# A GROUND-TRUTH FRAMEWORK FOR UNCERTAINTY DISENTANGLEMENT WITH POSTERIOR RISK

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Reliable uncertainty estimates are critical in safety-sensitive applications. For such estimates to be useful in practice, it is crucial to understand the sources underlying a model's uncertainty, motivating the disentanglement of total uncertainty into epistemic and aleatoric uncertainty. Existing notions of uncertainty differ in the sources they capture and, consequently, in their definitions of aleatoric and epistemic uncertainty, with no universally accepted definition. We define uncertainty through sample-conditional pointwise posterior risk, which is the expected loss of a predictor under the distribution of plausible ground-truth functions given the observed sample. This definition unifies probabilistic and risk-based concepts of uncertainty. To assess state-of-the-art uncertainty disentanglement methods, we develop a framework that directly compares their estimates against ground-truth uncertainty defined primarily by posterior risk, alongside commonly used alternative uncertainty definitions. We find that Spectral-normalized Neural Gaussian Processes and Variational Latent Gaussian Processes most closely recover the ground-truth uncertainty, while most methods track posterior variance more closely than posterior risk, missing the predictors' bias. Beyond method rankings, we investigate how strongly estimated aleatoric and epistemic uncertainty are entangled and how sensitive uncertainty quality is to modeling choices, yielding practical guidance for uncertainty disentanglement. To support further method development and validation, we release 13 semi-synthetic UCI/OpenML datasets with known posteriors, enabling the computation of ground-truth uncertainty. Code and data will be made publicly available upon acceptance.

## 1 INTRODUCTION

AI systems are increasingly deployed in safety-critical domains where uncertainty is essential for safety and interpretability. In such settings, uncertainty estimates can guide referral to human experts, additional testing, data acquisition, and decisions about when a model's output can be trusted (Snoek et al., 2019; Kompa et al., 2021; Kendall & Gal, 2017). Crucially, *why* a model is uncertain matters as much as *how much*: uncertainty can be irreducible due to intrinsic task randomness (aleatoric uncertainty/AU), or reducible because the model is poorly optimized, lacks coverage, or is poorly specified (epistemic uncertainty/EU) (Hüllermeier & Waegeman, 2021). This distinction is useful, for example, in applied clinical machine learning, where EU is used to flag unreliable cases for review (Faghani et al., 2023; Huang et al., 2024). Proposed uncertainty concepts differ in the sources they cover and, consequently, in how they define and distinguish aleatoric and epistemic uncertainty (Kotelevskii et al., 2024; Lahlou et al., 2021; Jiménez et al., 2026). Yet a universally accepted definition remains lacking. Lacking this definition risks underestimating uncertainty because the employed definition itself may omit certain sources.

**Probabilistic and risk-based concepts of uncertainty.** Whilst probabilistic and risk-based concepts mostly agree on the definition of AU, they differ substantially on EU. Probabilistic approaches commonly quantify EU using the predictor-induced posterior over functions. We distinguish the *predictors' posterior* from a *reference posterior* induced by the data-generating process and the observed sample. The two coincide only under exact Bayesian inference using the data-generating prior and likelihood. Otherwise, the spread of the predictors' posterior does not faithfully represent the distribution over plausible ground-truth functions. Moreover, a misspecified prior and/or approximate



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

inference can introduce systematic bias into the model's predictions, in ways this notion of EU does not capture. Conversely, risk-based approaches directly address the bias of a predictor by fixing one ground truth and measuring EU as excess risk of the predictor relative to it. As a definition, this acknowledges model misspecification but misses uncertainty induced by the limited sample. As a result, EU can be arbitrarily small in data-sparse regions whenever the predictor happens to match the ground-truth function, despite substantial uncertainty from the learner. This is unintuitive from the learner's perspective, where limited data should generally imply greater uncertainty because the learner cannot know that it matched the ground-truth function.

**Sample-conditional posterior risk.** We propose a notion of ground-truth uncertainty that keeps the reference posterior separate from the estimator while allowing for estimator misspecification, optimization error, and EU due to a lack of data coverage to contribute to uncertainty. This notion is the *sample-conditional posterior risk*: the expected loss of the deployed predictor under the reference posterior over plausible ground-truth functions. Intuitively, it asks how reliable the deployed predictor is across the ground-truth functions that remain plausible given what the learner has observed. Importantly, posterior risk should not serve as a new definition for an estimator but rather as a reliable, actionable ground-truth notion of what uncertainty should be from the learner's perspective. Defining a ground-truth enables, to our knowledge, the first direct evaluation of these different sources of EU together with AU and, by extension, uncertainty disentanglement. Most existing benchmarks instead evaluate EU through proxy tasks such as out-of-distribution (OOD) detection or correlations between estimated uncertainty components (Mucsányi et al., 2024). Such evaluations are appropriate when performance on the proxy task itself is the objective: if EU is intended specifically for OOD detection, its usefulness for OOD detection is directly relevant. However, proxy tasks are less informative when the goal is to assess whether an estimator faithfully recovers EU as a general-purpose uncertainty quantity. In this case, success on a downstream task does not establish that the estimated EU is correct, and even perfect EU need not solve the proxy task (Li et al., 2025; Paplhám et al., 2026). Without a ground-truth EU target, the proxy can therefore become an implicit definition of what EU should be, rather than a test of whether EU has been estimated faithfully. Our benchmark makes this distinction testable by directly comparing uncertainty estimates with ground truth and examining how this direct evaluation relates to proxy-task performance.

**Semi-synthetic evaluation with reference posteriors.** To make sample-conditional posterior risk operational, we require a setting in which the reference posterior over ground-truth functions is explicit and samplable. We therefore instantiate the theory in a semi-synthetic tabular regression setting with real covariates and synthetic targets. To cover different priors, we instantiate the reference posterior through three distinct prior families over latent functions: Gaussian Processes (GPs), Heteroscedastic Bayesian Additive Trees (HBART), and Bayesian Neural Networks (BNNs). Importantly, the reference distribution is determined by the data-generating process and the observed sample, rather than by the estimator being evaluated. With this construction, the posterior is correctly specified with respect to the actual data-generating process. This enables direct computation of ground-truth uncertainty and therefore more fine-grained analysis than proxy evaluations. To summarize, we make the following contributions:

- **A sample-conditional posterior-risk formulation of uncertainty.** We define uncertainty as the risk of the deployed predictor under a method-independent reference posterior conditioned on the observed sample, and locate posterior-spread and fixed-truth excess-risk notions as boundary cases.
- **A suite of semi-synthetic tasks for uncertainty disentanglement.** We release a large-scale suite of semi-synthetic tasks for uncertainty disentanglement evaluation, spanning 13 datasets, three prior families, four aleatoric noise modes, and two data-coverage scenarios with the corresponding code. The methods' robustness to a change of ground-truth prior is also assessed. Each task includes a correctly specified reference posterior, enabling direct computation of ground-truth uncertainty and fine-grained method evaluation.
- **A direct evaluation of 17 state-of-the-art uncertainty disentanglement methods.** We benchmark 17 state-of-the-art uncertainty disentanglement methods directly against posterior risk, alongside the most prevalent uncertainty definitions. We also quantify excess uncertainty entanglement and report the most robust and reliable methods for uncertainty disentanglement identified by our framework.

Table 1: Notation used throughout the paper

| Symbol | Meaning |
|---|---|
| $X, Y;\ x, y$ | Random input $X$ and outcome $Y$, and their realizations $x, y$. Their respective spaces are $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$ for regression or $\mathcal{Y} = \{0, \dots, K\}$ for classification. |
| $f$ | Deterministic generative function $f \in \mathcal{F}$, where $\mathcal{F}$ denotes the space of possible generative functions and where $F$ denotes a random variable over $\mathcal{F}$. |
| $P(\cdot)$ | Joint distribution over $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. |
| $P(\cdot \mid x, f)$ | Pointwise first-order distribution around the ground-truth function $f$ with $Y \mid (X = x, F = f) \sim P(\cdot \mid x, f)$, with density $p(y \mid x, f)$ distributed w.r.t. the aleatoric noise model. |
| $\mathcal{S}$ | Random sample $\mathcal{S} := \{(x_i, y_i)\}_{i=1}^N$ with $(x_i, y_i) \overset{\text{i.i.d.}}{\sim} P(\cdot \mid x, f)$. |
| $a,\ \hat{a}_{\mathcal{S}}$ | Action $a \in \mathcal{A}$, where $\mathcal{A} \subseteq \mathcal{Y}$ for a point prediction and $\mathcal{A} = \mathcal{P}(\mathcal{Y})$ for a probabilistic prediction. The predictor $\hat{a}_{\mathcal{S}} : \mathcal{X} \to \mathcal{A}$ trained on $\mathcal{S}$ and returns the action $\hat{a}_{\mathcal{S}}(x)$ at input $x$. |
| $\ell$ | Loss $\ell\colon \mathcal{A} \times \mathcal{Y} \to \mathbb{R}$, covering pointwise losses and proper scoring rules. |
| $\Pi$ | Distribution over generative functions $f$. |

## 2 Uncertainty and pointwise risk

**The Bayesian world view.** We assume that observations arise from a real-world process described by a function $f$, which, together with observation noise, induces the conditional outcome distribution $P(\cdot \mid x, f)$. We further assume a function-generating prior $\Pi_0$ describing which functions are plausible in the domain and their probabilities before observing any data. Conditioning this prior on some sample $\mathcal{S}$ using the corresponding observation model yields the correctly specified *reference posterior* $\Pi_{\text{ref}}(\cdot \mid \mathcal{S})$ over plausible functions.

### 2.1 Uncertainty as sample-conditional posterior risk

Following the pointwise risk notion of uncertainty (Lahlou et al., 2021; Kotelevskii et al., 2022; 2024; Fishkov et al., 2025), we take pointwise risk as our starting point to later on combine the Bayesian world view with the risk-based notion of uncertainty.

**Definition 2.1** (First-order risk-based notion)**.** *For a fixed input $x$, the* pointwise risk *of $\hat{a}$ is*

$$R(\hat{a}; x) := \mathbb{E}_{Y \sim P(\cdot \mid x, f)}\big[\ell\big(\hat{a}(x), Y\big)\big] = \int_{\mathcal{Y}} \ell\big(\hat{a}(x), y\big)\, P(dy \mid x, f). \tag{1}$$

*If $\ell$ is a strictly proper scoring rule, the Bayes act is uniquely $P(\cdot \mid x, f)$, where $f$ is a single deterministic ground-truth function $f^\star$.*

Our contribution is to expand $P(\cdot \mid x, f)$ by a construction that matches a learner's perspective. We argue that the Bayesian world-view is the only view that matches the learners' perspective in terms of quantifying uncertainty, as the learner does not observe the full $P(\cdot \mid x, f^\star)$, only the finite sample $\mathcal{S}$ with limited coverage. Therefore, the distribution $P(\cdot \mid x, f)$ is not clearly identified by $\mathcal{S}$, inducing a second-order distribution over plausible outcome distributions around functions $f$. According to the Bayesian world view $f$ is distributed according to the reference posterior $\Pi_{\text{ref}}(\cdot \mid \mathcal{S})$. This induces an expanded notion of risk-based uncertainty, defined as the risk of a predictor after seeing $\mathcal{S}$ over the distribution of plausible real-world processes $f$ according to $\Pi_{\text{ref}}(\cdot \mid \mathcal{S})$.

**Definition 2.2** (Sample-conditional pointwise posterior risk)**.** *For a fixed input $x$ and observed sample $\mathcal{S}$, the sample-conditional pointwise posterior risk of a deployed predictor $\hat{a}_{\mathcal{S}}$ is*

$$\begin{aligned} R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) &:= \mathbb{E}_{F \sim \Pi_{\text{ref}}(\cdot \mid \mathcal{S})} \mathbb{E}_{Y \sim P(\cdot \mid x, F)}\big[\ell\big(\hat{a}_{\mathcal{S}}(x), Y\big)\big] \\ &= \int_{\mathcal{F}} \int_{\mathcal{Y}} \ell\big(\hat{a}_{\mathcal{S}}(x), y\big) P(dy \mid x, f)\, \Pi_{\text{ref}}(df \mid \mathcal{S}) \\ &= \mathbb{E}_{Y \sim P(\cdot \mid x, \mathcal{S})}\big[\ell\big(\hat{a}_{\mathcal{S}}(x), Y\big)\big] \end{aligned} \tag{2}$$

*The final line uses the posterior predictive mixture $P(\cdot \mid x, \mathcal{S}) = \int_{\mathcal{F}} P(\cdot \mid x, f)\, \Pi_{\text{ref}}(df \mid \mathcal{S})$.*

We follow Kendall & Gal (2017); Lahlou et al. (2021) and define total uncertainty as the sum of aleatoric and epistemic uncertainty. Additionally, we define sample-conditional pointwise posterior risk as the total uncertainty.

**Definition 2.3** (Disentanglement of the pointwise posterior risk)**.** *For each generative function $f$, define its conditional Bayes risk as $R_f^\star(x) := \inf_{a\in\mathcal{A}} \mathbb{E}_{Y\sim P(\cdot\,|x,f)}[\ell(a,Y)]$. Aleatoric uncertainty is the posterior expectation of this risk, and epistemic uncertainty is the posterior expected excess risk:*

$$\begin{aligned} \mathrm{AU}_{\mathcal{S}}(x) &:= \mathbb{E}_{F\sim\Pi_{\mathrm{ref}}(\cdot\,|\mathcal{S})}[R_F^\star(x)]\,, \\ \mathrm{EU}_{\mathcal{S}}(\hat{a}_{\mathcal{S}};x) &:= \mathbb{E}_{F\sim\Pi_{\mathrm{ref}}(\cdot\,|\mathcal{S})}\big[\mathbb{E}_{Y\sim P(\cdot\,|x,F)}[\ell(\hat{a}_{\mathcal{S}}(x),Y)] - R_F^\star(x)\big] \\ &= R_{\mathcal{S}}(\hat{a}_{\mathcal{S}};x) - \mathrm{AU}_{\mathcal{S}}(x). \end{aligned} \tag{3}$$

*Thus, AU is the expected posterior Bayes risk that would remain even if the generative function were known, while EU measures the deployed predictor's expected posterior excess risk over predicting the generative function. Their sum equals the pointwise posterior risk, which quantifies total uncertainty.*

Following Adlam et al. (2022), we use the Bregman divergence framework and the generalized bias–variance decomposition of Pfau (2025). The Bregman form provides a general AU–EU decomposition across all proper losses considered here. We can thus define the split once and obtain each loss-specific instantiation by substituting the corresponding divergence, recovering known quantities, for example, the Mutual Information as part of EU (Depeweg et al., 2018; Smith & Gal, 2018), as shown in Appendix A. The abstraction therefore provides generality over the most prevalent proper losses (Bülte et al., 2026).

**Proposition 2.4** (Bregman decomposition of posterior risk)**.** *Let $\mathcal{D}$ be a convex representation of the action space. With a slight abuse of notation, throughout the Bregman analysis we identify $Y$ with its canonical representation in $\mathcal{D}$: the observed value under squared loss, its one-hot encoding under multiclass log loss, and the function $t \mapsto \mathbf{1}\{Y \le t\}$ under CRPS. Let $\phi : \mathcal{D} \to \mathbb{R}$ be strictly convex and differentiable, with Bregman divergence $D_\phi(u\|v) := \phi(u) - \phi(v) - \langle \nabla\phi(v), u - v\rangle$. Assume that $\ell(a,Y) = D_\phi(Y\|a)$ and that all quantities below are well-defined. Write $\mathbb{E}_F$ for expectation under $\Pi_{\mathrm{ref}}(\cdot\mid\mathcal{S})$ and $\mathbb{E}_{Y|F}$ for expectation under $P(\cdot\mid x,F)$. Then $a_f(x) := \mathbb{E}_{Y\sim P(\cdot\,|x,f)}[Y]$ is the conditional Bayes act, and $\bar{a}_{\mathrm{ref}}(x) := \mathbb{E}_F[a_F(x)]$ is the posterior Bregman centroid. The pointwise posterior risk decomposes as*

$$\begin{aligned} R_{\mathcal{S}}(\hat{a}_{\mathcal{S}};x) &= \underbrace{\mathbb{E}_F\big[\mathbb{E}_{Y|F}[D_\phi(Y\,\|\,a_F(x))]\big]}_{\textbf{\textit{AU}}} + \underbrace{\mathbb{E}_F[D_\phi(a_F(x)\,\|\,\hat{a}_{\mathcal{S}}(x))]}_{\textbf{\textit{EU}}} \\ &= \underbrace{\mathbb{E}_F[R_F^\star(x)]}_{\textit{Bayes error}} + \underbrace{\mathbb{E}_F[D_\phi(a_F(x)\,\|\,\bar{a}_{\mathrm{ref}}(x))]}_{\textit{generalized variance}} + \underbrace{D_\phi(\bar{a}_{\mathrm{ref}}(x)\,\|\,\hat{a}_{\mathcal{S}}(x))}_{\textit{deployment gap}}. \end{aligned} \tag{4}$$

*We call the final term the* deployment gap *to distinguish it from frequentist bias averaged over repeated samples.*

The Bregman decomposition identifies AU as the posterior expected Bayes risk: the irreducible loss that remains even when the underlying generative function is known, independently of the deployed predictor. EU combines two terms: a predictor-independent generalized variance, which captures the spread of plausible ground-truth functions given $\mathcal{S}$, and a deployment gap, which measures the Bregman divergence between the corresponding centroid and the deployed prediction. Under squared loss, these terms reduce to the posterior variance and the squared deviation of the deployed prediction from the posterior mean.

### 2.2 The relationship of posterior risk to concurrent notions of uncertainty

**Bayesian Posterior-spread notion.** A common Bayesian view identifies EU with the spread of an estimator posterior $\Pi_{\mathrm{est}}(\cdot\mid\mathcal{S})$ over functions or parameters. In the Bregman setting of Equation (4), let $\bar{a}_{\mathrm{est}}(x)$ denote its corresponding posterior centroid. At this posterior-optimal action, total uncertainty decomposes as

$$\widehat{\mathrm{TU}}_{\mathrm{Bayes}}(x) = \underbrace{\mathbb{E}_{F\sim\Pi_{\mathrm{est}}}\mathbb{E}_{Y|F}\left[D_\phi(Y\,\|\,a_F(x))\right]}_{\widehat{\mathrm{AU}}_{\mathrm{Bayes}}(x)} + \underbrace{\mathbb{E}_{F\sim\Pi_{\mathrm{est}}}\left[D_\phi(a_F(x)\,\|\,\bar{a}_{\mathrm{est}}(x))\right]}_{\widehat{\mathrm{EU}}_{\mathrm{Bayes}}(x)}.$$

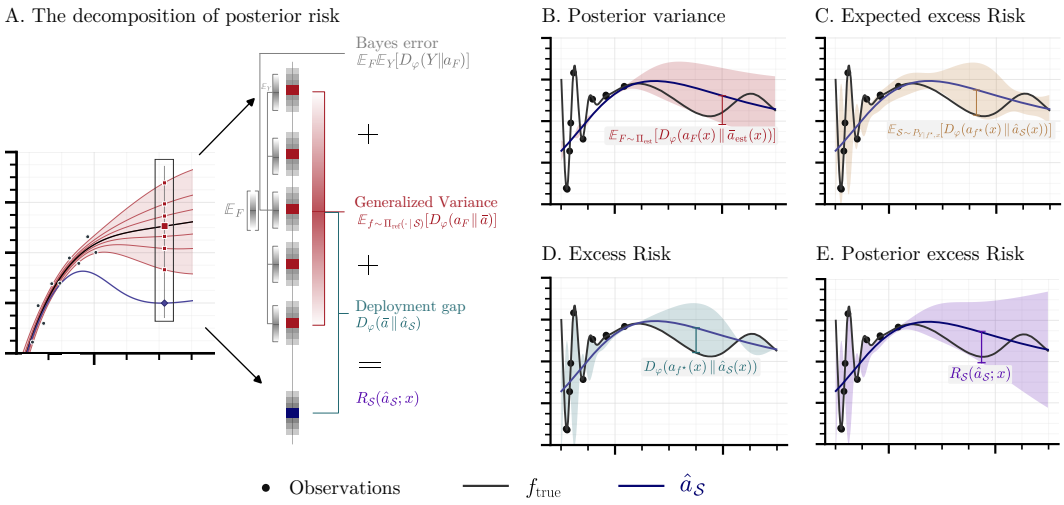



Figure 1: **Decomposition of posterior risk and pathologies of alternative uncertainty notions.** A: Visual decomposition of Equation (4). B–E: Different uncertainty notions retain only parts of this decomposition and can therefore behave pathologically. Posterior variance can remain small for a misspecified or underfit predictor (B), while excess-risk notions can assign low uncertainty when the predictor happens to be close to the realized ground truth despite limited data (C, D). Posterior excess risk retains both uncertainty over plausible functions and error of the deployed predictor (E).

Bayesian EU is therefore the generalized spread of the estimator posterior. It equals posterior-risk EU only when the estimator posterior matches the reference posterior, and its centroid is correct. Inference within the estimator's model does not guarantee this match. Differences in prior, likelihood, hypothesis class, or approximate inference can make the posteriors arbitrarily different in center and spread, and the estimator centroid need not be reference-optimal. In Figure 1 Panel B, insufficient model capacity accordingly produces an inaccurate, underfit prediction with low Bayesian uncertainty. Under correct specification, exact inference, and centroid deployment, the deployment gap vanishes by construction, leaving posterior spread as the only EU term.

Posterior-risk EU instead evaluates the deployed predictor under the reference posterior:

$$\mathrm{EU}_\mathcal{S}(\hat{a}_\mathcal{S};x) = \underbrace{\mathbb{E}_{F\sim\Pi_{\text{ref}}(\,\cdot\,|\mathcal{S})}\left[D_\phi(a_F(x)\,\|\,\bar{a}_{\text{ref}}(x))\right]}_{\text{generalized variance}} + \underbrace{D_\phi(\bar{a}_{\text{ref}}(x)\,\|\,\hat{a}_\mathcal{S}(x))}_{\text{deployment gap}}.$$

The estimator spread need not equal the first term, while the second measures the deviation of the deployed predictor from the reference-posterior centroid. Posterior-spread EU alone therefore does not account for misspecification and can assign low uncertainty to arbitrarily inaccurate predictions. It is not a reliable general ground-truth target, although it may approximate posterior-risk EU when posterior mismatch and the deployment gap are small. Posterior spread can nevertheless serve two roles. Under a probabilistic view, it estimates the reference-posterior spread that defines EU. Under a risk-based view, fixed-truth excess risk remains the target, while Bayesian machinery provides a practical estimator of that target (Kotelevskii et al., 2024; Fishkov et al., 2025). A Bayesian estimator therefore need not imply a probabilistic definition of EU.

**Risk-based notions.** First-order risk-based notions instead fix one ground truth $f^\star$ and define EU by deviation from its Bayes act, which is equivalent to setting $\Pi_{\text{ref}}(\,\cdot\mid\mathcal{S}) = \delta_{f^\star}$. With $a_{f^\star}(x) := \mathbb{E}_{Y\sim P(\,\cdot\,|x,f^\star)}[Y]$, Equation (4) gives

$$\mathrm{TU}_{f^\star}(\hat{a}_\mathcal{S};x) = \underbrace{\mathbb{E}_{Y\sim P(\,\cdot\,|x,f^\star)}\left[D_\phi(Y\,\|\,a_{f^\star}(x))\right]}_{\mathrm{AU}_{f^\star}(x)} + \underbrace{D_\phi(a_{f^\star}(x)\,\|\,\hat{a}_\mathcal{S}(x))}_{\mathrm{EU}_{f^\star}(\hat{a}_\mathcal{S};x)}.$$

Collapsing the reference posterior removes the generalized variance and measures the deployment gap relative to the fixed Bayes act $a_{f^\star}(x)$, rather than the reference-posterior centroid $\bar{a}_{\text{ref}}(x)$. Pointwise excess risk therefore measures realized estimator error, but not uncertainty from the incomplete

identification of $f^\star$ by $\mathcal{S}$. It can vanish in data-sparse regions whenever the deployed predictor happens to agree with the realized Bayes act, as shown in Figure 1 Panel C, although the sample cannot establish that this agreement has occurred. It can thus assign low EU precisely where uncertainty induced by limited data remains substantial.

Second-order risk-based notions instead introduce a distribution over repeated training samples $\mathcal{S}_i$, rather than plausible ground-truth functions, and measure expected error across repeated experiments (Jürgens et al., 2024; Jiménez et al., 2026). They provide a frequentist second-order analogue of posterior risk, but average over hypothetical samples instead of conditioning on the realized $\mathcal{S}$. They therefore do not represent uncertainty induced by the coverage of that realized sample.

## 3 Related Work

Beyond the notions in Section 2.2, concurrent work by Alamri et al. (2026) decomposes subjective risk under the predictive distribution, recovering several existing AU and EU measures under strictly proper losses. This is orthogonal to our goal of fixing a method-independent reference posterior to define ground-truth targets for estimator evaluation rather than unifying estimator-side measures.

On the evaluation side of uncertainty literature, most works utilize proxy tasks. These studies report that predictive entropy can conflate ambiguous in-distribution and OOD inputs (Kirsch et al., 2021), deterministic uncertainty methods can combine competitive OOD detection with poor calibration under distribution shift (Postels et al., 2022), and regression uncertainty estimators can become strongly overconfident under real-world shifts (Gustafsson et al., 2023). Other indirect diagnostics examine interactions between AU and EU estimates (Valdenegro-Toro & Saromo, 2022) or use their rank correlation as one disentanglement criterion (Mucsányi et al., 2024). These findings identify limitations of particular estimators or scores, but do not distinguish inaccurate EU estimation from a mismatch between EU and the proxy objective. Li et al. (2025) argue that a supervised classifier's predictive uncertainty, including conventional model-posterior EU, does not generally determine OOD membership. Paplhám et al. (2026) similarly show that Bayes-optimal strategies for OOD detection and active learning do not generally coincide with scores used to quantify EU. Proxy performance therefore depends on both estimator quality and objective alignment.

More direct evaluations use controlled simulations of physical systems with quantified measurement uncertainty (Caldeira & Nord, 2020) or compare approximate methods against HMC-derived EU on synthetic problems (Schweighofer et al., 2023). Concurrent work by Paplhám et al. (2026) also derives an equivalent decomposition of Bayesian expected risk into posterior-expected conditional Bayes risk and posterior-expected regret. Their benchmark evaluates the resulting AU and EU estimates through risk–regret–coverage trade-offs on datasets with densely repeated human annotations. Their empirical targets are estimated from annotator-derived conditional label distributions, whereas ours are analytic pointwise targets derived from a declared generative process and its sample-conditional reference posterior and further separate EU into posterior spread and the deployment gap.

## 4 Evaluation protocol

We define posterior risk as the ground-truth target for uncertainty estimates. To test how closely state-of-the-art methods align with this notion of uncertainty, and to identify the ground-truth notion they most closely approximate, we construct a set of 13 semi-synthetic datasets with reference posteriors that allow us to calculate ground-truth uncertainty measures. We instantiate each dataset using three different prior families (GP, HBART, and BNN; Appendix B.2), four aleatoric noise models (Appendix B.3), and two training-sample coverage scenarios (Appendix B.4). This results in 312 tasks, each repeated over 10 seeds. In each task, we rank-correlate the models' uncertainty estimates with the ground-truth uncertainty using Spearman's $\rho$ to directly assess the reliability of their uncertainty approximations.

**Semi-synthetic tasks and ground-truth targets.** Ground-truth uncertainty targets require both the aleatoric noise model and the reference posterior $\Pi_{\mathrm{ref}}(\cdot \mid \mathcal{S})$, neither of which is available for real data. We therefore retain the covariates of 13 real tabular datasets and generate the targets from a known prior over functions with known noise (Appendix B). Conditioning the same prior on the training sample yields a reference posterior that is correctly specified by construction with respect to

the data-generating process. As ground-truth AU, we use the specified noise model's variance. For EU, we define posterior excess risk, instantiated under squared loss, as the primary target. To assess alternative notions of EU, we also use reference-posterior variance and squared deviation from the ground-truth function as secondary targets.

**Metrics.** Because the absolute scale of risk depends on the evaluation loss, which is not specified to the methods, and many downstream tasks require only a reliable ordering of inputs, we use Spearman's $\rho$ as our primary metric. We average $\rho$ over all task instances of a dataset and report the min–max seed range across repeated experiments. To determine the stability of the method ranking across configurations, we report Kendall's $W$. For each axis (dataset, noise mode, prior family, and covariate scenario), we fix that axis while averaging over all others, thereby measuring its influence on the model rankings. Correlation with both ground-truth targets does not establish disentanglement, because an estimate of total uncertainty may correlate with both. We therefore measure excess entanglement,

$$\Delta_{\text{ent}} = \rho(\widehat{\text{EU}}, \widehat{\text{AU}}) - \rho(\text{EU}^\star, \text{AU}^\star), \tag{5}$$

where hats denote a method's estimates and stars denote the corresponding ground-truth values. Positive values indicate that the estimates depend on one another more strongly than the ground-truth quantities do. To test whether OOD detection serves as a good proxy for EU estimation, we additionally perform an OOD detection experiment using the White Wine UCI dataset as in-distribution data and the Red Wine dataset as OOD data. We then compare the ranking of models according to their AUROC in OOD detection, based on their EU estimates, with the ranking obtained from our direct Spearman-correlation-based benchmark.

**Methods.** We evaluate 17 methods from four families: Bayesian linear regression (BLR), GP-based models, tree ensembles and Neural Networks (NNs). Each method returns a prediction, an AU estimate, and an EU estimate, and all methods are trained using standard architecture and training settings for each task instance (Appendix C).

## 5 Evaluation

We begin by asking how closely the methods' uncertainty estimates track ground-truth EU and AU, which EU targets they capture most faithfully, and whether their relative performance is stable across tasks. Figure 2 summarizes performance across the 13 datasets using the mean Spearman rank correlation with the oracle EU and AU targets. Across methods, correlations range from $\rho = 0.16$ to $\rho = 0.69$ for EU and from $\rho = 0.42$ to $\rho = 0.86$ for AU. VLGP and SNGP-Het achieve the highest agreement with both targets, whereas NN-DEUP and NN-EDL perform the worst, which, for NN-EDL, is in line with the failure modes identified by Jürgens et al. (2024). For every method, EU estimates agree more closely with reference-posterior variance than with posterior excess risk and show weak agreement with pointwise excess risk. Despite these differences in absolute performance, the method ranking is stable across the benchmark axes of prior family, noise mode, and covariate scenario. Kendall's $W$, computed separately along each axis, is lowest across prior families ($W = 0.81$ for the EU and $0.93$ for the AU ranking, Table 7). Part of the variation across prior families is a matched-prior advantage: methods rank better when the ground truth is drawn from a prior like their own. The best and worst methods, however, are the same under every oracle (Figure 7 in Appendix D.1). Additionally rankings were stable across datasets (EU Kendall's $W = 0.78$, AU $W = 0.89$ see Figure 8, 9). Strong performance therefore does not require an exactly specified prior, and several methods remain effective across all generative settings considered.

**Are methods AU-EU estimates overly entangled?** Figure 3A shows the excess entanglement (Equation (5)) of each method and demonstrates that its magnitude depends on the definition of EU. When EU is defined as posterior excess risk, methods generally exhibit stronger EU–AU coupling than the corresponding ground truth. This discrepancy is smaller when EU is defined as posterior variance. Across the benchmark, ground-truth posterior variance and AU have a mean Spearman correlation of $\rho = 0.42$, which reflects the true degree of the ground-truth correlations. Because most evaluated methods aim to estimate posterior variance, this target provides an interesting comparison. Relative to posterior variance, excess coupling is largest for NGBoost-NIG, followed by NN-EDL

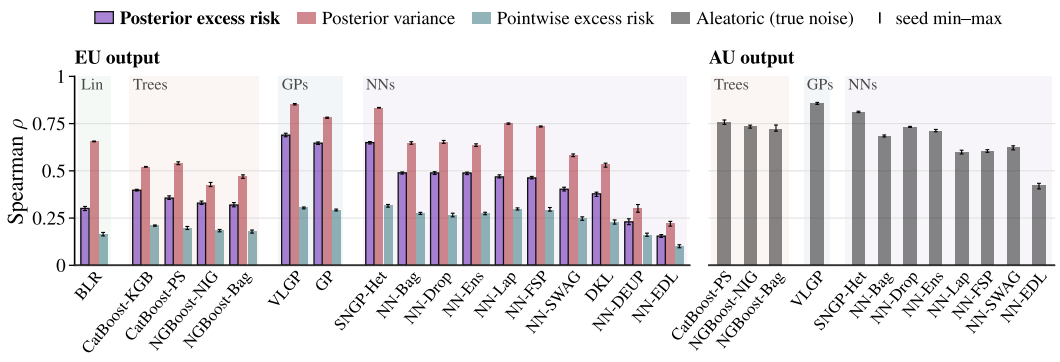



Figure 2: **Main benchmark results across the 13 datasets.** Bars show the mean Spearman rank correlation between estimated and oracle epistemic or aleatoric uncertainty. Error bars show the minimum-to-maximum range across repeated experiments with 10 different random seeds. SNGP-Het and VLGP achieve the highest rank agreement for both EU and AU. Across methods, EU estimates track posterior variance more closely than posterior risk and track pointwise excess risk poorly.

and NN-SWAG. Motivated by the improved training stability reported by Immer et al. (2023), we also evaluated natural-parameter heads but found that they increase excess entanglement for some methods. Appendix D.2 discusses possible reasons for the excess entanglement of EDL methods and natural-parameter heads. Accounting for the ground-truth EU–AU correlation suggests that excess entanglement is, at least w.r.t. posterior variance, less severe than previously reported (Mucsányi et al., 2024). Moreover, method rankings by excess entanglement differ from those by agreement with ground-truth EU and AU. This discrepancy demonstrates the need for direct targets because low excess entanglement alone does not imply faithful uncertainty estimates in our experiments.

**Is OOD detection a good proxy for EU?** We compare our benchmark ranking with a red- vs. white-wine OOD task (OpenML, 2022f;h). Methods are trained on white wine and use their EU to separate red from held-out white wine (Figure 3B). Whether the two rankings agree depends on the type of shift. Red wine as a whole lies far from white wine, so that Mahalanobis distance alone detects it (AUROC 0.98). For this distance-type shift, the OOD detection ranking closely matches our benchmark (Kendall's $W = 0.89$, where 1 means identical rankings). We then match red- and white wine evaluation points in distance to the training data. The red wines still differ in feature combinations and have higher GP-oracle uncertainty compared to white wines (Appendix D.3). But the OOD method rankings no longer agree ($W = 0.49$–$0.64$, where 0.5 means no association). The ensembles now detect red wine best, because the disagreement between their members grows on red wine. The EU of VLGP and SNGP-Het, the best methods in our benchmark, hardly changes, although they track the ground-truth EU within the white-wine data far more closely than the ensembles. This has two consequences. First, a method that detects one type of shift well is not guaranteed to detect others. Second, OOD detection cannot directly evaluate how well EU is estimated, because which methods it favours depends on the chosen OOD task.

**Does Spectral Normalization help NNs in estimating uncertainty?** SNGP-Het tracks the posterior excess risk far better than DKL ($\rho = 0.65$ vs. $0.38$). Both place a GP on learned features, but SNGP spectrally normalizes its backbone and uses a differently constructed and approximated GP output layer. Without such a constraint, feature extractors are known to collapse: inputs from unrelated regions are mapped onto the features of training points, so no output layer can recognize them as novel (Liu et al., 2020; van Amersfoort et al., 2022). Therefore the question arose if spectral normalization (SN) can help the other NN-based methods in estimating uncertainty. We retuned eight NN methods with SN and, as a control, strong weight decay (WD) to make sure that the SN effect was not attributable to stronger regularization (Table 13). SN raises local distance preservation for every method (pooled $+0.21$ vs. $+0.01$ for strong WD; Appendix D.4). For the four methods whose EU is read from a single network's features, SN improves the correlation with the posterior excess risk by $+0.10$ to $+0.11$ and with the posterior variance by up to $+0.26$. WD recovers at most half of this gain. For NN-Lap and NN-FSP, the gain is mediated by distance preservation and grows

**A.** EU–AU coupling vs. truth

| | $\rho(\widehat{\mathrm{EU}}, \widehat{\mathrm{AU}})$ | Excess entanglement: $\Delta_{\mathrm{ent}}$ post. exc. risk | post. variance | pointw. exc. risk |
|---|---|---|---|---|
| CatBoost-PS | 0.52 | +0.38 | +0.10 | +0.48 |
| NGBoost-NIG | 1.00 | +0.71 | +0.58 | +0.85 |
| NGBoost-Bag | 0.33 | +0.10 | –0.09 | +0.20 |
| VLGP | 0.51 | +0.21 | +0.09 | +0.38 |
| SNGP-Het | 0.46 | +0.17 | +0.04 | +0.32 |
| NN-Bag | 0.35 | +0.09 | –0.07 | +0.21 |
| NN-Bag (NP) | 0.55 | +0.29 | +0.13 | +0.40 |
| NN-Drop | 0.50 | +0.24 | +0.08 | +0.37 |
| NN-Drop (NP) | 0.48 | +0.24 | +0.06 | +0.35 |
| NN-Ens | 0.36 | +0.10 | –0.07 | +0.22 |
| NN-Ens (NP) | 0.53 | +0.26 | +0.11 | +0.37 |
| NN-Lap | 0.40 | +0.20 | –0.02 | +0.28 |
| NN-FSP | 0.42 | +0.22 | 0.00 | +0.30 |
| NN-SWAG | 0.57 | +0.34 | +0.15 | +0.44 |
| NN-SWAG (NP) | 0.54 | +0.24 | +0.11 | +0.34 |
| NN-EDL | 0.57 | +0.35 | +0.15 | +0.44 |

Linear Trees GP & DKL NNs

**B.** Wine OOD detection

| rank | all points | EU AUROC | rank | distance-matched | EU AUROC |
|---|---|---|---|---|---|
| 1 | GP | 0.979 | 1 | NN-Ens | 0.666 |
| 2 | VLGP | 0.979 | 2 | NN-Bag | 0.639 |
| 3 | BLR | 0.979 | 3 | CatBoost-KGB | 0.572 |
| 4 | SNGP-Het | 0.976 | 4 | NN-EDL | 0.569 |
| 5 | CatBoost-KGB | 0.969 | 5 | GP | 0.566 |
| 6 | NN-Ens | 0.968 | 6 | NGBoost-NIG | 0.558 |
| 7 | NN-Bag | 0.960 | 7 | SNGP-Het | 0.547 |
| 8 | NN-FSP | 0.950 | 8 | CatBoost-PS | 0.540 |
| 9 | NN-Drop | 0.937 | 9 | VLGP | 0.534 |
| 10 | NN-Lap | 0.933 | 10 | NN-Drop | 0.528 |
| 11 | NN-SWAG | 0.910 | 11 | NN-DEUP | 0.521 |
| 12 | DKL | 0.867 | 12 | NGBoost-Bag | 0.519 |
| 13 | NGBoost-Bag | 0.862 | 13 | NN-FSP | 0.497 |
| 14 | CatBoost-PS | 0.795 | 14 | NN-SWAG | 0.488 |
| 15 | NGBoost-NIG | 0.671 | 15 | BLR | 0.485 |
| 16 | NN-DEUP | 0.590 | 16 | DKL | 0.469 |
| 17 | NN-EDL | 0.482 | 17 | NN-Lap | 0.424 |
| | Mahalanobis | 0.982 | | Mahalanobis | 0.537 |
| | 10-NN distance | 0.981 | | 10-NN distance | 0.567 |
| | Classifier† | 0.997 | | Classifier† | 0.954 |

Figure 3: **A. EU–AU coupling.** $\rho(\widehat{\mathrm{EU}}, \widehat{\mathrm{AU}})$: Spearman correlation between a method's EU and AU outputs, averaged over all conditions. $\Delta_{\mathrm{ent}}$: excess over the ground-truth coupling for three EU definitions; $\Delta_{\mathrm{ent}} > 0$ means the outputs are more coupled than the truth. (NP): natural-parameter head. **B. Wine OOD detection.** AUROC of the EU output for red vs. held-out white wine (models trained on white wine), on all points and after matching Mahalanobis distance to the training data, where distance alone scores 0.5. †Supervised red-vs-white classifier, an upper bound on separability.

with depth, since deeper plain networks preserve distances worst (Table 14). For ensembles and MC dropout, whose EU arises from disagreement between models, SN leaves EU quality unchanged. SN preserves distances for uncertainty methods that read from a single representation, especially Laplace methods. Its gains grew with depth even though our backbones had at most four residual blocks, suggesting SN might be even more important for larger networks.

## 6 Limitations

Our ground-truth targets depend on the chosen prior family, so results may differ for priors outside our benchmark even though method rankings were broadly stable across those tested. Computational constraints limited the BNN priors to small networks, which may favor smoother functions than larger architectures would have produced as exact BNN inference at larger scales remains costly (Izmailov et al., 2021). Moreover, the Bregman decomposition covers most losses considered here but not every practically relevant objective, including general weighted losses. Corresponding decompositions are left for future work. Finally, our experiments cover only regression, chosen to permit controlled heteroscedastic noise and complement classification-focused prior work (Kirsch et al., 2021; Postels et al., 2022; Valdenegro-Toro & Saromo, 2022; Mucsányi et al., 2024). Extending the benchmark to classification is left for future work.

## 7 Conclusion

We introduced posterior risk as a ground-truth measure of uncertainty capturing a predictor's expected loss over plausible ground-truth functions. Its method-independent reference posterior retains uncertainty induced by limited data without assuming that an estimator's posterior is correct. Posterior risk also captures errors arising from misspecification or optimization through the deployment gap, while its Bayes-risk component represents irreducible outcome uncertainty. It therefore provides a common target for evaluating the main sources of AU and EU and enables more fine-grained analysis than estimator-internal or fixed-truth targets. On our semi-synthetic benchmark, SNGP-Het and VLGP showed the strongest overall agreement with posterior-risk targets. Most methods tracked posterior variance more closely than posterior risk. Their AU and EU estimates were also frequently more coupled than their ground-truth counterparts, although the extent depended on the EU definition. Spectral normalization improved uncertainty quality through distance preservation, particularly for Laplace approximation methods. Direct evaluation and proxy tasks also yielded different method rankings, supporting the use of ground-truth targets alongside downstream benchmarks.

## AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing methods and refining the conceptual framing of the paper as well as feedback on research methodology and experiments. We have not used generative AI tools for other tasks with required disclosure. Additionally, we used generative AI tools to create and edit code, identify relevant literature, and draft and edit parts of the manuscript. We have reviewed all AI-assisted work. We checked suggested references against the original sources and verified AI-assisted code, discussed generated feedback with colleagues and checked mathematical statements, and manuscript text for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REPRODUCIBILITY STATEMENT

We provide the code and scripts for constructing the benchmark datasets and reproducing the experimental pipeline and main results in the supplementary materials. Section 4 describes the benchmark construction and evaluation protocol, and Section 5 describes the experimental comparisons. Appendix A gives the theoretical derivations, Appendix B specifies the datasets and generation presets, and Appendix C documents the models, hyperparameter search spaces, tuning procedure, and compute requirements. Additional experimental results are reported in Appendix D.

## REFERENCES


Ben Adlam, Neha Gupta, Zelda Mariet, and Jamie Smith. Understanding the bias-variance tradeoff of Bregman divergences, February 2022. URL http://arxiv.org/abs/2202.04167. arXiv:2202.04167 [stat].

Raghad Alamri, Michele Caprio, and Gavin Brown. Subjective risk decomposition: A new view for uncertainty quantification, 2026. URL https://arxiv.org/abs/2607.15196.

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression, 2020. URL https://arxiv.org/abs/1910.02600.

Davide Ballabio, Matteo Cassotti, Viviana Consonni, and Roberto Todeschini. QSAR fish toxicity. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C5JG7B.

Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6(58):1705–1749, 2005. URL http://jmlr.org/papers/v6/banerjee05b.html.

Lukas Biewald. Experiment tracking with weights and biases. https://www.wandb.ai/, 2020. Software available from wandb.ai.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. doi: 10.1023/A:1018054314350.

Thomas Brooks, D. Pope, and Michael Marcolini. Airfoil Self-Noise. UCI Machine Learning Repository, 1989. DOI: https://doi.org/10.24432/C5VW2C.

Christopher Bülte, Yusuf Sale, Gitta Kutyniok, and Eyke Hüllermeier. Uncertainty quantification for regression: A unified framework based on kernel scores, 2026. URL https://arxiv.org/abs/2510.25599.

João Caldeira and Brian Nord. Deeply uncertain: comparing methods of uncertainty quantification in deep learning algorithms. *Machine Learning: Science and Technology*, 2, 12 2020. doi: 10.1088/2632-2153/aba6f3.

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1), March 2010. ISSN 1932-6157. doi: 10.1214/09-aoas285. URL http://dx.doi.org/10.1214/09-AOAS285.

Tristan Cinquin, Marvin Pförtner, Vincent Fortuin, Philipp Hennig, and Robert Bamler. Fsp-laplace: Function-space priors for the laplace approximation in bayesian deep learning, 2024. URL https://arxiv.org/abs/2407.13711.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009. doi: 10.1016/j.dss.2009.05.016.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux - effortless bayesian deep learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 20089–20103. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a7c9585703d275249f30a088cebba0ad-Paper.pdf.

DELVE. Computer Activity Database. http://www.cs.toronto.edu/~delve/copyright.html, n.d. Referenced by the OpenML Croissant metadata for cpu_activity as the dataset citation/source attribution.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning, 2018. URL https://arxiv.org/abs/1710.07283.

Timothy Dozat. Incorporating nesterov momentum into adam. 2015. URL https://api.semanticscholar.org/CorpusID:620137.

Tony Duan, Anand Avati, Daisy Yi Ding, Khanh K. Thai, Sanjay Basu, Andrew Ng, and Alejandro Schuler. Ngboost: natural gradient boosting for probabilistic prediction. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Shahriar Faghani, Mana Moassefi, Pouria Rouzrokh, Bardia Khosravi, Francis Baffour, Michael D Ringler, and Bradley J. Erickson. Quantifying uncertainty in deep learning of radiologic images. *Radiology*, 308(2), aug 2023. doi: 10.1148/radiol.222217. URL https://doi.org/10.1148/radiol.222217.

Alexander Fishkov, Kajetan Schweighofer, Mykyta Ielanskyi, Nikita Kotelevskii, Mohsen Guizani, and Maxim Panov. Uncertainty quantification for regression using proper scoring rules, 2025. URL https://arxiv.org/abs/2509.26610.

Vincent Fortuin, Mark Collier, Florian Wenzel, James Allingham, Jeremiah Liu, Dustin Tran, Balaji Lakshminarayanan, Jesse Berent, Rodolphe Jenatton, and Effrosyni Kokiopoulou. Deep classifiers with label noise modeling and distance awareness, 2022. URL https://arxiv.org/abs/2110.02609.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Zoubin Ghahramani. Kin Family of Datasets. https://www.cs.toronto.edu/~delve/data/kin/desc.html, 1996. DELVE dataset family; last updated 5 October 1996.

Tilmann Gneiting and Adrian Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378, 03 2007. doi: 10.1198/016214506000001437.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9d86d83f925f2149e9edb0ac3b49229c-Paper.pdf.

Fredrik K. Gustafsson, Martin Danelljan, and Thomas B. Schön. How reliable is your regression model's uncertainty under real-world distribution shifts?, 2023. URL https://arxiv.org/abs/2302.03679.

James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data, 2013. URL https://arxiv.org/abs/1309.6835.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo, 2011. URL https://arxiv.org/abs/1111.4246.

Ling Huang, Su Ruan, Yucheng Xing, and Mengling Feng. A review of uncertainty quantification in medical image analysis: Probabilistic and non-probabilistic methods. *Medical Image Analysis*, 97:103223, 2024. doi: 10.1016/j.media.2024.103223. URL https://doi.org/10.1016/j.media.2024.103223.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021. doi: 10.1007/S10994-021-05946-3. URL https://doi.org/10.1007/s10994-021-05946-3.

Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Mohammad Emtiyaz Khan. Scalable marginal likelihood estimation for model selection in deep learning, 2021. URL https://arxiv.org/abs/2104.04975.

Alexander Immer, Emanuele Palumbo, Alexander Marx, and Julia Vogt. Effective bayesian heteroscedastic regression with deep neural networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53996–54019. Curran Associates, Inc., 2023. doi: 10.52202/075280-2349. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a901d5540789a086ee0881a82211b63d-Paper-Conference.pdf.

Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like?, 2021. URL https://arxiv.org/abs/2104.14421.

Sebastián Jiménez, Mira Jürgens, and Willem Waegeman. Position: Epistemic uncertainty estimation methods are fundamentally incomplete, 2026. URL https://arxiv.org/abs/2505.23506.

Mira Jürgens, Nis Meinert, Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. Is epistemic uncertainty faithfully represented by evidential deep learning methods?, 2024. URL https://arxiv.org/abs/2402.09056.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?, 2017. URL https://arxiv.org/abs/1703.04977.

Andreas Kirsch, Jishnu Mukhoti, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. On pitfalls in ood detection: Predictive entropy considered harmful. In *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*, 2021. URL https://www.gatsby.ucl.ac.uk/~balaji/udl2021/accepted-papers/UDL2021-paper-092.pdf. Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning.

Benjamin Kompa, Jasper Snoek, and Andrew L. Beam. Second opinion needed: communicating uncertainty in medical machine learning. *NPJ Digital Medicine*, 4:4, January 2021. ISSN 2398-6352. doi: 10.1038/s41746-020-00367-3. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC7785732/.

Nikita Kotelevskii, Aleksandr Artemenkov, Kirill Fedyanin, Fedor Noskov, Alexander Fishkov, Artem Shelmanov, Artem Vazhentsev, Aleksandr Petiushko, and Maxim Panov. Nonparametric uncertainty quantification for single deterministic neural network, 2022. URL https://arxiv.org/abs/2202.03101.

Nikita Kotelevskii, Vladimir Kondratyev, Martin Takáč, Éric Moulines, and Maxim Panov. From risk to uncertainty: Generating predictive uncertainty measures via bayesian estimation, 2024. URL https://arxiv.org/abs/2402.10727.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Ion Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. Deup: Direct epistemic uncertainty prediction, 2021. URL https://arxiv.org/abs/2102.08501.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017. URL https://arxiv.org/abs/1612.01474.

Yucen Lily Li, Daohan Lu, Polina Kirichenko, Shikai Qiu, Tim G. J. Rudner, C. Bayan Bruss, and Andrew Gordon Wilson. Out-of-distribution detection methods answer the wrong questions, 2025. URL https://arxiv.org/abs/2507.01831.

Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness, 2020. URL https://arxiv.org/abs/2006.10108.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Miguel Lázaro-Gredilla and Michalis Titsias. Variational heteroscedastic gaussian process regression. pp. 841–848, 01 2011.

David J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3), May 1992. doi: 10.1162/neco.1992.4.3.415. URL https://doi.org/10.1162/neco.1992.4.3.415.

Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning, 2019. URL https://arxiv.org/abs/1902.02476.

Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. Uncertainty in gradient boosting via ensembles, 2021. URL https://arxiv.org/abs/2006.10562.

Nis Meinert, Jakob Gawlikowski, and Alexander Lavin. The unreasonable effectiveness of deep evidential regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8): 9134–9142, 2023. ISSN 2159-5399. doi: 10.1609/aaai.v37i8.26096. URL http://dx.doi.org/10.1609/aaai.v37i8.26096.

Bálint Mucsányi, Michael Kirchhof, and Seong Joon Oh. Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks, 2024. URL https://arxiv.org/abs/2402.19460.

Warwick Nash, Tracy Sellers, Simon Talbot, Andrew Cawthorn, and Wes Ford. Abalone. UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C55C7W.

Craig A. Olson. A comparison of parametric and semiparametric estimates of the effect of spousal health insurance coverage on weekly hours worked by wives. *Journal of Applied Econometrics*, 13 (5):543–565, 1998. doi: 10.1002/(SICI)1099-1255(1998090)13:5⟨543::AID-JAE507⟩3.0.CO;2-J.

OpenML. cpu_activity. https://www.openml.org/d/44978, 2022a. OpenML dataset, Data ID 44978, version 6. License: Public. Original source: https://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html.

OpenML. diamonds. https://www.openml.org/d/44979, 2022b. OpenML dataset, Data ID 44979, version 8. License: Public. Original source: https://ggplot2.tidyverse.org/reference/diamonds.html.

OpenML. health_insurance. https://www.openml.org/d/44993, 2022c. OpenML dataset, Data ID 44993, version 3. License: GPL (¿= 2). Original source: https://rdrr.io/cran/Ecdat/man/HI.html.

OpenML. kin8nm. https://www.openml.org/d/44980, 2022d. OpenML dataset, Data ID 44980, version 4. License: Public. Original source: https://www.cs.toronto.edu/~delve/data/kin/desc.html.

OpenML. physiochemical_protein. https://www.openml.org/d/44963, 2022e. OpenML dataset, Data ID 44963, version 7. License: CC BY 4.0. Original source: https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure.

OpenML. red_wine. https://www.openml.org/d/44972, 2022f. OpenML dataset, Data ID 44972, version 11. License: CC BY 4.0. Original source: https://archive.ics.uci.edu/ml/datasets/wine.

OpenML. sarcos. https://www.openml.org/d/44976, 2022g. OpenML dataset, Data ID 44976, version 9. License: Public. Original source: http://gaussianprocess.org/gpml/data/.

OpenML. white_wine. https://www.openml.org/d/44971, 2022h. OpenML dataset, Data ID 44971, version 7. License: CC BY 4.0. Original source: https://archive.ics.uci.edu/ml/datasets/wine.

OpenML. space_ga. https://www.openml.org/d/45402, 2023. OpenML dataset, Data ID 45402, version 5. License: Public. Source listed in OpenML metadata: https://www.openml.org/search?type=data&sort=runs&id=507&status=active.

R. Kelley Pace and Ronald Barry. Quick computation of spatial autoregressive estimators. *Geographical Analysis*, 29(3):232–247, 1997. doi: 10.1111/j.1538-4632.1997.tb00959.x.

Jakub Paplhám, Willem Waegeman, Eyke Hüllermeier, and Vojtěch Franc. Evaluating epistemic uncertainty: Beyond ood detection and active learning, 2026. URL https://arxiv.org/abs/2607.14817.

Giacomo Petrillo. Very fast bayesian additive regression trees on gpu, 2025. URL https://arxiv.org/abs/2410.23244.

David Pfau. A generalized bias-variance decomposition for bregman divergences, 2025. URL https://arxiv.org/abs/2511.08789.

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro, 2019. URL https://arxiv.org/abs/1912.11554.

Thomas Pinder and Daniel Dodd. Gpjax: A gaussian process framework in jax. *Journal of Open Source Software*, 7(75):4455, 2022. doi: 10.21105/joss.04455. URL https://doi.org/10.21105/joss.04455.

Janis Postels, Mattia Segù, Tao Sun, Luca Daniel Sieber, Luc Van Gool, Fisher Yu, and Federico Tombari. On the practicality of deterministic epistemic uncertainty. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17870–17909. PMLR, 2022. URL https://proceedings.mlr.press/v162/postels22a.html.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features, 2019. URL https://arxiv.org/abs/1706.09516.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Prashant Rana. Physicochemical Properties of Protein Tertiary Structure. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5QW3H.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

John Rice. Bandwidth choice for nonparametric regression. *The Annals of Statistics*, 12(4):1215–1230, 1984. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/2240998.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, Günter Klambauer, and Sepp Hochreiter. Quantification of uncertainty with adversarial models, 2023. URL https://arxiv.org/abs/2307.03217. NeurIPS 2023.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks, 2022. URL https://arxiv.org/abs/2203.09168.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection, 2018. URL https://arxiv.org/abs/1803.08533.

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua V. Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*

*(NeurIPS 2019)*, pp. 13969–13980, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8558cb408c1d76621371888657d2eb1d-Abstract.html.

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating bayesian inference algorithms with simulation-based calibration, 2020. URL https://arxiv.org/abs/1804.06788.

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

UCI. Seoul Bike Sharing Demand. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C5F62R.

Aleksei Ustimenko, Artem Beliakov, and Liudmila Prokhorenkova. Gradient boosting performs gaussian process inference, 2023. URL https://arxiv.org/abs/2206.05608.

Matias Valdenegro-Toro and Daniel Saromo. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1508–1516, 2022. doi: 10.1109/CVPRW56347.2022.00157. URL https://doi.org/10.1109/CVPRW56347.2022.00157.

Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty, 2022. URL https://arxiv.org/abs/2102.11409.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved r^ for assessing convergence of mcmc (with discussion). *Bayesian Analysis*, 16(2), 2021. ISSN 1936-0975. doi: 10.1214/20-ba1221. URL http://dx.doi.org/10.1214/20-BA1221.

Sethu Vijayakumar and Stefan Schaal. Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional spaces. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 288–293, 2000.

Tobias Weber, Bálint Mucsányi, Lenard Rommel, Thomas Christie, Lars Kasüschke, Marvin Pförtner, and Philipp Hennig. Laplax: Laplace approximations in jax. https://github.com/laplax-org/laplax, 2025.

Hadley Wickham. Prices of over 50,000 round cut diamonds. https://ggplot2.tidyverse.org/reference/diamonds.html, n.d. Dataset documentation for the diamonds dataset in ggplot2. Original source referenced by OpenML dataset 44979.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning, 2015. URL https://arxiv.org/abs/1511.02222.

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

## APPENDIX CONTENTS

**A Loss-specific instantiations of the posterior-risk decomposition** **18**

A.1 Common setup and proof of Equation 4 . . . . . 18

A.2 Squared loss . . . . . 19

A.3 Continuous ranked probability score (CRPS) . . . . . 20

A.4 Log score for continuous outcomes . . . . . 20

A.5 Multiclass log loss . . . . . 21

A.6 Brier score . . . . . 22

A.7 When is the aleatoric term independent of the reference posterior? . . . . . 23

**B Semi-synthetic regression tasks and reference posteriors** **24**

B.1 Covariates . . . . . 24

B.2 Priors over ground-truth functions . . . . . 24

B.3 Aleatoric noise . . . . . 25

B.4 Training sample and data coverage . . . . . 26

B.5 Generating one task instance . . . . . 27

B.6 Oracle targets . . . . . 28

B.7 Correctness and convergence checks . . . . . 29

B.8 Summary tables . . . . . 30

**C Models** **31**

C.1 Probabilistic Regression Baselines . . . . . 31

C.2 Deep Kernel Learning and Distance-Aware Models . . . . . 33

C.3 Tree-Based Models . . . . . 33

C.4 Bayesian Neural Network Models . . . . . 35

**D Supplementary Results** **39**

D.1 Stability of model rankings across tasks . . . . . 39

D.2 Disentanglement . . . . . 40

D.3 OOD Analysis . . . . . 44

D.4 Spectral Normalization . . . . . 46

# A Loss-specific instantiations of the posterior-risk decomposition

## A.1 Common setup and proof of Equation 4

**Conventions.** Throughout this appendix we fix an input $x$ and the observed sample $\mathcal{S}$. As in the main text, $\mathbb{E}_F$ denotes expectation over $F \sim \Pi_{\mathrm{ref}}(\cdot \mid \mathcal{S})$ and $\mathbb{E}_{Y|F}$ expectation over $Y \sim P(\cdot \mid x, F)$; for a fixed generative function $f$ we write $\mathbb{E}_{Y|f}$ for expectation over $Y \sim P(\cdot \mid x, f)$. The posterior predictive is $P(\cdot \mid x, \mathcal{S}) = \mathbb{E}_F\big[P(\cdot \mid x, F)\big]$ (Equation (2)). Within each subsection the deployed action $\hat{a}_{\mathcal{S}}(x)$ is abbreviated by a loss-specific symbol, stated in its setting card. As in Equation (4), $\phi$ is strictly convex; for the function-valued cases (CRPS and densities) this holds after identifying functions that agree almost everywhere. We assume that all displayed expectations and integrals are finite and that expectations may be exchanged (Fubini). Wherever the gradient of $\phi$ is used, it is assumed to exist at the second argument of $D_\phi$; boundary cases of the KL divergence and the CRPS are read through the explicit divergence formulas in the setting cards.

**Point and distributional scoring.** Under *point (value) scoring*, the action is a point $a \in \mathcal{A} \subseteq \mathcal{Y}$ and the conditional Bayes act is a functional of the conditional law; for the point loss considered below it is the conditional mean. Under *distributional scoring*, the action is a predictive distribution $a \in \mathcal{A} = \mathcal{P}(\mathcal{Y})$ scored by a strictly proper scoring rule, and the conditional Bayes act is the conditional law itself. Equation (4) covers both cases whenever outcome and action admit representatives in a common convex set $\mathcal{D}$ with $\ell(a, Y) = D_\phi(Y\|a)$. The continuous log score does not; it is covered by Lemma A.1 below.

**Compensation identity.** For any $\mathcal{D}$-valued random variable $U$ with mean $\bar{u} := \mathbb{E}[U] \in \mathcal{D}$ and any $a \in \mathcal{D}$,

$$\mathbb{E}\big[D_\phi(U\|a)\big] = \mathbb{E}\big[D_\phi(U\|\bar{u})\big] + D_\phi(\bar{u}\|a). \tag{6}$$

In the difference of the two expectations the terms $\phi(U)$ cancel and $\mathbb{E}[U - \bar{u}] = 0$ removes the linear term at $\bar{u}$, leaving $\phi(\bar{u}) - \phi(a) - \langle \nabla\phi(a), \bar{u} - a\rangle = D_\phi(\bar{u}\|a)$. Because $\phi$ is strictly convex, $D_\phi(\bar{u}\|a) \geq 0$ with equality if and only if $a = \bar{u}$, so the mean is the unique minimizer of $a \mapsto \mathbb{E}[D_\phi(U\|a)]$ (Banerjee et al., 2005). For a merely convex $\phi$, the mean would still be a minimizer, but not necessarily the only one.

**Proof of Equation (4).** Applying equation 6 to $U = Y \sim P(\cdot \mid x, f)$ for fixed $f$ shows that $a_f(x) = \mathbb{E}_{Y|f}[Y]$ is the conditional Bayes act, that $\mathbb{E}_{Y|f}\big[D_\phi(Y\|a_f(x))\big] = R^\star_f(x)$, and that

$$\mathbb{E}_{Y|f}\big[D_\phi\big(Y\|\hat{a}_{\mathcal{S}}(x)\big)\big] = R^\star_f(x) + D_\phi\big(a_f(x)\|\hat{a}_{\mathcal{S}}(x)\big).$$

Taking $\mathbb{E}_F$ gives the first line of Equation (4), $R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \mathrm{AU}_{\mathcal{S}}(x) + \mathbb{E}_F\big[D_\phi\big(a_F(x)\|\hat{a}_{\mathcal{S}}(x)\big)\big]$. Applying equation 6 once more, now to $U = a_F(x)$ with mean $\bar{a}_{\mathrm{ref}}(x)$, splits the epistemic term into the generalized variance and the deployment gap. □

**The centroid is the Bayes act of the posterior predictive.** Since

$$\bar{a}_{\mathrm{ref}}(x) = \mathbb{E}_F\,\mathbb{E}_{Y|F}[Y] = \mathbb{E}_{Y\sim P(\cdot|x,\mathcal{S})}[Y],$$

the posterior Bregman centroid is the Bayes act of the posterior predictive. For every distributional score below it is therefore the posterior predictive itself; for the point loss it is the corresponding functional of it.

**Outcomes without a representative in $\mathcal{D}$.** For the log score with continuous outcomes, the natural representative of an observation is the point mass $\delta_Y$, which has no Lebesgue density, so $\ell(a, Y) = D_\phi(Y\|a)$ is unavailable pointwise. The same three-term decomposition nonetheless holds for every proper scoring rule.

**Lemma A.1** (Scoring-rule form of the decomposition)**.** *Let $\ell$ be a proper scoring rule on a convex class $\mathcal{P}$ of distributions that contains $P_f := P(\cdot \mid x, f)$ for $\Pi_{\mathrm{ref}}$-almost every $f$, the posterior predictive $\bar{P}_{\mathrm{ref}} := P(\cdot \mid x, \mathcal{S})$, and the deployed action $\hat{Q} := \hat{a}_{\mathcal{S}}(x)$, and let the action space be $\mathcal{A} = \mathcal{P}$. Define the generalized entropy and the score divergence*

$$H_\ell(P) := \mathbb{E}_{Y\sim P}\big[\ell(P, Y)\big], \qquad d_\ell(P\|Q) := \mathbb{E}_{Y\sim P}\big[\ell(Q, Y)\big] - H_\ell(P) \;\geq\; 0.$$

*Then* $R_f^\star(x) = H_\ell(P_f)$*, and*

$$R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \underbrace{\mathbb{E}_F\big[H_\ell(P_F)\big]}_{\mathrm{AU}_{\mathcal{S}}(x)} + \underbrace{\mathbb{E}_F\big[d_\ell(P_F \| \bar{P}_{\mathrm{ref}})\big]}_{\textit{generalized variance}} + \underbrace{d_\ell(\bar{P}_{\mathrm{ref}} \| \hat{Q})}_{\textit{deployment gap}}, \tag{7}$$

*where* $\bar{P}_{\mathrm{ref}}$ *minimizes* $Q \mapsto \mathbb{E}_F\big[d_\ell(P_F \| Q)\big]$.

*Proof.* Propriety gives $R_f^\star(x) = \inf_Q \mathbb{E}_{Y\sim P_f}[\ell(Q, Y)] = H_\ell(P_f)$. Because $P \mapsto \mathbb{E}_{Y\sim P}[\ell(Q, Y)]$ is linear,

$$R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \mathbb{E}_{Y\sim\bar{P}_{\mathrm{ref}}}[\ell(\hat{Q}, Y)] = H_\ell(\bar{P}_{\mathrm{ref}}) + d_\ell(\bar{P}_{\mathrm{ref}} \| \hat{Q}),$$

and, by the same linearity,

$$H_\ell(\bar{P}_{\mathrm{ref}}) = \mathbb{E}_F\, \mathbb{E}_{Y\sim P_F}[\ell(\bar{P}_{\mathrm{ref}}, Y)] = \mathbb{E}_F\big[H_\ell(P_F) + d_\ell(P_F \| \bar{P}_{\mathrm{ref}})\big].$$

Finally, $\mathbb{E}_F[d_\ell(P_F \| Q)] = \mathbb{E}_{Y\sim\bar{P}_{\mathrm{ref}}}[\ell(Q, Y)] - \mathbb{E}_F[H_\ell(P_F)]$ is minimized at $Q = \bar{P}_{\mathrm{ref}}$ by propriety. □

The score divergence $d_\ell$ is the generalized Bregman divergence generated by the convex function $-H_\ell$ and the subgradient selected by the scoring rule (Gneiting & Raftery, 2007); when $-H_\ell$ is differentiable, it reduces to an ordinary Bregman divergence. When observations and predictive distributions admit representatives in a common Bregman domain $\mathcal{D}$, as for the CRPS, multiclass log loss and Brier score, identifying each distribution with its representative gives $d_\ell = D_\phi$, and Lemma A.1 coincides term by term with Equation (4). For the continuous log score, $d_\ell = \mathrm{KL}$ on densities, but the pointwise loss has no such representation because $\delta_Y$ has no Lebesgue density; Lemma A.1 then writes AU as a generalized entropy rather than as an expected divergence.

**Structure of the following subsections.** Each subsection has the same four parts: a setting card that fixes every object entering Equation (4), followed by the *derivation*, the resulting *decomposition*, and the form of the *aleatoric term*. Table 2 collects the results.

### A.2 Squared loss

| | |
|---|---|
| **Scoring type** | Point (value) scoring. |
| **Outcome space** $\mathcal{Y}$ | $\mathcal{Y} = \mathbb{R}$. |
| **Action space** $\mathcal{A}$, deployed action | $\mathcal{A} = \mathbb{R}$; deployed point prediction $\hat{a} := \hat{a}_{\mathcal{S}}(x)$. |
| **Loss** $\ell(a, y)$ | $\ell(a, y) = (y - a)^2$. |
| **Representation of** $Y$ **in** $\mathcal{D}$ | $\mathcal{D} = \mathbb{R}$; the outcome $Y$ itself. |
| **Generator and divergence** | $\phi(u) = u^2$; $D_\phi(u \| v) = (u - v)^2$. |
| **Conditional Bayes act** $a_f(x)$ | Conditional mean $\mu_f(x) := \mathbb{E}_{Y\|f}[Y]$. |
| **Posterior centroid** $\bar{a}_{\mathrm{ref}}(x)$ | Posterior mean $m_{\mathrm{ref}}(x) := \mathbb{E}_F[\mu_F(x)] = \mathbb{E}[Y \mid x, \mathcal{S}]$. |

**Derivation.** Write $\sigma_\varepsilon^2(x, f) := \mathrm{Var}_{Y|f}(Y)$ for the conditional noise variance under $f$, and $\sigma_{\mathrm{ref}}^2(x) := \mathrm{Var}_F\big[\mu_F(x)\big]$ for the posterior variance of the conditional mean. For fixed $f$,

$$\mathbb{E}_{Y|f}\big[(Y - \hat{a})^2\big] = \sigma_\varepsilon^2(x, f) + \big(\mu_f(x) - \hat{a}\big)^2,$$

and averaging over $F$,

$$\mathbb{E}_F\big[(\mu_F(x) - \hat{a})^2\big] = \sigma_{\mathrm{ref}}^2(x) + \big(m_{\mathrm{ref}}(x) - \hat{a}\big)^2.$$

**Decomposition.**

$$R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \underbrace{\mathbb{E}_F\big[\sigma_\varepsilon^2(x, F)\big]}_{\mathrm{AU}_{\mathcal{S}}(x)} + \underbrace{\underbrace{\sigma_{\mathrm{ref}}^2(x)}_{\text{generalized variance}} + \underbrace{\big(\hat{a}_{\mathcal{S}}(x) - m_{\mathrm{ref}}(x)\big)^2}_{\text{deployment gap}}}_{\mathrm{EU}_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x)}. \tag{8}$$

The first two terms sum to the posterior predictive variance $\mathrm{Var}(Y \mid x, \mathcal{S})$ by the law of total variance; the deployment gap vanishes if and only if $\hat{a}_{\mathcal{S}}(x) = m_{\mathrm{ref}}(x)$.

**Aleatoric term.** If the conditional variance does not depend on $f$, $\sigma_\varepsilon^2(x, f) = \sigma_\varepsilon^2(x)$ (for instance under additive noise $Y = f(x) + \varepsilon$ with an $f$-independent noise law, where $\mu_f(x) = f(x)$), then $\mathrm{AU}_{\mathcal{S}}(x) = \sigma_\varepsilon^2(x)$, independently of $\mathcal{S}$. If instead $\sigma_\varepsilon^2(x, f)$ varies over the support of the reference posterior, AU is the posterior mean of the noise variance and can depend on $\mathcal{S}$.

### A.3 Continuous ranked probability score (CRPS)

| | |
|---|---|
| **Scoring type** | Distributional scoring; strictly proper on distributions with finite first moment. |
| **Outcome space** $\mathcal{Y}$ | $\mathcal{Y} = \mathbb{R}$. |
| **Action space** $\mathcal{A}$, deployed action | $\mathcal{A}$: CDFs on $\mathbb{R}$ with finite first moment; deployed predictive CDF $\hat{G} := \hat{a}_{\mathcal{S}}(x)$. |
| **Loss** $\ell(a, y)$ | $\ell(G, y) = \mathrm{CRPS}(G, y) = \int_{\mathbb{R}} \big(G(t) - \mathbf{1}\{y \le t\}\big)^2 \, dt$. |
| **Representation of** $Y$ **in** $\mathcal{D}$ | $\mathcal{D}$: functions $G : \mathbb{R} \to [0, 1]$ with $\int_{\mathbb{R}} \lvert G(t) - \mathbf{1}\{t \ge 0\}\rvert \, dt < \infty$; the step function $t \mapsto \mathbf{1}\{Y \le t\}$. |
| **Generator and divergence** | $\phi(G) = \int_{\mathbb{R}} \big(G(t)^2 - \mathbf{1}\{t \ge 0\}\big) \, dt$; $D_\phi(u \Vert v) = \int_{\mathbb{R}} \big(u(t) - v(t)\big)^2 dt$. The subtracted term only makes $\phi$ finite and cancels in $D_\phi$. |
| **Conditional Bayes act** $a_f(x)$ | Conditional CDF $G_f(t) := P(Y \le t \mid x, f) = \mathbb{E}_{Y \mid f}\big[\mathbf{1}\{Y \le t\}\big]$. |
| **Posterior centroid** $\bar{a}_{\mathrm{ref}}(x)$ | Posterior predictive CDF $\bar{G}_{\mathrm{ref}}(t) := \mathbb{E}_F[G_F(t)] = P(Y \le t \mid x, \mathcal{S})$. |

**Derivation.** For fixed $f$ and each $t$,

$$\mathbb{E}_{Y \mid f}\big[\mathbf{1}\{Y \le t\}\big] = G_f(t), \qquad \mathbb{E}_{Y \mid f}\big[(\mathbf{1}\{Y \le t\} - G_f(t))^2\big] = G_f(t)\big(1 - G_f(t)\big),$$

so the cross term vanishes pointwise in $t$ and

$$\mathbb{E}_{Y \mid f}\big[\mathrm{CRPS}(\hat{G}, Y)\big] = \int_{\mathbb{R}} G_f(t)\big(1 - G_f(t)\big) \, dt + \int_{\mathbb{R}} \big(G_f(t) - \hat{G}(t)\big)^2 dt.$$

Averaging over $F$ and expanding around $\bar{G}_{\mathrm{ref}}$, the cross term again vanishes pointwise in $t$ because $\mathbb{E}_F[G_F(t)] = \bar{G}_{\mathrm{ref}}(t)$. Both steps are instances of equation 6 for the squared $L^2$ distance, which is well defined on $\mathcal{D}$ because differences of its elements lie in $L^2(\mathbb{R})$, even though CDFs themselves do not.

**Decomposition.**

$$R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \underbrace{\mathbb{E}_F \int_{\mathbb{R}} G_F(t)\big(1 - G_F(t)\big) \, dt}_{\mathrm{AU}_{\mathcal{S}}(x)} + \underbrace{\mathbb{E}_F \int_{\mathbb{R}} \big(G_F(t) - \bar{G}_{\mathrm{ref}}(t)\big)^2 dt}_{\text{generalized variance}} + \underbrace{\int_{\mathbb{R}} \big(\bar{G}_{\mathrm{ref}}(t) - \hat{G}(t)\big)^2 dt}_{\text{deployment gap}}. \tag{9}$$

**Aleatoric term.** The conditional Bayes risk is $\int_{\mathbb{R}} G_f(1 - G_f) \, dt = \frac{1}{2}\,\mathbb{E}\lvert Y - Y'\rvert$ for $Y, Y' \overset{\text{iid}}{\sim} P(\cdot \mid x, f)$, which is invariant under shifts of $P(\cdot \mid x, f)$. Under additive noise $Y = \mu_f(x) + \varepsilon$ whose law given $x$ does not depend on $f$, AU therefore equals $\frac{1}{2}\,\mathbb{E}\lvert\varepsilon - \varepsilon'\rvert$, independently of $\mathcal{S}$; for Gaussian noise, $\mathrm{AU}_{\mathcal{S}}(x) = \sigma_\varepsilon(x)/\sqrt{\pi}$.

**Computation.** For CDFs $G_1, G_2$ with independent draws $X_i, X_i' \sim G_i$, $\int_{\mathbb{R}}(G_1 - G_2)^2 dt = \mathbb{E}\lvert X_1 - X_2\rvert - \frac{1}{2}\mathbb{E}\lvert X_1 - X_1'\rvert - \frac{1}{2}\mathbb{E}\lvert X_2 - X_2'\rvert$. The three terms can therefore be estimated by Monte Carlo from posterior draws $F_m \sim \Pi_{\mathrm{ref}}(\cdot \mid \mathcal{S})$ with conditional samples $Y_m, Y_m' \sim P(\cdot \mid x, F_m)$, posterior-predictive samples generated from independent posterior draws, and samples from $\hat{G}$. When the conditional CDFs $G_{F_m}$ can be evaluated directly, the integrals can instead be computed by numerical quadrature.

### A.4 Log score for continuous outcomes

| | |
|---|---|
| **Scoring type** | Distributional scoring; strictly proper. |
| **Outcome space** $\mathcal{Y}$ | $\mathcal{Y} = \mathbb{R}$. |
| **Action space** $\mathcal{A}$, deployed action | $\mathcal{A}$: densities with respect to Lebesgue measure $\lambda$; deployed predictive density $\hat{q} := \hat{a}_{\mathcal{S}}(x)$, with $\hat{q} > 0$ almost everywhere under $\bar{p}_{\mathrm{ref}}$. |
| **Loss** $\ell(a, y)$ | $\ell(q, y) = -\log q(y)$. |
| **Representation of** $Y$ **in** $\mathcal{D}$ | None: the representative $\delta_Y$ has no density. The decomposition follows from Lemma A.1. |
| **Generator and divergence** | On densities, $\phi(p) = \int p \log p \, d\lambda$; $D_\phi(p\|q) = \mathrm{KL}(p\|q) = d_\ell(p\|q)$. Generalized entropy: differential entropy $H(p) = -\int p \log p \, d\lambda$. |
| **Conditional Bayes act** $a_f(x)$ | Conditional density $p_f(y) := p(y \mid x, f)$. |
| **Posterior centroid** $\bar{a}_{\mathrm{ref}}(x)$ | Posterior predictive density $\bar{p}_{\mathrm{ref}}(y) := \mathbb{E}_F[p_F(y)] = p(y \mid x, \mathcal{S})$. |

**Derivation.** For fixed $f$, adding and subtracting $\log p_f$ gives $\mathbb{E}_{Y|f}\big[-\log \hat{q}(Y)\big] = H(p_f) + \mathrm{KL}(p_f\|\hat{q})$. Averaging over $F$ and using $\mathbb{E}_F \int p_F \log(\bar{p}_{\mathrm{ref}}/\hat{q})\, d\lambda = \int \bar{p}_{\mathrm{ref}} \log(\bar{p}_{\mathrm{ref}}/\hat{q})\, d\lambda$,

$$\mathbb{E}_F\big[\mathrm{KL}(p_F\|\hat{q})\big] = \mathbb{E}_F\big[\mathrm{KL}(p_F\|\bar{p}_{\mathrm{ref}})\big] + \mathrm{KL}(\bar{p}_{\mathrm{ref}}\|\hat{q}),$$

which is equation 7 with $H_\ell = H$ and $d_\ell = \mathrm{KL}$.

**Decomposition.**

$$R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \underbrace{\mathbb{E}_F\big[H(p_F)\big]}_{\mathrm{AU}_{\mathcal{S}}(x)} + \underbrace{\mathbb{E}_F\big[\mathrm{KL}(p_F\|\bar{p}_{\mathrm{ref}})\big]}_{\text{generalized variance}} + \underbrace{\mathrm{KL}(\bar{p}_{\mathrm{ref}}\|\hat{q})}_{\text{deployment gap}}, \tag{10}$$

where the generalized variance equals the mutual information $I(Y; F \mid x, \mathcal{S}) = H(\bar{p}_{\mathrm{ref}}) - \mathbb{E}_F[H(p_F)]$.

**Aleatoric term.** AU is an expected differential entropy. It may be negative and shifts by $\log c$ when $y$ is rescaled by $c > 0$, whereas both epistemic terms are Kullback–Leibler divergences, non-negative and invariant under invertible measurable transformations of $\mathcal{Y}$. Under additive Gaussian noise whose variance does not depend on $f$, $H(p_f) = \frac{1}{2}\log\big(2\pi e\, \sigma_\varepsilon^2(x)\big)$ and AU is independent of $\mathcal{S}$.

**Closed form for Gaussian reference posteriors.** Suppose the observation model is Gaussian with $f$-independent noise, $Y = \mu_F(x) + \varepsilon$ with $\varepsilon \sim \mathcal{N}\big(0, \sigma_\varepsilon^2(x)\big)$ independent of $F$, and the reference posterior of the conditional mean is Gaussian, $\mu_F(x) \sim \mathcal{N}\big(m_{\mathrm{ref}}(x), \sigma_{\mathrm{ref}}^2(x)\big)$ (as for a GP reference prior). Then $\bar{p}_{\mathrm{ref}} = \mathcal{N}(m_{\mathrm{ref}}, v_{\mathrm{ref}})$ with $v_{\mathrm{ref}} := \sigma_\varepsilon^2 + \sigma_{\mathrm{ref}}^2$. For a Gaussian deployed predictive $\hat{q} = \mathcal{N}(\hat{\mu}, \hat{s}^2)$,

$$\underbrace{\tfrac{1}{2}\log\Big(1 + \frac{\sigma_{\mathrm{ref}}^2}{\sigma_\varepsilon^2}\Big)}_{\text{generalized variance}}, \qquad \underbrace{\tfrac{1}{2}\Big[\log\frac{\hat{s}^2}{v_{\mathrm{ref}}} + \frac{v_{\mathrm{ref}} + (\hat{\mu} - m_{\mathrm{ref}})^2}{\hat{s}^2} - 1\Big]}_{\text{deployment gap}}.$$

For non-Gaussian reference posteriors, $\bar{p}_{\mathrm{ref}}$ is a non-Gaussian mixture. Terms involving $\log \bar{p}_{\mathrm{ref}}$ then require either an evaluable mixture density or the Monte Carlo estimate $\hat{p}(y) = M^{-1}\sum_{m=1}^{M} p_{F_m}(y)$ with $F_m \sim \Pi_{\mathrm{ref}}(\cdot \mid \mathcal{S})$; by Jensen's inequality, $\log \hat{p}$ is biased downward for finite $M$, but it is consistent as $M \to \infty$.

### A.5 Multiclass log loss

| | |
|---|---|
| **Scoring type** | Distributional scoring; strictly proper. |
| **Outcome space** $\mathcal{Y}$ | $\mathcal{Y} = \{1, \dots, K\}$; binary classification is $K = 2$. |
| **Action space** $\mathcal{A}$, deployed action | $\mathcal{A} = \Delta^{K-1}$, the probability simplex; deployed class-probability vector $\hat{\pi} := \hat{a}_{\mathcal{S}}(x)$ in its relative interior. |
| **Loss** $\ell(a, y)$ | $\ell(\pi, y) = -\log \pi_y$. |
| **Representation of** $Y$ **in** $\mathcal{D}$ | $\mathcal{D} = \Delta^{K-1}$; the one-hot vector $e_Y$. |
| **Generator and divergence** | $\phi(u) = \sum_k u_k \log u_k$ with $0 \log 0 := 0$; $D_\phi(u\|v) = \sum_k u_k \log(u_k/v_k) = \mathrm{KL}(u\|v)$, so that $D_\phi(e_Y\|\pi) = -\log \pi_Y$. |
| **Conditional Bayes act** $a_f(x)$ | Conditional class probabilities $\pi_f(x) := \mathbb{E}_{Y\|f}[e_Y] = \big(P(Y = k \mid x, f)\big)_{k=1}^K$. |
| **Posterior centroid** $\bar{a}_{\mathrm{ref}}(x)$ | Posterior predictive probabilities $\bar{\pi}_{\mathrm{ref}}(x) := \mathbb{E}_F[\pi_F(x)] = \big(P(Y = k \mid x, \mathcal{S})\big)_{k=1}^K$. |

**Derivation.** Equation (4) applies directly. The conditional Bayes risk is the Shannon entropy of the conditional class probabilities,

$$\mathbb{E}_{Y|f}\big[-\log \pi_{f,Y}(x)\big] = H\big(\pi_f(x)\big).$$

**Decomposition.**

$$R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \underbrace{\mathbb{E}_F\big[H(\pi_F(x))\big]}_{\mathrm{AU}_{\mathcal{S}}(x)} + \underbrace{\mathbb{E}_F\big[\mathrm{KL}(\pi_F(x)\|\bar{\pi}_{\mathrm{ref}}(x))\big]}_{\text{generalized variance}} + \underbrace{\mathrm{KL}(\bar{\pi}_{\mathrm{ref}}(x)\|\hat{\pi})}_{\text{deployment gap}}, \tag{11}$$

where the generalized variance equals $I(Y; F \mid x, \mathcal{S}) = H(\bar{\pi}_{\mathrm{ref}}) - \mathbb{E}_F[H(\pi_F)]$.

**Aleatoric term.** Unlike in regression with an $f$-independent noise law, $H(\pi_f(x))$ need not be constant in $f$. AU is then a genuine posterior average and can depend on $\mathcal{S}$. It cannot if $H(\pi_f(x))$ is constant over the support of the reference prior (Section A.7). For example, if the reference prior supports only labelings that are deterministic at $x$, $\pi_f(x) \in \{e_1, \dots, e_K\}$, then $\mathrm{AU}_{\mathcal{S}}(x) = 0$ for every sample and all uncertainty at $x$ is epistemic, however uncertain the posterior is about which class is correct. By concavity of $H$, $\mathbb{E}_F[H(\pi_F)] \le H(\bar{\pi}_{\mathrm{ref}})$, and the gap is exactly the generalized variance: $H(\bar{\pi}_{\mathrm{ref}}) = \mathrm{AU}_{\mathcal{S}}(x) + I(Y; F \mid x, \mathcal{S})$. Evaluating at $\hat{\pi} = \bar{\pi}_{\mathrm{ref}}$ removes the deployment gap and yields the information-theoretic decomposition $H(\bar{\pi}_{\mathrm{ref}}) = \mathbb{E}_F[H(\pi_F)] + I(Y; F \mid x, \mathcal{S})$, with the spread taken under the reference posterior rather than an estimator posterior (cf. Section 2.2).

### A.6 Brier score

| | |
|---|---|
| **Scoring type** | Distributional scoring; strictly proper. |
| **Outcome space** $\mathcal{Y}$ | $\mathcal{Y} = \{1, \dots, K\}$. |
| **Action space** $\mathcal{A}$, deployed action | $\mathcal{A} = \Delta^{K-1}$; deployed class-probability vector $\hat{\pi} := \hat{a}_{\mathcal{S}}(x)$. |
| **Loss** $\ell(a, y)$ | $\ell(\pi, y) = \|\pi - e_y\|_2^2$. |
| **Representation of** $Y$ **in** $\mathcal{D}$ | $\mathcal{D} = \Delta^{K-1}$; the one-hot vector $e_Y$. |
| **Generator and divergence** | $\phi(u) = \|u\|_2^2$; $D_\phi(u\|v) = \|u - v\|_2^2$. |
| **Conditional Bayes act** $a_f(x)$ | $\pi_f(x)$, as in Section A.5. |
| **Posterior centroid** $\bar{a}_{\mathrm{ref}}(x)$ | $\bar{\pi}_{\mathrm{ref}}(x)$, as in Section A.5. |

**Derivation.** Equation (4) applies directly. The conditional Bayes risk is $\mathbb{E}_{Y|f}\big[\|e_Y - \pi_f(x)\|_2^2\big] = 1 - \|\pi_f(x)\|_2^2$.

**Decomposition.**

$$R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \underbrace{\mathbb{E}_F\big[1 - \|\pi_F(x)\|_2^2\big]}_{\mathrm{AU}_{\mathcal{S}}(x)} + \underbrace{\mathbb{E}_F\big[\|\pi_F(x) - \bar{\pi}_{\mathrm{ref}}(x)\|_2^2\big]}_{\text{generalized variance}} + \underbrace{\|\bar{\pi}_{\mathrm{ref}}(x) - \hat{\pi}\|_2^2}_{\text{deployment gap}}. \tag{12}$$

Table 2: Instantiations of Equation (4). EU is the sum of the last two columns. For every distributional score the centroid is the posterior predictive.

| **Loss** | **Centroid** | **AU** | **Generalized variance** | **Deployment gap** |
|---|---|---|---|---|
| Squared | $m_{\mathrm{ref}}$ | $\mathbb{E}_F[\sigma_\varepsilon^2(x,F)]$ | $\sigma^2_{\mathrm{ref}}(x)$ | $(\hat{a}-m_{\mathrm{ref}})^2$ |
| CRPS | $\bar{G}_{\mathrm{ref}}$ | $\mathbb{E}_F\int G_F(1-G_F)\,dt$ | $\mathbb{E}_F\int(G_F-\bar{G}_{\mathrm{ref}})^2dt$ | $\int(\bar{G}_{\mathrm{ref}}-\hat{G})^2dt$ |
| Log score (cont.) | $\bar{p}_{\mathrm{ref}}$ | $\mathbb{E}_F[H(p_F)]$ | $I(Y;F\mid x,\mathcal{S})$ | $\mathrm{KL}(\bar{p}_{\mathrm{ref}}\|\hat{q})$ |
| Multiclass log | $\bar{\pi}_{\mathrm{ref}}$ | $\mathbb{E}_F[H(\pi_F)]$ | $I(Y;F\mid x,\mathcal{S})$ | $\mathrm{KL}(\bar{\pi}_{\mathrm{ref}}\|\hat{\pi})$ |
| Brier | $\bar{\pi}_{\mathrm{ref}}$ | $\mathbb{E}_F[1-\|\pi_F\|_2^2]$ | $\mathbb{E}_F\|\pi_F-\bar{\pi}_{\mathrm{ref}}\|_2^2$ | $\|\bar{\pi}_{\mathrm{ref}}-\hat{\pi}\|_2^2$ |

**Aleatoric term.** AU is the posterior expected Gini impurity of the conditional class probabilities. As for the log loss, $1-\|\pi_f(x)\|_2^2$ need not be constant in $f$, so AU can depend on $\mathcal{S}$. It vanishes for every sample if the reference prior supports only labelings that are deterministic at $x$.

### A.7 When is the aleatoric term independent of the reference posterior?

By Equation (3), $\mathrm{AU}_{\mathcal{S}}(x)=\mathbb{E}_F\big[R_F^\star(x)\big]$. If the conditional Bayes risk $R_f^\star(x)$ is constant over the support of the reference prior, AU takes the same value for every observed sample. For the three regression losses, this holds under additive noise whose law given $x$ does not depend on $f$, since the variance, the CRPS entropy $\frac{1}{2}\mathbb{E}|Y-Y'|$, and the differential entropy are all invariant under shifts. The condition is sufficient, not necessary, and it is not met in two situations: when the reference prior places uncertainty on the noise law itself, and in classification, where $R_f^\star(x)$ is a function of $\pi_f(x)$. In both situations AU is the posterior expectation of an unknown irreducible risk and can depend on the observed sample, but it need not. For instance, if the reference prior supports only deterministic labelings, $\pi_f(x)\in\{e_1,\dots,e_K\}$, then $R_f^\star(x)=0$ for every $f$ under both the log loss and the Brier score, so AU vanishes for every sample and all uncertainty at $x$ is epistemic.

# B SEMI-SYNTHETIC REGRESSION TASKS AND REFERENCE POSTERIORS

Computing the ground-truth uncertainty of Section 2 requires two objects that a real dataset does not provide: the noise model $P_{y|x,f}$, which defines aleatoric uncertainty, and the reference posterior $\Pi_{\mathrm{ref}}(\cdot \mid \mathcal{S})$ over ground-truth functions, which is one of the central objects for epistemic uncertainty. We therefore build *semi-synthetic* tasks: real covariates, but the targets are generated by a process we fully control. This appendix describes how these tasks and their reference posteriors are produced.

**The central idea.** Every task is produced in three steps. We choose a prior $\Pi_0$ over functions, draw one function $f^\star \sim \Pi_0$ and add noise to it to obtain training targets, and then condition the *same* prior $\Pi_0$ on the resulting training sample $\mathcal{S}$. Because the prior used for conditioning is exactly the prior that generated $f^\star$, the reference posterior $\Pi_{\mathrm{ref}}(\cdot \mid \mathcal{S})$ is correctly specified by construction: seen from the learner's side, $f^\star$ is simply one draw from it. This holds for *any* choice of prior hyperparameters. The hyperparameters therefore only decide how realistic and how varied the tasks are. What *can* make it wrong is an implementation error or a sampler that has not converged, and Appendix B.7 describes how we test for both.

**Overview of the generation pipeline.**

*Once per dataset* (Appendices B.1–B.4):

1. Load the real covariates, scale them to $[0, 1]$, and split the rows into an evaluation pool and a pool of candidate training inputs.
2. For each prior family (GP, HBART, BNN), fix one scale constant so that prior draws vary by about one unit over the evaluation pool.
3. For each of the four noise modes, fix the noise variance function $\sigma_\varepsilon^2(x)$.
4. Fix the training-set size $n$.

*Once per task instance*, i.e. per (dataset, prior family, noise mode, coverage scenario, seed) (Appendix B.5):

5. Select $n$ training inputs according to the coverage scenario.
6. Draw the ground truth $f^\star \sim \Pi_0$.
7. Draw noisy targets $y_i = f^\star(x_i) + \varepsilon_i$ at the training inputs.
8. Condition $\Pi_0$ on $\mathcal{S}$ and store $D = 1000$ draws from $\Pi_{\mathrm{ref}}(\cdot \mid \mathcal{S})$ at the evaluation points.
9. Compute the oracle targets from the stored draws (Appendix B.6).

## B.1 COVARIATES

We use 13 tabular regression datasets from UCI and OpenML (Table 4), with between 4 and 21 covariates and between 908 and 53 940 rows. Numeric columns are kept as they are; categorical columns with exactly two levels are coded as $0/1$; categorical columns with more levels are dropped, because an integer code would impose an ordering that the covariates do not have. Rows with missing values are removed. Each covariate is then min-max scaled to $[0, 1]$.

The original targets of these datasets are *not* used as targets.

Each dataset is split once, with a fixed seed, into two disjoint parts. The *evaluation pool* $X_{\mathrm{eval}}$ consists of $n_{\mathrm{eval}} = \min(1500, \lfloor N/2 \rfloor)$ randomly chosen rows, where $N$ is the number of rows. All uncertainty targets and all method predictions are evaluated on this pool. The remaining rows form the *candidate pool*, from which training inputs are drawn. No evaluation point is ever a training input.

## B.2 PRIORS OVER GROUND-TRUTH FUNCTIONS

We use three prior families $\Pi_0$. They produce different kinds of functions: functions with correlation structure induced by varying kernels (GP), piecewise constant functions with axis-aligned jumps (HBART), and functions represented by a neural network (BNN). Using three families ensures that

no method is evaluated only on functions that happen to match its own inductive bias. Table 6 summarises all settings.

**A common scale.** Left at their default settings, the three priors produce functions of very different size: one prior's draws might vary by $\pm 0.3$, another's by $\pm 5$. To make them comparable, we multiply every prior by a constant $a$, chosen per dataset and prior family so that a typical draw varies by about one unit across the evaluation pool. Concretely, we take 100 draws from the prior, compute each draw's standard deviation over the evaluation pool, and set $a$ so that the median of these standard deviations is one. All tasks then share the same signal scale, so a given noise level (Appendix B.3) makes tasks equally noisy whichever prior generated them. The constant $a$ is fixed once, before any task is generated, and simply becomes part of the prior. Individual draws are never rescaled.

**Gaussian process (GP).** $f \sim \mathcal{GP}\big(\mu,\, a^2 k\big)$. To introduce variation in the GP tasks we set a combination of kernels individually for each of the datasets. The resulting kernels differ considerably between datasets (Table 4): stationary Matérn, RBF, rational-quadratic and powered-exponential kernels, sums with a linear or periodic component, and arc-cosine kernels. The mean $\mu$ is zero. The kernel's output variance is the scale constant $a^2$ (Step 2).

**Heteroscedastic Bayesian additive regression trees (HBART).** A sum of $m = 200$ regression trees with the standard BART prior (Chipman et al., 2010): a node at depth $k$ is split with probability $0.95\,(1+k)^{-2}$, up to a maximum depth of 6. The split covariate is chosen uniformly at random and the split value uniformly from a grid of 255 equally spaced thresholds in $[0, 1]$. The leaf values are independent $\mathcal{N}(0, a^2/m)$, so that the sum of $m$ leaves has prior standard deviation $a$. The "heteroscedastic" part of the name refers to the likelihood: when conditioning, each observation is weighted by its own, known noise variance $\sigma_\varepsilon^2(x_i)$ (Appendix B.5).

**Bayesian neural network (BNN).** A fully connected network with one hidden layer of $H = 50$ $\tanh$ units,

$$u(x) = \frac{1}{\sqrt{H}} \sum_{h=1}^{H} v_h \tanh\Big(\tfrac{1}{\sqrt{d}}\, w_h^\top (2x - 1) + c_h\Big),$$
$$f(x) = b + a\Big(u(x) - \tfrac{1}{R}\textstyle\sum_{r=1}^{R} u(x_r^{\mathrm{ref}})\Big),$$

with all weights and biases $w_h, c_h, v_h, b$ independent standard normal. The $1/\sqrt{\text{fan-in}}$ factors are the usual scaling under which the network prior approaches a GP as the width grows. The second term is centred on a fixed set of $R = 65$ reference inputs drawn uniformly from $[0, 1]^d$. Without this centring, most of the prior variance of $f$ would be a constant offset, and the scale $a$ would control the overall level of $f$ rather than how much it varies across inputs. With it, $a$ controls the variation and the bias $b$ controls the level. The same reference inputs are used when drawing $f^\star$ and when conditioning. In practice this prior produces smooth functions over the data.

### B.3 Aleatoric noise

Observations follow a additive Gaussian noise model:

$$y = f(x) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}\big(0, \sigma_\varepsilon^2(x)\big).$$

The noise variance $\sigma_\varepsilon^2(x)$ is not drawn at random: it is a fixed, known function of $x$. For a given dataset and noise mode it is identical for all three prior families, so tasks from different priors differ only in the function $f$. We write it as

$$\log \sigma_\varepsilon^2(x) = c + s\, h(x),$$

where $h$ is a *shape* function standardised to mean zero and standard deviation one over the evaluation pool, $s$ controls how strongly the noise varies, and $c$ sets the overall noise level.

**Four noise modes.** All shapes are functions of a single score $z(x)$, the first principal component of the column-standardised covariates, itself standardised to mean zero and unit variance. This gives a direction along which the noise changes that is defined for any number of covariates, and it makes every shape a closed-form function of $x$. Before standardisation, the shapes are

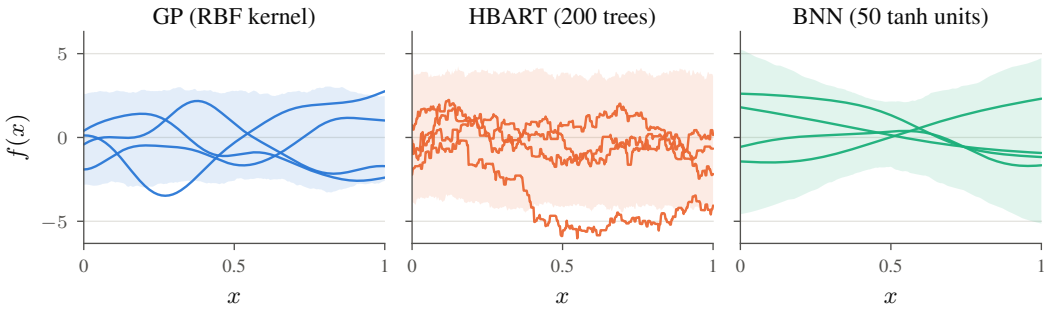



Figure 4: **Prior draws from the three prior families** on a one-dimensional input. Lines show four independent draws $f \sim \Pi_0$; shaded regions show the pointwise central 95% interval over 500 draws. Each prior is scaled with the benchmark's rule, so that a typical draw has standard deviation one across the inputs. The GP panel uses an RBF kernel for illustration; in the benchmark the GP kernel is set separately for each dataset. HBART produces piecewise-constant functions with a large per-draw offset; the BNN prior produces smooth functions.

| Mode | Shape before standardisation |
|---|---|
| homoscedastic | $h(x) = 0$ (constant noise) |
| smooth monotone | $1/\big(1 + e^{-2z(x)}\big)$ |
| multimodal | $\exp\big(-\frac{(z(x)+1.1)^2}{2\cdot 0.55^2}\big) + \exp\big(-\frac{(z(x)-1.1)^2}{2\cdot 0.55^2}\big)$ |
| discontinuous | $\mathbf{1}\{z(x) > 0\}$ |

The homoscedastic mode is the baseline. The smooth monotone mode lets the noise increase steadily along one direction. In the multimodal mode there are two separate high-noise regions, so the noise is not monotone in any single direction. The discontinuous mode has a jump in noise level across a hyperplane, which no smooth noise model can represent exactly.

**Strength and level.** For the three heteroscedastic modes we set $s = 1$, so the log noise variance has standard deviation one over the evaluation pool; for the homoscedastic mode $s = 0$. The level $c$ is set so that the average noise variance over the evaluation pool equals a target noise-to-signal ratio $\rho$:

$$\frac{1}{n_{\text{eval}}} \sum_{x \in X_{\text{eval}}} \sigma_\varepsilon^2(x) = \rho \quad \Longleftrightarrow \quad c = \log \rho - \log \Big( \tfrac{1}{n_{\text{eval}}} \textstyle\sum_{x \in X_{\text{eval}}} e^{sh(x)} \Big).$$

Since prior draws have variance about one (Appendix B.2), $\rho$ is the ratio of noise variance to signal variance. Each dataset has its own $\rho$, drawn once, log-uniformly from $[0.2, 0.5]$ (Table 4). The noise therefore accounts for between roughly $17\%$ and $30\%$ of the total variance of $y$, depending on the dataset. As an example, on White Wine Quality the noise standard deviation is constant at $0.47$ in the homoscedastic mode and ranges over $[0.17, 0.78]$ (smooth monotone), $[0.11, 0.72]$ (multimodal) and $[0.22, 0.60]$ (discontinuous) in the other modes. The noise modes are illustrated in Figure 5

### B.4 Training sample and data coverage

**Training-set size.** Each dataset has a fixed training-set size $n$ (Table 4). It is drawn once as an integer in $[1500, 5000]$, limited to at most $60\%$ of the rows, and reduced further where necessary so that both coverage scenarios below can be realised with the same $n$. The two scenarios thus differ only in *where* the training inputs lie, not in how many there are. The resulting sizes range from 336 to 4 771.

**Coverage scenarios.** To control where training data is available, we sort all rows of the dataset by their Mahalanobis distance to the mean of the covariates (using the empirical covariance, shrunk by $10\%$ towards a scaled identity for numerical stability) and split them at the quintiles into five *shells*, from the innermost to the outermost fifth. Training inputs are then drawn uniformly without replacement from the candidate pool, with a prescribed share from each shell:

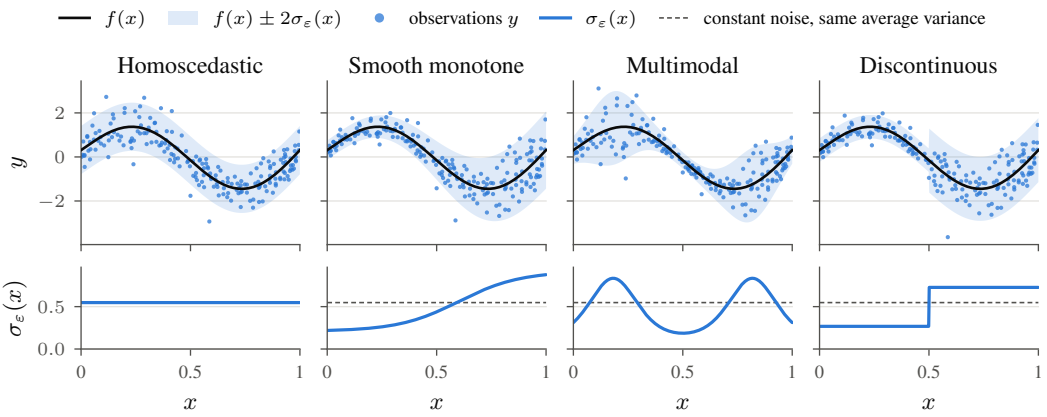



Figure 5: **The four aleatoric noise modes** on a one-dimensional input. Top: a fixed function $f$, the band $f \pm 2\sigma_\varepsilon(x)$ and noisy observations, generated from the same underlying random draws in every panel. Bottom: the noise standard deviation $\sigma_\varepsilon(x)$; the dashed line is the constant noise level with the same average variance. All modes have the same average noise variance (here a noise-to-signal ratio of $\rho = 0.3$) and, apart from the homoscedastic mode, the same spread of $\log \sigma_\varepsilon^2$. They differ only in where the noise is high: increasing along one direction, in two separate regions, or with a jump across a hyperplane.

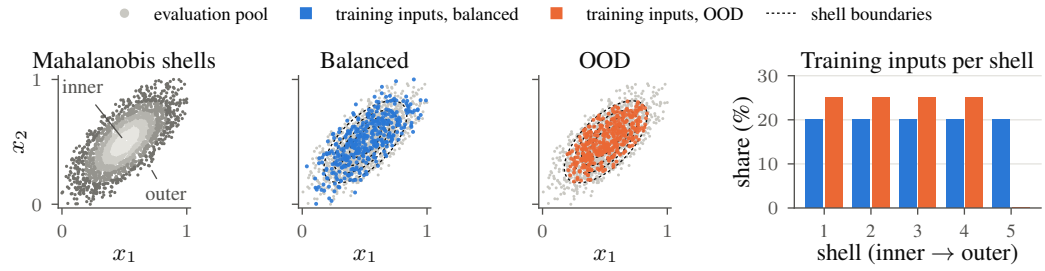



Figure 6: **The two data-coverage scenarios**, illustrated on a synthetic two-dimensional covariate cloud. Left: all rows are split into five shells at the quintiles of their Mahalanobis distance to the covariate mean (light: inner, dark: outer). Middle: the evaluation pool (grey) with the training inputs of each scenario; dashed lines are the shell boundaries. In the balanced scenario the training inputs cover all shells equally; in the OOD scenario none come from the outermost shell, so about 20% of the evaluation points lie in a region without training data. Right: realised share of the training inputs per shell. Both scenarios use the same number of training inputs.

- **Balanced:** 20% from every shell. The training inputs cover the input space in the same way as the evaluation pool; there is no covariate shift.
- **OOD:** 25% from each of the four inner shells and none from the outermost shell. The roughly 20% of evaluation points that lie in the outermost shell have no training data nearby, so they require extrapolation.

In the OOD scenario, epistemic uncertainty should rise in the empty outer shell, while the aleatoric target, which is a fixed function of $x$, is unaffected by where the training data lies. The coverage scenarios are illustrated in Figure 6.

### B.5 Generating one task instance

A task instance is defined by a dataset, a prior family, a noise mode, a coverage scenario and a seed. It is generated as follows.

1. **Training inputs.** Select $n$ training inputs $x_1, \ldots, x_n$ from the candidate pool according to the coverage scenario (Appendix B.4).
2. **Ground truth.** Draw one function $f^\star \sim \Pi_0$, evaluated jointly at the evaluation pool and the training inputs.
3. **Noisy targets.** Draw $y_i = f^\star(x_i) + \sigma_\varepsilon(x_i)\,\epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0,1)$, and set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$. This is all a method (learner) sees.
4. **Reference posterior.** Condition $\Pi_0$ on $\mathcal{S}$, with the noise variances $\sigma_\varepsilon^2(x_i)$ treated as known, and store $D = 1000$ draws $f^{(1)}, \ldots, f^{(D)} \sim \Pi_{\mathrm{ref}}(\cdot \mid \mathcal{S})$ evaluated at every point of the evaluation pool (and at the training inputs, for the checks in Appendix B.7).

The reference posterior is a posterior over the function $f$ only. The noise is known to it, just as the noise model is known in Section 2. This is what makes the aleatoric target exact.

Each instance draws its own random numbers, derived from the seed and from the instance's dataset, prior family, noise mode and scenario. Instances are therefore independent replicates: two noise modes or two scenarios with the same seed share neither the ground truth nor the training sample.

**Conditioning.** The three prior families require different conditioning procedures.

- **GP: exact.** With known noise variances the likelihood is Gaussian, and the posterior is available in closed form (Rasmussen & Williams, 2005). With $K$ the prior covariance of the training inputs, $\Sigma = \mathrm{diag}\left(\sigma_\varepsilon^2(x_1), \ldots, \sigma_\varepsilon^2(x_n)\right)$ and $k_x$ the prior covariances between $x$ and the training inputs,

$$m_{\mathcal{S}}(x) = \mu + k_x^\top (K + \Sigma)^{-1}(y - \mu\mathbf{1}),$$
$$\mathrm{Cov}\left(f(x), f(x') \mid \mathcal{S}\right) = a^2 k(x, x') - k_x^\top (K + \Sigma)^{-1} k_{x'}.$$

  The only difference from textbook GP regression is that each observation has its own noise variance, so noisy observations automatically count less.
- **HBART: MCMC.** We use the standard BART sampler (Chipman et al., 2010) as implemented in `bartz` (Petrillo, 2025), with each observation's known noise variance in the likelihood. We run 250 independent chains for 5 000 warm-up iterations. To help the chains move between very different tree configurations, the first 75% of the warm-up uses a tempered likelihood whose temperature decreases geometrically from 3 to 1 in ten stages; the rest of the warm-up and all retained draws use the untempered likelihood. Each chain then contributes 4 draws, spaced 25 iterations apart, giving $D = 1000$.
- **BNN: MCMC.** We sample all network weights with the No-U-Turn sampler (Hoffman & Gelman, 2011) in NumPyro (Phan et al., 2019): 4 chains, each with 500 warm-up iterations and 250 retained draws ($D = 1000$), target acceptance probability 0.9. Each retained weight vector is turned into a function draw by evaluating the network at the evaluation pool.

The code path that draws $f^\star$ and the code path that conditions use the same prior definition for each family (the same kernel function, tree prior and network), so that the generating and the conditioning prior cannot drift apart.

**Size of the benchmark.** The full grid has 13 datasets $\times$ 3 prior families $\times$ 4 noise modes $\times$ 2 coverage scenarios $=$ 312 conditions. For each condition we generate 10 seeds, which gives 3 120 task instances with reference posteriors.

### B.6 Oracle targets

Under squared loss the posterior-risk decomposition of Appendix A.2 reads, at an evaluation point $x$,

$$R_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \underbrace{\sigma_\varepsilon^2(x)}_{\mathrm{AU}_{\mathcal{S}}(x)} + \underbrace{\sigma_f^2(x) + \left(\hat{a}_{\mathcal{S}}(x) - m_{\mathcal{S}}(x)\right)^2}_{\mathrm{EU}_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x)}.$$

Every term can be read off the stored artefacts:

- **Aleatoric uncertainty** $\mathrm{AU}_{\mathcal{S}}(x) = \sigma_\varepsilon^2(x)$ is known exactly. Because the noise does not depend on $f$, the posterior expectation of the Bayes risk is simply the noise variance.
- **Posterior mean** (the reference Bayes act $\bar{a}_{\mathrm{ref}}$): $m_{\mathcal{S}}(x) \approx \frac{1}{D}\sum_{d=1}^{D} f^{(d)}(x)$.
- **Posterior variance** (the generalised-variance term): $\sigma_f^2(x) \approx \frac{1}{D}\sum_{d=1}^{D}\big(f^{(d)}(x) - m_{\mathcal{S}}(x)\big)^2$. It does not depend on the method.
- **Epistemic uncertainty of a method**: given a method's prediction $\hat{a}_{\mathcal{S}}(x)$, $\mathrm{EU}_{\mathcal{S}}(\hat{a}_{\mathcal{S}}; x) = \sigma_f^2(x) + \big(\hat{a}_{\mathcal{S}}(x) - m_{\mathcal{S}}(x)\big)^2$, where the second term is the deployment gap. This target is computed separately for every method after it has been trained on $\mathcal{S}$.

Since the realised ground truth $f^\star$ is stored as well, the fixed-truth excess risk $\big(\hat{a}_{\mathcal{S}}(x) - f^\star(x)\big)^2$ of Section 2.3 can be computed on the same instances for comparison. Because the full set of posterior draws is stored rather than only its mean and variance, the targets for other losses (CRPS and log score, Appendices A.3 and A.4) can also be computed without regenerating any data. With $D = 1000$ draws, the Monte Carlo error of $\sigma_f^2(x)$ is about $5\%$ of its value at a single point, and this error averages out over the points and instances that any reported score aggregates. For the GP, $m_{\mathcal{S}}$ and $\sigma_f^2$ are also available in closed form.

### B.7 Correctness and convergence checks

We assess the calibration and convergence of the reference posteriors on all $3{,}120$ benchmark instances (Table 3).

*Calibration.* If $f^\star \sim \Pi_0$ and the draws follow the posterior, the rank of $f^\star(x)$ among the $D$ draws is uniform (Talts et al., 2020). For all three prior families the pooled rank histograms are flat. The coverage of the central $90\%$ interval is $0.90$ both pooled (with confidence intervals that resample whole instances, since points within an instance are dependent) and within the 104 combinations of dataset, noise mode and coverage scenario (interquartile range $0.89$–$0.91$). Because these checks could also be passed by draws that ignore the data, we note that the posterior mean tracks the realised ground truth closely (median correlation with $f^\star$ between $0.94$ and $1.00$). *Convergence.* The GP posterior is exact. For the BNN, an instance is used only if the split-$\hat{R}$ (Vehtari et al., 2021) of its four chains, computed in function space at every evaluation point, is at most $1.05$ everywhere. Seeds that fail are replaced by the next seed, for all prior families alike ($5.3\%$ of seeds with balanced and $8.6\%$ with OOD coverage). In the retained instances, $98\%$ of point-wise $\hat{R}$ values are below $1.01$, and restricting the BNN tasks to instances with $\hat{R} \le 1.01$ everywhere leaves the method ranking unchanged (Kendall's $\tau = 0.98$; no method moves by more than one rank). HBART pools 250 independent chains after a tempered warm-up. On two of the largest datasets (Protein and kin8nm), a four times longer warm-up changes the $90\%$ coverage by less than $0.003$ and the posterior standard deviation by at most about $5\%$. *Precision.* The Monte Carlo error of the posterior variance, estimated by batch means over chains so that within-chain dependence is accounted for, has a median of $4$–$6\%$ and a 95th percentile below $9\%$.

Table 3: Reference-posterior diagnostics over the benchmark instances (1,040 per prior family). Rank histogram: largest deviation of a bin share from $5\%$ (20 bins). Coverage: share of evaluation points whose ground truth lies in the central $90\%$ posterior interval; pooled with a $95\%$ bootstrap interval over instances, and the range over the 104 dataset $\times$ noise mode $\times$ coverage cells. MCSE: relative Monte Carlo standard error of the posterior variance by batch means (20 batches of whole chains or chain segments), over all evaluation points.

| | GP | HBART | BNN |
|---|---|---|---|
| Conditioning | exact | 250 chains | NUTS, 4 chains |
| $90\%$ coverage, pooled | $0.899$ $[0.898, 0.901]$ | $0.902$ $[0.901, 0.903]$ | $0.901$ $[0.898, 0.905]$ |
| $90\%$ coverage, range over cells | 0.880–0.932 | 0.881–0.921 | 0.838–0.957 |
| MCSE of $\sigma_f^2$, median / 95th pct. | $4.4\%$ / $5.7\%$ | $5.6\%$ / $8.6\%$ | $4.7\%$ / $6.1\%$ |
| Point-wise $\hat{R}$, median / 99th pct. | – | – | $1.000$ / $1.013$ |
| Corr. $(m_{\mathcal{S}}, f^\star)$, median | 0.94 | 0.98 | 1.00 |

**Released artefacts.** For every dataset we release the processed covariates, the evaluation/candidate split, the scale constant of each prior family and the noise parameters of each mode. For every task instance we release the training indices and targets $y$, the noise variances at the training and evaluation points, the ground truth $f^\star$ at the training and evaluation points, the $D \times n_{\text{eval}}$ matrix of posterior draws, and the sampler diagnostics.

### B.8 SUMMARY TABLES

Table 4: Datasets. $d$: number of covariates; $N$: rows after cleaning; $n_{\text{eval}}$: size of the evaluation pool; $n$: training-set size (the same in both coverage scenarios); $\rho$: noise-to-signal ratio; GP kernel: the kernel family fitted to the real targets and used by the GP prior (Mat: Matérn; RQ: rational quadratic; PE: powered exponential; lin: linear).

| Dataset | Source | $d$ | $N$ | $n_{\text{eval}}$ | $n$ | $\rho$ | GP kernel |
|---|---|---|---|---|---|---|---|
| Abalone | UCI 1 | 7 | 4 177 | 1 500 | 2 084 | 0.28 | Arc-cosine |
| Airfoil Self-Noise | UCI 291 | 5 | 1 503 | 751 | 584 | 0.32 | Mat-$\frac{1}{2}$ + lin |
| Concrete Strength | UCI 165 | 8 | 1 030 | 515 | 348 | 0.23 | RQ + lin |
| CPU Activity | OpenML 44978 | 21 | 8 192 | 1 500 | 1 782 | 0.25 | Mat-$\frac{5}{2}$ + PE |
| Diamonds | OpenML 44979 | 6 | 53 940 | 1 500 | 2 000 | 0.32 | RBF |
| Health Insurance | OpenML 44993 | 4 | 22 272 | 1 500 | 3 279 | 0.42 | Arc-cosine |
| kin8nm | OpenML 44980 | 8 | 8 192 | 1 500 | 3 136 | 0.33 | Mat-$\frac{3}{2}$ |
| Protein Structure | OpenML 44963 | 9 | 45 730 | 1 500 | 3 223 | 0.36 | RQ + lin |
| QSAR Fish Toxicity | UCI 504 | 6 | 908 | 454 | 336 | 0.41 | Mat-$\frac{3}{2}$ |
| Sarcos | OpenML 44976 | 21 | 48 933 | 1 500 | 4 771 | 0.42 | Mat-$\frac{5}{2}$ |
| Seoul Bike Sharing | UCI 560 | 9 | 8 760 | 1 500 | 1 945 | 0.21 | Periodic + lin |
| Space GA | OpenML 45402 | 6 | 3 107 | 1 500 | 1 260 | 0.21 | PE |
| White Wine Quality | OpenML 44971 | 11 | 4 898 | 1 500 | 2 668 | 0.23 | RBF |

Table 5: References for the datasets in Table 4.

| Dataset | Reference |
|---|---|
| Abalone | Nash et al. (1994) |
| Airfoil Self-Noise | (Brooks et al., 1989) |
| Concrete Strength | Yeh (1998) |
| CPU Activity | (OpenML, 2022a; DELVE, n.d.) |
| Diamonds | OpenML (2022b); Wickham (n.d.) |
| Health Insurance | OpenML (2022c); Olson (1998) |
| kin8nm | OpenML (2022d); Ghahramani (1996) |
| Protein Structure | OpenML (2022e); Rana (2013) |
| QSAR Fish Toxicity | Ballabio et al. (2015) |
| Sarcos | OpenML (2022g); Vijayakumar & Schaal (2000) |
| Seoul Bike Sharing | UCI |
| Space GA | OpenML (2023); Pace & Barry (1997) |
| White Wine Quality | OpenML (2022h); Cortez et al. (2009) |

Table 6: Prior families and how their reference posteriors are computed. $a$ is the per-dataset scale constant (Appendix B.2). All samplers return $D = 1000$ posterior draws of $f$ at the evaluation pool.

| Family | Prior over $f$ | Conditioning on $\mathcal{S}$ (noise known) |
|---|---|---|
| GP | $\mathcal{GP}(\mu, a^2 k)$; kernel family, lengthscales and constant mean fitted to the real targets by marginal likelihood | Exact closed form; $D$ joint draws from the Gaussian posterior |
| HBART | Sum of 200 trees; split probability $0.95(1 + k)^{-2}$ at depth $k$, maximum depth 6; uniform split covariate; 255 split thresholds per covariate; leaves $\mathcal{N}(0, a^2/200)$ | BART MCMC (`bartz`); 250 chains; 5 000 warm-up iterations, the first 75% with likelihood temperature decreasing from 3 to 1; 4 draws per chain, thinned by 25 |
| BNN | One hidden layer of 50 $\tanh$ units; standard normal weights and biases with $1/\sqrt{\text{fan-in}}$ scaling; output centred on 65 fixed reference inputs and multiplied by $a$; output bias $\mathcal{N}(0, 1)$ | NUTS (NumPyro) on all weights; 4 chains $\times$ (500 warm-up + 250 draws); target acceptance 0.9; double precision |

# C MODELS

This appendix specifies the 21 regression methods of the benchmark. All methods expose the same prediction interface: a predictive mean $\hat{y}(x)$, an aleatoric uncertainty estimate, an epistemic uncertainty estimate, and a total uncertainty estimate, all reported as variances. For every method the total uncertainty is the sum of its aleatoric and epistemic estimates. Whenever a method is trained in a standardized target space, predictive means and variances are transformed back to the original target scale before evaluation.

**Hyperparameter tuning.** Hyperparameters were selected separately for every method and every benchmark condition, i.e. every combination of dataset, noise mode, prior family and coverage scenario ($13 \times 4 \times 3 \times 2 = 312$ conditions). Each selection is one Bayesian-optimization sweep of 30 trials in Weights & Biases (Biewald, 2020) over the search space listed with the method; entries with a single value are fixed. A sweep uses the training sample of the condition's first instance (seed 0): a uniformly random 80% of it trains a candidate configuration, and the remaining 20% scores it by the Gaussian negative log-likelihood of the targets under $\mathcal{N}\big(\hat{y}(x), \hat{\sigma}^2_{\text{tot}}(x)\big)$, where $\hat{\sigma}^2_{\text{tot}}$ is the method's total uncertainty. The winning configuration is then used for all seeds of that condition. Tuning never sees the evaluation points or any oracle quantity. Because the objective scores only the total predictive variance, nothing in the tuning rewards a particular split into aleatoric and epistemic parts. In total, the benchmark ran 6,552 sweeps.

**Compute.** All experiments were run on a GPU cluster. The methods implemented in JAX (GP, VLGP, DKL, SNGP-Het and all neural-network methods) were fitted on single NVIDIA GPUs (RTX 2080 Ti, V100 or A100); BLR and the tree-based methods ran on CPU nodes. Summed over all jobs, the 65,457 final fits (3,117 per method) took about 2,850 hours of wall-clock time (470 on GPUs, 2,380 on CPUs, dominated by CatBoost-KGB with a median of 24 minutes per fit), and the tuning sweeps took about 3,150 hours (2,050 on GPUs, 1,100 on CPUs). These are sums of per-fit and per-sweep wall-clock times; the jobs ran in parallel.

## C.1 PROBABILISTIC REGRESSION BASELINES

**Bayesian linear regression (BLR).** The Bayesian linear regression baseline is a conjugate Gaussian linear model with an optional intercept term. It places an isotropic Gaussian prior on the regression weights,

$$w \sim \mathcal{N}(0, \alpha^{-1} I),$$

and assumes a homoscedastic Gaussian observation model with fixed noise standard deviation $\sigma$,

$$y \mid x, w \sim \mathcal{N}(x^\top w, \sigma^2).$$

The intercept, if used, is a constant column of the design matrix with the same prior as the other weights. Targets are standardized before fitting, so $\sigma$ is measured in units of the target standard

deviation and one search range serves all datasets. Given the design matrix $X$, the posterior over the weights is available in closed form,

$$S_N = \big(\alpha I + \beta X^\top X\big)^{-1}, \qquad m_N = \beta S_N X^\top y, \qquad \beta = \sigma^{-2},$$

up to a small diagonal jitter for numerical stability. The predictive distribution for a test input $x$ is then Gaussian with mean

$$\mu(x) = x^\top m_N$$

and variance

$$\sigma^2_{\text{pred}}(x) = \sigma^2 + x^\top S_N x.$$

The quadratic term $x^\top S_N x$ is interpreted as epistemic uncertainty, the fixed noise variance $\sigma^2$ as aleatoric uncertainty, and the total predictive uncertainty as their sum (MacKay, 1992). **Hyperparameter search space:** $\alpha \in [10^{-4}, 10]$ (log-uniform), $\sigma \in [10^{-2}, 2]$ (log-uniform), and `fit_intercept` $\in \{\text{True}, \text{False}\}$.

**Exact Gaussian process (GP).** Exact GP regression (Rasmussen & Williams, 2005) with a constant mean function and the kernel

$$k(x, x') = \sigma_m^2\, k_\nu\big(x, x'; \ell_1, \ldots, \ell_d\big) + \sigma_l^2\, x^\top x',$$

the sum of a Matérn kernel $k_\nu$ with one lengthscale per input dimension (automatic relevance determination) and a linear kernel. The same kernel family is used on every dataset, rather than the kernel that generated the data, so that the baseline does not inherit knowledge of the oracle. The Gaussian likelihood has a single, homoscedastic noise variance $\sigma_n^2$. Inputs and targets are standardized. The kernel hyperparameters, the constant mean and $\sigma_n^2$ are fitted jointly by maximizing the exact log marginal likelihood with AdamW (Loshchilov & Hutter, 2019), a learning-rate schedule with linear warm-up and cosine decay to zero, and gradient-norm clipping, implemented in GPJax (Pinder & Dodd, 2022). The predictive mean is the posterior mean, epistemic uncertainty is the posterior variance of the latent function, and aleatoric uncertainty is the learned noise variance $\sigma_n^2$, which is constant in $x$. **Hyperparameter search space:** `matern_nu` $\in \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$, `lengthscale` $\in [0.1, 10]$ (log-uniform; initial value of every ARD lengthscale), `matern_variance` $\in [0.1, 5]$ (log-uniform), `linear_variance` $\in [10^{-3}, 2]$ (log-uniform), `noise_std` $\in [0.02, 1]$ (log-uniform; initial noise standard deviation), `num_iters` $\in [150, 500]$ (integer), `warmup_steps` $\in [0, 100]$ (integer), `peak_learning_rate` $\in [10^{-3}, 5 \cdot 10^{-2}]$ (log-uniform), `gradient_clip` $\in [0.5, 2]$ (uniform), and `weight_decay` $\in [10^{-6}, 10^{-2}]$ (log-uniform).

**Variational latent Gaussian process (VLGP).** A heteroscedastic sparse variational GP with two latent processes (Lázaro-Gredilla & Titsias, 2011): $f$ for the regression function and $g(x) = \log \sigma^2(x)$ for the noise variance,

$$y \mid f, g \sim \mathcal{N}\big(f(x), e^{g(x)}\big).$$

The process $f$ uses the Matérn-plus-linear kernel and constant mean of the exact GP. The process $g$ has its own Matérn kernel, whose linear component is initialized at a negligible variance of $10^{-6}$, and a constant prior mean $g_0$. Each process has $M$ inducing inputs, placed at $k$-means centres of the training inputs and shared by both, with a free Gaussian variational distribution over the inducing values (Titsias, 2009; Hensman et al., 2013). All parameters are fitted by maximizing the heteroscedastic evidence lower bound of Lázaro-Gredilla & Titsias (2011) (as implemented in GPJax) with AdamW, a warm-up cosine schedule and gradient clipping. Two initialization choices keep the fit stable. The variational covariances start at the prior covariance, so the Kullback–Leibler term is zero at the start. The noise anchor $g_0$, which is also the initial variational mean of $g$, is a nearest-neighbour estimate of the noise level (Rice, 1984),

$$g_0 = \log\Big(\tfrac{1}{2n} \textstyle\sum_{i=1}^{n} \big(y_i - y_{\text{nn}(i)}\big)^2\Big),$$

where $\text{nn}(i)$ is the training point closest to $x_i$. Differencing neighbouring targets largely cancels the smooth signal, so unlike $\log \text{Var}(y)$ this estimate is not dominated by the signal variance. The predictive mean is the variational posterior mean of $f$. Epistemic uncertainty is the variational posterior variance of $f(x)$, and aleatoric uncertainty is the posterior mean of the noise variance,

$$\hat{\sigma}^2_{\text{ale}}(x) = \mathbb{E}_q\big[e^{g(x)}\big] = \exp\big(m_g(x) + \tfrac{1}{2} v_g(x)\big),$$

where $m_g(x)$ and $v_g(x)$ are the variational mean and variance of $g(x)$. Among the GP baselines, VLGP is the only one whose aleatoric estimate varies with $x$. **Hyperparameter search space:** `matern_nu` $\in \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$, `lengthscale` $\in [0.1, 10]$ (log-uniform), `matern_variance` $\in [0.1, 5]$ (log-uniform), `linear_variance` $\in [10^{-3}, 2]$ (log-uniform), `noise_matern_nu` $\in \{\frac{3}{2}, \frac{5}{2}\}$, `noise_lengthscale` $\in [0.5, 20]$ (log-uniform), `noise_kernel_variance` $\in [0.05, 2]$ (log-uniform), `n_inducing` $\in \{64, 128, 256\}$, `num_iters` $\in [1000, 4000]$ (integer), `warmup_steps` $\in [0, 200]$ (integer), `peak_learning_rate` $\in [2 \cdot 10^{-4}, 2 \cdot 10^{-3}]$ (log-uniform), `gradient_clip` $\in [0.5, 2]$ (uniform), and `weight_decay` $\in [10^{-6}, 10^{-2}]$ (log-uniform).

### C.2 Deep Kernel Learning and Distance-Aware Models

**Deep kernel learning (DKL).** DKL (Wilson et al., 2015) combines a neural feature extractor with a Gaussian-process output layer. A multilayer perceptron with `n_layers` hidden layers of width `hidden_dim`, ReLU activations and a linear last layer maps standardized inputs to features $\phi_\theta(x) \in \mathbb{R}^{d_\phi}$. An exact GP with zero mean and an ARD Matérn-$\frac{5}{2}$ kernel is placed on these features, with a Gaussian likelihood with one homoscedastic noise variance. Network weights, kernel hyperparameters and noise are trained jointly by maximizing the exact log marginal likelihood with AdamW, a warm-up cosine schedule and gradient clipping. The predictive mean is the GP posterior mean at $\phi_\theta(x)$, epistemic uncertainty is the posterior latent variance, and aleatoric uncertainty is the learned noise variance, which is constant in $x$. Targets are standardized, and all reported means and variances are transformed back to the original target scale. **Hyperparameter search space:** `n_layers` $\in \{1, 2, 3\}$, `hidden_dim` $\in \{16, 32, 64, 128\}$, `feature_space_dim` $\in [2, 8]$ (integer), `noise_std` $\in [0.05, 1.0]$ (log-uniform; initial value), `base_kernel_variance` $\in [0.1, 5.0]$ (log-uniform; initial value), `base_kernel_lengthscale` $\in [0.1, 5.0]$ (log-uniform; initial value), `num_iters` $\in [150, 500]$ (integer), `warmup_steps` $\in [0, 150]$ (integer), `peak_learning_rate` $\in [5 \cdot 10^{-4}, 5 \cdot 10^{-2}]$ (log-uniform), `end_learning_rate`$= 0$, `gradient_clip` $\in [0.5, 2.0]$ (uniform), `weight_decay` $\in [10^{-6}, 10^{-2}]$ (log-uniform), and `standardize_inputs=True`, `standardize_targets=True`.

**Heteroscedastic SNGP (SNGP-Het).** SNGP (Liu et al., 2020) uses the shared extractor $h(x)$ with a random-feature approximation of a GP output layer. With $M$ random Fourier features $\varphi(h) = \sqrt{2/M}\,\cos(\Omega h + b)$, where the rows of $\Omega$ are drawn from $\mathcal{N}(0, \ell^{-2} I)$ and $b$ uniformly from $[0, 2\pi]$ and both stay fixed (Rahimi & Recht, 2007), the predictive mean is $\mu(x) = \varphi(h(x))^\top \beta + b_0$. For regression we add a linear log-variance head on the same features, $\log \sigma^2(x) = h(x)^\top w_g + b_g$, initialized at $w_g = 0$ and $b_g$ equal to the nearest-neighbour noise estimate described for VLGP. Network and both heads are trained jointly by minimizing the Gaussian negative log-likelihood plus the ridge penalty $\frac{\lambda}{2n}\|\beta\|^2$. After training, the Laplace posterior over $\beta$ is computed once, as in SNGP; for a Gaussian likelihood with fixed $\sigma^2(x)$ it is exact,

$$\Sigma^{-1} = \lambda I + \sum_{i=1}^{n} \frac{\varphi_i \varphi_i^\top}{\sigma^2(x_i)}, \qquad \varphi_i = \varphi\big(h(x_i)\big).$$

Epistemic uncertainty is $\varphi(h(x))^\top \Sigma\, \varphi(h(x))$, and aleatoric uncertainty is the point estimate $\sigma^2(x)$. Heteroscedastic SNGP has been published only for classification (Fortuin et al., 2022); this regression version is the standard mean–variance construction on SNGP's extractor. Its variance head is not distance-aware and carries no uncertainty of its own, and the negative log-likelihood is not reweighted (e.g. by the $\beta$-NLL of Seitzer et al., 2022). **Hyperparameter search space:** `width`, `depth`, `coeff`, `dropout`, `lengthscale_factor`, `epochs`, `batch_size`, `learning_rate` and `weight_decay` as for DUE; `num_inducing` (number of random features $M$) $\in \{512, 1024, 2048\}$; and `ridge` ($\lambda$) $\in [10^{-3}, 10]$ (log-uniform).

### C.3 Tree-Based Models

**NGBoost with Normal-Inverse-Gamma outputs (NGBoost-NIG).** This model uses NGBoost (Duan et al., 2020) for evidential regression. The boosting model predicts the four parameters of a Normal-Inverse-Gamma (NIG) distribution,

$$\mu(x), \qquad \lambda(x) = e^{\theta_2(x)}, \qquad \alpha(x) = e^{\theta_3(x)} + 1, \qquad \beta(x) = e^{\theta_4(x)},$$

with regression trees of depth `max_depth` as base learners. It minimizes the evidential score

$$\mathcal{S}(x_i, y_i) = \mathcal{L}_{\mathrm{NIG}}(x_i, y_i) + \lambda_{\mathrm{evi}} \frac{1}{n} \sum_{j=1}^{n} |y_j - \mu_j| \, (2\alpha_j + \lambda_j) + \lambda_{\mathrm{KL}} \frac{1}{n} \sum_{j=1}^{n} \mathrm{KL}\big(\mathrm{NIG}(\mu_j, \lambda_j, \alpha_j, \beta_j) \,\|\, \mathrm{NIG}(0, 1, 2, 1)\big),$$

where $\mathcal{L}_{\mathrm{NIG}}$ is the NIG marginal negative log-likelihood (the same expression as for the evidential BNN below, with $v = \lambda$), the second term is the evidential regularizer of Amini et al. (2020), and the third pulls each prediction towards a weak reference NIG. Boosting uses ordinary gradients: the only Fisher information available for this score is a rank-one empirical estimate, whose inverse applied to the gradient is not a natural gradient. Targets are standardized. Uncertainty follows the NIG decomposition, with aleatoric uncertainty $\beta(x)/(\alpha(x)-1)$, epistemic uncertainty $\beta(x)/\big(\lambda(x)(\alpha(x)-1)\big)$, and total uncertainty their sum. **Hyperparameter search space:** `n_estimators` $\in [200, 800]$ (integer), `learning_rate` $\in [0.01, 0.2]$ (log-uniform), `max_depth` $\in [2, 6]$ (integer), `minibatch_frac` $\in [0.5, 1.0]$ (uniform), `col_sample` $\in [0.5, 1.0]$ (uniform), `natural_gradient=False`, `use_svgd=False`, `evid_strength` ($\lambda_{\mathrm{evi}}$) $\in [10^{-4}, 1]$ (log-uniform), `kl_strength` ($\lambda_{\mathrm{KL}}$) $\in [10^{-4}, 1]$ (log-uniform), and `standardize_y=True`. The space also contains `length_scale` $\in [0.1, 5.0]$, which only affects the SVGD variant and is therefore inactive.

**NGBoost bagging ensemble (NGBoost-Bag).** The NGBoost bagging baseline is an ensemble of `n_regressors` NGBoost regressors (Duan et al., 2020) with Gaussian predictive distributions, each trained under the Gaussian log score on its own resample of the training data. A resample has $\lfloor$`sample_fraction` $\cdot\, n\rfloor$ rows, drawn with or without replacement according to `replace` (Breiman, 1996). Each member predicts a Gaussian location and scale. The benchmark prediction is the average of the member means. Aleatoric uncertainty is the average member predictive variance, epistemic uncertainty is the variance across member means, and total uncertainty is the sum of both terms. Optionally, the target can be standardized before fitting, with all reported means and variances rescaled to the original target units. **Hyperparameter search space:** `n_estimators` $\in [200, 800]$ (integer), `learning_rate` $\in [0.005, 0.1]$ (log-uniform), `max_depth` $\in [2, 6]$ (integer), `minibatch_frac` $\in [0.5, 1.0]$ (uniform), `col_sample` $\in [0.5, 1.0]$ (uniform), `natural_gradient` $\in \{\mathrm{True}, \mathrm{False}\}$, `n_regressors` $\in [5, 25]$ (integer), `sample_fraction` $\in [0.6, 0.95]$ (uniform), `replace` $\in \{\mathrm{True}, \mathrm{False}\}$, and `standardize_y` $\in \{\mathrm{True}, \mathrm{False}\}$.

**CatBoost posterior sampling (CatBoost-PS).** This tree ensemble trains `n_regressors` CatBoost regressors (Prokhorenkova et al., 2019) with the `RMSEWithUncertainty` objective, under which each member predicts a mean and a variance. Each member uses CatBoost's posterior-sampling mode, which trains it by stochastic gradient Langevin boosting so that it approximates a sample from the posterior over models (Malinin et al., 2021), with a Bayesian bootstrap of strength `bagging_temperature` and its own random seed. If `bagging_frac` $< 1$, each member is in addition trained on a bootstrap sample of $\lfloor$`bagging_frac` $\cdot\, n\rfloor$ rows. The benchmark averages member means for prediction, averages the member-provided variances for aleatoric uncertainty, and uses the variance across member means for epistemic uncertainty. When target standardization is enabled, all means and variances are transformed back to the original target scale. **Hyperparameter search space:** `iterations` $\in [100, 600]$ (integer), `learning_rate` $\in [0.01, 0.3]$ (log-uniform), `depth` $\in [3, 8]$ (integer), `l2_leaf_reg` $\in [0.5, 10]$ (log-uniform), `random_strength` $\in [0, 2]$ (uniform), `bagging_temperature` $\in [0, 2]$ (uniform), `n_regressors` $\in [5, 20]$ (integer), `bagging_frac` $\in [0.6, 1.0]$ (uniform), and `standardize_y` $\in \{\mathrm{True}, \mathrm{False}\}$.

**CatBoost kernel gradient boosting (CatBoost-KGB).** The CatBoost KGB baseline uses CatBoost's `sample_gaussian_process` routine, which implements the kernel gradient boosting sampling procedure of Ustimenko et al. (2023). In this view, gradient boosting with symmetric trees is used to approximately sample functions from a Gaussian-process posterior under a prior of the form $f \sim \mathcal{GP}(0, \sigma^2 K + \delta^2 I)$. The wrapper draws `n_regressors` sampled tree functions and forms predictions by averaging them; epistemic uncertainty is the variance across the sampled functions. The routine returns function values only and no observation-noise estimate, so the aleatoric uncertainty is estimated from held-out residuals, as for DEUP: a second

ensemble with the same hyperparameters is trained on a random 90% of the training data, and the aleatoric variance is its mean squared error on the remaining 10%. This estimate is constant in $x$. The final ensemble is then trained on all training data. If target standardization is enabled, predictive means and variances are transformed back to the original target scale. We use the implementation available in the CatBoost library (Prokhorenkova et al., 2019). **Hyperparameter search space:** `posterior_iterations` $\in [200, 1000]$ (integer), `prior_iterations` $\in [50, 300]$ (integer), `learning_rate` $\in [0.03, 0.3]$ (log-uniform), `depth` $\in [4, 8]$ (integer), `sigma` $\in [0.05, 0.5]$ (log-uniform), `delta` $\in [0.01, 0.5]$ (log-uniform), `random_strength` $\in [0.05, 1.0]$ (log-uniform), `eps` $\in [10^{-5}, 10^{-3}]$ (log-uniform), `n_regressors` $\in [5, 20]$ (integer), and `standardize_y` $\in \{\text{True}, \text{False}\}$.

### C.4 Bayesian Neural Network Models

**Shared neural network setup.** All neural-network methods use the same residual tabular network, inspired by TabResNet (Gorishniy et al., 2021), implemented in JAX/Flax in single precision. Given an input $x$, the backbone first forms

$$h^{(0)} = \mathrm{Dropout}\big(\sigma(W_{\mathrm{in}}x + b_{\mathrm{in}})\big),$$

and then applies $L$ residual blocks of the form

$$u^{(\ell)} = \mathrm{Dropout}\big(\sigma(W_1^{(\ell)}h^{(\ell)}+b_1^{(\ell)})\big), \qquad h^{(\ell+1)} = \sigma\big(W_2^{(\ell)}u^{(\ell)}+b_2^{(\ell)}+h^{(\ell)}\big), \qquad \ell = 0, \dots, L-1,$$

with the ReLU activation $\sigma$, He initialization and no normalization layers. The dropout rate is zero except for MC dropout. The final representation is passed to a linear output head. Width and depth are set by `resnet_width` and `resnet_blocks`.

Inputs and targets are standardized during training. If $\tilde{y} = (y - m_y)/s_y$ denotes the standardized target, predictive quantities are transformed back to the original target scale via

$$\mu(x) = s_y\tilde{\mu}(x) + m_y, \qquad v(x) = s_y^2\tilde{v}(x),$$

and analogously for aleatoric, epistemic, and total variances.

Gaussian-output methods (all except EDL and DEUP) predict a mean and a raw scale parameter,

$$(\mu(x), \rho(x)), \qquad s(x) = \mathrm{softplus}(\rho(x)) + \varepsilon, \qquad v(x) = s(x)^2,$$

with $\varepsilon = 10^{-6}$, and are trained on the Gaussian negative log-likelihood plus a weak penalty on implausibly large variances,

$$\ell(y, x) = \frac{1}{2}\left(\log v(x) + \frac{(y - \mu(x))^2}{v(x)}\right) + \lambda_{\mathrm{var}}\left[\max\big\{0,\ \log v(x) - \log\big(20\,\mathrm{Var}(\tilde{y})\big)\big\}\right]^2,$$

up to the additive constant $\frac{1}{2}\log(2\pi)$, with $\lambda_{\mathrm{var}} = 10^{-3}$. At prediction time, member variances are clipped to $[\varepsilon,\ 50\,\mathrm{Var}(\tilde{y})]$. The implementation also supports the natural parameterization of the same Gaussian, $\mu = -\eta_1/(2\eta_2)$ and $v = -1/(2\eta_2)$. It is not used for any reported result: in it the mean is $\eta_1$ times the model's own variance, so every perturbation of the network that the ensemble-, dropout-, Laplace- and SWAG-type methods use to measure epistemic uncertainty moves the mean in proportion to the aleatoric estimate, which couples the two estimates.

All networks are trained for `n_epochs` epochs of shuffled minibatches with NAdamW (Dozat, 2015; Loshchilov & Hutter, 2019): a constant learning rate `lr`, first-moment coefficient $\beta_1 =$`momentum`, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, decoupled weight decay `weight_decay`, and gradients clipped to a global norm of 1. For ensembles the clipping is applied per member, and the loss is averaged over the batch but summed over members, so each member receives exactly the gradient it would receive if trained alone.

For methods with $M$ stochastic forward passes, ensemble members or posterior samples $\{(\mu_m(x), v_m(x))\}_{m=1}^{M}$, we report

$$\hat{\mu}(x) = \frac{1}{M}\sum_{m=1}^{M}\mu_m(x), \qquad \hat{v}_{\mathrm{ale}}(x) = \frac{1}{M}\sum_{m=1}^{M}v_m(x),$$

$$\hat{v}_{\mathrm{epi}}(x) = \mathrm{Var}_m(\mu_m(x)), \qquad \hat{v}_{\mathrm{tot}}(x) = \hat{v}_{\mathrm{ale}}(x) + \hat{v}_{\mathrm{epi}}(x).$$

The number of members $M$ is a different resource for different methods. For bagging and deep ensembles, a member is a full retraining, so $M = 10$. For the post-hoc samplers, a member is one forward pass, so $M$ is set high (128 for MC dropout and SWAG, 256 for the Laplace variants): the sample variance of $M$ draws carries multiplicative Monte Carlo noise that alone caps the achievable rank correlation, at roughly 0.7 for $M = 10$ and 0.97 for $M = 128$. **Shared hyperparameter search space:** for all neural-network methods we tuned `resnet_width`$\in \{32, 64, 128, 256, 512\}$, `resnet_blocks`$\in \{1, 2, 3, 4\}$, `n_epochs`$\in [50, 200]$ (integer), `batch_size`$\in \{64, 128, 256, 512\}$, `lr`$\in [10^{-4}, 5 \cdot 10^{-3}]$ (log-uniform), `weight_decay`$\in [10^{-6}, 10^{-2}]$ (log-uniform), and `momentum`$\in [0.7, 0.95]$ (uniform). The main benchmark tunes no optimizer-specific parameter. The optimizer spaces were: Adam with `adam_b2`$\in [0.95, 0.999]$ (uniform) and `adam_eps`$\in [10^{-9}, 10^{-6}]$ (log-uniform).

**BNN bagging (NN-Bag).** The bagging BNN uses the shared network with $M$ independently initialized members, trained on the Gaussian objective above. In contrast to the deep ensemble below, each member is trained on a fixed subsample of the training data of size $\lfloor$`sample_fraction` $\cdot\, n \rfloor$, drawn with or without replacement according to `replace`. The predictive mean is the average of the member means, aleatoric uncertainty is the average of the member predictive variances, and epistemic uncertainty is the variance across member means; total uncertainty is their sum (Breiman, 1996). **Additional hyperparameter search space:** `n_members`$= 10$, `sample_fraction`$\in [0.6, 0.95]$ (uniform), and `replace`$\in \{\mathrm{True}, \mathrm{False}\}$.

**Deep ensemble BNN (NN-Ens).** The deep ensemble uses the shared network with $M$ independently initialized members of identical architecture. All members are trained on the full training set with separate parameters and optimization trajectories, using the Gaussian objective above. The predictive mean is obtained by averaging member means, aleatoric uncertainty by averaging member predictive variances, and epistemic uncertainty by the variance across member means; total uncertainty is the sum of the aleatoric and epistemic terms (Lakshminarayanan et al., 2017). **Additional hyperparameter search space:** `n_members`$= 10$.

**MC dropout BNN (NN-Drop).** The MC-dropout BNN uses a single instance of the shared network, trained with dropout (after the input layer and after the first layer of every residual block) and the Gaussian objective above. At test time, dropout remains active and $M$ stochastic forward passes are drawn with different dropout masks. These passes are treated like ensemble members: the predictive mean is the average of the sampled means, aleatoric uncertainty is the average of the sampled predictive variances, and epistemic uncertainty is the variance across sampled means; total uncertainty is their sum (Gal & Ghahramani, 2016). **Additional hyperparameter search space:** `n_members`$= 128$ and `dropout`$\in [0.05, 0.35]$ (uniform).

**Evidential BNN (NN-EDL).** The evidential BNN uses the shared network with a four-dimensional evidential head that predicts the parameters of a Normal-Inverse-Gamma (NIG) distribution (Amini et al., 2020),

$$\mu(x), \qquad v(x) > 0, \qquad \alpha(x) > 1, \qquad \beta(x) > 0,$$

where the constraints are enforced through softplus transformations. The model is trained with the NIG marginal negative log-likelihood together with an evidential regularizer,

$$\mathcal{L}_{\mathrm{EDL}}(x, y) = \mathcal{L}_{\mathrm{NIG}}(x, y) + \lambda_{\mathrm{evi}}\, \mathcal{R}_{\mathrm{evi}}(x, y),$$

where $\lambda_{\mathrm{evi}}$ corresponds to `evi_reg`. The NIG marginal negative log-likelihood is

$$\mathcal{L}_{\mathrm{NIG}}(x, y) = \frac{1}{2}\big(\log \pi - \log v(x)\big) - \alpha(x) \log\big(2\beta(x)(1 + v(x))\big) \tag{13}$$

$$+ \Big(\alpha(x) + \frac{1}{2}\Big) \log\Big(v(x)\big(y - \mu(x)\big)^2 + 2\beta(x)(1 + v(x))\Big) \tag{14}$$

$$+ \log \Gamma\big(\alpha(x)\big) - \log \Gamma\Big(\alpha(x) + \frac{1}{2}\Big), \tag{15}$$

and the evidential regularizer is

$$\mathcal{R}_{\mathrm{evi}}(x, y) = |y - \mu(x)|\, \big(2v(x) + \alpha(x)\big).$$

At test time, the predictive mean is $\mu(x)$, aleatoric uncertainty is

$$\sigma^2_{\text{ale}}(x) = \frac{\beta(x)}{\alpha(x) - 1},$$

and epistemic uncertainty is

$$\sigma^2_{\text{epi}}(x) = \frac{\beta(x)}{v(x)(\alpha(x) - 1)},$$

with total uncertainty given by

$$\sigma^2_{\text{tot}}(x) = \sigma^2_{\text{ale}}(x) + \sigma^2_{\text{epi}}(x).$$

**Additional hyperparameter search space:** `evi_reg`$\in [10^{-3}, 3 \cdot 10^{-1}]$ (log-uniform).

**DEUP BNN (NN-DEUP).** The DEUP model follows Algorithm 1 of Lahlou et al. (2021) for a fixed training set and consists of two networks with the shared architecture. First, a mean regressor $m(x)$ is trained with squared-error loss on a random 80% of the training data. The held-out 20%, $\mathcal{V}$, gives a single global aleatoric variance,

$$\hat{\sigma}^2_{\text{ale}} = \frac{1}{|\mathcal{V}|} \sum_{(x_i, y_i) \in \mathcal{V}} \big(y_i - m(x_i)\big)^2,$$

since the benchmark provides no repeated measurements from which an input-dependent noise level could be estimated. Second, an error predictor $e(x)$ is trained under squared loss on the same held-out points to predict the out-of-sample loss $e_i = \big(y_i - m(x_i)\big)^2$, i.e. the pointwise risk of $m$. At prediction time, the first network provides the predictive mean, and epistemic uncertainty is the predicted risk minus the aleatoric level, $u(x) = e(x) - \hat{\sigma}^2_{\text{ale}}$, with $e(x)$ bounded above by $50\,\text{Var}(\tilde{y})$. Because $\hat{\sigma}^2_{\text{ale}}$ is the mean squared error, $u(x)$ is negative wherever the predicted risk is below average. Rather than flooring these values at zero, which would tie all of them, $u$ is shifted by one constant, $\hat{\sigma}^2_{\text{epi}}(x) = u(x) - \min\{0, \min_{x'} u(x')\}$, with the minimum taken over the evaluation points. This preserves the ranking of $u$ and changes only its level. Aleatoric uncertainty is the constant $\hat{\sigma}^2_{\text{ale}}$, and total uncertainty is the sum. The two stages are tuned in sequence: a sweep over the shared search space selects the mean network (with the error network at its defaults), and a second 30-trial sweep then selects the error network by its held-out mean squared error in predicting the loss, with the mean network fixed at the first sweep's winner. **Additional hyperparameter search space:** the first-stage regressor uses the shared search space above. The second-stage error network was tuned over `lr_excess`$\in [10^{-4}, 10^{-1}]$ (log-uniform), `weight_decay_excess`$\in [10^{-6}, 10^{-2}]$ (log-uniform), `n_epochs_excess`$\in [30, 200]$ (integer), and `batch_size_excess`$\in \{32, 64, 128, 256\}$.

**Laplace BNN (NN-Lap).** The Laplace BNN is a post-hoc method applied to a point-estimate network (Daxberger et al., 2021). We first train the shared Gaussian-output network on all training data to obtain a MAP solution. Following standard last-layer Laplace practice (Daxberger et al., 2021; Weber et al., 2025), we then approximate a Gaussian posterior over the weights and bias of the mean output only, $\omega = (w_\mu, b_\mu)$, while keeping the backbone and the variance output fixed at their MAP values. With last-layer features $\phi(x) = [h(x); 1]$, the posterior precision is the full generalized Gauss–Newton matrix, weighted by the MAP noise variance of each training point,

$$\Lambda = \tau I + \sum_{i=1}^{n} \frac{\phi(x_i)\phi(x_i)^\top}{v_{\text{MAP}}(x_i)},$$

which is exact for this linear-Gaussian last layer. The prior precision $\tau$ maximizes the Laplace approximation of the log marginal likelihood of the training data (MacKay, 1992; Immer et al., 2021) over a logarithmic grid of 100 values in $[10^{-4}, 10^4]$; the posterior is not tempered and no data are held out. The approximation is constructed with `laplax` (Weber et al., 2025). At prediction time we draw $M$ samples $\omega_m \sim \mathcal{N}(\omega_{\text{MAP}}, \Lambda^{-1})$ and propagate them through the fixed backbone. Epistemic uncertainty is the variance of the sampled means, a Monte Carlo estimate of $\phi(x)^\top \Lambda^{-1} \phi(x)$; aleatoric uncertainty is the MAP variance output; total uncertainty is their sum. **Additional hyperparameter search space:** `n_members`$= 256$.

**Laplace BNN with spectral normalization (NN-Lap-SN).** The same method on a spectrally normalized backbone: every dense layer of the backbone (input layer and both layers of every residual block) is constrained to $\|W\|_2 \leq$`sn_coeff`, as in the distance-aware feature extractor above, and the method is tuned under the constraint. The largest singular value is tracked by persistent power iteration during training and refined by additional power iterations at prediction time. **Additional hyperparameter search space:** `n_members`$= 256$, `spectral_norm=True` and `sn_coeff`$\in [0.5, 3]$ (log-uniform).

**Function-space prior Laplace BNN (NN-FSP).** FSP-Laplace (Cinquin et al., 2024) replaces the isotropic weight prior of the Laplace BNN with a Gaussian-process prior on the network's mean function. MAP training adds the corresponding function-space regularizer at a set $C$ of context points,

$$n \cdot \bar{\ell}_{\text{batch}} + \frac{1}{2}\big(\mu_\theta(C) - m(C)\big)^\top K_{CC}^{-1}\big(\mu_\theta(C) - m(C)\big),$$

where $\bar{\ell}_{\text{batch}}$ is the minibatch average of the Gaussian objective above, the prior mean is $m = 0$, and the prior kernel on standardized inputs is

$$k(x, x') = \sigma_k^2\big(k_{3/2}(r) + \tilde{x}^\top \tilde{x}'\big), \qquad \tilde{x} = x/\ell_k, \quad r = \|\tilde{x} - \tilde{x}'\|,$$

a Matérn-$\frac{3}{2}$ plus linear kernel with $\sigma_k = \ell_k = 1$. During training, $C$ consists of 64 training inputs resampled at every step. After training, the backbone and the variance output are fixed at their MAP values, and a Gaussian posterior over the weights and bias of the mean output is constructed with the low-rank function-space Laplace approximation of Cinquin et al. (2024, Algorithm 2). The prior term uses 256 context points from a Halton sequence covering the bounding box of the training inputs, with at most 128 eigendirections of $K_{CC}$. The data term is $\sum_i J(x_i)^\top J(x_i)/v_{\text{MAP}}(x_i)$ over all training points, where $J$ is the Jacobian of the mean output with respect to these parameters. Leading posterior directions are then dropped until the posterior variance at the context points no longer exceeds the prior variance. The posterior is not tempered and no data are held out. At prediction time, $M$ weight samples give epistemic uncertainty as the variance of the sampled means, while the MAP variance output provides the aleatoric uncertainty. **Additional hyperparameter search space:** `n_members`$= 256$.

**Function-space prior Laplace BNN with spectral normalization (NN-FSP-SN).** FSP-Laplace on the spectrally normalized backbone described for NN-Lap-SN, tuned under the constraint. **Additional hyperparameter search space:** `n_members`$= 256$, `spectral_norm=True` and `sn_coeff`$\in [0.5, 3]$ (log-uniform).

**SWAG BNN (NN-SWAG).** The SWAG BNN (Maddox et al., 2019) first removes a random calibration split $\mathcal{C}$ of relative size `calibration_fraction` from the training data and trains the shared Gaussian-output network on the rest, as above. Starting from this solution, it then enters a collection phase in which training continues along a single SGD trajectory with constant learning rate `swag_lr`, momentum `swag_momentum` and no weight decay. After every `epoch_per_checkpoint` epochs, for `n_checkpoints` checkpoints, the first and second parameter moments $E[\theta]$ and $E[\theta^2]$ are updated online; the pretrained solution enters as the first point. These moments define a diagonal Gaussian posterior approximation over all network parameters,

$$\sigma^2_{\text{SWAG}} = E[\theta^2] - E[\theta]^2, \qquad \theta \sim \mathcal{N}\big(\mu_{\text{SWAG}}, \text{diag}(\sigma^2_{\text{SWAG}})\big)\,,$$

where $\mu_{\text{SWAG}} = E[\theta]$. At prediction time, $M$ networks are sampled from this posterior and evaluated with deterministic forward passes. The predictive mean is the average of the sampled means, aleatoric uncertainty is the average sampled variance output, and epistemic uncertainty is the variance across sampled means. If $\mathcal{C}$ is non-empty, a single global variance scale is fitted on it,

$$c = \frac{\frac{1}{|\mathcal{C}|}\sum_{(x_i,y_i)\in\mathcal{C}}\big(y_i - \hat{\mu}(x_i)\big)^2}{\frac{1}{|\mathcal{C}|}\sum_{(x_i,y_i)\in\mathcal{C}}\hat{\sigma}^2_{\text{tot}}(x_i)},$$

clipped to $[10^{-3}, 10^3]$, and both aleatoric and epistemic variances are multiplied by $c$ at test time; otherwise $c = 1$. The scale does not change any ranking. **Additional hyperparameter search space:** `n_members`$= 128$, `n_checkpoints`$\in [10, 40]$ (integer), `epoch_per_checkpoint`$\in [2, 12]$ (integer), `swag_lr`$\in [10^{-3}, 10^{-1}]$ (log-uniform), `swag_momentum`$\in \{0.0, 0.9, 0.95\}$, `calibration_fraction`$\in [0, 0.25]$ (uniform), and `grad_clip_norm`$\in [0.5, 2.0]$ (uniform), which replaces the default clipping norm of 1 in both phases.

# D SUPPLEMENTARY RESULTS

## D.1 STABILITY OF MODEL RANKINGS ACROSS TASKS

The method ranking depends little on the experimental condition (Table 7): across the levels of each factor, Kendall's $W$ is at least 0.81 for the epistemic and 0.93 for the aleatoric criterion. The least stable factor is the oracle engine ($W = 0.81$ on ranks), so we check whether methods rank better when the ground truth is drawn from a prior like their own (Figure 7). This matched-prior advantage is real: GP-based methods and NNs gain about one rank at their matched oracle (mean rank 4.0 vs. 4.6 and 9.1 vs. 9.8), and trees about four ranks under HBART (9.0 vs. 12.8). The top three (GP, VLGP, SNGP-Het) and the bottom two (NN-DEUP, NN-EDL) are the same under every oracle. The rankings of methods on EU defined as posterior risk largely agree between datasets as well (Kendal's W=078). We showcase the results for representative datasets (abalone, airfoil) in Figure 8 and Figure 9.

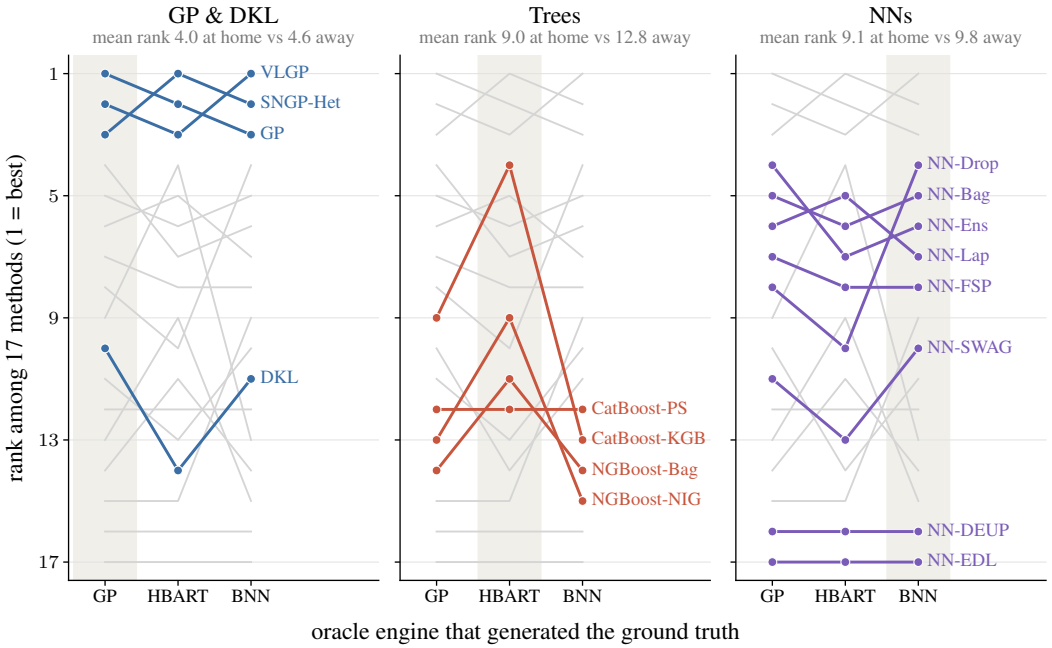



Figure 7: Rank of each method under each oracle engine, by its mean Spearman $\rho$ between EU output and posterior excess risk (1 = best of 17). One panel per family. The other methods are shown in grey, and the shaded column marks the oracle whose prior matches the family (GP for GP & DKL, HBART for trees, BNN for NNs). BLR has no matching oracle and appears only in grey.

Table 7: Stability of the method rankings across the levels of each experimental factor, measured by Kendall's coefficient of concordance $W$ (tie-corrected; $W = 1$: every level ranks the methods identically, $W = 0$: no agreement). Each level contributes one ranking of the methods, from their mean rank within the conditions of that level (ranks) or from their mean Spearman $\rho$ over them ($\rho$). Epistemic: EU output against the posterior excess risk; aleatoric: AU output against the true noise variance, for the methods with a heteroscedastic AU and the three noise modes whose noise varies.

| Criterion | Factor | Levels | Methods | $W$ (ranks) | $W$ ($\rho$) |
|---|---|---|---|---|---|
| Epistemic | Oracle engine | 3 | 19 | 0.81 | 0.92 |
| | Noise mode | 4 | 19 | 0.94 | 0.97 |
| | Covariate coverage | 2 | 19 | 0.98 | 0.99 |
| Aleatoric | Oracle engine | 3 | 14 | 0.93 | 0.94 |
| | Noise mode | 3 | 14 | 0.96 | 0.99 |
| | Covariate coverage | 2 | 14 | 0.99 | 1.00 |

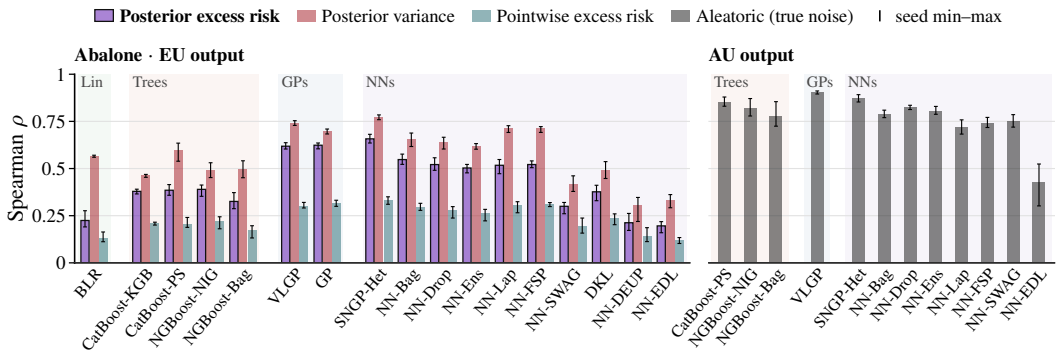



Figure 8: The complementary figure to Figure 2 of the main text, but reduced to only one dataset (abalone)

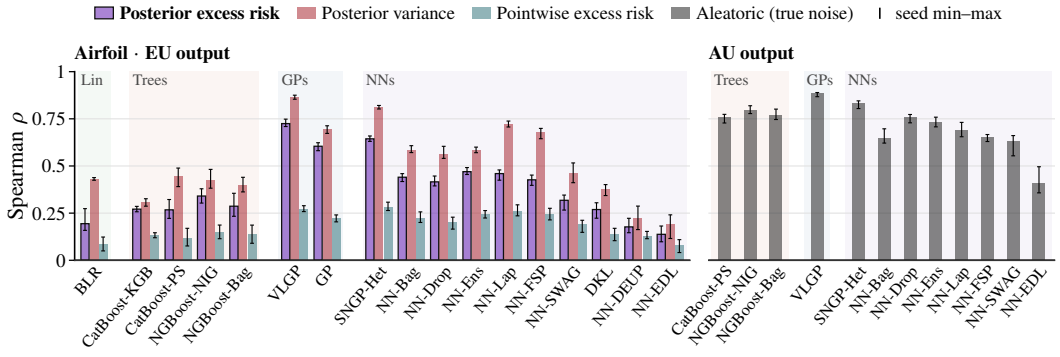



Figure 9: The complementary figure to Figure 2 of the main text, but reduced to only one dataset (airfoil)

### D.2 Disentanglement

This subsection reports the full excess-entanglement results underlying Figure 3a. We compare each method's EU–AU coupling with the corresponding ground-truth coupling under posterior excess risk and posterior variance, both overall and across oracle engines and coverage settings. We then examine the pronounced coupling of NIG-based evidential methods and present possible explanations for the additional coupling observed with natural-parameter heads and VLGP.

**Normal–Inverse-Gamma evidential objectives.** The excess entanglement of NIG-based evidential methods follows from properties previously analyzed by Amini et al. (2020), Meinert et al. (2023), and Jürgens et al. (2024). We summarize these results to interpret the coupling observed in our benchmark. For an NIG distribution with parameters $(\gamma, \nu, \alpha, \beta)$, the commonly used uncertainty estimates are

$$\widehat{\mathrm{AU}}(x) = \frac{\beta(x)}{\alpha(x) - 1}, \qquad \widehat{\mathrm{EU}}(x) = \frac{\beta(x)}{\nu(x)[\alpha(x) - 1]} = \frac{\widehat{\mathrm{AU}}(x)}{\nu(x)} \tag{16}$$

Both estimates therefore contain the same input-dependent scale $c(x) = \beta(x)/[\alpha(x) - 1]$. Unless $\nu(x)$ varies sufficiently to alter the ranking induced by $c(x)$, AU and EU inherit a strong positive association.

This coupling is reinforced by the non-identifiability of the NIG decomposition. Marginalizing the NIG distribution produces a Student-$t$ predictive distribution with location $\gamma$, $2\alpha$ degrees of freedom, and scale

$$s^2 = \frac{\beta(1 + \nu)}{\alpha\nu}.$$

Table 8: Excess disentanglement per method with the ground-truth EU defined as the posterior excess risk, $\mathrm{EU}^\star = \mathbb{E}_{\mathrm{post}}\big[(\hat{a}_{\mathcal{S}}(x) - f(x))^2\big]$. $\Delta\rho = \rho(\mathrm{EU}, \mathrm{AU}) - \rho(\mathrm{EU}^\star, \mathrm{AU}^\star)$ (Spearman, over the evaluation points), averaged over all conditions, and separately per oracle engine and per coverage scenario. $\Delta\rho > 0$: EU and AU outputs more strongly coupled than the truth.

| | | | | | $\Delta\rho$ by oracle engine | | | $\Delta\rho$ by coverage | |
|---|---|---|---|---|---|---|---|---|---|
| | Method | $\rho(\mathrm{EU}, \mathrm{AU})$ | $\rho(\mathrm{EU}^\star, \mathrm{AU}^\star)$ | $\Delta\rho$ | GP | HBART | BNN | bal. | OOD |
| Trees | CatBoost-PS | 0.52 | 0.14 | **+0.38** | +0.42 | +0.35 | +0.39 | +0.37 | +0.40 |
| | NGBoost-NIG | 1.00 | 0.29 | **+0.71** | +0.77 | +0.64 | +0.72 | +0.68 | +0.74 |
| | NGBoost-Bag | 0.33 | 0.23 | **+0.10** | +0.13 | +0.03 | +0.14 | +0.07 | +0.12 |
| GPs | VLGP | 0.51 | 0.31 | **+0.21** | +0.23 | +0.30 | +0.09 | +0.20 | +0.21 |
| NNs | SNGP-Het | 0.46 | 0.29 | **+0.17** | +0.18 | +0.21 | +0.13 | +0.20 | +0.14 |
| | NN-Bag | 0.35 | 0.26 | **+0.09** | +0.09 | +0.16 | +0.02 | +0.07 | +0.11 |
| | NN-Drop | 0.50 | 0.26 | **+0.24** | +0.23 | +0.29 | +0.19 | +0.22 | +0.25 |
| | NN-Ens | 0.36 | 0.26 | **+0.10** | +0.09 | +0.14 | +0.05 | +0.08 | +0.11 |
| | NN-Lap | 0.40 | 0.21 | **+0.20** | +0.18 | +0.22 | +0.20 | +0.24 | +0.16 |
| | NN-FSP | 0.42 | 0.20 | **+0.22** | +0.21 | +0.24 | +0.21 | +0.24 | +0.19 |
| | NN-SWAG | 0.57 | 0.22 | **+0.34** | +0.31 | +0.44 | +0.29 | +0.29 | +0.39 |
| | NN-EDL | 0.57 | 0.21 | **+0.35** | +0.30 | +0.34 | +0.42 | +0.35 | +0.36 |

Table 9: Excess disentanglement per method with the ground-truth EU defined as the posterior variance, $\mathrm{EU}^\star = \mathrm{Var}_{\mathrm{post}}\, f(x)$. $\Delta\rho = \rho(\mathrm{EU}, \mathrm{AU}) - \rho(\mathrm{EU}^\star, \mathrm{AU}^\star)$ (Spearman, over the evaluation points), averaged over all conditions, and separately per oracle engine and per coverage scenario. $\Delta\rho > 0$: EU and AU outputs more strongly coupled than the truth.

| | | | | | $\Delta\rho$ by oracle engine | | | $\Delta\rho$ by coverage | |
|---|---|---|---|---|---|---|---|---|---|
| | Method | $\rho(\mathrm{EU}, \mathrm{AU})$ | $\rho(\mathrm{EU}^\star, \mathrm{AU}^\star)$ | $\Delta\rho$ | GP | HBART | BNN | bal. | OOD |
| Trees | CatBoost-PS | 0.52 | 0.42 | **+0.10** | +0.19 | +0.03 | +0.09 | +0.04 | +0.16 |
| | NGBoost-NIG | 1.00 | 0.42 | **+0.58** | +0.65 | +0.54 | +0.53 | +0.52 | +0.63 |
| | NGBoost-Bag | 0.33 | 0.42 | **−0.09** | −0.04 | −0.15 | −0.08 | −0.15 | −0.03 |
| GPs | VLGP | 0.51 | 0.42 | **+0.09** | +0.16 | +0.04 | +0.07 | +0.06 | +0.12 |
| NNs | SNGP-Het | 0.46 | 0.42 | **+0.04** | +0.09 | −0.04 | +0.06 | +0.05 | +0.03 |
| | NN-Bag | 0.35 | 0.42 | **−0.07** | −0.02 | −0.11 | −0.10 | −0.12 | −0.03 |
| | NN-Drop | 0.50 | 0.42 | **+0.08** | +0.13 | +0.03 | +0.07 | +0.03 | +0.12 |
| | NN-Ens | 0.36 | 0.42 | **−0.07** | 0.00 | −0.12 | −0.08 | −0.11 | −0.02 |
| | NN-Lap | 0.40 | 0.42 | **−0.02** | +0.04 | −0.07 | −0.02 | −0.01 | −0.02 |
| | NN-FSP | 0.42 | 0.42 | **0.00** | +0.06 | −0.06 | −0.01 | −0.01 | +0.01 |
| | NN-SWAG | 0.57 | 0.42 | **+0.15** | +0.20 | +0.15 | +0.09 | +0.06 | +0.23 |
| | NN-EDL | 0.57 | 0.42 | **+0.15** | +0.17 | +0.06 | +0.21 | +0.10 | +0.19 |

For fixed $(\gamma, \alpha, s^2)$, every $\nu > 0$ yields the same predictive distribution when $\beta = s^2\alpha\nu/(1+\nu)$. Along this set of equivalent parameterizations,

$$\widehat{\mathrm{AU}} = s^2 \frac{\alpha}{\alpha - 1} \frac{\nu}{1+\nu}, \qquad \widehat{\mathrm{EU}} = s^2 \frac{\alpha}{\alpha - 1} \frac{1}{1+\nu},$$

while

$$\widehat{\mathrm{AU}} + \widehat{\mathrm{EU}} = s^2 \frac{\alpha}{\alpha - 1}$$

remains unchanged. The marginal likelihood can therefore identify the predictive distribution without uniquely identifying its allocation into AU and EU. This non-injectivity of the NIG predictive mapping and the resulting non-identifiability of its second-order parameters were established by Jürgens et al. (2024).

The evidential regularizer introduces an additional coupling mechanism. The objective proposed by Amini et al. (2020) contains

$$R_{\mathrm{evi}}(x, y) = \lambda |y - \gamma(x)| [2\nu(x) + \alpha(x)].$$

At the correct mean under Gaussian noise,

$$\mathbb{E}[|Y - \gamma(x)| \mid x] = \sigma(x)\sqrt{\frac{2}{\pi}}.$$

The expected contribution of the regularizer to the gradient with respect to $\nu$ is therefore

$$\mathbb{E}\left[\frac{\partial R_{\text{evi}}}{\partial \nu} \,\middle|\, x\right] = 2\lambda\sigma(x)\sqrt{\frac{2}{\pi}}.$$

All else fixed, the regularizer thus places stronger pressure toward smaller $\nu$ in regions with larger aleatoric noise. Because $\widehat{\mathrm{EU}} = \widehat{\mathrm{AU}}/\nu$, this transfers variation in AU into the epistemic estimate. Meinert et al. (2023) previously derived and demonstrated this residual-induced leakage under heteroscedastic noise.

These established properties account for the pronounced excess entanglement of NN-EDL and NGBoost-NIG. Equation (16) does not by itself imply perfect rank correlation, since variation in $\nu(x)$ can change the ordering of EU relative to AU. The observed correlation of $\rho = 1.00$ for NGBoost-NIG therefore additionally indicates that $\nu(x)$ does not substantially alter the ranking induced by the shared scale $c(x)$.

**Possible source of excess entanglement: natural-parameter heads.** The standard and natural-parameter heads represent the same pointwise Gaussian family. Their different coupling cannot therefore be attributed to a different population-optimal conditional distribution. Natural parameterization instead changes the coordinates in which the finite network is optimized and in which variation across ensemble members or stochastic predictions is expressed (Immer et al., 2023). We consider the following algebraic argument as a post hoc explanation of the observed results, not as a general consequence of natural-parameter training.

For stochastic prediction $m$, the natural parameters satisfy

$$v_m(x) = -\frac{1}{2\eta_{2m}(x)}, \qquad \mu_m(x) = -\frac{\eta_{1m}(x)}{2\eta_{2m}(x)} = \eta_{1m}(x)v_m(x).$$

The resulting estimates are

$$\widehat{\mathrm{AU}}(x) = \mathbb{E}_m[v_m(x)], \qquad \widehat{\mathrm{EU}}(x) = \mathrm{Var}_m[\eta_{1m}(x)v_m(x)].$$

Writing $\bar{\eta}_1(x) = \mathbb{E}_m[\eta_{1m}(x)]$ and $\bar{v}(x) = \mathbb{E}_m[v_m(x)]$, a first-order expansion gives

$$\mathrm{Var}_m[\mu_m(x)] \approx \bar{v}(x)^2\,\mathrm{Var}_m[\eta_{1m}(x)] + \bar{\eta}_1(x)^2\,\mathrm{Var}_m[v_m(x)] + 2\bar{\eta}_1(x)\bar{v}(x)\,\mathrm{Cov}_m[\eta_{1m}(x), v_m(x)].$$

If the stochastic predictions approximately agree on $v_m(x)$, the last two terms are small and

$$\widehat{\mathrm{EU}}(x) \approx \widehat{\mathrm{AU}}(x)^2\,\mathrm{Var}_m[\eta_{1m}(x)].$$

Under the additional condition that $\mathrm{Var}_m[\eta_{1m}(x)]$ varies more slowly across inputs than $\widehat{\mathrm{AU}}(x)$, disagreement in the first natural parameter is amplified in regions with large predicted AU. By contrast, the standard head outputs $\mu_m(x)$ directly and does not impose this multiplicative dependence on $v_m(x)$.

The optimization geometry provides a second possible contribution. For Gaussian NLL, the mean gradient in standard coordinates is

$$\frac{\partial \ell}{\partial \mu} = \frac{\mu - y}{v},$$

whereas the corresponding gradient in natural coordinates is

$$\frac{\partial \ell}{\partial \eta_1} = \mu - y.$$

Natural coordinates therefore remove the explicit precision weighting from the first-output gradient (Immer et al., 2023). Combined with $\mu = \eta_1 v$, this may allow differences between independently trained members to produce greater mean disagreement in high-variance regions.

This explanation is consistent with the increased coupling observed for bagging and deep ensembles, but it is not supported uniformly across stochastic methods. Natural parameterization does not increase coupling for dropout or SWAG, showing that the covariance terms and the source of stochastic variation can offset the proposed amplification. Establishing this mechanism would require decomposing the three terms above from the member-level outputs or intervening on the variance head. We therefore treat it only as a possible explanation of the observed pattern.

**Possible source of excess entanglement: posterior uncertainty in the VLGP noise process.** VLGP represents the conditional mean and heteroscedastic noise through two latent functions. The signal process $f$ models the conditional mean, while the noise process $g = \log \sigma^2$ models the input-dependent observation variance. Both functions are inferred jointly using sparse variational GPs under

$$Y \mid f, g, x \sim \mathcal{N}(f(x), \exp(g(x))),$$

as in heteroscedastic GP regression (Lázaro-Gredilla & Titsias, 2011). Consequently, the model is uncertain not only about the signal function but also about the noise function. However, these two sources of model uncertainty enter the reported AU and EU asymmetrically. Writing

$$q(f(x)) = \mathcal{N}(m_f(x), v_f(x)), \qquad q(g(x)) = \mathcal{N}(m_g(x), v_g(x)),$$

VLGP reports

$$\widehat{\mathrm{EU}}(x) = v_f(x), \qquad \widehat{\mathrm{AU}}(x) = \mathbb{E}_q[\exp(g(x))] = \exp\left(m_g(x) + \frac{v_g(x)}{2}\right).$$

The uncertainty about $f$ is therefore reported directly as EU, whereas uncertainty about $g$ changes the reported AU through the posterior expectation of the noise variance. This decomposition is internally coherent because the law of total variance gives

$$\mathrm{Var}_q(Y \mid x) = \mathbb{E}_q[\exp(g(x))] + \mathrm{Var}_q[f(x)].$$

It nevertheless differs from the oracle used in our benchmark. The oracle AU is computed from the realized noise law, whereas VLGP must infer that law and marginalizes over its method-side uncertainty.

We consider the following argument a post hoc explanation of the observed excess entanglement rather than an established property of VLGP. Its AU estimate factorizes as

$$\widehat{\mathrm{AU}}(x) = \underbrace{\exp(m_g(x))}_{\text{plug-in noise variance}} \underbrace{\exp(v_g(x)/2)}_{\text{noise-posterior factor}}.$$

The second factor is the log-normal correction arising from uncertainty about the noise process. At fixed $m_g(x)$, it increases the reported AU relative to the plug-in estimate $\exp(m_g(x))$. Since the exponential function is strictly increasing,

$$\rho_{\mathrm{S}}\left(\widehat{\mathrm{EU}}, \widehat{\mathrm{AU}}\right) = \rho_{\mathrm{S}}\left(v_f, m_g + \frac{v_g}{2}\right).$$

Thus, $v_g$ can increase EU–AU coupling if it is positively associated with $v_f$ across inputs. Such an association is plausible in our implementation because the two latent GPs are inferred from the same observations and use the same inducing locations. In regions with limited data coverage, both the signal and the noise function may consequently be weakly constrained, producing large values of both $v_f$ and $v_g$. The term $v_g/2$ would then introduce into AU part of the same coverage-dependent pattern represented by the signal posterior variance.

This argument does not establish that $v_f$ and $v_g$ must be positively associated. The two processes use different kernels, the posterior mean $m_g$ may dominate the AU ranking, and their joint optimization may weaken or reverse the proposed association. Moreover, some correlation between AU and $v_f$ is appropriate because observations with larger noise constrain the signal function less strongly. The proposed explanation concerns only the additional coupling that may arise because the method-side uncertainty $v_g$ enters the reported AU.

The hypothesis can be evaluated without retraining by replacing the reported AU with the plug-in estimate

$$\widehat{\mathrm{AU}}_{\mathrm{plug}}(x) = \exp(m_g(x))$$

and recomputing excess entanglement. Reporting $\rho_{\mathrm{S}}(v_f, v_g)$ would test the required association directly. A reduction in excess entanglement after removing the factor $\exp(v_g/2)$ would support the proposed explanation. Without this intervention, uncertainty in the noise process remains a plausible but unverified source of VLGP's excess entanglement.

Table 10: Excess disentanglement per method with the ground-truth EU defined as the pointwise excess risk, $\mathrm{EU}^\star = (\hat{a}_{\mathcal{S}}(x) - f^\star(x))^2$. $\Delta\rho = \rho(\mathrm{EU}, \mathrm{AU}) - \rho(\mathrm{EU}^\star, \mathrm{AU}^\star)$ (Spearman, over the evaluation points), averaged over all conditions, and separately per oracle engine and per coverage scenario. $\Delta\rho > 0$: EU and AU outputs more strongly coupled than the truth.

| | | | | | $\Delta\rho$ by oracle engine | | | $\Delta\rho$ by coverage | |
|---|---|---|---|---|---|---|---|---|---|
| | Method | $\rho(\mathrm{EU}, \mathrm{AU})$ | $\rho(\mathrm{EU}^\star, \mathrm{AU}^\star)$ | $\Delta\rho$ | GP | HBART | BNN | bal. | OOD |
| Trees | CatBoost-PS | 0.52 | 0.05 | **+0.48** | +0.51 | +0.45 | +0.47 | +0.47 | +0.48 |
| | NGBoost-NIG | 1.00 | 0.15 | **+0.85** | +0.90 | +0.82 | +0.82 | +0.84 | +0.85 |
| | NGBoost-Bag | 0.33 | 0.13 | **+0.20** | +0.23 | +0.16 | +0.22 | +0.19 | +0.22 |
| GPs | VLGP | 0.51 | 0.13 | **+0.38** | +0.41 | +0.41 | +0.32 | +0.40 | +0.36 |
| NNs | SNGP-Het | 0.46 | 0.14 | **+0.32** | +0.34 | +0.31 | +0.31 | +0.38 | +0.27 |
| | NN-Bag | 0.35 | 0.14 | **+0.21** | +0.23 | +0.25 | +0.15 | +0.21 | +0.21 |
| | NN-Drop | 0.50 | 0.13 | **+0.37** | +0.38 | +0.39 | +0.33 | +0.37 | +0.36 |
| | NN-Ens | 0.36 | 0.14 | **+0.22** | +0.24 | +0.23 | +0.17 | +0.22 | +0.22 |
| | NN-Lap | 0.40 | 0.12 | **+0.28** | +0.29 | +0.29 | +0.27 | +0.33 | +0.23 |
| | NN-FSP | 0.42 | 0.12 | **+0.30** | +0.32 | +0.31 | +0.28 | +0.34 | +0.27 |
| | NN-SWAG | 0.57 | 0.12 | **+0.44** | +0.45 | +0.52 | +0.37 | +0.40 | +0.48 |
| | NN-EDL | 0.57 | 0.13 | **+0.44** | +0.41 | +0.42 | +0.49 | +0.45 | +0.43 |

### D.3 OOD Analysis

A common way to evaluate epistemic uncertainty without a ground truth is out-of-distribution (OOD) detection. We use a natural covariate shift, white vs. red wine, to test how such an OOD score relates to our benchmark ranking, and whether the answer depends on the kind of shift.

**Experimental setup.** We reuse the benchmark's white-wine instances with balanced covariates and smooth-monotone heteroscedastic noise, for the three oracle engines (GP, HBART, BNN). Every method is fitted exactly as in the benchmark (same tuned configuration and training seed) and queried on two sets of 1,500 points: held-out white wines, and red wines, drawn from the 1,599 rows of the red-wine data (OpenML 44972) (OpenML, 2022f), which has the same eleven physicochemical features.

**Evaluation.** The OOD score is the method's EU output. Its AUROC for red (positive) vs. held-out white (negative) is computed per instance and averaged over the 15 instances (three oracle engines times 5 seeds), with $95\%$ intervals from a bootstrap over instances. We evaluate two versions of the task. *All points* uses both full sets. *Distance-matched* removes the distance shortcut: for each instance we fit a Mahalanobis distance to the white training inputs, pool the distances of the white and red query points, cut them into 20 bins at the pooled quantiles, and keep in every bin the same number of white and red points (the smaller of the two counts, drawn at random). Both classes then have the same distance distribution; on average 134 points per class remain, the white wines far from the training centre and the red wines closest to it. A method whose EU only measures distance from the training data scores $\approx 0.5$ on this version.

Three reference scores bracket the methods: the Mahalanobis distance itself, the mean distance to the 10 nearest training inputs, and a supervised gradient-boosting classifier trained on red vs. white labels (5-fold cross-validated), which bounds how separable the two sets are.

**Analysis.** We compare the OOD ranking of the 19 methods (mean EU-AUROC) with the benchmark ranking (mean Spearman $\rho$ between EU and the ground-truth target over all regression runs) using Kendall's coefficient of concordance $W$ (tie-corrected). $95\%$ intervals come from a bootstrap over the wine instances with the benchmark ranking held fixed. We repeat the comparison for the three definitions of the EU target (posterior excess risk, posterior variance, pointwise excess risk).

To check whether the distance-matched red wines are genuinely more uncertain, we evaluate the ground truth directly on the GP-oracle instances, where the posterior is available in closed form. We compare the ratio of the median EU on matched red and matched white wines between each method and the true posterior variance.

**Results.** On all points the task is easy: Mahalanobis distance alone reaches AUROC $0.982$ and the supervised classifier $0.997$ (Table 11). The best methods (NN-Lap-SN, GP, VLGP, BLR, SNGP-Het, NN-FSP-SN; $0.976$–$0.982$) match but do not exceed distance, and the ranking agrees closely with the benchmark: $W = 0.89$ $[0.85, 0.91]$ for the posterior excess risk, $0.92$ $[0.91, 0.93]$ for the posterior variance and $0.84$ $[0.81, 0.85]$ for the pointwise excess risk (Table 12). The methods ranked last are those whose EU is coupled to their AU (NN-EDL $0.48$, NN-DEUP $0.59$, NGBoost-NIG $0.67$).

After distance matching, all methods fall far below the separability bound ($0.954$). Only NN-Ens ($0.666$ $[0.645, 0.687]$) and NN-Bag ($0.639$ $[0.619, 0.655]$) clearly exceed both distance references (Mahalanobis $0.537$, 10-NN $0.567$); NN-Lap-SN ($0.580$) and CatBoost-KGB ($0.572$) exceed Mahalanobis but not 10-NN, and VLGP ($0.534$) and SNGP-Het ($0.547$) do not exceed either. The ranking decouples from the benchmark ($W = 0.64$ $[0.52, 0.75]$, $0.49$ $[0.37, 0.63]$ and $0.53$ $[0.42, 0.65]$ for the three targets). The ensembles' advantage holds under all three oracles (NN-Ens $0.689$, $0.645$, $0.664$ for BNN, GP and HBART), whereas the exact GP drops to $0.468$ under the BNN oracle.

The ground truth confirms that the matched red wines are more uncertain although they are not farther from the training data: under the GP oracle the true posterior variance is $1.36\times$ higher on matched red than on matched white wines ($1.22$–$1.60$ across instances) and separates the two with AUROC $0.643$ ($0.986$ on all points). The ensembles reproduce this shift (median EU ratio $1.49$ for NN-Ens, $1.30$ for NN-Bag), as do the exact GP ($1.27$) and CatBoost-KGB ($1.24$), whereas VLGP ($1.05$), NN-Lap-SN ($1.04$), SNGP-Het ($0.85$) and NN-FSP-SN ($0.71$) barely change or rate red wine as less uncertain. At the same time, VLGP, the GP, NN-Lap-SN and SNGP-Het track the true posterior variance beyond distance more closely than the ensembles (partial Spearman correlation given the Mahalanobis distance $0.90$, $0.84$, $0.77$ and $0.76$ vs. $0.37$ for NN-Ens and $0.43$ for NN-Bag): they reproduce the posterior's variation within the white-wine data, but not its rise at the red wines. The matched red wines differ from white wine in feature combinations rather than in distance, most strongly in lower total sulfur dioxide ($-1.8$ training standard deviations) together with higher pH, fixed acidity and volatile acidity ($+0.8$ to $+0.9$).

Finally, adding the AU output to the OOD score lowers the AUROC for most NN methods (e.g. NN-Lap-SN $0.982 \to 0.735$, NN-FSP $0.950 \to 0.619$): separating EU from AU matters for OOD detection in practice.

**Limitations.** This is a single dataset shift, so it illustrates rather than establishes how OOD detection relates to EU quality. The distance-matched set is small (134 points per class).

Table 11: Wine OOD test: methods trained on white wine (benchmark instances; balanced covariates, smooth-monotone noise, 3 oracle engines $\times$ 5 seeds) score held-out white wine against red wine (1500 each) by their epistemic uncertainty. AUROC, mean over the 15 instances, 95% bootstrap CI over instances. *Distance-matched*: white and red subsampled to equal Mahalanobis-distance distributions w.r.t. the training inputs (134 per class), so distance from the training data alone scores $\approx 0.5$.

| | All points (far OOD) | | Distance-matched (near OOD) | |
|---|---|---|---|---|
| Rank | Method | AUROC | Method | AUROC |
| 1 | NN-Lap-SN | 0.982 [0.979, 0.984] | NN-Ens | 0.666 [0.645, 0.687] |
| 2 | GP | 0.979 [0.977, 0.982] | NN-Bag | 0.639 [0.619, 0.655] |
| 3 | VLGP | 0.979 [0.977, 0.981] | NN-Lap-SN | 0.580 [0.557, 0.602] |
| 4 | BLR | 0.979 [0.976, 0.981] | CatBoost-KGB | 0.572 [0.558, 0.588] |
| 5 | SNGP-Het | 0.976 [0.972, 0.980] | NN-EDL | 0.569 [0.483, 0.652] |
| 6 | NN-FSP-SN | 0.976 [0.968, 0.983] | GP | 0.566 [0.528, 0.604] |
| 7 | CatBoost-KGB | 0.969 [0.966, 0.972] | NGBoost-NIG | 0.558 [0.520, 0.595] |
| 8 | NN-Ens | 0.968 [0.965, 0.972] | SNGP-Het | 0.547 [0.507, 0.585] |
| 9 | NN-Bag | 0.960 [0.955, 0.965] | CatBoost-PS | 0.540 [0.479, 0.597] |
| 10 | NN-FSP | 0.950 [0.936, 0.962] | NN-FSP-SN | 0.539 [0.469, 0.604] |
| 11 | NN-Drop | 0.937 [0.919, 0.954] | VLGP | 0.534 [0.517, 0.553] |
| 12 | NN-Lap | 0.933 [0.918, 0.947] | NN-Drop | 0.528 [0.471, 0.590] |
| 13 | NN-SWAG | 0.910 [0.888, 0.932] | NN-DEUP* | 0.521 [0.492, 0.562] |
| 14 | DKL | 0.867 [0.807, 0.923] | NGBoost-Bag | 0.519 [0.478, 0.562] |
| 15 | NGBoost-Bag | 0.862 [0.825, 0.900] | NN-FSP | 0.497 [0.462, 0.537] |
| 16 | CatBoost-PS | 0.795 [0.687, 0.885] | NN-SWAG | 0.488 [0.436, 0.546] |
| 17 | NGBoost-NIG | 0.671 [0.614, 0.733] | BLR | 0.485 [0.455, 0.512] |
| 18 | NN-DEUP* | 0.590 [0.527, 0.669] | DKL | 0.469 [0.394, 0.553] |
| 19 | NN-EDL | 0.482 [0.332, 0.626] | NN-Lap | 0.424 [0.376, 0.474] |
| *Reference scores* | | | | |
| | Mahalanobis distance | 0.982 | Mahalanobis distance | 0.537 |
| | 10-NN distance | 0.981 | 10-NN distance | 0.567 |
| | Supervised classifier† | 0.997 | Supervised classifier† | 0.954 |

† gradient boosting trained on red vs. white labels, 5-fold cross-validated; an upper bound on separability, not an uncertainty method.

Table 12: Agreement between the wine OOD ranking and the benchmark ranking of the same 19 methods, for three definitions of the epistemic target. Kendall's $W$ (tie-corrected); for two rankings $W = (1 + \rho_S)/2$, so 0.5 is no association and 1 identical rankings. Benchmark ranking: mean Spearman $\rho$ between EU and the target over all regression runs (balanced and OOD covariates). 95% CI: bootstrap over the 15 wine instances, benchmark rankings held fixed. The three benchmark rankings agree among themselves with $W = 0.93$, which keeps the $m = 4$ row high even when the OOD ranking is unrelated to them; the pairwise rows are the informative ones.

| Benchmark EU target | All points (far OOD) | Distance-matched (near OOD) |
|---|---|---|
| Posterior risk | 0.89 [0.85, 0.91] | 0.64 [0.52, 0.75] |
| Posterior variance | 0.92 [0.91, 0.93] | 0.49 [0.37, 0.63] |
| Pointwise excess risk | 0.84 [0.81, 0.85] | 0.53 [0.42, 0.65] |
| All four rankings ($m = 4$) | 0.87 [0.85, 0.88] | 0.63 [0.54, 0.71] |

### D.4 Spectral Normalization

In the main benchmark SNGP-Het tracks the posterior excess risk much better than DKL ($\rho = 0.65$ vs. 0.38). Both place a Gaussian-process layer on learned features. Beyond their output layers, SNGP spectrally normalizes its feature extractor so that distances in input space are approximately preserved in feature space (Liu et al., 2020). This appendix tests whether spectral normalization (SN) improves the EU of the other NN methods, and whether any improvement is caused by distance preservation rather than by the additional regularization SN introduces.

**Experimental setup.** We study the eight NN methods of the benchmark: NN-Lap , NN-FSP , NN-DEUP, NN-EDL, NN-SWAG, NN-Drop, NN-Ens and NN-Bag. Each is fitted in three arms:

- *plain*: the benchmark's own tuned configuration, without SN;
- *SN*: every dense layer of the residual backbone is spectrally normalized, $W \leftarrow W \cdot \min\big(1, c/\sigma_{\max}(W)\big)$, with $\sigma_{\max}$ estimated by power iteration (one step per training update, three refinement steps at inference). The bound $c$ is searched log-uniformly in $[0.5, 3]$ and all other hyperparameters are re-tuned under the constraint;
- *WD*: no SN, but weight decay is forced into $[10^{-2}, 1]$ (the benchmark's range is $[10^{-6}, 10^{-2}]$) and all other hyperparameters are re-tuned. This control asks whether a conventional regularizer of comparable or greater strength reproduces the effect of SN.

Balanced covariates, smooth-monotone heteroscedastic noise, all 13 datasets, the three oracle engines (GP, HBART, BNN).

**Evaluation.** Let $h(x)$ denote the backbone's output, i.e. the features the uncertainty head operates on, and $\tilde{x}$ the input standardized with the training mean and standard deviation (the space the network sees). For each instance we draw 20,000 random pairs $(i, j)$ of evaluation points, the same pairs for all three arms, and compute input and feature distances

$$d_{ij}^{\text{in}} = \|\tilde{x}_i - \tilde{x}_j\|_2, \qquad d_{ij}^{\text{feat}} = \|h(x_i) - h(x_j)\|_2.$$

*Local distance preservation* is the Spearman correlation between the two, restricted to the $25\%$ closest input pairs,

$$\text{pres}_{\text{loc}} = \rho_S\big(d_{ij}^{\text{in}}, d_{ij}^{\text{feat}} \;:\; d_{ij}^{\text{in}} \le q_{0.25}(d^{\text{in}})\big).$$

A value of 1 means that nearby inputs keep their relative distances in feature space; small values mean the feature map reorders or collapses neighbourhoods. For ensembles and bagging each member's backbone is scored and the scores are averaged. SWAG is scored at its SWA mean, MC dropout with dropout switched off.

For each instance and method we form the paired differences $\Delta = \text{arm} - \text{plain}$ for SN and for WD (Table 13). Point estimates are means over the 195 instances. Confidence intervals are $95\%$ percentile intervals from a non-parametric bootstrap that resamples the 13 *datasets* with replacement (2,000 replicates), since datasets, not instances, are the unit to which the conclusions should generalize.

**Analysis.** To ask whether SN helps through distance preservation, we regress, per method, the change in EU quality on the change in preservation, the change in accuracy, the arm, and the change in depth, pooling the SN and WD contrasts:

$$\Delta\rho \;=\; \beta_{\text{pres}}\,\Delta\text{pres}_{\text{loc}} + \beta_{\text{fit}}\,\Delta\log\text{NMSE} + \beta_{\text{SN}}\,\mathbb{1}[\text{SN}] + \beta_{\text{depth}}\,\Delta\text{depth} + \alpha_{\text{dataset}} + \varepsilon,$$

with dataset fixed effects $\alpha_{\text{dataset}}$, all variables standardized (so coefficients are in standard deviations), contrasts winsorized at the 1st and 99th percentile within each method, and $95\%$ intervals from a dataset bootstrap. If SN acts through preservation, $\beta_{\text{pres}}$ is positive and the SN indicator loses its effect once $\Delta\text{pres}$ is in the model. If it acts through regularization or fit, $\beta_{\text{fit}}$ carries it instead. A pooled model over all methods adds method fixed effects.

Because SN changes preservation much more than WD, the arm indicator and $\Delta\text{pres}$ are correlated in the model above. We therefore also test, within the SN arm alone, whether instances in which SN raised preservation more also gained more: $\Delta\rho_{\text{var}} \sim \Delta\text{pres}_{\text{loc}} + \Delta\log\text{NMSE} + \alpha_{\text{dataset}}$, standardized, with dataset-bootstrap intervals.

For Table 14 we correlate the plain network's tuned depth, width, epochs and number of gradient steps with its preservation, and its depth with the SN gains, using Spearman's correlation *within* datasets (dataset means removed from both variables). Depth is chosen by the tuner rather than set by us, so these are associations within datasets, not interventions.

**Results.** SN raises local preservation in all eight methods ($+0.15$ to $+0.35$; pooled $+0.21$ $[0.17, 0.25]$), whereas WD leaves it unchanged (pooled $+0.01$ $[-0.01, 0.02]$).

For the four methods whose EU is read from a single network's features (NN-Lap, NN-FSP, NN-DEUP, NN-EDL), SN improves $\rho_{\text{risk}}$ by $+0.10$ to $+0.11$ and $\rho_{\text{var}}$ by $+0.08$ to $+0.26$. WD yields at

most about half of the gain, and only for NN-Lap and NN-FSP (Table 13). For the ensembles and MC dropout, whose EU arises from disagreement between models or masks, preservation rises without improving EU (NN-Ens $+0.00$, NN-Drop $-0.01$, NN-Bag $-0.03$ in $\rho_{\text{risk}}$). NN-SWAG deteriorates ($-0.08$ in $\rho_{\text{risk}}$, $-0.135$ in $\rho_{\text{var}}$), see below.

For posterior-variance tracking, preservation carries the effect for NN-Lap ($\beta_{\text{pres}} = +0.40$ $[0.20, 0.52]$, $\beta_{\text{SN}} = -0.00$ $[-0.18, 0.20]$) and NN-FSP ($+0.51$ $[0.22, 0.78]$, $\beta_{\text{SN}} = -0.08$ $[-0.27, 0.14]$), with a smaller contribution from accuracy ($\beta_{\text{fit}} = -0.30$ and $-0.26$). Pooled over all methods, the SN advantage in $\rho_{\text{var}}$ ($+0.11$ $[0.08, 0.15]$ without mediators) vanishes once the mediators are included ($\beta_{\text{SN}} = +0.01$ $[-0.09, 0.09]$) and is carried by preservation ($\beta_{\text{pres}} = +0.15$ $[0.05, 0.25]$), not by accuracy ($\beta_{\text{fit}} = -0.08$ $[-0.17, 0.02]$).

Within the SN arm, a one-standard-deviation larger rise in preservation goes with a $0.44$ $[0.03, 0.69]$ (NN-Lap) and $0.64$ $[0.38, 0.89]$ (NN-FSP) standard-deviation larger gain in $\rho_{\text{var}}$ at equal change in accuracy. For NN-DEUP and NN-EDL there is no such relation ($+0.09$ and $-0.10$, both not significant): their gains are not explained by preservation.

Within datasets, deeper plain networks preserve distances worse in all eight methods (Table 14; pooled $\rho_S = -0.57$), whereas width, epochs and gradient steps show no consistent relation. SN restores more preservation in deeper networks (pooled $+0.41$; positive for seven of eight methods) and the gain in $\rho_{\text{var}}$ grows with depth for NN-FSP ($+0.53$), NN-Lap ($+0.38$) and NN-DEUP ($+0.28$). The full benchmark (312 keys) agrees: for NN-FSP the within-dataset correlation of the SN gain with the plain depth is $+0.36$, and controlling for the plain network's preservation removes the depth effect ($+0.035$ to $+0.005$ per standard deviation), consistent with depth acting through preservation. When tuned with SN, networks also select deeper backbones (e.g. $2.6$ vs. $1.4$ residual blocks on average for NN-FSP).

**Benchmark comparison.** Figure 10 compares the SN variants with the benchmark methods on the same evaluation subset. Several SN variants approach SNGP-Het (NN-Laplace and FSP-Laplace) in EU quality, substantially narrowing the performance gap observed for their plain counterparts.

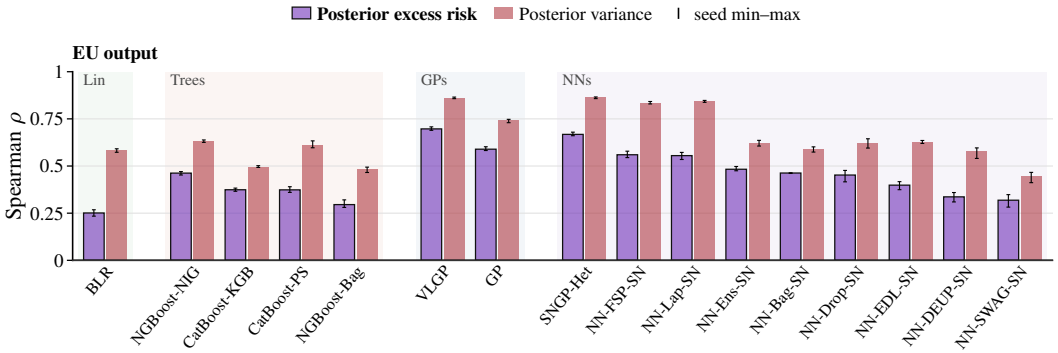



Figure 10: Benchmark results including SN variants on the spectral-normalization study subset: 13 datasets, three oracle engines, and five seeds, with balanced covariates and smooth-monotone heteroscedastic noise.

Table 13: Spectral normalisation (SN) vs. strong weight decay (WD), each tuned separately (30-trial Bayes sweep on validation NLL; WD searched in $[10^{-2}, 1]$, 100–1000$\times$ the usual range), as paired changes against the plain network. 13 datasets $\times$ 3 oracles $\times$ 5 seeds (195 instances per method; balanced covariates, smooth-monotone heteroscedastic noise). $\Delta\rho$: change in Spearman correlation of the EU output with the posterior risk and with the posterior variance; $\Delta$pres.: change in local distance preservation of the backbone features; NMSE: ratio of the arm's to the plain network's error (geometric mean). Bold: 95% CI from bootstrapping datasets excludes 0 (NMSE: 1).

| | SN − plain | | | | WD − plain | | | |
|---|---|---|---|---|---|---|---|---|
| Method | $\Delta\rho_{\text{risk}}$ | $\Delta\rho_{\text{var}}$ | $\Delta$pres. | NMSE | $\Delta\rho_{\text{risk}}$ | $\Delta\rho_{\text{var}}$ | $\Delta$pres. | NMSE |
| NN-Lap | **+0.103** | **+0.077** | **+0.172** | **0.48** | **+0.049** | **+0.033** | **+0.021** | **0.70** |
| NN-FSP | **+0.108** | **+0.092** | **+0.155** | **0.38** | **+0.046** | **+0.041** | −0.013 | **0.64** |
| NN-DEUP | **+0.108** | **+0.258** | **+0.345** | **0.44** | +0.020 | +0.023 | +0.012 | 0.91 |
| NN-EDL | **+0.109** | **+0.154** | **+0.184** | **0.48** | +0.023 | +0.035 | −0.024 | **0.78** |
| NN-SWAG | **−0.080** | **−0.135** | **+0.237** | **0.62** | −0.016 | −0.012 | **+0.041** | **0.88** |
| NN-Drop | −0.006 | −0.002 | **+0.244** | **0.83** | **−0.020** | **−0.023** | **+0.045** | **0.93** |
| NN-Ens | +0.002 | −0.021 | **+0.179** | **0.78** | −0.001 | **−0.025** | +0.025 | 0.93 |
| NN-Bag | **−0.026** | **−0.065** | **+0.150** | **0.75** | **−0.035** | **−0.060** | −0.029 | **0.89** |
| All (pooled) | **+0.040** | **+0.045** | **+0.208** | **0.57** | +0.008 | +0.001 | +0.010 | **0.83** |

Table 14: Deeper networks preserve distances worse and gain more from SN. Left: Spearman correlation of the plain network's distance preservation with its tuned training hyperparameters. Right: Spearman correlation of the plain network's depth with the gain from SN in distance preservation and in $\rho(\text{EU}, \text{posterior variance})$. Computed across the 195 instances per method (13 datasets $\times$ 3 oracles $\times$ 5 seeds) within datasets (dataset means removed); pooled: additionally centred per method. Distance preservation: Spearman correlation of pairwise input and feature distances over the $25\%$ closest input pairs. Bold: $p < 0.05$.

| | Plain preservation vs. | | | | Depth vs. SN gain in | |
|---|---|---|---|---|---|---|
| Method | Depth | Width | Epochs | Grad. steps | Preservation | $\rho_{\text{var}}$ |
| NN-Lap | **−0.70** | **+0.17** | +0.13 | −0.01 | **+0.63** | **+0.38** |
| NN-FSP | **−0.70** | **+0.51** | +0.09 | +0.10 | **+0.50** | **+0.53** |
| NN-DEUP | **−0.59** | **−0.62** | **−0.34** | **−0.41** | **+0.39** | **+0.28** |
| NN-EDL | **−0.44** | **+0.33** | **−0.37** | **−0.32** | **+0.37** | +0.07 |
| NN-SWAG | **−0.36** | +0.08 | **−0.18** | **−0.25** | −0.07 | −0.02 |
| NN-Drop | **−0.33** | +0.04 | +0.01 | **−0.14** | **+0.27** | +0.00 |
| NN-Ens | **−0.87** | +0.06 | **−0.27** | −0.08 | **+0.55** | **−0.19** |
| NN-Bag | **−0.76** | **+0.43** | **+0.27** | **+0.36** | **+0.49** | +0.01 |
| Pooled (8 methods) | **−0.57** | **+0.08** | **−0.12** | **−0.13** | **+0.41** | **+0.09** |