A Mutual Information Lower Bound for Multimodal Regression Active Learning

Leonardo Ferreira Guilhoto*
Graduate Group on Applied Mathematics & Computational Science
Philadelphia, PA 19104
guilhoto@sas.upenn.edu

Akshat Kaushal*
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
akaush@seas.upenn.edu

Paris Perdikaris
Department of Mechanical Engineering and Applied Mechanics
University of Pennsylvania
Philadelphia, PA 19104
pgp@seas.upenn.edu

Abstract

Active learning for continuous regression has lacked an acquisition function that targets epistemic uncertainty when the predictive distribution is multimodal: variance misses modal disagreement, and information-theoretic targets like BALD are designed for discrete outputs. We introduce a Two-Index framework that makes this separation explicit: one stochastic index selects among competing model hypotheses (epistemic source), while a second governs within-hypothesis randomness (aleatoric source). An entropy decomposition within the framework identifies the mutual information between the output and the epistemic index as a principled acquisition objective, and we prove this quantity vanishes as the model is trained on growing datasets, confirming that it captures exactly the uncertainty data can resolve. Because this mutual information is intractable for continuous outputs, we derive the Mutual Information Lower Bound (MI-LB) acquisition function, a closed-form approximation for Mixture Density Network ensembles. On benchmarks featuring multimodal systems, MI-LB matches or beats every baseline evaluated and is the only method to do so consistently – geometric and Fisher-based baselines compete only when the input space already encodes the multimodality, and collapse otherwise.

1 Introduction

Active learning in continuous, multivariate settings requires acquisition functions that can distinguish uncertainty the model can reduce by collecting more data (epistemic) from uncertainty intrinsic to the data-generating process (aleatoric). Existing approaches either operate in restricted settings, such

*Equal contribution.

Accepted at ICML 2026 Workshop on Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning.

as discrete classification [1] or unimodal Gaussian outputs [2], or rely on scalar summaries like predictive variance that conflate the two sources. The problem is worse in multimodal settings: two model hypotheses may assign the same mean and variance to an output while disagreeing on the number and location of modes. Variance alone cannot detect this kind of distributional disagreement.

We address this gap with a framework that keeps the two sources of uncertainty separate by construction. The framework extends the Epistemic Neural Network formalism [3] by introducing two independent stochastic indices: an epistemic index $Z \sim P_Z$ that parameterizes a family of hypotheses, and an aleatoric index $\epsilon \sim P_\epsilon$ that governs within-hypothesis stochasticity. A learned map $g_\theta(x; z, \epsilon)$ transforms these indices and the input into the output space. This approach provides a common vocabulary for a broad class of uncertainty quantification methods, facilitating the exchange of analyses among model families.

The two-index structure yields an entropy decomposition that separates predictive uncertainty into an epistemic term, the mutual information $I(Y; Z \mid x)$, and an aleatoric term, the expected conditional entropy $\mathbb{E}_Z[H(Y \mid x, Z)]$. Despite being a natural acquisition target, the mutual information requires computing differential entropies that are intractable for general continuous outputs. When the model family is an ensemble of Mixture Density Networks (MDNs), the predictive and conditional distributions are both Gaussian mixtures, and known upper and lower bounds on Gaussian-mixture entropy [4] can be combined to produce a tractable lower bound on $I(Y; Z \mid x)$. We call this the *Mutual Information Lower Bound* (MI-LB) acquisition function. Across three benchmarks with controllable multimodality, MI-LB matches or beats every baseline evaluated and is the only method to do so consistently. Geometric and Fisher-based baselines compete only when the input geometry already encodes the multimodal structure, and collapse otherwise. To the best of our knowledge, MI-LB is the first acquisition function specifically designed for multimodal continuous settings.

The main contributions of this paper are:

- The MI-LB acquisition function for active learning in continuous multivariate settings, with a proof that it lower bounds the true epistemic mutual information $I(Y; Z \mid x)$ and a closed-form expression that is efficient to evaluate under standard MDN assumptions.
- The Two-Index framework for disentangling epistemic and aleatoric uncertainty, applicable to a broad class of model families, with a proof that the epistemic term $I(Y; Z \mid x)$ vanishes with sufficient data under standard well-specification and consistency conditions (Appendix D).
- Experiments on three multimodal benchmarks showing that MI-LB matches or beats every baseline evaluated (Random, Variance, BAIT, Core-Set), and is the only method to do so consistently – geometric and Fisher-based baselines compete only when the input geometry already encodes the multimodal structure, and collapse when the modal disagreement lives in output space.

1.1 Related Work

Decomposing predictive uncertainty into aleatoric and epistemic components has been a longstanding goal in the machine learning literature [2, 5]. Foundational approaches include Bayesian Neural Networks [6], Deep Ensembles [7], and the Epistemic Neural Network framework [3], which provides a unified interface that does not require explicit Bayesian inference [5, 3]. Despite these advances, recent benchmarking shows that practical disentanglement remains elusive: modern evidential, variational, and deterministic methods exhibit rank correlations often exceeding 0.94 between their aleatoric and epistemic estimates [8]. This problem is particularly pronounced in continuous, multimodal regression tasks, where fitting a single Gaussian to diverse target modes inflates predictive variance, motivating the use of Mixture Density Networks (MDNs) [9, 10].

In the context of active learning, selecting informative data points to minimize labeling costs has traditionally relied on geometric or gradient-based heuristics [11]. Geometric methods such as Core-Set minimize L2 distances in embedding space [12], while Fisher-based methods like BAIT optimize bounds on the maximum likelihood estimator error using network gradients [13]. Both have important failure modes in multimodal settings: Core-Set ensures coverage in the learned embedding space and can fail when multimodality manifests in the output distribution rather than in the geometry of the inputs [12], and BAIT's last-layer Fisher embedding provides poor candidate rankings when the predictive likelihood is highly multimodal, since the gradient signal at a single Monte-Carlo sample is

dominated by within-mode variance. In these regimes, black-box acquisition strategies based on the empirical predictive covariance of ensembles often outperform white-box gradient methods [14, 11].

Information-theoretic approaches such as Bayesian Active Learning by Disagreement (BALD) use mutual information to target epistemic uncertainty directly [1, 15]. BALD is effective in discrete classification, but extending it to continuous regression requires evaluating differential entropy for distributions like Gaussian mixtures, which lacks a closed-form solution [4].

2 The Two-Index Approach for Disentangling Uncertainties

2.1 The Two-Index Generative Model

We build on the Epistemic Neural Network (ENN) framework [3] and the distribution network formalism of [10] to construct a unified model that explicitly represents both epistemic and aleatoric uncertainty through two independent stochastic indices.

In the ENN framework, a model is specified by a parameterized function f_θ and a fixed reference distribution P_Z. An *epistemic index* $Z \sim P_Z$ is sampled and passed to the network alongside the input x, producing a prediction $f_\theta(x; Z)$. Crucially, P_Z is not updated during training. Instead, training optimizes θ so that the mapping from Z to predictions becomes increasingly invariant under P_Z as more data is collected. Sensitivity of predictions to Z reflects epistemic uncertainty: the model has not yet identified a unique function consistent with the data.

We extend this framework by introducing the *aleatoric index* $\epsilon \sim P_\epsilon$, a second fixed reference distribution independent of both Z and θ. For a given epistemic index $Z = z$ and input x, the model defines a conditional distribution over outputs Y through the generative mapping

$$Y = g_\theta(x;\, z,\, \epsilon), \qquad Z \sim P_Z, \quad \epsilon \sim P_\epsilon, \quad Z \perp \epsilon. \tag{1}$$

Here g_θ is a learned transformation that maps the input x and two independent sources of randomness into the output space. The role of ϵ is to represent irreducible stochasticity: for any fixed z, sampling $\epsilon \sim P_\epsilon$ produces draws from the aleatoric conditional distribution $p_\theta(y \mid x, z)$. The role of Z is to represent reducible uncertainty: variation in predictions across different realizations of Z reflects hypotheses about the data-generating process that have not yet been eliminated by the available data. The network g_θ learns a transformation from the tractable distributions P_Z and P_ϵ into a flexible, potentially multimodal distribution over the output Y.

Induced distributions. The joint generative model in (1) induces a hierarchy of distributions, made precise using the push-forward operator. For a measurable space $(\mathcal{Y}, \mathcal{B}(\mathcal{Y}))$ and a measurable map $h : \mathcal{E} \to \mathcal{Y}$, the *push-forward* of a measure μ on $\mathcal{E}$ under h is the measure $h_\#\mu$ on $\mathcal{Y}$ defined by

$$(h_\#\mu)(A) := \mu\big(\{\epsilon \in \mathcal{E} : h(\epsilon) \in A\}\big), \qquad A \in \mathcal{B}(\mathcal{Y}). \tag{2}$$

For a fixed epistemic realization $z \in \mathcal{Z}$ and input $x \in \mathcal{X}$, the *aleatoric conditional distribution* is defined as the push-forward of P_ϵ under the map $g_\theta(x; z, \cdot) : \mathcal{E} \to \mathcal{Y}$:

$$p_\theta(\cdot \mid x, z) \;:=\; \big(g_\theta(x;\, z,\, \cdot)\big)_\# P_\epsilon. \tag{3}$$

Marginalizing equation (3) over the epistemic index yields the *predictive distribution*

$$p_\theta(\cdot \mid x) \;:=\; \int_{\mathcal{Z}} p_\theta(\cdot \mid x, z)\, dP_Z(z). \tag{4}$$

The predictive distribution conflates both sources of uncertainty. Disentangling them requires tracking the two indices separately, as formalized in the decompositions of Section 2.2.

Interpretation. Each realization $Z = z$ instantiates a *hypothesis* $p_\theta(\cdot \mid x, z)$ for the true conditional $p^*(\cdot \mid x)$; within a fixed hypothesis, ϵ governs irreducible stochasticity. Disagreement across realizations of Z is the origin of epistemic uncertainty. The predictive distribution has the same mathematical form as Bayesian Model Averaging [16], but the framework does not require a strict Bayesian formulation of P_Z: as illustrated in Appendix B, it accommodates ensembles, BNNs, conditional flow matching, and VAEs under a common vocabulary.

2.2 Quantifying the Two Sources of Uncertainty

The two-index structure enables exact decompositions of predictive uncertainty into its epistemic and aleatoric components. We provide two such decompositions via variance and entropy.

Variance-Based Decomposition. In the scalar output ($Y \in \mathbb{R}$) case [5], the law of total variance yields an exact additive decomposition. Writing $Y = g_\theta(x; Z, \epsilon)$ with $Z \perp \epsilon$,

$$\underbrace{\mathrm{Var}_{Z,\epsilon}(Y|x)}_{\text{total variance}} = \underbrace{\mathbb{E}_Z\big[\mathrm{Var}_\epsilon(Y \mid Z, x)\big]}_{\text{aleatoric variance}} + \underbrace{\mathrm{Var}_Z\big(\mathbb{E}_\epsilon[Y \mid Z, x]\big)}_{\text{epistemic variance}}. \tag{5}$$

The aleatoric term $\mathbb{E}_Z[\mathrm{Var}_\epsilon(Y \mid Z, x)]$ averages the output variance within each hypothesis over epistemic realizations; it measures the average irreducible spread. The epistemic term $\mathrm{Var}_Z(\mathbb{E}_\epsilon[Y \mid Z, x])$ measures the variance of the conditional mean across hypotheses; it captures disagreement between different epistemic realizations of the model.

This decomposition is exact and interpretable, but it summarizes uncertainty through second moments alone. As noted in the introduction to this section, two hypotheses can share the same conditional mean and variance while assigning probability mass to entirely different modes. The variance-based epistemic term would report an incomplete summary in such a case. For multimodal or non-Gaussian distributions, a richer summary is needed, as further discussed in Appendix C.

Entropy-Based Decomposition. We complement (equation 5) with an entropy-based decomposition that captures distributional disagreement beyond second moments. Define the total predictive uncertainty as the differential entropy of $p_\theta(y \mid x)$:

$$H_{Z,\epsilon}(Y \mid x) = \mathbb{E}_{Z,\epsilon}\big[-\log p_\theta(Y \mid x)\big]. \tag{6}$$

Applying the chain rule of mutual information, this decomposes as

$$H_{Z,\epsilon}(Y \mid x) = \underbrace{\mathbb{E}_Z\big[H_\epsilon(Y \mid Z, x)\big]}_{\text{aleatoric uncertainty}} + \underbrace{I(Y;\, Z \mid x)}_{\text{epistemic uncertainty}}, \tag{7}$$

where $I(Y; Z \mid x)$ is the *mutual information* between the output Y and the epistemic index Z, conditioned on the input x. The aleatoric term $\mathbb{E}_Z[H_\epsilon(Y \mid Z, x)]$ is the expected entropy of the conditional distribution $p_\theta(\cdot \mid x, z)$, averaged over the epistemic index. The epistemic term $I(Y; Z \mid x)$ quantifies how much knowing Z reduces uncertainty about Y: it is large when different epistemic indices yield meaningfully different conditional distributions, and vanishes when all hypotheses agree. Because mutual information is sensitive to the full shape of each hypothesis distribution, it detects the kind of modal disagreement that variance misses.

2.3 Asymptotic Guarantees

The decompositions (5) and (7) have the correct asymptotic behavior by construction. As the size of the training dataset grows, θ is optimized to produce predictions that are consistent with the data under all probable realizations of Z. In the limit of infinite data and a well-specified model, the learned θ renders $g_\theta(x; z, \epsilon)$ approximately independent of z for P_Z-almost all z, so that:

$$I(Y;\, Z \mid x,\, \mathcal{D}_n) \xrightarrow{n\to\infty} 0, \qquad \mathbb{E}_Z\big[H_\epsilon(Y \mid Z, x,\, \mathcal{D}_n)\big] \xrightarrow{n\to\infty} H(p^*(\cdot \mid x)), \tag{8}$$

where $p^*(\cdot \mid x)$ is the true conditional distribution of the data-generating process. Epistemic uncertainty vanishes as the model identifies the correct hypothesis, while aleatoric uncertainty converges to the irreducible entropy of the true process. Conversely, at an out-of-distribution input x_{OOD}, training data provides no constraints on Z, so $I(Y; Z \mid x_{\mathrm{OOD}})$ can remain large, providing a principled basis for out-of-distribution detection. We note that the convergence of the aleatoric term to $H(p^*(\cdot \mid x))$ requires that the aleatoric model class — the family of distributions representable by $g_\theta(x; z, \cdot)$ for fixed z — be sufficiently expressive to capture p^*. This is a non-trivial condition that motivates the use of flexible density estimators, such as Mixture Density Networks or implicit generative models, rather than restricting to unimodal Gaussian outputs.

This is stated informally as the following theorem, which is made precise and proved in Appendix D:

Theorem 2.1 (Informal: Epistemic Uncertainty Vanishes with Data)**.** *If a model can represent the true data-generating process $p^*(\cdot \mid x)$ and its parameters converge to the correct values as the dataset grows to infinity, then $I(Y; Z \mid x) \to 0$. In other words, a well-trained model's predictions eventually agree across all epistemic index values, and all remaining uncertainty is aleatoric.*

2.4 Implications for Data Acquisition

The decomposition from (7) and Thm. 2.1 together identify $I(Y; Z \mid x)$ as the natural acquisition target for active learning. It isolates exactly the component of predictive uncertainty that additional data can reduce: epistemic disagreement among hypotheses. The variance-based epistemic term from (5) targets the same goal in principle, but summarizes disagreement through second moments alone; it can miss regions where hypotheses differ in modal structure rather than spread. The mutual information $I(Y; Z \mid x)$, by contrast, is sensitive to the full distribution of each hypothesis, making it the right objective when the conditional distribution $p^*(y \mid x)$ may be multimodal.

The remaining challenge is computational. Evaluating $I(Y; Z \mid x)$ requires computing differential entropies of the predictive and conditional distributions, which have no closed-form expression for general continuous outputs. In the next section, we show that when the model family is an ensemble of Gaussian mixtures, analytical entropy bounds can be used to construct a tractable lower bound on $I(Y; Z \mid x)$ that serves as an effective acquisition function.

3 The Mutual Information Lower Bound (MI-LB) Acquisition Function

3.1 Entropy Intractability in Continuous Settings

Re-arranging the entropy-based decomposition from (7), the mutual information can be written as

$$I(Y;\, Z \mid x) = \underbrace{H(Y \mid x)}_{\text{(i) marginal entropy}} - \underbrace{\mathbb{E}_Z\big[H(Y \mid x, Z)\big]}_{\text{(ii) expected aleatoric entropy}} . \tag{9}$$

Computing this quantity requires evaluating two differential entropy terms: term (i), the entropy of the predictive distribution $p_\theta(\cdot \mid x)$ from equation 4, and term (ii), the entropy of $Y \mid x, Z = z$ for a fixed z, averaged over P_Z. For *implicit* distribution networks [10] — models that produce samples of Y via the map $g_\theta(x; z, \epsilon)$ but do not provide an analytical form for $p_\theta(\cdot \mid x, z)$ — both terms must be estimated from samples using kernel density estimators, which may scale poorly with the dimension of $\mathcal{Y}$. For *explicit* distribution networks, the density $p_\theta(\cdot \mid x, z)$ is available analytically, enabling structured approximation of both terms. We exploit this structure in the case of Mixture Density Networks, where each $p_\theta(\cdot \mid x, z)$ is a mixture of Gaussians.

3.2 MDN Structure and the Marginal Mixture

Assume Z takes values in the discrete index set $\{1, \ldots, n_{\text{ens}}\}$ with $P_Z(Z = z) = w_z$, and suppose that for each z, the aleatoric conditional $p_\theta(\cdot \mid x, z)$ is a Gaussian mixture with K components:

$$p_\theta(y \mid x, z) = \sum_{i=1}^{K} \alpha_i^{(z)}(x)\, \mathcal{N}\Big(y;\, \mu_i^{(z)}(x),\, C_i^{(z)}(x)\Big)\,, \tag{10}$$

where $\alpha_i^{(z)}(x) > 0$, $\sum_i \alpha_i^{(z)}(x) = 1$, $\mu_i^{(z)}(x) \in \mathbb{R}^N$, and $C_i^{(z)}(x) \in \mathbb{R}^{N \times N}$ is a symmetric positive definite covariance matrix. Here N denotes the dimension of the output space $\mathcal{Y} \subseteq \mathbb{R}^N$. Under this formulation, the predictive marginal distribution (4) is

$$p_\theta(y \mid x) = \sum_{z=1}^{n_{\text{ens}}} w_z\, p_\theta(y \mid x, z) = \sum_{z=1}^{n_{\text{ens}}} \sum_{i=1}^{K} \underbrace{w_z\, \alpha_i^{(z)}(x)}_{=:\, \beta_{z,i}(x)}\, \mathcal{N}\Big(y;\, \mu_i^{(z)}(x),\, C_i^{(z)}(x)\Big)\,, \tag{11}$$

which is a Gaussian mixture with $n_{\text{ens}} \cdot K$ components and weights $\beta_{z,i}(x) = w_z\, \alpha_i^{(z)}(x)$. Both terms in (9) therefore require computing the differential entropy of a Gaussian mixture, for which no closed-form expression exists.

3.3 Entropy Bounds for Gaussian Mixtures

We make use of two bounds from [4]. Let $p(y) = \sum_{i=1}^{K} \pi_i\, \mathcal{N}(y; \mu_i, C_i)$ be a Gaussian mixture with K components, weights π_i, means $\mu_i \in \mathbb{R}^N$, and covariances $C_i \in \mathbb{R}^{N \times N}$. [4] shows that the

differential entropy $H(p)$ satisfies $H_{\text{lower}} \leq H(p) \leq H_{\text{upper}}$, where

$$H_{\text{lower}} = -\sum_{i=1}^{K} \pi_i \log\left(\sum_{j=1}^{K} \pi_j \, \mathcal{N}(\mu_i;\, \mu_j,\, C_i + C_j)\right), \tag{12}$$

$$H_{\text{upper}} = \sum_{i=1}^{K} \pi_i \left(-\log \pi_i + \frac{1}{2}\log\big((2\pi e)^N \, |C_i|\big)\right). \tag{13}$$

Both bounds are computable in $O(K^2)$ operations. The upper bound (13) decomposes as the weighted sum of the entropy of individual modes and is tight when the components have negligible overlap. The lower bound (12) accounts for inter-component similarity through pairwise Gaussian evaluations at the component means. Under the standard MDN assumption of diagonal covariances $C_i = \text{diag}(\sigma_{i,1}^2, \ldots, \sigma_{i,N}^2)$, the Gaussian evaluations in (12) and the determinants in (13) reduce to sums of univariate quantities, making both bounds particularly efficient to evaluate.

3.4 The MI-LB Acquisition Function

We define the *Mutual Information Lower Bound* (MI-LB) acquisition function as

$$\boxed{\textit{MI-LB}(x) := H_{\text{lower}}(Y \mid x) - \sum_{z=1}^{n_{\text{ens}}} w_z \, H_{\text{upper}}(Y \mid x, Z = z),} \tag{14}$$

where $H_{\text{lower}}(Y \mid x)$ is the lower bound (12) applied to the $n_{\text{ens}} \cdot K$-component marginal mixture (11) with component weights $\beta_{z,i}(x)$, means $\mu_i^{(z)}(x)$, and covariances $C_i^{(z)}(x)$; and $H_{\text{upper}}(Y \mid x, Z = z)$ is the upper bound (13) applied to the K-component conditional mixture (10) for each z. Explicitly,

$$\begin{aligned}\textit{MI-LB}(x) = &-\sum_{z=1}^{n_{\text{ens}}}\sum_{i=1}^{K} \beta_{z,i} \log\left(\sum_{l=1}^{n_{\text{ens}}}\sum_{j=1}^{K} \beta_{l,j} \, \mathcal{N}\Big(\mu_i^{(z)};\, \mu_j^{(l)},\, C_i^{(z)} + C_j^{(l)}\Big)\right) \\ &- \sum_{z=1}^{n_{\text{ens}}} w_z \sum_{i=1}^{K} \alpha_i^{(z)} \left(-\log \alpha_i^{(z)} + \frac{1}{2}\log\Big((2\pi e)^N \, |C_i^{(z)}|\Big)\right),\end{aligned} \tag{15}$$

where we suppress the dependence on x for readability. The fundamental property of MI-LB is that it constitutes a certified lower bound on the true epistemic uncertainty $I(Y; Z \mid x)$, which we state formally as follows, with a proof given in Appendix E.

Theorem 3.1 (MI-LB is a lower bound on mutual information)**.** *Under the model assumptions of Section 2, with Z supported on $\{1, \ldots, n_{\text{ens}}\}$ and $p_\theta(\cdot \mid x, z)$ a Gaussian mixture of the form (10) for each z, it holds that*

$$\textit{MI-LB}(x) \leq I(Y;\, Z \mid x) \qquad \textit{for all } x \in \mathcal{X}. \tag{16}$$

The practical consequence of Theorem 3.1 is that maximizing MI-LB over a candidate pool is a conservative acquisition strategy: any point selected by MI-LB is guaranteed to have true epistemic uncertainty at least as large as the reported score. This mirrors the logic of the Evidence Lower BOund (ELBO) in variational inference [17], where optimizing a lower bound on the log-evidence provides a principled surrogate without risking false certification of a poor solution. Consequently, the MI-LB acquisition rule selects

$$x^* = \underset{x \in \mathcal{X}_{\text{pool}}}{\arg\max} \; \textit{MI-LB}(x). \tag{17}$$

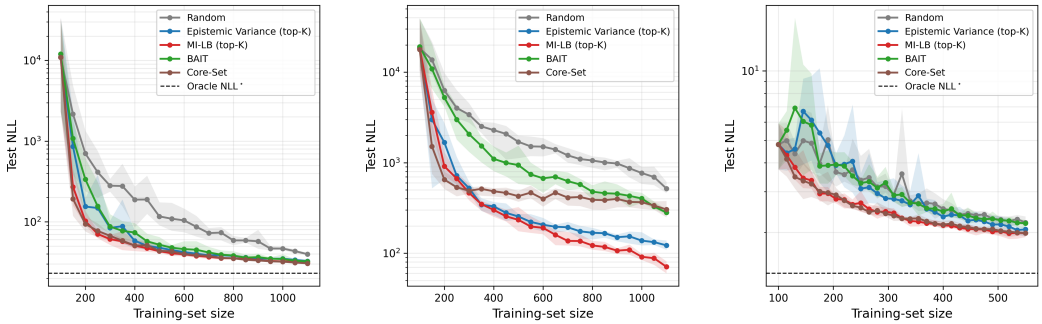
Connection to BALD. The Bayesian Active Learning by Disagreement (BALD) acquisition [1] is defined identically to (9) in the discrete classification setting, where Y follows a categorical distribution and entropy is Shannon entropy, admitting exact computation. MI-LB can therefore be understood as a continuous, multivariate alternative to BALD for regression, with the [4] entropy bounds replacing exact entropy to restore tractability in the absence of a finite output alphabet.

4 Experiments

We benchmark MI-LB, the acquisition function proposed in Section 3, against four baselines on three multimodal problems: **Random**, **Epistemic Variance**, **BAIT** [13] (last-layer Fisher trace), and **Core-Set** [12] (k-Center-Greedy on the MDN backbone). SBAL [14] and MaxDist [11] batch variants of Variance and MI-LB are deferred to the appendix. Among the three benchmarks, §4.2 (the coupled double-well) is the discriminating one: input geometry (σ, κ) does not encode the modal structure, which lives in output space via Kramers escape. §4.1 and §4.3 have phase boundaries in the input space and serve to characterise where geometric baselines remain competitive.

Table 1: Per-benchmark settings. All used a pool of 50,000 inputs, 100 initial labels, and an ensemble of size 8.

Benchmark	$\dim x$	$\dim y$	K_{MDN}	rounds $\times$ batch	total labels	NLL^*
Multimodal conditional(§4.1)	10	16	5	20×50	1,100	22.98
Coupled double-well (§4.2)	7	20	8	20×50	1,100	—
Ternary phases (§4.3)	8	1	4	30×15	550	1.33



(a) Multimodal conditional (§4.1) (b) Coupled double-well (§4.2) (c) Ternary phases (§4.3)

Figure 1: Test NLL vs. training-set size on each benchmark for MI-LB against Random, Epistemic Variance, BAIT, and Core-Set in top-k (5 seeds; bands min–max); dashed = oracle NLL^* where available. MI-LB matches or beats every baseline on every benchmark; geometric methods (Core-Set, BAIT) are competitive only when input geometry encodes the multimodality (a, c) and collapse when it lives in output space (b).

4.1 Synthetic Multimodal Conditional Problem

We construct a synthetic conditional distribution $p^*(y \mid x)$ as a mixture of $K{=}3$ Gaussians:

$$p^*(y \mid x) = \sum_{k=1}^{K} \pi_k(x)\, \mathcal{N}\big(y;\, \mu_k(x),\, \Sigma_k(x)\big), \tag{18}$$

where the mixing weights π_k, means μ_k, and diagonal covariances Σ_k are input-dependent functions (full specification in Appendix F). Inputs $x \in \mathbb{R}^{10}$ lie on a 4-dimensional manifold embedded via $x = \tanh(Al + b)$, $l \sim \mathcal{N}(0, I_4)$, respecting the manifold hypothesis [18], and outputs $y \in \mathbb{R}^{16}$. Because p^* is known in closed form, the oracle NLL^* is computable exactly as a calibration reference. A $K{=}5$ MDN ensemble trained on the full pool reaches NLL 24.72 vs. oracle 22.98, while a $K{=}1$ MDN attains only 47.76 – the conditional is multimodal and a mixture head is necessary (Fig. 2). The mixing weights $\pi_k(x)$ are gated by the radial coordinate $r = \|x_{1:L}\|$: for $r \lesssim r_0$ a single component dominates and $p^*(y \mid x)$ is effectively unimodal, while for $r \gtrsim r_0$ multiple components carry comparable mass and p^* becomes genuinely multimodal, producing a sharp unimodal–multimodal transition at $r \approx r_0$ (full form in App. F.2).

Why this is a useful benchmark. The coexistence of unimodal and multimodal regimes makes this problem a natural testbed for the Two-Index framework: $I(Y; Z \mid x)$ should be small in the interior, where all ensemble members agree on a single-component conditional, and large near the transition boundary $\|x_{1:L}\| \approx r_0$, where members must resolve the weights and locations of multiple

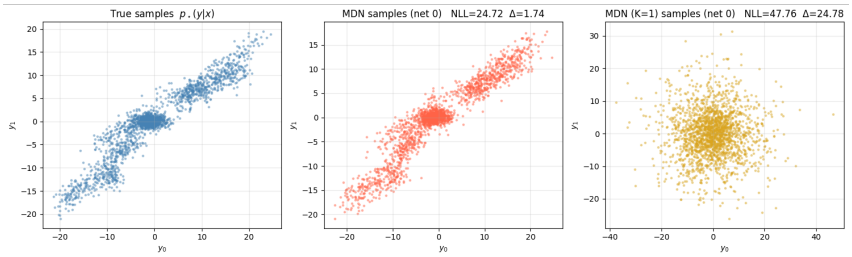


Figure 2: Predicted vs. true samples in the (y_0, y_1) plane on held-out inputs. **Left:** draws from the oracle $p^*(y \mid x)$, displaying the multimodal structure of the target conditional. **Middle:** MDN ensemble samples; the recovered geometry closely matches the oracle, with calibration gap $\Delta = 1.74$. **Right:** single-Gaussian MDN samples collapses to an isotropic blob that cannot represent disjoint modes, yielding $\Delta = 24.78$.

components. A well-calibrated acquisition function should therefore concentrate queries on this boundary. As a lower bound on $I(Y; Z \mid x)$, MI-LB is designed to detect this distributional disagreement, while variance-based scores summarize ensemble disagreement through second moments alone and underweight regions where hypotheses differ in modal structure rather than spread.

Results. Figure 1a compares MI-LB against the four baselines. From the very first acquisition rounds MI-LB pulls clearly ahead of Random, Variance, and BAIT and tracks the strongest baseline (Core-Set) throughout the budget, ending within seed bands at 31.1 ± 0.3 vs. 30.6 ± 0.6 at $n = 1100$. MI-LB matches the best geometric baseline on this benchmark with the tightest seed-to-seed std among all methods, tying Core-Set under top-k and beating it under MaxDist batch selection (Appendix F.2, Table 5). Core-Set is competitive here for a benchmark-specific reason: the input manifold $x = \tanh(Al + b)$ is 4-dimensional and the radial coordinate $\|x_{1:L}\|$ already encodes the unimodal–multimodal gate, so k-Center-Greedy in feature space implicitly samples the boundary that MI-LB targets explicitly. Sections 4.2–4.3 test this dependence and show that MI-LB retains its advantage when geometry no longer encodes multimodality, whereas Core-Set degrades sharply. Spatially, MI-LB concentrates on the unimodal–multimodal boundary, Variance is weaker on the same region, Core-Set covers the manifold uniformly, Random spreads everywhere (App. F.2, Fig. 6).

4.2 Coupled Double-Well System

We consider a chain of $P = 5$ particles evolving in coupled one-dimensional double-well potentials under overdamped Langevin dynamics,

$$dq_i = \Big[\underbrace{a\,(q_i - q_i^3)}_{\text{double-well force}} + \underbrace{\kappa \sum_{j \in \mathrm{nn}(i)} (q_j - q_i)}_{\text{nearest-neighbour coupling}} \Big]\, dt + \sigma\, dW_i, \qquad i = 1, \ldots, P, \tag{19}$$

where dW_i are independent Wiener increments and we fix $a = 1$. Inputs encode the initial particle configuration together with (σ, κ); outputs stack particle positions at four snapshot times. Full configuration, integration scheme, and evaluation protocol in Appendix F.3.

Why this is a useful benchmark. The conditional $p^*(y \mid x)$ undergoes a sharp change in mode structure controlled by σ^2/a: at low σ each particle stays in its starting well (p^* unimodal per particle); at high σ, noise triggers *Kramers escape* [19], making the single-particle marginal bimodal. Coupling κ raises the effective barrier for collective flips, coupling single-particle marginals into a joint distribution over $\{-1, +1\}^P$ basins. The Kramers exponent $a/(2\sigma^2)$ at barrier height $\Delta V = a/4$ places the crossover at $\sigma \gtrsim \sqrt{a/2} \approx 0.71$. A single-particle sweep (Fig. 3) confirms this scaling, and at $\sigma \gtrsim 1$ noise overwhelms the barrier so that mass spreads beyond both wells into the $|q| > 1$ tails.

With σ and κ as input coordinates, the phase boundary lives directly in input space, marking where informative acquisitions should concentrate. A Gaussian head must smear mass across both wells in the bimodal regime, inflating test NLL: a single-Gaussian ($K = 1$) head pays a 5.51-nat penalty against the $K = 8$ mixture head used on this benchmark (Appendix F.3, Fig. 8). This penalty is a

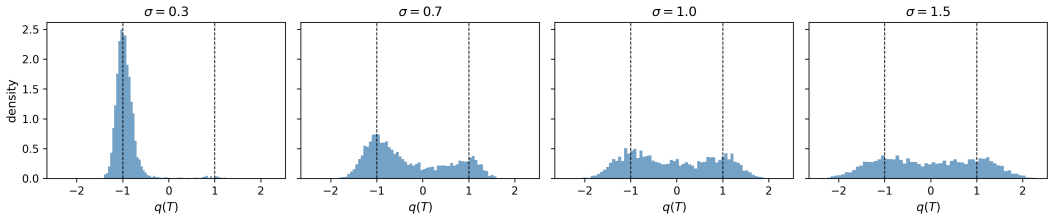


Figure 3: Terminal position $q(T)$ histograms for an uncoupled particle ($P = 1$, $\kappa = 0$, $q(0) = -0.5$) at four noise levels; dashed lines mark $q = \pm 1$. At $\sigma = 0.3$ the particle stays trapped near $q = -1$; at $\sigma = 0.7 \approx \sqrt{a/2}$ Kramers escape fills both wells; for $\sigma \geq 1$ noise dominates the barrier, spreading mass into the $|q| > 1$ tails.

property of the benchmark, not of any acquisition strategy: any method built on a single-Gaussian head fails here by construction.

Results. Figure 1b: MI-LB wins decisively at $\mathbf{71} \pm \mathbf{8}$ at $n = 1100$ - $1.7\times$ better than Variance (122 ± 11), $4\times$ better than Core-Set (304 ± 55) and BAIT (281 ± 15), and $7\times$ better than Random (518 ± 60). This is the discriminating benchmark: with the bimodal regime's mode structure living in *output* space (Kramers escape between wells), two ensemble members can agree on $\mathbb{E}[Y \mid x]$ while disagreeing on whether mass concentrates near $q = +1$, $q = -1$, or both – so feature-space coverage (Core-Set) and last-layer Fisher (BAIT) cannot rank candidates informatively, whereas MI-LB's entropy decomposition over the full output mixture targets exactly the disagreement that matters.

4.3 Synthetic Phase-Competition Benchmark

Active learning is a standard tool in materials discovery, where labels are expensive and composition–process inputs live on low-dimensional manifolds [20, 21, 22]. As in the previous benchmarks, discrete phases compete across the input space, producing boundary-localised multimodality. We build a synthetic CALPHAD-style [23] benchmark that reproduces this structure.

Each input is a composition on the 3-component simplex (with $x_C = 1 - x_A - x_B$) concatenated with $n_{\text{proc}} = 6$ continuous process parameters. Per-phase Gibbs free energies $G_\phi(x_A, x_B, x_C)$ are random quadratic forms in composition (independent of p), and the latent phase distribution is $\text{softmax}(-G_\phi/\tau_G)$ at temperature $\tau_G = 0.08$, producing sharp phase boundaries on the simplex slice. The scalar response is sampled from a per-phase Gaussian whose mean depends on both composition and process parameters; full simulator parameters in Appendix F.4.

Why this is a useful benchmark. The composition (x_A, x_B, x_C) acts as a categorical latent selecting a phase, confining multimodality to a thin boundary region on the simplex; the process subspace contributes only a smooth, phase-conditional shift. This factorisation isolates a clean test of boundary localisation (realised phase masses and boundary fraction in Appendix F.4).

Results. The tighter budget than Sections 4.1 and 4.2 reflects the 1-D scalar output, on which NLL saturates faster than the 16- and 20-dimensional outputs of the other benchmarks. Figure 1c: MI-LB reaches $\mathbf{1.99} \pm \mathbf{0.04}$ at $n = 550$, tied with Core-Set (1.99 ± 0.04); Variance (2.06), BAIT (2.19), and Random (2.21) trail. MaxDist on top of MI-LB narrowly wins overall (1.95 ± 0.06). BAIT collapses to Random-level, the 1-D scalar output makes its single-MC last-layer Fisher estimate too noisy to rank candidates. Core-Set's parity again reflects benchmark geometry (phase boundaries lie on the composition simplex) and Sections 4.2 shows it does not transfer when multimodality is in output space. MI-LB retains the tightest seed-to-seed spread in the data-scarce regime ($n \in [115, 250]$): $\sim 2.6\times$ tighter than Random, $\sim 4.5\times$ than Variance (Appendix Table 11).

5 Conclusion

Summary. We introduce the Two-Index generative framework to formally disentangle epistemic and aleatoric uncertainty across diverse model families. By leveraging this formalism, we derive the MI-LB acquisition function, providing a principled and computationally efficient Mutual Information lower bound for active learning in continuous, multimodal settings. Across three benchmarks MI-LB

matches or beats every baseline we evaluate (Random, Variance, BAIT, Core-Set) and is the only method that does so consistently – the geometric baselines are competitive only on benchmarks whose inputs already encode the multimodality. The discriminating test is the coupled double-well, where Kramers escape places the modal disagreement in output space: Core-Set and BAIT both collapse to within $1.7\times$ of Random, while MI-LB wins by $4\times$ – the empirical signature that distributional acquisition is doing something feature-space coverage cannot.

Limitations. MI-LB currently relies on the availability of explicit density components, such as those in Mixture Density Networks. Its performance is also tied to the quality of the entropy bounds, which may loosen in some settings. Furthermore, the framework assumes a well-specified model class capable of recovering the true irreducible stochasticity. Our empirical evaluation is also restricted to synthetic benchmarks with controllable multimodality, since no standard benchmarks exist for active learning in continuous multimodal regression; building such benchmarks is left to future work.

Future Work. Future research may focus on extending the MI-LB objective to implicit generative models, such as diffusion and flow matching, where densities are not analytically tractable. Another promising direction is to explore the framework's utility in safety-critical control tasks where distinguishing between lack of data and inherent noise is vital for risk-aware planning.

Acknowledgments and Disclosure of Funding

We would like to acknowledge support from the US Department of Energy under the Advanced Scientific Computing Research program (grant DE-SC0024563) and the US National Science Foundation (NSF) Soft AE Research Traineeship (NRT) Program (NSF grant 2152205). We also thank the developers of software that enabled this research, including JAX [24], Flax [25] Matplotlib [26] and NumPy [27].

Author Contributions

L.F.G. developed the theory for the two-index framework and MI-LB acquisition. A.K. developed the benchmarks, carried out experiments, and analyzed the data. P.P. provided funding and supervised this study. All authors helped to write and review the manuscript.

Impact Statement

Our contributions enable advances in deep learning and uncertainty quantification. This has the potential to impact a wide range of downstream applications. While we do not anticipate specific negative impacts from this work, as with any powerful predictive tool, there is potential for misuse. We encourage the research community to consider the ethical implications and potential dual-use scenarios when applying these technologies in sensitive domains and to avoid its application altogether to weaponry and other military technologies.

Declaration of LLM Usage

LLMs were used during the development of this paper, including for editing the writing, developing ideas, and as code assistants. We believe our use to be within the standard uses of this technology, and authors have verified the integrity of all material contained in this work.

References

[1] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning, 2011.

[2] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[3] Ian Osband, Zheng Wen, Mohammad Asghari, Morteza Ibrahimi, Xiyuan Lu, and Benjamin Van Roy. Epistemic neural networks. *CoRR*, abs/2107.08924, 2021.

[4] Marco F. Huber, Tim Bailey, Hugh Durrant-Whyte, and Uwe D. Hanebeck. On entropy approximation for gaussian mixture random vectors. In *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 181–188, 2008.

[5] Matias Valdenegro-Toro and Daniel Saromo Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1508–1516. IEEE, 2022.

[6] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR, 2021.

[7] Andrew Gordon Wilson and Pavel Izmailov. Deep ensembles as approximate bayesian inference. `https://cims.nyu.edu/~andrewgw/deepensembles/`, 2021.

[8] Bálint Mucsányi, Michael Kirchhof, and Seong Joon Oh. Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. *Advances in neural information processing systems*, 37:50972–51038, 2024.

[9] Christopher M Bishop. Mixture density networks. *Neural Computing Research Group Report*, 1994.

[10] Leonardo Ferreira Guilhoto, Akshat Kaushal, and Paris Perdikaris. Multimodal scientific learning beyond diffusions and flows, 2026.

[11] David Holzmüller, Viktor Zaverkin, Johannes Kästner, and Ingo Steinwart. A framework and benchmark for deep batch active learning for regression. *Journal of Machine Learning Research*, 24(164):1–81, 2023.

[12] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

[13] Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. Gone fishing: Neural active learning with fisher embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[14] Andreas Kirsch, Sebastian Farquhar, and Yarin Gal. A simple baseline for batch active learning with stochastic acquisition functions. *CoRR*, abs/2106.12059, 2021.

[15] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

[16] Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial (with comments by m. clyde, david draper and ei george, and a rejoinder by the authors. *Statistical science*, 14(4):382–417, 1999.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Lawrence Cayton et al. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12(1-17):1, 2005.

[19] H. A. Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.

[20] Turab Lookman, Prasanna V. Balachandran, Dezhen Xue, and Ruihao Yuan. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials*, 5(1):21, 2019.

[21] Dezhen Xue, Prasanna V. Balachandran, John Hogden, James Theiler, Deqing Xue, and Turab Lookman. Accelerated search for materials with targeted properties by adaptive design. *Nature Communications*, 7(1):11241, 2016.

[22] Eric Stach, Brian DeCost, A. Gilad Kusne, Jason Hattrick-Simpers, Keith A. Brown, Kristofer G. Reyes, Joshua Schrier, Simon Billinge, Tonio Buonassisi, Ian Foster, Carla P. Gomes, John M. Gregoire, Apurva Mehta, Joseph Montoya, Elsa Olivetti, Chiwoo Park, Eli Rotenberg, Semion K. Saikin, Sylvia Smullin, Valentin Stanev, and Benji Maruyama. Autonomous experimentation systems for materials development: A community perspective. *Matter*, 4(9):2702–2726, 2021.

[23] Hans Lukas, Suzana G. Fries, and Bo Sundman. *Computational Thermodynamics: The CALPHAD Method*. Cambridge University Press, USA, 1st edition, 2007.

[24] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[25] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2023.

[26] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[27] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[28] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[29] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[30] Leonardo Ferreira Guilhoto and Paris Perdikaris. Composite bayesian optimization in function spaces using neon—neural epistemic operator networks. *Scientific Reports*, 14(1):29199, 2024.

[31] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[32] Yulong Lu and Jianfeng Lu. A universal approximation theorem of deep neural networks for expressing probability distributions. *Advances in neural information processing systems*, 33:3094–3105, 2020.

[33] DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem, 2020.

[34] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

A Mathematical Notation

Table 2 summarizes the symbols and notation used in this work.

For operands that involve expectations, such as expectation $\mathbb{E}$, variance Var and entropy H, a sub-index indicates what is the random variable for which the expectation is being taken over. When the context is clear, this sub-index is often omitted.

Table 2: Summary of the symbols and notation used in this paper.

Symbol	**Meaning**
$x \in \mathcal{X}$	Input to the model
$Y \in \mathcal{Y}$	Output random variable
$\theta \in \Theta$	Parameters of a neural network
$Z \sim P_Z$	Epistemic index over model hypotheses
$\epsilon \sim P_\epsilon$	Aleatoric index for irreducible stochasticity
$g_\theta(x;\, z,\, \epsilon)$	Learned generative map to the output space
$p_\theta(\cdot \mid x, z)$	Aleatoric conditional distribution for fixed z from eq. (3)
$p_\theta(\cdot \mid x)$	Predictive distribution, marginal over P_Z from eq. (4)
$p^*(\cdot \mid x)$	True conditional distribution
$h_\# \mu$	Push-forward of measure μ under map h from eq. (2)
$\mathbb{E}$	Expectation operator
$\mathrm{Var}(\cdot)$	Variance operator
$H(\cdot)$	Differential entropy
$I(Y;\, Z \mid x)$	Epistemic mutual information
$\mathrm{KL}(p \,\|\, q)$	KL divergence from q to p
$\mathrm{TV}(p, q)$	Total variation distance between p and q
$\mathcal{N}(\mu, \Sigma)$	Normal distribution with mean μ and covariance Σ
$\mathcal{N}(y;\, \mu, \Sigma)$	PDF of $\mathcal{N}(\mu, \Sigma)$ evaluated at y
n_{ens}	Number of ensemble members
K	Number of components in a Gaussian mixture
$\alpha_i^{(z)}(x)$	Mixture weight of component i for member z
$\mu_i^{(z)}(x)$	Mean of component i for member z
$C_i^{(z)}(x)$	Covariance of component i for member z
w_z	Weight of ensemble member z under P_Z
MI-LB(x)	Mutual Information Lower Bound acquisition function
$\mathcal{D}_n$	Training dataset of size n
$\boldsymbol{I}_d$	Identity matrix of size d

B Concrete Examples of the Two Index Framework

To ground the two-index framework in practice, we show how three families of models already in common use fit naturally into the formalism of equation (1). In each case we identify the epistemic

index Z, the aleatoric index ϵ, the learned map g_θ, and the resulting conditional and predictive distributions.

Ensemble of Conditional Variational Autoencoders (C-VAEs). Consider an ensemble of n_{ens} independently trained C-VAE [28], indexed by $k \in \{1, \ldots, n_{\text{ens}}\}$. Each model k consists of a decoder $d_{\theta_k} : \mathcal{X} \times \mathcal{L} \to \mathcal{Y}$, where $\mathcal{L}$ is the latent space. At inference time, a latent code is sampled from the prior $\epsilon \sim P_\epsilon := \mathcal{N}(0, I)$ and passed through the decoder to produce a prediction.

In the two-index framework, we set:

$$\begin{aligned} Z &\sim P_Z := \text{Uniform}\{1, \ldots, n_{\text{ens}}\}, \\ \epsilon &\sim P_\epsilon := \mathcal{N}(0, I_d), \\ g_\theta(x;\, z,\, \epsilon) &:= d_{\theta_z}(x,\, \epsilon), \end{aligned}$$

where $\theta = (\theta_1, \ldots, \theta_{n_{\text{ens}}})$ collects all decoder parameters. The epistemic index $Z = k$ selects a member of the ensemble, capturing uncertainty about which model best represents the data-generating process. The aleatoric index ϵ is the latent noise injected into the decoder of the selected member, capturing the intrinsic stochasticity of the output given a fixed hypothesis. The aleatoric conditional from eq. (3) becomes

$$p_\theta(\cdot \mid x, k) = (d_{\theta_k}(x, \cdot))_\# \mathcal{N}(0, I_d),$$

which is the generative distribution of the k-th C-VAE. The predictive distribution equation 4 is the equally-weighted mixture of these per-member distributions:

$$p_\theta(\cdot \mid x) = \frac{1}{n_{\text{ens}}} \sum_{k=1}^{n_{\text{ens}}} p_\theta(\cdot \mid x, k).$$

Epistemic uncertainty, measured by $I(Y; Z \mid x)$, is large when the ensemble members disagree on their generative distributions, and vanishes when all decoders produce the same output distribution regardless of which member is selected.

Ensemble of Conditional Flow Matching Models. Flow matching models [29] learn a deterministic vector field that transports an initial noise sample $\epsilon \sim P_\epsilon$ to a target distribution over $\mathcal{Y}$, conditioned on input x. At inference time, the output is obtained by integrating the learned vector field from time $t = 0$ to $t = 1$, starting from ϵ. An ensemble of n_{ens} such models captures epistemic uncertainty through disagreement across members.

The two-index instantiation mirrors the C-VAE case, with a key structural difference: the aleatoric index ϵ is not a latent code injected at an intermediate layer but rather the *initial noise* supplied to the flow at inference time. Letting $\Phi_{\theta_k}(\cdot; x) : \mathcal{Y} \to \mathcal{Y}$ denote the learned flow map of member k (i.e., the solution operator of the ODE from $t = 0$ to $t = 1$), we set:

$$\begin{aligned} Z &\sim P_Z := \text{Uniform}\{1, \ldots, n_{\text{ens}}\}, \\ \epsilon &\sim P_\epsilon := \mathcal{N}(0, I_d), \\ g_\theta(x;\, z,\, \epsilon) &:= \Phi_{\theta_z}(\epsilon;\, x). \end{aligned}$$

The aleatoric conditional is

$$p_\theta(\cdot \mid x, k) = (\Phi_{\theta_k}(\cdot\,; x))_\# \mathcal{N}(0, I_d),$$

which, for an exactly learned flow, equals the target conditional distribution of member k.

Bayesian Neural Network for Classification. Bayesian Neural Networks (BNNs) are a special case of ENNs [3], with the epistemic index Z playing the role of a sample from the weight posterior $p(\theta \mid \mathcal{D})$. We instantiate this within the two-index framework for a classification setting with n_{cls} classes.

Let $f_w : \mathcal{X} \to \Delta^{n_{\text{cls}}-1}$ be a neural network with weights w, mapping inputs to the probability simplex. In a BNN, weights are treated as random variables with posterior $p(w \mid \mathcal{D})$ approximated by a distribution $Q_\theta(w)$ with learnable parameters θ (e.g., a mean-field Gaussian). The epistemic index Z is a draw from this approximate posterior. The aleatoric index ϵ serves as the source of randomness

for sampling a class label from the predicted categorical distribution. Concretely:

$$\begin{aligned} Z &\sim P_Z := Q_\theta(w), \\ \epsilon &\sim P_\epsilon := \mathrm{Uniform}(0,1), \\ g_\theta(x;\, z,\, \epsilon) &:= \min\{c \in \{1,\dots,n_{\mathrm{cls}}\} : \big(F_z(x)\big)_c \geq \epsilon\}, \end{aligned}$$

where $(F_z(x))_c := \sum_{j=1}^{c} [f_z(x)]_j$ is the c-th entry of the cumulative sum of the predicted class probabilities under weights z. This is the inverse CDF (quantile) transform: $g_\theta(x; z, \epsilon)$ samples a class label from the categorical distribution $\mathrm{Cat}(f_z(x))$ using ϵ as the uniform source of randomness. The aleatoric conditional is then

$$p_\theta(\cdot \mid x, z) = \mathrm{Cat}(f_z(x)),$$

and the predictive distribution is equal to the Bayesian Model Average (BMA):

$$p_\theta(\cdot \mid x) = \mathbb{E}_{Z\sim Q_\theta}\big[\mathrm{Cat}(f_Z(x))\big].$$

The epistemic term $I(Y; Z \mid x)$ measures disagreement among weight posterior samples about the predicted class probabilities, recovering the standard notion of epistemic uncertainty in BNN classification [3]. The aleatoric term $\mathbb{E}_Z[H(f_Z(x))]$ is the expected entropy of each categorical prediction, averaged over the weight posterior. It measures the label ambiguity that remains under a fixed weight configuration.

Epinet for Regression. The epinet [3] augments a base network with a small auxiliary network that takes an epistemic index z as an additional input, so that varying z at a fixed input traces out a family of predictions from a single trained model. Paired with a Gaussian aleatoric head, an epinet captures both sources of uncertainty within one network: the epinet generates the epistemic spread, and the predicted variance captures the aleatoric stochasticity.

Let $\mu_\theta^{\mathrm{base}} : \mathcal{X} \to \mathbb{R}^N$ be the base mean network, $\eta_\theta : \mathcal{X} \times \mathcal{Z}_{\mathrm{epi}} \to \mathbb{R}^N$ be the epinet, and $\sigma_\theta^2 : \mathcal{X} \to \mathbb{R}^N_{>0}$ be the predicted aleatoric variance. The mean prediction at epistemic index z is $\mu_\theta(x, z) := \mu_\theta^{\mathrm{base}}(x) + \eta_\theta(x, z)$. We set:

$$\begin{aligned} Z &\sim P_Z := \mathcal{N}(0, I_{d_{\mathrm{epi}}}), \\ \epsilon &\sim P_\epsilon := \mathcal{N}(0, I_N), \\ g_\theta(x;\, z,\, \epsilon) &:= \mu_\theta(x,\, z) + \sigma_\theta(x) \odot \epsilon, \end{aligned}$$

where $\odot$ denotes element-wise multiplication. The aleatoric conditional is

$$p_\theta(\cdot \mid x, z) = \mathcal{N}\big(\mu_\theta(x, z),\, \mathrm{diag}\big(\sigma_\theta^2(x)\big)\big)\,,$$

a Gaussian whose mean depends on z but whose covariance does not. The predictive distribution is the continuous mixture

$$p_\theta(\cdot \mid x) = \mathbb{E}_{Z\sim\mathcal{N}(0, I_{d_{\mathrm{epi}}})}\big[\mathcal{N}\big(\mu_\theta(x, Z),\, \mathrm{diag}\big(\sigma_\theta^2(x)\big)\big)\big]\,,$$

which is a Gaussian-mean mixture: the mean varies with Z while the covariance is fixed at $\mathrm{diag}(\sigma_\theta^2(x))$. The epistemic term $I(Y; Z \mid x)$ measures how much the mean prediction $\mu_\theta(x, Z)$ varies with the epistemic index. The aleatoric term has the closed form $\mathbb{E}_Z[H_\epsilon(Y \mid x, Z)] = \frac{1}{2}\log\Big((2\pi e)^N \prod_{i=1}^N \sigma_{\theta,i}^2(x)\Big)$, since the conditional covariance does not depend on Z. The two sources of uncertainty are therefore parameterized by disjoint sub-networks: the epinet η_θ governs the epistemic spread, and the variance head σ_θ governs the aleatoric noise. The same construction can be extended to the operator learning setting, where inputs and outputs are functions rather than vectors [30].

C Limitations of Variance for Multimodal Distributions

The Epistemic Variance acquisition function (Section 2.2), also known as Uncertainty Sampling, is the standard choice for active learning with real-valued predictions. There are settings, however, where it assigns identical scores to distributions with very different uncertainty profiles. We give a concrete example on the unit sphere that illustrates this failure mode.

For a distribution p over $\mathbb{R}^m$ with mean $\bar{y} = \mathbb{E}_p[Y]$, we define the *multivariate variance* as the trace of the covariance matrix:

$$\mathrm{Var}(p) := \mathrm{tr}\big(\mathrm{Cov}_p(Y)\big) = \mathbb{E}_p\big[\|Y - \bar{y}\|_2^2\big]. \tag{20}$$

This is the expected squared ℓ_2 distance from a sample to the mean, and it is the quantity that variance-based acquisition functions rank candidates by.

Consider the unit sphere $S^{m-1} := \{y \in \mathbb{R}^m : \|y\|_2 = 1\}$ and two inputs $x, x' \in \mathbb{R}^n$. For x, suppose $p_\theta(y \mid x)$ is the *uniform distribution* on S^{m-1}. For x', let $C_\delta(e_1)$ and $C_\delta(-e_1)$ denote the spherical caps of geodesic radius $\delta > 0$ centered at e_1 and $-e_1$ respectively, and define the *polar distribution* $p_\theta(y \mid x')$ as the distribution that selects one of the two caps with equal probability and then draws uniformly within it. Both distributions are continuous with respect to the surface measure on S^{m-1}, both have mean $\bar{y} = 0$ by symmetry, and every sample lies on S^{m-1}. Since $\|Y\|_2^2 = 1$ almost surely and $\bar{y} = 0$ for both distributions, equation (20) gives $\mathrm{Var}(p) = 1$ in both cases, exactly and for all m (see Figure 4 for an illustration). Any acquisition function that ranks inputs by variance assigns the same score to x and x'.

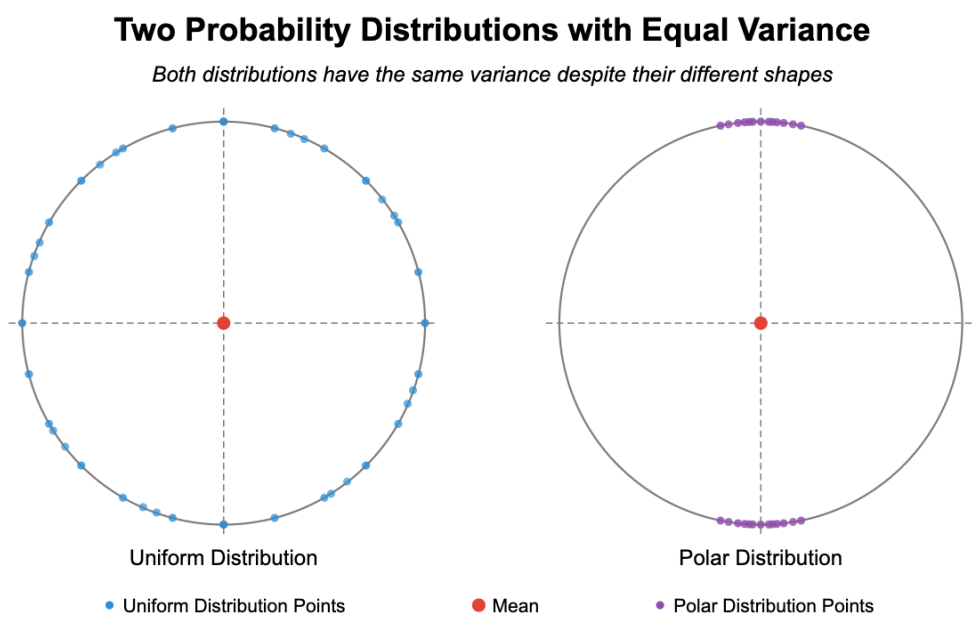


Figure 4: Two distributions on the unit circle with identical variance but different entropy. **Left:** the uniform distribution spreads samples evenly across the circle. **Right:** the polar distribution concentrates samples near two antipodal points. Both distributions have mean zero and multivariate variance equal to 1, yet the uniform distribution has strictly higher entropy.

The two distributions are not equally uncertain, however. The polar distribution confines the output to two small patches of the sphere, while the uniform distribution spreads mass over the entire surface. The entropy-based decomposition of Section 2.2 captures this gap. Writing $A_m = \mathrm{Area}(S^{m-1})$ for the total surface area and $A_\delta = \mathrm{Area}(C_\delta)$ for the area of each cap, the differential entropies with respect to the surface measure are

$$h\big(p_\theta(\cdot \mid x)\big) = \log A_m, \qquad h\big(p_\theta(\cdot \mid x')\big) = \log 2 + \log A_\delta. \tag{21}$$

The entropy gap is $\log(A_m/(2A_\delta))$, which grows without bound as $m \to \infty$ for any fixed cap radius δ. This follows from concentration of measure on the sphere: the fraction A_δ/A_m of the sphere covered by a cap of fixed geodesic radius $\delta < \pi/2$ vanishes exponentially in m, because the mass of the integrand $\sin^{m-2}(\phi)$ in the surface area element concentrates near the equator $\phi = \pi/2$. Entropy-based acquisition functions detect this growing gap, while variance-based scores cannot.

D Proof of Asymptotic Guarantees

In what follows, we formalize the insights from Section 2.3 and theorem 2.1 in order to provide guarantees on the asymptotic behavior under infinite data. First, we prove that epistemic uncertainty measured as the mutual information $I(Y; Z|x)$ converges to zero. Then, we show that the predicted aleatoric uncertainty $\mathbb{E}_Z\big[H_\epsilon(Y \mid Z, x, \mathcal{D}_n)\big]$ converges to the true quantity $H(p^*(\cdot \mid x))$. As a corollary of these two results, we also observe that the total predictive entropy also converges to the true entropy of the data-generating process.

D.1 Mutual Information Collapse

We begin by stating and proving a lemma that will be helpful in proving the main theorem.

Lemma D.1 (KL divergence–total variation bound under bounded likelihood ratio)**.** *Let p and q be probability measures on a measurable space $(\mathcal{Y}, \mathcal{B}(\mathcal{Y}))$ with $p \ll q$. Suppose there exists a constant $C \geq 1$ such that $dp/dq \leq C$ holds q-almost everywhere. Then*

$$\mathrm{KL}(p \,\|\, q) \leq 2M_C \cdot \mathrm{TV}(p, q),$$

where

$$M_C := \max\left(1,\ \frac{C \log C - C + 1}{C - 1}\right) \tag{22}$$

for $C > 1$, and $M_1 := 1$.

Proof. Let $h := dp/dq$, so that $0 \leq h \leq C$ holds q-a.e. Since $\int h\, dq = 1$, the KL divergence can be written as

$$\mathrm{KL}(p \,\|\, q) = \int h \log h \, dq = \int \phi(h)\, dq, \tag{23}$$

where $\phi(t) := t \log t - t + 1$. The function ϕ is non-negative and convex on $[0, \infty)$ with $\phi(1) = 0$ and $\phi'(1) = 0$.

We claim that $\phi(t) \leq M_C\, |t - 1|$ for all $t \in [0, C]$. On the interval $[0, 1]$: since $t \log t \leq 0$ for $t \in [0, 1]$, we have $\phi(t) = t \log t + (1 - t) \leq 1 - t = |t - 1|$. On the interval $[1, C]$: convexity of ϕ together with $\phi(1) = 0$ gives $\phi(t) \leq \phi(C) \cdot \frac{t-1}{C-1}$, where $\phi(C) = C \log C - C + 1$. Combining both cases yields $\phi(t) \leq M_C\, |t - 1|$ on $[0, C]$.

Integrating against q and applying the bound gives

$$\begin{aligned} \mathrm{KL}(p \,\|\, q) = \int \phi(h)\, dq &\leq M_C \int |h - 1|\, dq \\ &= 2M_C \cdot \mathrm{TV}(p, q), \end{aligned} \tag{24}$$

where the last equality uses $\mathrm{TV}(p, q) = \frac{1}{2} \int |dp - dq| = \frac{1}{2} \int |h - 1|\, dq$. □

Equipped with this lemma, we now state the first part of the theorem precisely.

Theorem D.2 (Mutual Information Collapse)**.** *Let $(\mathcal{Y}, \mathcal{B}(\mathcal{Y}))$ be a measurable space, let $(\mathcal{Z}, P_Z)$ and $(\mathcal{E}, P_\epsilon)$ be probability spaces, let $\Theta \subseteq \mathbb{R}^d$, and let $g_\theta : \mathcal{X} \times \mathcal{Z} \times \mathcal{E} \to \mathcal{Y}$ be measurable for each $\theta \in \Theta$. Fix $x \in \mathcal{X}$, define the aleatoric conditional $p_\theta(\cdot \mid x, z) := (g_\theta(x; z, \cdot))_\# P_\epsilon$ and the predictive marginal $p_\theta(\cdot \mid x) := \int p_\theta(\cdot \mid x, z)\, dP_Z(z)$. Suppose the following conditions hold.*

- *A1* ***Well-specification.*** *There exists $\theta^* \in \Theta$ such that $p_{\theta^*}(\cdot \mid x, z) = p^*(\cdot \mid x)$ for P_Z-almost every z.*
- *A2* ***Consistency.*** *$\hat{\theta}_n \to \theta^*$ in probability as $n \to \infty$.*
- *A3* ***Uniform continuity in total variation.*** *The map $\theta \mapsto p_\theta(\cdot \mid x, z)$ is continuous at θ^* in total variation, uniformly in z: for every $\varepsilon > 0$ there exists $\delta > 0$ such that $\|\theta - \theta^*\| < \delta$ implies $\sup_{z \in \mathcal{Z}} \mathrm{TV}(p_\theta(\cdot \mid x, z), p_{\theta^*}(\cdot \mid x, z)) < \varepsilon$.*

A4 ***Bounded likelihood ratio.*** *There exists $C \geq 1$ such that for all θ in a neighborhood of θ^* and P_Z-almost every z, the aleatoric conditional is absolutely continuous with respect to the predictive marginal and*

$$\frac{dp_\theta(\cdot \mid x, z)}{dp_\theta(\cdot \mid x)} \leq C \qquad p_\theta(\cdot \mid x)\text{-a.e.}$$

Then

$$I_{\hat{\theta}_n}(Y;\, Z \mid x) := \mathbb{E}_Z\Big[\mathrm{KL}\Big(p_{\hat{\theta}_n}(\cdot \mid x, Z) \,\Big\|\, p_{\hat{\theta}_n}(\cdot \mid x)\Big)\Big] \xrightarrow{p} 0 \quad \text{as } n \to \infty.$$

Before proving this theorem, we describe what assumptions A1–A4 intuitively mean. Assumption A1 relates to *universality* of the chosen model class: the neural network architecture should be expressive enough to contain a parameter θ^* that captures the desired map, and is a standard assumption in the machine learning literature, having been proved for many choices of architectures via universal approximation theorems [31, 32]. Assumption A2 relates to well-posedness of training: under infinite data, it should be possible to recover the true optimum of the problem. Assumption A3 is a regularity condition requiring that the aleatoric conditionals respond smoothly to parameter perturbations, uniformly across the epistemic index. Assumption A4 requires a uniform upper bound on the likelihood ratio $dp_\theta(\cdot \mid x, z)/dp_\theta(\cdot \mid x)$; for finite ensembles with prior weights $w_z > 0$, this is automatically satisfied with $C = 1/\min_z w_z$, since the marginal $p_\theta(\cdot \mid x) = \sum_z w_z\, p_\theta(\cdot \mid x, z) \geq w_z\, p_\theta(\cdot \mid x, z)$ dominates each conditional. For continuous epistemic indices, A4 is a mild regularity condition excluding pathological concentration of the conditional relative to the marginal.

Proof. Fix $x \in \mathcal{X}$ and write $I_\theta := I_\theta(Y; Z \mid x)$, $p_\theta(z) := p_\theta(\cdot \mid x, z)$, and $p_\theta := p_\theta(\cdot \mid x)$ for brevity. The strategy is to show that $\theta \mapsto I_\theta$ is continuous at θ^* with $I_{\theta^*} = 0$, after which the conclusion follows from A2 by the continuous mapping theorem.

By A1, $p_{\theta^*}(z) = p^*(\cdot \mid x)$ for P_Z-a.e. z, and marginalizing gives

$$\begin{aligned} p_{\theta^*} = \int p_{\theta^*}(z)\, dP_Z(z) &= \int p^*(\cdot \mid x)\, dP_Z(z) \\ &= p^*(\cdot \mid x), \end{aligned} \tag{25}$$

so that $p_{\theta^*}(z) = p_{\theta^*}$ for P_Z-a.e. z. Consequently $\mathrm{KL}(p_{\theta^*}(z) \,\|\, p_{\theta^*}) = 0$ for P_Z-a.e. z, and integrating against P_Z yields $I_{\theta^*} = 0$.

For continuity at θ^*, fix $\varepsilon > 0$. Since $p_\theta(\cdot \mid x)$ is a mixture over P_Z with $p_\theta(\cdot \mid x, z) \ll p_\theta(\cdot \mid x)$, Lemma D.1 together with A4 gives, for every θ in the neighborhood specified by A4 and P_Z-a.e. z,

$$\mathrm{KL}(p_\theta(z) \,\|\, p_\theta) \leq 2M_C \cdot \mathrm{TV}(p_\theta(z), p_\theta), \tag{26}$$

where M_C is the constant from equation 22. It therefore suffices to bound $\mathbb{E}_Z[\mathrm{TV}(p_\theta(Z), p_\theta)]$. The triangle inequality for total variation, applied with $p_{\theta^*}(z)$ and p_{θ^*} as intermediate points, gives

$$\begin{aligned} \mathrm{TV}(p_\theta(z), p_\theta) \leq\ & \mathrm{TV}(p_\theta(z), p_{\theta^*}(z)) \\ & + \mathrm{TV}(p_{\theta^*}(z), p_{\theta^*}) \\ & + \mathrm{TV}(p_{\theta^*}, p_\theta), \end{aligned} \tag{27}$$

in which the middle term vanishes for P_Z-a.e. z by equation 25. By A3 there exists $\delta > 0$ such that $\|\theta - \theta^*\| < \delta$ implies

$$\sup_{z \in \mathcal{Z}} \mathrm{TV}(p_\theta(z), p_{\theta^*}(z)) < \frac{\varepsilon}{4M_C}, \tag{28}$$

controlling the first term in equation 27 uniformly in z. For the third term, joint convexity of total variation together with the representation $p_\theta = \int p_\theta(z)\, dP_Z(z)$ yields

$$\begin{aligned} \mathrm{TV}(p_{\theta^*}, p_\theta) &\leq \int \mathrm{TV}(p_{\theta^*}(z), p_\theta(z))\, dP_Z(z) \\ &< \frac{\varepsilon}{4M_C}, \end{aligned} \tag{29}$$

where the strict bound again follows from equation 28 at the same δ. Integrating equation 27 against P_Z and substituting equation 28 and equation 29 gives

$$\begin{aligned}\mathbb{E}_Z[\mathrm{TV}(p_\theta(Z), p_\theta)] &< \frac{\varepsilon}{4M_C} + 0 + \frac{\varepsilon}{4M_C} \\ &= \frac{\varepsilon}{2M_C},\end{aligned} \tag{30}$$

which combined with equation 26 yields

$$I_\theta \leq 2M_C \cdot \mathbb{E}_Z[\mathrm{TV}(p_\theta(Z), p_\theta)] < 2M_C \cdot \frac{\varepsilon}{2M_C} = \varepsilon. \tag{31}$$

Hence $\theta \mapsto I_\theta$ is continuous at θ^*.

Since $\hat{\theta}_n \xrightarrow{p} \theta^*$ by A2 and I_θ is continuous at θ^* with $I_{\theta^*} = 0$, the continuous mapping theorem gives $I_{\hat{\theta}_n} \xrightarrow{p} I_{\theta^*} = 0$. □

D.2 Aleatoric Entropy Convergence

Theorem D.2 establishes that the epistemic mutual information collapses to zero. We now show that the aleatoric uncertainty estimator converges to the entropy of the true conditional. The argument follows the same three-step template as Theorem D.2: point evaluation at θ^*, continuity at θ^*, and the continuous mapping theorem. The new ingredient is a regularity condition that does not follow from A3 alone.

Assumption A3 controls $\theta \mapsto p_\theta(\cdot \mid x, z)$ in total variation, but differential entropy is not continuous in TV: distributions can be arbitrarily TV-close with entropies that differ by any prescribed amount. We therefore add the following assumption.

A5 **Uniform entropy continuity.** The map $\theta \mapsto H(p_\theta(\cdot \mid x, z))$ is continuous at θ^*, uniformly in z: for every $\varepsilon > 0$ there exists $\delta > 0$ such that $\|\theta - \theta^*\| < \delta$ implies $\sup_{z \in \mathcal{Z}} |H(p_\theta(\cdot \mid x, z)) - H(p_{\theta^*}(\cdot \mid x, z))| < \varepsilon$.

For Gaussian-mixture conditionals of the type used in Section 4.1, A5 follows from continuous parametrisation of the mixture weights, means, and covariances together with a uniform lower bound on component covariance eigenvalues in a neighborhood of θ^*. For finite ensembles, the supremum over z reduces to a maximum over finitely many continuous functions and is automatic.

Theorem D.3 (Aleatoric Entropy Consistency)**.** *Adopt the setup of Theorem D.2 and suppose A1, A2, and A5 hold. Then*

$$A_{\hat{\theta}_n}(x) := \mathbb{E}_Z\Big[H_\epsilon\Big(Y \mid Z, x; \hat{\theta}_n\Big)\Big] \xrightarrow{p} H(p^*(\cdot \mid x)) \quad \textit{as } n \to \infty.$$

Proof. Fix $x \in \mathcal{X}$ and write $A_\theta := \mathbb{E}_Z[H(p_\theta(\cdot \mid x, Z))]$ and $p_\theta(z) := p_\theta(\cdot \mid x, z)$ for brevity. The strategy mirrors Theorem D.2: show that $\theta \mapsto A_\theta$ is continuous at θ^* with $A_{\theta^*} = H(p^*(\cdot \mid x))$, then apply the continuous mapping theorem under A2.

By A1, $p_{\theta^*}(z) = p^*(\cdot \mid x)$ for P_Z-a.e. z, so $H(p_{\theta^*}(z)) = H(p^*(\cdot \mid x))$ for P_Z-a.e. z. Integrating against P_Z gives

$$A_{\theta^*} = H(p^*(\cdot \mid x)). \tag{32}$$

For continuity at θ^*, fix $\varepsilon > 0$. By A5 there exists $\delta > 0$ such that $\|\theta - \theta^*\| < \delta$ implies

$$\sup_{z \in \mathcal{Z}} |H(p_\theta(z)) - H(p_{\theta^*}(z))| < \varepsilon. \tag{33}$$

Applying the triangle inequality for integrals followed by equation 33 yields

$$\begin{aligned}|A_\theta - A_{\theta^*}| &\leq \int |H(p_\theta(z)) - H(p_{\theta^*}(z))| \, dP_Z(z) \\ &\leq \sup_{z \in \mathcal{Z}} |H(p_\theta(z)) - H(p_{\theta^*}(z))| \\ &< \varepsilon,\end{aligned} \tag{34}$$

so $\theta \mapsto A_\theta$ is continuous at θ^*.

Since $\hat{\theta}_n \xrightarrow{p} \theta^*$ by A2 and $\theta \mapsto A_\theta$ is continuous at θ^*, the continuous mapping theorem combined with equation 32 gives $A_{\hat{\theta}_n} \xrightarrow{p} H(p^*(\cdot \mid x))$. □

Theorems D.2 and D.3 together imply that the total predictive entropy converges to the entropy of the true conditional, which gives a complete consistency picture for the entropy decomposition equation 7.

Corollary D.4 (Total Predictive Entropy Consistency)**.** *Under A1–A5,*

$$H_{Z,\epsilon}(Y \mid x;\, \hat{\theta}_n) \xrightarrow{p} H(p^*(\cdot \mid x)) \quad \textit{as } n \to \infty.$$

Proof. By the entropy decomposition equation 7,

$$H_{Z,\epsilon}(Y \mid x;\, \hat{\theta}_n) = A_{\hat{\theta}_n}(x) + I_{\hat{\theta}_n}(Y;\, Z \mid x).$$

Theorem D.3 gives $A_{\hat{\theta}_n}(x) \xrightarrow{p} H(p^*(\cdot \mid x))$ and Theorem D.2 gives $I_{\hat{\theta}_n}(Y;\, Z \mid x) \xrightarrow{p} 0$. The continuous mapping theorem applied to the sum yields the result. □

E Proof of Theorem 3.1

Proof. We wish to show that *MI-LB*$(x) \leq I(Y; Z \mid x)$ for all $x \in \mathcal{X}$. Recalling the decomposition from (9),

$$I(Y;\, Z \mid x) = H(Y \mid x) - \mathbb{E}_Z\big[H(Y \mid x, Z)\big]. \tag{35}$$

We bound the two terms on the right-hand side in opposite directions using the results of [4].

Step 1: Lower bounding $H(Y \mid x)$.

The marginal predictive distribution $p_\theta(\cdot \mid x)$ is the Gaussian mixture with $n_{\text{ens}} \cdot K$ components given in equation 11. By the lower bound of [4],

$$H(Y \mid x) \geq H_{\text{lower}}(Y \mid x), \tag{36}$$

where $H_{\text{lower}}(Y \mid x)$ is equation 12 applied to the marginal mixture with weights $\beta_{z,i}$, means $\mu_i^{(z)}$, and covariances $C_i^{(z)}$.

Step 2: Upper bounding $\mathbb{E}_Z[H(Y \mid x, Z)]$.

For each fixed $z \in \{1, \ldots, n_{\text{ens}}\}$, the aleatoric conditional $p_\theta(\cdot \mid x, z)$ is a Gaussian mixture with K components as in equation 10. By the upper bound of [4],

$$H(Y \mid x, Z = z) \leq H_{\text{upper}}(Y \mid x, Z = z) \qquad \text{for each } z. \tag{37}$$

Multiplying by $w_z \geq 0$ and summing over z,

$$\mathbb{E}_Z\big[H(Y \mid x, Z)\big] = \sum_{z=1}^{n_{\text{ens}}} w_z\, H(Y \mid x, Z = z) \leq \sum_{z=1}^{n_{\text{ens}}} w_z\, H_{\text{upper}}(Y \mid x, Z = z). \tag{38}$$

Step 3: Combining the bounds.

Substituting equation 36 and equation 38 into equation 35, and using the fact that $f(a, b) = a - b$ is non-decreasing in a and non-increasing in b,

$$\begin{aligned} I(Y;\, Z \mid x) &= H(Y \mid x) - \mathbb{E}_Z\big[H(Y \mid x, Z)\big] \\ &\geq H_{\text{lower}}(Y \mid x) - \sum_{z=1}^{n_{\text{ens}}} w_z\, H_{\text{upper}}(Y \mid x, Z = z) \;=\; \textit{MI-LB}(x). \end{aligned}$$ □

F Experimental Details

F.1 Software

Our code is implemented in JAX [24] using the Flax [25] and Optax [33] libraries to define and train our neural networks. We used the `JaxMix` library [10] as a starting point for working with MDN ensembles. Our complete codebase is available at `https://github.com/PredictiveIntelligenceLab/JaxMix-AL`.

F.2 Synthetic Multimodal Conditional Problem

This appendix specifies the full data-generating process, model, trainer, and active-learning protocol used in Section 4.1. All hyperparameters listed here are the values used to produce Figures 1a and 6 and are exposed as defaults in `examples/multimodal_conditional/experiment_config.py`.

Input manifold. Latent codes $l \in \mathbb{R}^L$ are drawn $l \sim \mathcal{N}(0, I_L)$ and embedded into the input space via

$$x = \tanh\big(A\, l + b_m\big), \qquad A \in \mathbb{R}^{D\times L},\ A_{ij} \overset{\text{iid}}{\sim} \mathcal{N}\big(0, 1/L\big), \qquad b_m \sim \mathcal{N}(0, I_D).$$

Both A and b_m are drawn once using `manifold_seed` $= 1$ and held fixed across the benchmark; the $\tanh$ bounds x to $[-1, 1]^D$.

Random Fourier feature map. All mixture parameters are input-dependent through a fixed random Fourier feature map

$$h(x) = \cos\big(\Omega\, x + \varphi\big) \in \mathbb{R}^P, \qquad \Omega_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/D), \quad \varphi_i \overset{\text{iid}}{\sim} \text{Uniform}(0, 2\pi).$$

The parameters (Ω, φ) are drawn once with `dist_seed` $= 42$ and held fixed.

Per-component means and variances. For each of the K mixture components, uncentered means and log-variances are

$$\tilde{\mu}_k(x) = B_k\, h(x) + c_k, \qquad \log \Sigma_k(x) = C_k\, h(x) + b_k^{\text{var}},$$

with $B_k, C_k \in \mathbb{R}^{M\times P}$ having i.i.d. $\mathcal{N}(0, 1/P)$ entries, $c_k \sim \mathcal{N}(0, c_{\text{scale}}^2\, I_M)$, and $b_k^{\text{var}} = 0$. The per-component offsets c_k control how far apart the modes sit in output space. Component means are then centered to enforce $\mathbb{E}[y \mid x] = 0$:

$$\mu_k(x) = \tilde{\mu}_k(x) - \sum_{j=1}^{K} \pi_j(x)\, \tilde{\mu}_j(x),$$

and the covariance is diagonal, $\Sigma_k(x) = \text{diag}(\exp \log \Sigma_k(x))$. The variance-coupling coefficient α in the code is set to 0, so log-variances do not depend on π_k.

Structured mixing weights. To create a clean unimodal/multimodal phase boundary, we use the `structured` mixing mode. Letting $r = \|x_{1:L}\|$, define a radial gate and $K-1$ angular scores,

$$\begin{aligned} g(x) &= \tfrac{1}{2}\big(1 + \tanh\big(\beta\,(r - r_0)\big)\big), \\ s(x) &= \text{softmax}\big(\gamma\, V\, x_{1:L}\big) \in \Delta^{K-2}, \qquad V \in \mathbb{R}^{(K-1)\times L},\ V_{k,:}/\|V_{k,:}\| = V_{k,:}, \end{aligned}$$

where the rows of V are unit vectors drawn from $\mathcal{N}(0, I_L)$ and normalised, and then assemble the logits

$$\ell_0(x) = \text{scale} \cdot (1 - g(x)), \qquad \ell_k(x) = \text{scale} \cdot g(x) \cdot s_k(x) \ \ (k = 1, \ldots, K-1),$$

and set $\pi(x) = \text{softmax}(\ell(x))$. Inside the radius ($r < r_0$) the mass concentrates on component 0, making $p^*(y \mid x)$ effectively unimodal; outside the radius the angular scores activate components $1, \ldots, K-1$ in sectors.

Hyperparameters. Table 3 lists the distribution hyperparameters and Table 4 the model and active-learning hyperparameters. All values match the defaults in `experiment_config.py`.

Table 3: Distribution hyperparameters for the multimodal conditional problem.

Symbol	Value	Meaning
D	10	Input dimension
M	16	Output dimension
L	4	Latent manifold dimension
K	3	Number of true mixture components
P	128	Random Fourier features
c_{scale}	10.0	Std. of per-component offsets c_k
α	0.0	Variance-coupling coefficient (disabled)
τ	1.0	Softmax temperature (unused in structured mode)
β	8.0	Transition sharpness
r_0	1.3	Transition radius
γ	2.0	Angular sharpness
scale	3.0	Logit magnitude
`dist_seed`	42	PRNG seed for distribution parameters
`manifold_seed`	1	PRNG seed for manifold map (A, b_m)

Oracle NLL. Because $p^*(y \mid x)$ is known in closed form, the oracle negative log-likelihood $\text{NLL}^* = -\mathbb{E}_{(x,y)\sim p^*}[\log p^*(y \mid x)]$ can be evaluated exactly on the test set. Under the settings above it equals $\text{NLL}^* = 22.98$ (computed by `compute_true_nll` in `examples/multimodal_conditional/utils.py`).

Model architecture. Each ensemble member is a Mixture Density Network with shared backbone: a 2-layer MLP with 128 hidden units per layer and GELU activations [34]. The MLP output is linearly projected to $K_{\text{MDN}} = 5$ mixture components, each with a mean in $\mathbb{R}^M$ and a diagonal covariance; mixture weights come from a softmax head. Ensemble size is $n_{\text{ens}} = 8$, with members initialised from different PRNG seeds and trained independently on the same labeled set (the standard deep-ensemble construction used to instantiate $P_Z = \text{Uniform}\{1, \ldots, n_{\text{ens}}\}$).

Training schedule. All members are trained with AdamW (peak learning rate 5×10^{-4}, weight decay 10^{-2}) preceded by adaptive gradient clipping at 0.1. The learning rate follows a linear warmup over $\min(500,\ n_{\text{iter}}/5)$ steps to the peak value, after which it decays exponentially at rate 0.9 per $2{,}000$ steps. The number of gradient steps per active learning round is set adaptively to

$$n_{\text{iter}}(n_{\text{lab}}) = \min\big(10{,}000,\ 10 \cdot n_{\text{lab}}\big),$$

where n_{lab} is the current labeled-set size; this prevents overfitting on small labeled sets in early rounds while allowing full optimisation once $n_{\text{lab}} \geq 1000$. The training mini-batch size is 128 (full-batch when $n_{\text{lab}} < 128$). Optimiser, architecture, and active-learning hyperparameters are fixed across all acquisition functions; we do not perform per-method tuning.

Active-learning protocol. For each of 5 seeds we instantiate a fresh pool of $50{,}000$ candidate inputs, a held-out test set of $2{,}000$ inputs, and an initial labeled set of 100 inputs. Acquisition scores are evaluated on the remaining pool in chunks of 256. Each active-learning run performs 20 rounds, acquiring 50 queries per round, so the final labeled set contains $100 + 20 \cdot 50 = 1100$ examples, i.e. 2.2% of the pool. Test NLL is evaluated on the held-out set at the end of every round.

Acquisition functions. Five scoring functions are evaluated. **Random** draws scores i.i.d. $\text{Uniform}(0, 1)$. **Epistemic Variance** uses the trace of the covariance of conditional means across ensemble members. **MI-LB** is equation 15. **BAIT** [13] uses the paper's last-layer mean-head Fisher of ensemble member 0 with one MC sample $y \sim p_\theta(\cdot \mid x)$ per pool point. **Core-Set** [12] runs k-Center-Greedy on the shared MDN backbone activations of ensemble member 0 (the "final FC layer" recipe of §4.4). The latter two produce no scalar score and bypass the score-to-batch step.

Implementation of BAIT and Core-Set in the MDN setting. Both baselines are adapted from their original classification specifications. **BAIT** requires per-input Fisher embeddings $G(x)$ such that $F(x) = G(x)^\top G(x)$ approximates the per-sample Fisher information; following the last-layer recipe

Table 4: Model and active-learning hyperparameters for the multimodal conditional problem.

Symbol / option	Value	Meaning
n_{ens}	8	Ensemble size
K_{MDN}	5	MDN mixture components per member
hidden features	128	MLP width
depth	2	Number of MLP hidden layers
activation	GELU	Nonlinearity
training batch size	128	Mini-batch size for AdamW
peak learning rate	5×10^{-4}	AdamW peak LR (after warmup)
weight decay	10^{-2}	AdamW weight decay
gradient clip	0.1	Adaptive gradient clipping threshold
n_{iter} (cap)	10,000	Max gradient steps per round
iter-per-sample	10	Slope of adaptive schedule
candidate pool size	50,000	Unlabeled pool $\lvert\mathcal{X}_{\text{pool}}\rvert$
test set size	2,000	Held-out evaluation set
initial labelled	100	Initial labeled budget
AL rounds	20	Number of acquisition rounds
query batch size	50	Queries acquired per round
acquisition batch size	256	Chunk size for scoring the pool
seeds	$\{0, 1, 2, 3, 4\}$	Data / training seeds for reported curves

of [13] we restrict the Fisher to the mean-head weights of ensemble member 0. Drawing one Monte-Carlo sample $y \sim p_\theta(\cdot \mid x)$ and using the closed-form MDN log-likelihood gradient $\nabla_{W_\mu} \log p(y \mid x; \theta) = \gamma_k(y, x)\,(y - \mu_k(x))/\sigma_k^2(x) \otimes z(x)$, where γ_k are posterior mixture responsibilities and $z(x)$ is the shared backbone activation, the embedding is the flattened outer product, $G(x) \in \mathbb{R}^{1\times hKd}$. Selection minimises $\mathrm{tr}((F_{\text{train}} + \lambda I)^{-1} F_{\text{cand}})$ via the forward+backward greedy of [13] with ridge $\lambda = 10^{-3}$ and the labelled-set Fisher as the burden matrix. This single-ensemble, single-MC choice is a compute trade-off rather than a capacity claim: extending the Fisher embedding to all $n_{\text{ens}} = 8$ members (to capture cross-member disagreement and the mixture-weight signal the mean-head Fisher omits) or to multiple MC samples per pool point would multiply an already dominant selection overhead — BAIT alone consumes $\sim$12 of the $\sim$25 GPU-h total compute budget (Table 13), pushing a richer embedding into a 100+ GPU-h regime no other acquisition here requires. We therefore report the canonical last-layer recipe of [13] and flag the resulting Fisher-estimate variance as a known limitation in the empirical discussion below. **Core-Set** uses the activations of the shared MDN backbone of ensemble member 0 (the layer immediately before the mixture head) as h-dimensional features, then runs k-Center-Greedy [12] on Euclidean distances initialised at the labelled set. Both methods consume the full pool without subsampling.

Selection strategies. Three batch-selection rules convert per-point scores into a query batch of size 50:

- **Top-k (greedy).** Select the 50 points with the highest acquisition scores. Used for all main-text results.
- **SBAL [14].** We use the *softmax* variant of SBAL (as opposed to the softrank and power variants in Kirsch et al.), which handles negative scores without modification: sample 50 points without replacement from $\mathrm{softmax}(\mathrm{score}/T)$ via the Gumbel-top-k trick. The temperature T interpolates between exploitation ($T \to 0$, recovers top-k) and uniform exploration ($T \to \infty$, recovers Random). All SBAL runs reported here use $T = 1.0$.
- **MaxDist [11].** Acquisition-weighted farthest-point sampling in feature space (LCMD-TP variant). Given input features x standardised per dimension and normalised scores $\tilde{s} \in [0, 1]$, greedily pick the point maximising $d_{\min}(i) \cdot (1 + w\,\tilde{s}(i))$, where $d_{\min}(i)$ is the squared distance from point i to its nearest already-selected or already-training point. All MaxDist runs use score-weight $w = 1$. At $w = 0$ this reduces to pure farthest-point sampling, and as $w \to \infty$ it approaches top-k on $\tilde{s}$.

SBAL and MaxDist only apply to the deterministic acquisitions (Epistemic Variance and MI-LB); combining either with Random is undefined (Random has no meaningful score ranking or signal to weight against). Any acquisition of the form `sbal_X` or `maxdist_X` in the experiment log corresponds to applying the named selection strategy on top of base score X. BAIT and Core-Set bypass the score-to-batch step entirely (Fisher-trace forward+backward greedy and k-Center-Greedy respectively) and have no SBAL/MaxDist variants.

Results for SBAL and MaxDist variants. Figure 5 and Table 5 report final test NLL at $n = 1100$ across 5 seeds. At $T = 1.0$, MI-LB (SBAL) sits roughly halfway between MI-LB (top-k) and Random (33.99 vs 31.12 and 39.73), because softmax sampling at $T = 1.0$ places non-trivial mass outside the high-score region near the radial gate; lowering T tightens the distribution back onto MI-LB (top-k) and recovers it exactly as $T \to 0$. MI-LB (MaxDist) at $w = 1$ matches MI-LB (top-k) (30.49 vs 31.12), adding batch diversity without sacrificing score quality; larger w moves it toward top-k on the score, smaller w toward pure farthest-point sampling.

BAIT and Core-Set. Core-Set ties MI-LB within seed-to-seed noise (30.57 ± 0.56 vs 31.12 ± 0.34); MI-LB retains the tighter spread. The tie reflects benchmark geometry: pool inputs lie on a 4-D tanh manifold, so k-Center-Greedy spreads queries across that manifold and ends up sampling the gate region that MI-LB targets through entropy disagreement. BAIT effectively ties Variance (32.26 ± 1.55 vs 32.56 ± 1.14), but with the largest seed-to-seed spread of any acquisition we evaluate — the variance of the single-MC Fisher estimate at small budgets. Both baselines bypass the Two-Index decomposition; the next two appendices test whether the apparent tie generalises beyond a benchmark in which input geometry already encodes the multimodality.

Table 5: Final test NLL at $n = 1100$, mean $\pm$ std across 5 seeds (min–max in brackets). SBAL uses temperature $T = 1.0$; MaxDist uses score weight $w = 1$. Oracle NLL$^\star = 22.98$.

Acquisition	Mean $\pm$ std	Min – Max
Random	39.73 ± 0.90	38.89 – 40.97
Epistemic Variance (top-k)	32.56 ± 1.14	31.41 – 34.14
Epistemic Variance (SBAL)	31.01 ± 0.57	30.43 – 31.61
MI-LB (top-k)	31.12 ± 0.34	30.58 – 31.37
MI-LB (SBAL)	33.99 ± 0.74	33.21 – 35.01
MI-LB (MaxDist)	30.49 ± 0.46	29.84 – 30.91
BAIT	32.26 ± 1.55	30.40 – 34.34
Core-Set	$\mathbf{30.57 \pm 0.56}$	29.97 – 31.31

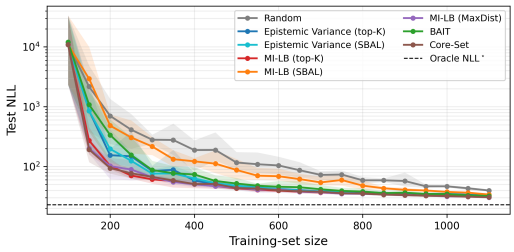


Figure 5: Learning curves on the multimodal benchmark for SBAL ($T = 1.0$) and MaxDist ($w = 1$) variants of Variance and MI-LB, plus the BAIT and Core-Set baselines, alongside the three top-k curves from Fig. 1a (5 seeds; bands min–max).

F.3 Coupled Double-Well System

This appendix specifies the full data-generating process, model, trainer, and active-learning protocol used in Section 4.2.

Simulator. $P = 5$ particles evolve according to the overdamped Langevin SDE equation 19 under open boundary conditions: particle 1 couples only to particle 2, particle P only to particle $P-1$, and

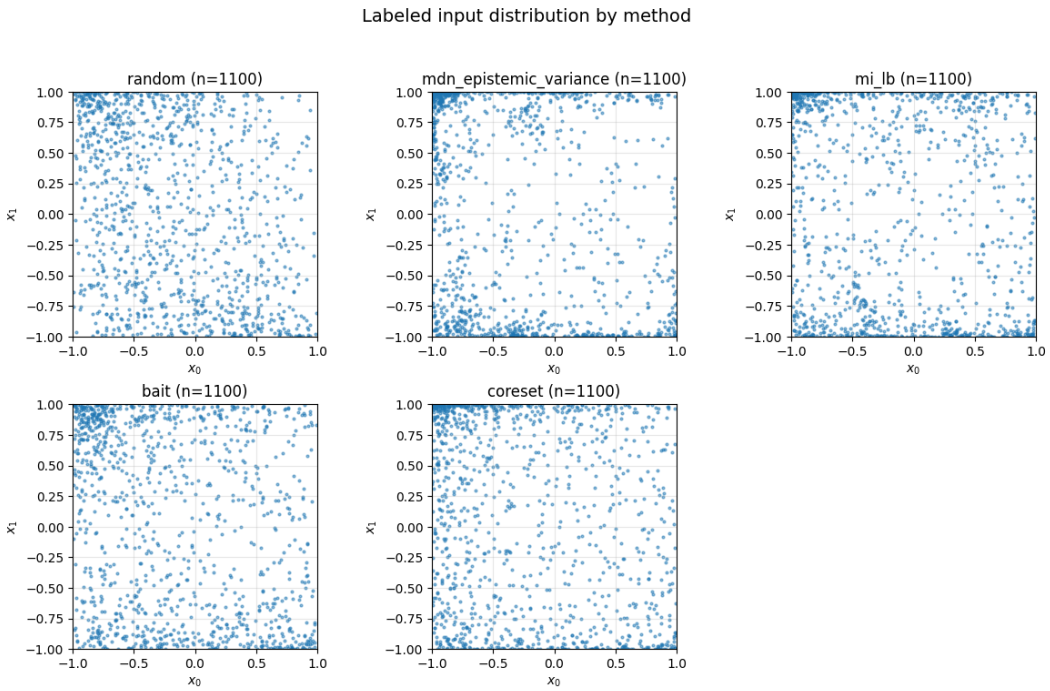


Figure 6: Spatial distribution of all labeled inputs (initial + acquired) at $n = 1100$ projected onto (x_0, x_1) for the five base acquisitions: Random, Epistemic Variance, and MI-LB **(top row)**, and the BAIT and Core-Set baselines **(bottom row)**. By the end of the budget the four informative acquisitions look broadly similar — all concentrate mass along the multimodal edge of the pool ($x_0 \to -1$) and the upper/lower boundary — while Random remains the clear outlier, spreading queries uniformly across the support. All five panels inherit the same off-center pool support from the fixed bias b in $x = \tanh(Al + b)$ (drawn once at benchmark instantiation; see *Input manifold* above).

interior particles to both neighbours. Barrier heights are uniform, $a = 1$, giving per-particle potential $V(q) = q^4/4 - q^2/2$ with minima at $q = \pm 1$ and barrier height $a/4 = 0.25$. We integrate with the Euler–Maruyama scheme

$$q_i^{(t+dt)} = q_i^{(t)} + \left[a\,(q_i^{(t)} - (q_i^{(t)})^3) + \kappa \sum_{j \in \mathrm{nn}(i)} (q_j^{(t)} - q_i^{(t)})\right] dt + \sigma\sqrt{dt}\,\eta_i^{(t)}, \qquad \eta_i^{(t)} \sim \mathcal{N}(0,1),$$

with $dt = 0.005$ and terminal time $T = 5.0$ (i.e. $n_{\mathrm{steps}} = 1000$).

Inputs and outputs. Each trajectory is summarised by the 7-dimensional input $x = (q_1(0), \ldots, q_5(0),\, \sigma,\, \kappa) \in \mathbb{R}^{P+2}$, drawn component-wise from

$$q_i(0) \sim \mathrm{Uniform}(-1.5, 1.5), \quad \sigma \sim \mathrm{Uniform}(0.3, 2.0), \quad \kappa \sim \mathrm{Uniform}(0, 3.0).$$

The output is the concatenation of $n_{\mathrm{snap}} = 4$ snapshots of the particle positions evenly spaced in $[T/n_{\mathrm{snap}}, T]$ (i.e. at $t = 1.25, 2.5, 3.75, 5.0$), flattened into $y \in \mathbb{R}^{n_{\mathrm{snap}} \cdot P} = \mathbb{R}^{20}$. Multi-snapshot outputs give the model direct access to trajectory-level structure and substantially improve training stability compared to single-endpoint outputs, at no additional simulation cost.

Hyperparameters. Table 6 lists the simulator and data hyperparameters and Table 7 the model and active-learning hyperparameters.

Model architecture. Each ensemble member is an MDN with a 3-hidden-layer MLP backbone, 128 hidden units per layer, GELU activations, and a final linear projection to $K_{\mathrm{MDN}} = 8$ mixture components with diagonal covariances over the 20-dimensional output. The ensemble size is $n_{\mathrm{ens}} = 8$, with members initialised from different PRNG seeds and trained independently on the same labeled set.

Table 6: Simulator and data hyperparameters for the coupled double-well benchmark.

Symbol / option	Value	Meaning
P	5	Number of coupled particles
a	1 (uniform)	Barrier-height coefficient
T	5.0	Terminal integration time
dt	0.005	Euler–Maruyama step (1000 steps)
n_{snap}	4	Snapshots recorded per trajectory
boundary	open	Chain ends couple only to one neighbour
$q_i(0)$ range	$[-1.5,\ 1.5]$	Initial-configuration prior
σ range	$[0.3,\ 2.0]$	Noise-intensity prior
κ range	$[0,\ 3.0]$	Coupling-strength prior

Training schedule. All members are trained with AdamW using the same optimiser configuration as in Appendix F.2: peak learning rate 5×10^{-4}, weight decay 10^{-2}, linear warmup followed by exponential decay at rate 0.9 per $2{,}000$ steps, and adaptive gradient clipping at 0.1. The number of gradient steps per round is set adaptively to $n_{\text{iter}}(n_{\text{lab}}) = \min(10{,}000,\ 10 \cdot n_{\text{lab}})$, with mini-batch size 128.

Active-learning protocol. For each of 5 seeds we instantiate a fresh pool of $50{,}000$ candidate inputs, a held-out test set of $2{,}000$ inputs, and an initial labeled set of 100 inputs. Acquisition scores are evaluated on the remaining pool in chunks of 256. Each run performs 20 rounds of 50 queries each, so the final labeled set contains $100 + 20 \cdot 50 = 1{,}100$ trajectories (2.2% of the pool). Test NLL is evaluated at the end of every round. Seeds $\{0, 1, 2, 3, 4\}$ are used for all reported curves.

Table 7: Model and active-learning hyperparameters for the coupled double-well benchmark.

Symbol / option	Value	Meaning
n_{ens}	8	Ensemble size
K_{MDN}	8	MDN mixture components per member
hidden features	128	MLP width
depth	3	Number of MLP hidden layers
activation	GELU	Nonlinearity
training batch size	128	Mini-batch size for AdamW
peak learning rate	5×10^{-4}	AdamW peak LR (after warmup)
weight decay	10^{-2}	AdamW weight decay
gradient clip	0.1	Adaptive gradient clipping threshold
n_{iter} (cap)	10,000	Max gradient steps per round
iter-per-sample	10	Slope of adaptive schedule
candidate pool size	50,000	Unlabeled pool $\lvert\mathcal{X}_{\text{pool}}\rvert$
test set size	2,000	Held-out evaluation set
initial labelled	100	Initial labeled budget
AL rounds	20	Number of acquisition rounds
query batch size	50	Queries acquired per round
acquisition batch size	256	Chunk size for scoring the pool
seeds	$\{0, 1, 2, 3, 4\}$	Data / training seeds

Acquisition functions and selection strategies. Five base acquisitions, defined in Appendix F.2 unchanged: Random, Variance, MI-LB, BAIT [13], Core-Set [12]. Selection strategies: top-k for Random / Variance / MI-LB; SBAL ($T = 1.0$) for Variance and MI-LB; MaxDist ($w = 1$) for MI-LB. BAIT and Core-Set bypass the score-to-batch step entirely.

Results for SBAL and MaxDist variants. Figure 7 and Table 8 report the full learning curves and the final test NLL at $n = 1100$ across 5 seeds. Two patterns are worth highlighting. First, MaxDist pairs well with MI-LB: MI-LB (MaxDist) at $w = 1$ finishes at 131 ± 15, within a factor of two of MI-LB (top-k) and comparable to Epistemic Variance (top-k). Acquisition-weighted farthest-point

sampling adds batch diversity without discarding the informative ranking. Second, SBAL severely degrades both base scores on this benchmark. MI-LB (SBAL) ends at 553 ± 92 and Epistemic Variance (SBAL) at $\mathbf{457} \pm \mathbf{46}$, both within striking distance of the Random baseline (518 ± 60) and roughly 4–8$\times$ worse than their respective top-k counterparts (122 and 71). This is the opposite of the multimodal-conditional result in Appendix F.2, where SBAL was competitive or helpful, and it reflects a quantitative rather than a qualitative difference: the total budget (1,100 points) is small relative to the pool, so softmax sampling at $T = 1.0$ spreads enough mass onto low-score points that the effective fraction of informative queries collapses toward the Random baseline; annealing T toward zero would tighten the distribution onto the top-k result and recover it in the limit $T \to 0$. This is consistent with the original observations of Kirsch et al. [14]: stochastic batch acquisition helps when the top of the score ranking is corrupted by approximation noise and diverse sampling hedges against that noise; it hurts when the top ranking is already approximately correct and exploration trades informative points for uninformative ones.

BAIT and Core-Set. Both geometric baselines fail decisively on this benchmark: Core-Set finishes at 304.3 ± 54.8 and BAIT at 281.3 ± 15.1, $\sim 4\times$ worse than MI-LB (70.8 ± 8.0) and within $1.7\times$ of Random (518.1 ± 60.1). Mode structure here lives in *output* space (Kramers escape between $q = \pm 1$): k-Center-Greedy spreads queries uniformly over (σ, κ) regardless of where the bimodal regime sits, and the single-MC last-layer Fisher embedding is dominated by within-mode noise. Their joint collapse is the strongest evidence in the paper that MI-LB's advantage is not subsumed by feature-space coverage or last-layer information geometry.

Table 8: Final test NLL at $n = 1100$, mean $\pm$ std across 5 seeds (min–max in brackets) for the coupled double-well benchmark. All SBAL runs use $T = 1.0$; MaxDist uses score weight $w = 1$.

Acquisition	Mean $\pm$ std	Min – Max
Random	518.1 ± 60.1	462.6 – 616.2
Epistemic Variance (top-k)	122.1 ± 11.3	109.1 – 138.5
Epistemic Variance (SBAL)	456.8 ± 45.7	392.9 – 515.3
MI-LB (top-k)	$\mathbf{70.8} \pm \mathbf{8.0}$	59.9 – 81.6
MI-LB (SBAL)	553.0 ± 91.9	423.5 – 630.2
MI-LB (MaxDist)	130.7 ± 14.9	113.5 – 147.0
BAIT	281.3 ± 15.1	268.8 – 305.3
Core-Set	304.3 ± 54.8	249.3 – 377.5

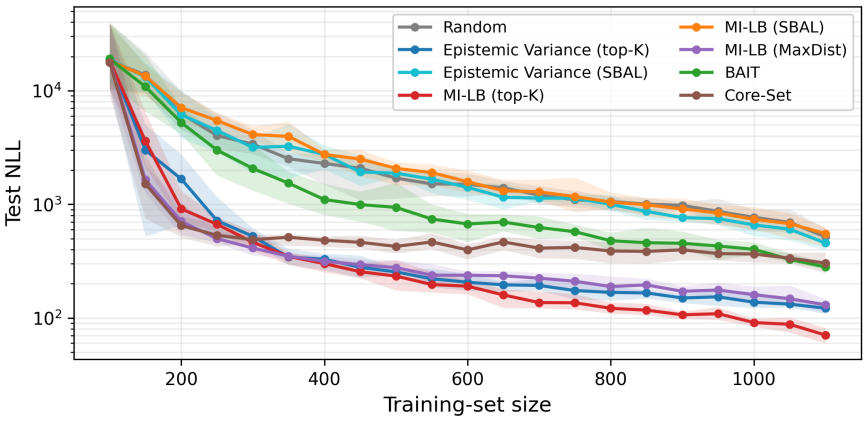


Figure 7: Learning curves for all eight (acquisition, selection-strategy) combinations on the coupled double-well benchmark including the BAIT and Core-Set baselines (5 seeds; bands min–max). MaxDist-MI-LB stays within $\sim 2\times$ of top-k MI-LB; both SBAL variants and the two geometric baselines (BAIT, Core-Set) collapse toward the Random regime.

Why a mixture head: $K = 1$ vs $K = 8$. To justify the choice of a mixture head ($K_{\text{MDN}} = 8$) over a single-Gaussian head ($K = 1$), we train both with the identical architecture used in the

AL experiments (depth-3 MLP, 128 hidden units, $n_{\text{ens}} = 8$) on a larger offline dataset of 200,100 trajectories drawn from the same input distribution as the active-learning experiments, and evaluate both on a held-out test set of 5,000 trajectories. The ensemble-averaged test NLL is $\mathbf{15.90}$ for $K = 8$ versus $\mathbf{21.41}$ for $K = 1$, a gap of 5.51 nats on the 20-dimensional output (roughly 0.28 nats per output dimension). Absolute NLL values here are below those in the AL experiments because training uses 200,100 examples rather than the 1,100 queried by the AL loop; the quantity of interest is the $K = 8$-vs-$K = 1$ gap, which is driven by the head structure, not the training-set size. The $K = 1$ model must place a single Gaussian per input, so wherever the conditional is bimodal it is forced to smear mass between the wells and pay an unavoidable log-likelihood penalty; the $K = 8$ mixture can allocate one component per well. Figure 8 shows the qualitative picture on the test distribution: ground-truth samples of $(q_0(T), q_1(T))$ concentrate near $(\pm 1, \pm 1)$ in the bimodal regime and the $K = 8$ MDN recovers the same multi-well envelope, while the $K = 1$ model collapses to a single broad ellipse near the origin. This is the failure mode anticipated in Section 4.2 and the quantitative reason the active-learning experiments on this benchmark use $K_{\text{MDN}} = 8$ throughout.

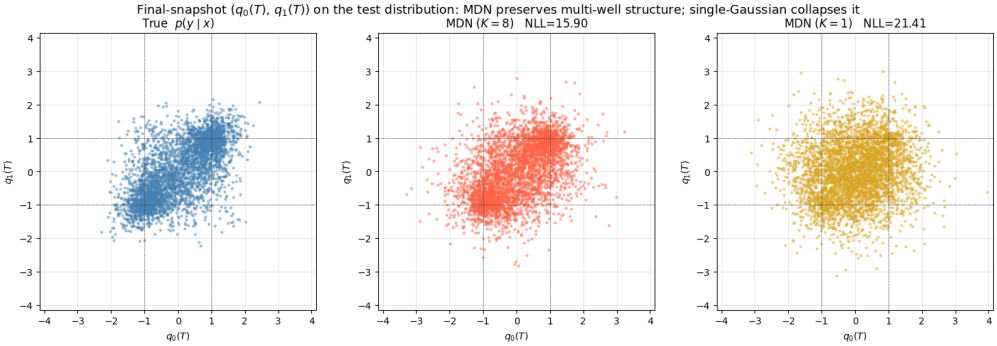


Figure 8: Coupled double-well benchmark: final-snapshot positions $(q_0(T), q_1(T))$ for 4,000 held-out test inputs drawn from the same joint distribution as the AL test set. **Left:** ground-truth simulator samples; mass concentrates near $(\pm 1, \pm 1)$, the signature of the per-particle bimodal regime at $\sigma \gtrsim \sqrt{a/2} \approx 0.71$, with a diagonal enhancement that reflects neighbour alignment under non-zero κ. **Middle:** samples from the $K = 8$ MDN ensemble (net 0); the mixture head preserves the same multi-well envelope (test NLL 15.90). **Right:** samples from a $K = 1$ single-Gaussian MDN trained with the same architecture and budget; per-input unimodality forces mass into a single broad ellipse centred near the origin (test NLL 21.41; +5.51 nats over $K = 8$). Dotted lines mark the well locations $q = \pm 1$.

F.4 Material Science Application: Alloy Phase Competition

This appendix details the synthetic phase-competition simulator and the active-learning protocol summarised in Section 4.3, and reports the SBAL and MaxDist batch-diversity variants deferred from the main text.

Phase model. The simulator implements a softmin-over-quadratic-free-energies model inspired by the CALPHAD framework [23]. For each of $N_\phi = 4$ nominal phases ϕ we draw a positive-definite 3×3 Hessian as $H_\phi = R_\phi R_\phi^\top + 0.3 \cdot I_3$ with R_ϕ a Gaussian matrix ($R_{\phi,ij} \sim \mathcal{N}(0, 0.5^2)$), and a linear bias $b_\phi \in \mathbb{R}^3$ with $b_{\phi,i} \sim \mathcal{N}(0, 2^2)$. Writing $x_3 = (x_A, x_B, x_C)$ for the composition coordinates extended onto the 3-simplex, the Gibbs free energy of phase ϕ is the quadratic form

$$G_\phi(x_3) = \tfrac{1}{2}\, x_3^\top H_\phi\, x_3 + b_\phi^\top x_3,$$

and the phase posterior at temperature τ_G is $\pi(\phi \mid x_3) \propto \exp(-G_\phi(x_3)/\tau_G)$. This posterior does not depend on the process parameters $p \in \mathbb{R}^{n_{\text{proc}}}$; phase assignment is a function of composition alone.

The conditional response mean for phase ϕ is

$$\mu_\phi(x_3, p) = c_\phi^\top x_3 + d_\phi + \tfrac{1}{2} \sin(\omega_\phi^\top x_3) + W_\phi^\top p,$$

with $c_\phi \in \mathbb{R}^3$, $d_\phi \in \mathbb{R}$, $\omega_\phi \in \mathbb{R}^3$, and $W_\phi \in \mathbb{R}^{n_{\text{proc}}}$ all drawn from independent Gaussians (entry-wise scales $c_{\mu\text{-scale}} = 6$ for c_ϕ, 2 for d_ϕ and ω_ϕ, 1 for W_ϕ). The per-phase log-variance is the affine

function $\log \sigma_\phi^2(x_3) = e_\phi^\top x_3 + f_\phi$ with $e_{\phi,i} \sim \mathcal{N}(0, 0.5^2)$ and $f_\phi \sim \mathcal{N}(-1, 0.3^2)$, so per-phase noise scales vary smoothly with composition around a typical $\sigma_\phi \approx 0.6$.

The output is sampled by first drawing $\phi \sim \pi(\cdot \mid x_3)$, then drawing $y \mid \phi, x_3, p \sim \mathcal{N}(\mu_\phi(x_3, p), \sigma_\phi^2(x_3))$; the ground-truth conditional $p^\star(y \mid x)$ is therefore a Gaussian mixture in y with composition-dependent mixing weights, composition- and-process-dependent component means, and composition-dependent component variances. The number of components with appreciable weight at any given x is determined by the realised phase structure of the chosen system seed, summarised below.

Inputs and outputs. The input is the 8-dimensional vector $x = (x_A, x_B, p_1, \ldots, p_6)$ with (x_A, x_B, x_C) drawn from a symmetric Dirichlet($\alpha = (1, 1, 1)$) on the 3-simplex and $p_i \sim \mathrm{Uniform}(-1, 1)$ component-wise. The output is the scalar response $y \in \mathbb{R}$.

Realised phase structure at the system seed. At the system seed used throughout (`system_seed` $= 12$, $N_\phi = 4$, $\tau_G = 0.08$), all four nominal phases carry appreciable mass on the simplex: a grid-evaluation of $\pi(\phi \mid x_3)$ on 20,301 uniform simplex points gives marginal phase masses 48.18%, 25.60%, 18.84%, 7.38% (sorted, summing to 100%). The boundary region defined by $\max_\phi \pi(\phi \mid x_3) < 0.7$ occupies 11.40% of the simplex slice; tightening to $\max_\phi \pi < 0.5$ contracts it to 1.14%, and relaxing to $\max_\phi \pi < 0.8$ expands it to 17.60%. The realised problem is thus a genuine 4-phase competition with curved boundaries between phase domains in the 8-dimensional input space. Inside any single phase domain the property response collapses to a single Gaussian whose mean varies smoothly with the 6-dimensional process subspace, while in the boundary region the conditional is a genuine multi-component mixture.

Model architecture. Each ensemble member is an MDN with a 2-hidden-layer MLP backbone, 64 hidden units per layer, GELU activations [34], and a final linear projection to $K_{\mathrm{MDN}} = 4$ mixture components with diagonal covariances over the scalar property output. The ensemble size is $n_{\mathrm{ens}} = 8$, with members initialised from different PRNG seeds and trained independently on the same labeled set.

Training schedule. All members are trained with AdamW (peak learning rate 2×10^{-4}, weight decay 5×10^{-2}) preceded by adaptive gradient clipping at 0.1, with the same warmup-then-exponential-decay schedule as in Appendix F.2. The number of gradient steps per round is set adaptively to $n_{\mathrm{iter}}(n_{\mathrm{lab}}) = \min(40{,}000,\ 200 \cdot n_{\mathrm{lab}})$ with a minimum of 2,000 steps per round, and mini-batch size 64. The smaller peak LR and stronger weight decay (relative to the multimodal and double-well benchmarks) reflect the larger per-round step budget; as in the other benchmarks, optimiser, architecture, and active-learning hyperparameters are fixed across all acquisition functions and are not tuned per method.

Hyperparameters. Tables 9 and 10 list the simulator/data and model/AL hyperparameters used for all 30 runs reported below.

Active-learning protocol. For each of 5 seeds we instantiate a fresh pool of 50,000 candidate inputs, a held-out test set of 2,000 inputs, and an initial labeled set of 100 inputs. Acquisition scores are evaluated on the remaining pool in chunks of 256. Each run performs 30 rounds of 15 queries each, so the final labeled set contains $100 + 30 \cdot 15 = 550$ inputs (1.1% of the pool). Test NLL is evaluated at the end of every round.

Acquisition functions and selection strategies. We evaluate the same five base acquisitions as in Appendix F.2 (Random, Epistemic Variance, MI-LB, BAIT [13], Core-Set [12]), paired with three selection strategies for the score-based ones: top-k for Random / Variance / MI-LB, SBAL for Variance and MI-LB, and MaxDist for MI-LB. SBAL temperature is $T = 1.0$; MaxDist score weight is $w = 1$. BAIT and Core-Set bypass the score-to-batch step entirely (Fisher-trace forward+backward greedy and k-Center-Greedy on the MDN backbone respectively). Definitions follow Appendix F.2 unchanged.

Seed-to-seed spread of the top-k base acquisitions. Section 4.3 reports that MI-LB is markedly more consistent across seeds than Random or Epistemic Variance in the data-scarce regime, with the

Table 9: Simulator and data hyperparameters for the ternary phase-competition benchmark.

Symbol / option	Value	Meaning
N_ϕ	4	Nominal phases (all four active at the chosen seed)
τ_G	0.08	Free-energy softmin temperature
n_{proc}	6	Process parameters per input
$c_{\mu\text{-scale}}$	6	Scale of per-phase composition coupling
H_ϕ raw scale	0.5	$R_{\phi,ij} \sim \mathcal{N}(0, 0.5^2)$
H_ϕ regulariser	$0.3 \cdot I_3$	PD floor on Hessian
b_ϕ scale	2	Linear-bias scale on composition
d_ϕ scale	2	Per-phase mean offset scale
ω_ϕ scale	2	Sinusoidal-modulation frequency scale
W_ϕ scale	1	Per-phase process-coupling scale
e_ϕ scale	0.5	Composition-dependence of $\log \sigma_\phi^2$
f_ϕ mean / std	-1 / 0.3	Per-phase log-variance offset
(x_A, x_B, x_C) prior	$\text{Dirichlet}(1, 1, 1)$	Composition prior on 3-simplex
p_i prior	$\text{Uniform}(-1, 1)$	Process-parameter prior
`system_seed`	12	PRNG seed for $H_\phi, b_\phi, c_\phi, \ldots$
candidate pool size	50,000	Unlabeled pool $\lvert\mathcal{X}_{\text{pool}}\rvert$
test set size	2,000	Held-out evaluation set
initial labelled	100	Initial labeled budget

Table 10: Model and active-learning hyperparameters for the ternary phase-competition benchmark.

Symbol / option	Value	Meaning
n_{ens}	8	Ensemble size
K_{MDN}	4	MDN mixture components per member
hidden features	64	MLP width
depth	2	Number of MLP hidden layers
activation	GELU	Nonlinearity [34]
training batch size	64	Mini-batch size for AdamW
peak learning rate	2×10^{-4}	AdamW peak LR (after warmup)
weight decay	5×10^{-2}	AdamW weight decay
gradient clip	0.1	Adaptive gradient clipping threshold
min iter per round	2,000	Floor on adaptive iteration count
n_{iter} (cap)	40,000	Max gradient steps per round
iter-per-sample	200	Slope of adaptive schedule
AL rounds	30	Number of acquisition rounds
query batch size	15	Queries acquired per round
acquisition batch size	256	Chunk size for scoring the pool
seeds	$\{0, 1, 2, 3, 4\}$	Data / training seeds

gap closing by $n \gtrsim 400$. Table 11 gives the underlying numbers: the seed-to-seed min–max range of test NLL averaged over each budget band. In the early band $n \in [115, 250]$, MI-LB's range is $\sim 2.6\times$ tighter than Random's and $\sim 4.5\times$ tighter than Epistemic Variance's; the inflated Variance spread comes from seeds in which the second-moment signal selects unhelpful queries before the ensemble has accumulated enough data to disagree informatively over modal structure. By the late band ($n \gtrsim 400$) all three ranges sit within a factor of ~ 2.4 of each other.

Table 11: Seed-to-seed min–max range of test NLL (in nats), averaged over each budget band, for the top-k base acquisitions on the synthetic phase-competition benchmark (`system_seed` $= 12$, 5 seeds).

Acquisition (top-k)	Early ($n \in [115, 250]$)	Late ($n \gtrsim 400$)
Random	1.82	0.24
Epistemic Variance	3.13	0.31
MI-LB	**0.69**	**0.13**

Results for SBAL and MaxDist variants. Table 12 reports the final test NLL at $n = 550$ across 5 seeds for all combinations, and Figure 9 shows the corresponding learning curves. Three observations.

First, MaxDist is the mean-best acquisition on this benchmark: MI-LB (MaxDist) at $w = 1$ finishes at 1.945 ± 0.061, narrowly ahead of MI-LB (top-k) at 1.985 ± 0.043. The seed-to-seed spread is large enough that the two are not separated cleanly—per-seed, MaxDist wins on 2 seeds, ties on 1, and is within 0.01 nat of top-k on the other 2—so the safest summary is that MaxDist matches MI-LB (top-k) and offers a small mean improvement, the same qualitative conclusion as on the coupled double-well benchmark of Appendix F.3.

Second, SBAL hurts MI-LB: MI-LB (SBAL) at $T = 1.0$ degrades from 1.985 ± 0.043 to 2.109 ± 0.054, with MI-LB (top-k) winning on all 5 seeds. This is consistent with the small-budget, narrow-high-information-manifold argument of Appendix F.3: at $T = 1.0$ enough mass leaks onto low-score points that the effective fraction of informative queries drops, and the realisable improvement from batch diversity is bounded above by the MaxDist result that preserves the informative ranking exactly.

Third, SBAL modestly helps Epistemic Variance: $\mathbf{2.029 \pm 0.050}$ for SBAL versus 2.062 ± 0.087 for top-k, with the SBAL variant winning on 4 of 5 seeds and producing a tighter seed-to-seed spread. Because the Variance-based score is itself already a noisier ranking signal than MI-LB on this benchmark, its top-k ordering is less informative, and stochastic relaxation does not cost what it does for MI-LB, at this temperature it actively helps by hedging against unhelpful early queries.

BAIT and Core-Set. Core-Set ties MI-LB within seed-to-seed noise (1.989 ± 0.043 vs 1.985 ± 0.043); the tie reflects benchmark geometry, with phase boundaries lying directly on the 2-D composition simplex so that k-Center-Greedy in feature space hits the same boundary MI-LB targets through entropy disagreement. BAIT is the only acquisition whose mean NLL falls within seed-to-seed noise of Random (2.190 ± 0.090 vs 2.209 ± 0.085): the 0.019 improvement is an order of magnitude smaller than the ~ 0.09 standard deviation of either run, whereas every other acquisition in Table 12 beats Random by at least 0.10 in mean NLL — well outside the noise band. The single-MC last-layer Fisher embedding produces a noisy candidate ranking on the 1-D scalar output, and the 30×15 small-budget protocol does not give the forward+backward greedy enough rounds to recover — the same failure mode anticipated in Appendix F.2 (where BAIT had the largest seed-to-seed spread of any reported method) but more pronounced here because of the lower-dimensional output.

Remark on per-acquisition simplex visualisation. A per-acquisition visualisation of selected points projected onto the (x_A, x_B) simplex slice would complement the boundary-seeking narrative of Section 4.3. The production runs reported in Table 12 did not log query indices to the experiment tracker, so we do not include such a figure here; the example notebook accompanying this benchmark provides a single-seed reproduction that materialises the simplex projection from local state.

Why a mixture head: $K = 1$ vs $K = 4$. To justify the choice of a mixture head ($K_{\text{MDN}} = 4$) over a single-Gaussian head ($K = 1$), we train both with the identical architecture used in the AL experiments (depth-2 MLP, 64 hidden units, $n_{\text{ens}} = 8$) on a larger offline dataset of 100,000 samples drawn from the same input distribution as the active-learning experiments, and evaluate both on a held-out test set of 10,000 samples. The ensemble-averaged test NLL is $\mathbf{1.34}$ for $K = 4$

Table 12: Final test NLL at $n = 550$, mean $\pm$ std across 5 seeds $\{0, 1, 2, 3, 4\}$ (min–max in brackets) for the synthetic phase-competition benchmark at `system_seed` $= 12$. SBAL temperature is $T = 1.0$ for both SBAL runs; MaxDist uses score weight $w = 1$.

Acquisition	Mean $\pm$ std	Min – Max
Random	2.209 ± 0.085	$2.136 - 2.346$
Epistemic Variance (top-k)	2.062 ± 0.087	$1.918 - 2.122$
Epistemic Variance (SBAL)	2.029 ± 0.050	$1.977 - 2.107$
MI-LB (top-k)	1.985 ± 0.043	$1.911 - 2.017$
MI-LB (SBAL)	2.109 ± 0.054	$2.057 - 2.174$
MI-LB (MaxDist)	$\mathbf{1.945 \pm 0.061}$	$1.870 - 2.023$
BAIT	2.190 ± 0.090	$2.087 - 2.265$
Core-Set	1.989 ± 0.043	$1.951 - 2.058$

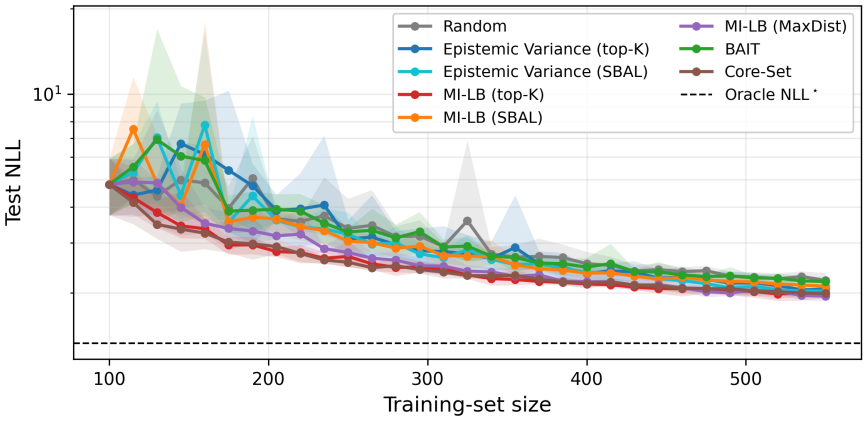


Figure 9: Synthetic phase-competition benchmark: test NLL vs. training-set size for all eight (acquisition, selection-strategy) combinations including the BAIT and Core-Set baselines (5 seeds; shaded bands min–max). SBAL $T = 1.0$, MaxDist $w = 1$. The dotted line marks the oracle floor $\mathrm{NLL}^* \approx 1.33$. The y-axis is clipped to $[1.2, 7.0]$ to suppress early-iteration training-instability spikes for the SBAL variants and BAIT.

versus $\mathbf{1.88}$ for $K = 1$, against an analytically computed oracle floor of $\mathrm{NLL}^* = 1.33$. The $K = 4$ ensemble therefore sits within 0.01 nat of the oracle—consistent with the fact that the simulator is itself a 4-component Gaussian mixture per input, so the $K = 4$ MDN model class contains the true distribution. The $K = 1$ model pays a 0.55-nat penalty at the same budget: forced to place a single Gaussian per input, it cannot resolve phase competition in the boundary region and must smear mass across phase means that differ by several nats (per-phase median means $+2.9$, $+6.9$, $+3.7$, -4.0 on this simulator instance). The penalty is smaller than on the coupled double-well benchmark (5.51 nats, Appendix F.3) because the per-phase response is a unimodal Gaussian with bounded variance, so the smearing cost is bounded; it is still clearly nonzero, and the active-learning experiments on this benchmark therefore use $K_{\mathrm{MDN}} = 4$ throughout.

Figure 10 shows the qualitative picture on a 120-point simplex grid with process parameters pinned to zero. The $K = 4$ ensemble's predicted conditional mean $\hat{\mathbb{E}}[Y \mid x]$ tracks the ground-truth $\mathbb{E}^*[Y \mid x]$ to within roughly ± 1 nat, with residuals smoothly distributed across the simplex. The $K = 1$ residual is visibly structured and larger (up to ± 4) at the intersections of the phase-boundary network, where the single-Gaussian head cannot simultaneously fit the correct mean and an appropriate variance—exactly the failure mode anticipated in Section 4.3.

F.5 Compute Resources

All experiments were run on a workstation with $7\times$ NVIDIA RTX A6000 GPUs (48 GB VRAM each), 128 CPU cores, and approximately 504 GiB of system memory. Each active-learning run uses a single A6000 GPU; peak GPU memory fits comfortably within the 48 GB budget for all configurations (small MLP backbones, ensemble size 8), so GPU memory is not a binding constraint.

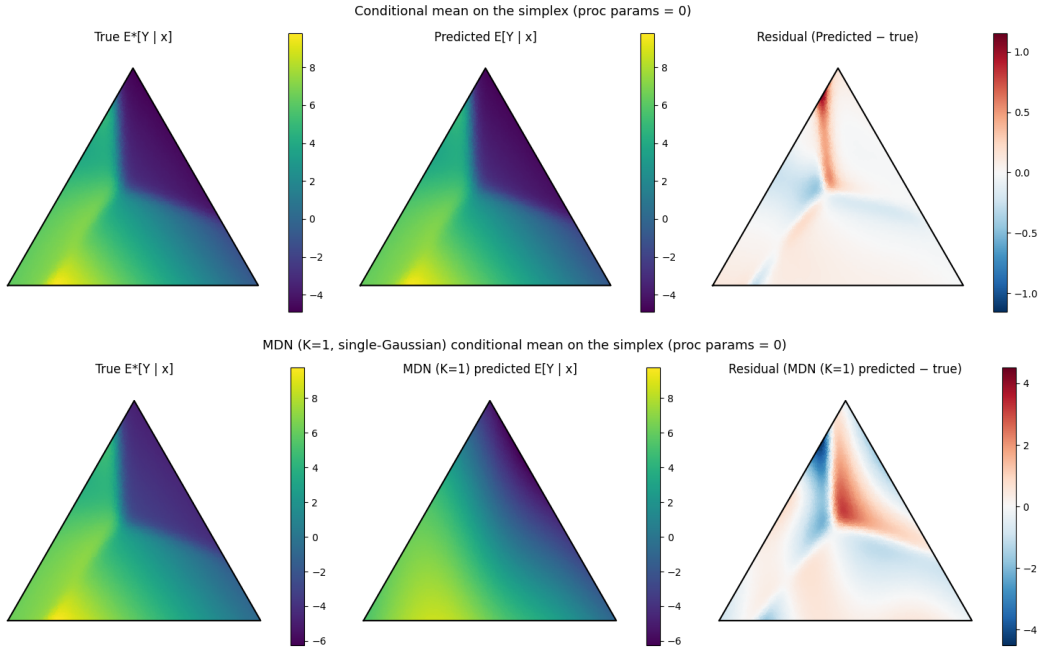


Figure 10: Synthetic phase-competition benchmark (`system_seed` $= 12$): predicted conditional mean on a 120-point simplex grid (process parameters pinned to zero) for the $K = 4$ MDN ensemble (**top row**) and a $K = 1$ single-Gaussian MDN (**bottom row**), both trained offline on 100,000 samples with the architecture used throughout the AL experiments. **Left column:** ground-truth $\mathbb{E}^*[Y \mid x]$. **Middle column:** ensemble-predicted $\hat{\mathbb{E}}[Y \mid x]$. **Right column:** residual (predicted − true). The $K = 4$ residuals sit within $\sim\pm 1$ and are smoothly distributed; the $K = 1$ residuals reach ± 4 and concentrate at the intersections of the phase-boundary network, where the single-Gaussian head cannot represent the competing phase components. The ensemble-averaged test NLL on a held-out set of 10,000 samples is 1.34 ($K = 4$) vs 1.88 ($K = 1$), against an oracle floor of 1.33.

Per benchmark we evaluate eight (acquisition, selection-strategy) combinations across 5 seeds, giving 40 runs per benchmark and 120 runs in total across the three benchmarks. Per-run wall-clock times, extracted from the W&B logs and reported as the median over 5 seeds, are summarised in Table 13. Total compute across the three benchmarks is approximately 25 GPU-hours, of which $\sim$12 h (roughly half) is the 15 BAIT runs, BAIT is the only acquisition with non-trivial selection overhead, since its forward+backward greedy on the per-input Fisher embeddings of the full 50,000-point pool scales with both pool size and query batch. All other acquisitions, including Core-Set, complete in roughly the same wall time as Random per benchmark.

Table 13: Per-run wall-clock time on a single NVIDIA RTX A6000, median over 5 seeds, for the three paper benchmarks. Each entry is the end-to-end runtime of one 20-round (multimodal, double-well) or 30-round (ternary) active-learning run. Total compute across all 120 runs is $\sim$25 GPU-hours, of which $\sim$12 h is BAIT ($\sim$8.6 h on the double-well benchmark alone).

Acquisition (selection)	Multimodal	Double-well	Ternary
Random (top-k)	2.7 min	3.1 min	12.8 min
Epistemic Variance (top-k)	3.2 min	3.4 min	13.6 min
MI-LB (top-k)	3.1 min	3.4 min	13.1 min
BAIT (top-k)	22.4 min	1.35 h	15.0 min
Core-Set (top-k)	3.5 min	4.4 min	14.2 min
Epistemic Variance (SBAL)	3.1 min	3.4 min	13.2 min
MI-LB (SBAL)	3.1 min	3.4 min	13.2 min
MI-LB (MaxDist)	3.7 min	3.9 min	13.6 min