big[w(y)\,\big(y - \mathbb{E}_{q_{n+m}(y|x)}[y]\big)^2\big]}{\mathbb{E}_{p_{\text{eval}}(y)}[w(y)]}.$$

Note that we do not record accuracy as an evaluation metric, as accuracy implies a zero-one loss that we do not optimise for during data acquisition. Therefore, there is no reason for acquisition methods to dominate on accuracy-related metrics.

In addition, we do not claim that each benchmark has a uniquely correct weighting. Rather, the experiments are designed to test whether the acquisition rule can be customised to different user-specified downstream preferences. Accordingly, all weights are fixed a priori from simple task-motivated heuristics and are not tuned on validation performance.

E.2 GP model setting

We use a Gaussian process (GP) regressor with a constant prior mean $m(x) \equiv c$ set to the median of the observed responses; we train on $y - c$ and add c back at prediction. Inputs are standardized feature-wise once on the initial train+pool set and the same transform is reused thereafter; the target is left in native units. The kernel is a sum of a linear (dot-product) term and a Matérn term with i.i.d. white noise, i.e., $m(x) \equiv c_t$ and $k(x, x') = \sigma^2_{\text{lin},t}\, z^\top z' + \sigma^2_{f,t}\, \kappa_\nu\big(\|z - z'\|/\ell_t\big) + \sigma^2_{n,t}\, \delta_{x,x'}$, where z denotes standardized inputs (we standardize X only), κ_ν is the Matérn correlation (we use $\nu = 3/2$ unless noted), and $\delta_{x,x'}$ is the Kronecker delta (Williams & Rasmussen, 2006; Stein, 1999). This is the usual universal-kriging decomposition of a linear trend plus a stationary residual (Cressie, 1993; Diggle & Ribeiro, 2007). Hyperparameters $(c_t, \sigma^2_{\text{lin},t}, \sigma^2_{f,t}, \sigma^2_{n,t}, \ell_t)$ are recomputed by robust plug-in rules every three acquisitions and held fixed in between; no marginal-likelihood maximisation or other gradient-based tuning is used. We fix ν=5/2 on YACHT and ν=3/2 on SLUMP/ESTATE; ν=5/2 induces smoother (twice m.s. differentiable) sample paths than ν=3/2 (once m.s. differentiable), which we found to better match the hydrodynamics response, whereas the latter is more robust for noisier, less-smooth tabular targets.

E.3 Robust hyperparameter estimation

Following the GP setting above, we set the location $c_t = \text{median}(y)$ (robust location) and estimate scales by robust/difference-based rules: (i) the observation noise is estimated from nearest-neighbor differences in

standardized input space, $\sigma_{n,t} = \frac{1.4826}{\sqrt{2}}\,\mathrm{median}_i\,|y_i - y_{j(i)}|$, $j(i) = \arg\min_{j\neq i}\|z_i - z_j\|$, a difference-based variance estimator in the spirit of Rice (1984) and Hall et al (1990), with the MAD-to-σ factor 1.4826 ensuring normal Fisher-consistency (Huber & Ronchetti, 2009; Tyler, 2011); (ii) we split linear vs. residual signal by a small-ridge fit on Z (standardized inputs) to $y - c_t$, taking $\sigma^2_{\mathrm{lin},t} = \mathrm{Var}(\hat{y})$ and $\sigma^2_{f,t} = \max\big((1.4826\,\mathrm{median}|r|)^2 - \sigma^2_{n,t}, \varepsilon\big)$ with residuals $r = (y - c_t) - \hat{y}$ and $\varepsilon = 10^{-12}$ (Hoerl & Kennard, 1970; Huber & Ronchetti, 2009); (iii) we set the length-scale ℓ_t by method-of-moments matching: choose a characteristic design spacing $r_{0,t}$ (the median nearest-neighbor distance in z) and solve $\kappa_\nu(r_{0,t}/\ell_t) = \rho$ with a target short-lag correlation $\rho = 0.5$ (a "practical range" style calibration; closed form for $\nu = 3/2$ gives $\ell_t \approx 1.032\,r_{0,t}$) (Cressie, 1993; Stein, 1999; Diggle & Ribeiro, 2007). These choices yield a prior that reflects the data scale and geometry while remaining fully predetermined between update rounds.

F Extra results

F.1 Regression

With the same model setup in Section 5.1.1, Figure 5 shows the data-gathering procedure when $w(z) = \exp(-z)$, which targets the precision in the low-value region. After a warm-start of exploring potential low value, the target region is secured and more data is acquired around the area.

F.2 Classification

We present Figure 6 to demonstrate the evolution for proportion of acquired class for Vehicle, Landsat and Vowel, respectively. The weighting has a one-to-one correspondence with the class in proportion plot. Curves show cumulative proportion of acquired labels per class for each method (Random, EPIG, EPIG$_w$) by the mean over 100 runs; shaded regions indicate the SEM range. The class-mix plots confirm that EPIG$_w$ allocates a larger share of queries to the high-weight classes compared to EPIG and Random.

G Resources

G.1 Software

Project	Citation	License	URL
NumPy	Harris et al (2020)	BSD (3-clause)	`numpy.org`
SciPy	Virtanen et al (2020)	BSD (3-clause)	`scipy.org`
Scikit-learn	Pedregosa et al (2011)	BSD (3-clause)	`scikit-learn.org`
Matplotlib	Hunter (2007)	PSF-based	`matplotlib.org`
pandas	McKinney (2010)	BSD (3-clause)	`pandas.pydata.org`
tqdm	da Costa-Luis (2019)	MIT	`github.com/tqdm/tqdm`
PyTorch	Paszke et al (2019)	BSD (3-clause)	`pytorch.org`
GPyTorch	Gardner et al (2018)	MIT	`gpytorch.ai`
h5py	h5py developers (2025)	BSD (3-clause)	`h5py.org`
PMLB	Olson et al (2017); Romano et al (2021)	MIT	`epistasislab.github.io/pmlb`
ucimlrepo (UCI client)	Kelly et al (2025)	MIT	`github.com/uci-ml-repo/ucimlrepo`

Table 3 Third-party software used in this work.

G.2 Datasets

Dataset	Citation	URL
Vehicle	Mowforth & Shepherd (1993)	`archive.ics.uci.edu/dataset/149/statlog+vehicle+silhouettes`
Landsat	Srinivasan (1993)	`archive.ics.uci.edu/dataset/146/statlog+landsat+satellite`
Vowel	Deterding et al (1988)	`archive.ics.uci.edu/dataset/152/connectionist+bench+vowel+recognition+deterding+data`
Slump	Yeh (2007)	`archive.ics.uci.edu/dataset/182/concrete+slump+test`
Yacht	Gerritsma et al (1981)	`archive.ics.uci.edu/dataset/243/yacht+hydrodynamics`
Estate	Yeh (2018)	`archive.ics.uci.edu/dataset/477/real+estate+valuation+data+set`

Table 4 UCI datasets used in this work. All are available under Creative Commons Attribution 4.0 (CC BY 4.0).

G.3 Compute

All experiments were executed on a single local workstation; no distributed or cloud resources were used. The software environment is listed in Table 3, and the datasets are summarized in Table 4.

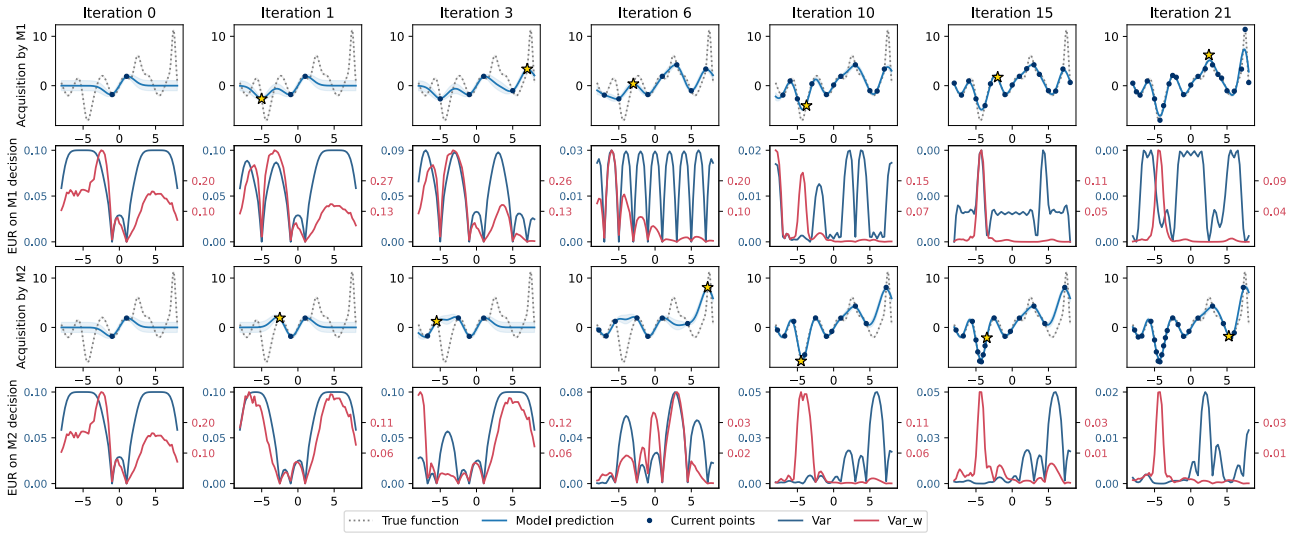

Figure 5 Expected variance reduction (EVR) vs. a variant with weighting $w(z) = \exp(-z)$ (EVR_w). Row 1: prediction with data acquired by EVR. Row 2: values of EVR and EVR_w given the Row 1 training set (the maximiser is labelled next). Row 3: prediction with data acquired by EVR_w. Row 4: values of EVR and EVR_w given the Row 3 training set.

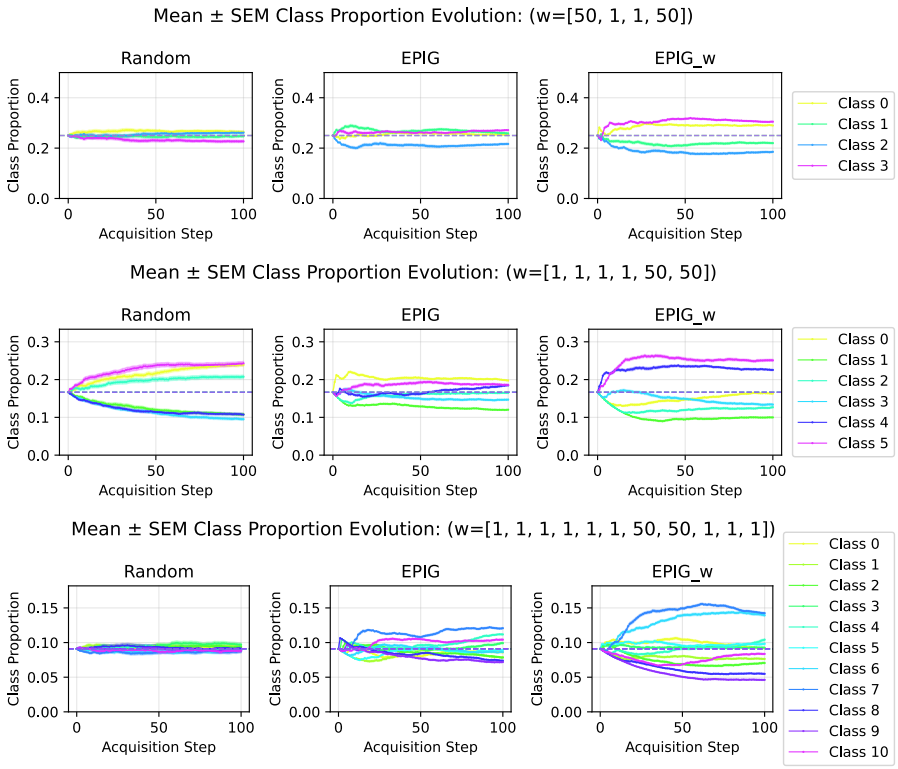

Figure 6 From top to bottom: evolution of acquired-class proportions on VEHICLE ($w = [50, 1, 1, 50]$), LANDSAT ($w = [1, 1, 1, 1, 50, 50]$) and VOWEL ($w = [1, 1, 1, 1, 1, 1, 50, 50, 1, 1, 1]$).