
Variational Uncertainty Decomposition for In-Context Learning

Anonymous Author(s)

Affiliation

Address

email

Abstract

As large language models (LLMs) gain popularity in conducting prediction tasks in-context, understanding the sources of uncertainty in in-context learning becomes essential to ensuring reliability. The recent hypothesis of in-context learning performing predictive Bayesian inference opens the avenue for Bayesian uncertainty estimation, particularly for decomposing uncertainty into epistemic uncertainty due to lack of in-context data and aleatoric uncertainty inherent in the in-context prediction task. However, the decomposition idea remains under-explored due to the intractability of the latent parameter posterior from the underlying Bayesian model. In this work, we introduce a variational uncertainty decomposition framework for in-context learning without explicitly sampling from the latent parameter posterior, by optimising auxiliary inputs as probes to obtain an upper bound to the aleatoric uncertainty of an LLM’s in-context learning procedure. Through experiments on synthetic and real-world tasks, we show quantitatively and qualitatively that the decomposed uncertainties obtained from our method exhibit desirable properties of epistemic and aleatoric uncertainty.¹

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable abilities in natural language generation [10, 51, 60], and are being extended to a wide range of applications such as question answering [69], retrieval-augmented generation [32], information analysis [57, 43], and bandit problems [27]. In particular, an emergent property of an LLM is *in-context learning* (ICL), where the model acquires task behavior at inference time, without the need for prior pre-training or fine-tuning [7]. With the rising importance and presence of LLMs, understanding where and why these models are uncertain is essential in assessing their trustworthiness and robustness. A straightforward method of assessing uncertainty is to directly prompt the LLM to quantify the uncertainty of its outputs. However, this can be unreliable due to the overconfidence of language models [61]. Therefore, being able to faithfully quantify and determine the sources of uncertainties from the LLMs’ output can assist practitioners in better understanding and addressing the model’s limitations.

Recent work has hypothesised that ICL exhibits properties of Bayesian inference [66]. If we concatenate a dataset of a predictive task $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ and a test input x^* into a prompt, then we can view ICL as (approximately) inferring an implicit latent parameter θ for an underlying posterior distribution $p(\theta|\mathcal{D})$ and computing a posterior predictive distribution $p(y^*|x^*, \mathcal{D})$. This interpretation allows estimation of uncertainty through a Bayesian framework, which measures a model’s *total (predictive) uncertainty* by computing the entropy $\mathbb{H}[y^*|x^*, \mathcal{D}]$ or, in regression settings, the total variance $\text{Var}[y^*|x^*, \mathcal{D}]$. The total uncertainty can then be decomposed further into two

¹See supplementary for appendix and code.

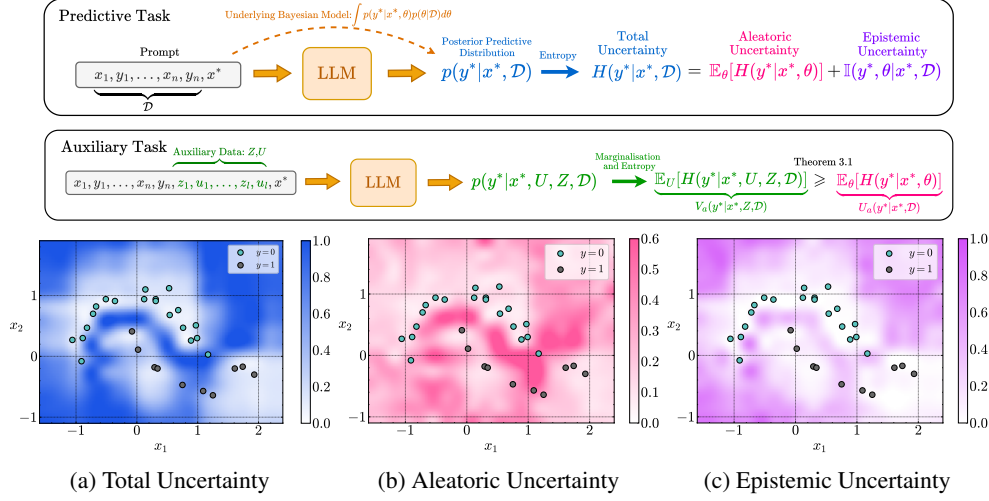


Figure 1: Uncertainty Decomposition with Auxiliary Data (Above).
Decomposition Example for Two-Moons Dataset (Below).

sources [25, 58]: *aleatoric uncertainty*, which captures noise inherent in the data generation process (thus irreducible), and *epistemic uncertainty* that accounts for uncertainty in the model due to the lack of knowledge (reducible with more data). In the bottom of Figure 1, we motivate the importance of a decomposition on the two-moons classification dataset. This decomposition provides valuable insights: aleatoric uncertainty pinpoints regions of ambiguity around the decision boundary, while epistemic uncertainty exposes areas lacking sufficient in-context data, guiding practitioners on where additional data or model refinement is needed. This notion of uncertainty decomposition has been explored in various domains, including computer vision [25, 26] and reinforcement learning [48, 11].

Obtaining high-quality Bayesian uncertainty estimates and decomposition for LLM-based ICL poses two major challenges. First, an LLM’s auto-regressive prediction procedure often does not satisfy the exchangeability condition [12, 70], which questions the existence of the implicit Bayesian model with latent parameter θ . Second, even if an implicit Bayesian model exists, one cannot explicitly simulate posterior samples $\theta \sim p(\theta|\mathcal{D})$, which are required by the uncertainty decomposition procedure in many existing Bayesian neural network methods [44, 6, 14, 21, 33]. In this regard, recent work on Martingale posterior [12] proposes generating a long sequence of future data and estimating a posterior distribution over θ via risk minimisation. But the Martingale posterior approach incurs a high computational cost and, still, the missing guarantee of exchangeability makes its uncertainty estimates questionable in aligning with the uncertainty from a coherent Bayesian model.

In this work, we propose a Variational Uncertainty Decomposition (VUD) framework for LLM-based ICL, focusing on addressing the mentioned two challenges. Our contributions are as follows:

- We propose an *optimisable* variational upper-bound to the aleatoric (predictive) uncertainty without explicit simulating the parameter posterior $p(\theta|\mathcal{D})$, by appending in optimisable auxiliary inputs \mathbf{Z} to the context and computing uncertainty measures with \mathbf{Z} conditioning. This variational estimator also induces a lower-bound on the epistemic uncertainty, which can be used in relevant tasks. An overview of our two-task variational decomposition pipeline can be found in the above of Figure 1.
- We propose novel LLM prompting and optimisation techniques for computing $p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$ and searching optimal \mathbf{Z} . Our design facilitates (approximate) exchangeability for ICL, making the variational uncertainty estimates better aligned with desirable Bayesian properties such as epistemic uncertainty reduction with increasing amount of data.

Experiments on synthetic regression and classification datasets show that our uncertainty decomposition framework is effective, behaving qualitatively similar to a Bayesian model. Quantitatively, the variational estimation of epistemic uncertainty also benefits downstream tasks such as bandit and out-of-distribution (OOD) detection applied to real-world natural language datasets.

2 Background

In-Context Learning and Bayesian Inference. A (pre-trained) LLM with weights ϕ parametrises a set of conditional distributions $\{p_\phi^i(\mathbf{t}_i|\mathbf{t}_{1:i-1})\}_{i \in \mathbb{N}^+}$ over tokens $\{\mathbf{t}_i\}_{i \in \mathbb{N}^+}$. Given a predictive

task of covariate-label pairs, $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, and test covariate \mathbf{x}^* , the ICL procedure with an LLM sets $(t_{2i-1}, t_{2i}) = (\mathbf{x}_i, \mathbf{y}_i)$ and $(t_{2n+1}, t_{2n+2}) = (\mathbf{x}^*, \mathbf{y}^*)$ and computes the predictive distribution as $p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = p_\phi^{2n+2}(t_{2n+2}|t_{1:2n+1})$. Now suppose the random variables $\mathbf{y}_{1:n}|\mathbf{x}_{1:n} \sim \prod_{i=1}^n p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{x}_{<i}, \mathbf{y}_{<i})$ (with $p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{x}_{<i}, \mathbf{y}_{<i}) = p_\phi^{2i}(t_{2i}|t_{1:2i-1})$) are *exchangeable*, namely for all permutations σ of $[n]$,

$$p(\mathbf{y}_{\sigma(1)}, \dots, \mathbf{y}_{\sigma(n)}|\mathbf{x}_{\sigma(1)}, \dots, \mathbf{x}_{\sigma(n)}) = p(\mathbf{y}_1, \dots, \mathbf{y}_n|\mathbf{x}_1, \dots, \mathbf{x}_n), \quad (1)$$

then by de Finetti's theorem [9] there exists a Bayesian model w.r.t. a parameter θ such that

$$p(\mathbf{y}_1, \dots, \mathbf{y}_n|\mathbf{x}_1, \dots, \mathbf{x}_n) = \int \prod_{i=1}^n p(\mathbf{y}_i|\mathbf{x}_i, \theta) p(\theta) d\theta. \quad (2)$$

Notably, the parameter θ here is defined *implicitly*. We discuss the link between ICL and Bayesian models as well as existing methods to promote exchangeability further in Appendix D and G. In particular, we design prompting and post-processing methods over LLM auto-regressive next token prediction in Section 3 to (approximately) achieve exchangeability (c.f. [12]).

Decomposing Predictive Uncertainty. Consider a *prescribed* Bayesian model $\mathbf{y}|\mathbf{x} \sim p(\mathbf{y}|\mathbf{x}, \theta)$ with prior $\theta \sim p(\theta)$. Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, we can (approximately) compute the posterior predictive distribution $p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \theta) p(\theta|\mathcal{D}) d\theta$. Then the predictive *total (entropic) uncertainty* is defined as $U(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathbb{H}[p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})]$, which can be decomposed further into *aleatoric uncertainty* $U_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$ and *epistemic uncertainty* $U_e(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$ [25]:

$$\underbrace{\mathbb{H}[p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})]}_{=:U(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})} = \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[\mathbb{H}[p(\mathbf{y}^*|\mathbf{x}^*, \theta)]]}_{=:U_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})} + \underbrace{\mathbb{I}(\mathbf{y}^*; \theta|\mathbf{x}^*, \mathcal{D})}_{=:U_e(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})}, \quad (3)$$

When $\mathbf{y}^* \in \mathbb{R}$, we can also compute the *total variance* of the prediction and perform a similar decomposition into *aleatoric and epistemic variances* by the tower rule property

$$\underbrace{\text{Var}[\mathbf{y}^*|\mathbf{x}, \mathcal{D}]}_{=:U^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})} = \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[\text{Var}[\mathbf{y}^*|\mathbf{x}^*, \theta]]}_{=:U_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})} + \underbrace{\text{Var}_{p(\theta|\mathcal{D})}[\mathbb{E}[\mathbf{y}^*|\mathbf{x}^*, \theta]]}_{=:U_e^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})}. \quad (4)$$

Typically, these decompositions are obtained by Monte Carlo estimation with (approximate) samples from $p(\theta|\mathcal{D})$ [29]. However, this approach poses a challenge when we don't have access to $p(\theta|\mathcal{D})$, which may occur if the Bayesian model is only implicitly defined [66] as in Eq. (2), or if sampling from $p(\theta|\mathcal{D})$ is prohibitively expensive.

3 Method

We present an alternative approach for uncertainty decomposition defined in (3) and (4), which sidesteps explicit posterior sampling of the parameter θ and thus, is suitable for implicitly defined Bayesian models. Although our practical algorithmic development focuses on LLM in-context learning on context $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ and test query \mathbf{x}^* , the decomposition technique applies to any Bayesian model *a la* de Finetti (2), including prescribed Bayesian models such as Bayesian linear regression and Gaussian processes (Appendix B).

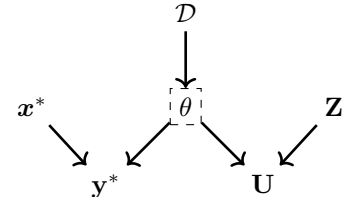


Figure 2: The DAG \mathcal{G} of the conditional independence assumptions.

3.1 Variational Estimates of Uncertainty Decomposition

Total Uncertainty Decomposition. Suppose we can directly compute (or approximate) the posterior predictive distribution $p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$ for arbitrary \mathcal{D} and \mathbf{x}^* . Now consider a set of *auxiliary inputs* $\mathbf{Z} = \{\mathbf{z}_l\}_{l=1}^L$, and corresponding outputs as $\mathbf{U} = \{\mathbf{u}_l\}_{l=1}^L$. Then we define the following *variational estimation* of the aleatoric uncertainty as:

$$V_a(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) := \mathbb{E}_{p(\mathbf{U}|\mathbf{Z}, \mathcal{D})}[\mathbb{H}[p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})]]. \quad (5)$$

To ensure consistency with an underlying Bayesian model (2), we assume that $\mathbf{x}^*, \mathbf{y}^*, \mathbf{Z}, \mathbf{U}, \mathcal{D}$ obey the conditional independence relations given by the directed acyclic graph (DAG) \mathcal{G} in Figure 2. This assumption allows us to prove the following theorem relating the variational estimation of the aleatoric uncertainty to the exact Bayesian estimate of aleatoric uncertainty.

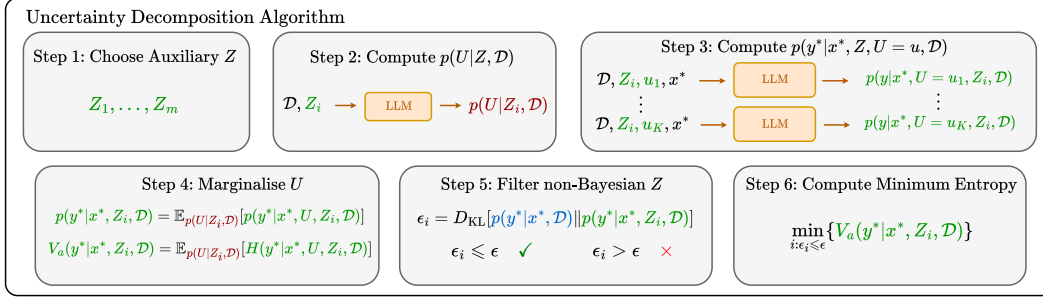


Figure 3: Variational Uncertainty Decomposition (VUD) Framework.

Theorem 3.1 (Aleatoric Uncertainty Upper-Bound). *If the conditional independence relations in \mathcal{G} hold, then the variational estimator provides an upper-bound to the aleatoric uncertainty:*

$$V_a(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \geq U_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}), \quad (6)$$

where the difference between the variational estimator and the true aleatoric uncertainty is:

$$\mathbb{E}_{p(\mathbf{y}^*, \mathbf{U}|\mathbf{x}^*, \mathbf{Z}, \mathcal{D})} [D_{\text{KL}}[p(\theta|\mathbf{y}^*, \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}) || p(\theta|\mathbf{U}, \mathbf{Z}, \mathcal{D})]] . \quad (7)$$

See Appendix A.1 for the proof. Importantly, the upper-bound (6) holds for *arbitrary* \mathbf{Z} which inspires the following optimisation procedure to obtain the best variational estimate:

$$V_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) := \min_{\mathbf{Z}} V_a(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}), \quad (8)$$

Since the aleatoric uncertainty is trivially upper-bounded by the total uncertainty in (3), we denote

$$\tilde{V}_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \min\{V_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}), \mathbb{H}[p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})]\},$$

as the *variational estimate of the aleatoric uncertainty*. We can obtain a *variational estimate for the epistemic uncertainty* by defining $V_e(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) := \mathbb{H}[p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})] - \tilde{V}_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$, which implies that $V_e(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) \leq U_e(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$. This motivates our Variational Uncertainty Decomposition approach illustrated in Figure 1.

Total Variance Decomposition. Similarly to (8), we can also construct a variational estimate for the aleatoric variance and derive a corresponding upper-bound. See Appendix A.2 for the proof.

Theorem 3.2 (Aleatoric Variance Upper-Bound). *If the conditional independence relation in \mathcal{G} holds, then the variational estimator provides an upper-bound to the estimation of aleatoric variance:*

$$V_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) := \mathbb{E}_{p(\mathbf{U}|\mathbf{Z}, \mathcal{D})} [\text{Var}[\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}]] \geq U_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}). \quad (9)$$

The best variational estimate is then $V_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) := \min_{\mathbf{Z}} V_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D})$, and a lower-bound of the epistemic variance is obtained as $V_e^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) := \text{Var}[\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}] - \tilde{V}_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$.

The effectiveness of this variational decomposition hinges on the choice of \mathbf{Z} to optimise (8). From the difference (7), we see that for an optimal choice of \mathbf{Z} , sufficient information regarding θ is provided by \mathbf{U} , \mathbf{Z} and \mathcal{D} , that also observing \mathbf{y}^* and \mathbf{x}^* does not provide much more certainty about θ . Therefore, the conditional entropy, $V_a(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D})$ is a suitable proxy for the true aleatoric uncertainty $U_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$, but remains an upper bound because some of the uncertainty in θ is absorbed into V_a . We discuss an alternative information theoretic interpretation in Section A.1.

3.2 Optimising the Variational Estimates and Promoting Exchangeability

The presented decomposition technique requires the model to be Bayesian *a la de* Finetti (2) and compatible with the DAG \mathcal{G} (Figure 2), which is not the case if naively prompting LLM for in-context learning. Specifically, exchangeability requires ensuring the following conditions [5, 70]:

- (C1) $p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{x}_{<i}, \mathbf{y}_{<i}) = p(\mathbf{y}_i|\mathbf{x}_i, \sigma(\mathbf{x}_{<i}, \mathbf{y}_{<i}))$ for all $i \in \mathbb{N}_+$ & all permutations σ on $[i]$;
- (C2) $p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) := \int p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})p(\mathbf{U}|\mathbf{Z}, \mathcal{D})d\mathbf{U} = p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$.

To promote exchangeability for LLM in-context learning, we propose two strategies tailored for the above conditions. First, to approximately achieve (C1), we construct the predictive distribution by shuffling the context and ensembling the LLM’s predictions, i.e., we define for context $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ and test query \mathbf{x}^* (with S_n a uniform distribution over the permutations on $[n]$):

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) := \frac{1}{L} \sum_{l=1}^L p_\phi^{2n+2}(\mathbf{y}^*|\mathbf{x}^*, \{\mathbf{x}_{\sigma_l(1)}, \mathbf{y}_{\sigma_l(1)}, \dots, \mathbf{x}_{\sigma_l(n)}, \mathbf{y}_{\sigma_l(n)}\}), \quad \sigma_l \sim S_n. \quad (10)$$

The other distributions $p(\mathbf{U}|\mathbf{Z}, \mathcal{D})$ and $p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})$ are defined in the same manner. For classification tasks, we evaluate the LLM logits to compute (10). However, in the regression case, we make a further Gaussian approximation to (10), which allows for easy computation of the entropy and marginalisation. Further details can be found in Appendix E.2. Then to approximately satisfy (C2), we restrict the search of \mathbf{Z} (Eq. (8)) to ensure the solution satisfies

$$D_{\text{KL}}[p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \parallel p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})] < \epsilon, \quad (11)$$

for some $\epsilon > 0$. Any metric or divergence on probability distributions will suffice for (11) but we choose KL divergence due to ease of computation. We filter out the \mathbf{Z} candidates that violate this KL constraint, hence we name this step as *KL filtering*. Choosing the number of permutations L and the threshold ϵ for KL filtering of \mathbf{Z} determines the accepted level of Bayesian approximation in the variational decomposition. While the selection of L is mainly determined by the computational resources, the choice of ϵ is further discussed in Appendix D.3.

Lastly, to reduce the search space of \mathbf{Z} for efficient computation, we restrict \mathbf{Z} to contain a single example in \mathbf{x} domain, i.e., $L = 1$ and $\mathbf{Z} = \mathbf{z}$, and design sampling techniques to obtain candidates for optimal \mathbf{Z} , including random sampling, setting $\mathbf{Z} = \mathbf{x}^*$, perturbing \mathbf{Z} around \mathbf{x}^* and a Bayesian optimisation strategy [59]. Empirically we find that perturbing \mathbf{Z} around \mathbf{x}^* works best for inputs that lie in a continuous space, which can partly be explained via the Gaussian process example in Appendix B. For natural language tasks such as question-answering (QA), we conduct the perturbation of \mathbf{z} by “rephrasing” \mathbf{x}^* with another LLM. Further details regarding the sampling procedures we explored for perturbing \mathbf{Z} are in Appendix C. Our overall step-by-step variational uncertainty decomposition framework (VUD) is depicted in Figure 3. Detailed decomposition algorithms for classification and regression tasks are provided in Appendix E.1.

4 Related Work

Our work takes inspiration from the growing body of literature connecting ICL to Bayesian inference [70, 66, 23, 35]. While much of the existing research centers on estimating a latent concept θ , often through methods like the Martingale posterior [12, 66], we take a different route by approximating conditional entropy and mutual information using auxiliary data. While our work is not the first to decompose predictive uncertainty in LLMs into aleatoric and epistemic components, prior approaches define these uncertainties differently from their traditional definitions in Bayesian deep learning [25, 11, 63]. Huo et al. [22] analyse how uncertainty changes when a prompt is modified with additional “clarifications.” While this is similar in spirit to our use of perturbations, we append perturbations to the ICL data rather than the predictive task itself. Moreover, their approach attributes aleatoric uncertainty solely to input ambiguity and does not incorporate a Bayesian framework, leading to a definition of uncertainty that diverges from the standard Bayesian interpretation. Ling et al. [34] assume a Bayesian approach but use alternative non-standard definitions of aleatoric and epistemic uncertainties. We provide a more detailed discussion of these related works, along with applications to OOD detection and bandit problems, in Appendix G.

5 Experiments

We evaluate the robustness and applicability of our method to classification and regression tasks. This includes ablation studies and visualisations on synthetic datasets, as well as downstream applications such as bandit problems and out-of-distribution (OOD) detection on question-answering (QA) tasks. We use the following LLMs in our experiments: Qwen2.5-14B/7B, [51] and Llama-3.1-8B [60]. Only for QA tasks, we use Qwen2.5-14B-Instruct. For conciseness, we show results for Qwen2.5-14B/14B-Instruct in the main text and the results for the remaining LLMs are given in Appendix F. Prompts and sampling details are provided in Appendix H.

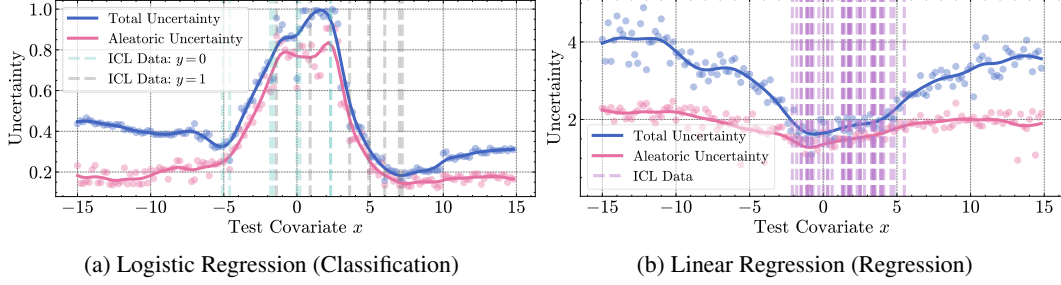


Figure 4: Uncertainty Decompositions for Logistic and Linear Regressions.

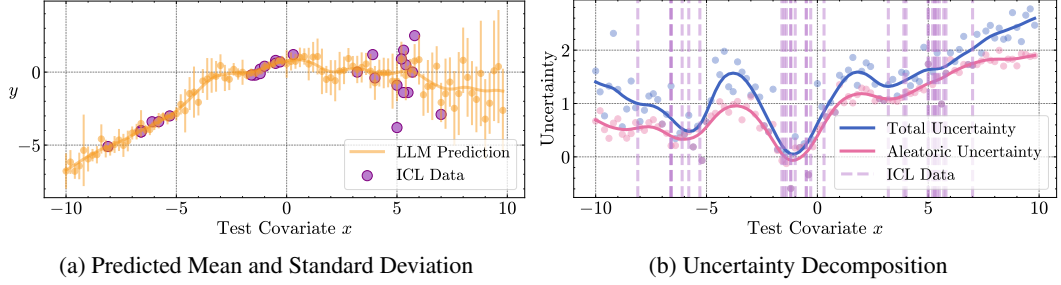


Figure 5: Uncertainty Decompositions for Regression Tasks with Gaps in ICL Data.

5.1 Synthetic Regression & Classification Datasets

We visualise the uncertainty decompositions on synthetic regression & classification datasets and conduct ablation studies on the effects of KL filtering and \mathbf{Z} choices. Further ablations regarding permuting the in-context examples and various LLMs are in Appendix D and C.

Visualisations. In Figures 4a and 4b, we visualise the VUD uncertainty decompositions for a 1-D logistic regression (classification) and a 1-D linear regression (regression) task, each conditioned on a set of $|\mathcal{D}| = 15$ in-context examples (vertical lines). We consider more complex tasks of the Two Moons dataset (class.) in Figure 1, a dataset with designated “gaps” and heteroscedastic noises in the in-context learning data (reg.) in Figure 5, and the Spirals dataset (multi-class class.) in Figure 6.

Across these examples, we observe similar qualitative characteristics of the uncertainty decomposition. The epistemic uncertainty (represented by the gap between the total and aleatoric uncertainty in the 1-D examples), is lowest in regions near demonstrations and increases as the distance to the in-context learning data increases. In the classification examples, the aleatoric uncertainty is sharply localised near the decision boundary of the problem where $p(y^*|x^*, \mathcal{D}) \approx 0.5$. In the regression setting of Figure 4b, we observe minimal change in the aleatoric uncertainty, which reflects the homoscedastic noise of the data observations. However, in Figure 5 where we have heteroscedastic noise, the model accurately distinguishes between regions of high and low heteroscedastic noise. These examples indicate that the model can correctly distinguish between uncertainty from inherent data noise and uncertainty arising from missing contextual information.

Ablations. In Figure 7, we analyse the behavior of uncertainty decompositions as a function of in-context dataset size $|\mathcal{D}|$ under a logistic regression setting. We consider both in-distribution test inputs ($x = 0, 5$, solid lines) and out-of-distribution test inputs ($x = -15, -10, -5, 10, 15$, dotted lines). As expected, Figure 7a shows decreasing epistemic uncertainty across all test covariates with increasing $|\mathcal{D}|$, since additional training examples reduce model uncertainty. The largest epistemic uncertainty occurs at out-of-distribution inputs ($x = -15, -10, -5, 10, 15$), while in-distribution inputs ($x = 0, 5$) consistently exhibit lower values. The decay is most rapid for in-distribution test points, suggesting that the model becomes confident more quickly when the test point distribution overlaps with the training data. In contrast, aleatoric uncertainty reported in Figure 7b remains relatively stable as $|\mathcal{D}|$ grows, particularly for out-of-distribution covariates. Notably, aleatoric uncertainty is highest for the decision boundary at $x = 0$, where the class overlap is greatest, and remains consistently elevated across all dataset sizes. Out-of-distribution points show slightly lower but stable aleatoric values, reflecting lower intrinsic class ambiguity at extreme covariates. The mild increase in aleatoric uncertainty for in-distribution points at small dataset sizes is likely due to model underfitting, which resolves as more data is provided.

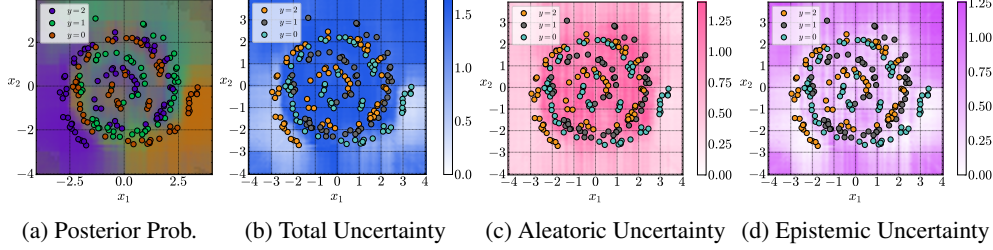


Figure 6: Uncertainty Decompositions for Spirals Classification Task.

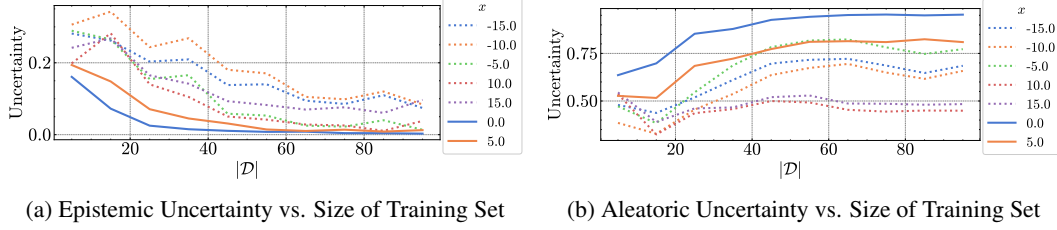


Figure 7: Uncertainty decompositions for logistic regression task with varying dataset size. Solid and dotted lines indicate in-distribution and out-of-distribution predictive points respectively.

In Figure 8, we compare the computed aleatoric uncertainty across different \mathbf{Z} sampling methods under the logistic regression setting. These include Perturb, where small noise is added to the test example to create \mathbf{Z} ; Repeated, where \mathbf{Z} is chosen to be the test example itself; Random, where \mathbf{Z} is sampled uniformly from the dataset; and Bayesian Optimisation (BO) [59], where \mathbf{Z} is actively selected to minimise a utility function related to the uncertainty. The aleatoric uncertainties reported in Figure 8a show that all these approaches track the total uncertainty curve around the decision boundary, indicating strong performance in capturing the local uncertainty landscape. Among them, Repeated returns the lowest variational aleatoric uncertainty estimate. Perturb also provides lower estimates, closely following the peak and providing stable estimates across the covariate space. Random sampling shows an upward trend in low ICL density regions far from the decision boundary, indicating poor stability. Regarding the KL divergence (11) achieved by the selected \mathbf{Z} in Figure 8b, Random and BO consistently have the lowest KL divergence across the majority of test samples, followed by the Perturb method which is significantly faster than BO. The Repeated sampling method yields higher KL values than Perturb, indicating greater deviation from the predictive posterior and is thus less aligned with Bayesian principles. These evidences support Perturb as a scalable and well-performing approach for sampling candidate \mathbf{Z} in (8)’s optimisation procedure.

5.2 Downstream Applications of Uncertainty Decomposition

We conduct quantitative experiments on two applications of uncertainty decomposition: bandit problems and out-of-distribution detection in real-world question-answering tasks.

Bandits. Bandit problems in reinforcement learning necessitate the ability to distinguish between aleatoric and epistemic uncertainty to balance exploration and exploitation. In a bandit problem, for a trial t , an agent must choose an arm $a_t \in \mathcal{A}$ which gives a reward r_t . The goal is to minimise the overall regret over all the trials $\sum_t \mu_t^* - \mathbb{E}[r_t]$, where μ_t^* is the mean reward from the optimal arm. We consider the Upper Confidence Bound (UCB) bandit algorithms [2], where $a_t = \operatorname{argmax}_a Q_t(a) + \alpha U_t(a)$, where Q_t is the estimated reward from arm a and U_t is the uncertainty in arm a at trial t , and α is the exploration rate. We use the LLM posterior mean as Q_t , and compare the performance of epistemic and total variance as U_t . In this setting, epistemic variance guides exploration to choose arms where additional data is beneficial, whereas total variance may prioritise actions where the reward has high intrinsic noise. We use the multi-armed bandit “Buttons” task [27], with 5 arms, where each arm a yields a Bernoulli reward with mean p_a . The base reward level p controls the overall success probability, with the optimal arm set to $p_a^* = p + \frac{\Delta}{2}$ and all other arms set to $p_a = p - \frac{\Delta}{2}$, where Δ denotes the reward gap between the optimal and suboptimal arms. We set $\Delta = 0.2$, which is the “hard” setting in [27]. When $p > 0.5$, the reward for the optimal arm will have the lowest (aleatoric) variance, and UCB algorithms using total variance will choose more suboptimal actions. We use mean regret and worst-case mean regret (from the 30% of worst performing seeds)

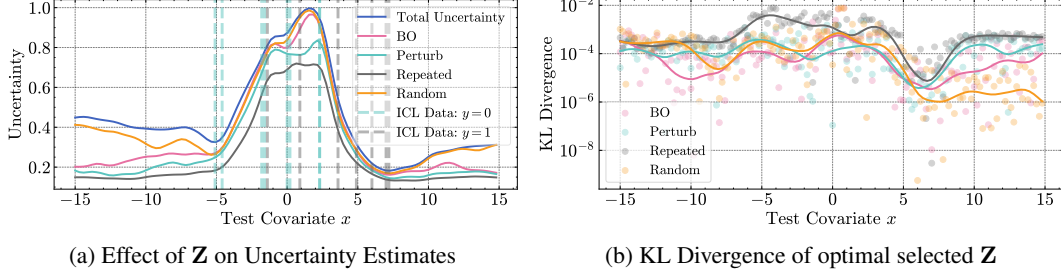


Figure 8: Ablation of \mathbf{Z} choice on Aleatoric Uncertainty and KL Divergence.

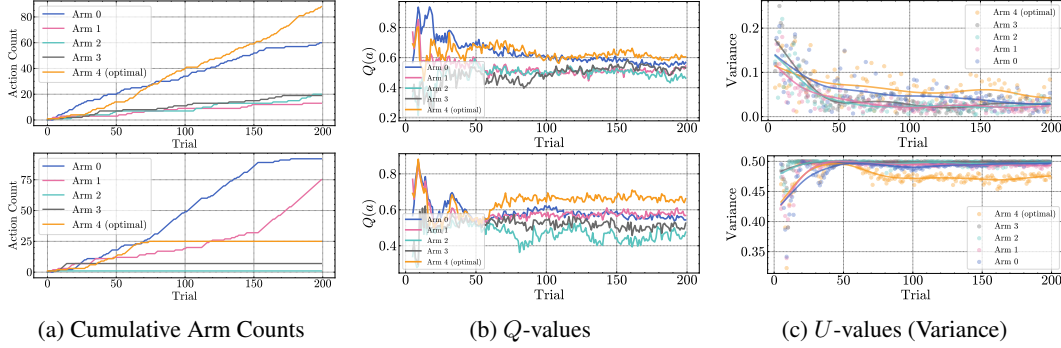


Figure 9: Example Run ($p = 0.6, \alpha = 5$) with Epistemic (above) and Total Variance (below).

as the primary performance metrics as well as metrics of median reward, suffix-fail frequency and $K \cdot \text{MinFrac}$ used in [27]. We also include UCB1 and Greedy as a non-LLM baseline, and the instruction prompting method from [27] as an LLM-based non-uncertainty baseline. See Appendix F.3 for further details on metrics, results and implementation of the UCB algorithm.

Figure 9 shows a typical run of epistemic variance (EV) and total variance (TV) for a particular seed. In both examples, the Q value for the optimal arm is the highest in the last 50 trials (Figure 9b) and thus should be chosen. But when we consider the arms chosen, the optimal arm is not picked in the last 50 trials for the TV run (Figure 9a). This is because the epistemic variance decreases with the number of observations for EV but not for TV (Figure 9c). Table 1 shows our experimental results on the Buttons task. We see for $p > 0.5$, the worst-case regret is significantly lower for EV than TV, indicating that the UCB algorithms is more robust for EV. Furthermore, EV generally results in lower mean regret for $p > 0.5$ with the exception of $p = 0.6, \alpha = 2$. However, it is important to note bandit algorithms have high variance in mean regret due to the stochasticity of the reward.

OOD Detection in Question-Answering. We perform out-of-distribution (OOD) detection via area under the ROC curve (AUC) [17] in natural language question-answering tasks. Our goal is to demonstrate that leveraging epistemic uncertainty from our decomposition yields higher OOD detection accuracy than directly utilising the total uncertainty. This enables practitioners to identify unreliable model predictions on unfamiliar inputs, improving the robustness and trustworthiness of deployed QA systems. In our experiments, we leverage BoolQA [8], HotpotQA [69], and PubMedQA [24] interchangeably of equivalent sample size as the in-distribution (ID) and out-of-distribution (OOD) datasets [39]. We formulate these datasets as binary classification tasks (yes/no). For our reference baseline, we extend the Deep Ensembles framework [22] to our OOD detection task by ensembling the output distributions of multiple different in-context example sets. For both methods, we leverage a training set size of $|\mathcal{D}| = 15$ ICL samples and a test set size of $|\mathbf{x}_{\text{ID}}^* + \mathbf{x}_{\text{OOD}}^*| = 120$ for our ID and OOD samples and average our experimental results across 3 seeds. For our method, we generate $|\mathbf{Z}| = 20$ perturbations by prompting the LLM to rephrase with relevant context from the test sample. For Deep Ensembles, we leverage 5 different in-context learning sets. Further details regarding setup can be found in Appendix F.4.

Before our discussion, a note that OOD detection from an ICL perspective can be particularly challenging. Traditionally, OOD detection leverages the entire training set to train the model [16, 17]. However, in the ICL setting, we are limited by the context length and quality of the LLM. Another issue that persists is guaranteeing that the QA datasets are semantically different enough where their distribution differs. Despite the difficulties, in Table 2, we observe that for our method, epistemic

Table 1: Buttons Bandit Problem. TV is Total Variance and EV is Epistemic Variance.

	METHOD	MEAN WORST-CASE REGRET ↓	MEAN REGRET ↓	MEDIAN REWARD ↑	SuffFailFreq($T/2$) ↓	$K \cdot \text{MinFrac}$ ↓
$p = 0.5$	UCB	0.128 \pm .019	0.094 \pm .027	0.510	0.0	0.29
	GREEDY	0.199 \pm .000	0.101 \pm .092	0.525	0.460	0.03
	INSTRUCT BASELINE	0.161 \pm .020	0.107 \pm .043	0.495	0.0	0.26
	TV ($\alpha = 2$)	0.196 \pm .005	0.100 \pm .074	0.492	0.3	0.03
	EV ($\alpha = 2$)	0.147\pm.000	0.087\pm.051	0.522	0.0	0.12
	TV ($\alpha = 5$)	0.198 \pm .000	0.100\pm.074	0.492	0.7	0.04
	EV ($\alpha = 5$)	0.152\pm.011	0.124 \pm .024	0.510	0.0	0.60
$p = 0.6$	UCB1	0.127 \pm .018	0.094 \pm .027	0.610	0.0	0.28
	GREEDY	0.199 \pm .000	0.092 \pm .090	0.645	0.396	0.03
	INSTRUCT BASELINE	0.111 \pm .007	0.076 \pm .043	0.620	0.0	0.18
	TV ($\alpha = 2$)	0.198 \pm .001	0.035\pm.054	0.670	0.1	0.04
	EV ($\alpha = 2$)	0.149\pm.039	0.068 \pm .042	0.642	0.0	0.145
	TV ($\alpha = 5$)	0.199 \pm .000	0.158 \pm .065	0.555	0.8	0.04
	EV ($\alpha = 5$)	0.140\pm.013	0.105\pm.027	0.600	0.0	0.42
$p = 0.7$	UCB1	0.122 \pm .017	0.094 \pm .027	0.710	0.0	0.27
	GREEDY	0.199 \pm .000	0.085 \pm .089	0.760	0.369	0.03
	INSTRUCT BASELINE	0.132 \pm .043	0.087 \pm .040	0.703	0.0	0.18
	TV ($\alpha = 2$)	0.199 \pm .000	0.076 \pm .087	0.725	0.3	0.03
	EV ($\alpha = 2$)	0.092\pm.004	0.050\pm.033	0.735	0.0	0.11
	TV ($\alpha = 5$)	0.195 \pm .003	0.151 \pm .073	0.603	0.7	0.04
	EV ($\alpha = 5$)	0.135\pm.007	0.092\pm.037	0.682	0.0	0.24

Table 2: Out-of-Distribution Detection AUC scores on QA tasks. Higher AUC values for epistemic uncertainty highlights the effectiveness of the uncertainty decomposition.

		AUC ↑ (DEEP ENSEMBLES)			AUC ↑ (OURS)		
ID/OOD		BOOLQA	HOTPOTQA	PUBMEDQA	BOOLQA	HOTPOTQA	PUBMEDQA
BOOLQA	TU	—	0.343\pm.000	0.604 \pm .000	—	0.355 \pm .000	0.570\pm.000
	EU	—	0.347 \pm .001	0.619\pm.002	—	0.600\pm.001	0.395 \pm .000
HOTPOTQA	TU	0.677\pm.000	—	0.684\pm.000	0.712 \pm .002	—	0.754 \pm .002
	EU	0.659 \pm .000	—	0.638 \pm .001	0.780\pm.002	—	0.775\pm.002
PUBMEDQA	TU	0.666\pm.000	0.360\pm.000	—	0.679\pm.004	0.382 \pm .002	—
	EU	0.606 \pm .002	0.329 \pm .001	—	0.471 \pm .001	0.483\pm.001	—

uncertainty (EU) yields higher AUC scores in more ID/OOD settings than total uncertainty (TU), implying better OOD detection results via our decomposition. When compared to Deep Ensembles, we notice that 1) the AUC scores for EU are considerably lower and 2) the AUC of the decomposed EU often underperforms when compared to its own TU.

6 Conclusion

In this work, we introduce the Variational Uncertainty Decomposition framework for ICL in LLMs. Motivated by a Bayesian view of ICL, we use auxiliary data to derive a variational upper bound to the aleatoric uncertainty and variance. This permits the estimation of the aleatoric uncertainty and variance, without requiring an estimation of the latent Bayesian parameter θ . Through extensive experiments using synthetic toy and real-world datasets, we demonstrate that our method provides a sensible decomposition that qualitatively and quantitatively respects properties of epistemic and aleatoric uncertainties. These results show that our method is capable of accurately distinguishing between aleatoric and epistemic uncertainty across a variety of LLMs.

Limitations. We assume that ICL behaves in a Bayesian manner. Whilst there is some evidence to support this Bayesian hypothesis [66, 70, 42], it has also been observed that in longer sampling horizons this Bayesian hypothesis breaks down [12, 35]. We address this by considering short sampling horizons, permutations, and a filtering step to remove “non-Bayesian” samples. However, whilst the filtering condition is necessary for a Bayesian model, it is not sufficient and doesn’t guarantee Bayesian behaviour. Therefore, we view our method as approximately Bayesian where ϵ is a quantification of the Bayesian approximation. Secondly, we focus on regression and classification tasks where the output of the task is a real number or a small set of classes and our prompt structure ensures short responses. In many real-world settings, the LLM output is in natural language where responses can differ in tokens but have the same semantic meaning. Therefore, uncertainty quantification methods that consider semantics [28] can be integrated with the VUD algorithm to obtain a posterior over the natural language response, and we leave this as future work.

Broader Impact. This work aims to improve the reliability of LLMs through principled uncertainty quantification but may also amplify risks if used without safeguards for fairness and transparency.

References

- [1] Dilip Arumugam and Thomas L Griffiths. Toward efficient exploration by large language model agents. *arXiv preprint arXiv:2504.20997*, 2025.
- [2] Peter Auer. Using upper confidence bounds for online learning. In *Proceedings 41st annual symposium on foundations of computer science*, pages 270–279. IEEE, 2000.
- [3] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International conference on machine learning*, pages 263–272. PMLR, 2017.
- [4] Oleksandr Balabanov and Hampus Linander. Uncertainty quantification in fine-tuned llms using lora ensembles. *arXiv preprint arXiv:2402.12264*, 2024.
- [5] Patrizia Berti, Luca Pratelli, and Pietro Rigo. Limit theorems for a class of identically distributed random variables. *The Annals of Probability*, 32(3A):2029–2052, 2004.
- [6] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.
- [7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
- [8] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [9] Bruno de Finetti. Funzione caratteristica di un fenomeno aleatorio. In *Atti del Congresso Internazionale dei Matematici*, pages 179–190, 1929.
- [10] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojuan Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanbiao Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha,

368 Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou,
369 Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang,
370 Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong
371 Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu,
372 Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report,
373 2025.

374 [11] Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft.
375 Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning,
376 2018.

377 [12] Fabian Falck, Ziyu Wang, and Chris Holmes. Is in-context learning in large language models
378 bayesian? a martingale perspective, 2024.

379 [13] Edwin Fong, Chris Holmes, and Stephen G Walker. Martingale posterior distributions. *Journal*
380 *of the Royal Statistical Society Series B: Statistical Methodology*, 85(5):1357–1391, 2023.

381 [14] Alex Graves. Practical variational inference for neural networks. *Advances in neural information*
382 *processing systems*, 24, 2011.

383 [15] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and
384 David Sontag. Tabllm: Few-shot classification of tabular data with large language models,
385 2023.

386 [16] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza
387 Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-
388 world settings, 2022.

389 [17] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution
390 examples in neural networks, 2018.

391 [18] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedziec, Rishabh Krishnan, and
392 Dawn Song. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint*
393 *arXiv:2004.06100*, 2020.

394 [19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier
395 exposure. *arXiv preprint arXiv:1812.04606*, 2018.

396 [20] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv*
397 *preprint arXiv:1309.6835*, 2013.

398 [21] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable
399 learning of bayesian neural networks. In *International conference on machine learning*, pages
400 1861–1869. PMLR, 2015.

401 [22] Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decom-
402 posing uncertainty for large language models through input clarification ensembling, 2024.

403 [23] Hong Jun Jeon, Jason D Lee, Qi Lei, and Benjamin Van Roy. An information-theoretic analysis
404 of in-context learning. *arXiv preprint arXiv:2401.15530*, 2024.

405 [24] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA:
406 A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent
407 Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods*
408 *in Natural Language Processing and the 9th International Joint Conference on Natural Lan-*
409 *guage Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China, November 2019.
410 Association for Computational Linguistics.

411 [25] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for
412 computer vision?, 2017.

413 [26] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh
414 losses for scene geometry and semantics, 2018.

- 415 [27] Akshay Krishnamurthy, Keegan Harris, Dylan J. Foster, Cyril Zhang, and Aleksandrs Slivkins.
416 Can large language models explore in-context?, 2024.
- 417 [28] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances
418 for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*,
419 2023.
- 420 [29] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable
421 predictive uncertainty estimation using deep ensembles. *Advances in neural information*
422 *processing systems*, 30, 2017.
- 423 [30] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.
- 424 [31] Hyungi Lee, Eunggu Yun, Giung Nam, Edwin Fong, and Juho Lee. Martingale posterior neural
425 processes. *arXiv preprint arXiv:2304.09431*, 2023.
- 426 [32] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman
427 Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and
428 Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.
- 429 [33] Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. Stochastic expectation
430 propagation. *Advances in neural information processing systems*, 28, 2015.
- 431 [34] Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyu Sun, Mika Oishi,
432 Takao Osaki, Katsushi Matsuda, Jie Ji, et al. Uncertainty quantification for in-context learning
433 of large language models. *arXiv preprint arXiv:2402.10189*, 2024.
- 434 [35] Shang Liu, Zhongze Cai, Guanting Chen, and Xiaocheng Li. Towards better understand-
435 ing of in-context learning ability from in-context uncertainty quantification. *arXiv preprint*
436 *arXiv:2405.15115*, 2024.
- 437 [36] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically
438 ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv*
439 *preprint arXiv:2104.08786*, 2021.
- 440 [37] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances*
441 *in neural information processing systems*, 31, 2018.
- 442 [38] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction.
443 *arXiv preprint arXiv:2002.07650*, 2020.
- 444 [39] Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. Uncertainty in gradient
445 boosting via ensembles, 2021.
- 446 [40] Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyu Sun, Wei Li, and Yixuan Li. Delving into out-of-
447 distribution detection with vision-language representations. *Advances in neural information*
448 *processing systems*, 35:35087–35102, 2022.
- 449 [41] Giovanni Monea, Antoine Bosselut, Kianté Brantley, and Yoav Artzi. Llms are in-context
450 reinforcement learners. *arXiv preprint arXiv:2410.05362*, 2024.
- 451 [42] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter.
452 Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.
- 453 [43] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad
454 Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large
455 language models, 2024.
- 456 [44] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science &
457 Business Media, 2012.
- 458 [45] Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta
459 learning via sequence modeling. *arXiv preprint arXiv:2207.04179*, 2022.

- [46] Allen Nie, Yi Su, Bo Chang, Jonathan N Lee, Ed H Chi, Quoc V Le, and Minmin Chen. Evolve: Evaluating and optimizing llms for exploration. *arXiv preprint arXiv:2410.06238*, 2024.
- [47] Emre Onal, Klemens Flöge, Emma Caldwell, Arsen Sheverdin, and Vincent Fortuin. Gaussian stochastic weight averaging for bayesian low-rank adaptation of large language models. *arXiv preprint arXiv:2405.03425*, 2024.
- [48] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.
- [49] Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.
- [50] Madhur Panwar, Kabir Ahuja, and Navin Goyal. In-context learning through the bayesian prism. *arXiv preprint arXiv:2306.04891*, 2023.
- [51] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.
- [52] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.
- [53] Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. Out-of-distribution detection and selective generation for conditional language models. *arXiv preprint arXiv:2209.15558*, 2022.
- [54] James Requeima, John Bronskill, Dami Choi, Richard Turner, and David K Duvenaud. Llm processes: Numerical predictive distributions conditioned on natural language. *Advances in Neural Information Processing Systems*, 37:109609–109671, 2024.
- [55] Karthik Abinav Sankararaman, Sinong Wang, and Han Fang. Bayesformer: Transformer with uncertainty estimation. *arXiv preprint arXiv:2206.00826*, 2022.
- [56] Remo Sasso, Michelangelo Conserva, and Paulo Rauber. Posterior sampling for deep reinforcement learning. In *International Conference on Machine Learning*, pages 30042–30061. PMLR, 2023.
- [57] Jacob Si, Wendy Yusi Cheng, Michael Cooper, and Rahul G. Krishnan. Interpretabnet: Distilling predictive signals from tabular data by salient feature interpretation, 2024.
- [58] Freddie Bickford Smith, Jannik Kossen, Eleanor Trollope, Mark van der Wilk, Adam Foster, and Tom Rainforth. Rethinking aleatoric and epistemic uncertainty. *arXiv preprint arXiv:2412.20892*, 2024.
- [59] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms, 2012.
- [60] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.
- [61] Bingbing Wen, Chenjun Xu, Robert Wolfe, Lucy Lu Wang, Bill Howe, et al. Mitigating overconfidence in large language models: A behavioral lens on confidence estimation and calibration. In *NeurIPS 2024 Workshop on Behavioral Machine Learning*, 2024.
- [62] Rand R Wilcox. *Introduction to robust estimation and hypothesis testing*. Academic press, 2011.

- 507 [63] Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. Quantifying
508 aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual
509 information appropriate measures? In *Uncertainty in artificial intelligence*, pages 2282–2292.
510 PMLR, 2023.
- 511 [64] Yue Wu, Xuan Tang, Tom M Mitchell, and Yuanzhi Li. Smartplay: A benchmark for llms as
512 intelligent agents. *arXiv preprint arXiv:2310.01557*, 2023.
- 513 [65] Yijun Xiao and William Yang Wang. On hallucination and predictive uncertainty in conditional
514 language generation. *arXiv preprint arXiv:2103.15025*, 2021.
- 515 [66] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of
516 in-context learning as implicit bayesian inference, 2022.
- 517 [67] Keyang Xu, Tongzheng Ren, Shikun Zhang, Yihao Feng, and Caiming Xiong. Unsupervised
518 out-of-domain detection via pre-trained transformers. *arXiv preprint arXiv:2106.00948*, 2021.
- 519 [68] Adam X. Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank
520 adaptation for large language models, 2024.
- 521 [69] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhut-
522 dinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop
523 question answering, 2018.
- 524 [70] Naimeng Ye, Hanming Yang, Andrew Siah, and Hongseok Namkoong. Exchangeable sequence
525 models can naturally quantify uncertainty over latent concepts. *arXiv preprint arXiv:2408.03307*,
526 2024.
- 527 [71] Liyi Zhang, R Thomas McCoy, Theodore R Sumers, Jian-Qiao Zhu, and Thomas L Griffiths.
528 Deep de finetti: Recovering topic distributions from large language models. *arXiv preprint*
529 *arXiv:2312.14226*, 2023.
- 530 [72] Siyan Zhao, Tung Nguyen, and Aditya Grover. Probing the decision boundaries of in-context
531 learning in large language models. *Advances in Neural Information Processing Systems*,
532 37:130408–130432, 2024.
- 533 [73] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use:
534 Improving few-shot performance of language models. In *International conference on machine*
535 *learning*, pages 12697–12706. PMLR, 2021.
- 536 [74] Wenxuan Zhou, Fangyu Liu, and Muhao Chen. Contrastive out-of-distribution detection for
537 pretrained transformers. *arXiv preprint arXiv:2104.08812*, 2021.

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 abstract and introduction state the claims made and are justified thoroughly with proofs and experiments throughout the paper.

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: We discuss the limitations of our method in the conclusion.

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: Complete proofs of our theory are included and discussed in the Appendix.

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: Detailed experimental setup, methodologies, and chosen parameters are outlined in the main text and the appendix.

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).
 - (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 anonymized and zipped along with our submission.

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The above is specified in the main text with further details in the 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: Standard deviations are included throughout our downstream experiments on real world examples.

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: Implementation details regarding compute resources are included in the appendix.

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 authors have reviewed and comply with the NeurIPS Code of Ethics

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?

Answer: [Yes]

Justification: A broader impact statement is provided in the conclusion.

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: Our paper demonstrates foundational research tested on publicly available datasets that were designed to test machine learning algorithms.

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: We properly acknowledge and cite all assets and resources used in the paper.

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.

799 • If this information is not available online, the authors are encouraged to reach out to
800 the asset’s creators.

801 **13. New assets**

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

804 Answer: [NA]

805 Justification: The paper does not release new assets.

806 Guidelines:

807 • The answer NA means that the paper does not release new assets.

808 • Researchers should communicate the details of the dataset/code/model as part of their
809 submissions via structured templates. This includes details about training, license,
810 limitations, etc.

811 • The paper should discuss whether and how consent was obtained from people whose
812 asset is used.

813 • At submission time, remember to anonymize your assets (if applicable). You can either
814 create an anonymized URL or include an anonymized zip file.

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

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

819 Answer: [NA]

820 Justification: The paper does not involve crowdsourcing nor research with human subjects.

821 Guidelines:

822 • The answer NA means that the paper does not involve crowdsourcing nor research with
823 human subjects.

824 • Including this information in the supplemental material is fine, but if the main contribu-
825 tion of the paper involves human subjects, then as much detail as possible should be
826 included in the main paper.

827 • According to the NeurIPS Code of Ethics, workers involved in data collection, curation,
828 or other labor should be paid at least the minimum wage in the country of the data
829 collector.

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

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

836 Answer: [NA]

837 Justification: The paper does not involve crowdsourcing nor research with human subjects.

838 Guidelines:

839 • The answer NA means that the paper does not involve crowdsourcing nor research with
840 human subjects.

841 • Depending on the country in which research is conducted, IRB approval (or equivalent)
842 may be required for any human subjects research. If you obtained IRB approval, you
843 should clearly state this in the paper.

844 • We recognize that the procedures for this may vary significantly between institutions
845 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
846 guidelines for their institution.

847 • For initial submissions, do not include any information that would break anonymity (if
848 applicable), such as the institution conducting the review.

849 **16. Declaration of LLM usage**

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

854 Answer: [Yes]

855 Justification: Exploring uncertainty decomposition in LLMs via in-context learning is a core
856 part of our methodology. We have delineated all details in our paper.

857 Guidelines:

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

Appendix

Contents

A Proofs	22
A.1 Variational Uncertainty Decomposition	22
A.2 Variational Estimates of Variance Decomposition	23
B Theoretical Examples	25
B.1 Bayesian linear regression	25
B.2 Gaussian process regression	25
C Sampling Methods for Auxiliary Data	26
C.1 Methods	26
C.2 Ablations on Logistic Regression Data	27
D Promoting Exchangeability in In-Context Learning	28
D.1 Enforcing Bayesian Behaviour and De Finetti’s	28
D.2 Permutation Invariant Conditional Generation	28
D.3 Determining threshold for KL-Filtering	29
E Algorithms and Pseudocode	29
E.1 Pseudocode for Variational Uncertainty Decomposition Algorithm	29
E.2 Computing Approximate Posterior Predictive Distributions	30
E.3 Code	31
F Experiments	31
F.1 Code Implementation	31
F.2 Synthetic Toy Experiments	32
F.3 Bandits	33
F.4 Out-of-Distribution Detection	43
G Further Related Works	43
H Example Prompts	45
H.1 Synthetic Toy Prompts	45
H.2 Bandit Prompts	45
H.3 OOD Detection Prompts	47

A Proofs

A.1 Variational Uncertainty Decomposition

Theorem 3.1 (Aleatoric Uncertainty Upper-Bound). *If the conditional independence relations in \mathcal{G} hold, then the variational estimator provides an upper-bound to the aleatoric uncertainty:*

$$V_a(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \geq U_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}), \quad (6)$$

895 where the difference between the variational estimator and the true aleatoric uncertainty is:

$$\mathbb{E}_{p(\mathbf{y}^*, \mathbf{U} | \mathbf{x}^*, \mathbf{Z}, \mathcal{D})} [D_{\text{KL}}[p(\theta | \mathbf{y}^*, \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}) || p(\theta | \mathbf{U}, \mathbf{Z}, \mathcal{D})]] . \quad (7)$$

896 *Proof.* We begin by decomposing the variational estimator V_a , noting that from \mathcal{G} we get,
897 $p(\mathbf{y}^* | \mathbf{x}^*, \theta) = p(\mathbf{y}^* | \mathbf{x}^*, \theta, \mathbf{U}, \mathbf{Z}, \mathcal{D})$ and $p(\theta | \mathbf{x}, \mathbf{U}, \mathbf{Z}, \mathcal{D}) = p(\theta, \mathbf{U}, \mathbf{Z}, \mathcal{D})$:

$$\begin{aligned} V_a(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) &:= -\mathbb{E}_{p(\mathbf{U} | \mathbf{Z}, \mathcal{D}) p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})} [\log p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})] \\ &= -\mathbb{E}_{p(\mathbf{U} | \mathbf{Z}, \mathcal{D}) p(\mathbf{y}^* | \mathbf{x}^*, \theta) p(\theta | \mathbf{U}, \mathbf{Z}, \mathcal{D})} \left[\log \frac{p(\mathbf{y}^* | \mathbf{x}^*, \theta) p(\theta | \mathbf{U}, \mathbf{Z}, \mathcal{D})}{p(\theta | \mathbf{y}^*, \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})} \right] \quad (*) \\ &= -\mathbb{E}_{p(\mathbf{U} | \mathbf{Z}, \mathcal{D}) p(\mathbf{y}^* | \mathbf{x}^*, \theta) p(\theta | \mathbf{U}, \mathbf{Z}, \mathcal{D})} [\log p(\mathbf{y}^* | \mathbf{x}^*, \theta)] \\ &\quad + \mathbb{E}_{p(\mathbf{U} | \mathbf{Z}, \mathcal{D}) p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}) p(\theta | \mathbf{y}^*, \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})} \left[\log \frac{p(\theta | \mathbf{y}^*, \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})}{p(\theta | \mathbf{U}, \mathbf{Z}, \mathcal{D})} \right] \quad (**) \\ &= \mathbb{E}_{p(\theta | \mathcal{D})} [\mathbb{H}[p(\mathbf{y}^* | \mathbf{x}^*, \theta)]] \\ &\quad + \mathbb{E}_{p(\mathbf{y}^*, \mathbf{U} | \mathbf{x}^*, \mathbf{Z}, \mathcal{D})} [D_{\text{KL}}[p(\theta | \mathbf{y}^*, \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}) || p(\theta | \mathbf{U}, \mathbf{Z}, \mathcal{D})]] \\ &\geq \mathbb{E}_{p(\theta | \mathcal{D})} [\mathbb{H}[p(\mathbf{y}^* | \mathbf{x}^*, \theta)]] := U_a(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}). \end{aligned}$$

898 Here steps (*) and (**) are obtained via Bayes' rule and the assumptions in \mathcal{G} . \square

899 *Alternative Proof.* Firstly, it is useful to define the corresponding definition of the variational approx-
900 imation to the epistemic uncertainty as:

$$\begin{aligned} V_e(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) &:= \mathbb{I}(\mathbf{y}^*; \mathbf{U} | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \\ &= \mathbb{H}[\mathbb{E}_{p(\mathbf{U} | \mathbf{Z}, \mathcal{D})} [p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})]] - V_a(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \\ &= \mathbb{H}[p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D})] - V_a(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \\ &= \mathbb{H}[p(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D})] - V_a(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}), \end{aligned} \quad (*)$$

901 where (*) follows from the conditional independence assumption $\mathbf{y}^* \perp \mathbf{Z} | \mathbf{x}, \mathcal{D}$. Therefore, we have

$$V_e(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) - U_e(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) = U_a(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) - V_a(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \quad (**)$$

902 If we have the conditional independence relation $\mathbf{y}^* \perp \mathbf{U} | \theta, \mathbf{x}, \mathbf{Z}, \mathcal{D}$, then by the data processing
903 inequality (DPE):

$$V_a(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) := \mathbb{I}(\mathbf{y}^*; \mathbf{U} | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \stackrel{\text{DPE}}{\leq} \mathbb{I}(\mathbf{y}^*; \theta | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \stackrel{(\dagger)}{=} \mathbb{I}(\mathbf{y}^*; \theta | \mathbf{x}^*, \mathcal{D}) =: U_a(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}),$$

904 where (\dagger) follows from the conditional independence relation $(\mathbf{y}^*, \theta) \perp \mathbf{Z} | \mathbf{x}, \mathcal{D}$. \square

905 **Remark.** From this information-theoretic perspective, we see that choosing an optimal \mathbf{Z} , is
906 equivalent to maximising the mutual information between \mathbf{y}^* and \mathbf{U} . This further motivates choosing
907 \mathbf{Z} that repeats \mathbf{x}^* or are perturbations of \mathbf{x}^* .

908 A.2 Variational Estimates of Variance Decomposition

909 To prove Theorem 3.2, we first prove the following lemma.

910 **Lemma A.1** For any random variables X, Y, Z where the conditional variances $\text{Var}(Y | X)$ and
911 $\text{Var}(Y | X, Z)$ exist,

$$\mathbb{E}[\text{Var}(Y | X)] = \mathbb{E}[\text{Var}(\mathbb{E}[Y | X, Z] | X)] + \mathbb{E}[\text{Var}(Y | X, Z)] \geq \mathbb{E}[\text{Var}(Y | X, Z)].$$

912 *Proof.* By the law of total expectation, $\mathbb{E}[\mathbb{E}(Y^2 | X)] = \mathbb{E}[\mathbb{E}(Y^2 | X, Z)] = \mathbb{E}[Y^2]$. Therefore,

$$\begin{aligned} \mathbb{E}[\text{Var}(Y | X)] - \mathbb{E}[\text{Var}(Y | X, Z)] &= \mathbb{E}[\mathbb{E}(Y^2 | X) - \mathbb{E}(Y | X)^2] - \mathbb{E}[\mathbb{E}(Y^2 | X, Z) - \mathbb{E}(Y | X, Z)^2] \\ &= \underbrace{\mathbb{E}[\mathbb{E}(Y^2 | X)] - \mathbb{E}[\mathbb{E}(Y^2 | X, Z)]}_{=0} - \mathbb{E}[\mathbb{E}(Y | X)^2] + \mathbb{E}[\mathbb{E}(Y | X, Z)^2] \\ &= \mathbb{E}[\mathbb{E}(Y | X, Z)^2] - \mathbb{E}[\mathbb{E}(Y | X)^2]. \end{aligned}$$

913 To show that the LHS is positive we first decompose $\mathbb{E}(Y|X, Z)$ as

$$\mathbb{E}(Y|X, Z) = (\mathbb{E}(Y|X, Z) - \mathbb{E}(Y|X)) + \mathbb{E}(Y|X).$$

914 Now, the expectation of the product of these terms is 0 as

$$\begin{aligned} \mathbb{E}[(\mathbb{E}(Y|X, Z) - \mathbb{E}(Y|X)) \cdot \mathbb{E}(Y|X)] &= \mathbb{E}[\mathbb{E}[(\mathbb{E}(Y|X, Z) - \mathbb{E}(Y|X)) \cdot \mathbb{E}(Y|X)|X]] \\ &= \mathbb{E}[\mathbb{E}[(\mathbb{E}(Y|X, Z) - \mathbb{E}(Y|X))|X] \cdot \mathbb{E}(Y|X)] \\ &= \mathbb{E}[(\mathbb{E}(Y|X) - \mathbb{E}(Y|X)) \cdot \mathbb{E}(Y|X)] \quad (*) \\ &= \mathbb{E}[0 \cdot \mathbb{E}(Y|X)] \\ &= 0, \end{aligned}$$

915 where $(*)$ follows from the fact that $\sigma(X) \subset \sigma(X, Z)$. Therefore,

$$\begin{aligned} \mathbb{E}[\mathbb{E}(Y|X, Z)^2] &= \mathbb{E}\left[\left((\mathbb{E}(Y|X, Z) - \mathbb{E}(Y|X)) + \mathbb{E}(Y|X)\right)^2\right] \\ &= \mathbb{E}\left[\underbrace{(\mathbb{E}(Y|X, Z) - \mathbb{E}(Y|X))^2}_{=\text{Var}(\mathbb{E}[Y|X, Z]|X)} + 2\underbrace{(\mathbb{E}(Y|X, Z) - \mathbb{E}(Y|X)) \cdot \mathbb{E}(Y|X)}_{=0} + \mathbb{E}(Y|X)^2\right] \\ &= \mathbb{E}[\text{Var}(\mathbb{E}[Y|X, Z]|X)] + \mathbb{E}[\mathbb{E}(Y|X)^2]. \end{aligned}$$

916 Finally, this gives

$$\mathbb{E}[\text{Var}(Y|X)] - \mathbb{E}[\text{Var}(Y|X, Z)] = \mathbb{E}[\mathbb{E}(Y|X, Z)^2] - \mathbb{E}[\mathbb{E}(Y|X)^2] = \mathbb{E}[\text{Var}(\mathbb{E}[Y|X, Z]|X)] \geq 0,$$

917 where the final inequality follows from the non-negativity of variance. \square

918 **Theorem 3.2** (Aleatoric Variance Upper-Bound). *If the conditional independence relation in \mathcal{G} holds,*
919 *then the variational estimator provides an upper-bound to the estimation of aleatoric variance:*

$$V_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) := \mathbb{E}_{p(\mathbf{U}|\mathbf{Z}, \mathcal{D})}[\text{Var}[\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}]] \geq U_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}). \quad (9)$$

920 *Proof.* By the definition of V_a^Σ ,

$$\begin{aligned} V_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) &= \mathbb{E}_{p(\mathbf{U}|\mathbf{Z}, \mathcal{D})}[\text{Var}[\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}]] \\ &= \mathbb{E}_{p(\mathbf{U}|\mathbf{x}^*, \mathbf{Z}, \mathcal{D})}[\text{Var}[\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}]] \\ &\geq \mathbb{E}_{p(\mathbf{U}, \theta|\mathbf{x}^*, \mathbf{Z}, \mathcal{D})}[\text{Var}[\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \theta, \mathcal{D}]] \quad (*) \\ &= \mathbb{E}_{p(\mathbf{U}, \theta|\mathbf{x}^*, \mathbf{Z}, \mathcal{D})}[\text{Var}[\mathbf{y}^*|\mathbf{x}^*, \theta]] \quad (**) \\ &= \mathbb{E}_{p(\theta|\mathbf{x}^*, \mathbf{Z}, \mathcal{D})}[\text{Var}[\mathbf{y}^*|\mathbf{x}^*, \theta]] \\ &= \mathbb{E}_{p(\theta|\mathcal{D})}[\text{Var}[\mathbf{y}^*|\mathbf{x}^*, \theta]] \\ &= U_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}). \end{aligned}$$

921 Here, $(*)$ follows from Lemma A.1 and $(**)$ follows from the conditional independence relation
922 $\mathbf{y}^* \perp \mathbf{Z}, \mathbf{U}, \mathcal{D}|\mathbf{x}^*, \theta$. \square

923 **Remark.** From Lemma A.1, we also obtain that the discrepancy between $V_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D})$ and
924 $U_a^\Sigma(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$ is

$$\begin{aligned} &\mathbb{E}\left[\text{Var}(\mathbb{E}[\mathbf{y}^*|\theta, \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}]|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})\right|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}] \\ &= \mathbb{E}_{p(\mathbf{U}|\mathbf{Z}, \mathcal{D})}\left[\text{Var}_{p(\theta|\mathbf{U}, \mathbf{Z}, \mathcal{D})}(\mathbb{E}[\mathbf{y}^*|\theta, \mathbf{x}^*]|\mathbf{U}, \mathbf{Z}, \mathcal{D})\right|\mathbf{Z}, \mathcal{D}]. \end{aligned}$$

925 B Theoretical Examples

926 B.1 Bayesian linear regression

927 Consider a linear regression model with homogeneous output noise variance. Namely, we assume
 928 a normal prior $p(\theta) = \mathcal{N}(\theta; \mathbf{0}, \lambda^{-1} \mathbf{I}_d)$, and the likelihood model is $p(\mathbf{y}|\mathbf{x}, \theta) := \mathcal{N}(\mathbf{y}; \theta^\top \mathbf{x}, \sigma^2)$.
 929 Denote $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times d}$ and $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_m]^\top \in \mathbb{R}^{m \times d}$. Now consider the exact
 930 posterior predictive distributions which can be shown as:

$$p(\theta|\mathcal{D}) = \mathcal{N}(\theta; \boldsymbol{\mu}, \Lambda^{-1}), \quad \Lambda := \sigma^{-2} \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_d, \quad \boldsymbol{\mu} := \Lambda^{-1} \mathbf{X}^\top \mathbf{y},$$

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}(\mathbf{y}^*; \boldsymbol{\mu}^\top \mathbf{x}^*, (\mathbf{x}^*)^\top \Lambda^{-1} \mathbf{x}^* + \sigma^2).$$

931 Then using the closed-form expressions for the entropy of a Gaussian distribution, it is straightforward
 932 to show that for arbitrary \mathbf{y}^* , \mathbf{x}^* and \mathcal{D} :

$$U_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \frac{1}{2}(1 + \log 2\pi\sigma^2),$$

$$U_e(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \frac{1}{2} \log((\mathbf{x}^*)^\top \Lambda^{-1} \mathbf{x}^* + \sigma^2) - \frac{1}{2} \log \sigma^2,$$

933 Adding the auxiliary data \mathbf{Z}, \mathbf{U} :

$$p(\theta|\mathbf{U}, \mathbf{Z}, \mathcal{D}) = \mathcal{N}(\theta; \boldsymbol{\mu}(\mathbf{Z}), \Lambda^{-1}(\mathbf{Z})), \quad \Lambda(\mathbf{Z}) := \sigma^{-2}(\mathbf{X}^\top \mathbf{X} + \mathbf{Z}^\top \mathbf{Z}) + \lambda \mathbf{I}_d,$$

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D}) = \mathcal{N}(\mathbf{y}^*; \boldsymbol{\mu}(\mathbf{Z})^\top \mathbf{x}^*, (\mathbf{x}^*)^\top \Lambda^{-1}(\mathbf{Z}) \mathbf{x}^* + \sigma^2 \mathbf{I}_d).$$

934 Since the variance of $p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathcal{D})$ does not depend on \mathbf{y}^* and \mathbf{U} , this leads to

$$V_a(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) = \frac{1}{2}(1 + \log 2\pi) + \frac{1}{2} \log((\mathbf{x}^*)^\top \Lambda^{-1}(\mathbf{Z}) \mathbf{x}^* + \sigma^2),$$

$$V_e(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \frac{1}{2} \log((\mathbf{x}^*)^\top \Lambda^{-1} \mathbf{x}^* + \sigma^2) - \frac{1}{2} \log((\mathbf{x}^*)^\top \Lambda^{-1}(\mathbf{Z}) \mathbf{x}^* + \sigma^2),$$

935 It is easy to show for all possible \mathbf{Z} :

$$V_a(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) - U_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \frac{1}{2} \log(\sigma^{-2}(\mathbf{x}^*)^\top \Lambda^{-1}(\mathbf{Z}) \mathbf{x}^* + 1) \geq 0.$$

936 Now consider the optimum of the variational estimate:

$$V_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) := \frac{1}{2}(1 + \log 2\pi\sigma) + \min_{\mathbf{Z}} \frac{1}{2} \log(\sigma^{-2}(\mathbf{x}^*)^\top \Lambda^{-1}(\mathbf{Z}) \mathbf{x}^* + 1),$$

937 where $\Lambda(\mathbf{Z}) := \sigma^{-2}(\mathbf{X}^\top \mathbf{X} + \mathbf{Z}^\top \mathbf{Z}) + \lambda \mathbf{I}_d$. Now, if γ is the minimum eigenvalue of $(\mathbf{X}^\top \mathbf{X} + \mathbf{Z}^\top \mathbf{Z})$
 938 and $\gamma > 0$, then $(\mathbf{x}^*)^\top \Lambda^{-1} \mathbf{x}^* \leq \frac{1}{\gamma} \|\mathbf{x}^*\|_2^2$. If $m \geq d$, we can choose \mathbf{z}_j (e.g. unit vectors) such that
 939 $\lambda > 0$, and then scaling \mathbf{z}_j by a constant ensures $\gamma \rightarrow \infty$ and $(\mathbf{x}^*)^\top \Lambda^{-1} \mathbf{x}^* \rightarrow 0$. Therefore, for
 940 appropriately chosen \mathbf{Z} , $V_a(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Z}, \mathcal{D}) \rightarrow U_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$.

941 B.2 Gaussian process regression

942 Here we assume a Gaussian process model [52] with a kernel function as the prior covariance:

$$y = f(\mathbf{x}) + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1), \quad f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot)).$$

943 For regression problems we have closed form solution to the posterior predictive:

$$p(y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(y^*; \boldsymbol{\mu}, k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{X}}(\mathbf{K}_{\mathbf{X}\mathbf{X}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X}*} + \sigma^2),$$

944 leading to the following uncertainty estimates (with $\mathcal{D} = (\mathbf{X}, \mathbf{y})$):

$$U_a(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \frac{1}{2}(1 + \log 2\pi\sigma^2),$$

$$U_e(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \frac{1}{2} \log(k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{X}}(\mathbf{K}_{\mathbf{X}\mathbf{X}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X}*} + \sigma^2) - \frac{1}{2} \log \sigma^2.$$

Now consider sparse variational Gaussian process (SVGP) [20] with inducing inputs/outputs \mathbf{Z}, \mathbf{u} and an approximating distribution $q(\mathbf{u}) := \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S})$. Then we have the approximate posterior predictive as:

$$q(y^* | \mathbf{x}^*) = \mathcal{N}(y^*; \mu(\mathbf{x}^*), k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{Z}} \mathbf{K}_{\mathbf{ZZ}}^{-1} (\mathbf{K}_{\mathbf{ZZ}} - \mathbf{S}) \mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbf{K}_{\mathbf{Z}*} + \sigma^2),$$

so that the uncertainty estimates are

$$U_a(\mathbf{y}^* | \mathbf{x}^*; q) = \frac{1}{2} (1 + \log 2\pi\sigma^2),$$

$$U_e(\mathbf{y}^* | \mathbf{x}^*; q) = \frac{1}{2} \log(k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{Z}} \mathbf{K}_{\mathbf{ZZ}}^{-1} (\mathbf{K}_{\mathbf{ZZ}} - \mathbf{S}) \mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbf{K}_{\mathbf{Z}*} + \sigma^2) - \frac{1}{2} \log \sigma^2.$$

For regression problems we have the optimal $\mathbf{S} = \mathbf{K}_{\mathbf{ZZ}} (\mathbf{K}_{\mathbf{ZZ}} + \sigma^{-2} \mathbf{K}_{\mathbf{ZX}} \mathbf{K}_{\mathbf{XZ}})^{-1} \mathbf{K}_{\mathbf{ZZ}}$ [20], and therefore

$$U_e(\mathbf{y}^* | \mathbf{x}^*; q) = \frac{1}{2} \log(k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{Z}} (\mathbf{K}_{\mathbf{ZZ}}^{-1} - (\mathbf{K}_{\mathbf{ZZ}} + \sigma^{-2} \mathbf{K}_{\mathbf{ZX}} \mathbf{K}_{\mathbf{XZ}})^{-1}) \mathbf{K}_{\mathbf{Z}*} + \sigma^2) - \frac{1}{2} \log \sigma^2.$$

On the other hand, using the variational uncertainty decomposition method, we can show that

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{U}, \mathbf{Z}, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mathbf{y}^*; \mu(\mathbf{Z}, \mathbf{U}), k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{X}} (\mathbf{K}_{\mathbf{XX}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X}*} - \Delta(\mathbf{x}^*, \mathbf{Z}) + \sigma^2),$$

$$\Delta(\mathbf{x}^*, \mathbf{Z}) = \mathbf{A}^\top (\mathbf{K}_{\mathbf{ZZ}} + \sigma^2 \mathbf{I} - \mathbf{K}_{\mathbf{ZX}} (\mathbf{K}_{\mathbf{XX}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{XZ}})^{-1} \mathbf{A},$$

$$\mathbf{A} = \mathbf{K}_{\mathbf{Z}*} - \mathbf{K}_{\mathbf{ZX}} (\mathbf{K}_{\mathbf{XX}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X}*},$$

leading to the following uncertainty estimates (with $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ and $C := \frac{1}{2} (1 + \log 2\pi)$):

$$V_a(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) = C + \frac{1}{2} \log(k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{X}} (\mathbf{K}_{\mathbf{XX}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X}*} - \Delta(\mathbf{x}^*, \mathbf{Z}) + \sigma^2),$$

$$V_e(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}) = C + \frac{1}{2} \log(k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{X}} (\mathbf{K}_{\mathbf{XX}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X}*} + \sigma^2) - V_a(\mathbf{y}^* | \mathbf{x}^*, \mathbf{Z}, \mathcal{D}).$$

Note that if we choose $\mathbf{Z} = \mathbf{x}^*$ then we have

$$V_a(\mathbf{y}^* | \mathbf{x}^*, \mathbf{x}^*, \mathcal{D}) = U_a(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) + \frac{1}{2} \log \left(2 - \frac{\sigma^2}{k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{X}} (\mathbf{K}_{\mathbf{XX}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X}*} + \sigma^2} \right),$$

$$V_e(\mathbf{y}^* | \mathbf{x}^*, \mathbf{x}^*, \mathcal{D}) = U_e(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) - \frac{1}{2} \log \left(2 - \frac{\sigma^2}{k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{X}} (\mathbf{K}_{\mathbf{XX}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X}*} + \sigma^2} \right).$$

This means if test data \mathbf{x}^* is close to the training data \mathbf{X} , then $k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}_{*\mathbf{X}} (\mathbf{K}_{\mathbf{XX}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X}*}$ will be close to zero, and then $V_e(\mathbf{y}^* | \mathbf{x}^*, \mathbf{x}^*, \mathcal{D}) \approx U_e(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D})$ provides a good estimate of the epistemic uncertainty.

C Sampling Methods for Auxiliary Data

In this section, we discuss in detail the methods used to sample auxiliary data \mathbf{Z} to find the best variational estimate of the aleatoric uncertainty and variance. As noted in Section 3.1, we restrict \mathbf{Z} to a single example in the \mathbf{x} domain to reduce the search space.

C.1 Methods

Bayesian Optimisation. The optimisation problem (8) can be directly optimised via Bayesian Optimisation. However, this is a constrained optimisation problem where \mathbf{Z} needs to satisfy an "approximately Bayesian" criterion (11) which we discuss in Section 3.2. To overcome this issue, we treat the problem as an unconstrained Bayesian optimisation to obtain auxiliary examples $\{z_i\}_{i=1}^m$ and then apply the criterion to remove auxiliary examples that do not satisfy (11).

In the synthetic examples we consider, the covariates \mathbf{x}_i are real and continuous. Therefore, we use a Gaussian process with an RBF kernel to model the objective function and take the log expected improvement as the acquisition function. In order to provide a warm start to the Bayesian optimisation process, we provide 5 initial samples that are randomly sampled.

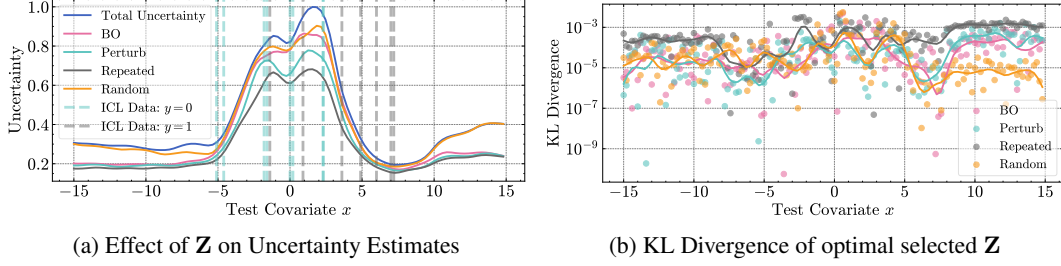


Figure 10: Comparing the computed V_a across different methods that sample \mathbf{Z} for the Qwen2.5-7B model.

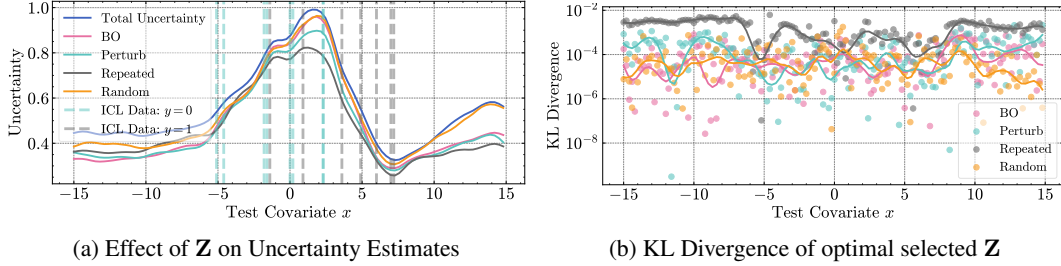


Figure 11: Comparing the computed V_a across different methods that sample \mathbf{Z} for the Llama3-8B model.

Table 3: V_a rank for different sampling methods

MODELS	BAYESIAN OPTIMISATION	PERTURBATIONS	REPEATED TASK	RANDOM SAMPLING
QWEN7B	2.93	2.01	1.29	3.77
QWEN14B	3.09	2.03	1.16	3.68
LLAMA8B	2.41	1.92	2.09	3.57

Perturbations. Given the covariates \mathbf{x}^* of the data point we wish to decompose the uncertainty for, we can choose $\mathbf{Z} = \{\mathbf{z}_j\}_{j=1}^m$ to be “close” to \mathbf{x}^* . To perturb a categorical covariate x_k^* , we sample uniformly from the list of categories with probability p and keep the original covariate with probability $1 - p$. For a real covariate x_k^* , we sample from a normal distribution, similarly to random sampling, but we choose the mean as x_k^* and the standard deviation as a scaled population standard deviation estimate of the covariate $\gamma \cdot \sigma_k^{\mathcal{D}}$ where $\gamma = 0.1$.

Repeated. Given the test covariates \mathbf{x}^* , we set $\mathbf{Z} = \mathbf{x}^*$. Since we repeat the covariates, we only have 1 auxiliary example per test example.

Random Sampling. The most basic sampling procedure to generate auxiliary data \mathbf{Z} is to randomly sample the data. If a covariate is a categorical variable, we sample uniformly from the list of categories. If a covariate is a real variable, we assume a normal distribution with mean and standard deviation given by the population mean and standard deviation estimates of the covariate, $\mu_k^{\mathcal{D}}$ and $\sigma_k^{\mathcal{D}}$ from the in-context data \mathcal{D} .

C.2 Ablations on Logistic Regression Data

We compare the performance of the four approaches outlined in Section C.1 for choosing \mathbf{Z} for 15 auxiliary examples (with the exception of the Repeated where we have a single auxiliary example). We plot the uncertainty decompositions for the \mathbf{Z} sampling approaches and the corresponding KL divergence for the chosen \mathbf{Z} that minimises (8) in Figures 8 10 and 11. In Tables 3 and 4 we quantify the performance of each of the sampling methods by computing the mean rank of each method over the test samples. For the 3 LLMs that we consider, we consistently observe that Repeated has the lowest V_a , followed by Perturb. However, Perturb has the highest KL divergence which indicates that this method is less aligned with the Bayesian assumptions that we make.

Table 4: KL-divergence rank for different sampling methods

MODELS	BAYESIAN OPTIMISATION	PERTURBATIONS	REPEATED TASK	RANDOM SAMPLING
QWEN7B	2.27	2.31	3.43	1.99
QWEN14B	1.98	2.27	3.51	3.51
LLAMA8B	1.93	2.41	3.77	1.89

D Promoting Exchangeability in In-Context Learning

D.1 Enforcing Bayesian Behaviour and De Finetti’s

In order to apply de Finetti’s theorem in (2) to a sequence of random variables $\{(x_i, y_i)\}_{i \geq 1}$, we need to ensure the exchangeability of conditional distribution as in (1). This is challenging as the position of tokens in a prompt can affect the prediction of the next token due to aspects of the transformer architecture such as positional encoding and rotational embeddings [73, 36, 15]. In order to promote exchangeability during autoregressive generation we employ *permutation-invariant conditional generation*.

D.2 Permutation Invariant Conditional Generation

Consider the following property of exchangeable sequences:

Proposition D.1 *If a sequence of random variables $(X_n)_{n \in \mathbb{N}^+}$ are exchangeable, then given a permutation $\sigma : [n] \rightarrow [n]$,*

$$p(X_{n+1} = x_{n+1} | X_{\sigma(1)} = x_1, \dots, X_{\sigma(n)} = x_n) = p(X_{n+1} = x_{n+1} | X_1 = x_1, \dots, X_n = x_n). \quad (12)$$

Proof.

$$\begin{aligned} p(X_{n+1} = x_{n+1} | X_{\sigma(1)} = x_1, \dots, X_{\sigma(n)} = x_n) &= \frac{p(X_{\sigma(1)} = x_1, \dots, X_{\sigma(n)} = x_n, X_{n+1} = x_{n+1})}{p(X_{\sigma(1)} = x_1, \dots, X_{\sigma(n)} = x_n)} \\ &= \frac{p(X_1 = x_1, \dots, X_n = x_n, X_{n+1} = x_{n+1})}{p(X_1 = x_1, \dots, X_n = x_n)} \\ &= p(X_{n+1} = x_{n+1} | X_1 = x_1, \dots, X_n = x_n). \end{aligned}$$

□

Proposition D.1 shows that the distribution of the next term in the sequence is not dependent on the ordering of the previous terms. However, due to architectural choices in LLMs such as positional encoding and rotational embeddings, the order of the in-context examples affects the posterior predictive distribution of the model. Therefore, we permute the order of the in-context examples to add this inductive bias, which we name *permutation invariant conditional generation*.

Suppose we have in-context examples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where x_i are the covariates for data point i and y_i is the corresponding label. Then, a prompt with *ordered* in-context examples $((x_i, y_i))_{i=1}^n$ to predict the label y^* for x^* is of the form

```

1014 {x_1} <output>{y_2}</output>\n
1015 {x_2} <output>{y_3}</output>\n
1016 .
1017 .
1018 .
1019 {x_n} <output>{y_n}</output>\n
1020 {x^*} <output>
```

We denote the probability distribution of y^* extracted by taking the logits of the label y^* as

$$p(y^* | x^*, ((x_i, y_i))_{i=1}^n). \quad (13)$$

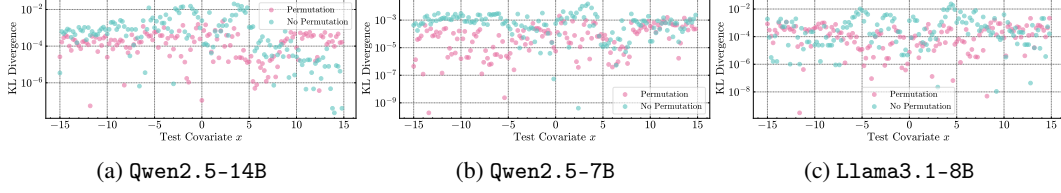


Figure 12: Permutation Ablation for Logistic Regression Dataset.

Clearly, this is dependent on the ordering $((\mathbf{x}_i, \mathbf{y}_i))_{i=1}^n$. However, we can consider the distribution \tilde{p} defined as

$$\tilde{p}(\mathbf{y}^* | \mathbf{x}^*, \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n) = \mathbb{E}_{\sigma} \left[p(\mathbf{y}^* | \mathbf{x}^*, ((\mathbf{x}_{\sigma(i)}, \mathbf{y}_{\sigma(i)})_{i=1}^n)) \right], \quad (14)$$

which is the distribution obtained by taking the expectation over the uniform distribution over all permutations $\sigma : [n] \mapsto [n]$ of the n in-context examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$. Note that now the probability \tilde{p} does not depend on the order of the in-context examples.

This allows us to define a probability distribution over the sequence $X_i = (\mathbf{x}_i, \mathbf{y}_i)$. Assume \mathbf{x}_{n+1} is independent of $\mathbf{X}_{1:n}$, with density

$$\hat{p}(\mathbf{x}_{n+1} | X_1, \dots, X_n) = p_x(\mathbf{x}_{n+1}),$$

for some density p_x . Then \mathbf{y}_{n+1} given \mathbf{x}_{n+1} and $X_{1:n}$ has density,

$$\hat{p}(\mathbf{y}_{n+1} | \mathbf{x}_{n+1}, X_{1:n}) = \tilde{p}(\mathbf{y}_{n+1} | \mathbf{x}_{n+1}, \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n).$$

This gives conditional density

$$\hat{p}(X_{n+1} | X_1, \dots, X_n) = p(\mathbf{y}_{n+1} | \mathbf{x}_{n+1}, X_{1:n}) p(\mathbf{x}_{n+1} | X_1, \dots, X_n) \quad (15)$$

$$= \tilde{p}(\mathbf{y}_{n+1} | \mathbf{x}_{n+1}, \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n) p_x(\mathbf{x}_{n+1}) \quad (16)$$

By construction for any permutation $\sigma : [n] \rightarrow [n]$,

$$\hat{p}(X_{n+1} | X_1, \dots, X_n) = \hat{p}(X_{n+1} | X_{\sigma(1)}, \dots, X_{\sigma(n)}).$$

Therefore, sampling from the LLM by perturbing the inputs guarantees permutation invariant conditional generation. We obtain Monte Carlo estimates to (14) when computing posterior predictive distributions, which we discuss in Appendix E.2.

Effect of no permutation. In Figure 12, we plot the KL divergent of the uncertainty decomposition when we permute and not permute the in-context labels. We see that permuting the in-context labels results in lower KL divergences which suggests the behaviour is more Bayesian.

D.3 Determining threshold for KL-Filtering

The choice of ϵ controls the level of approximation permitted in the uncertainty decomposition method. A small ϵ ensures that the auxiliary data \mathbf{Z} that we choose obey our Bayesian assumption but at the cost of rejecting more \mathbf{Z} and obtaining a larger variational upper bound to the aleatoric uncertainty or variance. Furthermore, as shown in Figure 8, the range of KL values for the different auxiliary examples may vary when we vary \mathbf{x}^* . Therefore, to guarantee that we have enough valid auxiliary examples, we set ϵ as the r^{th} smallest element in the set of KL divergences $\{\epsilon_j\}_{j=1}^m$ where $\epsilon_j := D_{KL}[\tilde{p}(\mathbf{y} | \mathbf{x}, \mathcal{D}), \tilde{p}(\mathbf{y} | \mathbf{x}, \mathcal{D}, \mathbf{z}_j)]$. Therefore, we can control the strictness of the filtering by varying r , where a smaller r gives a stricter decomposition.

E Algorithms and Pseudocode

E.1 Pseudocode for Variational Uncertainty Decomposition Algorithm

Algorithm 1 is pseudocode for multi-class classification problems and Algorithm 2 is the pseudocode for regression. They are similar in approach but vary during the marginalisation step: for classification,

1051 we can compute $\tilde{p}(\mathbf{y}|\mathbf{x}, \mathbf{u} = k, \mathbf{z}_j, \mathcal{D})$ for each class k , and directly compute the marginal distribution
 1052 using the tower property. However, for regression, this is computationally infeasible so we use a
 1053 Monte Carlo estimate for the entropy $\mathbb{E}_{\mathbf{u} \sim \tilde{p}(\mathbf{u}|\mathbf{z}_j, \mathcal{D})}[H[\tilde{p}(\mathbf{y}|\mathbf{x}, \mathbf{u}, \mathbf{z}_j, \mathcal{D})]]$, over different samples
 1054 of \mathbf{u} . To obtain the marginal distribution, we bootstrap samples from the mixture of Gaussians
 1055 $\{\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, \mathbf{u}_t^{(j)}\})\}_{t=1}^T$ and fit a Gaussian to these samples.

Algorithm 1 Multi-Class Classification for Aleatoric Uncertainty Estimation

Require: Features \mathbf{x} ; ICL Dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ where $\mathbf{y}_i \in [K]$

- 1: $\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D}) \leftarrow \text{CLASSDIST}(\mathbf{x}, \mathcal{D})$
- 2: $H_{y|\mathbf{x}} \leftarrow H[\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D})]$
- 3: **for** $j = 1, \dots, m$ **do**
- 4: $\mathbf{z}_j \leftarrow \text{NEWAUX}(\mathbf{x}, \mathbf{z}_{[1:j-1]})$ {Get new auxiliary variable}
- 5: $\tilde{p}(\mathbf{u}|\mathbf{z}_j, \mathcal{D}) \leftarrow \text{CLASSDIST}(\mathbf{z}_j, \mathcal{D})$
- 6: **for** $k = 1, \dots, K$ **do**
- 7: $\tilde{p}_j(\mathbf{y}|\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, k\}) \leftarrow \text{CLASSDIST}(\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, k\})$
- 8: $H_{kt} \leftarrow H[\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, k\})]$
- 9: **end for**
- 10: $\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathbf{z}_j) \leftarrow \sum_{k=1}^K \tilde{p}_j(\mathbf{y}|\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, k\}) \cdot \tilde{p}(\mathbf{u} = k|\mathbf{z}_j, \mathcal{D})$
- 11: $H_j \leftarrow \sum_{k=1}^K H_{kt} \cdot \tilde{p}(\mathbf{u} = k|\mathbf{z}_j, \mathcal{D})$
- 12: $\epsilon_j \leftarrow D_{\text{KL}}[\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D}) \parallel \tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathbf{z}_j)]$
- 13: **end for**
- 14: Compute threshold ϵ
- 15: $V_a \leftarrow \min(\min(\{H_j : \epsilon_j < \epsilon\}), H_{y|\mathbf{x}})$
- 16: **return** V_a

Algorithm 2 Regression for Aleatoric Uncertainty Estimation

Require: Features \mathbf{x} ; ICL Dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ where $\mathbf{y}_i \in \mathbb{R}$

- 1: $\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D}) \leftarrow \text{REGDIST}(\mathbf{x}, \mathcal{D})$
- 2: $H_{y|\mathbf{x}} \leftarrow H[\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D})]$
- 3: **for** $j = 1, \dots, m$ **do**
- 4: $\mathbf{z}_j \leftarrow \text{NEWAUX}(\mathbf{x}, \mathbf{z}_{[1:j-1]})$ {Get new auxiliary variable}
- 5: $U^{(j)} \leftarrow \{\mathbf{u}_t^{(j)}\}_{t=1}^T$ where $\mathbf{u}_t^{(j)} \sim \text{REGDIST}(\mathbf{z}_j, \mathcal{D})$
- 6: **for** $t = 1, \dots, T$ **do**
- 7: $\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, \mathbf{u}_t^{(j)}\}) \leftarrow \text{REGDIST}(\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, \mathbf{u}_t^{(j)}\})$
- 8: $H_{jt} \leftarrow H[\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, \mathbf{u}_t^{(j)}\})]$
- 9: **end for**
- 10: $\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathbf{z}_j) \leftarrow \text{NORMAPPROX}(\{\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, \mathbf{u}_t^{(j)}\})\}_{t=1}^T)$
- 11: $H_j \leftarrow \frac{1}{T} \sum_t H_{jt}$
- 12: $\epsilon_j \leftarrow D_{\text{KL}}[\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D}) \parallel \tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathbf{z}_j)]$
- 13: **end for**
- 14: Compute threshold ϵ
- 15: $V_a \leftarrow \min(\min(\{H_j : \epsilon_j < \epsilon\}), H_{y|\mathbf{x}})$
- 16: **return** V_a

1056 Note that these algorithms can also be extended to the decomposition of total variance by replacing
 1057 the entropic uncertainty terms with the corresponding variance terms.

1058 E.2 Computing Approximate Posterior Predictive Distributions

1059 **Classification.** Algorithm 3 describes the process of obtaining the logits for a predictive task
 1060 $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ given in-context learning data $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ and the covariates of the predictive task \mathbf{x} .
 1061 We permute the ICL data and take an average of the predictive distribution to obtain a Monte Carlo
 1062 estimate of a conditional permutation-invariant distribution (which we discuss further in Appendix

1063 **D.** Furthermore, by the construction of the prompt, the we only need to obtain the logits for the first
 1064 token that is generated, which remains constant with respect to the choice of LLM seed.

Algorithm 3 Compute Permutation Invariant Classification Distribution z : CLASSDIST

Require: Features \mathbf{x} ; ICL Dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ where $\mathbf{y}_i \in [K]$
 1: **function** CLASSDIST(\mathbf{x}, \mathcal{D})
 2: **for** $l = 1, \dots, L$ **do**
 3: $\sigma_l \sim S_K$
 4: $\mathbf{p}_y^{(l)} \leftarrow \text{LLM}(\text{PROMPT}(\mathbf{x}_{\sigma_l(1)}, \mathbf{y}_{\sigma_l(1)}, \dots, \mathbf{x}_{\sigma_l(K)}, \mathbf{y}_{\sigma_l(K)}, \mathbf{x}))$ {Class logits of next token \mathbf{y} }
 5: **end for**
 6: $\bar{\mathbf{p}}_y \leftarrow \frac{1}{L} \sum_l \mathbf{p}^{(l)}$
 7: **return** $\bar{\mathbf{p}}_y$

1065 **Regression.** In Algorithm 4, we outline the procedure for constructing an approximate distribution
 1066 for $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$. Similarly to the classification case, we permute the ICL data. However, as \mathbf{y} can take
 1067 any value in \mathbb{R} , the tokenisation of \mathbf{y} may require more than one token and as the logits of a token
 1068 depend on the previous tokens generated, the logits of the tokens will vary with the choice of LLM
 1069 seed. Standard approaches to approximate the distribution require a forward pass over every value
 1070 that \mathbf{y} takes [54] which is prohibitively expensive. Therefore, for each permutation, we sample a
 1071 single \mathbf{y} (varying the LLM seed for every permutation) and fit a normal distribution to these samples.

1072 **Variance Reduction.** To reduce the variance of the estimated mean and standard deviation, we use
 1073 a trimmed mean, removing the top k and bottom k of our samples, and the interquartile range to
 1074 estimate the mean and standard deviation respectively [62]. In our experiments, we set $k = 1$.

1075 **Marginalisation.** In Algorithm 2, we are required to compute the marginal distribution $\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathbf{z}_j)$
 1076 given the Gaussian distributions $\{\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, \mathbf{u}_t^{(j)}\})\}_{t=1}^T$. We compute this marginal distribution
 1077 by bootstrap sampling from the distributions $\tilde{p}(\mathbf{y}|\mathbf{x}, \mathcal{D} \cup \{\mathbf{z}_j, \mathbf{u}_t^{(j)}\})$ and fitting a Gaussian distribution
 1078 to the bootstrap samples.

Algorithm 4 Approximate Permutation Invariant Regression Distribution: REGDIST. Optionally, at
 line 6, can remove the top k and bottom k values to obtain a more robust measure of the mean and
 variance.

Require: Features \mathbf{x} ; ICL Dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ where $\mathbf{y}_i \in \mathbb{R}$
 1: **function** REGDIST(\mathbf{x}, \mathcal{D})
 2: **for** $l = 1, \dots, L$ **do**
 3: $\sigma_l \sim S_K$
 4: $\mathbf{y}^{(l)} \leftarrow \text{LLM}(\text{PROMPT}(\mathbf{x}_{\sigma_l(1)}, \mathbf{y}_{\sigma_l(1)}, \dots, \mathbf{x}_{\sigma_l(K)}, \mathbf{y}_{\sigma_l(K)}, \mathbf{x}))$ {Sample next label \mathbf{y} }
 5: **end for**
 6: $\mathbf{Y} \leftarrow \{\mathbf{y}^{(l)}\}_{l=1}^L$
 7: **return** Normal(mean(\mathbf{Y}), std(\mathbf{Y}))

1079 E.3 Code

1080 We provide code for all experiments and algorithms in the following github repository:
 1081 <https://anonymous.4open.science/r/VUD/README.md>

1082 F Experiments

1083 F.1 Code Implementation

1084 The following delineates the foundation of our experiments:

- 1085 • Codebase: Python & PyTorch
- 1086 • CPU: AMD EPYC 7443P
- 1087 • GPU: NVIDIA A6000 48GB

1088 We leverage Qwen-2.5-14B/14B-Instruct/7B [51] and Llama3.1-8B [60] in our experiments. The
 1089 following delineates the configurations of our LLM.

- 1090 • Temperature: 1.0
- 1091 • Log Probs: 10
- 1092 • Max Tokens: 10 (Qwen-2.5-14B/7B and Llama3.1-8B), 512 (Qwen-2.5-14B-Instruct)

1093 F.2 Synthetic Toy Experiments

1094 **Datasets.** We qualitatively evaluate the decompositions of the variational uncertainty decomposition
 1095 algorithm on a variety of synthetic classification and regression settings. In this section, we give
 1096 details on the ground-truth distributions used to create the synthetic datasets.

1097 **Logistic Regression.** We consider a 1-D logistic regression problem with coefficient $\beta = 0.25$
 1098 and bias $\beta_0 = -0.5$. The covariates are generated from a Gaussian distribution with mean 1.5 and
 1099 standard deviation 3.

1100 **Linear Regression.** We consider a 1-D linear regression problem with coefficient $\beta = -1$, bias
 1101 $\beta_0 = 3$ and Gaussian noise with zero mean and standard deviation $\sigma = 2$. The covariates are
 1102 generated from a Gaussian distribution with mean 1 and standard deviation 2.

1103 **Heteroscedastic "Gaps" Regression.** We model the "gaps" as the combination of 3 linear regression
 1104 datasets, constructed similarly to the prior linear regression datasets. The parameters of the 3 clusters
 1105 are given in Table 5. To generate the small in-context learning dataset, we sample from this combined
 1106 dataset.

Table 5: Heteroscedastics "Gaps" Dataset Parameters

CLUSTER	DATASET SIZE	COEFFICIENT	BIAS	NOISE	$\mathbb{E}[x]$	$\text{Var}[x]$
1	50	0.75	1.0	0.1	-7	0.75
2	50	0.75	1.0	0.1	-1	0.75
3	100	0	-0.5	2	5	1

1107 **Moons Dataset.** We use the `make_moons` two-moons dataset generator in the `scikit-learn`
 1108 package (<https://scikit-learn.org/stable/>) We set the noise parameter in the "Moons 1" and "Moons 2"
 1109 datasets to $\sigma = 0.1$ and $\sigma = 0.4$ respectively. Figure 1 in the main text shows the decomposition for
 1110 "Moons 1" dataset.

1111 **Spirals Dataset.** We use a n -arm spiral dataset generator ([https://github.com/corneauf/N-Arm-Spiral-](https://github.com/corneauf/N-Arm-Spiral-Dataset)
 1112 `Dataset`) to generate the spirals. We set the number of arms to 3 and noise to be 1.2. We also scale the
 1113 covariate down by a factor of 4 so that all the points would appear in $[-4, 4] \times [-4, 4]$. For further
 1114 details see the codebase in Appendix E.1.

1115 F.2.1 Experimental Details for Visualisations

1116 **Logistic Regression.** We use `Perturb` with 15 auxiliary data points and perturbation scale $\lambda = 0.1$
 1117 to decompose the uncertainty for the logistic regression task. In Figures 4a and 13, we plot the
 1118 uncertainty decomposition for an ICL dataset of size $|\mathcal{D}| = 15$ and in Figures 14, we plot the
 1119 decomposition for $|\mathcal{D}| = 75$. We plot x^* values in the range $[-15, 15]$ with step-size 0.2. In Figures
 1120 7, 15 and 16, we plot the epistemic and aleatoric uncertainties as the dataset size increases for
 1121 in-distribution ($x = 0, 5$; solid lines) and out-of-distribution ($x^* = -15, -10, -5, 10, 15$; dotted
 1122 lines) points. As the uncertainty at a given x^* is dependent on the particular dataset, we average the
 1123 uncertainty at x^* over 10 datasets of the same size d to obtain the estimate of the mean aleatoric
 1124 uncertainty at d .

1125 **Linear Regression.** We use `Perturb` with 5 auxiliary data points and perturbation scale $\lambda = 0.1$ to
 1126 decompose the uncertainty for the logistic regression task. We reduce the number of auxiliary data
 1127 points due to the increased computational cost of computing distributions for regression problems.
 1128 In order to obtain smoother uncertainty computations, we average the uncertainties obtained over 3
 1129 sampled datasets of size $|\mathcal{D}| = 15$. We compute uncertainties for x^* in range $[-15, 15]$ with step-size

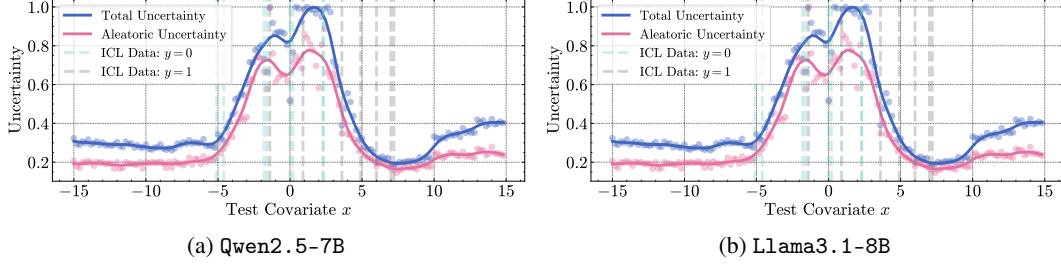


Figure 13: Uncertainty Decomposition for Logistic Regression $|\mathcal{D}| = 15$.

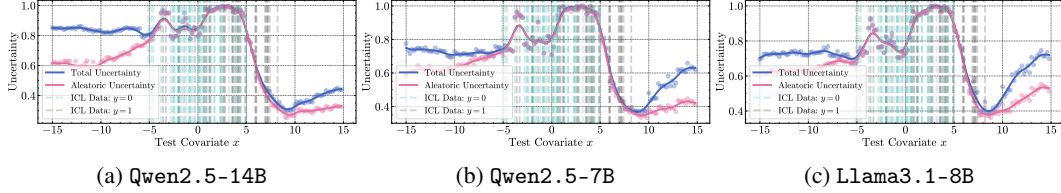


Figure 14: Logistic Regression with $|\mathcal{D}| = 75$.

0.2 and plot the obtained decompositions for entropic uncertainty and variance in Figure 4b, 17 and 18. We also provide an example decomposition for the uncertainty and variance for a single seed for completion in Figures 19, 20 and 21.

Heteroscedastic "Gaps" Dataset. We use Perturb with 5 auxiliary data points and perturbation scale $\lambda = 0.1$. We sample a single dataset of size $|\mathcal{D}| = 30$. We compute uncertainties for \mathbf{x}^* in range $[-15, 15]$ with step-size 0.2 and plot the obtained decompositions in Figures 22, 23 and 24.

Moons Dataset. We use Perturb with 15 auxiliary data points and perturbation scale $\lambda = 0.1$. For the "Moons 1" dataset, we sample a single dataset of size $|\mathcal{D}| = 30$ and compute uncertainties for \mathbf{x}^* in range $[-1.5, 2.5) \times [-1.5, 2.5)$ with step-size 0.2 for each interval. The decompositions are given in Figures 1, 25 and 26. For the "Moons 2" dataset, we sample a single dataset of size $|\mathcal{D}| = 30$ and compute uncertainties for \mathbf{x}^* in range $[-3.0, 3.5) \times [-2.5, 3.0)$ with step-size 0.2 for each interval. The decompositions are given in Figures 27, 28 and 29.

Spirals Dataset. Due to the complexity of this task, we sample a dataset of size $|\mathcal{D}| = 200$ and we compute uncertainties for \mathbf{x}^* in the range of $[-4, 4) \times [-4, 4)$ with interval 0.1. To mitigate the cost of increases prompt size and the number of test data points, we use Repeated to obtain \mathbf{Z} . The decomposition for Qwen2.5-14B is given in Figure 6. We provide decompositions for Qwen2.5-7B and Llama3.1-8B are shown in Figure 30 and 31.

F.3 Bandits

Definitions. In a bandit problem, we have multiple trials (or equivalently rounds), where the agent must choose an action (or equivalently an arm) which gives a reward. The agent has access to the actions made and rewards obtained for the previous trials. We denote run or seed to refer to a particular chain of trials.

LLM-UCB Algorithm. In the UCB algorithm, we have:

$$a_t = \operatorname{argmax}_{a \in \mathcal{A}} \{Q_t(a) + \alpha U_t(a)\},$$

where $Q_t(a)$ is the expected reward from action (i.e. arm) at t , $U_t(a)$ is the uncertainty in the reward from action a at t and α is the exploration rate [30]. In the LLM-UCB algorithm that we use to compare the epistemic and total variance decomposition in Section 5, we set $Q_t(a) = p(r|a, \mathcal{D}_t)$, where $\mathcal{D}_t = \{(a_i, r_i)\}_{i=1}^{t-1}$ is the prior action, reward pairs already observed in a run. In the epistemic variance setting $U_t(a) = \mathbb{E}_U[\operatorname{Var}[r|a, Z, \mathcal{D}_t]]$ and in the total variance setting $U_t(a) = \operatorname{Var}[r|a, \mathcal{D}_t]$. For each α and p , we run 10 seeds.

Non-LLM Benchmark. We use the standard UCB1 algorithm and the Greedy algorithm [30] as a non-LLM benchmark to ensure that the LLM-UCB algorithm has comparable performance to

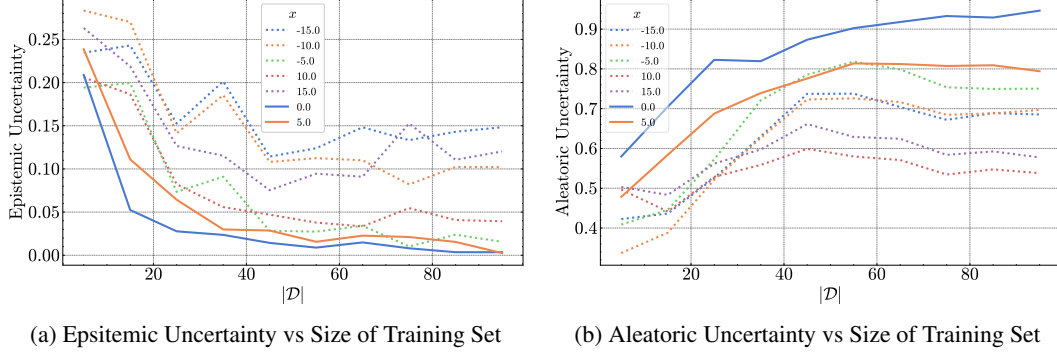


Figure 15: Epistemic Uncertainty and Aleatoric Uncertainty vs Dataset Size for Qwen2.5-7B model.

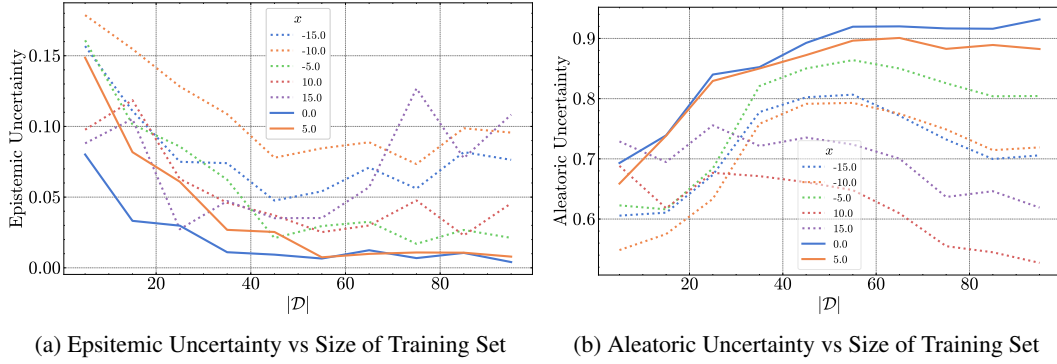


Figure 16: Epistemic Uncertainty and Aleatoric Uncertainty vs Dataset Size for Llama3.1-8B model.

standard bandit algorithms. An exploration rate of $\alpha = 0.75$ is used for the UCB1 algorithm. We run 5000 seeds for both UCB1 and Greedy for each α and p .

Instruction Prompting Benchmark. In [27] an instruction-tuned LLM is prompted to attempt the Buttons bandit task and there is a thorough investigation of the impact of the prompt configuration on the LLM performance. The authors conclude the most successful prompt configuration is: *BSSC0*, which consists of: a suggestive framing that the LLM is solving a bandit task; a summarised history of prior actions (including average rewards per action and counts per action); reinforced chain-of-thought prompting; and a temperature parameter of 0. For fair comparison of model performance, we use Qwen2.5-14B-Instruct, Qwen2.5-7B-Instruct, and Llama3.1-8B-Instruct [51, 60] to benchmark the performance of the LLM-UCB algorithm for the base models Qwen2.5-14B, Qwen2.5-7B, and Llama3.1-8B respectively. See Appendix H.2 for an example prompt. For each α and p , we run 10 seeds.

Number of Trials. For all the bandit experiments, we run the algorithm for $T = 200$ trials.

Role of p and aleatoric variance. The means of the optimal and suboptimal arm(s) in the Buttons setting are $p_a^* = p + \frac{\delta}{2}$ and $p_a = p - \frac{\delta}{2}$ respectively. Now, the variance for a Bernoulli random variable of mean q is $q(1 - q)$. This is a quadratic with a maximum at $q = \frac{1}{2}$. Therefore, if $p > \frac{1}{2}$,

$$|p_a - \frac{1}{2}| = |(p - \frac{1}{2}) - \frac{\Delta}{2}| < |(p - \frac{1}{2})| + |\frac{\Delta}{2}| = p - \frac{1}{2} + \frac{\Delta}{2} = |p_a^* - \frac{1}{2}|.$$

Therefore, the true variance of the suboptimal arm is higher than the true variance of the optimal arm.

Choice of α . In our experiments, we choose $\alpha = 2, 5$. In UCB1 smaller choices of α are typically chosen [30], however this is primarily due to the slow decay of $U_t(a)$ in the UCB1 algorithm. The decrease in epistemic uncertainty with the number of trials is significantly faster, and therefore, we use higher α . Since the total uncertainty is the sum of the epistemic uncertainty and the aleatoric uncertainty, the difference in the uncertainties is α multiplied by aleatoric uncertainty.

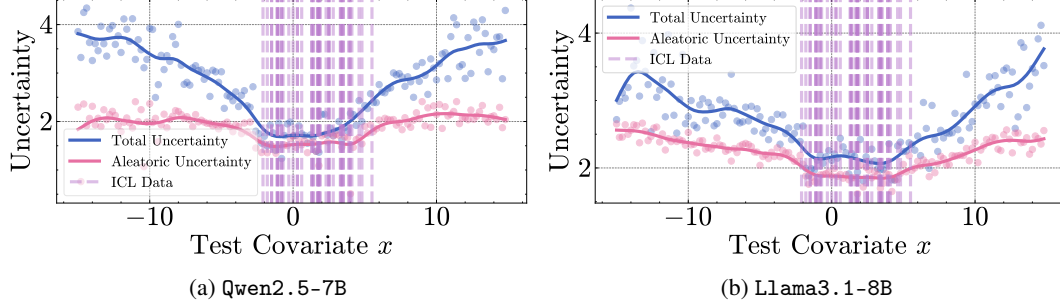


Figure 17: Linear Regression (Entropic) Uncertainty Decomposition for Qwen2.5-7B and Llama3.1-8B models.

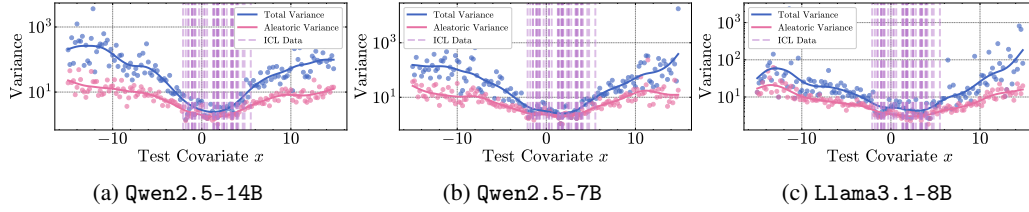


Figure 18: Linear Regression Variance Decomposition for Qwen2.5-14B, Qwen2.5-7B and Llama3.1-8B models.

Metrics. We use multiple metrics to assess the performance of the bandit algorithms. Suffix-fail frequency and $K \cdot \text{MinFrac}$ are metrics introduced in [27] to assess the performance of bandit runs.

- **Mean regret:** For a run of T trials, the mean regret is defined as $\frac{1}{T} \sum_{i=1}^T \mathbb{E}[r(a_t)] - \mu^*$, where μ^* is the optimal reward and $\mathbb{E}[r(a_t)]$ is the mean reward for arm a_t . We report the mean and standard deviation across the different seeds.
- **Mean worst-case regret:** We take the mean and standard deviation over the 30% of seeds with the highest mean regret. For algorithms where there is a large discrepancy between the mean regret and worst case mean regret, this indicates that the variability in the performance of the bandit algorithm is high.
- **Median reward:** For each seed run, we compute the mean reward $\frac{1}{T} \sum_{i=1}^T r_t$. We then report the median mean reward across all the seeds.
- **Suffix-fail frequency:** For a given run, there is a t -suffix failure, if the optimal arm is not chosen in the trials $[t, T]$. The suffix fail frequency $\text{SuffFailFreq}(t)$ is the proportion of t -suffix failures across all the seeds. This metric measures a particular failure mode of bandit-algorithms due to lack of exploration, where as a result, the optimal arm is not chosen.
- **$K \cdot \text{MinFrac}$:** For a given run j , let $S_a^{(j)}$ be the action counts. Given T runs, J seeds, and K arms, $K \cdot \text{MinFrac} = \frac{K}{TJ} \sum_{j=1}^J \min_a S_a^{(j)}$. This metric measures *uniform-like* failures of bandit algorithms, where due to excessive exploration, the algorithm behaves closely to one that uniformly chooses an action.

Results. In Tables 6 and 7, we provide the results for the Qwen2.5-7B and Llama3.1-8B models. We also plot the average cumulative regret across different seeds for $p = 0.5, 0.6, 0.7$ and $\alpha = 2, 5$ in Figures 32-49. Each line in these figures corresponds to the cumulative regret for a particular seed. Here, we see that in general, the algorithm that uses the epistemic variance estimate generally has more consistent performance than the algorithm that uses total variance.

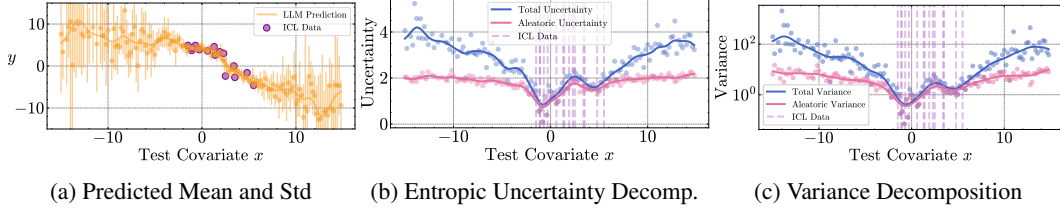


Figure 19: Uncertainty Decompositions for Linear Regression, single seed. Model: Qwen2.5-14B.

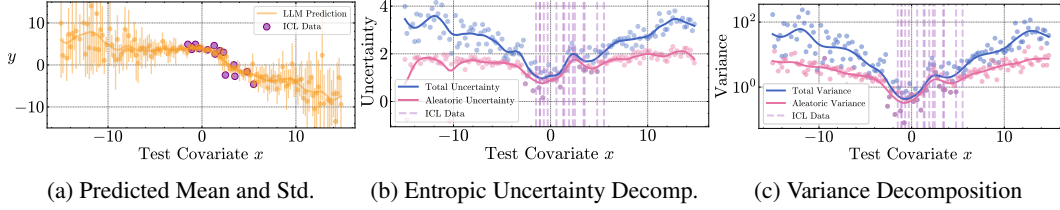


Figure 20: Uncertainty Decompositions for Linear Regression, single seed. Model: Qwen2.5-7B.

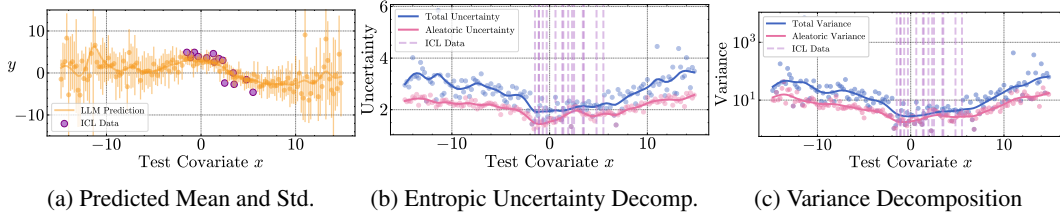


Figure 21: Uncertainty Decompositions for Linear Regression, single seed. Model: Llama3.1-8B.

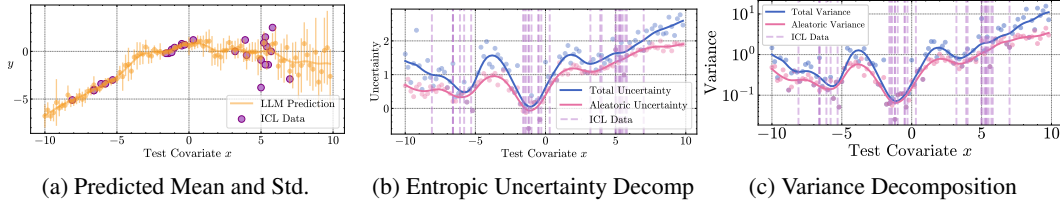


Figure 22: Uncertainty Decompositions for Regression Tasks with Gaps in ICL Data. Model: Qwen2.5-14B

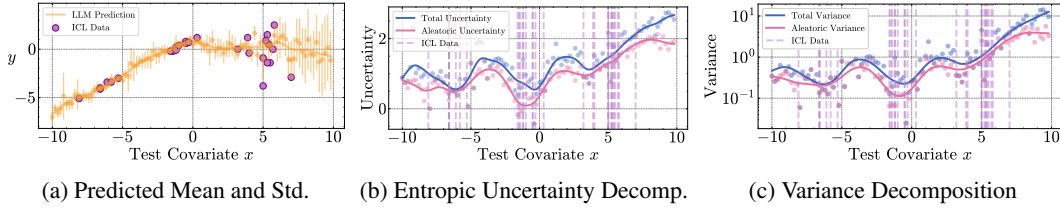


Figure 23: Uncertainty Decompositions for Regression Tasks with Gaps in ICL Data. Model: Qwen2.5-7B

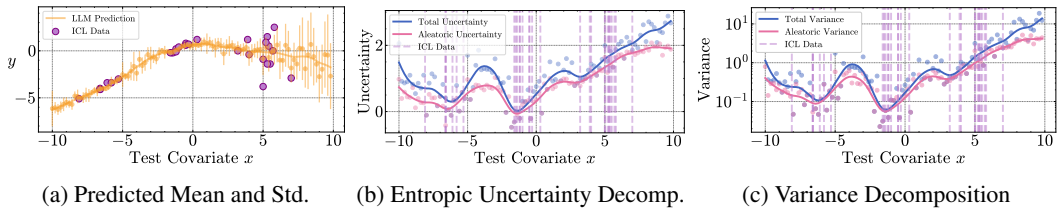


Figure 24: Uncertainty Decompositions for Regression Tasks with Gaps in ICL Data. Model: Llama3.1-8B

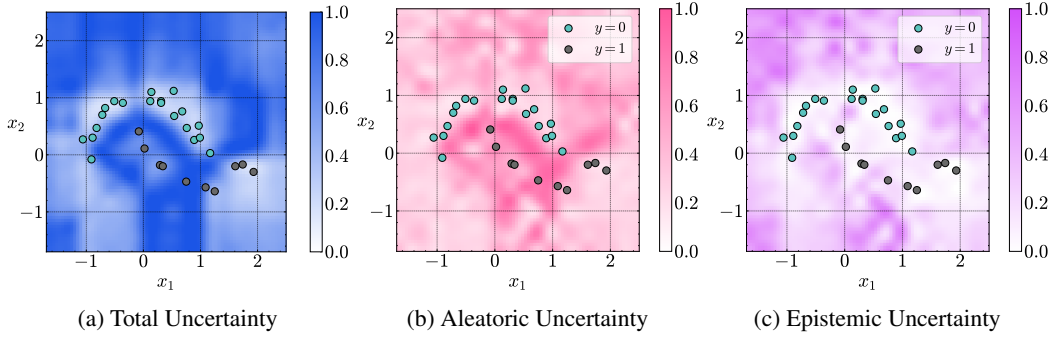


Figure 25: Uncertainty Decomposition for "Moons 1" Dataset. Model: Qwen2.5-7B.

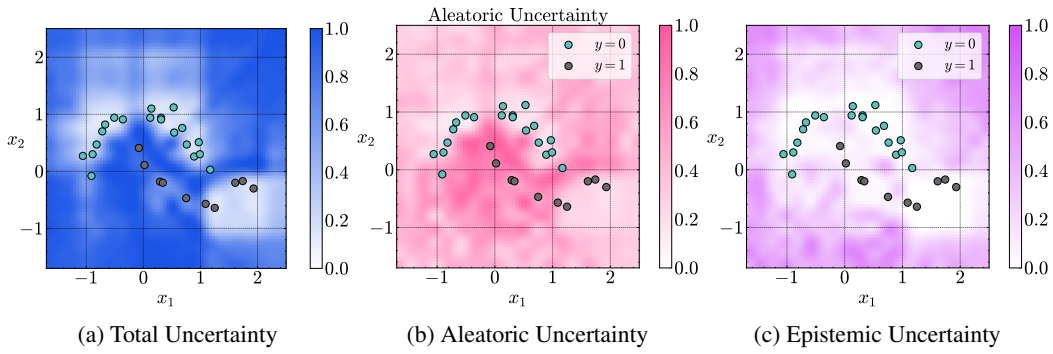


Figure 26: Uncertainty Decomposition for "Moons 1" Dataset. Model: Llama3.1-8B.

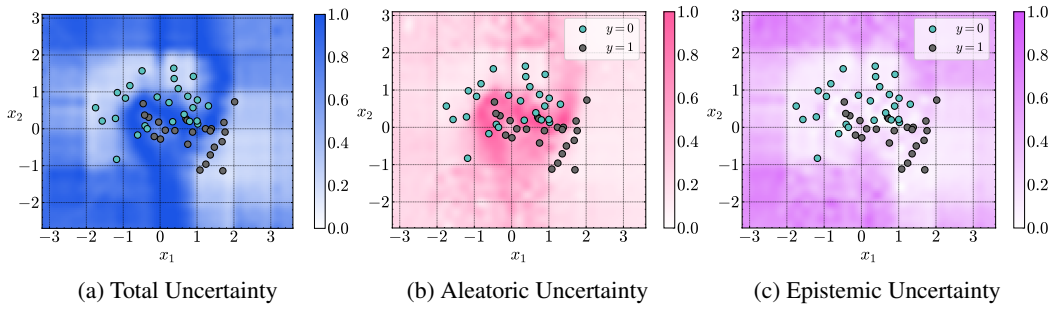


Figure 27: Uncertainty Decomposition for "Moons 2" Dataset. Model: Qwen2.5-14B.

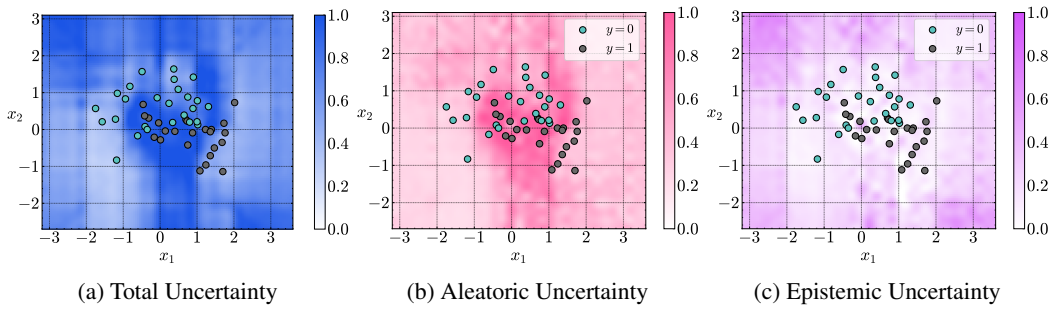


Figure 28: Uncertainty Decomposition for "Moons 2" Dataset. Model: Qwen2.5-7B.

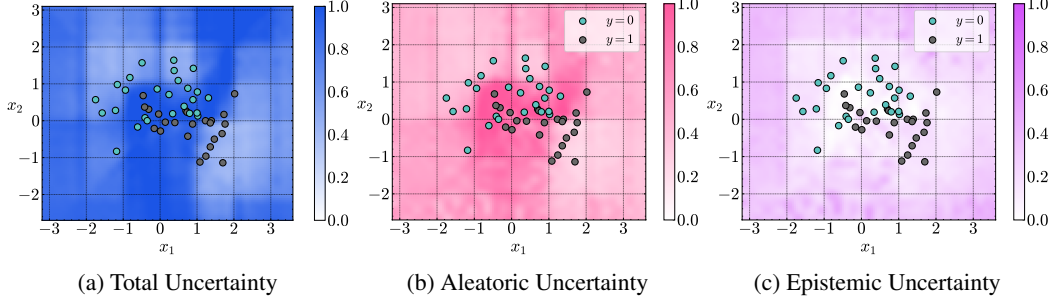


Figure 29: Uncertainty Decomposition for "Moons 2" Dataset. Model: Llama3.1-8B.

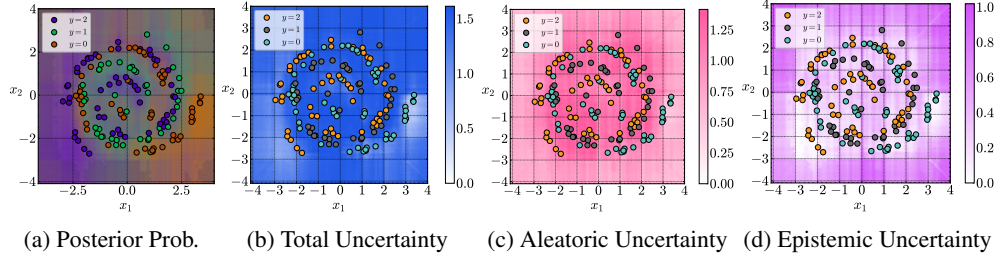


Figure 30: Uncertainty Decompositions for Spirals Classification Task. Model: Qwen2.5-7B

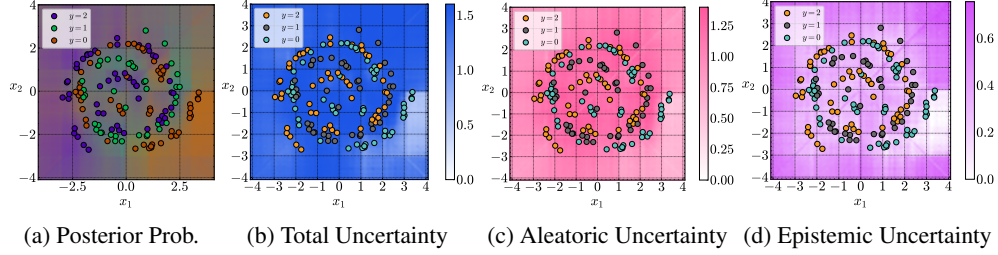


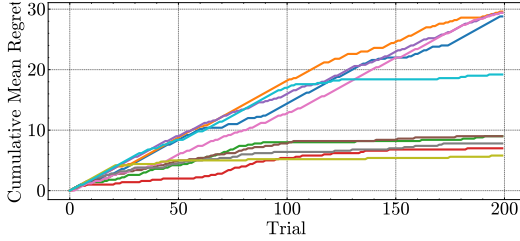
Figure 31: Uncertainty Decompositions for Spirals Classification Task. Model: Llama3.1-8B

Table 6: Buttons Bandit Problem. TV is Total Variance and EV is Epistemic Variance. Model: Qwen2.5-7B

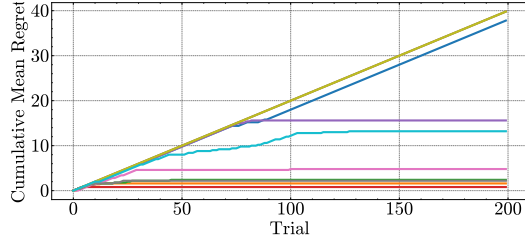
	METHOD	MEAN WORST-CASE REGRET \downarrow	MEAN REGRET \downarrow	MEDIAN REWARD \uparrow	SuffFailFreq($T/2$) \downarrow	$K \cdot \text{MinFrac} \downarrow$
$p = 0.5$	UCB	$0.128 \pm .019$	$0.094 \pm .027$	0.510	0.0	0.29
	GREEDY	$0.199 \pm .000$	$0.101 \pm .092$	0.525	0.460	0.03
	INSTRUCT BASELINE	$0.161 \pm .020$	$0.107 \pm .043$	0.495	0.0	0.26
	TV ($\alpha = 2$)	$0.175 \pm .027$	$0.068 \pm .074$	0.565	0.1	0.03
	EV ($\alpha = 2$)	$0.144 \pm .042$	$0.091 \pm .044$	0.535	0.0	0.24
	TV ($\alpha = 5$)	$0.196 \pm .003$	$0.075 \pm .081$	0.545	0.2	0.04
	EV ($\alpha = 5$)	$0.160 \pm .010$	$0.132 \pm .020$	0.463	0.0	0.57
$p = 0.6$	UCB1	$0.127 \pm .018$	$0.094 \pm .027$	0.610	0.0	0.28
	GREEDY	$0.199 \pm .000$	$0.092 \pm .090$	0.645	0.396	0.03
	INSTRUCT BASELINE	$0.111 \pm .007$	$0.076 \pm .043$	0.620	0.0	0.18
	TV ($\alpha = 2$)	$0.199 \pm .000$	$0.090 \pm .089$	0.627	0.3	0.03
	EV ($\alpha = 2$)	$0.088 \pm .002$	$0.061 \pm .026$	0.627	0.0	0.12
	TV ($\alpha = 5$)	$0.198 \pm .001$	$0.167 \pm .032$	0.570	0.5	0.07
	EV ($\alpha = 5$)	$0.156 \pm .016$	$0.117 \pm .030$	0.583	0.0	0.43
$p = 0.7$	UCB1	$0.122 \pm .017$	$0.094 \pm .027$	0.710	0.0	0.27
	GREEDY	$0.199 \pm .000$	$0.085 \pm .089$	0.760	0.369	0.03
	INSTRUCT BASELINE	$0.132 \pm .043$	$0.087 \pm .040$	0.703	0.0	0.18
	TV ($\alpha = 2$)	$0.198 \pm .001$	$0.088 \pm .091$	0.728	0.4	0.02
	EV ($\alpha = 2$)	$0.141 \pm .040$	$0.070 \pm .056$	0.720	0.0	0.09
	TV ($\alpha = 5$)	$0.195 \pm .004$	$0.149 \pm .073$	0.608	0.8	0.04
	EV ($\alpha = 5$)	$0.143 \pm .014$	$0.116 \pm .026$	0.667	0.0	0.38

Table 7: Buttons Bandit Problem. TV is Total Variance and EV is Epistemic Variance. Model: Llama3.1-8B

	METHOD	MEAN WORST-CASE REGRET \downarrow	MEAN REGRET \downarrow	MEDIAN REWARD \uparrow	SuffFailFreq($T/2$) \downarrow	$K \cdot \text{MinFrac} \downarrow$
$p = 0.5$	UCB	$0.128 \pm .019$	$0.094 \pm .027$	0.510	0.0	0.29
	GREEDY	$0.199 \pm .000$	$0.101 \pm .092$	0.525	0.460	0.03
	INSTRUCT BASELINE	$0.161 \pm .020$	$0.107 \pm .043$	0.495	0.0	0.26
	TV ($\alpha = 2$)	$0.160 \pm .055$	$0.071 \pm .071$	0.557	0.2	0.05
	EV ($\alpha = 2$)	0.149 $\pm .009$	0.097 $\pm .043$	0.505	0.0	0.21
	TV ($\alpha = 5$)	$0.149 \pm .036$	$0.066 \pm .061$	0.555	0.1	0.05
	EV ($\alpha = 5$)	0.169 $\pm .002$	0.153 $\pm .019$	0.432	0.0	0.73
$p = 0.6$	UCB1	$0.127 \pm .018$	$0.094 \pm .027$	0.610	0.0	0.28
	GREEDY	$0.199 \pm .000$	$0.092 \pm .090$	0.645	0.396	0.03
	INSTRUCT BASELINE	$0.111 \pm .007$	$0.076 \pm .043$	0.620	0.0	0.18
	TV ($\alpha = 2$)	$0.088 \pm .076$	$0.035 \pm .054$	0.670	0.1	0.04
	EV ($\alpha = 2$)	0.140 $\pm .045$	0.077 $\pm .051$	0.635	0.0	0.17
	TV ($\alpha = 5$)	$0.198 \pm .001$	$0.138 \pm .078$	0.568	0.6	0.04
	EV ($\alpha = 5$)	0.139 $\pm .004$	0.113 $\pm .022$	0.588	0.0	0.50
$p = 0.7$	UCB1	$0.122 \pm .017$	$0.094 \pm .027$	0.710	0.0	0.27
	GREEDY	$0.199 \pm .000$	$0.085 \pm .089$	0.760	0.369	0.03
	INSTRUCT BASELINE	$0.132 \pm .043$	$0.087 \pm .040$	0.703	0.0	0.18
	TV ($\alpha = 2$)	$0.168 \pm .041$	$0.063 \pm .075$	0.728	0.1	0.04
	EV ($\alpha = 2$)	0.111 $\pm .021$	0.053 $\pm .042$	0.745	0.0	0.08
	TV ($\alpha = 5$)	$0.197 \pm .002$	$0.165 \pm .041$	0.613	0.5	0.04
	EV ($\alpha = 5$)	0.127 $\pm .021$	0.087 $\pm .035$	0.688	0.0	0.35

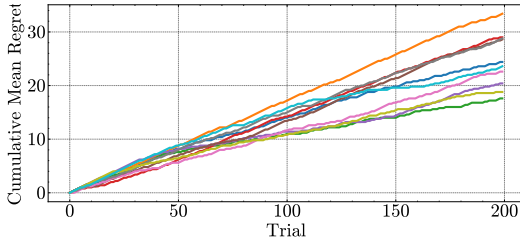


(a) Epistemic Variance

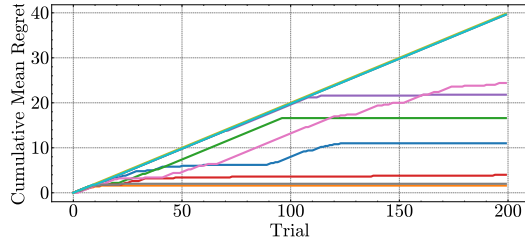


(b) Total Variance

Figure 32: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-14B, $p = 0.5$, $\alpha = 2$).

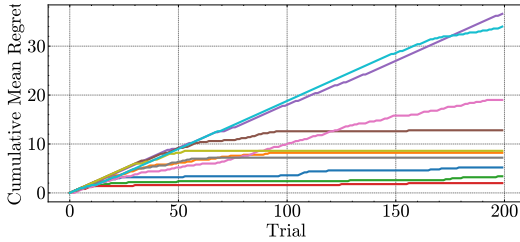


(a) Epistemic Variance

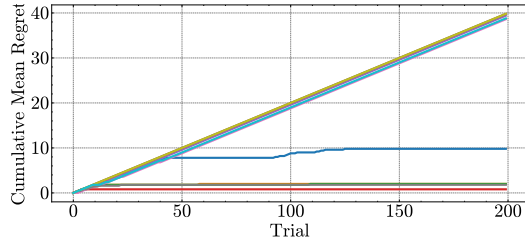


(b) Total Variance

Figure 33: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-14B, $p = 0.5$, $\alpha = 5$).



(a) Epistemic Variance



(b) Total Variance

Figure 34: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-14B, $p = 0.6$, $\alpha = 2$).

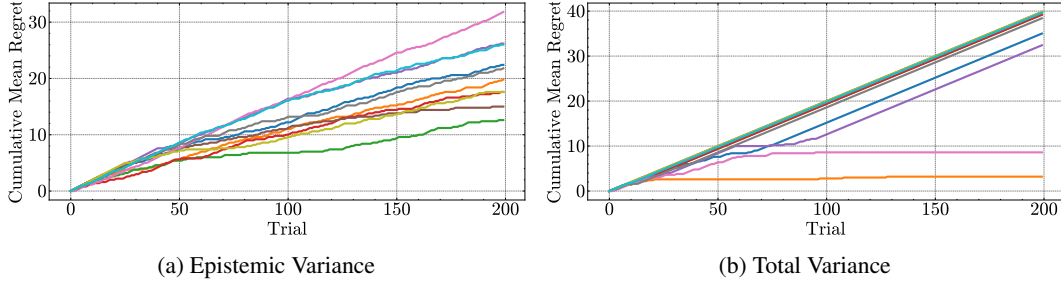


Figure 35: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-14B, $p = 0.6$, $\alpha = 5$).

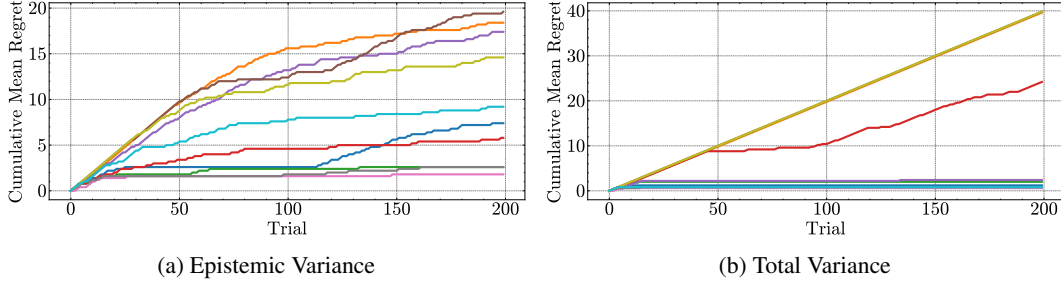


Figure 36: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-14B, $p = 0.7$, $\alpha = 2$).

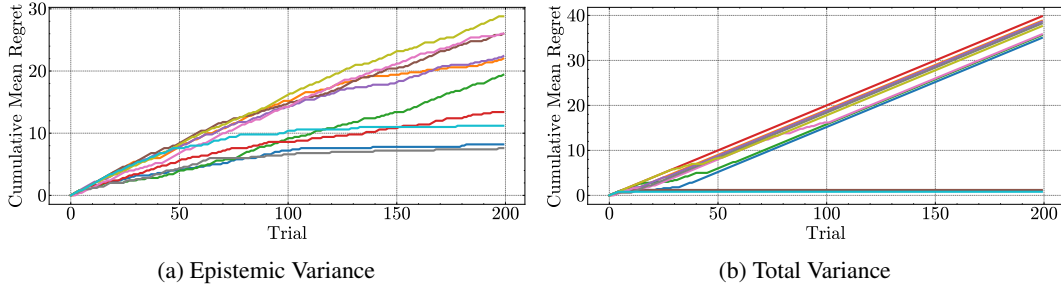


Figure 37: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-14B, $p = 0.7$, $\alpha = 5$).

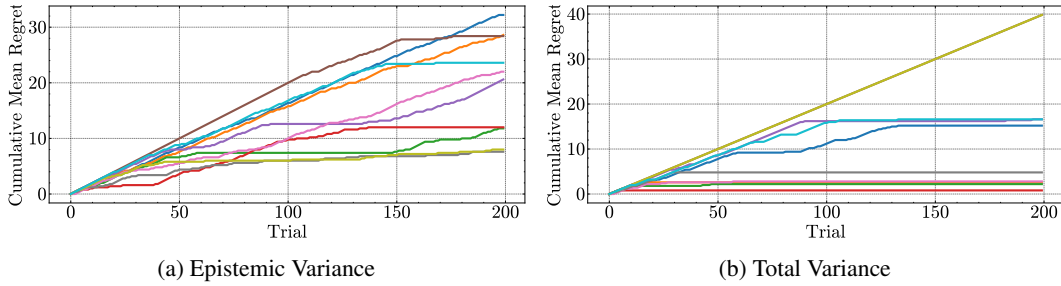


Figure 38: Cumulative Mean Regret for Bandit Experiments (Model Llama3.1-8B, $p = 0.5$, $\alpha = 2$).

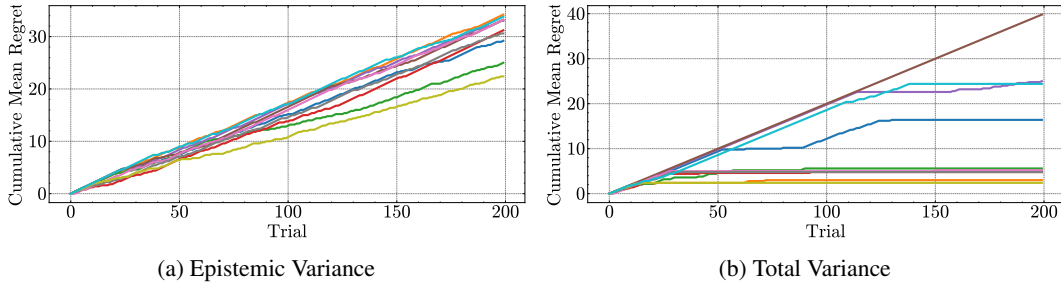


Figure 39: Cumulative Mean Regret for Bandit Experiments (Model Llama3.1-8B, $p = 0.5$, $\alpha = 5$).

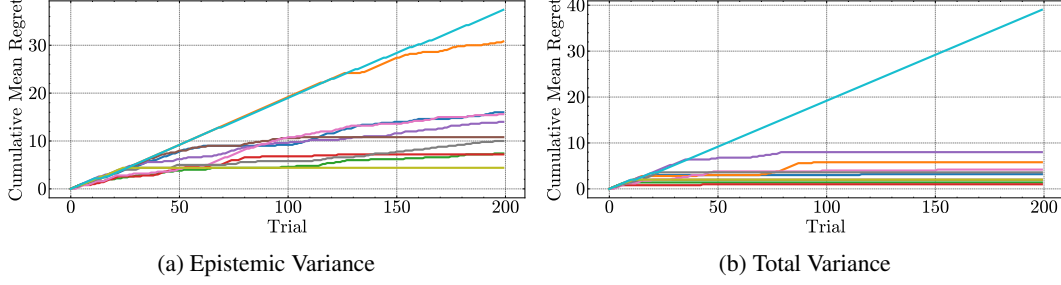


Figure 40: Cumulative Mean Regret for Bandit Experiments (Model Llama3.1-8B, $p = 0.6$, $\alpha = 2$).

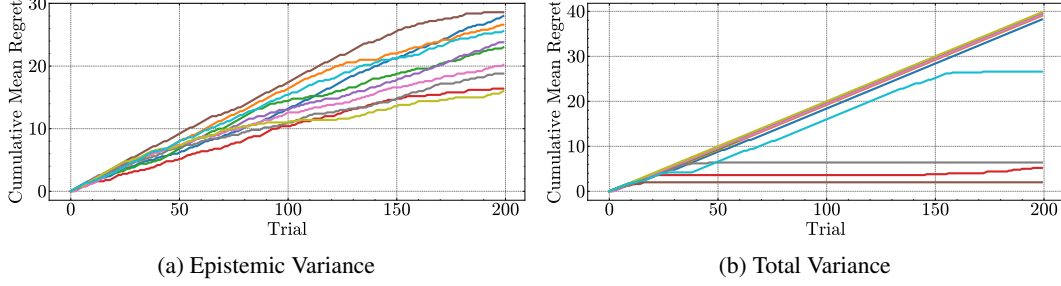


Figure 41: Cumulative Mean Regret for Bandit Experiments (Model Llama3.1-8B, $p = 0.6$, $\alpha = 5$).

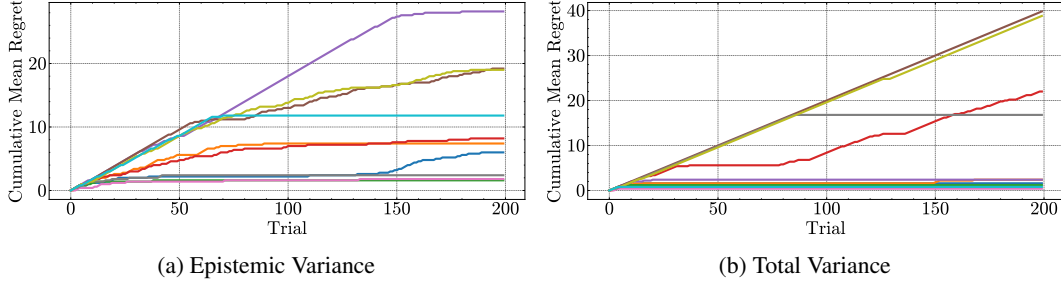


Figure 42: Cumulative Mean Regret for Bandit Experiments (Model Llama3.1-8B, $p = 0.7$, $\alpha = 2$).

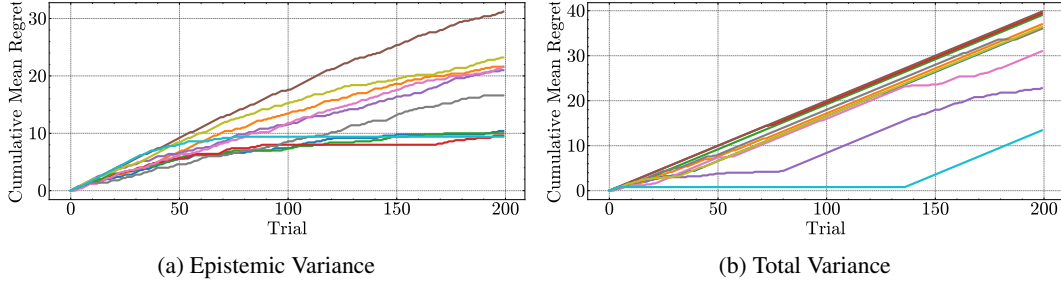


Figure 43: Cumulative Mean Regret for Bandit Experiments (Model Llama3.1-8B, $p = 0.7$, $\alpha = 5$).

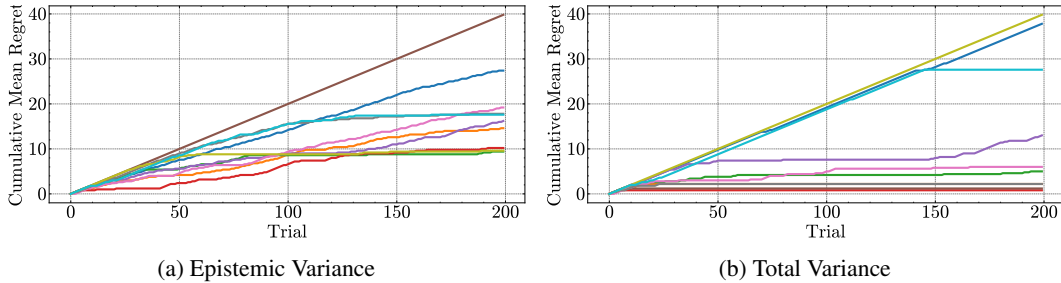


Figure 44: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-7B, $p = 0.5$, $\alpha = 2$).

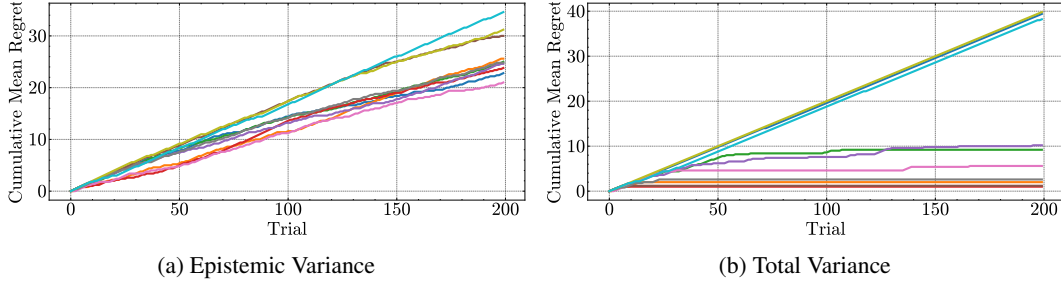


Figure 45: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-7B, $p = 0.5$, $\alpha = 5$).

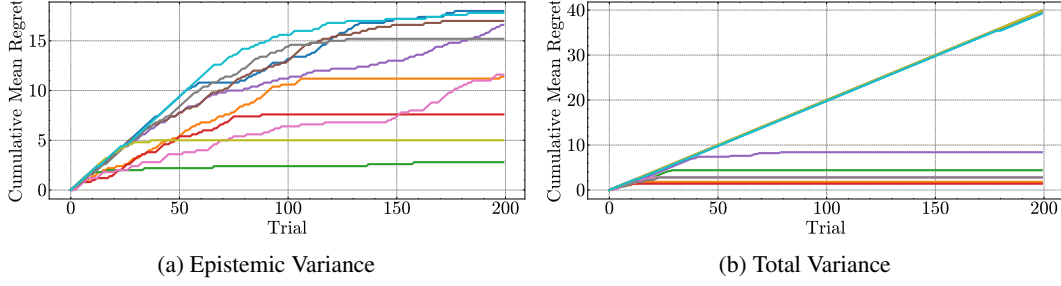


Figure 46: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-7B, $p = 0.6$, $\alpha = 2$).

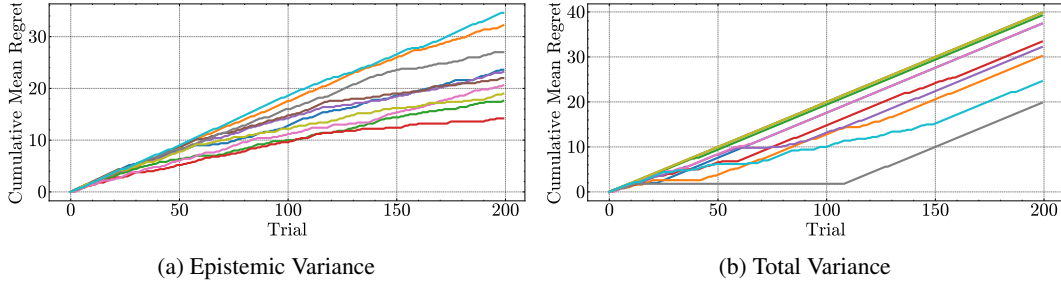


Figure 47: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-7B, $p = 0.6$, $\alpha = 5$).

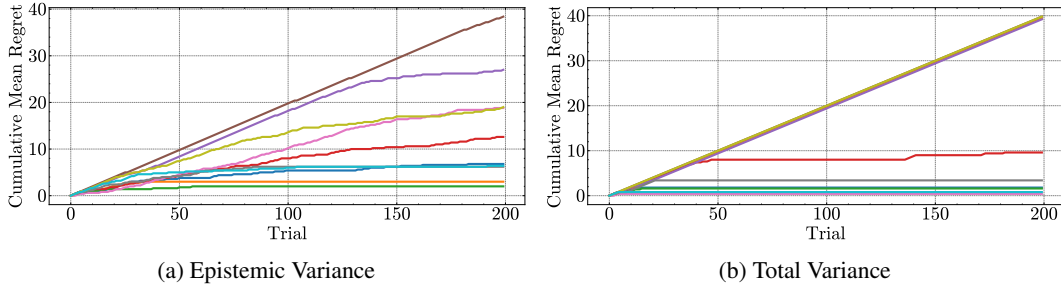


Figure 48: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-7B, $p = 0.7$, $\alpha = 2$).

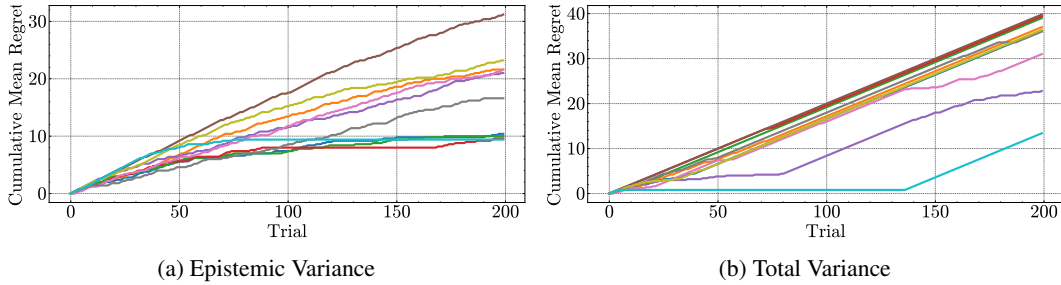


Figure 49: Cumulative Mean Regret for Bandit Experiments (Model Qwen2.5-7B, $p = 0.7$, $\alpha = 5$).

1208 F.4 Out-of-Distribution Detection

1209 **Datasets.** In our experiments, we leverage the BoolQA [8], HotpotQA [69], and PubMedQA [24]
1210 datasets. BoolQA is a reading comprehension dataset that studies yes/no questions. HotpotQA is a
1211 dataset with Wikipedia-based questions that contain complex reasoning explanations for answers.
1212 PubMedQA is a biomedical question answering dataset collected from PubMed abstracts to answer
1213 research questions with yes/no/maybe. For each dataset, we preprocess them by extracting the
1214 “yes/no” questions, followed by formulating each sample in a “Question:... Context:...” format and
1215 mapping its labels into integers: {“no”:0, “yes”:1}.

1216 **Benchmarks.** We perform out-of-distribution (OOD) detection via area under the ROC curve (AUC)
1217 [17]. The test set consists of the concatenated ID and OOD datasets of equal size, each labeled
1218 respectively under a binary “is_ood” column. For each sample in the test set, we compute the aleatoric,
1219 epistemic, and total uncertainties using our VUD method. Using the epistemic and total uncertainties,
1220 we fit them against the “is_ood” column using an AUROC curve. This yields our results in Table 2.

1221 G Further Related Works

1222 **Bayesian Interpretations of In-Context Learning.** Works in recent years [66, 50, 42] suggested
1223 that the behaviour of transformers during in-context learning emulates Bayesian inference. In our
1224 work, this Bayesian behaviour of ICL is a key assumption that is necessary for the validity of the
1225 variational uncertainty decomposition algorithm. However, there is also evidence to suggest that
1226 this Bayesian behaviour is only approximate during long-term generation in LLMs, this Bayesian
1227 assumption breaks down [12, 35]. We try to enforce permutation-invariant generation and filter
1228 non-Bayesian generation from auxiliary data to maintain the Bayesian assumption that we make.

1229 **Permutation Invariance and Exchangeability in LLMs.** The generation in language models is
1230 dependent on the position of tokens [36, 73]. This is a clear violation of exchangeability, which is
1231 necessary for the application of De Finetti’s. [71] assumes the exchangeability of LLM generation
1232 to apply De Finetti which allows for the estimation of the topic distributions from LLMs. However,
1233 they do not apply permutations during ICL to the training data. [70] discusses the importance of
1234 exchangeability for quantifying uncertainty in ICL. They investigate methods to promote permutation
1235 invariance during pre-training and fine-tuning or architectural modifications to the transformer through
1236 causal masking. Whilst they suggest using permuted data as a data augmentation technique during
1237 training, our permutation invariant conditional generation is purely applied during inference. This
1238 incurs a greater cost during inference time but does not require fine-tuning of the LLM. Assuming
1239 that the generation of the data in the transformer is exchangeable, then we can directly estimate the
1240 distribution of the latent variable θ through Martingale posterior distributions [12, 13, 31]. However,
1241 this estimate requires the long-run trajectory of auto-regressively generated samples to obtain an
1242 approximate sample of a single parameter θ .

1243 **Uncertainty quantification for LLMs.** Prior work in uncertainty quantification has focused on
1244 quantifying the total uncertainty for predictive tasks or uncertainty quantification over different
1245 aspects of a generated response. One of the challenges of uncertainty quantification in LLMs and
1246 generative models in general is that a response to a question may appear different but be semantically
1247 similar (for example, “Paris” and “The capital of France is Paris” are equally valid responses to
1248 the question “What is the capital of France?” [28]. In our work, we focus on predictive tasks in a
1249 regression or multi-class setting and use the prompt structure to elicit simple responses from a small
1250 set of classes or a real number.

1251 **Uncertainty decomposition for LLM in-context predictions.** Uncertainty decomposition for
1252 LLMs has also been explored in previous works; however, the definitions of aleatoric and epistemic
1253 uncertainty vary from the traditional definitions in prior Bayesian literature. [22] considers the
1254 aleatoric uncertainty of a response as the ambiguity in the input. Therefore, given a distribution
1255 of “clarifications” $q(\mathbf{C}|\mathbf{x}^*)$ for a particular prompt, the aleatoric uncertainty is defined as the *mean*
1256 conditional uncertainty of a particular clarification $\mathbb{E}_{q(\mathbf{C}|\mathbf{x}^*)}[H[\mathbf{y}^*|\mathbf{x}^* \oplus \mathbf{C}]]$. In contrast, we seek
1257 to find the minimal conditional entropy given auxiliary data, which acts as an upper bound to the
1258 underlying Bayesian conditional entropy. Furthermore, the focus of [22] is primarily zero-shot and
1259 few-shot prediction, whereas we consider tasks where a training dataset is provided in context.

[34] approaches uncertainty decomposition of in-context learning by also employing the interpretation that ICL performs Bayesian inference. However, they define epistemic uncertainty as the conditional entropy $\mathbb{E}_{p(\theta|\mathcal{D})}[H[\mathbf{y}^*|\mathbf{x}^*, \theta]]$ and aleatoric uncertainty as the mutual information $\mathbb{I}(\mathbf{y}^*; \theta|\mathbf{x}^*, \mathcal{D})$. This reverses the traditional definitions of Bayesian uncertainty decomposition [25] and therefore, we do not use this as a baseline.

Bayesian Approaches to Transformers. In this work, we view in-context learning as implicit Bayesian inference. However, prior work has connected the transformer architecture with Bayesian inference more explicitly via Bayes-by-backprop approaches [55, 38, 6]. In particular, low-rank adaptation [68, 4, 47] has allowed for parameter-efficient avenues for Bayesian deep learning in transformers. Alternatively, neural processes have been integrated with transformers [45] to provide another approach to Bayesian uncertainty quantification in transformers.

Applications to In-Context Exploration. Techniques used to quantify uncertainty in LLM predictions can be used to drive in-context exploration-exploitation tasks. In reinforcement learning and bandit tasks, efficient exploration algorithms such as Upper Confidence Bound [30, 3] and Thompson Sampling (TS) [49, 48, 56] require modelling the epistemic posterior distribution over possible outcomes either implicitly, through visitation counts, or explicitly, for example via ensembles. By modelling the epistemic uncertainty, the agent is able to reason about potential outcomes with uncertainty due to lack of data and explore in promising directions.

Previous work that analyses the in-context exploration capabilities of LLMs includes [27], where the exploration capabilities of LLMs are compared to those of standard algorithms on small-scale tasks, and [41], which investigates the exploration capability of LLMs on natural language bandit tasks. The work in [46] further explores and benchmarks LLMs’ abilities on a number of bandit tasks and offers ways to improve the efficiency of exploration by introducing algorithmic enhancements that better align LLMs with the exploration-exploitation task. This line of work focusing on bandits is complemented by [64], which extends the benchmarking to include multi-step tasks in addition to bandits. Finally, the work in [1] adapts the TS heuristic to the LLM setting, enabling LLM agents to tackle sequential decision-making tasks analogous to that of the full reinforcement learning setting.

Uncertainty-aware exploration has also been used in active-learning settings to obtain smoother decision boundaries of LLMs by identifying the data points that will give smoother boundaries [72].

OOD Detection. Detecting out-of-distribution (OOD) inputs is critical for real-world applications such as medical diagnosis and autonomous driving, where models can make confidently wrong predictions on inputs far from the training distribution. Foundational work demonstrated that softmax confidence often fails under distributional shift, establishing simple baselines for OOD detection in deep neural networks [19]. However, epistemic uncertainty has been shown to be useful in OOD and hallucination detection [65, 25]. This led to uncertainty-based methods which estimate epistemic uncertainty such as deep ensembles [29], where the uncertainty is measured through model diversity, and prior networks where distributional uncertainty is used in addition to epistemic uncertainty [37]. In NLP, pre-trained language models have been used for OOD detection [18] through non-Bayesian approaches such as contrastive learning [74], unsupervised detection with transformers [67], and conditional generation strategies to improve OOD discriminability [53]. Extensions to multimodal settings further explore OOD detection in vision-language tasks [40].

1301 H Example Prompts

1302 H.1 Synthetic Toy Prompts

Prompt Template for Synthetic Classification Experiments

```
x1 = -1.75; x2 = 0.57 <output>0<\output>  
x1 = -0.16; x2 = -0.21 <output>1<\output>  
x1 = 0.4; x2 = -0.05 <output>1<\output>  
x1 = 0.2; x2 = 0.4 <output>
```

Prompt Template for Synthetic Regression Experiments

```
x = -0.7 <output> 4.9 <\output>  
x = -1.1 <output> 3.7 <\output>  
x = 4.8 <output> -1.6 <\output>  
x = 0.2 <output>
```

1305 H.2 Bandit Prompts

Prompt Template for Bandit Classification Experiments (LLM-UCB Algorithm)

```
action = 0 <reward>1<\reward>  
action = 1 <reward>0<\reward>  
action = 3 <reward>1<\reward>  
action = 1 <reward>
```

Prompt Template for Bandit Classification Experiments (Instruct Baseline)

<|system|>

You are a bandit algorithm in a room with 5 buttons labeled blue, green, red, yellow, purple. Each button is associated with a Bernoulli distribution with a fixed but unknown mean; the means for the buttons could be different. For each button, when you press it, you will get a reward that is sampled from the button's associated distribution. You have 200 time steps and, on each time step, you can choose any button and receive the reward. Your goal is to maximize the total reward over the 10 time steps.

At each time step, I will show you a summary of your past choices and rewards. Then you must make the next choice, which must be exactly one of blue, green, red, yellow, purple. Let's think step by step to make sure we make a good choice. You must provide your final answer within the tags <Answer>COLOR</Answer> where COLOR is one of blue, green, red, yellow, purple.

<|user|>

So far you have played 7 times with your past choices and rewards summarized as follows:

blue button: pressed 3 times with average reward 0.67

green button: pressed 2 times with average reward 0.50

red button: pressed 0 times

yellow button: pressed 1 times with average reward 0.00

purple button: pressed 1 times with average reward 1.00

Which button will you choose next? Remember, YOU MUST provide your final answer within the tags <Answer>COLOR</Answer> where COLOR is one of blue, green, red, yellow, purple. Let's think step by step to make sure we make a good choice.

<|assistant|>

1307

Prompt Template for Question-Answering Tasks (Prediction)

You are given a set of in-context examples and a new input.
Your task is to predict the label of the new input.

Please carefully review the following examples and their labels inside
<output>{labels}</output> tags:

Question: is marley from...

Context: when john senses...

<output>1</output>

Question: are all the...

Context: following the unsuccessful...

<output>0</output>

...

Now, predict the label for this new input:

Question: did the titans...

Context: despite bertier's paralysis...

IMPORTANT: Output ONLY the label inside <output></output> tags.

Do not add any explanation, text, or formatting.

Your response must strictly follow this format:

<output>{label_prediction}</output>

1309

Prompt Template for Question-Answering Tasks (Z Perturbations)

Please rephrase the following:

Question: do the titans ...

Context: while celebrating ...

While rephrasing the above, incorporate context from the following and
make sure its intertwined/interconnected:

Question: did zz top play ...

Context: "doubleback" is a song ...

Use the following format when rephrasing:

<rep> Question: {Rephrased Question}?

Context: {Rephrased Context}. </rep>

1310