# Measuring Coherence in Predictive Models

**Jamie Reason*** University of Oxford
**Vik Shirvaikar** University of Oxford
**Mark van der Wilk** University of Oxford
**Chris C. Holmes** University of Oxford

## Abstract

Many sequential decision-making procedures, such as active learning, implicitly assume that a model's current predictions agree with those it expects to make after updating on new data, a property we call coherence. Most models in use do not update by exact conditioning and are not guaranteed to be coherent, which can lead to worse decisions. Yet there is currently no general way to measure incoherence or to tell when it matters. We develop tools to measure incoherence and characterise when it affects decisions. We introduce the incoherence ratio, an MMD-based measure of how far any updating procedure departs from coherence, computed only from the procedure's own predictions and updates, and accompany it with an anytime-valid test of coherence. In Bayesian experimental design, acquisition objectives that are equivalent for coherent models choose different data under every approximate model we study, and the incoherence ratio tracks how far incoherence distorts them. We also introduce action coherence, which captures only the incoherence that changes a decision. What a resulting disagreement costs, however, is determined primarily by predictive accuracy. Sequential decisions depend on how a model updates as well as on what it predicts, so coherence, like accuracy, should be checked rather than assumed.

## 1 Introduction

In sequential decision-making problems, actions are often chosen using a model that is updated as new data arrive. Active learning [Settles, 2009], for example, selects the data whose acquisition will most improve the model's predictions after updating. Making this choice requires the model to anticipate how its own predictions will change, which relies on its current predictions agreeing with those it expects to make after updating. We call a model with this property coherent. Many methods for sequential decision-making are derived assuming exact Bayesian conditioning [Rainforth et al., 2024, Garnett, 2023], which guarantees coherence by construction. The models they are applied to, however, typically update by other means and are not guaranteed to be coherent. Rather than assuming coherence, we measure how severely it is violated, and identify when and how incoherence affects decisions.

Bayesian experimental design shows how incoherence can reach a decision. The expected predictive information gain (EPIG) [Bickford Smith et al., 2023] selects the design whose outcome most reduces uncertainty in the model's prediction at a target of interest. It can be written in several standard forms, as a reduction in predictive entropy, as a mutual information, or as an expected divergence from the current predictive, and these are used interchangeably because they agree under coherence. For an incoherent model, however, they become different acquisition rules that can rank candidate designs in opposite orders (Section 3). Empirically, the entropy form acquires worse data on average than the mutual-information form for every approximate model we study (Section 6).

Coherence has a long history in both philosophy [van Fraassen, 1984, Lewis, 1999] and statistics [Goldstein, 1983], where it corresponds to the martingale property of predictive distributions

*Correspondence to: jamie.reason@reuben.ox.ac.uk.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

[Fong et al., 2023] and underpins the predictive approach to Bayesian inference [de Finetti, 1937]. As models whose updates are not explicit conditioning have become central to machine learning, a growing body of work has asked whether they nonetheless behave like Bayesian updates [Mlodozeniec et al., 2024]. In-context learning in large language models has been shown to violate the martingale property [Falck et al., 2024], approximate Bayesian neural networks have been found to update incoherently [Pituk et al., 2025], and methods for prior-data fitted networks are built without assuming coherent updates [Nagler and Rügamer, 2025]. These works establish that coherence can fail in important models. Separately, work on sequential decisions has shown that decision quality depends on a model's joint predictive distribution rather than its marginal predictions [Wen et al., 2022, Osband et al., 2022]. A model's update rule implicitly defines such a joint, and coherence is the requirement that it agree with the model's current predictions. Because coherence concerns how predictions update rather than how accurate they are, a model can predict accurately at every input and still be incoherent. What is still missing is a general way to measure how incoherent an arbitrary updating procedure is, and an account of when incoherence matters for the decisions these models are used to make.

We address both. We first set out what coherence means for any procedure that predicts and updates, without reference to a prior, likelihood or parameter posterior (Section 2). The requirement is local to the history, target and update relevant to the problem, and, in keeping with a predictivist view, concerns only predictions of observable quantities. To measure incoherence, we introduce the incoherence ratio $\Psi$, a scale-invariant measure of how far a procedure's updates depart from conditioning, estimated from its own predictions and updates without a test set or reference posterior (Section 4). We also give an anytime-valid sequential test of coherence (Section 5). To determine when incoherence matters, we introduce action coherence, a relaxation of coherence that asks only whether incoherence changes the action a procedure selects, and so isolates exactly the incoherence that reaches a decision (Section 3).

Our experiments show what incoherence reveals about models and when it reaches decisions (Section 6). Our measure of incoherence falls as an approximation is refined and ranks models consistently across datasets. For every approximate model we study, objectives that agree under coherence select different data on a substantial share of decisions, so incoherence reaches the action. Our measure of incoherence tracks how susceptible a model's decisions are to this disagreement, while accuracy determines what it costs. Incoherence is therefore a warning rather than a verdict. Sequential decisions rely on how a model updates, not only on what it predicts, and coherence is the property that captures this.

## 2 Predictive coherence

A predictive procedure comprises a probabilistic model and its prescribed update rule. At a realised history $h$, the procedure returns predictive laws for the variables required by the prediction and update problem. We write $Y$ for the prediction target and $Z$ for the future observations that will be incorporated before the procedure is queried for $Y$ again. We call $Z$ the *update block*, which may contain one or more observations. An ordered update design $d = (d_1, \ldots, d_K)$ specifies which observations comprise the block and their order. The update is written $\mathsf{U}(h, z)$ and maps a history and an observed update block to an updated history. In supervised learning, for example, the initial history may be the observed dataset $h = (x_{1:n}, y_{1:n})$, and the target $Y$ the response at a fixed test input $x_*$. An update design $d = x_{n+1:n+K}$ specifies a collection of additional inputs, with corresponding update block $Z = Y_{n+1:n+K}$. Here the components of $Z$ are responses of the same kind as $Y$. More generally, however, the construction assumes neither that the observations are exchangeable nor that the components of $Z$ take values in the same space as $Y$.

We write $P_Y(\cdot; h)$ and $P_Z(\cdot; h)$ for the procedure's current predictive laws for the target and the complete update block, respectively. When the procedure does not supply a joint law for $Z = (Z_1, \ldots, Z_K)$, we construct one prequentially in the order specified by the design $d$: $P_Z(dz; h) = \prod_{k=1}^{K} P_{Z_k}(dz_k; h_{k-1})$, where $h_0 \equiv h$ and $h_k := \mathsf{U}(h_{k-1}, z_k)$ is the updated history after observing $z_{\leq k}$. We assume throughout that updating on the whole block agrees with updating component-wise along $d$, so that $h_K = \mathsf{U}(h, z)$. Throughout, a semicolon indicates that a predictive law is indexed by a history, and we reserve the vertical bar for probabilistic conditioning within a specified joint distribution. Subscripts name the variable whose law is described and superscripts carry the indices on which the law depends, namely the update design and its realisation. We suppress any index that

is held fixed in the surrounding discussion, and reinstate it where it varies. In this section the design is fixed and suppressed.

**Update-induced joint.** The model's predictive law for the update block and its post-update target predictions together define the *update-induced joint*,

$$Q(dy, dz; h) := P_Y(dy; \mathsf{U}(h, z))\, P_Z(dz; h). \tag{1}$$

By construction, $Q_Z(\cdot; h) = P_Z(\cdot; h)$ and $Q_{Y|Z=z}(\cdot; h) = P_Y(\cdot; \mathsf{U}(h, z))$ for $P_Z(\cdot; h)$-almost every $z$. Thus $Q$ is the unique joint distribution whose $Z$-marginal represents the model's current beliefs about the update block and whose conditional distributions represent the predictions that the procedure will actually return after updating. The target marginal of this joint is

$$Q_Y(dy; h) = \int P_Y(dy; \mathsf{U}(h, z))\, P_Z(dz; h). \tag{2}$$

This is the model's post-update predictive averaged under its current beliefs about the update block, which we call the *expected future predictive*. The update-induced joint is the operationally relevant distribution for a procedure which first observes $Z$ and then $Y$. When the history $h$ is fixed, we suppress it, writing $P_Y$ and $P_Z$ for the current predictives, $P_Y^z := P_Y(\cdot; \mathsf{U}(h, z))$ for the post-update predictive, and $Q_Y$ for the expected future predictive, so that $Q_Y = \mathbb{E}_{P_Z}[P_Y^Z]$ and $Q_{Y|z} \equiv P_Y^z$.

**Predictive coherence.** A predictive procedure is coherent for the target $Y$, history $h$ and update block $Z$ when the current predictive and expected future predictive agree, $P_Y = Q_Y$. Coherence is exactly the statement that the current predictive is recovered as the $Y$-marginal of the update-induced joint. If coherence fails, no joint can simultaneously have $P_Z$ as its $Z$-marginal, $P_Y^z$ as its conditional distribution given $Z = z$, and $P_Y$ as its $Y$-marginal. Exact conditioning under a model-specified joint is sufficient for predictive coherence. The general definition, however, does not require the model to specify a joint law over $(Y, Z)$ independently of the update-induced joint $Q$.

**Local coherence.** Predictive coherence is a local condition: it is defined relative to a realised history $h$, a target $Y$, and a specified update setup comprising the ordered design $d$ and its corresponding update block $Z$. Coherence for one such setup does not imply coherence at another history, for another target, or under another update design. This locality is both a limitation and a feature. It limits the scope of any coherence guarantee, but focuses the requirement on the predictions and updates that are relevant to the problem under consideration. Incoherence outside this scope need not affect the intended use of the procedure. From a predictivist perspective, coherence concerns the compatibility of predictions about the observable quantities $Y$ and $Z$ across the histories reachable within this update setup. It therefore makes no general claim about the model's latent parameters, posterior approximation, or global Bayesianity. We call a predictive procedure globally coherent only if the coherence condition holds for every admissible history, target, and update design.

**Incoherence admits a Dutch book.** Incoherence can have direct consequences for tasks that rely on a predictive procedure. The canonical demonstration is a *diachronic Dutch book*: a sequence of contracts that the procedure regards as individually fair at the histories where they are offered, but whose combined payoff is a sure loss. The classical construction of Lewis [1999] shows qualitatively that such exploitation is possible whenever the procedure's current valuation of some payoff differs from its expected valuation after updating. We quantify the maximum severity of this exploitation under the betting protocol and a bounded-liability constraint. Writing $D_{\mathrm{TV}} := \mathrm{TV}(P_Y, Q_Y)$, the maximum fraction of initial wealth which can be extracted as a sure loss is $D_{\mathrm{TV}}/(1 + D_{\mathrm{TV}})$. This Dutch-book result concerns the internal consistency of the procedure's valuations and it does not imply that either predictive distribution is empirically inaccurate. Appendix A.1 gives the classical construction and derives the maximum sure-loss fraction.

### 2.1 Bayesian models and coherence

Exact Bayesian conditioning provides the canonical sufficient condition for predictive coherence, but it is often unavailable. The familiar prior-likelihood construction in Bayesian deep learning has an intractable parametric posterior which must be approximated. Practical approximations include sampling-based methods such as SG-MCMC [Ma et al., 2015], variational inference over a restricted

family [Blei et al., 2017], such as the diagonal Gaussian used by IVON [Shen et al., 2024], and the linearised Laplace approximation [Immer et al., 2021]. Whilst each targets a coherent Bayesian model, the approximations can induce incoherence in the resulting predictive procedure.

Amortised approaches such as neural processes [Garnelo et al., 2018a,b] and prior-data fitted networks [Müller et al., 2022] bypass explicit parametric inference and learn a direct map from an observed dataset to a predictive distribution. They do not perform approximate inference within an explicit Bayesian model, but approximate the posterior predictive defined by their training prior, and in practice are not exactly coherent either. Similarly, deep ensembles can be interpreted as approximating Bayesian model averaging [Lakshminarayanan et al., 2017, Wilson and Izmailov, 2020], despite their members not generally being samples from the posterior of a specified BNN [Wild et al., 2023]. These constructions demonstrate that a procedure can be approximately Bayesian without an explicit target model. A procedure's Bayesian provenance does not determine how severe its incoherence will be. We therefore separate construction from empirical behaviour.

## 3 Cost of incoherence

Incoherence matters when an action is taken before an update but valued by the predictions the procedure will report after it. We consider a decision problem spanning a single update. An initial action $a \in \mathcal{A}$ is taken, a block $Z^a$ is observed, the procedure is updated, a terminal action $b \in \mathcal{B}$ is chosen from the post-update predictive, and a loss $\ell(a, b, y)$ is realised at the target. The design of the block may depend on the initial action, so we index by the action, writing the update-induced joint as $Q^a(dy, dz) = P_Y^{a,z}(dy)P_Z^a(dz)$. Prediction-oriented experimental design is a problem of this form, and problems spanning several updates, as in model-based reinforcement learning, are covered when actions are chosen greedily. The terminal action depends on the procedure only through the post-update predictive, which is a conditional of $Q^a$. Of the predictions consumed in valuing the initial action, only the current predictive $P_Y$ can therefore disagree with $Q^a$.

**Action coherence.** We *coherentise* an objective by substituting $Q_Y^a$ for $P_Y$ wherever it appears, writing $J$ for the stated objective and $J^+$ for its coherentisation, and $a^*$ and $a^+$ for the actions they select. A procedure is *action coherent* at $h$ for $(\mathcal{A}, J)$ when coherentising leaves the decision unchanged, $a^* = a^+$, with ties broken identically. Predictive coherence for every block in $\{Z^a : a \in \mathcal{A}\}$ implies action coherence for every objective, but not conversely, so action coherence is a relaxation that registers only the incoherence that reaches the action. Objectives that coincide under coherence share the same coherentisation, so two such objectives selecting different actions is sufficient for at least one of them to be action incoherent. Coherentisation is internal to the procedure and is not a correction towards the truth, since $Q^a$ may describe the true process poorly. It removes inconsistency, not inaccuracy.

**Severity of incoherence.** What reaches the action is not the size of the discrepancy between $Q_Y^a$ and $P_Y$, but how it varies across candidates relative to how sharply the objective separates them, and how the objective consumes $P_Y$. A large discrepancy that is near-constant across candidates barely moves the decision, so detecting predictive incoherence establishes that a decision may be affected without indicating by how much. Incoherence also decouples objectives that coincide under coherence, and no member of the decoupled family is the correct one, since the current predictive and the expected future predictive disagree and neither is guaranteed to be the more accurate. Which member selects better actions under the true law is an empirical question we address in Section 6.

### 3.1 Objective splitting in experimental design

Bayesian experimental design (BED) [Rainforth et al., 2024] selects designs which maximise information about a downstream quantity. In prediction-oriented design, that quantity is the procedure's prediction at deployment. Here $\mathcal{A}$ is a set of candidate designs, $Z^a$ is the observation acquired at the chosen design, and the terminal action reports a predictive for a target $Y$ under the log score. The expected predictive information gain (EPIG) [Bickford Smith et al., 2023] acquires the design which most reduces predictive uncertainty at the target, $\mathrm{EPIG}_H(a) = H[P_Y] - \mathbb{E}_{Z \sim P_Z^a}[H[P_Y^{a,Z}]]$. Read decision-theoretically, $\mathrm{EPIG}_H$ is a design-free constant minus the procedure's expected log loss after the update, so maximising it minimises that loss. Read information-theoretically, EPIG is the mutual information between the target and the acquired observation under the joint the procedure induces over

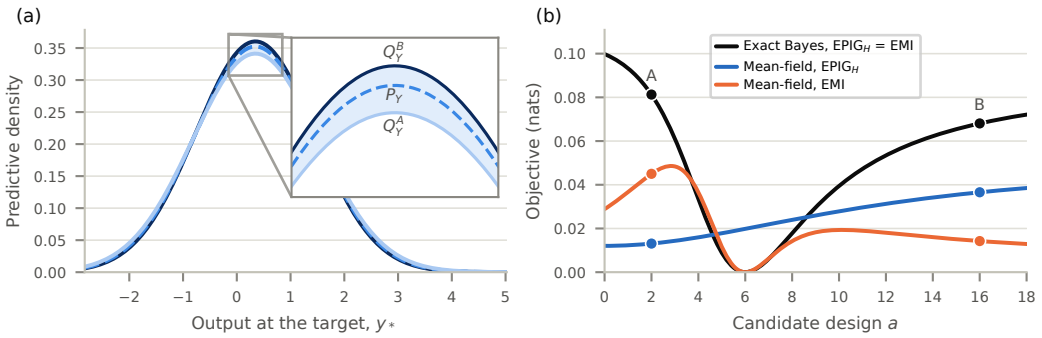


Figure 1: Objective splitting under mean-field VI for Bayesian linear regression. **(a)** The mean-field model's current predictive $P_Y$ (dashed) and its expected future predictives $Q_Y^A$ and $Q_Y^B$ after acquiring candidates $A$ and $B$. The inset magnifies the peak, where neither expected future predictive matches the current one, so the model is incoherent for both candidates. **(b)** $\mathrm{EPIG}_H$ and EMI against the candidate design $a$. Under exact Bayes the two coincide. Under mean-field VI they separate and rank the marked candidates in opposite orders. $\mathrm{EPIG}_H$ prefers $B$, whereas EMI prefers $A$.

them. Under incoherence that joint has $Y$-marginal $Q_Y^a$ rather than $P_Y$, and the mutual information becomes $\mathrm{EMI}(a) = H[Q_Y^a] - \mathbb{E}_{Z\sim P_Z^a}[H[P_Y^{a,Z}]]$, the coherentisation of $\mathrm{EPIG}_H$. The two differ only in their baseline, which is constant in $a$ for $\mathrm{EPIG}_H$ but varies with $a$ once coherentised, and so can change the ranking. A third standard form, the expected divergence from the current predictive, is related to both by

$$\mathrm{EPIG}_{\mathrm{KL}}(a) := \mathbb{E}_{Z\sim P_Z^a}\Big[\mathrm{KL}\Big(P_Y^{a,Z} \,\|\, P_Y\Big)\Big] = \mathrm{EMI}(a) + \mathrm{KL}(Q_Y^a \,\|\, P_Y)\,, \tag{3}$$

$$\mathrm{EPIG}_H(a) = \mathrm{EPIG}_{\mathrm{KL}}(a) + \mathbb{E}_{Q_Y^a}[\log p_Y] - \mathbb{E}_{P_Y}[\log p_Y], \tag{4}$$

where $p_Y$ is the density of $P_Y$ (Appendix A.2). All three forms coincide under coherence and coherentise to EMI (Appendix A.2), but under incoherence they are different acquisition rules, so even a choice ordinarily made for numerical convenience becomes a choice of acquisition rule. They also differ in how far they move. The KL form departs from EMI by a non-negative term that is second order in the discrepancy between $Q_Y^a$ and $P_Y$, approximately half their $\chi^2$ divergence, whereas the entropy form adds a signed term that is first order. For small incoherence the KL form therefore stays much closer to EMI, as Section 6 confirms.

We make the split concrete with two-parameter Bayesian linear regression under a Gaussian likelihood and prior, comparing exact conditioning against mean-field variational inference, whose factorised posterior cannot represent the intercept–slope correlation induced by training inputs clustered away from the origin. Both objectives are available in closed form, so any discrepancy is attributable to incoherence rather than estimation noise. Sweeping the candidate design (Figure 1), the objectives coincide under exact conditioning and separate under mean-field VI, where they rank two candidates in opposite orders. Section 6 measures how often this occurs, and what it costs, across model families on a neural testbed.

## 4 Measuring incoherence

Section 3 showed that incoherence can reach a decision. Since few procedures update by exact conditioning, the question that matters in practice is how incoherent a procedure is, rather than whether it is coherent at all. This section develops a scalar measure of that severity, computed from the procedure's own predictions and updates.

### 4.1 The incoherence ratio

Fix a history $h$, a target $Y$ and a design $d$ with block $Z$. Each draw from $Q_Y$ costs an update of the procedure, so updates are the budget, and we treat the procedure as a black box that can be updated and sampled. A *round* draws a block $Z_r \sim P_Z$, updates the fitted procedure on it, drawing any

randomness internal to the update afresh, and returns $P_Y^{Z_r}$, so rounds are independent and identically distributed. The subscript on $Z_r$ indexes rounds, not the components of the block. The Dutch book of Section 2 prices incoherence by $\mathrm{TV}(P_Y, Q_Y)$, but total variation is non-linear in the mixture $Q_Y$ and cannot be estimated without bias from $R$ post-update predictives. We instead use kernel mean embeddings [Gretton et al., 2012], which are linear in the mixture. For a kernel $k$ with feature map $\phi$ and $\|\phi(y)\|^2 \leq \kappa$, write $\mu_P$, $\mu_Q$ and $\mu_r$ for the embeddings of $P_Y$, $Q_Y$ and $P_Y^{Z_r}$, and $\delta_r := \mu_r - \mu_P$ for the displacement of round $r$. Then $\Delta := \mu_Q - \mu_P = \mathbb{E}[\delta_r]$ and $\mathrm{MMD}^2(P_Y, Q_Y) = \|\Delta\|^2$. For a characteristic kernel [Sriperumbudur et al., 2010], $\Delta = 0$ if and only if the procedure is coherent. Since $\mathrm{MMD} \leq 2\sqrt{\kappa}\, D_{\mathrm{TV}}$, however, the embedding discrepancy bounds the sure loss only from below, and a small $\Delta$ certifies only that incoherence is small in the features the kernel measures (Appendix B.2).

**The incoherence ratio.** The squared discrepancy $\|\Delta\|^2$ grows with how far a procedure's updates move its predictive, so procedures that update by different amounts cannot be compared on it. The bias–variance identity $\mathbb{E}\|\delta_r\|^2 = \|\Delta\|^2 + V$, with $V := \mathbb{E}\|\delta_r - \Delta\|^2$, separates the systematic part of an update from its response to the block, and we define the *incoherence ratio* as

$$\Psi := \frac{\|\Delta\|^2}{V} = \frac{\mathrm{MMD}^2(P_Y, Q_Y)}{\mathbb{E}\big[\mathrm{MMD}^2(P_Y^{Z_r}, Q_Y)\big]}, \tag{5}$$

the squared signal-to-noise ratio of the update. It is invariant to rescaling the displacements, and $\Psi/(1+\Psi)$ is the systematic share of the expected squared displacement, so $\Psi = 1$ means that bias and spread contribute equally. For a characteristic kernel, $\Psi$ vanishes exactly under coherence, and a shift that ignores the block, with $V = 0$ and $\Delta \neq 0$, has $\Psi = \infty$ (Appendix B.2). $\Psi$ is a self-consistency check that needs no test set, reference posterior or oracle. Blocks are drawn from $P_Z$ rather than from data because coherence concerns the procedure's beliefs about its own future, and measuring on observed blocks would mix inconsistency with inaccuracy. Like coherence, $\Psi$ is a property of the procedure for the update problem $(h, Y, d)$, including the size of the block, which should match the update the procedure will face. Its level depends on the kernel and the design, so it ranks procedures within a problem rather than on an absolute scale. Since $V$ includes any randomness internal to the update, a noisier update has a lower $\Psi$ for the same bias, and Appendix B.2 separates the spread driven by the block.

**Estimation.** $\Psi$ is estimated from $R$ rounds, which can be shared with the test of Section 5. U-statistics over rounds estimate $\|\Delta\|^2$ and $V$ without bias, reducing with exact embeddings to $\|\hat{\Delta}\|^2 - \hat{V}/R$ and $\hat{V}$, and the ratio has an $O(R^{-1})$ bias and a leave-one-round-out standard error (Appendix B.1). Near coherence the numerator is a degenerate U-statistic and estimates can be negative, which we report unclipped. Displacements at several evaluation points are pooled as a ratio of summed numerators to summed denominators, which keeps scale invariance. For finite targets we use the Dirac kernel, and for continuous targets random Fourier features [Rahimi and Recht, 2007], which make estimation cheap but are not characteristic. We set their lengthscale to the noise scale of the problem, and in Section 6 $\Psi$ is shown to be robust to this choice.

### 4.2 Measuring action incoherence

$\Psi$ is decision-agnostic. Whether incoherence reaches an action is defined by the coherentised objective $J^+$, which is rarely computable, since it consumes $Q_Y^a$ through a non-linear functional at every candidate. Because objectives that coincide under coherence share a coherentisation (Section 3), the rate at which two tractable objectives from this class disagree witnesses action incoherence, although agreement does not certify its absence. The two measures answer different questions. The disagreement rate concerns a particular pair of objectives and decision problem. $\Psi$ needs neither, so it can compare procedures before a decision problem is fixed, audit an approximation choice, and attribute a failed sequential procedure to inconsistent updates rather than inaccurate predictions. The two are linked in one direction, since $\Psi = 0$ for every block in $\{Z^a : a \in \mathcal{A}\}$, with a characteristic kernel, rules out action incoherence. A positive $\Psi$ diagnoses susceptibility, the distortion of the objective across candidates. Whether that distortion changes a choice also depends on the margin between candidates, and what a changed choice costs is evaluated under the true law rather than the procedure's own beliefs (Section 3), so neither is determined by $\Psi$. Coherentisation removes inconsistency rather than inaccuracy, so realised performance on an oracle or held-out set remains the decision-relevant comparison, and Section 6 examines how susceptibility relates to it.

# 5 Testing coherence

The incoherence ratio grades how incoherent a procedure is, but it cannot establish that a procedure is coherent. Near coherence its numerator is a degenerate U-statistic, and $\hat{\Psi} \pm 2\,\mathrm{SE}$ is not a confidence interval. Whether a procedure is exactly coherent matters most when its incoherence ratio is near zero and its joint is implicit in its update rule, so that coherence cannot be settled by construction. A heuristic procedure that is rejected is shown to be incoherent with high probability. We therefore test $H_0 : P_Y = Q_Y$. The classical MMD statistic is degenerate under this null and cannot feasibly be permutation-calibrated, since post-update samples are tied to their blocks. We instead adopt testing by betting [Shafer, 2021, Ramdas et al., 2023, Grünwald et al., 2024], whose anytime validity lets the test stop once the evidence is decisive and so spend no more updates than is needed, and adapt sequential kernel tests [Shekhar and Ramdas, 2024, Podkopaev et al., 2023] to a null in which one side is a mixture indexed by updates.

## 5.1 Sequential kernel test of coherence

With the kernel, embeddings and displacements of Section 4.1, coherence implies the kernel null $H_0^k : \mathbb{E}[\delta_r] = 0$ for any kernel, so validity does not depend on the kernel, and for a characteristic kernel the two nulls coincide. A restricted kernel measures coherence only in the features it selects, for instance the mean and variance under a quadratic kernel, and the test inherits that restriction.

Each round draws $Z_r \sim P_Z$, updates the fitted procedure, and estimates $\delta_r$ by $\hat{\delta}_r$, the difference between the embeddings of $P_Y^{Z_r}$ and $P_Y$, computed exactly or from samples (Appendix B.3). Unlike the estimator of $\Psi$, the test draws fresh samples of $P_Y$ in every round. Rounds are then independent given the fitted procedure, so $\mathbb{E}[\hat{\delta}_r \mid \mathcal{F}_{r-1}] = \Delta$, where $\mathcal{F}_{r-1}$ is everything observed before round $r$. A pool shared across rounds would couple the bets through its Monte Carlo error and break this property. Before round $r$, the test forms the running mean $D_{r-1} := \frac{1}{r-1}\sum_{i<r} \hat{\delta}_i$ and the witness $u_r := D_{r-1}/\|D_{r-1}\|$, or $u_r := 0$ if $\|D_{r-1}\| \le \varepsilon$ for a fixed noise floor $\varepsilon$. It then bets on the projection of the new displacement onto the witness,

$$b_r := \frac{\langle u_r, \hat{\delta}_r \rangle}{2\sqrt{\kappa}}, \qquad E_r := E_{r-1}\,(1 + \lambda_r b_r), \qquad E_0 := 1, \tag{6}$$

where $b_r \in [-1, 1]$ since $\|\hat{\delta}_r\| \le 2\sqrt{\kappa}$, and the stake $\lambda_r \in [0, 1/2]$ is set from earlier rounds by online Newton steps [Hazan et al., 2007, Cutkosky and Orabona, 2018, Waudby-Smith and Ramdas, 2024]. Since $u_r$ and $\lambda_r$ depend only on $\mathcal{F}_{r-1}$, $\mathbb{E}[b_r \mid \mathcal{F}_{r-1}] = \langle u_r, \Delta\rangle/(2\sqrt{\kappa})$, which is zero under the null. The wealth $(E_r)$ is then a non-negative martingale with $\mathbb{E}[E_r] = 1$, and Ville's inequality [Ville, 1939] gives $\mathbb{P}_{H_0}(\sup_r E_r \ge 1/\alpha) \le \alpha$. Rejecting the first time $E_r \ge 1/\alpha$ therefore controls the type I error at every stopping time [Shafer et al., 2011], and no update is spent after the evidence is decisive.

Under an alternative with $\|\Delta\| > \varepsilon$, the witness converges to $\Delta/\|\Delta\|$ and $\log E_r$ grows linearly in $r$, so the test is consistent above the noise floor. Its rate is set mainly by the absolute bias $\|\Delta\|/(2\sqrt{\kappa})$ rather than by $\Psi$, because bets are scaled by the worst-case bound $2\sqrt{\kappa}$, which keeps the test valid without assumptions on the procedure but makes it conservative when updates are small (Appendix B.4). Failing to reject within a budget of updates does not certify coherence, since incoherence below the noise floor, outside the features of the kernel, or too small to accumulate evidence within the budget goes undetected.

**Testing at multiple evaluation points.** Coherence is local, but a procedure is used at many evaluation points, and incoherence at any of them matters. We test the joint null that $P_Y = Q_Y$ at every evaluation point $l = 1, \ldots, L$. A single update gives a displacement at every point, so testing at $L$ points costs no more updates than testing at one. Running the construction separately at each point gives wealths $E_{l,r}$, which share the same updates and are dependent, so their product is not a valid test martingale. Their average is, since an average of martingales with respect to a common filtration is itself a martingale, and we reject the first time $\bar{E}_r := \frac{1}{L}\sum_l E_{l,r} \ge 1/\alpha$. Since $\bar{E}_r \ge \max_l E_{l,r}/L$, incoherence confined to a single point costs at most $\log L$ in log-evidence relative to testing at that point alone (Appendix B.3).

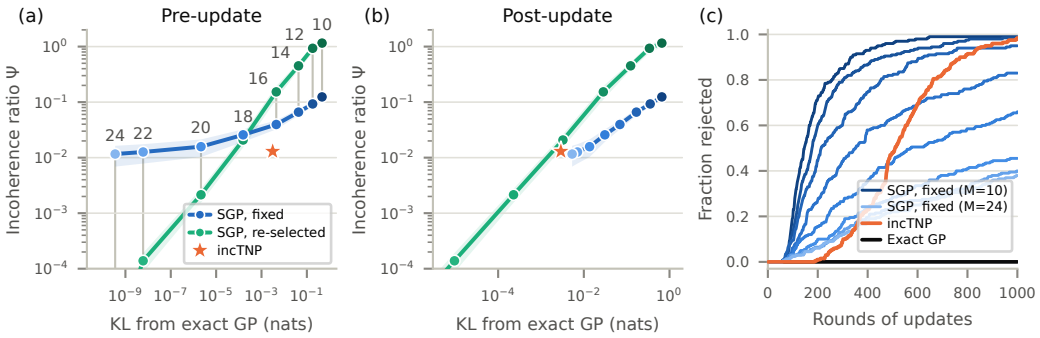


Figure 2: Sparse GPs with fixed (blue) or re-selected (green) inducing points, and an incTNP. **(a, b)** $\Psi$ against KL from the exact GP before and after the update. Vertical lines join procedures with the same number of inducing points, and bands show $\pm 2$ standard errors. **(c)** Fraction of tasks rejected at $\alpha = 0.05$. The exact GP is never rejected.

## 6 Experiments

The first two experiments ask whether incoherence measures what it should and what it reveals about models, in a controlled sparse GP setting and across models on real tabular datasets. The last two ask whether incoherence reaches decisions and what it costs, within mean-field Bayesian linear regression and across models on a synthetic neural-network testbed. Throughout, the block is pre-specified with $\Psi$ pooled over a set of evaluation points, and where we test coherence we do so at level $\alpha = 0.05$. Full experimental details are in Appendix C.

**Incoherence within sparse GPs.** We consider a controlled sparse GP setting, where the number of inducing points $M$ controls the resolution of the approximation. Functions are drawn from a GP prior, and each task observes $n = 24$ points before being updated on a block of $K = 8$ additional points, with incoherence measured at 64 evaluation points. We draw 20 functions (environments) and form 10 tasks from each by subsampling the observed, block and evaluation points, reporting the mean over tasks. Sparse GPs use the variational approximation of Titsias [2009], with the kernel hyperparameters fixed at their true values. Inducing points are chosen deterministically by greedy variance selection [Burt et al., 2020], and we compare an update procedure in which the inducing points are chosen before the update and held fixed with one in which they are re-selected during the update. The two procedures make identical initial predictions, so they only differ in how they update. We also include an incremental transformer neural process (incTNP) [Mortimer et al., 2026] trained on the same GP prior, which updates by extending its context (Appendix C.1). Within each update procedure, increasing the resolution of the approximation reduces incoherence and improves pre-update accuracy (Figure 2a). With fixed inducing points, incoherence plateaus since even when $M = n$, the update cannot adequately represent the new data. Re-selection avoids this, but with a fixed budget it must give up coverage of the existing data to accommodate the block, so its update departs from the current approximation, making it less coherent when $M$ is small and more coherent when $M$ is large. Re-selection is nevertheless more accurate after the update, even where it is the less coherent procedure (Figure 2b), so across update procedures incoherence does not order post-update accuracy. Rather, it registers that current and future predictions disagree, not which of them is wrong. The incTNP is as coherent as the fixed sparse GPs with nearly all the inducing points while fitting less accurately, and after the update it is more accurate than all of them, lying close to the re-selected curve, consistent with an approximation that adapts to new data. The test never rejects the exact GP, and within the fixed procedure rejects more coherent approximations more slowly (Figure 2c). Across families its speed to reject is not set by $\Psi$ alone, as the incTNP is rejected far faster than fixed sparse GPs of similar $\Psi$.

**Incoherence across models on tabular data.** We measure incoherence for models of several kinds on three regression and five classification datasets, over ten random splits (one for TabPFN), updating every model on blocks of further inputs with outputs drawn from the model itself (Appendix C.2). Exact methods have $\Psi$ indistinguishable from zero and are never rejected. The estimates are robust

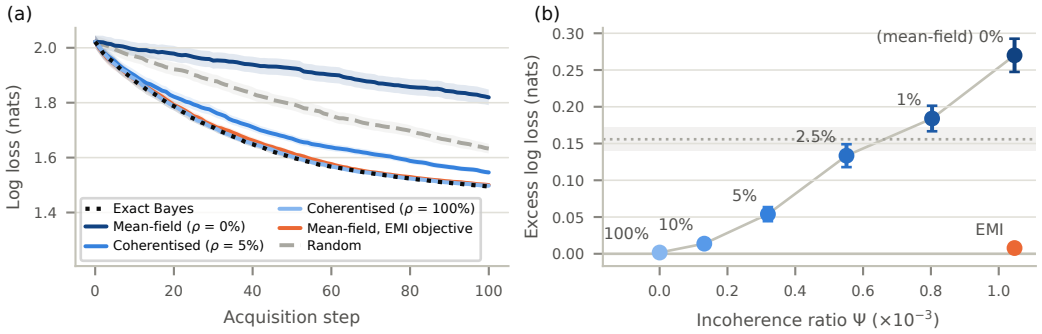


Figure 3: Acquisition with mean-field Bayesian linear regression. Every curve is the test log loss of the mean-field model on the data it acquired. **(a)** Test log loss by acquisition step. **(b)** Excess log loss over acquisition with the exact predictive, averaged over steps, against $\Psi$. Points are labelled by the share $\rho$ of the pool over which the acquisition predictive is made coherent, with $\rho = 0\%$ the mean-field model with EPIG. EMI changes only the objective, so it is plotted at the mean-field model's $\Psi$. Bands and error bars show $\pm 2$ standard errors.

to the kernel: on the regression datasets, halving or doubling the kernel lengthscale swaps only models whose $\Psi$ is within $\sim 15\%$ of each other. $\Psi$ responds to how a model updates, even when its current predictions are unchanged. Refitting a GP's hyperparameters at each update leaves its current predictions identical to the exact GP's but introduces a small incoherence. Refitting TabPFN's preprocessing on the updated context changes its $\Psi$ by up to a factor of six at similar log loss, although the two variants also differ before the update. An approximately Bayesian provenance is also no guarantee of coherence. For example, the update in BART reruns a sampler that mixes poorly within a run, and its median $\Psi$ of $0.71$ is comparable to those of nearest neighbours ($0.47$) and CatBoost ($0.90$). Plug-in logistic regression, by contrast, is about as coherent as the Laplace approximations, with a median $\Psi$ below $0.01$. The ordering of models is largely shared across datasets, with rank correlations of at least $0.9$ between them, but the level of $\Psi$ is not (kernel ridge ranges from $0.008$ to $0.12$), so $\Psi$ ranks models within a problem rather than on an absolute scale. Finally, accuracy does not imply coherence. TabPFN [Hollmann et al., 2025] has the lowest log loss on seven of the eight datasets, yet on every classification dataset its $\Psi$ is an order of magnitude or more above that of the full Laplace approximation, which is nearly coherent but less accurate (Table 5).

**Incoherence and acquisition within mean-field BLR.** Incoherence means that a model's current and future predictions disagree, and we now ask whether that disagreement reaches decisions. To isolate its effect, we hold the model and its predictions fixed and vary only the coherence of what the acquisition function consumes. We use Bayesian linear regression on random Fourier features and compare exact Bayes with mean-field variational inference, which we compute analytically so that any incoherence comes from the approximation rather than from optimisation error. Starting from 25 points, each model acquires 100 further points one at a time greedily from a pool, choosing each with either the entropy form of EPIG or EMI, and we average over 100 environments. With EPIG, the mean-field model acquires worse data than random acquisition (Figure 3a). With EMI, the same model nearly matches exact Bayes, with an excess of 0.008 nats, so the difference lies entirely in the objective. EMI is tractable here, but in general it requires the entropy of a mixture, so we also consider making the predictive itself coherent, building a ladder in which the mean-field predictive is made coherent with its own refit over a growing random share $\rho$ of the pool (Appendix C.3). Excess loss falls monotonically with $\Psi$ along the ladder, and coherentising over a small random subset already recovers most of the gap to exact Bayes (Figure 3b). This suggests that coherence need only hold over a representative selection of the pool rather than over every point the model might acquire.

**Incoherence and acquisition across models.** We now ask how incoherence affects acquisition across different approximate models in a more realistic setting. We use a synthetic neural-network testbed [Osband et al., 2022] of classification tasks generated by random networks, in which choosing one of several candidate batches to acquire is small enough that every labelling can be enumerated, so all quantities are exact (Appendix C.4). We consider eight approximate models (Laplace approximations, variational methods, deep ensembles, MC dropout and an incTNP) over 1000 environments.

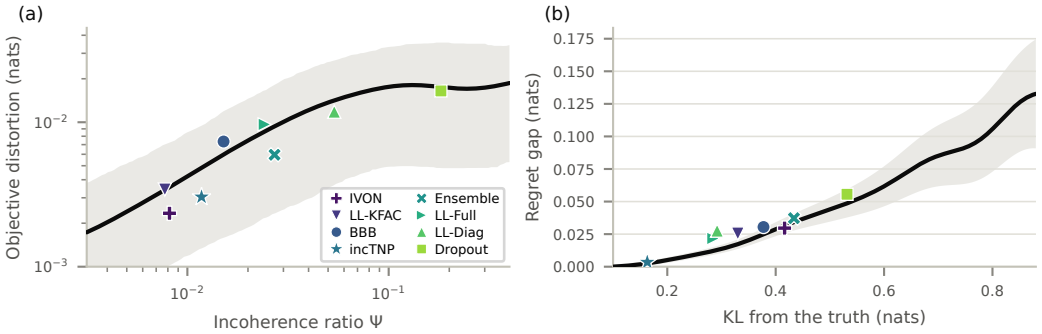


Figure 4: Acquisition on the neural testbed for eight approximate models. Exact Bayes is omitted, as its objectives never disagree. **(a)** Distortion of the acquisition objective against $\Psi$. **(b)** For decisions where EPIG and EMI choose differently, the excess loss of EPIG's choice against the model's KL from the truth before the update. Curves are local linear fits and markers show each model's mean. Bands show the 10th–90th percentiles in (a) and $\pm 2$ standard errors in (b).

We measure the cost of a disagreement between objectives as the excess loss of EPIG's choice over EMI's, and the distortion of the objective as the spread of EPIG minus EMI across candidates. Exact Bayes over a large discrete bank of networks, included as a coherent baseline, never separates the objectives, whereas for every approximate model the entropy form of EPIG disagrees with EMI on 26–80% of decisions and has a higher loss on average. The KL form of EPIG disagrees with EMI on only 1–20% of decisions, as the identity of Section 3.1 predicts, making it a potentially tractable substitute for EMI when a model is incoherent. $\Psi$ tracks how far incoherence distorts the objective (Figure 4a), and within an environment the ranking of models by $\Psi$ has a rank correlation of 0.73 with their distortion. This makes $\Psi$ a measure of how susceptible a model's decisions are to incoherence. It does not tell us whether a choice will flip, which also depends on the margin between the best candidates, or what a flip will cost, which is uncorrelated with $\Psi$ and instead tracks the model's accuracy (Figure 4b, Appendix C.4). Bayes by backprop and the incTNP, for example, are equally incoherent, yet the entropy form of EPIG disagrees with EMI on 80% of Bayes by backprop's decisions against 26% of the incTNP's, and the disagreements cost $0.025$ against $0.001$ nats.

## 7 Discussion

Many sequential decision procedures act as though a model's current and future beliefs are consistent. Measuring incoherence quantifies how far approximate models violate this, and flags when a procedure's decisions are exposed to it. Our experiments show this is more than a theoretical concern, as acquisition objectives that should coincide frequently disagreed and chose different data. Incoherence indicates when this disagreement can occur, but not what it costs, which depends on accuracy. Susceptibility can also depend on the objective. On the neural testbed, the KL form of EPIG stayed much closer to EMI than the entropy form, suggesting it as a cheap way to mitigate some of the effect of incoherence in BED. Whilst our experiments are restricted to BED, our framework applies to any single-update sequential decision problem. Understanding which properties of a problem determine how much incoherence matters would show where these diagnostics are most needed. Extending the framework to procedures that plan through a sequence of actions and updates would cover longer-horizon problems, where the effects of incoherence may compound. Beyond the choice of decision problem, the main limitations of measuring incoherence are the computational cost of many model updates and the scope of what it can tell us. The best models are both accurate and coherent, but coherence alone guarantees little, since exact conditioning in a misspecified model is perfectly coherent and can still be badly wrong. Measuring incoherence can reveal a potential problem, and help in explaining and addressing it, but it cannot substitute for held-out evaluation when choosing a model. Once an accurate model is chosen, making it exactly coherent is currently intractable, and approximate coherentisation over a representative subset of the pool has so far been shown to help only in a simple setting. It nonetheless points to a promising direction. Models that excel at static prediction are rarely designed for sequential decisions, and approaches that account for incoherence could help them perform well there too.

## Acknowledgments and Disclosure of Funding

JR was supported by the EPSRC Centre for Doctoral Training in Statistics and Machine Learning (StatML), grant EP/Y034813/1. VS was supported by the EPSRC Centre for Doctoral Training in Modern Statistics and Statistical Machine Learning (StatML), grant EP/S023151/1. The authors declare no competing interests.

## References

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

Tom Rainforth, Adam Foster, Desi Ivanova, and Freddie Bickford Smith. Modern Bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024. doi: 10.1214/23-STS915.

Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. Prediction-oriented Bayesian active learning. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 7331–7348. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/bickfordsmith23a.html`.

Bas C. van Fraassen. Belief and the will. *The Journal of Philosophy*, 81(5):235–256, 1984.

David Lewis. *Why Conditionalize?*, pages 403–407. Cambridge Studies in Philosophy. Cambridge University Press, 1999. doi: 10.1017/CBO9780511625343.024.

Michael Goldstein. The prevision of a prevision. *Journal of the American Statistical Association*, 78 (384):817–819, 1983.

Edwin Fong, Chris Holmes, and Stephen G. Walker. Martingale posterior distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(5):1357–1391, 2023. doi: 10.1093/jrsssb/qkad005.

Bruno de Finetti. La prévision: ses lois logiques, ses sources subjectives. *Annales de l'Institut Henri Poincaré*, 7:1–68, 1937.

Bruno Kacper Mlodozeniec, David Krueger, and Richard E. Turner. Implicitly Bayesian prediction rules in deep learning. In *Proceedings of the 6th Symposium on Advances in Approximate Bayesian Inference*, volume 253 of *Proceedings of Machine Learning Research*, pages 79–110. PMLR, 2024. URL `https://proceedings.mlr.press/v253/mlodozeniec24a.html`.

Fabian Falck, Ziyu Wang, and Christopher C. Holmes. Is in-context learning in large language models Bayesian? A martingale perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12784–12805. PMLR, 2024.

Gábor Pituk, Vik Shirvaikar, and Tom Rainforth. Do Bayesian neural networks actually behave like Bayesian models? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 49420–49458. PMLR, 2025. URL `https://openreview.net/forum?id=x5RQnF7Vw9`.

Thomas Nagler and David Rügamer. Uncertainty quantification for prior-data fitted networks using martingale posteriors. *arXiv preprint arXiv:2505.11325*, 2025.

Zheng Wen, Ian Osband, Chao Qin, Xiuyuan Lu, Morteza Ibrahimi, Vikranth Dwaracherla, Mohammad Asghari, and Benjamin Van Roy. From predictions to decisions: The importance of joint predictive distributions. *arXiv preprint arXiv:2107.09224*, 2022.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Xiuyuan Lu, Morteza Ibrahimi, Dieterich Lawson, Botao Hao, Brendan O'Donoghue, and Benjamin Van Roy. The neural testbed: Evaluating joint predictions. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yi-An Ma, Tianqi Chen, and Emily B. Fox. A complete recipe for stochastic gradient MCMC. In Corinna Cortes, Neil Lawrence, Daniel Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 2917–2925. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/hash/9a4400501febb2a95e79248486a5f6d3-Abstract.html.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773.

Yuesong Shen, Nico Daheim, Bai Cong, Peter Nickl, Gian Maria Marconi, Bazan Clement Emile Marcel Raoul, Rio Yokota, Iryna Gurevych, Daniel Cremers, Mohammad Emtiyaz Khan, and Thomas Möllenhoff. Variational learning is effective for large deep networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 44665–44686. PMLR, 2024. URL https://proceedings.mlr.press/v235/shen24b.html.

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of Bayesian neural nets via local linearization. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 703–711. PMLR, 2021. URL https://proceedings.mlr.press/v130/immer21a.html.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes. In *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018a.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1704–1713. PMLR, 10–15 Jul 2018b. URL https://proceedings.mlr.press/v80/garnelo18a.html.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, pages 6402–6413. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in Neural Information Processing Systems*, volume 33, pages 4697–4708. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/322f62469c5e3c7dc3e58f5a4d1ea399-Abstract.html.

Veit David Wild, Sahra Ghalebikesabi, Dino Sejdinovic, and Jeremias Knoblauch. A rigorous link between deep ensembles and (Variational) Bayesian methods. In *Advances in Neural Information Processing Systems*, volume 36, pages 39782–39811. Curran Associates, Inc., 2023. doi: 10.52202/075280-1729. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/7d25b1db211d99d5750ec45d65fd6e4e-Abstract-Conference.html.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R.G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11(50):1517–1561, 2010. URL http://jmlr.org/papers/v11/sriperumbudur10a.html.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

Glenn Shafer. Testing by betting: A strategy for statistical and scientific communication. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 184(2):407–431, 2021. doi: 10.1111/rssa.12647.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

Peter Grünwald, Rianne de Heide, and Wouter M. Koolen. Safe testing. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(5):1091–1128, 2024.

Shubhanshu Shekhar and Aaditya Ramdas. Nonparametric two-sample testing by betting. *IEEE Transactions on Information Theory*, 70(2):1178–1203, 2024.

Aleksandr Podkopaev, Patrick Blöbaum, Shiva Kasiviswanathan, and Aaditya Ramdas. Sequential kernelized independence testing. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27957–27993. PMLR, 2023.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2–3):169–192, 2007.

Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in Banach spaces. In *Proceedings of the 31st Conference on Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1493–1529. PMLR, 2018.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(1):1–27, 2024.

Jean Ville. *Étude critique de la notion de collectif*. Gauthier-Villars, Paris, 1939.

Glenn Shafer, Alexander Shen, Nikolai Vereshchagin, and Vladimir Vovk. Test martingales, Bayes factors and $p$-values. *Statistical Science*, 26(1):84–101, 2011. doi: 10.1214/10-STS347.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/titsias09a.html.

David R. Burt, Carl Edward Rasmussen, and Mark van der Wilk. Convergence of sparse variational inference in Gaussian processes regression. *Journal of Machine Learning Research*, 21(131):1–63, 2020. URL http://jmlr.org/papers/v21/19-1015.html.

Philip Mortimer, Cristiana Diaconu, Tommy Rochussen, Bruno Mlodozeniec, and Richard E. Turner. Incremental Transformer Neural Processes. In *Forty-third International Conference on Machine Learning*, 2026.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, Jan 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6. URL https://doi.org/10.1038/s41586-024-08328-6.

Paul Teller. Conditionalization and observation. *Synthese*, 26(2):218–258, Dec 1973. ISSN 1573-0964. doi: 10.1007/BF00873264. URL https://doi.org/10.1007/BF00873264.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10(1–2):1–141, 2017.

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948.

A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

Robert J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, 1980.

Ian Waudby-Smith, David Arbour, Ritwik Sinha, Edward H. Kennedy, and Aaditya Ramdas. Time-uniform central limit theory and asymptotic confidence sequences. *The Annals of Statistics*, 52(6): 2613–2640, 2024. doi: 10.1214/24-AOS2408.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. `https://kellerjordan.github.io/posts/muon/`, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

I-Cheng Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 1998a. DOI: `https://doi.org/10.24432/C5PK67`.

I-Cheng Yeh. Modeling of strength of high performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797–1808, 1998b. doi: 10.1016/S0008-8846(98)00165-3.

R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33 (3):291–297, 1997. doi: 10.1016/S0167-7152(96)00140-X.

J. Gerritsma, R. Onnink, and A. Versluis. Yacht Hydrodynamics. UCI Machine Learning Repository, 1981. DOI: `https://doi.org/10.24432/C5XG7R`.

Ross Quinlan. Statlog (Australian Credit Approval). UCI Machine Learning Repository, 1987. DOI: `https://doi.org/10.24432/C59012`.

Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. Heart Disease. UCI Machine Learning Repository, 1989. DOI: `https://doi.org/10.24432/C52P4X`.

Robert Detrano, Andras Janosi, Walter Steinbrunn, Matthias Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern H. Guppy, Stella Lee, and Victor Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5):304–310, 1989. doi: 10.1016/0002-9149(89)90524-9.

Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: `https://doi.org/10.24432/C5NC77`.

Pete Mowforth and Barry Shepherd. Statlog (Vehicle Silhouettes). UCI Machine Learning Repository, 1987. DOI: `https://doi.org/10.24432/C5HG6N`.

William Wolberg, Olvi Mangasarian, Nick Street, and W. Street. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository, 1993. DOI: `https://doi.org/10.24432/C5DW2B`.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198.

Radford M. Neal. MCMC using Hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin L. Jones, and Xiao-Li Meng, editors, *Handbook of Markov Chain Monte Carlo*, pages 113–162. Chapman and Hall/CRC, 2011.

David J. C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.5.720. URL `https://doi.org/10.1162/neco.1992.4.5.720`.

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010. doi: 10.1214/09-AOAS285.

Jingyu He, Saar Yalov, and P. Richard Hahn. XBART: Accelerated Bayesian additive regression trees. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1130–1138. PMLR, 2019.

Andrew Herren, P. Richard Hahn, Jared Murray, and Carlos Carvalho. StochTree: BART-based modeling in R and Python. *arXiv preprint arXiv:2512.12051*, 2025.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 6638–6648. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf`.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256. PMLR, 2010.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2408–2417. PMLR, 2015.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Nicol N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738, 2002. doi: 10.1162/08997660260028683.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/blundell15.html`.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059. PMLR, 2016.

# A Theoretical foundations

## A.1 Diachronic Dutch book

We follow the diachronic Dutch book of Teller [1973] and Lewis [1999], recast in the notation of Section 2. Fix the history $h$, the target $Y$ and the update design $d$ with block $Z$, and suppress them

where unambiguous. The procedure acts as an agent who regards a contract as fair at the price its current predictive assigns to the contract's payoff, and who will buy or sell any such contract at that price. For a payoff $f : \mathcal{Y} \to [0, 1]$, paid once $Y$ is observed, the agent's fair prices before and after the update are

$$\mathcal{V}_0(f) = \mathbb{E}_{P_Y}[f(Y)], \qquad \mathcal{V}_z(f) = \mathbb{E}_{P_Y^z}[f(Y)]. \tag{7}$$

Before the update, the agent also prices a contract paying the random future price $\mathcal{V}_Z(f)$ once $Z$ is revealed. Its fair price is

$$\overline{\mathcal{V}}(f) = \mathbb{E}_{P_Z}[\mathcal{V}_Z(f)] = \mathbb{E}_{Q_Y}[f(Y)], \tag{8}$$

where the second equality follows from (2). Restricting payoffs to $[0, 1]$ keeps every contract a bounded liability.

**Construction.** Suppose $\Delta(f) := \mathcal{V}_0(f) - \overline{\mathcal{V}}(f) > 0$. If instead $\Delta(f) < 0$, replace $f$ by $1 - f$, which reverses the sign. At a stake $\alpha > 0$, a bookmaker executes three trades, each at the agent's price.

(A) At $t = 0$, sell the agent a contract paying $\alpha f(Y)$ at $t = 2$, for $\alpha\mathcal{V}_0(f)$.

(B) At $t = 0$, buy from the agent a contract paying $\alpha\mathcal{V}_Z(f)$ at $t = 1$, for $\alpha\overline{\mathcal{V}}(f)$.

(C) At $t = 1$, after $Z = z$ is observed, buy from the agent a contract paying $\alpha f(Y)$ at $t = 2$, for $\alpha\mathcal{V}_z(f)$.

Starting from wealth 1, the agent holds $W_0 = 1 - \alpha\Delta(f)$ after the trades at $t = 0$. At $t = 1$ it pays $\alpha\mathcal{V}_z(f)$ to settle B and receives the same amount for C, and at $t = 2$ the payoffs of A and C cancel. Its terminal wealth is therefore $W_2 = W_0 = 1 - \alpha\Delta(f)$ and whatever $Z$ and $Y$ turn out to be, this is a sure loss of $\alpha\Delta(f)$.

**Bounded liability.** The agent must be able to settle B before receiving payment for C, so $W_0 - \alpha\mathcal{V}_z(f) \geq 0$ for every $z$. Since $\mathcal{V}_z(f) \leq 1$, this holds for all $z$, and for every payoff, whenever

$$1 - \alpha\Delta(f) - \alpha \geq 0 \quad \iff \quad \alpha \leq \frac{1}{1 + \Delta(f)}. \tag{9}$$

At the largest stake admissible under this constraint the sure loss is

$$1 - W_2 = \frac{\Delta(f)}{1 + \Delta(f)}. \tag{10}$$

**Maximal sure loss.** The loss in (10) is increasing in $\Delta(f)$, so the largest sure loss over payoffs $f : \mathcal{Y} \to [0, 1]$ is attained by maximising $\Delta(f)$. Since $\sup_{f:\mathcal{Y}\to[0,1]} |\mathbb{E}_{P_Y}[f] - \mathbb{E}_{Q_Y}[f]| = \mathrm{TV}(P_Y, Q_Y)$, attained by the indicator of the set on which $P_Y$ exceeds $Q_Y$, the maximal fraction of initial wealth that can be extracted as a sure loss is

$$\frac{D_{\mathrm{TV}}}{1 + D_{\mathrm{TV}}}, \qquad D_{\mathrm{TV}} := \mathrm{TV}(P_Y, Q_Y). \tag{11}$$

It vanishes if and only if $P_Y = Q_Y$, that is, exactly when the procedure is coherent for $Y$, $h$ and $Z$. The result concerns the internal consistency of the procedure's prices and says nothing about whether either predictive is accurate.

### A.2 Forms of EPIG

This section relates the three forms of EPIG in Section 3.1 and shows why the KL form stays closer to EMI than the entropy form. Fix the history $h$ and the candidate design $a$, and write $p_Y$ and $q_Y^a$ for the densities of $P_Y$ and $Q_Y^a$. We assume that $P_Y^{a,z} \ll P_Y$ for $P_Z^a$-almost every $z$, and that the entropies involved are finite.

**Relating the three forms.** Since $Q_Y^a$ is the average of the post-update predictives, averaging $\log p_Y$ over the post-update predictives gives $\mathbb{E}_{Z\sim P_Z^a}\mathbb{E}_{P_Y^{a,z}}[\log p_Y] = \mathbb{E}_{Q_Y^a}[\log p_Y]$ by Fubini's theorem. Expanding the KL divergence inside $\mathrm{EPIG}_{\mathrm{KL}}$ therefore gives

$$\mathrm{EPIG}_{\mathrm{KL}}(a) = -\mathbb{E}_{Z\sim P_Z^a}\big[H[P_Y^{a,Z}]\big] - \mathbb{E}_{Q_Y^a}[\log p_Y].$$

Writing $-\mathbb{E}_{Q_Y^a}[\log p_Y] = H[Q_Y^a] + \mathrm{KL}(Q_Y^a \,\|\, P_Y)$, and comparing with the definitions of EMI and $\mathrm{EPIG}_H$, gives

$$\begin{aligned}\mathrm{EPIG}_{\mathrm{KL}}(a) &= \mathrm{EMI}(a) + \mathrm{KL}(Q_Y^a \,\|\, P_Y),\\ \mathrm{EPIG}_H(a) &= \mathrm{EPIG}_{\mathrm{KL}}(a) + \mathbb{E}_{Q_Y^a}[\log p_Y] - \mathbb{E}_{P_Y}[\log p_Y].\end{aligned}$$

When the procedure is coherent, $Q_Y^a = P_Y$, both additional terms vanish, and all three forms equal EMI. Replacing $P_Y$ by $Q_Y^a$ in either $\mathrm{EPIG}_H$ or $\mathrm{EPIG}_{\mathrm{KL}}$ gives EMI directly, so all three forms coherentise to EMI.

**Size of the departures.** Both forms depart from EMI by terms that vanish under coherence, but at different rates. Write $q_Y^a = p_Y(1+\eta g_a)$, where $\mathbb{E}_{P_Y}[g_a] = 0$ and $g_a$ is bounded, so that $\eta$ sets the size of the incoherence. Expanding $(1+x)\log(1+x) = x + x^2/2 + O(x^3)$ gives

$$\mathrm{KL}(Q_Y^a \,\|\, P_Y) = \tfrac{\eta^2}{2}\,\mathbb{E}_{P_Y}[g_a^2] + O(\eta^3) = \tfrac{1}{2}\chi^2(Q_Y^a \,\|\, P_Y) + O(\eta^3),$$

so the KL form departs from EMI only at second order. The entropy form adds a further term, $\mathbb{E}_{Q_Y^a}[\log p_Y] - \mathbb{E}_{P_Y}[\log p_Y] = \eta\,\mathbb{E}_{P_Y}[g_a \log p_Y]$, which is of first order and can take either sign.

**Consequence for rankings.** Only differences between candidates affect which design is chosen. Let $a^+$ maximise EMI, and let $m := \min_{a\neq a^+}\{\mathrm{EMI}(a^+) - \mathrm{EMI}(a)\}$ be its margin over the next best candidate. Since KL divergence is non-negative, the first identity gives

$$\mathrm{EPIG}_{\mathrm{KL}}(a^+) - \mathrm{EPIG}_{\mathrm{KL}}(a) \geq m - \mathrm{KL}(Q_Y^a \,\|\, P_Y) \qquad \text{for every } a \neq a^+,$$

so the KL form selects the same design as EMI whenever $m > \max_a \mathrm{KL}(Q_Y^a \,\|\, P_Y)$. The margin need only exceed a second-order quantity. The entropy form has no comparable guarantee, because its additional term can favour any candidate at first order.

## B Measuring and testing incoherence

Throughout, fix a history $h$, a target $Y$ and a design $d$ with block $Z$, and a kernel $k$ on $\mathcal{Y}$ with feature map $\phi$ and $\|\phi(y)\|^2 \leq \kappa$. Round $r$ draws $Z_r \sim P_Z$ and returns the post-update predictive $P_Y^{Z_r}$, whose mean embedding [Muandet et al., 2017] we write $\mu_r$. As in Section 4.1,

$$\begin{aligned}&\delta_r := \mu_r - \mu_P,\\ &\Delta := \mu_Q - \mu_P = \mathbb{E}[\delta_r],\\ &V := \mathbb{E}\big[\|\delta_r - \Delta\|^2\big] = \mathbb{E}\big[\|\mu_r - \mu_Q\|^2\big].\end{aligned}$$

Here $Q_Y$ averages the post-update predictive over both the block and any randomness internal to the update. Rounds start from the same fitted procedure and draw both afresh, so $\mu_1, \ldots, \mu_R$ are independent and identically distributed with mean $\mu_Q$.

### B.1 Estimation

Every estimator below is a U-statistic [Hoeffding, 1948], an average of a kernel over distinct pairs, of rounds and, where samples are used, of samples. Excluding the pairing of an element with itself is what makes each one unbiased, as for the unbiased MMD estimator of Gretton et al. [2012]. We use two regimes. When $P_Y$ and every $P_Y^{Z_r}$ have closed-form embeddings, the only randomness is over rounds. Otherwise each predictive is summarised by samples, and the U-statistics also run over pairs of samples, respecting the block structure in which samples are grouped by round.

**Closed-form regime.** The numerator and denominator are order-two U-statistics over rounds,

$$\hat{N} := \frac{1}{R(R-1)} \sum_{r \neq r'} \langle \delta_r, \delta_{r'} \rangle,$$
$$\hat{V} := \frac{1}{2R(R-1)} \sum_{r \neq r'} \|\mu_r - \mu_{r'}\|^2,$$

with kernels $\langle \delta_r, \delta_{r'} \rangle$ and $\frac{1}{2}\|\mu_r - \mu_{r'}\|^2$, and $\hat{\Psi} := \hat{N}/\hat{V}$. For $r \neq r'$, independence of the rounds gives $\mathbb{E}[\langle \delta_r, \delta_{r'} \rangle] = \|\Delta\|^2$ and $\mathbb{E}\big[\frac{1}{2}\|\mu_r - \mu_{r'}\|^2\big] = V$, so both estimators are unbiased. Writing $\hat{\mu}_Q := \frac{1}{R}\sum_r \mu_r$, they take the familiar forms

$$\hat{V} = \frac{1}{R-1} \sum_{r=1}^{R} \|\mu_r - \hat{\mu}_Q\|^2,$$
$$\hat{N} = \|\hat{\mu}_Q - \mu_P\|^2 - \hat{V}/R,$$

the sample variance of the embeddings and a bias-corrected plug-in. The correction is needed because the covariance of $\hat{\mu}_Q$ has trace $V/R$, so $\mathbb{E}\big[\|\hat{\mu}_Q - \mu_P\|^2\big] = \|\Delta\|^2 + V/R$.

**Sampled regime.** Round $r$ provides $S$ samples $y_{r,1}, \ldots, y_{r,S}$ from $P_Y^{Z_r}$, drawn independently across rounds, and a single reference sample of $T$ draws $y'_1, \ldots, y'_T$ from $P_Y$ is shared by every round. Define the kernel averages

$$K_{rr'} := \frac{1}{S^2} \sum_{s,s'} k(y_{r,s}, y_{r',s'}) \quad (r \neq r'),$$
$$\tilde{K}_r := \frac{1}{S(S-1)} \sum_{s \neq s'} k(y_{r,s}, y_{r,s'}),$$
$$K_{rP} := \frac{1}{ST} \sum_{s,t} k(y_{r,s}, y'_t),$$
$$K_P := \frac{1}{T(T-1)} \sum_{t \neq t'} k(y'_t, y'_{t'}).$$

None of these pairs a sample with itself, so, given the blocks, they are unbiased for $\langle \mu_r, \mu_{r'} \rangle$, $\|\mu_r\|^2$, $\langle \mu_r, \mu_P \rangle$ and $\|\mu_P\|^2$ respectively. The estimators replace each inner product in the closed-form U-statistics by the corresponding kernel average,

$$\hat{N} := \frac{1}{R(R-1)} \sum_{r \neq r'} K_{rr'} - \frac{2}{R} \sum_r K_{rP} + K_P,$$
$$\hat{V} := \frac{1}{R} \sum_r \tilde{K}_r - \frac{1}{R(R-1)} \sum_{r \neq r'} K_{rr'},$$
$$\hat{\Psi} := \hat{N}/\hat{V}.$$

Taking expectations over the blocks, $\mathbb{E}[\langle \mu_r, \mu_{r'} \rangle] = \|\mu_Q\|^2$ for $r \neq r'$ by independence, and $\mathbb{E}[\langle \mu_r, \mu_P \rangle] = \langle \mu_Q, \mu_P \rangle$. Hence $\mathbb{E}[\hat{N}] = \|\mu_Q - \mu_P\|^2 = \|\Delta\|^2$ and $\mathbb{E}[\hat{V}] = \mathbb{E}\big[\|\mu_r\|^2\big] - \|\mu_Q\|^2 = V$. With exact embeddings in place of the kernel averages, both estimators reduce to those of the closed-form regime.

**Reusing the reference sample.** The reference sample enters $\hat{N}$ only, through $K_{rP}$ and $K_P$, and does not appear in $\hat{V}$. Reusing it across rounds correlates the terms of $\hat{N}$ but introduces no bias. Its Monte Carlo error, however, does not shrink with $R$. As $R \to \infty$, $\hat{N}$ still fluctuates about $\|\Delta\|^2$ by a term of order $\|\Delta\|\, T^{-1/2} + T^{-1}$, so the reference sample should be large relative to $S$, which is cheap. Splitting it in two and comparing the two resulting values of $\hat{N}$ gives a simple check.

**Cost.** With closed-form embeddings, or with an explicit feature map of dimension $D$ such as random Fourier features [Rahimi and Recht, 2007], each round is summarised by a vector in $\mathbb{R}^D$, and the estimators cost $O(RD)$, or $O(RSD)$ when each embedding is estimated from $S$ samples, with memory independent of $R$. With only a kernel, every round's samples must be stored and compared across rounds, at cost $O(R^2S^2)$.

**Pooling over evaluation points.** A round gives a displacement at every evaluation point $l = 1, \dots, L$ at no extra cost. Writing $\hat{N}_l$ and $\hat{V}_l$ for the estimators at point $l$, we report

$$\hat{\Psi} := \frac{\sum_l \hat{N}_l}{\sum_l \hat{V}_l},$$

which estimates $\Psi$ under the direct-sum kernel over evaluation points and so keeps its scale invariance. The pooled value weights each point by the size of its updates, and the per-point values $\hat{N}_l/\hat{V}_l$ localise incoherence.

**Standard errors.** Let $\hat{\Psi}_{(-r)}$ denote the estimate computed without round $r$, and $\bar{\Psi}$ the mean of these $R$ values. We report the leave-one-round-out standard error

$$\mathrm{SE} := \left( \frac{R-1}{R} \sum_{r=1}^{R} \big(\hat{\Psi}_{(-r)} - \bar{\Psi}\big)^2 \right)^{1/2},$$

which captures the variability of numerator and denominator jointly. In the sampled regime it is computed with the reference sample held fixed, so it measures variability over rounds conditional on that sample. Since $\hat{N}$ and $\hat{V}$ are unbiased, the ratio has an $O(R^{-1})$ bias by the delta method [van der Vaart, 1998].

**Behaviour near coherence.** When $\Delta = 0$, the conditional expectation $\mathbb{E}[\langle \delta_r, \delta_{r'} \rangle \mid \delta_r] = \langle \delta_r, \Delta \rangle$ vanishes, so $\hat{N}$ is a degenerate U-statistic [Serfling, 1980]. It is then of order $R^{-1}$ rather than $R^{-1/2}$, its limiting distribution is a centred weighted sum of $\chi^2$ variables, and it is negative with appreciable probability. We report $\hat{N}$ and $\hat{\Psi}$ unclipped, so that averages over tasks stay unbiased, and read negative values as indistinguishable from zero. The standard error also falls as $R^{-1}$ in this regime (Table 2). This is an empirical observation, since leave-one-out variance estimates are not in general consistent for degenerate U-statistics, but it remains a useful scale. $\hat{\Psi} \pm 2\,\mathrm{SE}$ is not, however, a confidence interval for a procedure that may be exactly coherent. That question is the test's.

### B.2 Properties

**Proposition 1** (Scale invariance)**.** *If every displacement is rescaled, $\delta_r \mapsto c\,\delta_r$ for some $c > 0$, then $\Psi$ is unchanged.*

*Proof.* By linearity, $\Delta \mapsto c\,\Delta$, and $V \mapsto c^2 V$. Hence $\Psi = \|\Delta\|^2/V \mapsto c^2\|\Delta\|^2/(c^2 V) = \Psi$. □

**The systematic share.** Since $\mathbb{E}\big[\|\delta_r\|^2\big] = \|\Delta\|^2 + V$,

$$\frac{\Psi}{1+\Psi} = \frac{\|\Delta\|^2}{\|\Delta\|^2 + V} = \frac{\|\Delta\|^2}{\mathbb{E}\big[\|\delta_r\|^2\big]} \in [0, 1)$$

is the share of the expected squared displacement that is systematic, so $\Psi = 1$ means that bias and spread contribute equally.

**The limit $V \to 0$.** If $V = 0$, every block produces the same displacement, so $\mu_r = \mu_Q$ for $P_Z$-almost every block, and for a characteristic kernel [Sriperumbudur et al., 2010] $P_Y^z = Q_Y$ for $P_Z$-almost every $z$. The procedure is then coherent if and only if $\Delta = 0$. If $\Delta \neq 0$, the update is a shift that does not depend on the block. This admits a Dutch book whatever the block turns out to be, and $\Psi = \infty$ is the correct reading. When $V$ is small but positive, as for a procedure that barely responds to the block, $\Psi$ reports whether the little movement there is tracks the block or drifts systematically. The estimators also return $\|\Delta\|^2$ and $V$ separately, for a reader who wants the absolute size of the bias.

**Randomised updates.** Let $\xi_r$ denote the randomness internal to the update in round $r$, so that $\delta_r = \delta(Z_r, \xi_r)$, and let $\bar{\delta}(z) := \mathbb{E}[\delta(z, \xi)]$ be the mean displacement for block $z$. The law of total variance gives $V = V_Z + V_\xi$, where

$$V_Z := \mathbb{E}\big[\|\bar{\delta}(Z_r) - \Delta\|^2\big] \quad \text{(spread driven by the block)},$$
$$V_\xi := \mathbb{E}\big[\|\delta_r - \bar{\delta}(Z_r)\|^2\big] \quad \text{(update noise)}.$$

Adding zero-mean noise to the update in embedding space leaves $\Delta$, and hence $Q_Y$ and the Dutch book, unchanged, while increasing $V$ and lowering $\Psi$. $\Psi$ therefore measures systematic incoherence against the total spread of the update. When update noise should be excluded, the block-driven ratio

$$\Psi_Z := \frac{\|\Delta\|^2}{V_Z} \geq \Psi$$

is the appropriate quantity, and the two coincide for deterministic updates. Estimating $\Psi_Z$ costs a second update per block. Given two independent updates $\delta_r^{(1)}$ and $\delta_r^{(2)}$ on the same block, $\mathbb{E}\big[\langle \delta_r^{(1)}, \delta_r^{(2)} \rangle\big] = \|\Delta\|^2 + V_Z$, so

$$\hat{V}_Z := \frac{1}{R} \sum_{r=1}^{R} \langle \delta_r^{(1)}, \delta_r^{(2)} \rangle - \hat{N}$$

is unbiased for $V_Z$, with $\hat{N}$ computed across rounds as before. Under the log score, a randomised update also scores no better in expectation than the update that reports its average predictive, by convexity of $-\log$, so update noise is not a free way to appear coherent.

**Embedding discrepancy and total variation.** Every $f$ in the unit ball of the RKHS satisfies $\|f\|_\infty \leq \sqrt{\kappa}$, so

$$\mathrm{MMD}(P, Q) = \sup_{\|f\| \leq 1} \big|\mathbb{E}_P[f] - \mathbb{E}_Q[f]\big| \leq 2\sqrt{\kappa}\, D_{\mathrm{TV}}(P, Q).$$

Since the sure loss $D_{\mathrm{TV}}/(1+D_{\mathrm{TV}})$ of Appendix A.1 increases with $D_{\mathrm{TV}}$, the embedding discrepancy bounds the sure loss from below, by the same expression with $D_{\mathrm{TV}}$ replaced by $\|\Delta\|/2\sqrt{\kappa}$, but gives no upper bound in general. For a finite $\mathcal{Y}$ with the Dirac kernel, $\phi(y) = e_y$, so $\|\Delta\| = \|q - p\|_2$ and $D_{\mathrm{TV}} = \frac{1}{2}\|q - p\|_1$, where $p$ and $q$ are the probability vectors of $P_Y$ and $Q_Y$. The inequalities $\|v\|_1/\sqrt{|\mathcal{Y}|} \leq \|v\|_2 \leq \|v\|_1$ then give

$$\frac{2D_{\mathrm{TV}}}{\sqrt{|\mathcal{Y}|}} \leq \|\Delta\| \leq 2D_{\mathrm{TV}},$$

so in the classification experiments the two agree up to a factor of $\sqrt{|\mathcal{Y}|}$.

### B.3 The sequential test

This section gives the details of the test of Section 5.1, as implemented in our experiments.

**Displacement estimates.** With closed-form embeddings, $\hat{\delta}_r := \mu_r - \mu_P$ is exact. Otherwise, with an explicit feature map, round $r$ draws $S$ samples $y_{r,1}, \ldots, y_{r,S}$ from $P_Y^{Z_r}$ and a fresh set of $T$ samples $y'_{r,1}, \ldots, y'_{r,T}$ from $P_Y$, and sets

$$\hat{\delta}_r := \frac{1}{S} \sum_{s=1}^{S} \phi(y_{r,s}) - \frac{1}{T} \sum_{t=1}^{T} \phi(y'_{r,t}).$$

Each average of features has norm at most $\sqrt{\kappa}$, so $\|\hat{\delta}_r\| \leq 2\sqrt{\kappa}$, which keeps every bet in $[-1, 1]$. Both kernels in our experiments have $\kappa = 1$. The Dirac kernel has $\phi(y) = e_y$, and random Fourier features are normalised so that $\|\phi(y)\|^2 = 1$.

**Why the test needs fresh samples.** Given the fitted procedure, the block, the update's internal randomness and both sets of samples are drawn afresh in each round, so $\hat{\delta}_r$ is independent of $\mathcal{F}_{r-1}$ and $\mathbb{E}[\hat{\delta}_r \mid \mathcal{F}_{r-1}] = \mu_Q - \mu_P = \Delta$. If instead every round reused one reference sample, its estimate $\hat{\mu}_P$ would be fixed by the past, and $\mathbb{E}[\hat{\delta}_r \mid \mathcal{F}_{r-1}] = \mu_Q - \hat{\mu}_P$. Under the null this is the Monte Carlo error of $\hat{\mu}_P$ rather than zero, so the bets would have a drift that the witness learns, and the test would eventually reject a coherent procedure. The batch estimator of Appendix B.1 has no such problem, since its unbiasedness is an unconditional statement.

**Witness and noise floor.** The witness is $u_r := D_{r-1}/\|D_{r-1}\|$, with $D_{r-1}$ the mean of the earlier displacement estimates, and $u_r := 0$ when $\|D_{r-1}\| \le \varepsilon$. We use $\varepsilon = 10^{-8}$, which guards against normalising a numerically zero vector, and with $D_0 := 0$ the first round never bets. Any rule for $u_r$ that depends only on $\mathcal{F}_{r-1}$ preserves validity, so the floor affects power but not the type I error.

**Stakes.** The stake is chosen by online Newton steps on the log-wealth [Hazan et al., 2007], in the form used for betting by Cutkosky and Orabona [2018] and Waudby-Smith and Ramdas [2024]. Each evaluation point keeps its own stake and curvature, initialised at $\lambda_1 := 0$ and $A_0 := 1$. After round $r$, with $b_r$ observed and the wealth already updated using $\lambda_r$,

$$\begin{aligned} g_r &:= \frac{b_r}{1 + \lambda_r b_r}, \\ A_r &:= A_{r-1} + g_r^2, \\ \lambda_{r+1} &:= \mathrm{clip}_{[0,1/2]}\Big(\lambda_r + c\,\frac{g_r}{A_r}\Big), \qquad c := \frac{2}{2 - \log 3} \approx 2.28, \end{aligned}$$

where $g_r$ is the derivative of $\log(1 + \lambda b_r)$ in $\lambda$ at $\lambda_r$. The curvature is updated before the step, so the step divides by $1 + \sum_{s \le r} g_s^2$. In one dimension clipping to $[0, 1/2]$ coincides with projection in the curvature norm. The stake $\lambda_{r+1}$ depends only on rounds up to $r$, so it is predictable. The cap at $1/2$ keeps $1 + \lambda_r b_r \ge 1/2$, which the logarithmic regret guarantee of online Newton steps against the best constant stake requires.

**Validity.** Fix an evaluation point. Under the null, $\mathbb{E}[\hat{\delta}_r \mid \mathcal{F}_{r-1}] = 0$, and since $u_r$ and $\lambda_r$ are $\mathcal{F}_{r-1}$-measurable,

$$\mathbb{E}[E_r \mid \mathcal{F}_{r-1}] = E_{r-1}\big(1 + \lambda_r\,\mathbb{E}[b_r \mid \mathcal{F}_{r-1}]\big) = E_{r-1}.$$

Since $1 + \lambda_r b_r \ge 1/2$, the wealth stays positive, so $(E_r)$ is a non-negative martingale with $E_0 = 1$. Ville's inequality gives $\mathbb{P}_{H_0}(\sup_r E_r \ge 1/\alpha) \le \alpha$, so rejecting the first time $E_r \ge 1/\alpha$ has type I error at most $\alpha$ at any stopping time.

**Multiple evaluation points.** Let $\mathcal{F}_r$ contain everything observed at every evaluation point up to round $r$. Each per-point wealth $(E_{l,r})$ is a non-negative martingale with respect to this common filtration, by the argument above, whatever the dependence between points. Their average $\bar{E}_r$ is therefore also a non-negative martingale with $\bar{E}_0 = 1$, and Ville's inequality applies to it. Equivalently, $\bar{E}_r$ is the product of predictable convex combinations of the per-point factors,

$$\bar{E}_R = \prod_{r=1}^{R} \sum_{l=1}^{L} w_{l,r}\,(1 + \lambda_{l,r} b_{l,r}), \qquad w_{l,r} := \frac{E_{l,r-1}}{\sum_{l'=1}^{L} E_{l',r-1}},$$

since each factor equals $\sum_l E_{l,r} / \sum_l E_{l,r-1}$ and the product telescopes, so the mixture concentrates on the points where evidence has accumulated. Since every $E_{l,r}$ is positive, $\bar{E}_r \ge \max_l E_{l,r}/L$, so $\log \bar{E}_r \ge \max_l \log E_{l,r} - \log L$. We compute $\log \bar{E}_r$ with a log-sum-exp over points.

**Consistency.** Under an alternative with $\|\Delta\| > \varepsilon$ at some point, the running mean $D_{r-1}$ converges to $\Delta$ by the strong law of large numbers, so the witness converges to $\Delta/\|\Delta\|$ and the bets have asymptotic mean $\|\Delta\|/(2\sqrt{\kappa}) > 0$. The log-wealth at that point then grows at the positive rate $g^\star$ of Appendix B.4, and by the bound above so does $\log \bar{E}_r$, so the test rejects with probability one.

### B.4 Severity and detectability

The incoherence ratio measures severity and the test of Section 5 detects incoherence, so it is natural to ask whether $\Psi$ determines how quickly the test rejects. It does not, and this section explains why.

**The oracle bet.** Above the noise floor, the test's witness converges to $u := \Delta/\|\Delta\|$. Consider the bet with this witness fixed,

$$b_r := \frac{\langle u, \hat{\delta}_r \rangle}{2\sqrt{\kappa}} \in [-1, 1],$$

where $\hat{\delta}_r$ is the estimated displacement. The bets are independent, with mean and variance

$$m := \frac{\|\Delta\|}{2\sqrt{\kappa}}, \qquad s^2 := \frac{\langle u, \hat{\Sigma} u \rangle}{4\kappa},$$

where $\hat{\Sigma}$ is the covariance of $\hat{\delta}_r$, equal to the covariance $\Sigma$ of the exact displacements plus any Monte Carlo noise from estimating the embeddings. Write $\rho := m^2/s^2$ for the signal-to-noise ratio along the bias. With exact embeddings, $\langle u, \Sigma u \rangle \le \operatorname{tr} \Sigma = V$, so $\rho \ge \Psi$, with equality only when the spread lies entirely along the bias. The test restricts stakes to $\lambda \in [0, 1/2]$, and online Newton steps [Hazan et al., 2007] achieve logarithmic regret against the best constant stake in that set [Cutkosky and Orabona, 2018, Waudby-Smith and Ramdas, 2024]. The log-wealth therefore grows asymptotically at rate

$$g^\star := \max_{\lambda \in [0, 1/2]} \mathbb{E}\big[\log(1 + \lambda b_r)\big],$$

and the test rejects after roughly $\log(1/\alpha)/g^\star$ updates.

**Two regimes.** For $x \ge -1/2$,

$$x - x^2 \;\le\; \log(1 + x) \;\le\; x - \tfrac{1}{2}x^2 + \tfrac{1}{3}x^3.$$

Apply these with $x = \lambda b_r$, using $\mathbb{E}[b_r^2] = m^2 + s^2$ and $b_r^3 \le b_r^2$. The lower bound gives $\mathbb{E}[\log(1 + \lambda b_r)] \ge \lambda m - \lambda^2(m^2 + s^2)$. The stake $\lambda = m/(2(m^2 + s^2))$ attains $\frac{1}{4}\rho/(1 + \rho)$ when it is at most $1/2$, and otherwise $\lambda = 1/2$ attains more than $m/4$. The upper bound gives $\mathbb{E}[\log(1 + \lambda b_r)] \le \lambda m - \lambda^2(m^2 + s^2)/3$ for $\lambda \le 1/2$, whose maximum is $\frac{3}{4}\rho/(1 + \rho)$, and $\log(1 + x) \le x$ gives $\mathbb{E}[\log(1 + \lambda b_r)] \le m/2$. Together,

$$g^\star \ge \tfrac{1}{4} \min\Big\{\frac{\rho}{1 + \rho},\; m\Big\},$$
$$g^\star \le \min\Big\{\tfrac{3}{4}\,\frac{\rho}{1 + \rho},\; \frac{m}{2}\Big\}.$$

The growth rate is therefore governed, up to constants, by the smaller of the systematic share along the bias, $\rho/(1 + \rho)$, and the absolute size of the bias relative to the kernel bound, $m = \|\Delta\|/(2\sqrt{\kappa})$. The second arises because bets are scaled by the worst-case bound $\|\hat{\delta}_r\| \le 2\sqrt{\kappa}$, which keeps every bet in $[-1, 1]$ and the wealth non-negative whatever the procedure does, and because the stake cap, which the regret guarantee of online Newton steps requires, prevents the test from betting more heavily on small bets. Displacements produced by a single update are typically far smaller than $2\sqrt{\kappa}$, so $m$ is small and the cap binds unless the signal-to-noise ratio is very low. In that regime the rate lies between $m/4$ and $m/2$, so it scales with the absolute bias and barely depends on the spread. Since $\Psi$ is scale invariant, two procedures with the same $\Psi$ but different absolute bias are rejected at different speeds, as in Figure 2c. $\Psi$ bounds the rate from below only in the low signal-to-noise regime, where $g^\star \ge \frac{1}{4}\rho/(1 + \rho) \ge \frac{1}{4}\Psi/(1 + \Psi)$ with exact embeddings.

**Conservativeness.** The test is valid at every stopping time without assumptions on the procedure, and the price is power when updates are small relative to the kernel bound. Rescaling bets by an estimated displacement scale would increase power, but any bet that could then exceed one in magnitude would have to be clipped, which breaks the mean-zero property under the null unless the displacements are symmetric. A tighter almost-sure bound on the displacement would allow larger effective stakes, but requires assumptions on how far an update can move the predictive. Asymptotic anytime-valid inference [Waudby-Smith et al., 2024] avoids the range bound and adapts to the variance of the bets, at the cost of validity that holds only asymptotically. We keep the non-asymptotic test because its intended use is auditing whether a procedure is exactly coherent, where validity matters more than speed, and use $\Psi$ to grade severity.

# C Experimental details

This appendix gives the details needed to reproduce each experiment in Section 6, one subsection per experiment, and the code is available at `https://github.com/jreaso/predictive-coherence`.

**Compute.** The sparse GP, tabular and Bayesian linear regression experiments ran on one laptop (Apple M3 Max, 36 GB), on its CPU with single-threaded workers. The sparse GP experiment took about 3 core-hours, and training its incTNP about 4 hours on the laptop's GPU. The tabular experiment took about 84 core-hours for the models other than TabPFN and 67 for TabPFN. The Bayesian linear regression experiment takes under a minute per setting. The neural testbed ran on one NVIDIA H100 (80 GB) in the cloud: tuning and training its incTNP took under two hours, the main run 5.1 hours, and the second draw of the random updates 50 minutes. Preliminary and abandoned runs used a comparable amount of compute, which is not included in these figures.

## C.1 Sparse GPs

We approximate a GP with sparse GPs of increasing resolution, set by the number of inducing points $M$, and with an amortised model, the incremental transformer neural process (incTNP). Because the data are drawn from the GP prior, the exact GP is the Bayesian ground truth, and we compare two procedures for updating the sparse GP against it.

**Prior and tasks.** We draw 20 datasets of 512 points from a GP prior on $[-2, 2]$ (squared-exponential kernel, lengthscale $0.25$, unit variance, noise standard deviation $0.1$) and split each at random 10 times into 24 context points, a block of size 8, and 64 evaluation points.

**Sparse GPs.** Inducing points are a subset of the context inputs, chosen by greedy variance selection [Burt et al., 2020], and $q(u)$ is the optimum of Titsias [2009]. The predictive includes the variance that the inducing points leave unexplained, but an update absorbs new data through the projected likelihood, which omits it, and this mismatch is the source of the incoherence. With fixed inducing points the update keeps them and updates $q(u)$, whereas re-selection chooses up to $M$ afresh from all 32 inputs and refits. The two share the same fit, so they predict identically before the update. We use $M = 6, 8, \ldots, 24$ with fixed inducing points and up to $M = 32$ with re-selection, and Figure 2 shows $M \geq 10$. Near $M = 24$, selection stops once the context is explained to numerical precision, so some tasks use fewer than $M$ inducing points.

**incTNP.** The incTNP uses the sequential architecture of Mortimer et al. [2026], which updates by extending a key–value cache rather than re-encoding the context, with five layers of width 128, eight attention heads and a Gaussian output for each target. We train it on sequences of 49 points from the same prior, predicting each point from those before it, for 400 000 steps with batches of 32. Hidden weight matrices are trained with Muon [Jordan et al., 2024] and the remaining parameters with AdamW [Loshchilov and Hutter, 2019], at learning rate $5 \times 10^{-4}$ and weight decay $0.01$, with 40 000 warm-up steps followed by cosine decay.

**Measuring and testing incoherence.** Each round draws the block's outputs from the model's own joint predictive, autoregressively for the incTNP, updates the fitted model on them, and embeds the updated predictive at the evaluation points. The kernel on outputs is a Gaussian kernel with lengthscale $0.1$, the noise standard deviation, approximated by 256 random Fourier features. Every predictive is Gaussian, so the embeddings are exact. We run 1000 rounds for each task and model, shared by the estimator and the test. For the exact and sparse GPs, $\Psi$ is also available in closed form, which Figure 2 plots, and the estimates agree with it within their standard errors. The figure reports the mean of $\Psi$ over tasks, with standard errors over environments. The test uses a noise floor of $10^{-8}$ and online Newton step stakes, and averages its wealth over the evaluation points.

**Accuracy.** Accuracy is the KL divergence from the exact GP's predictive to the model's, between their marginals at each evaluation point and averaged over the points. After the update, both are conditioned on the block's observed outputs. Figure 2 reports the geometric mean over tasks. $\Psi$ is the same in both panels, since it describes a single update of the fitted model.

**Results.** Table 1 gives the values behind Figure 2. Each run also recorded the estimate of $\Psi$ and its leave-one-round-out standard error after 10 and 100 rounds (Table 2). From 100 rounds the mean estimate matches the closed form. The standard error falls roughly as $R^{-1/2}$ for clearly incoherent models and roughly as $R^{-1}$ near coherence, where the U-statistic is degenerate.

Table 1: Results of the sparse GP experiment over 200 tasks. $\Psi$ is the mean over tasks with one standard error over environments, in closed form for the GP-based models and estimated from 1000 rounds for the incTNP. KL is the geometric mean over tasks, in nats per evaluation point, before and after the update, with values below $10^{-10}$ clamped before averaging. Rejected is the share of tasks rejected within 1000 rounds at level 0.05.

| Model | $M$ | $\Psi$ | KL before | KL after | Rejected |
|---|---|---|---|---|---|
| Exact GP | – | 0 | 0 | 0 | 0.00 |
| incTNP | – | $0.0130 \pm 0.0008$ | $3.1 \times 10^{-3}$ | $2.9 \times 10^{-3}$ | 0.99 |
| Fixed | 10 | $0.124 \pm 0.006$ | 0.47 | 0.66 | 0.99 |
| | 12 | $0.093 \pm 0.005$ | 0.18 | 0.35 | 0.99 |
| | 14 | $0.066 \pm 0.005$ | $4.3 \times 10^{-2}$ | 0.17 | 0.95 |
| | 16 | $0.040 \pm 0.004$ | $4.5 \times 10^{-3}$ | $7.0 \times 10^{-2}$ | 0.83 |
| | 18 | $0.026 \pm 0.003$ | $1.6 \times 10^{-4}$ | $3.1 \times 10^{-2}$ | 0.66 |
| | 20 | $0.016 \pm 0.003$ | $2.1 \times 10^{-6}$ | $1.4 \times 10^{-2}$ | 0.46 |
| | 22 | $0.013 \pm 0.002$ | $6.0 \times 10^{-9}$ | $7.1 \times 10^{-3}$ | 0.40 |
| | 24 | $0.012 \pm 0.002$ | $3.5 \times 10^{-10}$ | $5.3 \times 10^{-3}$ | 0.38 |
| Re-selected | 10 | $1.16 \pm 0.08$ | 0.47 | 0.66 | 1.00 |
| | 12 | $0.93 \pm 0.08$ | 0.18 | 0.34 | 1.00 |
| | 14 | $0.45 \pm 0.04$ | $4.3 \times 10^{-2}$ | 0.12 | 0.99 |
| | 16 | $0.153 \pm 0.021$ | $4.5 \times 10^{-3}$ | $2.9 \times 10^{-2}$ | 0.99 |
| | 18 | $0.021 \pm 0.003$ | $1.6 \times 10^{-4}$ | $3.3 \times 10^{-3}$ | 0.83 |
| | 20 | $0.0022 \pm 0.0003$ | $2.1 \times 10^{-6}$ | $2.3 \times 10^{-4}$ | 0.29 |
| | 22 | $(1.4 \pm 0.3) \times 10^{-4}$ | $6.0 \times 10^{-9}$ | $9.6 \times 10^{-6}$ | 0.02 |
| | 24 | $5 \times 10^{-6}$ | $3.5 \times 10^{-10}$ | $1.9 \times 10^{-7}$ | 0.00 |

Table 2: Estimates of $\Psi$ after $R$ rounds, as the mean over tasks $\pm$ the root mean square of the per-task leave-one-round-out standard errors. The estimates for different $R$ are prefixes of the same rounds.

| Model | Closed form | $R = 10$ | $R = 100$ | $R = 1000$ |
|---|---|---|---|---|
| Exact GP | 0 | $-0.005 \pm 0.090$ | $0.000 \pm 0.008$ | $0.0000 \pm 0.0007$ |
| incTNP | – | $0.017 \pm 0.093$ | $0.014 \pm 0.010$ | $0.013 \pm 0.002$ |
| Fixed, $M = 10$ | 0.124 | $0.127 \pm 0.141$ | $0.123 \pm 0.019$ | $0.124 \pm 0.005$ |
| Fixed, $M = 24$ | 0.012 | $0.016 \pm 0.111$ | $0.012 \pm 0.011$ | $0.012 \pm 0.002$ |
| Re-selected, $M = 20$ | 0.0022 | $0.001 \pm 0.093$ | $0.003 \pm 0.008$ | $0.0022 \pm 0.0008$ |

### C.2 Tabular data

We measure the incoherence of models in common use, on real datasets, and compare it with their accuracy. The models range from exactly Bayesian to no Bayesian provenance.

**Data and splits.** We use three regression and five classification datasets from OpenML (Table 3). The seven UCI datasets are released under the CC BY 4.0 licence, and california is listed as public on OpenML. Rows with missing values are dropped, categorical inputs are one-hot encoded, and inputs and regression outputs are standardised with the training set's mean and standard deviation. A split divides a dataset at random into a training set (half the rows, at most 1000), a test set (15%, at most 100) and a pool (the rest, at most 2000). Each round updates the model on a block a quarter the size of the training set, drawn without replacement from the pool. We use ten splits per dataset and 500 rounds per split, with the same splits and blocks for every model. TabPFN is run on the first split only, for 128 rounds, due to computational cost.

Table 3: Datasets and the sizes of each part of a split. The Inputs column counts features after one-hot encoding, and the pool is capped at 2000 rows. Sources: concrete [Yeh, 1998a,b], california [Pace and Barry, 1997], yacht [Gerritsma et al., 1981], australian [Quinlan, 1987], cleveland [Janosi et al., 1989, Detrano et al., 1989], german [Hofmann, 1994], vehicle [Mowforth and Shepherd, 1987] and wdbc [Wolberg et al., 1993], all fetched from OpenML [Vanschoren et al., 2013].

| Dataset | OpenML id | Rows | Inputs | Classes | Training | Test | Pool | Block |
|---|---|---|---|---|---|---|---|---|
| *Regression* | | | | | | | | |
| concrete | 44959 | 1030 | 8 | – | 515 | 100 | 415 | 128 |
| california | 44977 | 20640 | 8 | – | 1000 | 100 | 2000 | 250 |
| yacht | 42370 | 308 | 6 | – | 154 | 46 | 108 | 38 |
| *Classification* | | | | | | | | |
| australian | 40981 | 690 | 42 | 2 | 345 | 100 | 245 | 86 |
| cleveland | 194 | 297 | 25 | 5 | 148 | 44 | 105 | 37 |
| german | 31 | 1000 | 61 | 2 | 500 | 100 | 400 | 125 |
| vehicle | 54 | 846 | 18 | 4 | 423 | 100 | 323 | 105 |
| wdbc | 1510 | 569 | 30 | 2 | 284 | 85 | 200 | 71 |

**Models.** Every model is fitted on the training set and updated on the training set and block, and we describe them by how they update. Two update exactly. The GP, with an ARD squared-exponential kernel whose hyperparameters maximise the marginal likelihood, conditions exactly. Exact logistic regression, a softmax model with a Gaussian prior on its weights, represents its posterior by 8192 HMC draws [Neal, 2011] and updates by reweighting them, so it is exact Bayes with the draws as a discrete prior, and coherent by construction. The approximately Bayesian models recompute their approximation. The sparse GP takes the GP's hyperparameters and the optimal $q(u)$ of Titsias [2009], with inducing points chosen from the training inputs by greedy variance selection until the unexplained prior variance per input falls below the noise variance (29 to 100 points), and updates $q(u)$ with its inducing points fixed, as in Appendix C.1. The refitted GP refits the GP's hyperparameters on the training set and block before conditioning. The full and diagonal Laplace approximations [MacKay, 1992] share exact logistic regression's prior and move to the new mode, taking the full precision or its diagonal there. BART [Chipman et al., 2010] and XBART [He et al., 2019], run with `stochtree` version 0.4.5 [Herren et al., 2025] and its default priors and 200 trees, predict with the mixture over their draws (100 MCMC draws from one chain for BART, the last 25 of 40 grow-from-root sweeps for XBART) and update by rerunning the sampler with a fresh seed. The plug-in models refit. For regression, kernel ridge predicts a Gaussian at its fitted function and nearest neighbours a Gaussian at the mean of the $k$ nearest outputs, both with the mean squared leave-one-out residual as variance. For classification, logistic regression predicts the class probabilities at the mode of the Bayesian model above, and nearest neighbours the class frequencies among the $k$ neighbours. CatBoost [Prokhorenkova et al., 2018] predicts a mean and a variance for regression (`RMSEWithUncertainty` loss) and class probabilities for classification, and refits with its seed fixed at zero. TabPFN [Hollmann et al., 2025], with the v2.5 weights [Grinsztajn et al., 2025], released under the TabPFN-2.5 licence, and one estimator, conditions on the training set and updates by appending the block to its context. Its network-only variant switches off the package's input and target transforms, fingerprint feature, and feature and class shifts, keeping only the clipping of outlying inputs for classification. The variant with preprocessing uses the default configuration, whose transforms are refitted whenever the context changes, so the two variants differ before the update as well as in how they update. Plug-in models have no joint over a block, so by default their block's outputs are drawn independently from the current predictive. We also run kernel ridge, logistic regression and nearest neighbours with an autoregressive joint, which draws the outputs one at a time in random order, refitting after each. The two joints share the current predictive, and hence its accuracy. TabPFN draws its block autoregressively in the block's order.

**Tuning.** Kernel ridge's penalty and lengthscale, the number of neighbours, CatBoost's depth and number of trees, and the logistic prior scale are chosen once per dataset by five-fold cross-validated log likelihood on the first split's training set (Table 4). The first split's training set overlaps the other splits' test sets, by about half of their rows on every dataset but california.

Table 4: Tuned hyperparameters and noise scales. Kernel ridge's kernel is $\exp(-\gamma\|x - x'\|^2)$ on the standardised inputs, with $d$ the number of inputs. The noise scale is the smallest cross-validated RMSE among the regression models, in units of the training outputs' standard deviation.

| Dataset | Kernel ridge $\alpha, \gamma$ | Neighbours $k$ | CatBoost depth, trees | Logistic prior sd | Noise scale |
|---|---|---|---|---|---|
| concrete | $0.1, 1/d$ | 3 | 4, 300 | – | 0.309 |
| california | $0.1, 1/d$ | 8 | 2, 300 | – | 0.499 |
| yacht | $0.001, 1/d$ | 8 | 2, 300 | – | 0.042 |
| australian | – | 30 | 4, 300 | 0.3 | – |
| cleveland | – | 30 | 2, 100 | 0.3 | – |
| german | – | 20 | 4, 300 | 0.1 | – |
| vehicle | – | 30 | 2, 300 | 3 | – |
| wdbc | – | 20 | 6, 100 | 0.3 | – |

**Evaluation.** Classification uses the Dirac kernel. Regression uses a Gaussian kernel whose lengthscale is the dataset's noise scale, approximated by 256 random Fourier features [Rahimi and Recht, 2007] shared by every model. Every predictive has an exact embedding under these kernels. For each split, $\Psi$ is the ratio of the summed numerators to the summed denominators over the test inputs. A dataset's value is its mean over splits, with the standard deviation over splits divided by $\sqrt{10}$, and for TabPFN the leave-one-round-out standard error instead. The test uses a noise floor of $10^{-8}$ and online Newton step stakes, averages its wealth over the test inputs, and runs for every round. Accuracy is the log loss and error of the current predictive on the test set. For regression both are in units of the training outputs' standard deviation, with RMSE taken at the predictive mean. For classification the error is that of the most probable class, and probabilities are clamped below at $10^{-16}$, which matters only for nearest neighbours, whose predictive can give a label probability zero.

**Results.** Table 5 summarises each model's incoherence, accuracy and rejections, and Tables 6 and 7 give $\Psi$ and log loss by dataset. A few points supplement the main text.

- The exact models have $|\Psi|$ below $5 \times 10^{-4}$ on every dataset and are never rejected. Exact logistic regression on cleveland is $4.5$ standard errors below zero, because the estimate is skewed and the ten splits happened to fall below its mean. Rerunning one split 60 times gives an estimate centred on zero.
- The spread of $\Psi$ over splits is typically 1.8 to 15 times the within-split standard error (the median over datasets, for the models with ten splits), so TabPFN's standard errors understate its uncertainty across splits.
- Halving or doubling the kernel lengthscale swaps only models whose $\Psi$ is within 15% of each other. A lengthscale of one, between 2 and 24 times the noise scale, shrinks small values of $\Psi$ and swaps models up to 23% apart.
- The autoregressive joints leave a model's current predictive unchanged but change its $\Psi$, for example from $0.12$ to $0.19$ for kernel ridge on yacht.
- BART's incoherence is largely Monte Carlo error. On the first split of each regression dataset, rerunning the sampler on the training set alone with a fresh seed already displaces the predictive by 1.1 to 1.2 times as much as an update does, and a run's 100 draws are worth about two independent draws.
- Small but resolved incoherence often goes unrejected within the budget. The sparse GP's $\Psi$ exceeds twice its standard error on all 30 of its cells, but only 4 are rejected within 500 rounds, against a predicted median of about 1500 rounds. The test's predicted rounds for TabPFN range from 127 to 585, mostly beyond its budget of 128.

### C.3 Mean-field Bayesian linear regression

This experiment isolates the effect of incoherence on decisions. Every arm is evaluated with the same mean-field learner, and only the predictive that the acquisition objective consumes changes. Everything is Gaussian and computed in closed form, so the results carry no estimation noise beyond the choice of environments.

Table 5: Summary over datasets. $\Psi$ is the median over datasets of each model's mean over splits, with its range across datasets. Rank is the median rank of the model's test log loss among the models of its task (1 is best, ties averaged). Rejected is the share of the model's splits rejected within 500 rounds at level $0.05$. TabPFN's single split and 128 rounds make its rejections incomparable, so they are omitted.

| *Regression* Model | $\Psi$ | Range | Rank | Rej. |
|---|---|---|---|---|
| GP | 0.000 | 0.000–0.000 | 3.5 | 0.00 |
| Sparse GP | 0.007 | 0.004–0.018 | 9 | 0.13 |
| GP, refitted | 0.002 | 0.001–0.002 | 3.5 | 0.03 |
| BART | 0.71 | 0.63–0.86 | 7 | 1.00 |
| XBART | 0.38 | 0.27–0.79 | 5 | 1.00 |
| Kernel ridge | 0.013 | 0.008–0.119 | 8 | 0.37 |
| Nearest neighbours | 0.47 | 0.46–0.70 | 10 | 1.00 |
| CatBoost | 0.90 | 0.60–0.91 | 7 | 1.00 |
| TabPFN | 0.10 | 0.09–0.19 | 2 | – |
| TabPFN, preproc. | 0.12 | 0.08–0.22 | 1 | – |

| *Classification* Model | $\Psi$ | Range | Rank | Rej. |
|---|---|---|---|---|
| Logistic, exact | 0.000 | 0.000–0.000 | 4 | 0.00 |
| Laplace | 0.004 | 0.001–0.007 | 4 | 0.02 |
| Diagonal Laplace | 0.008 | 0.002–0.012 | 5 | 0.00 |
| Logistic regression | 0.007 | 0.001–0.016 | 3 | 0.10 |
| Nearest neighbours | 0.88 | 0.73–0.93 | 8 | 1.00 |
| CatBoost | 0.27 | 0.14–0.34 | 6 | 1.00 |
| TabPFN | 0.071 | 0.046–0.190 | 2 | – |
| TabPFN, preproc. | 0.13 | 0.054–0.42 | 2 | – |

Table 6: $\Psi$ by dataset, with one standard error over splits, or over rounds for TabPFN.

| Model | concrete | california | yacht |
|---|---|---|---|
| GP | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ |
| Sparse GP | $0.007 \pm 0.001$ | $0.004 \pm 0.000$ | $0.018 \pm 0.004$ |
| GP, refitted | $0.002 \pm 0.001$ | $0.001 \pm 0.000$ | $0.002 \pm 0.001$ |
| BART | $0.632 \pm 0.035$ | $0.710 \pm 0.051$ | $0.863 \pm 0.073$ |
| XBART | $0.378 \pm 0.034$ | $0.271 \pm 0.018$ | $0.787 \pm 0.075$ |
| Kernel ridge | $0.013 \pm 0.002$ | $0.008 \pm 0.001$ | $0.119 \pm 0.012$ |
| Nearest neighbours | $0.461 \pm 0.020$ | $0.470 \pm 0.017$ | $0.701 \pm 0.067$ |
| CatBoost | $0.900 \pm 0.048$ | $0.596 \pm 0.045$ | $0.912 \pm 0.239$ |
| TabPFN | $0.089 \pm 0.008$ | $0.102 \pm 0.008$ | $0.195 \pm 0.019$ |
| TabPFN, preproc. | $0.083 \pm 0.006$ | $0.118 \pm 0.009$ | $0.222 \pm 0.018$ |

| Model | australian | cleveland | german | vehicle | wdbc |
|---|---|---|---|---|---|
| Logistic, exact | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ | $0.000 \pm 0.000$ |
| Laplace | $0.003 \pm 0.001$ | $0.004 \pm 0.001$ | $0.001 \pm 0.000$ | $0.007 \pm 0.002$ | $0.004 \pm 0.002$ |
| Diagonal Laplace | $0.008 \pm 0.001$ | $0.006 \pm 0.001$ | $0.002 \pm 0.000$ | $0.009 \pm 0.001$ | $0.012 \pm 0.002$ |
| Logistic regression | $0.007 \pm 0.003$ | $0.003 \pm 0.001$ | $0.001 \pm 0.000$ | $0.012 \pm 0.008$ | $0.016 \pm 0.008$ |
| Nearest neighbours | $0.897 \pm 0.038$ | $0.849 \pm 0.036$ | $0.734 \pm 0.042$ | $0.883 \pm 0.026$ | $0.927 \pm 0.059$ |
| CatBoost | $0.145 \pm 0.011$ | $0.339 \pm 0.023$ | $0.211 \pm 0.012$ | $0.327 \pm 0.012$ | $0.273 \pm 0.043$ |
| TabPFN | $0.190 \pm 0.034$ | $0.046 \pm 0.010$ | $0.071 \pm 0.013$ | $0.142 \pm 0.020$ | $0.048 \pm 0.013$ |
| TabPFN, preproc. | $0.098 \pm 0.012$ | $0.054 \pm 0.008$ | $0.425 \pm 0.034$ | $0.248 \pm 0.022$ | $0.126 \pm 0.025$ |

**Environments.** The model is Bayesian linear regression on 100 random Fourier features [Rahimi and Recht, 2007] of a Gaussian kernel with lengthscale $0.1$ on a scalar input, with weights $w \sim \mathcal{N}(0, 3I)$ and observation noise variance $0.8$. The truth is drawn from this prior on the same features, so the setting is well specified. Each environment draws its features, the true weights, 25 initial inputs from $U[-1, 1]$, a pool of 1000 inputs and 100 target inputs from $U[-10, 10]$, noisy labels for the initial and pool inputs, and a random order of the pool. We use 100 environments, shared by every arm. There is no separate test set: the log loss we report is the expected log loss of a noisy observation at the 100 target inputs, in closed form, under the learner's predictive.

**Learner and acquisition.** The learner keeps the exact conjugate posterior. Its mean-field approximation, used for every reported loss, takes the exact posterior means and the inverse diagonal of the exact posterior precision as variances, which is the optimum of $\mathrm{KL}(q \,\|\, p)$ over factorised Gaussians. After each acquisition the approximation is recomputed from the new exact posterior rather than updated in a streaming fashion. Each arm acquires 100 points greedily, one at a time and without replacement, by maximising its score over the remaining pool. The entropy form of EPIG and EMI are both closed form here. The mean-field refit's variance does not depend on the outcome, and its mean is affine in it, so the expected future predictive $Q_Y$ is Gaussian. Both objectives are averaged

Table 7: Test log loss in nats per output, the mean over splits (TabPFN on its one split), with the lowest on each dataset in bold. Regression log loss is in units of the training outputs' standard deviation.

| Model | concrete | california | yacht |
|---|---|---|---|
| GP | 0.283 | 0.738 | −1.803 |
| Sparse GP | 0.457 | 0.889 | −1.749 |
| GP, refitted | 0.283 | 0.738 | −1.803 |
| BART | 0.151 | 0.773 | −1.540 |
| XBART | 0.089 | 0.738 | −0.128 |
| Kernel ridge | 0.412 | 0.762 | −0.231 |
| Nearest neighbours | 0.892 | 0.956 | 1.045 |
| CatBoost | 0.400 | 0.877 | −1.601 |
| TabPFN | −0.254 | 0.291 | **−3.021** |
| TabPFN, preproc. | **−0.308** | **0.237** | −2.959 |

| Model | australian | cleveland | german | vehicle | wdbc |
|---|---|---|---|---|---|
| Logistic, exact | 0.355 | 1.056 | 0.488 | 0.434 | 0.088 |
| Laplace | 0.352 | 1.043 | 0.490 | 0.449 | 0.094 |
| Diagonal Laplace | 0.356 | 1.043 | 0.493 | 0.443 | 0.107 |
| Logistic regression | 0.356 | 1.074 | **0.488** | 0.428 | 0.086 |
| Nearest neighbours | 0.409 | 1.421 | 0.598 | 0.840 | 0.225 |
| CatBoost | 0.323 | 1.190 | 0.490 | 0.500 | 0.097 |
| TabPFN | **0.211** | 0.946 | 0.497 | **0.340** | 0.049 |
| TabPFN, preproc. | 0.218 | **0.916** | 0.512 | 0.348 | **0.034** |

uniformly over the targets. *Exact Bayes* acquires with the exact posterior, under which EPIG and EMI coincide, and random acquisition takes the pool in its random order.

**The coherentised predictive.** Rung $\rho$ of the ladder scores candidates with EPIG under a predictive made coherent for the mean-field refit on a reference set $\mathcal{S}$, the first $\mathrm{round}(\rho N)$ points of the $N$ remaining pool points in the environment's fixed order. All predictives share the exact posterior mean and differ only in their weight covariance $\Omega$, giving $\mathcal{N}(\phi(x)^\top \mu, \phi(x)^\top \Omega \phi(x) + \sigma^2)$. Writing $\Sigma_+$ for the exact posterior covariance after observing $\mathcal{S}$, $\Omega_+$ for the mean-field covariance after observing $\mathcal{S}$, $B = \sigma^{-2}\Sigma_+ \Phi_{\mathcal{S}}^\top$ and $G = B\Phi_{\mathcal{S}}$, coherence for the refit on $\mathcal{S}$ requires the discrete Lyapunov equation

$$\Omega = \Omega_+ + \sigma^2 BB^\top + G\Omega G^\top,$$

whose solution is unique since the spectral radius of $G$ is below one. We solve it by summing $\sum_n G^n(\Omega_+ + \sigma^2 BB^\top)(G^\top)^n$ by repeated doubling. A candidate $c$ is scored by comparing the current predictive with the mean-field refit on $\mathcal{S} \cup \{c\}$, averaged over outcomes at $\mathcal{S}$ drawn from the coherentised predictive given the outcome at $c$. A candidate inside $\mathcal{S}$ is therefore counted both as observed and as part of the block. The reference set is recomputed at every step as the remaining pool shrinks, and its order is never redrawn. The shares are $\rho \in \{0, 1, 2.5, 5, 10, 100\}\%$, and at $\rho = 0$ the predictive is the mean-field one, so the rung is mean-field EPIG. The coherentised predictive is used only to score candidates.

**Measuring incoherence.** $\Psi$ is computed in closed form before any acquisition, under a Gaussian kernel of lengthscale $\sqrt{0.8}$, the noise standard deviation, and pooled over the 100 targets. For each rung, the update is the mean-field refit on the whole remaining pool, with outcomes drawn from the rung's predictive. This block is larger than the single-candidate update the acquisition faces, which Section 4.1 would otherwise call for, because each rung is defined by its coherence with the refit on a share of the pool, and that pool-scale coherence is what the ladder varies. Figure 3b plots the mean over environments of these values, and places EMI at the value for $\rho = 0$. This pool-scale $\Psi$ is about $10^{-3}$ for the mean-field model, whereas its $\Psi$ for an update on a single candidate, the scale of the other experiments, is $0.046$.

**Results.** Table 8 gives the values behind Figure 3. Every adjacent pair of rungs differs by at least four standard errors in loss, and coherentising over $1\%$, $2.5\%$, $5\%$ and $10\%$ of the pool recovers $32\%$, $51\%$, $80\%$ and $95\%$ of mean-field EPIG's excess loss. Mean-field EPIG and EMI choose different points at the first step in every environment, and their 100 acquired points overlap by 13 on average.

Table 8: Results of the Bayesian linear regression experiment over 100 environments. Excess loss is the mean-field learner's log loss less that after acquiring with the exact predictive, averaged over the 100 steps, with one standard error over environments. $\Psi$ is measured before acquisition for an update on the whole pool.

| Acquisition | Excess loss (nats) | $\Psi$ ($\times 10^{-3}$) |
|---|---|---|
| Mean-field, EPIG ($\rho = 0$) | $0.270 \pm 0.011$ | $1.05 \pm 0.05$ |
| Coherentised, $\rho = 1\%$ | $0.184 \pm 0.009$ | $0.80 \pm 0.04$ |
| Coherentised, $\rho = 2.5\%$ | $0.134 \pm 0.008$ | $0.55 \pm 0.03$ |
| Coherentised, $\rho = 5\%$ | $0.054 \pm 0.005$ | $0.32 \pm 0.02$ |
| Coherentised, $\rho = 10\%$ | $0.014 \pm 0.003$ | $0.13 \pm 0.01$ |
| Coherentised, $\rho = 100\%$ | $0.002 \pm 0.002$ | 0 |
| Mean-field, EMI | $0.008 \pm 0.002$ | – |
| Random | $0.156 \pm 0.008$ | – |

Table 9: Excess loss over acquiring with the exact predictive, as in Table 8, under changes to the baseline.

| Setting | Mean-field EPIG | Mean-field EMI | Random | $\rho = 5\%$ | $\rho = 100\%$ |
|---|---|---|---|---|---|
| Baseline | $0.270 \pm 0.011$ | $0.008 \pm 0.002$ | $0.156 \pm 0.008$ | $0.054 \pm 0.005$ | $0.002 \pm 0.002$ |
| Low noise | $1.966 \pm 0.070$ | $0.920 \pm 0.027$ | $0.929 \pm 0.037$ | $0.033 \pm 0.007$ | $0.009 \pm 0.004$ |
| High noise | $0.065 \pm 0.004$ | $0.003 \pm 0.001$ | $0.042 \pm 0.003$ | $0.017 \pm 0.003$ | $0.001 \pm 0.002$ |
| Distant targets | $0.385 \pm 0.019$ | $0.006 \pm 0.002$ | $0.270 \pm 0.013$ | $0.017 \pm 0.003$ | $-0.001 \pm 0.002$ |
| Misspecified | $0.537 \pm 0.018$ | $0.016 \pm 0.003$ | $0.454 \pm 0.016$ | $0.204 \pm 0.009$ | $0.008 \pm 0.002$ |

**Robustness.** We rerun the experiment with lower ($0.1$) and higher ($3$) noise variance, with targets drawn from $U[4, 8]$ away from most of the pool, and with a truth drawn from an independent set of random features, so that the model is misspecified (Table 9). Mean-field EPIG acquires worse data than random in every setting. With low noise the mean-field variances are badly wrong (an excess log loss of $1.31$ nats before acquisition, against $0.045$ in the baseline), and EMI then does no better than random, whereas the coherentised predictive still comes close to exact Bayes. EMI still consumes the mean-field model's variances, while coherentising changes them.

**The two-parameter example (Figure 1).** The model is $y = w_0 + w_1 x$ plus Gaussian noise of variance one, with prior $w \sim \mathcal{N}(0, 4I)$, fitted to five inputs drawn from $U[4, 6]$ with outputs $0.2x$ plus standard Gaussian noise. With the inputs this far from the origin, the exact posterior correlation between intercept and slope is $-0.97$, which the mean-field approximation cannot represent. The target is $x_* = 3.5$, and the candidate design ranges over $[-10, 30]$ in steps of $0.1$, of which the figure shows $[0, 18]$. Panel (a) shows the mean-field model's current predictive at the target and its expected future predictives after acquiring $A = 2$ or $B = 16$, each averaged over the outcome at the design. Under exact conditioning EPIG and EMI coincide and prefer $A$ ($0.081$ against $0.068$ nats). Under mean-field VI the entropy form of EPIG prefers $B$ ($0.037$ against $0.013$), whereas EMI prefers $A$ ($0.045$ against $0.014$).

### C.4 Neural testbed

This experiment compares approximate models on a single acquisition decision, in a setting small enough that every expectation over labels can be computed exactly. Its generative process follows the spirit of the Neural Testbed [Osband et al., 2022], but it is our own implementation, with three classes, clustered inputs and a batch acquisition task in place of the testbed's joint-prediction benchmark.

**Environments.** The truth is a random ReLU network with two inputs, two hidden layers of 32 units and three outputs, with Xavier-normal weights [Glorot and Bengio, 2010] and biases uniform on $\pm 1/\sqrt{\text{fan-in}}$. Its class probabilities are $\mathrm{softmax}((f(x) - o)/0.35)$, with offsets $o$ chosen so that each class has average probability $1/3$ over a grid on $[-6, 6]^2$. Inputs come from four Gaussian clusters centred at $(\pm 3, \pm 3)$. The 24 training inputs come from three of the clusters with standard deviation $0.5$, the pool of 512 from all four, and the 64 targets and 4096 test inputs from two, one of which

holds no training data, each with standard deviation $1.5$. An environment is kept if every class has mean true probability at least $0.1$ over its test inputs, which left 1000 environments from 1160 draws.

**The decision.** Each environment poses one decision among ten candidate batches of five pool inputs, the same for every model. Eight batches are designed from a diagonal Laplace approximation fitted to the training data, by rules that pick the most informative inputs overall, within the target clusters or within the training clusters, an informative input and its nearest neighbours, inputs spread over the pool or the target clusters, inputs near the decision boundary, or a mix of target and training inputs. The other batches are sampled with random cluster proportions, and a designed batch that duplicates an earlier one is replaced by a sampled one, giving about three sampled batches per environment.

**Models.** Every network model uses the generative architecture with a softmax likelihood, without the truth's temperature or offsets, and is fitted to the 24 training points. Exact Bayes is a bank of 8192 networks drawn from the generative prior, including its temperature and offsets, conditioned by Bayes' rule, with a new bank for each environment. The linearised Laplace approximations [Immer et al., 2021] (full, Kronecker-factored [Martens and Grosse, 2015, Ritter et al., 2018] and diagonal) share a MAP fit (Adam [Kingma and Ba, 2015], learning rate $0.05$, 1000 steps, Gaussian prior with four times the generative prior's scale) and use the generalised Gauss–Newton curvature [Schraudolph, 2002]. They update by conditioning the linearised model on the batch, through a Laplace approximation in the batch's logits found by 16 damped Newton steps. Bayes by backprop [Blundell et al., 2015] and IVON [Shen et al., 2024] fit factorised Gaussians under the generative prior with the KL term weighted by $0.1$ (Bayes by backprop with Adam at learning rate $0.003$ and 8 samples, IVON at learning rate $0.03$ with 4 samples, both for 2000 steps). They update by variational continual learning [Nguyen et al., 2018], with the current posterior as the prior and only the five new points in the objective, for 100 steps at learning rates $3 \times 10^{-4}$ and $10^{-3}$. The deep ensemble [Lakshminarayanan et al., 2017] has 8 members trained with Adam (learning rate $0.05$, 1000 steps, weight penalty 3), and MC dropout [Gal and Ghahramani, 2016] uses rate $0.2$ (learning rate $0.05$, 500 steps, weight penalty 1). Both update by continuing training on all 29 points, for 1000 and 50 steps. The Laplace and variational predictives are mixtures over 128 parameter draws, and MC dropout's over 128 masks, all fixed once and shared by the current and every updated predictive. The incTNP [Mortimer et al., 2026] has width 256, 8 layers and 8 heads, and is trained on tasks from this generative process to predict each training point from those before it and each target from the training points and part of a batch. Training uses Muon [Jordan et al., 2024] on hidden weight matrices and AdamW [Loshchilov and Hutter, 2019] elsewhere, at learning rate $5 \times 10^{-4}$ with cosine decay, for 90 000 steps of 128 tasks. It updates by appending the batch to its context. Hyperparameters were tuned on 64 separate environments by the KL divergence from the truth before and after an update, so tuning never sees incoherence or an objective. Several chosen values lie at the edge of their grids.

**Objectives, incoherence and cost.** For each candidate, all $3^5 = 243$ labellings $z$ are weighted by the model's joint $p(z)$, and the model is updated on each. The mixtures' joint draws one component and then labels the five inputs independently, and the incTNP's joint is autoregressive in the batch's order. Writing $p$ for the current predictive, $q_z$ for the predictive updated on $z$ and $\bar{q} = \sum_z p(z) q_z$, the objectives at each target are

$$\mathrm{EPIG}_H = H[p] - \textstyle\sum_z p(z)\, H[q_z], \qquad \mathrm{EPIG}_{\mathrm{KL}} = \textstyle\sum_z p(z)\, \mathrm{KL}(q_z \,\|\, p),$$
$$\mathrm{EMI} = \textstyle\sum_z p(z)\, \mathrm{KL}(q_z \,\|\, \bar{q}) = H[\bar{q}] - \textstyle\sum_z p(z)\, H[q_z],$$

averaged over the 64 targets. In the notation of Section 3, $p$ is $P_Y$, $p(z)$ is $P_Z^a(z)$, $q_z$ is $P_Y^{a,z}$ and $\bar{q}$ is $Q_Y^a$. $\Psi$ uses the Dirac kernel on the three classes, summed over the targets, for the update on each candidate, and an environment's $\Psi$ is the median over its candidates. The updates of Bayes by backprop, IVON and MC dropout are random, so for them we draw each update twice from independent streams and estimate $\|\Delta\|^2$ by the inner product of the two expected displacements, which is unbiased, and $V$ by their mean squared displacement less that estimate. These estimates can be negative, in 1, 6 and 36 of the 1000 environments respectively, and are kept. The distortion is the standard deviation over the ten candidates of $\mathrm{EPIG}_H - \mathrm{EMI}$. A candidate's loss is the KL divergence from the truth to the updated predictive at the 4096 test inputs, averaged over the batch's labellings under the truth, and the cost of a decision is the loss of $\mathrm{EPIG}_H$'s choice less that of EMI's. Accuracy before the update is the KL divergence from the truth to the fitted predictive, averaged over the test inputs.

Table 10: Results on the neural testbed over 1000 environments. $\Psi$ and $\|\Delta\|$ are medians over environments of each environment's median over candidates. Disagreement is the share of decisions where each form of EPIG chooses differently from EMI. Cost is the excess loss of entropy-form EPIG's choice over EMI's, over all decisions, and KL is the divergence from the truth before the update, both in nats with one standard error.

| | | | Disagreement | | | |
|---|---|---|---|---|---|---|
| Model | $\Psi$ | $\|\Delta\|$ | $\mathrm{EPIG}_H$ | $\mathrm{EPIG}_{\mathrm{KL}}$ | Cost | KL |
| Exact Bayes | 0 | 0 | 0.00 | 0.00 | 0 | $0.171 \pm 0.003$ |
| IVON | 0.0058 | 0.041 | 0.28 | 0.01 | $0.0084 \pm 0.0018$ | $0.366 \pm 0.007$ |
| Laplace, KFAC | 0.0067 | 0.059 | 0.40 | 0.01 | $0.0101 \pm 0.0020$ | $0.310 \pm 0.007$ |
| Bayes by backprop | 0.0102 | 0.094 | 0.80 | 0.10 | $0.0245 \pm 0.0027$ | $0.362 \pm 0.007$ |
| incTNP | 0.0103 | 0.097 | 0.26 | 0.02 | $0.0009 \pm 0.0005$ | $0.168 \pm 0.003$ |
| Ensemble | 0.0181 | 0.118 | 0.52 | 0.08 | $0.0195 \pm 0.0033$ | $0.406 \pm 0.009$ |
| Laplace, full | 0.0191 | 0.171 | 0.33 | 0.04 | $0.0071 \pm 0.0019$ | $0.270 \pm 0.004$ |
| Laplace, diagonal | 0.0474 | 0.318 | 0.38 | 0.08 | $0.0104 \pm 0.0020$ | $0.269 \pm 0.004$ |
| MC dropout | 0.117 | 0.425 | 0.54 | 0.20 | $0.0300 \pm 0.0069$ | $0.478 \pm 0.012$ |

**Figure 4.** The curves are Gaussian-kernel local linear regressions, with bandwidth $0.16$ in $\log_{10} \Psi$ in (a) and $0.06$ nats in (b). Markers are each model's mean over environments, in (b) over those where the objectives disagree. The bands are kernel-weighted 10th and 90th percentiles in (a) and $\pm 2$ approximate standard errors of the fit in (b). Environments with $\Psi \leq 0$ are left out of the fit in (a).

**Results.** Table 10 gives the results by model. Entropy-form EPIG has a higher loss than EMI on average for every approximate model, by more than two standard errors for all but the incTNP ($0.0009 \pm 0.0005$). A single disagreement goes either way, since EPIG's choice is the better one in 37–49% of them, and EMI's advantage lies in the size of the losses it avoids. Whether the objectives disagree depends on the margin between EMI's two best candidates relative to the distortion. Across quintiles of that ratio, the disagreement rate falls from $0.74$ to $0.06$. The cost of a disagreement has rank correlation $0.01$ with $\Psi$ and $0.21$ with the KL from the truth, pooled over models, and $0.24$ and $0.91$ across the models' means.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims are made in the abstract and reiterated in the introduction, with forward references to the corresponding sections of the main body.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations to the diagnostics introduced and the scope of the experiments are explained in Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper includes one proposition, establishing scale invariance of the proposed diagnostic, whose proof is given in Appendix B.2. The Dutch-book bound and the identities relating the forms of EPIG are derived in Appendix A, with their assumptions stated there. Other theoretical properties used in the paper are cited from prior work.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Full experimental details, including hyperparameters and data splits, are provided in the appendices, with sufficient information to reproduce the main experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available at `https://github.com/jreaso/predictive-coherence` under the MIT licence, with instructions for reproducing every experiment. All datasets used are publicly available and cited, and full experimental settings are given in Appendix C.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Key experimental settings are described in the main body, with full details including hyperparameters and data splits provided in the appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Figures report bands or error bars of two standard errors over environments or tasks, and the appendix tables report one standard error over splits or environments, as stated in each caption. Estimates of the incoherence ratio carry a leave-one-round-out standard error, defined in Appendix B.1.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are described in the appendix, including hardware used for experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: This work is methodological in nature and there is no direct societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release models or datasets that pose risks for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets are cited in Appendix C.2, where their licences are stated, and the software packages and pretrained weights used are cited and used under their published licences.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code is released at `https://github.com/jreaso/predictive-coherence` under the MIT licence, with a README describing how to run each experiment.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the research methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.