Epistemic Uncertainty Estimation in Regression Ensemble Models with Pairwise Epistemic Estimators

Lucas Berry, David Meger
Department of Computer Science
McGill University
lucas.berry@mail.mcgill.ca

Abstract

This work introduces a novel approach, Pairwise Epistemic Estimators (PairEpEsts), for epistemic uncertainty estimation in ensemble models for regression tasks using pairwise-distance estimators (PaiDEs). By utilizing the pairwise distances between model components, PaiDEs establish bounds on entropy. We leverage this capability to enhance the performance of Bayesian Active Learning by Disagreement (BALD). Notably, unlike sample-based Monte Carlo estimators, PairEpEsts can estimate epistemic uncertainty up to 100 times faster and demonstrate superior performance in higher dimensions. To validate our approach, we conducted a varied series of regression experiments on commonly used benchmarks: 1D sinusoidal data, *Pendulum*, *Hopper*, *Ant*, and *Humanoid*, demonstrating PairEpEsts' advantage over baselines in high-dimensional regression active learning.

1 Introduction

In this paper, we propose Pairwise Epistemic Estimators (PairEpEsts) as a non-sample based method for estimating epistemic uncertainty in deep ensembles with probabilistic outputs for regression tasks. Epistemic uncertainty, often distinguished from aleatoric uncertainty, reflects a model's ignorance and can be reduced by increasing the amount of data available [26, 16, 29]. The significance of epistemic uncertainty is particularly pronounced in safety-critical systems, where a single erroneous prediction could lead to catastrophic consequences [43]. Moreover, leveraging epistemic uncertainty proves beneficial as an acquisition criterion for active learning strategies [27].

Previously Monte Carlo (MC) methods have been employed for estimating epistemic uncertainty, in regression tasks, due to the absence of closed-form expressions in most modeling scenarios [15, 4]. However, as the output dimension increases, these MC methods require a large number of samples to get accurate estimates. PairEpEsts leverage Pairwise-Distance Estimators (PaiDEs), which provide a non-sample-based alternative for estimating information-theoretic criteria in ensemble regression models with probabilistic outputs [35].

Ensembles can be seen as committees, with each component acting as a member [50]. PairEpEsts synthesize the consensus amongst a committee of probabilistic learners by calculating the distributional distance between each pair of committee members. Aggregating these distances allows accurate estimation of the mutual information between the model output and weight distribution. When pairwise distances are efficiently computable, PairEpEsts offer a fast, sample-free method to estimate epistemic uncertainty. Figure 1 shows committee members in agreement with low uncertainty (small distance) and disagreement with high uncertainty (large distance).

This study demonstrates the use of PairEpEsts to estimate epistemic uncertainty in regression ensembles with probabilistic outputs. Unlike classification, regression poses unique challenges because entropy is harder to estimate for mixtures of continuous distributions than for categorical

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

https://doi.org/10.52202/085713-2161

ones. To address this, we focus on Normalizing Flows (NFs), which can capture heteroscedastic and multimodal aleatoric uncertainty [49, 33]. This capability is particularly relevant for robotic locomotion, where nonlinear stochastic dynamics make data acquisition costly and active learning essential. The proposed framework extends epistemic uncertainty estimation to high-dimensional regression tasks that have been largely underserved in the literature. Our contributions are threefold:

- We introduce the framework PairEpEsts, which applies PaiDEs to estimate epistemic uncertainty in deep ensembles with probabilistic outputs (Section 4).
- We extend previous epistemic uncertainty estimation methods to settings with higher-dimensional output [4], and demonstrate how PairEpEsts outperform MC and other active learning baselines in these settings with rigorous statistical testing (Section 5).
- We provide an analysis of the time-saving advantages offered by PairEpEsts compared to MC estimators for epistemic uncertainty estimation (Section 5.4).

2 Problem Statement

Following a supervised learning framework for regression, let $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ denote a dataset, where $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}^d$, and our objective is to approximate a complex multi-modal conditional probability $p(y|x)$. Let $f_\theta(y; x)$ denote our approximation to the conditional probability density, where θ is a set of parameters to be learned and is distributed as $p(\theta)$.

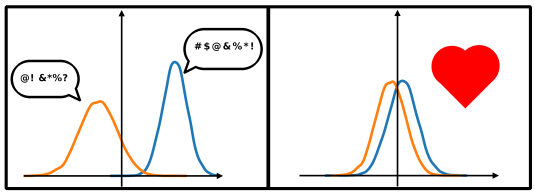


Figure 1: Epistemic uncertainty in an ensemble of probabilistic learners is high when the mixture components disagree (left) and low when there is agreement (right).

Leveraging a learned model, $f_\theta(y; x)$, we wish to estimate uncertainty which is typically viewed from a probabilistic perspective [14, 29]. When capturing uncertainty in supervised learning, one common measure is that of conditional differential entropy,

$$H(y|x) = -\int p(y|x) \ln p(y|x) dy.$$

Utilizing conditional differential entropy, we can establish an estimate for epistemic uncertainty as introduced by Houlsby et al. [27], expressed as:

$$I(y, \theta|x) = H(y|x) - H(y|x, \theta), \tag{1}$$

where $I(\cdot)$ refers to mutual information (MI). Equation 1 demonstrates that epistemic uncertainty, $I(y, \theta|x)$, can be represented by the difference between total uncertainty, $H(y|x)$, and aleatoric uncertainty, $H(y|x, \theta)$. MI measures the information gained about one variable by observing another.

To enable our methods to capture epistemic uncertainty, we employ ensembles to create $p(\theta)$. Ensembles leverage multiple models to obtain the estimated conditional probability by weighting the output distribution from each ensemble component,

$$f_\theta(y; x) = \sum_{j=1}^{M} \pi_j f_{\theta_j}(y; x) \qquad \sum_{j=1}^{M} \pi_j = 1, \tag{2}$$

where M, $0 \leq \pi_j \leq 1$ and θ_j are the number of model components, the component weights and the parameters for the jth component, respectively. In order to create an ensemble, one of two ways is typically chosen: randomization [5] or boosting [20]. While boosting has led to widely used machine learning methods [10], randomization has been the preferred method in deep learning [39].

Utilizing our ensemble we can estimate epistemic uncertainty in decision-making scenarios such as active learning. When all components produce the same $f_{\theta_i}(y, x)$, $I(y, \theta|x)$ is zero, indicating no epistemic uncertainty. Conversely, when the components exhibit varied output distributions, epistemic uncertainty is high. In an active learning framework, one chooses, at each iteration, what data points to add to a training dataset such that the model's performance improves as much as possible [41, 52].

In our context, one chooses the x's that maximize Equation 1 and adds those data points to the training set as in Bayesian Active Learning by Disagreement (BALD) [27]. It's worth noting that in the realm of continuous outputs as in regression and ensemble models, Equation 1 often lacks a closed-form solution, as the entropy of most mixtures of continuous distributions cannot be expressed in closed form [35]. Hence, prior methods have resorted to MC estimators for the estimation of epistemic uncertainty [15, 48]. One such MC method samples K points from the model, $y_j \sim f_\theta(y; x)$, and then estimates the total uncertainty,

$$\hat{H}_{MC}(y|x) = \frac{-1}{K} \sum_{j=1}^{K} \ln f_\theta(y_j; x).$$

The MC approach approximates the intractable integral over the continuous domain requiring only point-wise evaluation of the ensemble and its components. However, as the number dimensions increases, MC methods typically require a greater number of samples which creates a greater computational burden [51]. Note that the aleatoric uncertainty can be analytically computed as the average of the entropies of each ensemble component distribution.

3 Pairwise-Distance Estimators

Unlike MC methods, PaiDEs eliminate sampling by using generalized distance functions between ensemble components. They estimate the entropy of mixture distributions when pairwise distances have closed-form expressions, enabling estimation of $H(y|x)$ from Equation 1. We extend PaiDEs, from Kolchinsky and Tracey [35], to supervised learning and epistemic uncertainty estimation.

3.1 Properties of Entropy

One can treat a mixture model as a two-step process: first, a component is drawn and, second, a sample is taken from the corresponding component. Let $p(y, \theta|x)$ denote the joint of our output and model components given input x,

$$p(y, \theta|x) = p(\theta_j|x)p(y|\theta_j, x) = \pi_j p(y|\theta_j, x).$$

Using this representation, following principles of information theory [14], we can write its entropy as,

$$H(y, \theta|x) = H(\theta|y, x) + H(y|x). \tag{3}$$

Additionally, one can show the following bounds for $H(y|x)$,

$$H(y|\theta, x) \leq H(y|x) \leq H(y, \theta|x). \tag{4}$$

Intuitively, the lower bound can be justified by the fact that conditioning on more variables can only decrease or keep entropy the same. The upper bound follows from Equation 3, and $H(\theta|x, y) \geq 0$ since θ is modeled as a discrete random variable (ensemble index), unlike x and y which are continuous.

3.2 PaiDEs Definition

Let $D_\rho(p_i \parallel p_j)$ denote a generalized distance function between the probability distributions p_i and p_j, which for our case represent $p_i = p(y|x, \theta_i)$ and $p_j = p(y|x, \theta_j)$, respectively. More specifically, D is referred to as a premetric, $D_\rho(p_i \parallel p_j) \geq 0$ and $D_\rho(p_i \parallel p_j) = 0$ if $p_i = p_j$. The distance function need not be symmetric nor obey the triangle inequality. As such, PaiDEs can be defined as,

$$\hat{H}_\rho(y|x) := H(y|\theta, x) - \sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-D_\rho(p_i \parallel p_j)\right). \tag{5}$$

PaiDEs have many options for $D_\rho(p_i \parallel p_j)$ (Kullback-Leibler divergence, Wasserstein distance, Bhattacharyya distance, Chernoff α-divergence, Hellinger distance, etc.).

https://doi.org/10.52202/085713-2161

Theorem 3.1. *As from Kolchinsky and Tracey [35], using the extreme distance functions,*

$$D_{min}(p_i \parallel p_j) = 0 \quad \forall i, j$$
$$D_{max}(p_i \parallel p_j) = \begin{cases} 0, & \text{if } p_i = p_j, \\ \infty, & otherwise, \end{cases}$$

one can show that PaiDEs lie within bounds for entropy established in Equation 4.

Refer to Appendix A for the proof of Theorem 3.1. This provides a general class of estimators but a distance function still needs to be chosen.

3.3 Tighter Bounds for PaiDEs

Let the Chernoff α-divergence, where $\alpha \in [0, 1]$, be defined as [44],

$$D_{C_\alpha}(p_i \parallel p_j) = -\ln \int p^\alpha(y|x, \theta_i) p^{1-\alpha}(y|x, \theta_j) dy.$$

Corollary 3.2. *As from Kolchinsky and Tracey [35], when applying Chernoff α-divergence as our distance function in Equation 5, we achieve a tighter lower bound than $H(y|\theta, x)$ from Equation 4,*

$$H(y|\theta, x) \leq \hat{H}_{C_\alpha}(y|x) \leq H(y|x). \tag{6}$$

Refer to Appendix A for the proof of Corollary 3.2. In addition, the tightest lower bound can be shown to be $\alpha = 0.5$ for certain situations [35]. This is known as the Bhattacharyya distance,

$$D_B(p_i||p_j) = -\ln \int \sqrt{p(y|x, \theta_i) p(y|x, \theta_j)} dy. \tag{7}$$

In addition to the improved lower bound, there is an improved upper bound as well. Let Kullback-Liebler (KL) divergence be defined as follows,

$$D_{KL}(p_i \parallel p_j) = \int p(y|x, \theta_i) \ln \frac{p(y|x, \theta_i)}{p(y|x, \theta_j)} dy.$$

Note that the KL divergence is a not metric but does suffice as a generalized distance function.

Corollary 3.3. *As from Kolchinsky and Tracey [35], when applying Kullback-Liebler divergence as our distance function in Equation 5, we achieve a tighter upper bound than $H(y, \theta|x)$ from Equation 4,*

$$H(y|x) \leq \hat{H}_{KL}(y|x) \leq H(y, \theta|x). \tag{8}$$

Refer to Appendix A for the proof of Corollary 3.3.

4 Estimating Epistemic Uncertainty with Pairwise Epistemic Estimators

Our extension of prior methodologies enables the estimation of epistemic uncertainty, without the need for sampling. We illustrate this capability for NFs in the main paper and for Probabilistic Network Ensembles (PNEs) in Appendix E [39]. Note that our proposed estimators are applicable to any ensemble model whose component output distributions have closed-form pairwise distances. Since many models have these properties, we focus on Nflows Base for its expressiveness with multimodal aleatoric uncertainty and on widely used PNEs.

4.1 Pairwise Epistemic Estimators

By applying our definition of PaiDEs to Equation 1, we obtain the following expression:

$$\hat{I}_\rho(y, \theta) = \hat{H}_\rho(y|x) - H(y|x, \theta) = -\sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-D_\rho(p_i \parallel p_j)\right). \tag{9}$$

PaiDEs estimate epistemic uncertainty using only pairwise component distances, removing reliance on sampling. We propose two estimators:

$$\hat{I}_B(y,\theta) = -\sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-D_B(p_i \parallel p_j)\right),$$

$$\hat{I}_{KL}(y,\theta) = -\sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-D_{KL}(p_i \parallel p_j)\right),$$

where $D_B(p_i \parallel p_j)$ and $D_{KL}(p_i \parallel p_j)$ are defined for Gaussians in Appendix B. We refer to $\hat{I}_B(y,\theta)$ as PairEpEst-Bhatt and $\hat{I}_{KL}(y,\theta)$ as PairEpEst-KL.

4.2 Nflows Base

In this study, we utilize an ensemble technique known as Nflows Base, which has previously demonstrated robust performance in estimating both aleatoric and epistemic uncertainty on simulated robotic datasets by leveraging NFs to create ensembles [4].

NFs have traditionally been applied to unsupervised tasks [54, 53]. However, NFs have also been adapted for supervised learning tasks, particularly for regression [59, 1]. Using the structure described in Winkler et al. [59], a supervised NF is defined as:

$$p_{y|x}(y|x) = p_{b|x,\theta}(g_\phi^{-1}(y,x))|\det(J(g_\phi^{-1}(y,x)))|,$$
$$\log(p_{y|x}(y|x)) = \log(p_{b|x,\theta}(g_\phi^{-1}(y,x))) + \log(|\det(J(g_\phi^{-1}(y,x)))|),$$

where $p_{y|x}$ is the output distribution, $p_{b|x,\theta}$ is the base distribution with parameters θ, J refers to the Jacobian, and $g_\phi^{-1} : y \times x \mapsto b$ is the bijective mapping with parameters ϕ. For a complete review of NFs, refer to Papamakarios et al. [47]. Nflows Base creates an ensemble in the base distribution,

$$p_{y|x,\theta}(y|x,\theta_j) = p_{b|x,\theta_j}(g_\phi^{-1}(y,x))|\det(J(g_\phi^{-1}(y,x)))|,$$

where $p_{b|x,\theta_j}(b|x,\theta_j) = N(\mu_{\theta_j}(x), \Sigma_{\theta_j}(x))$, $\mu_{\theta_j}(x)$ and $\Sigma_{\theta_j}(x)$ denote the mean and covariance for input x and conditioned on θ_j. To encourage diversity in the ensemble, each member is initialized with different random weights, trained on a bootstrapped subset of the data, and assigned a fixed dropout mask sampled at the start of training (with $p = 0.5$), similar to [1]. This mask remains constant throughout training and inference. By constructing the ensemble within the base distribution, one can make use of closed-form pairwise-distance formulae as Berry and Meger [4] showed that estimating epistemic uncertainty in the base distribution is equivalent to estimating it in the output distribution.

Exploiting these closed-form pairwise distance formulae, Berry and Meger [4] showed that Nflows Base outperforms naive NF ensemble methods when estimating epistemic uncertainty. The aleatoric uncertainty from Equation 1 can be estimated in the base distribution space, and therefore can be computed analytically. However, this approach does not extend to the total uncertainty in Equation 1, which is why MC techniques have traditionally been used to estimate epistemic uncertainty.

4.3 Integrating Pairwise Epistemic Estimators with Nflows Base Ensembles

By combining Nflows Base and PairEpEsts, we construct an expressive non-parametric model capable of capturing intricate aleatoric uncertainty in the output distribution while efficiently estimating epistemic uncertainty in the base distribution. Equation 9 then becomes,

$$\hat{I}_\rho(y,\theta) = -\sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-D_\rho(p_{b|x,\theta_j} \parallel p_{b|x,\theta_i})\right). \tag{10}$$

Unlike previously proposed methods, we are able to estimate epistemic uncertainty without taking a single sample. Figure 7 in the Appendix shows an example of the distributional pairs that need to be considered in order to estimate epistemic uncertainty for an Nflows Base model. By applying PairEpEsts in the base distribution space we are able to capture epistemic uncertainty in the output space [4]. This approach enables the use of established formulae for computing distributional distances.

https://doi.org/10.52202/085713-2161

PairEpEsts can in principle be extended to skewed or heavy-tailed base distributions, provided that the pairwise premetric admits a closed-form expression. Recent work on tail-adaptive normalizing flows [40, 25] offers a promising direction for incorporating heavy-tailed behavior into normalizing flows. Integrating these approaches with PairEpEsts represents an important avenue for future research, particularly in domains where tail risk is critical.

4.4 Integrating Pairwise Epistemic Estimators with Probabilistic Network Ensembles

In addition to Nflows Base ensembles, we extend PairEpEsts to PNEs, which are commonly used for uncertainty quantification in regression [12, 38, 39]. PNEs comprise multiple components that produce Gaussian predictive distributions with learned means and variances.

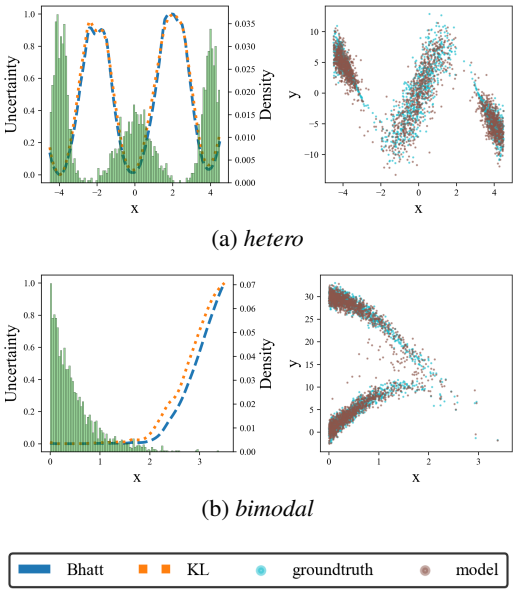
Applying PairEpEsts to PNEs leverages the closed-form pairwise distances between these Gaussian outputs to estimate epistemic uncertainty without sampling. Formally, the estimator is:

$$\hat{I}_\rho(y, \theta) = -\sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\big(-D_\rho(p_{y|x,\theta_j} \parallel p_{y|x,\theta_i})\big),$$

where $p_{y|x,\theta_i}$ denotes the i-th output distribution. Our experiments in Appendix E show that PairEpEsts with PNEs achieve comparable or better epistemic uncertainty estimates than baselines, especially in high-dimensional settings. Although PNEs are less expressive than normalizing flow ensembles, their combination with PairEpEsts offers a practical and efficient framework for scalable uncertainty quantification across diverse ensemble models.

5 Experimental Results

To evaluate our method, we tested each of our PairEpEsts (KL and Bhatt) on two 1D environments, as has been previously proposed in the literature [15]. Additionally, we present 4 multi-dimensional environments. In contrast to previous papers [48, 2], we evaluate each method on high dimensional output space (up to 257) to demonstrate the utility of PairEpEsts in this setting. Note that, for all experiments, the model components are assumed to be uniform, $\pi_j = \frac{1}{M}$, independent of x. All model hyperparameters are contained in Appendix B and the code can be found in the supplementary material. The experimental design follow previous literature [4].



(a) *hetero*

(b) *bimodal*

Figure 2: In the right graphs, the brown dots are sampled from Nflows Base and the cyan dots are the ground-truth data. The left graphs depict the epistemic uncertainty estimates corresponding to our two proposed estimators and the density of the ground-truth data in the green histogram.

5.1 Data

We evaluated our estimators on two 1D benchmarks, *hetero* and *bimodal*. The ground-truth data for *hetero* and *bimodal* can be seen in Figure 2 on the right graphs with the cyan dots. For *hetero*, there are two regions with low density (2 and -2) as can be seen by the green bar chart in the left graph which corresponds to the density of the ground-truth data. In these regions, one would expect a model to have high epistemic uncertainty. For *bimodal*, the number of data points drops off as x increases, thus we would expect a model to have epistemic uncertainty grow as x does. All details for data generation are contained in Appendix C.

In addition to the 1D environments, we tested our methods on four multi-dimensional environments: *Pendulum*, *Hopper*, *Ant*, and *Humanoid* [55]. Replay buffers were collected and the dynamics for each environment was modeled, $f_\theta(s_t, a_t) = \hat{s}_{t+1}$. The choice of multi-dimensional environments is

Table 1: Mean RMSE on the test set for the last (100^{th}) Acquisition Batch for Nflows Base. Experiments were across ten different seeds and the results are expressed as mean ± standard deviation. The best means are in bold and results that are statistically significant are highlighted.

	hetero	*bimodal*	*Pendulum*	*Hopper*	*Ant*	*Humanoid*
Output Dim.	1	1	3	11	32	257
Random	1.56 ± 0.14	6.4 ± 0.62	0.15 ± 0.04	0.97 ± 0.2	1.05 ± 0.1	6.59 ± 1.54
BatchBALD	1.54 ± 0.16	6.42 ± 0.65	0.13 ± 0.05	0.87 ± 0.27	0.94 ± 0.03	5.33 ± 1.2
BADGE	$\mathbf{1.44} \pm 0.12$	6.01 ± 0.04	0.34 ± 0.15	1.11 ± 0.27	0.91 ± 0.04	7.31 ± 3.11
BAIT	1.51 ± 0.14	6.26 ± 0.33	0.17 ± 0.05	1.06 ± 0.5	0.94 ± 0.04	11.01 ± 0.23
MC (BALD)	1.54 ± 0.17	6.01 ± 0.04	0.06 ± 0.01	0.32 ± 0.02	1.01 ± 0.05	10.71 ± 0.43
KL (ours)	1.47 ± 0.15	6.01 ± 0.04	$\mathbf{0.05} \pm 0.04$	0.3 ± 0.03	$\mathbf{0.9} \pm 0.07$	3.4 ± 0.48
Bhatt (ours)	1.55 ± 0.31	$\mathbf{6.0} \pm 0.04$	$\mathbf{0.05} \pm 0.01$	$\mathbf{0.29} \pm 0.03$	0.93 ± 0.08	$\mathbf{3.37} \pm 0.44$

$p < 0.05$ $p < 0.01$ $p < 0.001$

motivated by their common use as benchmarks and their higher-dimensional output space, providing a robust validation of our methods. Note that, for *Ant* and *Humanoid*, the dimensions representing their contact forces were eliminated due to a bug in Mujoco-v2.

In the Mujoco environment, the input consists of the current state and action, which are used to predict the next state based on the dynamics of the environment. In the hetero- and bimodal environments, we simplify the task to take a single input variable x, and predict the output variable y.

5.2 1D Experiments

Our 1D environments provide empirical proof that PairEpEsts can accurately measure epistemic uncertainty. Figure 2 depicts that both PairEpEsts are proficient at estimating the epistemic uncertainty as each method shows an increase in epistemic uncertainty around 2 and -2 on the *hetero* setting. This can be seen from the blue and orange lines with both estimators performing similarly.

A similar pattern can be seen for the *bimodal* setting in Figure 2, which shows that both PairEpEsts can accurately capture epistemic uncertainty. Each estimator shows the pattern of increasing epistemic uncertainty where the data is more scarce. Both examples show accurate epistemic uncertainty estimation with no loss in aleatoric uncertainty representation, as demonstrated in the right graphs in Figure 2: the brown dots closely match the cyan dots. Note that for visual clarity, the epistemic uncertainty has been scaled to 0-1, while ensuring that the relative properties of the estimators are preserved. Further discussion on over- and underestimation is provided in Section 6. While simple count-based heuristics may suffice in toy 1D settings like Figure 2, they fail to scale to higher-dimensional spaces where data is sparse and frequency statistics are uninformative. Moreover, they cannot exploit learned representations of the model, which are essential for structured, high-dimensional control tasks such as Hopper, Ant, and Humanoid. For these reasons, principled estimators such as PairEpEsts are required in realistic settings.

5.3 Active Learning

While the 1D experiments provide evidence of our estimators' effectiveness for estimating epistemic uncertainty, the active learning experiments extend this evaluation to higher-dimensional data and for a decision-making task. In order to benchmark our method, we compare against four state-of-the-art active learning frameworks: BatchBALD which maximizes information gain over batches using MC approximations [34], BADGE which uses gradient-based selection to identify the most informative data points for model improvement [3] and BAIT selects data based on uncertainty estimation through Bayesian active learning [2]. Additionally, a random baseline is included. The training set is initialized with 100 or 200 data points for 1D and multi-dimensional environments, respectively. In each acquisition batch, 10 points are added. For the MC estimator, 10^3 candidate inputs were sampled to construct each acquisition batch, with their epistemic uncertainties estimated using $K = 5000$.

https://doi.org/10.52202/085713-2161

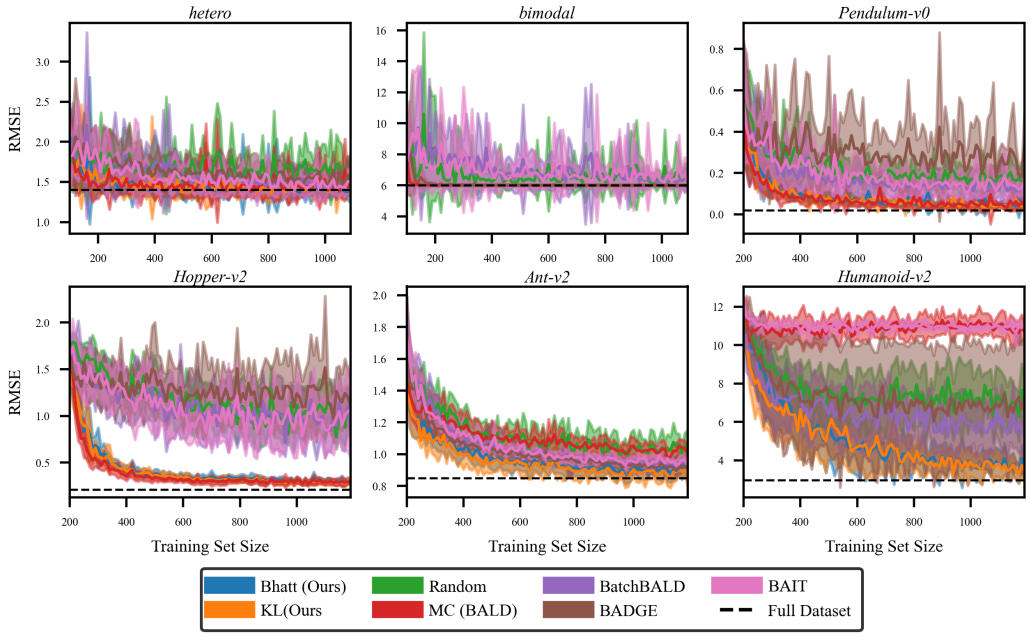


Figure 3: Mean RMSE on the test set as additional data is incorporated into the training set for Nflows Base. Both proposed methods perform comparably or significantly better than baselines, particularly in high-dimensional settings.

In the *Humanoid* environment, only 100 candidates were used due to computational constraints. In contrast, PairEpEsts sampled 10^4 candidate inputs and estimated their epistemic uncertainties for each environment including *Humanoid*. This underscores one advantage of our estimators over MC estimators, as they can estimate epistemic uncertainty over larger regions at a lower computational cost than their MC counterparts. Note that the other benchmarks each sampled 10^4 inputs as well and their respective acquisition functions applied. The RMSE on the test set was calculated at each acquisition batch.

Table 1 shows the performance of each framework on the 100th acquisition batch. Welch's t-test, with a Holm–Bonferroni correction (see Appendix G), compares our estimators to baselines in each environment. PairEpEsts achieve lower or comparable RMSEs, demonstrating their efficacy in estimating epistemic uncertainty. In higher dimensions, our estimators significantly outperform other methods, as shown in Humanoid. Learning curves are presented in Figure 3. Note that in the *bimodal* setting, all methods converge to similar performance and are indistinguishable in the plot. BatchBALD and BADGE, originally designed for classification, required adaptation for regression, facing challenges due to differences in uncertainty measure computation. Similarly, BAIT, designed for 1D regression, did not perform well in high-dimensional settings. The dashed line indicates the performance of our model when trained on the full dataset, serving as an upper bound for comparison.

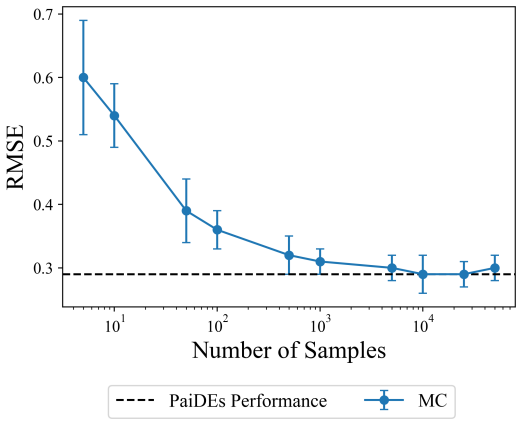


Figure 4: RMSE on the test set at the 100^{th} acquisition batch of the MC estimator on the *Hopper* environment for Nflows Base as the number of samples, K, per input, x, increases. Experiment were run across 10 seeds and the mean and stdev is being reported.

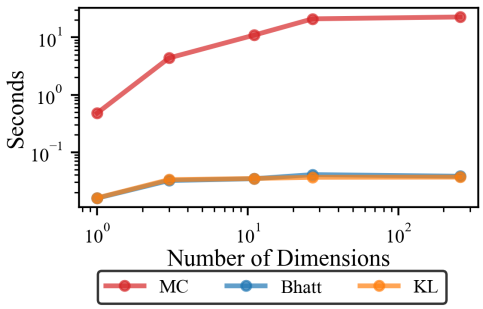


Figure 5: The amount of time taken for Nflows Base for each estimator across the different settings (1, 3, 11, 27, 257 dimensions). Results are averaged over 10 seeds. MC results use $K = 5000$ samples per acquisition candidate across all environments.

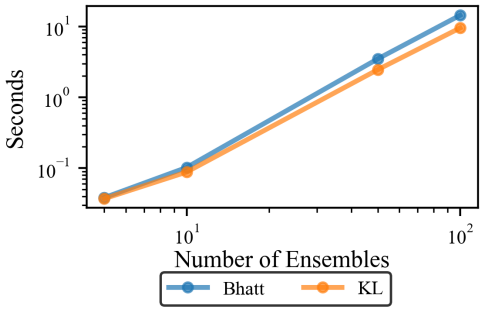


Figure 6: The amount of time taken for PaiDEs for Nflows Base as the number of ensemble components increases for *Humanoid-v2*. Results are averaged over 10 seeds.

For a deeper analysis of our method, we compared our estimators to MC estimators with a varying sample size, K, in the *Hopper* environment. As expected, MC estimators perform on par with PairEpEsts with a sufficient number of samples. This trend is illustrated in Figure 4. The MC estimator reaches the peak performance of our estimators given enough samples but does not perform better than PairEpEsts, suggesting that our estimators are sufficient to estimate epistemic uncertainty.

5.4 Time Analysis

In addition to benchmarking our estimators on active learning experiments, we provide an analysis of the time gains across our experiments. Figure 5 depicts the speed increase that can be gained using PairEpEsts over an MC approach. PairEpEsts provide a 1–2 order of magnitude runtime improvement over Monte Carlo approaches. The estimates are obtained from the active learning experiments.

6 Limitations

PairEpEsts have quadratic computational cost as the number of components grows. For asymmetric distances like KL-divergence, $M^2 - M$ pairwise distances are computed, while symmetric distances like Bhattacharyya require only $\frac{M^2-M}{2}$. Figure 6 shows timing as ensemble size increases. Bhattacharyya's cost could be further reduced using symmetry. Despite this, deep learning ensembles usually have few components (5–10) [45, 12], making the complexity manageable. The limitation is more relevant in large ensembles, such as weather forecasting [57].

PairEpEsts introduce a small bias absent in MC estimators. However, since active learning depends on the relative ordering of acquisition scores, this bias does not affect performance. As shown in Appendix D, the rankings are well preserved (Table 5). The relationships are illustrated in Figure 8, with epistemic uncertainty values normalized between 0 and 1 in Figure 2.

7 Related Work

Bayesian neural networks with information-based criteria are widely used for active learning in image classification [21, 32, 34], while others use gradient embeddings [3, 2]. Most focus on classification, with few methods extending to 1D regression [2]. Our work tackles active learning in high-dimensional regression, filling a gap relevant to robotic locomotion. Despite limitations of mutual information in classification [58], we demonstrate its value in high-dimensional regression.

Ensembles have been harnessed for epistemic uncertainty estimation [39, 11, 12]. Specifically related to our work, ensembles have been leveraged to quantify epistemic uncertainty in regression problems and active learning [15, 48, 4]. Depeweg et al. [15] employed Bayesian neural networks to model mixtures of Gaussians and demonstrated their ability to measure uncertainty in low-dimensional environments (1-2D). Building upon this foundation, Postels et al. [48] and Berry and Meger [4]

https://doi.org/10.52202/085713-2161

extended the research by developing efficient NF ensemble models that capture epistemic uncertainty. Our work advances this line of research by eliminating the need for sampling to estimate epistemic uncertainty, resulting in a faster and more effective method, especially in higher dimensions.

In addition to Bayesian neural networks and ensemble methods, the concept of Credal Bayesian Deep Learning has been introduced to enhance uncertainty quantification [7]. Furthermore, distributionally robust statistical verification has been proposed for high-dimensional autonomous systems using Imprecise Neural Networks, which provide uncertainty quantification guarantees [19].

Entropy estimators, which do not rely on sampling, is an active area of research [30, 31, 28, 35]. Kulak et al. [37] and Kulak and Calinon [36] demonstrated the utility of PaiDEs within Bayesian contexts, employing PaiDEs to estimate conditional predictive posterior entropy. In contrast, our approach provides a more general estimate of epistemic uncertainty, as defined in Equation 1, which can be applied to both ensemble, Bayesian methods and deep learning models.

Several methods have emerged in the literature for estimating epistemic uncertainty without relying on sampling techniques [56, 8]. Both Van Amersfoort et al. [56] and Charpentier et al. [8] focus on classification tasks with 1D categorical outputs. Charpentier et al. [9] extends the work of Charpentier et al. [8] to regression tasks but is limited to modeling outputs as members of the exponential family. In contrast, our approach can handle more complex output distributions by directly considering the outputs from NFs and can be applied to a larger space of regression models.

8 Conclusions

In this study, we introduced epistemic uncertainty estimators and applied them to Active Learning. We depicted how our method can be used to more efficiently quantify uncertainty by leveraging closed-form formulae instead of sampling. This led to improvements in computational speed and accuracy. As learning becomes pervasive in high-dimensional tasks in society, our method is well-placed to enable epistemic uncertainty awareness without unnecessary compute.

References

[1] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019.

[2] Jordan Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. Gone fishing: Neural active learning with fisher embeddings. *Advances in Neural Information Processing Systems*, 34:8927–8939, 2021.

[3] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2019.

[4] Lucas Berry and David Meger. Normalizing flow ensembles for rich aleatoric and epistemic uncertainty modeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6): 6806–6814, 2023.

[5] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[7] Michele Caprio, Souradeep Dutta, Kuk Jin Jang, Vivian Lin, Radoslav Ivanov, Oleg Sokolsky, and Insup Lee. Credal bayesian deep learning. *TLMR*, 2024.

[8] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems*, 33:1356–1367, 2020.

[9] Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. Natural posterior network: Deep bayesian uncertainty for exponential family distributions. *arXiv preprint arXiv:2105.04471*, 2021.

[10] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[11] Hyunsun Choi, Eric Jang, and Alexander A Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

[12] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[13] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms. *arXiv preprint arXiv:1904.06979*, 2019.

[14] Thomas M Cover and Joy A Thomas. *Elements of information theory*. Wiley-Interscience, 2006.

[15] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pages 1184–1193. PMLR, 2018.

[16] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

[17] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Cubic-spline flows. In *Workshop on Invertible Neural Networks and Normalizing Flows, International Conference on Machine Learning*, 2019.

[18] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. nflows: normalizing flows in PyTorch. https://doi.org/10.5281/zenodo.4296287, Nov 2020. Accessed: 2021-09-01.

[19] Souradeep Dutta, Michele Caprio, Vivian Lin, Matthew Cleaveland, Kuk Jin Jang, Ivan Ruchkin, Oleg Sokolsky, and Insup Lee. Distributionally robust statistical verification with imprecise neural networks. *HSCC*, 2025.

[20] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[21] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR, 2017.

[22] Ali Harakeh and Steven L Waslander. Estimating and evaluating regression predictive uncertainty in deep object detectors. *arXiv preprint arXiv:2101.05036*, 2021.

[23] David Haussler and Manfred Opper. Mutual information, metric entropy and cumulative relative entropy risk. *The Annals of Statistics*, 25(6):2451–2492, 1997.

[24] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE, 2007.

[25] Tennessee Hickling and Dennis Prangle. Flexible tails for normalizing flows. *arXiv preprint arXiv:2406.16971*, 2024.

[26] Stephen C Hora. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2-3):217–223, 1996.

[27] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[28] 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. IEEE, 2008.

 https://doi.org/10.52202/085713-2161

[29] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[30] Tony Jebara and Risi Kondor. Bhattacharyya and expected likelihood kernels. In *Learning theory and kernel machines*, pages 57–71. Springer, 2003.

[31] Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.

[32] 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.

[33] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[34] 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.

[35] Artemy Kolchinsky and Brendan D Tracey. Estimating mixture entropy with pairwise distances. *Entropy*, 19(7):361, 2017.

[36] Thibaut Kulak and Sylvain Calinon. Combining social and intrinsically-motivated learning for multi-task robot skill acquisition. *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

[37] Thibaut Kulak, Hakan Girgin, Jean-Marc Odobez, and Sylvain Calinon. Active learning of bayesian probabilistic movement primitives. *IEEE Robotics and Automation Letters*, 6(2): 2163–2170, 2021.

[38] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*, 2018.

[39] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[40] Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. Marginal tail-adaptive normalizing flows. In *International Conference on Machine Learning*, pages 12020–12048. PMLR, 2022.

[41] David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[42] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*, 2020.

[43] Adrián Morales-Torres, Ignacio Escuder-Bueno, Armando Serrano-Lombillo, and Jesica T Castillo Rodríguez. Dealing with epistemic uncertainty in risk-informed decision making for dam safety management. *Reliability Engineering & System Safety*, 191:106562, 2019.

[44] Frank Nielsen. Chernoff information of exponential families. *arXiv preprint arXiv:1102.2684*, 2011.

[45] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[46] John William Paisley. Two useful bounds for variational inference. 2010. URL https://api.semanticscholar.org/CorpusID:15745903.

[47] George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64, 2021.

[48] Janis Postels, Hermann Blum, Yannick Strümpler, Cesar Cadena, Roland Siegwart, Luc Van Gool, and Federico Tombari. The hidden uncertainty in a neural networks activations. *arXiv preprint arXiv:2012.03082*, 2020.

[49] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015.

[50] Lior Rokach. *Pattern classification using ensemble methods*, volume 75. World Scientific, 2010.

[51] Reuven Y Rubinstein and Peter W Glynn. How to deal with the curse of dimensionality of likelihood ratios in monte carlo simulation. *Stochastic Models*, 25(4):547–568, 2009.

[52] Burr Settles. Active learning literature survey. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2009.

[53] Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

[54] Esteban G Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

[55] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[56] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.

[57] Jonathan A Weyn, Dale R Durran, Rich Caruana, and Nathaniel Cresswell-Clay. Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. *Journal of Advances in Modeling Earth Systems*, 13(7):e2021MS002502, 2021.

[58] Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in Artificial Intelligence*, pages 2282–2292. PMLR, 2023.

[59] Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

 https://doi.org/10.52202/085713-2161

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction align with the paper's contributions and scope. The paper introduces PaiDEs as a non-sample based method for estimating epistemic uncertainty in regression tasks using ensembles, and it discusses its benefits in terms of computational efficiency and accuracy.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses the limitations of PaiDEs, particularly with respect to the quadratic computational complexity as the number of components increases, and the bias introduced by PaiDEs, which does not affect the relative relationships between data points Section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All theoretical results, assumptions, and proofs are provided in the appendix of the paper, with detailed discussions of the assumptions for each result.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The code is included in the supplementary material, and all necessary details, including the experimental setup and hyperparameters are provided in the paper to ensure the experiments can be reproduced.

   Guidelines:

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

https://doi.org/10.52202/085713-2161

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code is provided with the paper, and the data will be made publicly available upon publication.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All experimental details, including data splits, hyperparameters, and training settings, are thoroughly discussed in the paper and appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The paper reports error bars, p-values, and statistical significance tests, such as Welch's t-tests, across various experimental settings to ensure the robustness of the results.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper provides details about the computational resources used for experiments, including the type of compute workers (CPUs and GPUs), memory requirements, and execution times.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper follows the NeurIPS Code of Ethics, ensuring fairness, transparency, and responsibility in the handling of experiments.

   Guidelines:

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

10. **Broader impacts**

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

https://doi.org/10.52202/085713-2161

Answer: [Yes]

Justification: The paper discusses both the potential positive impacts of improving epistemic uncertainty estimation in active learning scenarios. To the best of our knowledge, there are no potential negative societal impact to epistemic uncertainty estimation.

Guidelines:

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

11. **Safeguards**

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

Answer: [NA]

Justification: The data used in this paper is just simulated robotics data.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All assets (such as datasets and code) used in the paper are properly credited, and the license and terms of use are respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: New assets introduced in the paper are well documented in the supplementary material.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No crowdsourcing or research with human subjects was involved in this study.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No research with human subjects was conducted in this paper.

https://doi.org/10.52202/085713-2161

Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: LLMs were not used as a core method in this paper, only in the writing process.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Proofs

The proofs follow from the steps of Kolchinsky and Tracey [35].

Proof of Theorem 3.1

Proof. Applying $D_{min}(p_i \parallel p_j)$ as our distance in PaiDEs,

$$\begin{aligned}\hat{H}(Y|X) &= H(Y|X,\Theta) - \sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-D_{min}(p_i \parallel p_j)\right) \\ &= H(Y|X,\Theta) - \sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \\ &= H(Y|X,\Theta).\end{aligned}$$

Note the last line follows from the fact that the component weights must sum to one, $\sum_{j=1}^{M} \pi_j = 1$. Next using $D_{max}(p_i \parallel p_j)$,

$$\begin{aligned}\hat{H}(Y|X) &= H(Y|X,\Theta) - \sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-D_{max}(p_i \parallel p_j)\right) \\ &= H(Y|X,\Theta) - \sum_{i=1}^{M} \pi_i \ln \left(\pi_i + \sum_{j\neq i} \pi_j \exp\left(-D_{max}(p_i \parallel p_j)\right)\right) \\ &= H(Y|X,\Theta) - \sum_{i=1}^{M} \pi_i \ln \pi_i \\ &= H(Y|X,\Theta) + H(\Theta|X) \\ &= H(Y,\Theta|X).\end{aligned}$$

This shows the upper bound with $D_{max}(p_i \parallel p_j)$ and the lower bound with $D_{min}(p_i \parallel p_j)$. Note that in our case $H(\Theta|X) = H(\Theta)$ as the distribution of weights does not depend on our input. □

Proof of Corollary 3.2

Proof. To show the upper bound from Equation 6 we use a derivation from Haussler and Opper [23],

$$\begin{aligned}H(Y|X) &= H(Y|X,\Theta) - \int \sum_{i=1}^{M} \pi_i p_i(y|x) \ln \frac{p_{Y|X}(y|x)}{p_i(y|x)} dy \\ &= H(Y|X,\Theta) - \int \sum_{i=1}^{M} \pi_i p_i(y|x) \ln \frac{p_{Y|X}(y|x)}{p_i(y|x)^{1-\alpha} \sum_j \pi_j p_j(y|x)^{1-\alpha}} dy \\ &\quad - \int \sum_{i=1}^{M} \pi_i p_i(y|x) \ln \frac{\sum_j \pi_j p_j(y|x)^{1-\alpha}}{p_i(y|x)^{\alpha}} dy \\ &\geq H(Y|X,\Theta) - \int \sum_{i=1}^{M} \pi_i p_i(y|x) \ln \frac{\sum_j \pi_j p_j(y|x)^{1-\alpha}}{p_i(y|x)^{\alpha}} dy \\ &\geq H(Y|X,\Theta) - \sum_{i=1}^{M} \pi_i \ln \int p_i(y|x)^{\alpha} \sum_j \pi_j p_j(y|x)^{1-\alpha} dy \\ &= H(Y|X,\Theta) - \sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-C_\alpha(p_i \parallel p_j)\right).\end{aligned}$$

The two inequalities follow from Jensen's inequality. □

https://doi.org/10.52202/085713-2161

Proof of Corollary 3.3

Proof. The lower bound can be shown using the definition of entropy for a mixture,

$$\begin{aligned}
H(Y|X) &= -\sum_{i=1}^{M} \pi_i E\left[\ln \sum_{j=1}^{M} \pi_j p_j(y|x)\right] \\
&\leq -\sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(E_{p_i}\left[\ln p_j(y|x)\right]\right) \\
&= -\sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-H(p_i \parallel p_j)\right) \\
&= \sum_{i=1}^{M} \pi_i H(p_i) - \sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-KL(p_i \parallel p_j)\right) \\
&= H(Y|X,\Theta) - \sum_{i=1}^{M} \pi_i \ln \sum_{j=1}^{M} \pi_j \exp\left(-KL(p_i \parallel p_j)\right),
\end{aligned}$$

where E_{p_i} refers to the expectation when Y is distributed as p_i and $H(p_i \parallel p_j)$ indicates the cross entropy. The inequality in line 2 follows from Hershey and Olsen [24] and Paisley [46]. □

A.1 PairEpEst-KL vs EPKL

In addition to the previous results from Kolchinsky and Tracey [35], we provide a unique proof to our paper demonstrating that the PairEpEst-KL estimator provides a strictly tighter upper bound on MI than the Expected Pairwise KL (EPKL) introduced by Malinin and Gales [42]. EPKL is another uncertainty measure that captures epistemic uncertainty by averaging the pairwise Kullback-Leibler divergences between predictive distributions of ensemble members. It quantifies how diverse the ensemble's predictions are, reflecting the model's uncertainty due to lack of knowledge or data. EPKL serves as an upper bound on MI. Our PairEpEst-KL estimator leverages a log-sum-exp formulation, yielding a sharper (tighter) upper bound on MI, which leads to more precise uncertainty quantification.

Specifically, consider an ensemble of M predictive distributions $p_1(y), \ldots, p_M(y)$. The EPKL is defined as the average pairwise Kullback-Leibler divergence between distinct ensemble components:

$$\text{EPKL} := \frac{1}{M(M-1)} \sum_{i \neq j} D_{\text{KL}}(p_i(y) \| p_j(y)).$$

Our PairEpEst-KL estimator, denoted by B, is given by

$$B := -\frac{1}{M} \sum_{i=1}^{M} \ln \left(\frac{1}{M} \sum_{j=1}^{M} \exp\left(- D_{\text{KL}}(p_i(y) \| p_j(y))\right)\right).$$

The following theorem formalizes the relationship between these two quantities and establishes that

$$B < \text{EPKL},$$

meaning that PairEpEst-KL yields a strictly sharper upper bound on MI compared to EPKL.

Theorem A.1. *Let $p_1(y), \ldots, p_M(y)$ be probability distributions, and define*

$$A := \frac{1}{M(M-1)} \sum_{i \neq j} D_{\text{KL}}(p_i(y) \| p_j(y)),$$

and

$$B := -\frac{1}{M} \sum_{i=1}^{M} \ln \left(\frac{1}{M} \sum_{j=1}^{M} \exp\left(- D_{\text{KL}}(p_i(y) \| p_j(y))\right)\right).$$

Then,

$$B \leq \frac{1}{M^2} \sum_{i,j} D_{\mathrm{KL}}(p_i(y) \| p_j(y)) < A,$$

and in particular,

$$B < A.$$

Proof. Define

$$s_{ij} := -D_{\mathrm{KL}}(p_i(y) \| p_j(y)) \leq 0.$$

By Jensen's inequality applied to the concave function $\ln(\cdot)$, for each fixed i,

$$\ln \left(\frac{1}{M} \sum_{j=1}^{M} e^{s_{ij}} \right) \geq \frac{1}{M} \sum_{j=1}^{M} s_{ij}.$$

Multiplying both sides by -1 reverses the inequality:

$$-\ln \left(\frac{1}{M} \sum_{j=1}^{M} e^{s_{ij}} \right) \leq -\frac{1}{M} \sum_{j=1}^{M} s_{ij}.$$

Substituting back for s_{ij}, we get

$$-\ln \left(\frac{1}{M} \sum_{j=1}^{M} \exp\big(- D_{\mathrm{KL}}(p_i(y) \| p_j(y)) \big) \right) \leq \frac{1}{M} \sum_{j=1}^{M} D_{\mathrm{KL}}(p_i(y) \| p_j(y)).$$

Averaging over i, it follows that

$$B = \frac{1}{M} \sum_{i=1}^{M} \left[-\ln \left(\frac{1}{M} \sum_{j=1}^{M} e^{-D_{\mathrm{KL}}(p_i \| p_j)} \right) \right] \leq \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} D_{\mathrm{KL}}(p_i \| p_j).$$

Since $D_{\mathrm{KL}}(p_i \| p_i) = 0$, the double sum over all i, j is equal to the sum over $i \neq j$:

$$\sum_{i=1}^{M} \sum_{j=1}^{M} D_{\mathrm{KL}}(p_i \| p_j) = \sum_{i \neq j} D_{\mathrm{KL}}(p_i \| p_j).$$

Note that

$$M^2 > M(M-1) \implies \frac{1}{M^2} < \frac{1}{M(M-1)},$$

so

$$B \leq \frac{1}{M^2} \sum_{i \neq j} D_{\mathrm{KL}}(p_i \| p_j) < \frac{1}{M(M-1)} \sum_{i \neq j} D_{\mathrm{KL}}(p_i \| p_j) = A,$$

which establishes the strict inequality

$$B < A,$$

completing the proof. □

B Compute and Hyper-parameter Details

The Nflows Base model employed one nonlinear transformation, g, with a single hidden layer containing 20 units, utilizing cubic spline flows as per [17]. The base network consisted of two hidden layers, each comprising 40 units with ReLU activation functions. It is important to note that all base distributions were Gaussian. The PNEs adopted an architecture of three hidden layers each with 50 units and ReLU activation functions. Model hyperparameters remained consistent across all experiments. The base distribution and the bijective mapping are trained simultaneously. Each base distribution network is mapped through the same bijective mapping, which is shared across all ensemble components. Every time a base distribution network takes a training step, the bijective mapping is updated accordingly. This approach ensures that the bijective mapping

 https://doi.org/10.52202/085713-2161

Algorithm 1 Active Learning Using PairEpEsts

Input: training dataset $\mathcal{D}_{train}$, oracle dataset $\mathcal{D}_{oracle}$, number of iterations T, sample size S, and batch size B
for i=1,2,...,T **do**
  Randomly initialize each ensemble component θ_j.
  Train our ensemble on $\mathcal{D}_{train}$ using bootstrapped samples.
  Sample S points from $\mathcal{D}_{oracle}$ uniformly, $\mathcal{D}_S$.
  Calculate the top B values according to Equation 9 from $\mathcal{D}_S$, $\mathcal{D}_B$.
  Add $\mathcal{D}_B$ to the training dataset while removing it from the oracle dataset, $\mathcal{D}_{train} = \mathcal{D}_B \cup \mathcal{D}_{train}$ and $\mathcal{D}_{oracle} = \mathcal{D}_{oracle} \setminus \mathcal{D}_B$
end for
train final model on $\mathcal{D}_{train}$.

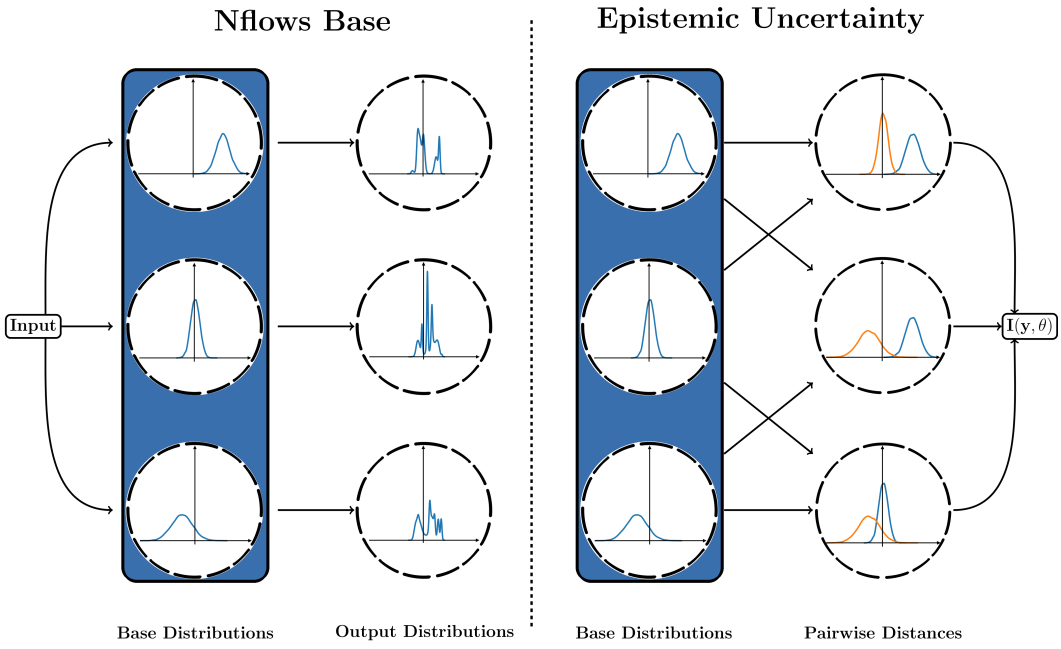


Figure 7: Nflows Base as an ensemble of 3 components with one bijective transformation on the left and an example of the pairwise comparisons needed to estimate epistemic uncertainty for said model on the right. Note the base distributions, instead of the output distributions, from Nflows Base are used to estimate the epistemic uncertainty which are highlighted by the blue bar.

is consistent across all components of the ensemble. This methodology is similar to previous approaches, such as those proposed by Osband et al. [45], where shared mappings are utilized within ensemble-based models. Training was conducted using 16GB RAM on Intel Gold 6148 Skylake @ 2.4 GHz CPUs and NVidia V100SXM2 (16G memory) GPUs. For each experimental setting, PNEs and Nflows Base were executed with five ensemble components. The MC estimator sampled 1000 and 5000 points for Nflows Base and PNEs, respectively, for each x conditioned on. The nflows library [18] was employed with minor modifications. Our code can be found at https://github.com/nwaftp23/pairflow-uncertainty.

Note that for the Bhatt estimator, the Bhattacharyya distance between two Gaussians is,

$$D_B(p_i||p_j) = \frac{1}{8}(\mu_{i|x} - \mu_{j|x})^{\mathrm{T}}\Sigma^{-1}(\mu_{i|x} - \mu_{j|x}) + \frac{1}{2}\ln\left(\frac{\det\Sigma}{\sqrt{\det\Sigma_{i|x}\det\Sigma_{j|x}}}\right),$$

$$\Sigma = \frac{\Sigma_{i|x} + \Sigma_{j|x}}{2}.$$

Also note that for the KL estimator, the KL divergence between two Gaussians is,

$$D_{KL}(p_i \parallel p_j) = \frac{1}{2}\left(\text{tr}(\Sigma_{j|x}^{-1}\Sigma_{i|x}) - D + \ln\left(\frac{\det \Sigma_{j|x}}{\det \Sigma_{i|x}}\right) + (\mu_{j|x} - \mu_{i|x})^{\mathrm{T}}\Sigma_{j|x}^{-1}(\mu_{j|x} - \mu_{i|x})\right),$$

where $\text{tr}(\cdot)$ refers to the trace of a matrix.

To complement the per-step acquisition runtimes reported in Figure 3, we also provide end-to-end active learning runtimes in Table 3 and Table 4. These results reflect the total wall-clock time required for a full active learning loop in each environment. These tables reinforce the efficiency advantage of PairEpEsts: across both architectures, KL and Bhatt estimators consistently require less total runtime than MC, particularly in higher-dimensional environments such as *Ant* and *Humanoid*. Notably, the runtime gap widens with dimensionality, reflecting the scalability of PairEpEsts relative to sampling-based methods.

To justify our choice of five ensemble components, we conducted an additional experiment analyzing the effect of ensemble size on active learning performance. This experiment was performed on the *Hopper* environment, with performance measured as RMSE at the final acquisition batch. Results are reported in Table 2. Performance improves markedly when increasing from 3 to 5 members, but remains stable for larger ensembles, suggesting that five components provide a strong balance between efficiency and accuracy.

Table 2: RMSE at the final acquisition batch for different ensemble sizes on *Hopper*.

	3	4	5	7	10
KL	0.51	0.43	0.30	0.31	0.29
Bhatt	0.50	0.45	0.29	0.28	0.30

Table 3: Total active learning runtimes (minutes) for Nflows Base across environments.

	Hetero	Bimodal	Pendulum	Hopper	Ant	Humanoid
KL	58.14	58.22	76.02	77.06	80.21	96.28
Bhatt	57.72	58.04	76.86	76.94	78.54	95.58
MC	61.87	61.82	94.44	98.66	117.40	155.34

Table 4: Total active learning runtimes (minutes) for PNEs across environments.

	Hetero	Bimodal	Pendulum	Hopper	Ant	Humanoid
KL	22.68	22.50	28.44	28.60	30.58	35.24
Bhatt	22.50	22.36	28.24	28.88	30.14	36.04
MC	24.42	24.62	35.22	46.42	66.96	84.30

C Data

The *hetero* dataset was generated using a two step process. Firstly, a categorical distribution with three values was sampled, where $p_i = \frac{1}{3}$. Secondly, x was drawn from one of three different Gaussian distributions ($N(-4, \frac{2}{5})$, $N(0, \frac{9}{10})$, $N(4, \frac{2}{5})$) based on the value of the categorical distribution. The corresponding y was then generated as follows:

$$y = 7\sin(x) + 3z\left|\cos\left(\frac{x}{2}\right)\right|.$$

On the other hand, the *bimodal* dataset was created by sampling x from an exponential distribution with parameter $\lambda = 2$, and then sampling n from a Bernoulli distribution with $p = 0.5$. Based on the value of n, the y value was determined as:

$$y = \begin{cases} 10\sin(x) + z & n = 0 \\ 10\cos(x) + z + 20 - x & n = 1 \end{cases}.$$

https://doi.org/10.52202/085713-2161

Note that for both *bimodal* and *hetero* data $z \sim N(0, 1)$.

Regarding the multi-dimensional environments, namely *Pendulum*, *Hopper*, *Ant*, and *Humanoid*, the training sets and test sets were collected using different approaches. The training sets were obtained by applying a random policy, while the test sets were generated using an expert policy. This methodology was employed to ensure diversity between the training and test datasets. Distribution shift is an inherent challenge in robotic dynamics, where data generated from different policies can vary significantly. These shifts, such as those observed in sim2real and imitation learning contexts. Notably, the OpenAI Gym library was utilized, with minor modifications [6].

D Additional Results

In addition to Table 1, we present Figure 3 showing the full active learning curves and Table 7 detailing additional acquisition batches. The trend of PairEpEsts outperforming or performing comparably to baselines is consistent across environments and acquisition batches. We also evaluated log-likelihood, a proper scoring rule [22], with results shown in Figure 9. Using 10 seeds, we report means and standard deviations. The PairEpEst-KL and PairEpEst-Bhatt estimators perform similarly or better than the MC estimator on all environments.

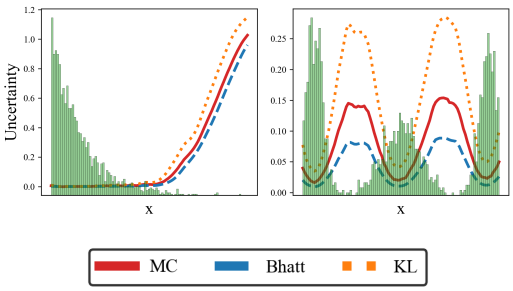


Figure 8: The bias introduced by PairEpEsts for Nflows Base compared to the MC method on the 1D environments.

For the high-dimensional *Humanoid* setting, however, log-likelihood did not improve as data was added. This limitation arises because computing ensemble log-likelihood requires evaluating each component's likelihood and summing them, a process prone to numerical underflow as values round to zero. As in Table 1, selected acquisition batches are reported in Table 8. We further compare the MC estimator and PairEpEsts on *Hopper* as the number of drawn samples increases (Figure 10). Additionally, we assessed the consistency of active learning rankings by computing Spearman's rank correlation between the MC estimator and PairEpEsts. As shown in Table 5, correlations are close to 1 across all models and datasets, indicating strong preservation of relative ordering despite small biases. All correlations are statistically significant with extremely low p-values (maximum $p = 1.11 \times 10^{-83}$).

As summarized in Table 6, Monte Carlo methods achieve comparable runtime to PairEpEsts with roughly 10 samples in *Hopper*, and with even fewer samples in higher-dimensional environments such as *Ant* and *Humanoid*. This suggests that small-sample MC can, in principle, be competitive in terms of computational cost. However, PairEpEsts provide robust and consistent uncertainty estimates without the need for sample tuning, making them particularly advantageous in high-dimensional tasks where both runtime efficiency and estimator stability are critical. Moreover, the accuracy of MC estimates deteriorates as the number of samples decreases, whereas PairEpEsts maintain reliability at low computational cost.

Table 5: Spearman's rank correlation coefficients comparing active learning rankings between MC and PairEpEsts, demonstrating that our estimators preserve relative ranking of points.

Model	Dataset	KL Corr.	Bhatt Corr.
NFlows Base	Bimodal	0.9958	0.9958
NFlows Base	Hetero	0.9976	0.9986
PNEs	Bimodal	0.9893	0.9893
PNEs	Hetero	0.9943	0.9972

Table 6: Number of MC samples that matched the runtime of PairEpEsts across environments.

Environment	Dimensionality	MC Samples (Runtime Matched)
Hetero	1	500
Bimodal	1	500
Pendulum	3	100
Hopper	11	10
Ant	27	5
Humanoid	257	5

Table 7: Mean RMSE on the test set for certain Acquisition Batches for Nflows Base. Experiments were across ten different seeds and the results are expressed as mean plus minus one standard deviation.

Env	Acquisition Batch	Random	BatchBALD	BADGE	BAIT	MC (BALD)	KL (ours)	Bhatt (ours)
hetero	10	1.76 ± 0.17	1.86 ± 0.35	1.84 ± 0.36	1.92 ± 0.28	$\mathbf{1.48} \pm 0.2$	1.66 ± 0.2	1.56 ± 0.26
	25	1.69 ± 0.45	1.66 ± 0.26	1.74 ± 0.25	1.71 ± 0.53	1.44 ± 0.1	$\mathbf{1.42} \pm 0.12$	$\mathbf{1.42} \pm 0.11$
	50	1.64 ± 0.22	1.54 ± 0.18	1.55 ± 0.1	1.6 ± 0.22	1.45 ± 0.12	1.49 ± 0.29	$\mathbf{1.42} \pm 0.11$
	100	1.56 ± 0.14	1.54 ± 0.16	$\mathbf{1.44} \pm 0.12$	1.51 ± 0.14	1.54 ± 0.17	1.47 ± 0.15	1.55 ± 0.31
bimodal	10	8.4 ± 3.38	7.37 ± 1.38	6.11 ± 0.1	6.91 ± 1.45	6.02 ± 0.05	$\mathbf{6.01} \pm 0.05$	$\mathbf{6.01} \pm 0.05$
	25	6.1 ± 0.08	6.74 ± 0.66	6.02 ± 0.04	6.85 ± 1.46	6.04 ± 0.04	6.02 ± 0.05	$\mathbf{6.01} \pm 0.04$
	50	6.57 ± 0.72	6.42 ± 0.66	6.01 ± 0.04	6.61 ± 0.97	6.01 ± 0.04	6.01 ± 0.04	$\mathbf{6.0} \pm 0.04$
	100	6.4 ± 0.62	6.42 ± 0.65	6.01 ± 0.04	6.26 ± 0.33	6.01 ± 0.04	6.01 ± 0.04	$\mathbf{6.0} \pm 0.04$
Pendulum	10	0.28 ± 0.06	0.32 ± 0.23	0.32 ± 0.27	0.42 ± 0.19	$\mathbf{0.12} \pm 0.04$	0.15 ± 0.06	0.17 ± 0.08
	25	0.22 ± 0.08	0.17 ± 0.06	0.31 ± 0.19	0.19 ± 0.07	$\mathbf{0.09} \pm 0.03$	$\mathbf{0.09} \pm 0.03$	$\mathbf{0.09} \pm 0.02$
	50	0.18 ± 0.08	0.15 ± 0.06	0.28 ± 0.14	0.17 ± 0.06	0.06 ± 0.03	0.06 ± 0.05	$\mathbf{0.05} \pm 0.01$
	100	0.15 ± 0.04	0.13 ± 0.05	0.34 ± 0.15	0.17 ± 0.05	$\mathbf{0.04} \pm 0.01$	0.05 ± 0.04	0.05 ± 0.01
Hopper	10	1.6 ± 0.19	1.3 ± 0.26	1.45 ± 0.28	1.36 ± 0.19	$\mathbf{0.55} \pm 0.09$	0.66 ± 0.08	0.69 ± 0.1
	25	1.24 ± 0.26	1.3 ± 0.33	1.37 ± 0.23	1.18 ± 0.29	$\mathbf{0.36} \pm 0.06$	0.38 ± 0.05	0.39 ± 0.06
	50	1.14 ± 0.16	1.06 ± 0.3	1.26 ± 0.23	1.05 ± 0.27	$\mathbf{0.31} \pm 0.04$	0.33 ± 0.03	0.34 ± 0.04
	100	0.97 ± 0.2	0.87 ± 0.27	1.11 ± 0.27	1.06 ± 0.5	$\mathbf{0.29} \pm 0.02$	0.3 ± 0.03	$\mathbf{0.29} \pm 0.03$
Ant	10	1.33 ± 0.1	1.31 ± 0.11	1.21 ± 0.14	1.25 ± 0.08	1.24 ± 0.13	$\mathbf{1.09} \pm 0.1$	1.13 ± 0.09
	25	1.13 ± 0.09	1.09 ± 0.03	1.05 ± 0.04	1.08 ± 0.04	1.13 ± 0.1	$\mathbf{1.0} \pm 0.07$	1.03 ± 0.08
	50	1.05 ± 0.05	1.01 ± 0.03	1.02 ± 0.06	0.98 ± 0.03	1.05 ± 0.03	$\mathbf{0.93} \pm 0.05$	0.94 ± 0.07
	100	1.05 ± 0.1	0.94 ± 0.03	0.91 ± 0.04	0.94 ± 0.04	1.01 ± 0.05	$\mathbf{0.9} \pm 0.07$	0.93 ± 0.08
Humanoid	10	8.79 ± 1.17	8.26 ± 1.91	8.6 ± 2.43	10.97 ± 0.15	10.69 ± 1.05	$\mathbf{6.82} \pm 1.34$	7.5 ± 1.63
	25	6.97 ± 1.48	6.59 ± 1.81	6.78 ± 3.24	11.01 ± 0.17	10.65 ± 0.81	$\mathbf{5.11} \pm 1.34$	5.3 ± 1.51
	50	7.0 ± 1.34	6.42 ± 1.55	7.87 ± 2.92	11.07 ± 0.25	10.94 ± 0.58	4.27 ± 0.94	$\mathbf{4.08} \pm 0.8$
	100	6.59 ± 1.54	5.33 ± 1.2	7.31 ± 3.11	11.01 ± 0.23	10.71 ± 0.43	3.4 ± 0.48	$\mathbf{3.37} \pm 0.44$

$p < 0.05$ $p < 0.01$ $p < 0.001$

https://doi.org/10.52202/085713-2161

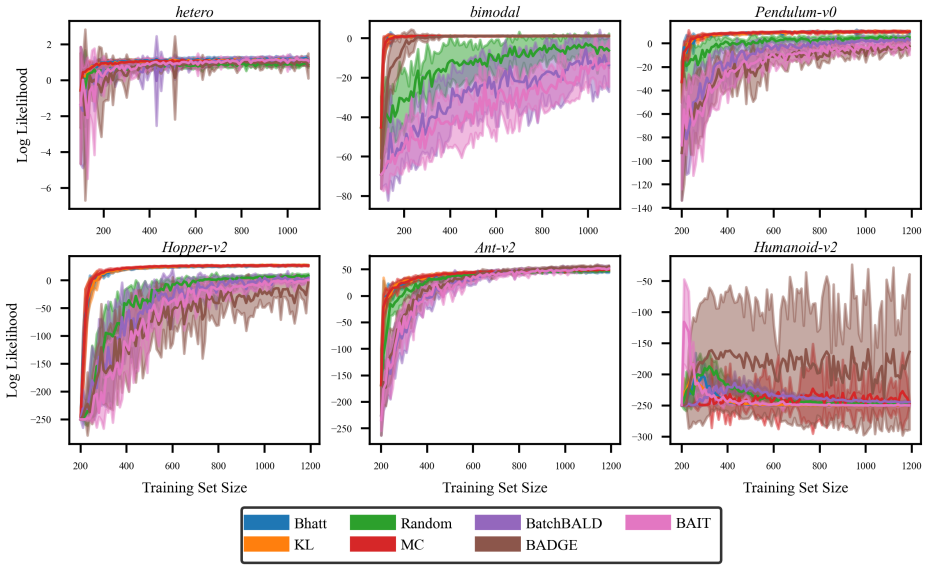


Figure 9: Mean Log Likelihood on test set as data was added to the training sets for Nflows Base.

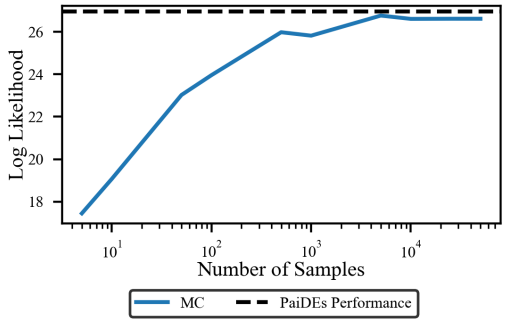


Figure 10: Log-Likelihood on the test set at the 100^{th} acquisition batch of the MC estimator on the *Hopper* environment as the number of samples increases, K, for Nflows Base. Experiment run across 10 seeds and the mean is being reported.

Table 8: Log Likelihood of a held out test set during training at different acquisition batches for Nflows Base. Experiments were across ten different seeds and the results are expressed as mean plus minus one standard deviation.

Env	Acq. Batch	Random	BatchBALD	BADGE	BAIT	MC (BALD)	KL (ours)	Bhatt (ours)
hetero	10	0.54 ± 0.27	0.21 ± 1.04	-0.67 ± 2.48	0.27 ± 0.7	0.95 ± 0.12	0.93 ± 0.14	**0.99** ± 0.12
	25	0.79 ± 0.18	0.72 ± 0.37	0.77 ± 0.26	0.66 ± 0.47	1.06 ± 0.06	1.06 ± 0.12	**1.11** ± 0.09
	50	0.8 ± 0.18	0.99 ± 0.17	0.95 ± 0.18	0.92 ± 0.25	1.08 ± 0.04	1.07 ± 0.16	**1.15** ± 0.14
	100	0.86 ± 0.14	1.11 ± 0.15	0.8 ± 0.7	1.1 ± 0.14	1.09 ± 0.1	1.12 ± 0.1	**1.14** ± 0.13
bimodal	10	-30.57 ± 12.61	-53.07 ± 7.95	-5.87 ± 8.38	-57.7 ± 11.26	1.08 ± 0.09	**1.11** ± 0.1	**1.11** ± 0.1
	25	-16.55 ± 9.04	-40.63 ± 18.86	1.0 ± 0.33	-41.4 ± 14.8	1.21 ± 0.09	1.17 ± 0.1	**1.27** ± 0.09
	50	-10.85 ± 8.5	-22.17 ± 13.04	1.21 ± 0.15	-32.98 ± 13.61	**1.22** ± 0.11	1.2 ± 0.1	1.21 ± 0.11
	100	-6.19 ± 8.66	-13.54 ± 13.5	**1.31** ± 0.16	-12.65 ± 11.91	1.26 ± 0.13	1.26 ± 0.1	1.26 ± 0.14
Pendulum	10	-5.55 ± 6.8	-47.16 ± 31.67	-53.04 ± 26.71	-61.82 ± 38.52	5.81 ± 3.29	**6.17** ± 1.86	5.8 ± 1.41
	25	-0.78 ± 5.92	-12.34 ± 16.57	-28.0 ± 16.65	-22.15 ± 8.47	**8.72** ± 0.45	8.45 ± 0.55	8.3 ± 0.7
	50	3.77 ± 1.8	-2.51 ± 5.06	-7.43 ± 6.72	-8.46 ± 4.94	9.77 ± 0.36	**9.95** ± 0.5	9.59 ± 0.27
	100	5.19 ± 2.45	3.04 ± 2.38	-2.29 ± 2.9	-4.41 ± 5.31	10.16 ± 0.7	**10.49** ± 0.23	9.58 ± 0.8
Hopper	10	-136.85 ± 53.59	-147.16 ± 77.99	-120.43 ± 77.99	-201.21 ± 50.44	**13.94** ± 5.54	11.59 ± 4.32	11.44 ± 2.31
	25	-28.0 ± 28.16	-94.48 ± 71.55	-118.09 ± 65.33	-88.87 ± 71.52	**23.18** ± 1.41	22.71 ± 0.86	22.29 ± 0.95
	50	-0.55 ± 6.82	-19.33 ± 25.02	-38.16 ± 25.17	-17.2 ± 9.63	**26.43** ± 1.37	25.72 ± 1.04	25.0 ± 1.6
	100	8.67 ± 3.39	4.53 ± 5.31	-4.16 ± 6.1	1.06 ± 3.61	26.62 ± 1.14	**26.96** ± 1.11	26.88 ± 0.91
Ant	10	-0.5 ± 16.68	-80.2 ± 46.22	-44.85 ± 20.71	-72.13 ± 37.99	24.48 ± 5.88	**25.41** ± 5.08	21.59 ± 9.0
	25	31.08 ± 5.4	17.06 ± 13.16	21.22 ± 10.61	23.12 ± 11.37	39.77 ± 3.34	**40.62** ± 2.85	37.58 ± 2.37
	50	43.37 ± 1.85	43.43 ± 4.81	40.05 ± 6.86	40.89 ± 7.56	**45.67** ± 1.76	45.07 ± 1.62	44.77 ± 1.39
	100	47.52 ± 1.36	54.38 ± 3.23	**55.48** ± 1.72	51.56 ± 3.68	49.29 ± 0.78	48.12 ± 2.05	46.52 ± 2.42
Humanoid	10	-200.61 ± 20.02	-243.1 ± 6.03	**-182.46** ± 99.5	-230.61 ± 9.92	-249.1 ± 0.69	-221.09 ± 10.68	-201.28 ± 22.78
	25	-223.54 ± 13.72	-221.22 ± 9.42	**-173.12** ± 89.96	-245.86 ± 1.8	-247.84 ± 1.74	-246.29 ± 1.92	-247.16 ± 1.04
	50	-244.15 ± 1.52	-236.24 ± 5.87	**-192.07** ± 77.55	-248.41 ± 0.93	-247.76 ± 2.11	-249.39 ± 0.43	-249.57 ± 0.62
	100	-247.82 ± 1.28	-244.58 ± 2.1	**-163.78** ± 124.9	-249.71 ± 0.32	-246.47 ± 2.99	-249.76 ± 0.4	-249.97 ± 0.05

$p < 0.05$ $p < 0.01$ $p < 0.001$

https://doi.org/10.52202/085713-2161

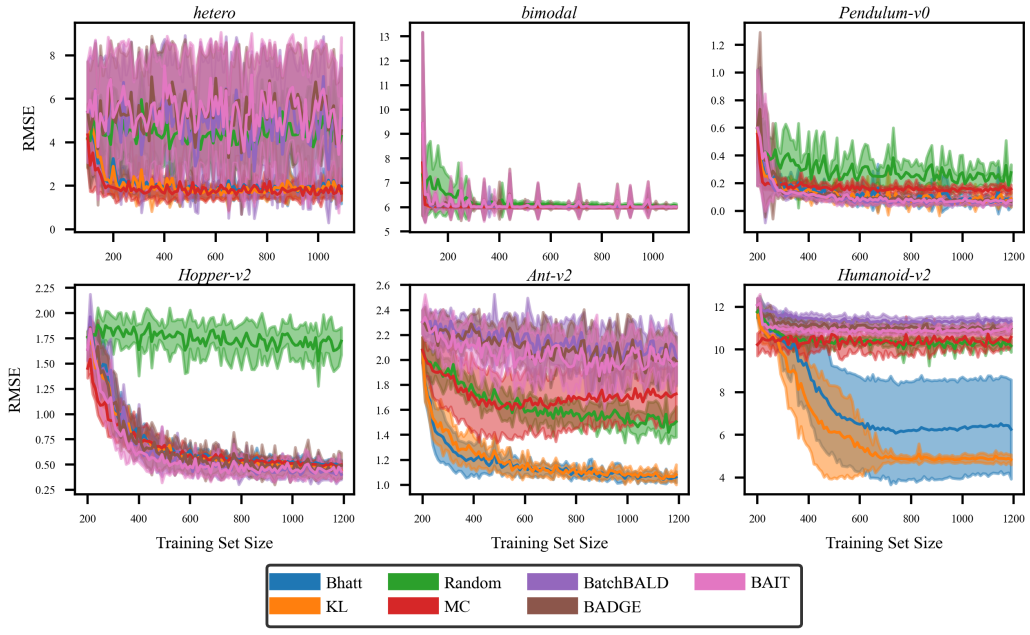


Figure 12: Mean RMSE on test set as data was added to the training sets for PNEs.

E Probabilistic Network Ensembles

In addition to ensembles of normalizing flows, we also explored the use of probabilistic network ensembles (PNEs) as a common approach for capturing uncertainty [12, 38, 39]. The PNEs were constructed by employing fixed dropout masks, where each ensemble component modeled a Gaussian distribution. The models were trained using negative log likelihood, with weights randomly initialized and bootstrapped samples from the training set to create diversity amongst the componenents. Our findings paralleled those of Nflows Base, in that PairEpEsts performed similarly or better than baselines and statistically significantly in higher dimensions. These results are presented in Figure 12 and Figure 13, as well as Table 9 and Table 10. Note that PNEs did not perform as well as Nflows Base as they are not as expressive this can be seen for the *bimodal* setting in Figure 11. Furthermore, we have included the 1D graphs illustrating the performance of PNEs in *hetero* and *bimodal* in Figure 11. Similarly to before, we also provide a comparison of the MC estimator as the number of samples increased in Figure 14. While we could have implemented ensembles with different output distributions to better fit the data, we decided to stick with Gaussians as they seem to be the most frequent choice [12, 38, 39]. Moreover, in order to pick the best distributional fit would require a hyper-parameter search or apriori knowledge whereas NFs will learn the best distributional fit.

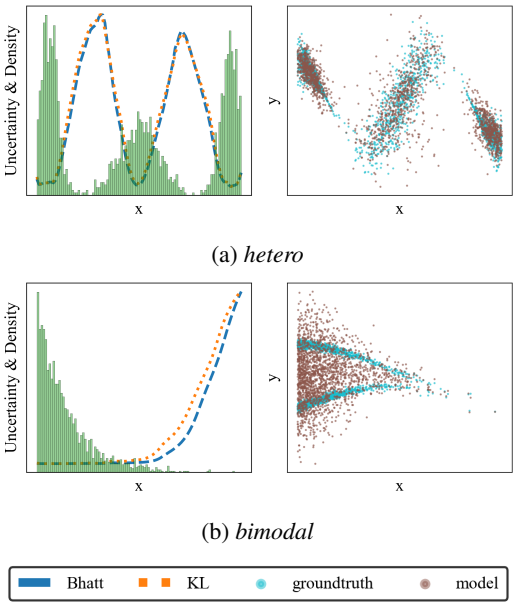


Figure 11: In the right graphs, the cyan dots are the ground-truth data and the brown dots are sampled from PNEs. The 2 lines depict epistemic uncertainty corresponding to different estimators. The left graphs depicts the ground-truth data as the blue dots and its corresponding density as the orange histogram. Note the legend refers to the lines in the right graphs.

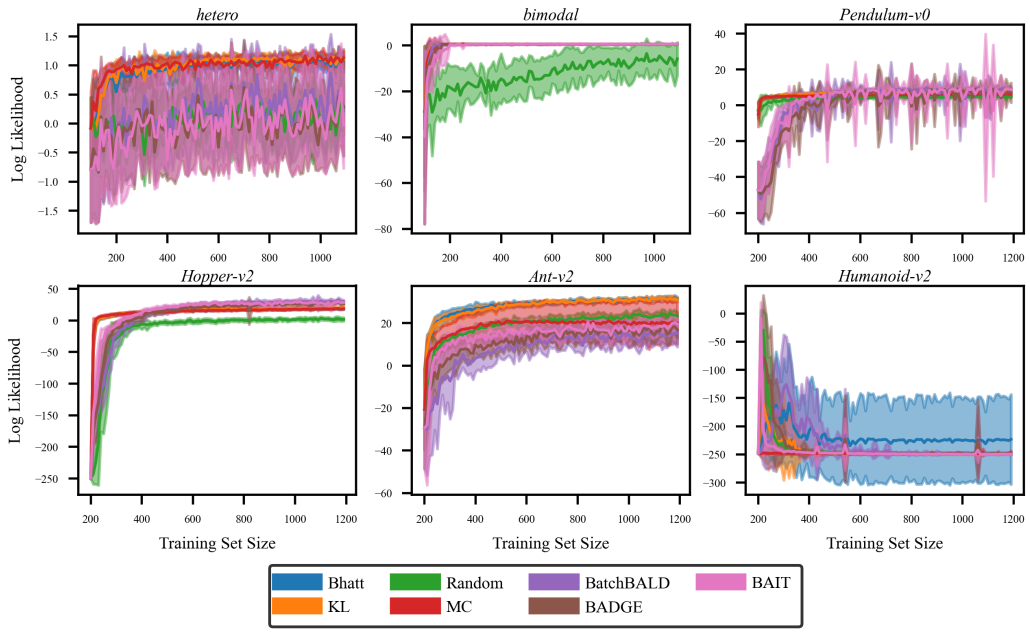


Figure 13: Mean Log Likelihood on the test set as data was added to the training sets for PNEs.

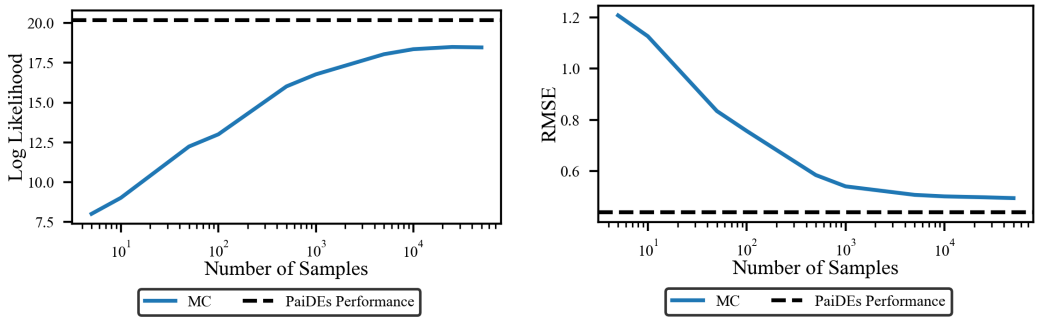


Figure 14: RMSE and Log-Likelihood on the test set at the 100^{th} acquisition batch of the MC estimator on the *Hopper* environment as the number of samples increases, K, for PNEs. Experiment run across 10 seeds and the mean is being reported.

https://doi.org/10.52202/085713-2161

Table 9: Mean RMSE on the test set for certain Acquisition Batches for PNEs. Experiments were across ten different seeds and the results are expressed as mean plus minus one standard deviation. Note that nothing was bolded for *bimodal* as many methods performed similarly.

Env	Acq. Batch	Random	BatchBALD	BADGE	BAIT	MC (BALD)	KL (ours)	Bhatt (ours)
hetero	10	3.9 ± 1.47	6.9 ± 1.15	6.91 ± 1.77	6.9 ± 1.58	$\mathbf{1.94 \pm 0.34}$	2.11 ± 0.34	2.36 ± 0.75
	25	4.42 ± 1.43	4.43 ± 2.81	4.06 ± 2.58	5.02 ± 2.91	$\mathbf{1.75 \pm 0.27}$	1.86 ± 0.39	1.96 ± 0.44
	50	3.96 ± 1.71	4.06 ± 2.64	5.12 ± 2.78	5.22 ± 2.93	1.78 ± 0.51	1.68 ± 0.34	$\mathbf{1.67 \pm 0.27}$
	100	4.28 ± 1.45	5.1 ± 2.92	4.54 ± 2.89	6.01 ± 2.82	$\mathbf{1.66 \pm 0.31}$	1.83 ± 0.51	1.98 ± 0.81
bimodal	10	6.52 ± 0.88	6.05 ± 0.07	6.19 ± 0.27	6.11 ± 0.15	6.01 ± 0.04	6.02 ± 0.04	6.01 ± 0.04
	25	6.1 ± 0.08	6.26 ± 0.75	6.31 ± 0.83	6.3 ± 0.88	6.01 ± 0.04	6.01 ± 0.04	6.01 ± 0.04
	50	6.13 ± 0.13	6.04 ± 0.07	6.02 ± 0.03	6.01 ± 0.04	6.01 ± 0.04	6.01 ± 0.04	6.01 ± 0.04
	100	6.06 ± 0.08	6.01 ± 0.03	6.03 ± 0.03	6.01 ± 0.04	6.01 ± 0.04	6.01 ± 0.04	6.01 ± 0.04
Pendulum	10	0.31 ± 0.11	0.15 ± 0.02	0.16 ± 0.05	$\mathbf{0.15 \pm 0.03}$	0.19 ± 0.04	0.18 ± 0.06	0.2 ± 0.07
	25	0.27 ± 0.09	$\mathbf{0.11 \pm 0.03}$	0.12 ± 0.03	$\mathbf{0.11 \pm 0.03}$	0.16 ± 0.02	0.12 ± 0.04	0.13 ± 0.04
	50	0.25 ± 0.07	$\mathbf{0.08 \pm 0.02}$	0.09 ± 0.02	$\mathbf{0.08 \pm 0.02}$	0.15 ± 0.03	$\mathbf{0.08 \pm 0.02}$	0.09 ± 0.03
	100	0.28 ± 0.06	$\mathbf{0.07 \pm 0.03}$	$\mathbf{0.07 \pm 0.02}$	0.08 ± 0.02	0.16 ± 0.03	0.09 ± 0.06	0.08 ± 0.05
Hopper	10	1.85 ± 0.14	1.44 ± 0.31	1.12 ± 0.29	1.16 ± 0.37	$\mathbf{0.94 \pm 0.16}$	1.19 ± 0.21	1.14 ± 0.18
	25	1.9 ± 0.14	0.6 ± 0.13	0.66 ± 0.19	$\mathbf{0.55} \pm 0.12$	0.73 ± 0.13	0.72 ± 0.08	0.71 ± 0.1
	50	1.83 ± 0.13	0.54 ± 0.15	0.55 ± 0.15	$\mathbf{0.51 \pm 0.12}$	0.57 ± 0.06	0.57 ± 0.06	0.57 ± 0.05
	100	1.72 ± 0.14	$\mathbf{0.4 \pm 0.05}$	0.49 ± 0.13	0.43 ± 0.05	0.5 ± 0.04	0.44 ± 0.04	0.49 ± 0.05
Ant	10	1.89 ± 0.12	2.27 ± 0.13	2.24 ± 0.12	2.15 ± 0.18	1.87 ± 0.25	1.62 ± 0.16	$\mathbf{1.52 \pm 0.18}$
	25	1.74 ± 0.13	2.15 ± 0.21	2.14 ± 0.18	2.12 ± 0.25	1.67 ± 0.25	1.43 ± 0.12	$\mathbf{1.39 \pm 0.12}$
	50	1.55 ± 0.1	2.2 ± 0.13	2.16 ± 0.14	2.11 ± 0.18	1.61 ± 0.22	$\mathbf{1.29 \pm 0.06}$	1.3 ± 0.04
	100	1.5 ± 0.12	2.01 ± 0.2	1.99 ± 0.2	1.97 ± 0.15	1.73 ± 0.22	1.26 ± 0.08	$\mathbf{1.24 \pm 0.04}$
Humanoid	10	10.73 ± 0.32	11.63 ± 0.19	11.22 ± 0.29	10.95 ± 0.17	10.41 ± 0.53	$\mathbf{9.95 \pm 1.1}$	10.65 ± 0.84
	25	10.38 ± 0.22	11.36 ± 0.26	11.07 ± 0.26	10.91 ± 0.25	10.44 ± 0.55	$\mathbf{6.54 \pm 2.13}$	8.22 ± 2.47
	50	10.29 ± 0.19	11.37 ± 0.3	10.98 ± 0.22	10.9 ± 0.25	10.34 ± 0.81	$\mathbf{4.91 \pm 0.24}$	6.33 ± 2.24
	100	10.17 ± 0.3	11.32 ± 0.16	10.99 ± 0.26	11.05 ± 0.21	10.57 ± 0.41	$\mathbf{4.84 \pm 0.24}$	6.25 ± 2.32

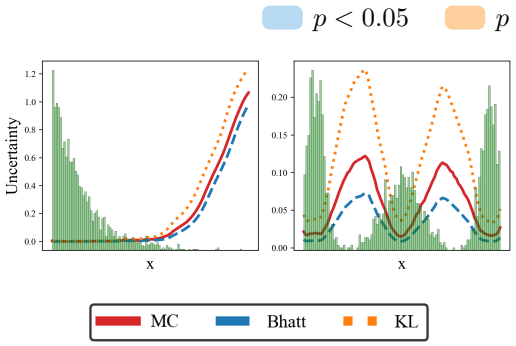


Figure 15: The bias introduced by PairEpEsts for PNEs compared to the MC method on the 1D environments.

Table 10: Log Likelihood on the test set during training at different acquisition batches for PNEs. Experiments were across ten different seeds and the results are expressed as mean plus minus one standard deviation. Note that no values were highlighted for Hopper despite the results being statistically significant. In this case, the results are statistically significantly worse and we found it misleading to highlight.

Env	Acq. Batch	Random	BatchBALD	BADGE	BAIT	MC (BALD)	KL (ours)	Bhatt (ours)
hetero	10	0.13 ± 0.57	-0.67 ± 0.55	-0.72 ± 0.61	-0.78 ± 0.59	**0.89 ± 0.1**	0.87 ± 0.15	0.74 ± 0.3
	25	0.03 ± 0.47	0.33 ± 0.76	0.38 ± 0.76	0.13 ± 0.82	**1.02 ± 0.11**	0.95 ± 0.14	0.92 ± 0.24
	50	0.23 ± 0.57	0.48 ± 0.71	0.14 ± 0.77	0.11 ± 0.85	1.01 ± 0.15	**1.1 ± 0.11**	1.09 ± 0.12
	100	0.23 ± 0.42	0.23 ± 0.78	0.28 ± 0.85	-0.03 ± 0.74	**1.1 ± 0.11**	1.11 ± 0.17	1.08 ± 0.2
bimodal	10	-17.71 ± 10.01	0.32 ± 0.33	0.41 ± 0.21	-1.36 ± 4.14	0.6 ± 0.1	0.59 ± 0.14	**0.61 ± 0.07**
	25	-20.78 ± 5.64	0.61 ± 0.19	0.48 ± 0.23	0.57 ± 0.21	0.66 ± 0.03	0.66 ± 0.03	**0.67 ± 0.04**
	50	-12.93 ± 8.55	0.66 ± 0.09	0.59 ± 0.16	0.61 ± 0.18	**0.69 ± 0.02**	**0.69 ± 0.03**	**0.69 ± 0.03**
	100	-5.79 ± 7.19	0.66 ± 0.1	0.64 ± 0.11	0.68 ± 0.08	**0.7 ± 0.02**	**0.7 ± 0.02**	**0.7 ± 0.02**
Pendulum	10	2.35 ± 2.94	-10.76 ± 7.28	-15.25 ± 8.86	-9.76 ± 9.09	4.94 ± 0.48	**5.51 ± 0.72**	5.21 ± 0.61
	25	4.19 ± 0.89	-0.4 ± 14.08	0.2 ± 13.03	4.68 ± 2.98	5.45 ± 0.43	6.21 ± 0.98	**6.23 ± 0.84**
	50	4.34 ± 0.75	8.34 ± 0.83	7.97 ± 0.74	**8.68 ± 0.72**	5.92 ± 0.32	7.65 ± 0.5	7.15 ± 0.76
	100	4.69 ± 0.56	**9.53 ± 0.96**	7.07 ± 5.49	8.8 ± 1.23	6.09 ± 0.42	7.66 ± 0.64	7.77 ± 0.68
Hopper	10	-31.94 ± 28.24	-26.65 ± 15.57	-12.96 ± 9.55	-10.81 ± 4.25	**9.36 ± 1.79**	8.55 ± 1.34	8.55 ± 1.88
	25	-5.74 ± 2.66	12.62 ± 4.32	13.88 ± 2.27	**17.29 ± 3.45**	13.31 ± 2.3	13.9 ± 1.3	13.93 ± 1.6
	50	-2.31 ± 3.08	27.28 ± 2.42	23.49 ± 2.97	**28.08 ± 2.76**	16.43 ± 1.27	17.6 ± 1.89	16.93 ± 1.43
	100	1.75 ± 2.83	30.35 ± 1.44	26.54 ± 2.19	**28.44 ± 1.52**	18.5 ± 0.65	20.19 ± 1.37	18.54 ± 1.23
Ant	10	11.83 ± 2.29	-4.91 ± 12.37	3.29 ± 6.43	10.2 ± 5.57	13.8 ± 5.87	22.13 ± 3.11	**23.32 ± 2.73**
	25	17.78 ± 2.28	5.79 ± 7.68	8.44 ± 3.68	11.97 ± 5.41	18.69 ± 9.06	26.98 ± 1.64	**27.66 ± 1.78**
	50	22.21 ± 2.11	12.24 ± 3.25	14.94 ± 3.3	16.16 ± 3.2	20.54 ± 8.88	**29.84 ± 1.09**	29.62 ± 0.7
	100	23.4 ± 1.85	15.26 ± 6.5	16.87 ± 3.75	19.6 ± 2.07	20.03 ± 9.87	31.24 ± 1.09	**31.63 ± 0.9**
Humanoid	10	-230.41 ± 19.87	**-156.59 ± 82.79**	-210.52 ± 53.23	-242.95 ± 2.96	-247.83 ± 1.71	-226.42 ± 46.74	-194.78 ± 73.74
	25	-245.45 ± 2.24	**-224.95 ± 37.12**	-245.82 ± 5.03	-247.3 ± 1.26	-247.83 ± 1.19	-247.39 ± 3.18	-232.34 ± 42.02
	50	-246.91 ± 1.28	-247.38 ± 3.28	-249.28 ± 0.52	-249.2 ± 0.52	-248.06 ± 1.82	-249.41 ± 0.73	**-223.49 ± 77.83**
	100	-247.74 ± 0.6	-248.97 ± 1.1	-249.4 ± 0.6	-249.66 ± 0.53	-247.25 ± 3.36	-249.91 ± 0.12	**-223.29 ± 79.8**

$p < 0.05$ $p < 0.01$ $p < 0.001$

https://doi.org/10.52202/085713-2161

F Introduction to Normalizing Flows

NFs are powerful non-parametric models that have demonstrated the ability to fit flexible multi-modal distributions [54, 53]. These models achieve this by transforming a simple base continuous distribution, such as Gaussian or Beta, into a more complex one using the change of variable formula. By enabling scoring and sampling from the fitted distribution, NFs find application across various problem domains. Let B represent the base distribution, a D-dimensional continuous random vector with $p_B(b)$ as its density function, and let $Y = g(B)$, where g is an invertible function with an existing inverse g^{-1}, and both g and g^{-1} are differentiable. Leveraging the change of variable formula, we can express the distribution of Y as follows:

$$p_Y(y) = p_B(g^{-1}(y))|\det(J(g^{-1}(y)))|, \tag{11}$$

where $J(\cdot)$ denotes the Jacobian, and $\det$ signifies the determinant. The first term on the right-hand side of Equation 11 governs the shape of the distribution, while $|\det(J(g^{-1}(y)))|$ normalizes it, ensuring the distribution integrates to one. Complex distributions can be effectively modeled by making $g(b)$ a learnable function with tunable parameters θ, denoted as $g_\theta(b)$. However, it is essential to select g carefully to guarantee its invertibility and differentiability. For examples of suitable choices, please refer to [47].

G Hypothesis Testing Details

We conducted Welch's t-tests to compare means (μ_i, μ_j) between different estimators, as this test relaxes the assumption of equal variances compared to other hypothesis tests [13]. The means for both KL and Bhatt were compared to each of the baseline methods: BatchBALD, BADGE, BAIT, MC and random. To control the family-wise error rate (FWER), we performed a Holm-Bonferroni correction across each setting, environment, and acquisition batch. This follows best practices to ensure our results are statistically significant and do not occur just by random chance.