Bayesian Prompt Ensembles: Model Uncertainty Estimation for Black-Box Large Language Models

Francesco Tonolini
Amazon
tonolini@amazon.com

Jordan Massiah
Amazon
jormas@amazon.com

Nikolaos Aletras
Amazon & University of Sheffield
aletras@amazon.com

Gabriella Kazai
Amazon
gkazai@amazon.com

Abstract

An important requirement for the reliable deployment of pre-trained large language models (LLMs) is the well-calibrated quantification of the uncertainty in their outputs. While the likelihood of predicting the next token is a practical surrogate of the data uncertainty learned during training, model uncertainty is challenging to estimate, i.e., due to lack of knowledge acquired during training. Prior efforts to quantify uncertainty of neural networks require specific architectures or (re-)training strategies, which are impractical to apply to LLMs with several billion parameters, or for black-box models where the architecture and parameters are not available. In this paper, we propose Bayesian Prompts Ensembles (BayesPE), a novel approach to effectively obtain well-calibrated uncertainty for the output of pre-trained LLMs. BayesPE computes output probabilities through a weighted ensemble of different, but semantically equivalent, task instruction prompts. The relative weights of the different prompts in the ensemble are estimated through approximate Bayesian variational inference over a small labeled validation set. We demonstrate that BayesPE approximates a Bayesian input layer for the LLM, providing a lower bound on the expected model error. In our extensive experiments, we show that BayesPE achieves significantly superior uncertainty calibration compared to several baselines over a range of natural language classification tasks, both in zero- and few-shot settings.

1 Introduction

Generative pre-trained large language models (LLMs) have proven to be very effective at performing a broad variety of natural language processing (NLP) tasks in zero- and few-shot settings (Brown et al., 2020; Touvron et al., 2023; Jiang et al., 2023a). These capabilities make them attractive alternatives to task-specific fine-tuned models, as they do not require large annotated data sets or gradient updates, but simply take as input relevant text instructions (i.e., prompts) to perform the inference of interest (Petroni et al., 2019; Shin et al., 2020; Sanh et al., 2021; Brown et al., 2020).

However, safely deploying LLMs requires well-calibrated estimates of the uncertainty in their outputs (Amodei et al., 2016; Hendrycks et al., 2021; Zhou et al., 2024). This is particularly challenging as LLMs are normally not trained to capture their own uncertainty and, due to their size and complexity, it is difficult and expensive to fine-tune or adapt them (Jiang et al., 2021; Kuhn et al., 2023; Osband et al., 2022). In addition, many state-of-the-art LLMs are proprietary and available only via APIs with their weights and architecture hidden from end users, leaving black-box integration as the only option (Katz et al., 2023; Ye et al., 2023a). This precludes the application of common model uncertainty estimation methods for deep learning, i.e., drop-out or deep ensembles (Zhou et al., 2022a; Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017).

To overcome these limitations, we propose Bayesian Prompts Ensembles (BayesPE), a simple yet effective approach for estimating the uncertainty of LLMs in black-box scenarios (i.e. without access to model weights). Similarly to Wightman et al. (2023) and Jiang et al. (2023b), we ensemble the output probabilities obtained by running the LLM classifier with different variations of the prompt task instructions. However, instead of simply combining all prompts with equal contribution, we learn their relative importance through approximate variational inference with a small validation set. Figure 1 schematically illustrates our approach. BayesPE adapts the use of different prompts to the task and LLM of choice, while obtaining well-calibrated estimates of the output uncertainty. We further show that BayesPE approximates a Bayesian input layer to the LLM and provides a lower bound to the reducible error of

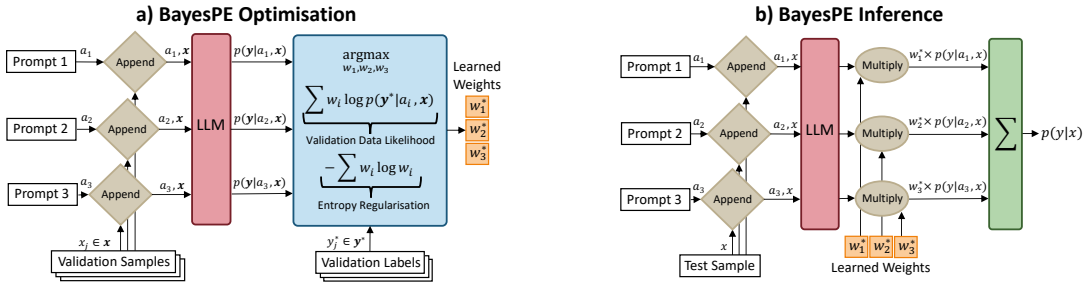

Figure 1: **(a) BayesPE optimisation:** Each validation text sample $x_j \in \mathbf{x}$ is appended to each of the semantically equivalent instruction prompts a_i. The LLM is used to compute the likelihood $p(y_j^*|a_i, x_j) \in p(\mathbf{y}^*|a_i, \mathbf{x})$ for each validation ground-truth label $y_j^* \in \mathbf{y}^*$. The returned likelihoods are then used to compute learned weights w_i^* through the BayesPE objective. **(b) BayesPE inference:** A new test sample x is appended to each prompt and all combinations are passed to the LLM. Each of the resulting class probability distributions $p(y|a_i, x)$ is weighted by the respective learned weight w_i^* and aggregated to obtain the final distribution $p(y|x)$.

the model. In our extensive empirical evaluation, we test BayesPE with several LLMs and a range of text classification tasks, both in zero- and few-shot scenarios. In each case, we measure several uncertainty estimation and calibration metrics and compare to existing competitive baselines.

2 Background & Related Work

Model Uncertainty in Deep Learning Developing deep neural networks (DNNs) that "know when they don't know" is a long standing challenge in deep learning (Amodei et al., 2016; Zhou et al., 2022a). Unlike other machine learning models, such as Gaussian processes (MacKay et al., 1998; Seeger, 2004; Damianou and Lawrence, 2013) and Markov chain Monte Carlo samplers (Andrieu et al., 2003), DNNs are not capable to self-assess output uncertainty originating from model misspecification, i.e., model or epistemic uncertainty, often resulting in wrong inferences with high confidence for out of distribution data (Gawlikowski et al., 2023). Conversely, Bayesian neural networks (BNNs) provide a principled way to capture this uncertainty by assigning distributions of values to their weights, instead of point estimates (MacKay, 1992; Goan and Fookes, 2020; Jospin et al., 2022). BNNs then estimate epistemic output uncertainty by sampling the weights' distributions at inference time.

While BNNs are arguably the most principled way to model uncertainty, they are difficult and often prohibitively expensive to train in practice (Graves, 2011; Blundell et al., 2015; Gal and Ghahramani, 2016; Wu et al., 2018). This motivated several approximations to BNNs to efficiently model epistemic uncertainty. Most successful approaches include drop-out, where variation in the weights is approximated by masking random hidden units (Gal and Ghahramani, 2016; Li and Gal, 2017), and deep ensembles, where uncertainty is estimated with multiple models trained on the same data with different random seeds (Lakshminarayanan et al., 2017; Hoffmann and Elster, 2021). In this work, we aim to approximate model uncertainty for black-box pre-trained LLMs. In this setting, uncertainty is intractable and we assume no access to the weights to modify or train the model.

Uncertainty Quantification for LLMs One way to estimate uncertainty for black-box LLMs is to use the probabilities of the generated tokens (Ott et al., 2018; Zhao et al., 2021; Mielke et al., 2022; Yu et al., 2022). Using relative tokens probabilities is proven to provide good calibration for several tasks (Jiang et al., 2021; Stengel-Eskin and Van Durme, 2022). Recent work also models the uncertainty of full sentences taking into account semantic equivalence. Malinin and Gales (2020) use input-output mutual information to capture uncertainty. Kuhn et al. (2023) compute distributions and uncertainty in a representation space specifically trained such that distances capture semantic similarity. Manakul et al. (2023) uses the LLM itself to evaluate similarity among multiple generations.

Using the LLM token probabilities is a reliable estimate of the uncertainty they learn from training data, i.e., data uncertainty. However, it does not necessarily capture model uncertainty, i.e., for inputs where the model is not competent (Osband et al., 2022). Arguably, model uncertainty is more critical to safety, as it assesses when we can trust the predictions of the LLM (Hüllermeier

and Waegeman, 2021). Analogously to approximate Bayesian approaches for deep learning, several studies propose sampling weights with dropout (Fomicheva et al., 2020; Wang et al., 2022). Other approaches build and train specific components instead, designing Bayesian transformer blocks (Sankararaman et al., 2022) or adding an uncertainty component to existing pre-trained LLMs and fine-tuning the whole architecture (Osband et al., 2022). While effective, these approaches need to have access to the model weights and are computationally expensive to apply at scale. A recently proposed approach for black-box scenarios is that of prompting LLMs themselves to generate estimates of their own confidence (Kadavath et al., 2022; Lin et al., 2022; Zhou et al., 2023). However, it is not trivial to construct prompts that result in accurate confidence assessments (Zhou et al., 2024).

Prompt Ensembles Recent works have explored aggregating prompts to improve predictive performance of LLM prompting (Bai et al., 2022; Shi et al., 2023). Similarly to our own work, Hou et al. (2023) assign different weights to different prompts in the ensemble and optimize them with a validation set. However, they train specific verbalizers for each prompt and their objective is based on classification accuracy maximization. In our own work, we perform direct prompting instead and prompts weights are optimized with a Bayesian objective, balancing data likelihood and uniform exploitation of the prompts for improved calibration. Prompts ensembles have also been explored to improve calibration, studying different strategies to generate diverse prompts for this purpose (Wightman et al., 2023; Jiang et al., 2023b). However, these works aggregate all prompts in the ensemble with equal importance. As LLMs are highly sensitive to prompt engineering (Jiang et al., 2020; White et al., 2023), this approach runs the risk of including low performing prompts in the ensemble and degrade overall performance. Conversely, we hypothesize that by learning the importance of different prompts with a Bayesian approach, we can strike a balance between prompt variation and data likelihood maximization, in analogy with the inference of BNN weight distributions given a prior and data.

Automatic Prompt Engineering Given the difficulty in prompt selection, recent work proposed to automatically generate them (Shin et al., 2020; Zhou et al., 2022b; Yang et al., 2023). While related to the scope of this work, our aim is in an orthogonal direction, as we focus on how to optimally use a series of pre-defined prompts for uncertainty estimation, rather than generating and optimizing a prompt for improving performance in a particular task. In fact, the two approaches are complementary (see Appendix C.3). Some works in the domain of vision-language models also propose Bayesian treatments of automatic prompt generation, balancing out the evidence from a validation set and regularizing priors (Derakhshani et al., 2023; Liu et al., 2023). However, these methods specifically apply to vision-language models, as they formulate the Bayesian combination of prior and posterior in the common text-image embedding space.

3 Bayesian Prompt Ensembling

Problem Description We consider the problem of text classification with a black-box decoder-only LLM via prompting. In a zero-shot setting, the text input (prompt) to the LLM can be separated into two parts: (i) task instructions a, e.g., "is the sentiment of the following review positive or negative?"; and (ii) test input x, e.g., a movie review in a test set. The prompt a is constant to all inputs for a given task, while test inputs x change, e.g., different reviews to be classified as positive or negative. The class probabilities $p(y|x, a)$ for the corresponding label y are inferred by passing the instruction-test data point pair $\{a, x\}$ as input to the LLM, e.g., "is the sentiment of the following review positive or negative? {review}", to obtain the probabilities for the next token or sequence that correspond to the particular classes in the vocabulary, e.g., 'positive' and 'negative'. In a few-shot setting, the task instructions can be accompanied with demonstrations (i.e. a small set of labeled data points).

We also assume that we have access to N semantically equivalent task instruction prompts $a_i \in \mathbf{a}$, and a small validation data set $\mathcal{D}$ of M data points $x_j \in \mathbf{x}$ with corresponding ground-truth labels $y_j^* \in \mathbf{y}^*$. Our main aim is to optimally exploit the LLM through prompting $p(y|x, a)$, the available labeled validation data $\mathcal{D}$ and the given batch of prompt instructions a to obtain well-calibrated classification probabilities $p(y|x, \mathcal{D})$. By well-calibrated, we mean that the LLM assigns, on average, the same confidence to its own prediction as the hidden ground-truth, i.e.,

$\mathbb{E}_{p(y|x,\mathcal{D})} \log p(y|x,\mathcal{D}) \simeq \log p(y^*|x,\mathcal{D})$ for new test input data $\{x, y^*\}$. This results in a model which produces high confidence when it generate the correct class token and low confidence when it is likely to generate the incorrect class token. Experimentally, we measure this property with a variety of established calibration metrics. In this paper, we do not aim to optimize the definition or automatic generation of the prompts $\mathbf{a}$. We assume these to be fixed and equally relevant a-priori. We focus on the problem of how to optimally employ them for well-calibrated inference. We study performance with different existing prompt generation strategies in Appendix C.3.

Modeling We model the batch of task instruction prompts $\mathbf{a}$ as a latent variable, resulting in the LLM having the form $p(y|x,\mathcal{D}) = \int p(a|\mathcal{D})p(y|x,a)da$. As the LLM is fixed, the conditional $p(y|x,a)$ is independent of the validation data $\mathcal{D}$. The posterior $p(a|\mathcal{D})$ is the desired distribution of task instructions a given $\mathcal{D}$ which is intractable to compute directly. We model the batch of prompts as samples from a prior distribution $a_i \sim p(a)$ of semantically equivalent task instruction prompts for the task of interest. We further assume that all prompts in the batch have approximately the same a-priori importance, i.e., $p(a_i) \approx C, \forall a_i$.

Approximate Variational Inference As the posterior of prompts given the validation data $p(a|\mathcal{D})$ is intractable to compute, we wish to find a distribution $q^*(a)$ to approximate it. We can formulate this task as a KL divergence minimization using variational inference (Graves, 2011; Gal et al., 2016):

$$q^*(a) = \underset{q(a)}{\operatorname{argmin}} \quad KL[q(a)|p(a|\mathcal{D})] = \\ \underset{q(a)}{\operatorname{argmax}} \quad \mathbb{E}_{q(a)}[\log p(\mathbf{y}^*|a,\mathbf{x})] - KL[q(a)|p(a)]. \tag{1}$$

The reformulation on the second line of Equation 1 is a standard result of variational inference, its derivation is detailed in Appendix A.1. Once $q^*(a)$ has been computed, the classification likelihood for a new input x can be estimated as:

$$p(y|x,\mathcal{D}) \simeq p(y|x) = \int q^*(a)p(y|x,a)da. \tag{2}$$

We argue that the classification likelihood can be interpreted as the addition of a Bayesian input layer, where $a \sim q^*(a)$ acts as a stochastic variable that transforms the LLM input x to $\{a, x\}$, i.e., the prompt including instructions and text input, by appending the sampled instruction prompt. Figure 2(a) illustrates this analogy.

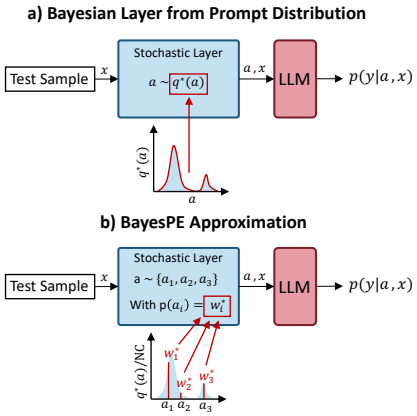

Figure 2: **(a) Prompt sampling as a Bayesian layer:** sampled instruction prompts $a \sim q^*(a)$ act as stochastic parameters of a transformation applied to a text input $x \rightarrow a, x$. This operation is analogous to a Bayesian input layer. **(b) BayesPE approximation:** We restrict sampling to the existing prompts $a_i \in \mathbf{a}$ and therefore only need the values of the density $q^*(a)$ at these points, which we re-parameterize in the weights $w_i^* \in \mathbf{w}^*$.

Fitting a parametric variational density $q(a)$ through the maximization of Equation 1 is challenging, as this distribution would be a generative language model itself with parameters to be trained; sampling a prompt $a \sim q(a)$ means generating text with defined properties. While this is certainly an interesting research direction for future work, we aim instead to exploit the set of available predefined prompts a_i to efficiently build the prompt ensemble. To this end, we estimate the likelihood of Equation 2 using importance sampling from the prior $p(a)$:

$$p(y|x) \approx \frac{1}{N}\sum_{i}^{N} \frac{q^*(a_i)}{C} p(y|a_i,x). \tag{3}$$

A full derivation is presented in Appendix A.2. Note that, with this importance sampling approximation, the likelihood computation only depends on the value of the variational density $q^*(a)$ at the N points $a = a_i$, i.e., the pre-defined prompts. Figure 2(b) illustrates this approximation.

Therefore, we only need to optimize the discrete values $q(a_i)$ in the variational inference of Equation 1. We re-parameterize these values as normalized weights $w_i = \frac{q(a_i)}{NC} \in \mathbf{w}$, with $w_i \in [0, 1]$ and

$\sum w_i = 1$ to re-formulate Equation 1 as:

$$\underset{\mathbf{w}}{\text{argmax}} \underbrace{\sum_{i}^{N} w_i \log p(\mathbf{y}^*|a_i, \mathbf{x})}_{\text{Validation Data Likelihood}} \underbrace{- \sum_{i}^{N} w_i \log w_i}_{\text{Entropy}} \tag{4}$$

A full derivation is presented in Appendix A.3. The BayesPE objective of Equation 4 consists of two terms. The first term encourages high average likelihood of validation data across the weighted prompts. The second term is the entropy of the weights $w_i \in \mathbf{w}$ and pushes them to be uniformly distributed. As $\log p(\mathbf{y}^*|a_i, \mathbf{x}) = \sum_j^M \log p(y_j^*|a_i, x_j)$, the relative importance of the validation likelihood term increases with M, i.e., the size of the validation data. In the limit of $M \to \infty$, the entropy term vanishes, leading to always sampling the prompt with the highest validation likelihood $a_{i=\text{best}}$ while the rest are discarded ($w^*_{i=\text{best}} = 1$, $w^*_{i\neq\text{best}} = 0$). Conversely, with no validation data, i.e., $M = 0$, only the entropy term remains, leading to uniformly sample task instruction prompts ($w_i^* = 1/N, \forall i$), which is equivalent to the standard ensembling approach proposed by Wightman et al. (2023) and Jiang et al. (2023b).

Once the BayesPE weights $w_i^* \in \mathbf{w}^*$ have been learned through the optimization of Equation 4, we can infer class probability distributions for new inputs x:

$$p(y|x) \approx \sum_{i}^{N} w_i^* p(y|a_i, x) \tag{5}$$

To sum up, we compute the final class probabilities $p(y|x, \mathcal{D})$ given a new test input x by first prompting the LLM N times, with each of the available instruction prompts $a_i \in \mathbf{a}$ to obtain conditional class probabilities $p(y|a_i, x)$. Second, we compute the weighted sum of the class probabilities using the weights $w_i^* \in \mathbf{w}^*$, learned through the optimization of Equation 4. BayesPE can also be interpreted as a Bayesian mixture of experts (Waterhouse et al., 1995), where weights are obtained through approximate MAP inference (details in Appendix A.4 and C.4).

Lower Bound on Model Error BayesPE provides a lower bound on the expected model error for the LLM, even when its weights are not known. Specifically, if prompts $a \sim q^*(a)$ are semantically equivalent to each other, the following inequality is valid:

$$\mathbb{E}_{q^*(a)}\mathbb{E}_{p(\theta|\mathcal{D}_T)} KL[p(y|x, \theta, a)||p(y|x, \theta^*, a)] \geq \mathbb{E}_{p(\theta|\mathcal{D}_T)}\mathbb{E}_{q^*(a)} KL[p(y|x, \theta, a)||p(y|x, \theta)] \tag{6}$$

A full derivation is presented in Appendix A.5. In the notation above, we have added the weights θ of the pre-trained LLM which are not available in black-box settings. $p(\theta|\mathcal{D}_T)$ is the posterior of these weights, given the data set that was used to train the LLM $\mathcal{D}_\mathcal{T}$. We assume the pre-trained LLM weights to be a sample from this posterior $\theta \sim p(\theta|\mathcal{D}_T)$. θ^* are the optimal weights for the LLM, for which its reducible model error is null. Equation 6 states that the expected divergence between the output given any particular prompt $p(y|x, \theta, a)$ and the aggregate output $p(y|x, \theta)$, i.e., the uncertainty with respect to the input prompts, is a lower bound to the expected divergence between the given LLM $p(y|x, \theta, a)$ and the LLM with optimal weights $p(y|x, \theta^*, a)$, i.e., the expected reducible model error. This bound indicates that BayesPE approximates the reducible model uncertainty, as opposed to the data uncertainty. Bayesian neural networks provide an analogous bound on the expected model error (Appendix A.6).

4 Experimental Setup

Data: We evaluate BayesPE on a total of ten tasks. Three sentiment analysis tasks: Amazon Reviews (He and McAuley, 2016), Imdb (Pal et al., 2020) and SST-2 (Socher et al., 2013). Three topic modeling tasks: DBPedia 14, Yahoo Answers (Zhang et al., 2015) and TREC (Hovy et al., 2001). Two semantic relation tasks: MRPC (Warstadt et al., 2018) and SNLI (Bowman et al., 2015). Finally, two spam detection tasks: SMS (Almeida et al., 2011) and YouTube (Alberto et al., 2015). These data sets cover a variety of common NLP tasks, number of classes and text input lengths (details in Appendix B.1).

Models: We evaluate the BayesPE framework with five different pre-trained and instruction fine-tuned LLMs, spanning families and sizes: Falcon-7b-Instruct (Penedo et al., 2023), Falcon-40b-Instruct (Almazrouei et al., 2023), MPT-7b-Instruct (Team, 2023b), MPT-30b-Instruct (Team, 2023a) and Mistral-7b-Instruct (Jiang et al., 2023a). More details in Appendix B.2.

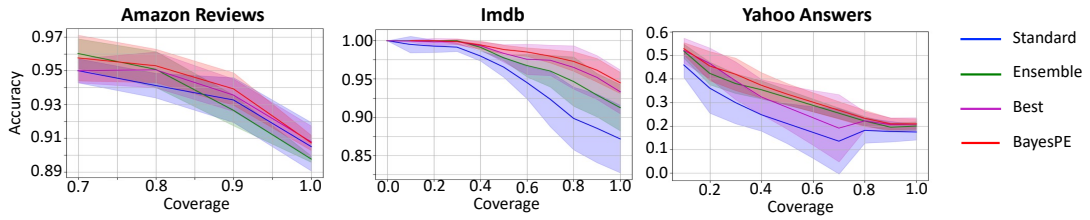

Figure 3: Calibration curves for few-shot classification with MPT-30b-Instruct using BayesPE and competing baselines. A threshold on the largest class probability is varied from zero to one. For each threshold value, coverage and accuracy are calculated and plotted against each other.

Baselines: We compare BayesPE to the following baselines:[1]

- **Standard:** The LLM is run once with a single prompt, drawn at random $a_i \sim \mathbf{a}$.
- **Ensemble:** Class probabilities from all instructions prompts a_i are summed together (Wightman et al., 2023; Jiang et al., 2023b).
- **Best:** The prompt a_i with the highest validation accuracy is chosen.

Uncertainty calibration metrics: We compute five metrics: negative log-likelihood (NLL), ROC-AUC score (AUC), expected calibration error (ECE), maximum calibration error (MCE) and Brier score (Brier) (Gawlikowski et al., 2023; Abdar et al., 2021).

Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)	F1 (↑)
Amazon Reviews						
Standard	0.356	0.981	0.069	**0.307**	0.083	**0.926**
Ensemble	0.294	**0.986**	0.061	**0.290**	0.076	**0.926**
Best	**0.292**	0.982	**0.056**	**0.292**	**0.070**	**0.927**
BayesPE	**0.267**	**0.986**	**0.056**	**0.356**	**0.070**	**0.926**
Imdb						
Standard	2.112	0.966	0.375	0.536	0.374	0.921
Ensemble	1.425	0.972	0.355	0.677	0.341	0.925
Best	**1.356**	**0.973**	**0.277**	**0.342**	**0.289**	0.925
BayesPE	**1.203**	**0.974**	**0.260**	**0.333**	**0.277**	**0.935**
Yahoo Answers						
Standard	4.024	0.687	0.462	0.527	0.110	0.300
Ensemble	**3.279**	**0.700**	**0.328**	**0.391**	**0.098**	**0.333**
Best	4.512	0.683	0.450	0.494	0.112	**0.332**
BayesPE	**3.223**	0.690	**0.271**	**0.410**	**0.096**	**0.332**

Table 1: Few-shot classification with MPT-30b-Instruct using BayesPE and competing baselines. Metric values which are best within statistical significance (two-sample p $>$ 0.05) are printed in bold.

[1]In early experiments, we also tested confidence self-assessing, following Kadavath et al. (2022), but found the probabilities to be consistently less calibrated than the tokens probabilities of the class words themselves (Appendix B.4).

Prompts and Validation Data: We consider a fixed set of N initially provided prompts a_i for each task. We do not aim to investigate prompt definition or engineering and, unless stated otherwise, we simply draw these at random from a pool of prompts generated with GPT-3.5-turbo (Ye et al., 2023b) with no manual editing. The prompts are used to compute Ensemble, Best and BayesPE, while Standard is computed by drawing just one prompt at random. For few-shot experiments, the prompts include five labeled examples from a different pool of validation data. These are drawn at random for each prompt in the set (details in Appendix B.3).

In each task, unless stated otherwise, we use $N = 5$ initial prompts, 50 validation examples to select the prompt for Best and train the BayesPE weights (details in Appendix B.4), and 200 test examples to compute metrics. The set of prompts and validation examples are drawn at random ten times in each of our experiments to obtain error bars and compute statistical significance of results.

Method	AUC (↑)	ECE (↓)	MCE (↓)
Ensemble	-3.9%	-20.0%	-16.6%
Best	+15.8%	-26.3%	-17.0%
BayesPE	**+23.7%**	**-32.1%**	**-23.3%**

Table 2: Average relative difference in AUC score, ECE and MCE that each methods provided compared to standard prompting over all 100 experiments in our evaluation.

Method	NLL	AUC	ECE	MCE	Brier
Standard	28(0)	48(0)	42(1)	53(3)	27(0)
Ensemble	37(1)	66(10)	55(3)	39 (1)	36(2)
Best	55(3)	63(7)	62(3)	73(1)	65(2)
BayesPE	**96(25)**	**72(4)**	**83(16)**	**75(7)**	**96(17)**

Table 3: Number of times (out of 100 experiments) each method was the best performing or not statistical significantly different to the best method for each metric. Parenthesis denotes how many times a method was found to be significantly better than any other.

5 Results

Uncertainty Calibration We evaluate performance of BayesPE and baselines on all tasks with all LLMs in zero- and few-shot settings. Table 1 shows detailed results for three of the ten tested tasks using the overall highest performing configuration, i.e., MPT-30b-Instruct with few-shot prompting. While we are primarily interested in uncertainty quantification, we also report macro F1-Score, to verify that predictive performance is not hindered by our approach. Full results are reported in Appendix C.5. Figure 3 shows calibration curves (accuracy vs. coverage) for the same three experiments.

In Table 1, we observe that BayesPE is competitive or better (within statistical significance) across all tasks and all metrics, with the exception of AUC score in Yahoo Answers, for which it is only 0.01 lower than the best baseline. The calibration curves of figure 3 also show BayesPE clearly outperforming competing baselines. For example, on the Imdb data set, BayesPE achieves 97% accuracy at 80% coverage, while competing methods need to reduce coverage to $45 - 55\%$ to achieve the same. We also see that, among competing baselines, the best approach varies with the task (Ensemble for Yahoo Answers and Best for Amazon Reviews and Imdb), meaning no single best baseline can be selected a-priori. Contrarily, BayesPE is always competitive, making it a far more reliable method. We further note that BayesPE often gives a substantial improvement in calibration (NLL and ECE) compared to standard prompting, e.g., NLL of 3.2 vs. 4.0 and ECE of 0.27 vs. 0.46 for the Yahoo Answers data set. Our approach is also competitive or best across the three tasks also in F1-score, meaning that it maintains or improves predictive performance as well.

The trends described above are observed across all experiments. While inspecting all results in detail is difficult (included in Appendix C.5), we present aggregate results, both in terms of average differences in calibration and statistical significance. Table 2 reports the average difference in AUC score, ECE and MCE each method provided across all 100 experiments (ten tasks $\times$ and five LLMs in $\times$ zero- and few-shot settings), compared to standard prompting[2]. BayesPE provides substantial improvement in calibration, with appreciably larger reductions in errors than Ensemble and Best, e.g., -32% in ECE vs. -20-26%.

Table 3 reports in how many of our 100 experiments each method was found to be the best, or within statistical significance of the best (two-sample p> 0.05 across ten repeats of each experiment). The number in parenthesis reports in how many experiments each method was significantly better than any other (two-sample p< 0.05 with second best). Overall, BayesPE is significantly better or competitive for every metric in the majority of experiments and in more experiments than any competing baseline. In particular, for metrics tied to expected calibration performance (NLL, ECE and Brier) it is substantially more consistent than the baselines, giving best or competitive NLL and Brier score in 96% of the experiments and ECE in 83% of the experiments. In comparison, other baselines were found competitive for these metrics in only $30 - 65\%$ of the experiments.

Inference Efficiency The driving factor for the run-time cost of prompting an LLM is the number of forward passes needed to obtain class probabilities (inference). With our approach, after finding the optimal weights $\mathbf{w}^*$ for a set of N ensembles through a validation set, we can fix a forward passes budget at run-time $L <= N$. We apply a simple greedy approach to select the run-time prompts and their weights by simply taking the L prompts with the highest weights w_i^* to construct the weighted ensemble. Using three data sets and Mistral-7b-Instruct as the base LLM in few-shot operation, we evaluate NLL with respect to the number of run-time forward passes. We use $N = 20$ initial prompts to compute the BayesPE weights and the best prompt (Best) baseline with a validation set and then test with a varying budget of $L = 1 : 10$ prompts at run-time. Experimental details are given in Appendix B.5. Figure 4 shows the results (additional results in supplementary C.2).

Ensemble and BayesPE display a common trend of decreasing NLL as the forward passes budget is increased. This is expected, as in both cases different prompts are increasingly added to the ensemble at increasing LLM forward passes, which results in increasingly better calibration of the final class probabilities. BayesPE is consistently better than or competitive with Ensemble for all experimental settings and all inference budgets. Even for

[2]NLL and Brier score are not comparable across experiments, as their values depend on the number of classes and is not bounded. Therefore, differences are meaningful only within the same experiment and not aggregated.

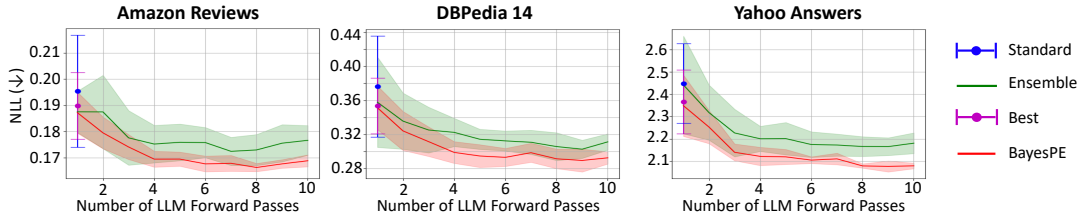

Figure 4: Negative log likelihood (NLL) vs. number of forward passes through the LLM at run-time. Using any prompt (Standard) and running the best performing prompt on validation (Best) can be performed only with a single prompt and hence a single forward pass. Standard ensembling (Ensemble) and our BayesPE approach exploit more prompts as the forward passes budget is increased.

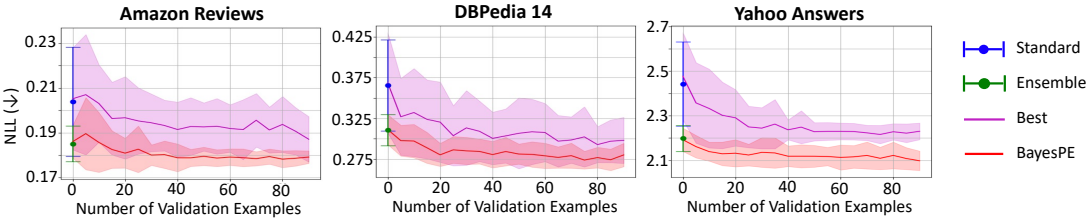

Figure 5: Negative log likelihood (NLL) vs. number of labeled validation examples. Using any prompt (Standard) and aggregating all prompts equally (Ensemble) does not require any validation set and are shown as single points at zero validation examples. Choosing the highest accuracy prompt on validation (Best) and BayesPE benefit from increased number of validation examples.

tasks such as Amazon Reviews in few-shot, where performance with a single prompt is comparable across methods (NLL$\sim$ 0.189), as the budget is increased, BayesPE results in a substantially better improvement in calibration (NLL $\sim$ 0.169 vs. 178 for Ensembles). This means that BayesPE is a more efficient and effective method of obtaining well-calibrated class probabilities with the LLM at any run-time costs constrain. Note that, at a budget of $L = 1$, BayesPE is equivalent to picking the best performing prompt on validation (Best). As the budget is increased, BayesPE consistently improves performance (lower NLL). This means that, compared to picking the best prompt on validation, BayesPE consistently utilizes additional computational spend to improve calibration performance, e.g. NLL $\sim$ 0.169 on the Amazon Reviews task for BayesPE with a budget of 5 forward passes, compared to an NLL of 1.89 for picking the best prompt on validation.

Labeled Data Efficiency The second substantial cost factor in prompting LLMs with BayesPE is the number of labeled examples needed in the validation set. We study NLL as a function of the number of validation examples used to estimate LLM uncertainty. As in the experiments above, we use three data sets and Mistral-7b-Instruct as the base LLM in few-shot operation. Experimental details are given in Appendix B.5. Figure 5 shows the results (additional results in supplementary C.2).

As expected, we observe that with no validation examples, BayesPE is equivalent to Ensemble, as the likelihood term in Equation 4 is null. BayesPE is competitive or better than any baselines for any budget of validation labeled data, e.g., for DBPedia 14 in Zero-Shot we report an NLL of 0.28 for BayesPE with 50 validation examples vs. 0.31 for for both Best and Ensemble. With the Amazon Reviews task, adding five validation examples initially causes to over-fit and degrade performance. However, as the validation data set increases to tens of examples, this over-fitting is mitigated and the BayesPE NLL matches or improves on that of the Ensemble baseline. BayesPE is also consistently better than the Best baseline in all experiments at all values of labeled data budget, e.g., NLL of 2.13 vs. 2.3 with 20 validation examples and 2.1 vs. 2.2 with 90 validation examples for Yahoo Answers. This means that BayesPE provides consistently better uncertainty quantification, independently of the available amount of validation examples.

6 Conclusion

We proposed Bayesian Prompt Ensembles (BayesPE), an effective method to approximately estimate the model uncertainty for classification

with black-box LLMs. We showed that BayesPE constitutes a discrete approximation to a Bayesian input layer and that its uncertainty gives a lower bound on the model error, analogous to that of BNNs. BayesPE is significantly better or comparable to competitive baselines across the majority of experiments. In our analysis, we showed how BayesPE improves uncertainty quantification over baselines as both the inference and labeled data budget increase.

Limitations

Semantic Equivalence Assumption: To develop the theory of BayesPE, we make the assumption that starting prompt are semantically equivalent. This is challenging to ensure in practice, and even to define. For the purpose of prompting, we formulate this property as stated in Equation 7, but as this definition involves a model with no reducible error, it is impossible to ensure in practice with generated prompts. In our experiments, prompts obtained with automatic rephrasing (a-priori or with validation using APE) proved to be a good enough approximation to obtain good calibration from BayesPE. However, better defining semantic equivalence, testing it and ensuring it for input prompts remains to be studied.

Classification Tasks Only: In our extensive experimental benchmark, we evaluate and compare BayesPE for several classification tasks using LLM prompting, but not with text generation. This is because of computational cost of experiments and complexity of comparing with ground-truth. With classification tasks, we can measure several well-established calibration and uncertainty metrics that can be readily computed given the output class probability, while in text generation these measures are only recently emerging and are much more computationally expensive to compute and with high variance, as they inevitably involve repeated sampling.

Access to Words Probabilities: In the evaluations presented in this paper we use different LLMs in black-box implementation, i.e, with no access to the weights, but we do use words probabilities to build classification distributions. However, some proprietary LLMs, e.g., GPT-4, do not even make these available and only return text answers to input prompts. BayesPE and other ensemble methods can still be applied, but require sampling multiple times, extracting classes from the resulting text outputs and aggregating the results to build output probabilities for each prompt. This is more computationally expensive and less robust than using directly class words probabilities when available.

Prompt Ensemble Generation: In this work, we focus on how to optimally exploit a set of prompts assumed to be semantically equivalent, but we do not propose a specific method to generate suitable prompts to build this ensemble. This is an important problem to be investigated in future work, as prompts automatically generated in the different ways we consider in this paper may introduce biases and be overall sub-optimal for the downstream task.

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297.

Túlio C Alberto, Johannes V Lochter, and Tiago A Almeida. 2015. Tubespam: Comment spam filtering on youtube. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 138–143. IEEE.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Tiago A. Almeida, Jose Maria Gomez Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: New collection and results. In *Proceedings of the 2011 ACM Symposium on Document Engineering (DOCENG'11)*.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. 2003. An introduction to mcmc for machine learning. *Machine learning*, 50:5–43.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional

ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Andreas Damianou and Neil D Lawrence. 2013. Deep gaussian processes. In *Artificial intelligence and statistics*, pages 207–215. PMLR.

Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor G Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. 2023. Bayesian prompt learning for image-language model generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15237–15246.

Giancarlo Ferrari-Trecate, Christopher Williams, and Manfred Opper. 1998. Finite-dimensional approximation of gaussian processes. *Advances in neural information processing systems*, 11.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Yarin Gal et al. 2016. *Uncertainty in deep learning*. PhD thesis, University of Cambridge.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589.

Ethan Goan and Clinton Fookes. 2020. Bayesian neural networks: An introduction and survey. *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, pages 45–87.

Alex Graves. 2011. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. 2021. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*.

Lara Hoffmann and Clemens Elster. 2021. Deep ensembles from a bayesian perspective. *arXiv preprint arXiv:2105.13283*.

Bairu Hou, Joe O'connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023. Promptboosting: Black-box text classification with ten forward passes. In *International Conference on Machine Learning*, pages 13309–13324. PMLR.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mingjian Jiang, Yangjun Ruan, Sicong Huang, Saifei Liao, Silviu Pitis, Roger Baker Grosse, and Jimmy Ba. 2023b. Calibrating language models via augmented prompt ensembles.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. *Transactions of the Association for Computational Linguistics*, 8:423–438.

Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. 2022. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma,

Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. 2023. Gpt-4 passes the bar exam. *Available at SSRN 4389233*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

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

Yingzhen Li and Yarin Gal. 2017. Dropout inference in bayesian neural networks with alpha-divergences. In *International conference on machine learning*, pages 2052–2061. PMLR.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Xinyang Liu, Dongsheng Wang, Miaoge Li, Zhibin Duan, Yishi Xu, Bo Chen, and Mingyuan Zhou. 2023. Patch-token aligned bayesian prompt learning for vision-language models. *arXiv preprint arXiv:2303.09100*.

David JC MacKay. 1992. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472.

David JC MacKay et al. 1998. Introduction to gaussian processes. *NATO ASI series F computer and systems sciences*, 168:133–166.

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

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Toby J Mitchell and John J Beauchamp. 1988. Bayesian variable selection in linear regression. *Journal of the american statistical association*, 83(404):1023–1032.

Ian Osband, Seyed Mohammad Asghari, Benjamin Van Roy, Nat McAleese, John Aslanides, and Geoffrey Irving. 2022. Fine-tuning language models via epistemic neural networks. *arXiv preprint arXiv:2211.01568*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Aditya Pal, Abhilash Barigidad, and Abhijit Mustafi. 2020. Imdb movie reviews dataset.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Karthik Abinav Sankararaman, Sinong Wang, and Han Fang. 2022. Bayesformer: Transformer with uncertainty estimation. *arXiv preprint arXiv:2206.00826*.

Matthias Seeger. 2004. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Elias Stengel-Eskin and Benjamin Van Durme. 2022. Calibrated interpretation: Confidence estimation in semantic parsing. *arXiv preprint arXiv:2211.07443*.

MosaicML NLP Team. 2023a. Introducing mpt-30b: Raising the bar for open-source foundation models (www.mosaicml.com/blog/mpt-30b).

MosaicML NLP Team. 2023b. Introducing mpt-7b: A new standard for open-source, commercially usable llms (www.mosaicml.com/blog/mpt-7b).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yuxia Wang, Daniel Beck, Timothy Baldwin, and Karin Verspoor. 2022. Uncertainty estimation and reduction of pre-trained models for text regression. *Transactions of the Association for Computational Linguistics*, 10:680–696.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Steve Waterhouse, David MacKay, and Anthony Robinson. 1995. Bayesian methods for mixtures of experts. *Advances in neural information processing systems*, 8.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Gwenyth Portillo Wightman, Alexandra DeLucia, and Mark Dredze. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 326–362.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E Turner, Jose Miguel Hernandez-Lobato, and Alexander L Gaunt. 2018. Deterministic variational inference for robust bayesian neural networks. *arXiv preprint arXiv:1810.03958*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023a. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023b. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Yue Yu, Rongzhi Zhang, Ran Xu, Jieyu Zhang, Jiaming Shen, and Chao Zhang. 2022. Cold-start data selection for few-shot language model fine-tuning: A prompt-based uncertainty propagation approach. *arXiv preprint arXiv:2209.06995*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Kaitlyn Zhou, Jena D Hwang, Xiang Ren, and Maarten Sap. 2024. Relying on the unreliable: The impact of language models' reluctance to express uncertainty. *arXiv preprint arXiv:2401.06730*.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: Expressions of overconfidence and uncertainty in language models. *arXiv preprint arXiv:2302.13439*.

Xinlei Zhou, Han Liu, Farhad Pourpanah, Tieyong Zeng, and Xizhao Wang. 2022a. A survey on epistemic (model) uncertainty in supervised learning: Recent advances and applications. *Neurocomputing*, 489:449–465.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022b. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

A Detailed Derivations

A.1 Derivation of Variational Inference for prompts distribution

We derive here the variational inference maximisation formula of equation 1. Starting from the KL divergence to be minimised:

$$
\begin{aligned}
&KL[q(a)|p(a|\mathcal{D})] = KL[q(a)|p(a|\mathbf{x}, \mathbf{y}^*)] = \\
&\int q(a) \log \frac{q(a)}{p(a|\mathbf{x}, \mathbf{y}^*)} da = \\
&\int q(a) \log \frac{q(a) p(\mathbf{y}^*)}{p(\mathbf{y}^*|a, \mathbf{x}) p(a)} da = \\
&\int q(a) \log \frac{q(a)}{p(a)} da + \log p(\mathbf{y}^*) \\
&- \int q(a) \log p(\mathbf{y}^*|a, \mathbf{x}) da.
\end{aligned}
$$

Because the log probability $\log p(\mathbf{y}^*)$ does not depend on $q(a)$, we can re-formulate the minimisation as:

$$
\begin{aligned}
&\underset{q(a)}{\operatorname{argmin}} \int q(a) \log \frac{q(a)}{p(a)} da + \log p(\mathbf{y}^*) \\
&\qquad - \int q(a) \log p(\mathbf{y}^*|a, \mathbf{x}) da = \\
&\underset{q(a)}{\operatorname{argmax}} - \int q(a) \log \frac{q(a)}{p(a)} da \\
&\qquad + \int q(a) \log p(\mathbf{y}^*|a, \mathbf{x}) da = \\
&\underset{q(a)}{\operatorname{argmax}} \quad \mathbb{E}_{q(a)}[\log p(y|a, x)] - KL[q(a)|p(a)]
\end{aligned}
$$

A.2 Derivation of Importance Sampling Approximation of Classification Likelihood

We derive here the importance sampling approximation of equation 3. After performing variational inference optimisation for the distribution $q(a)$ in equation 1, obtaining the optimised $q^*(a)$, the final classification likelihood is approximated as folows:

$$
\begin{aligned}
p(y|x, \mathcal{D}) &= \int p(a|\mathcal{D}) p(y|x, a) da \\
&\simeq \int q^*(a) p(y|x, a) da.
\end{aligned}
$$

We can now perform importance sampling with the prior $p(a)$:

$$
\begin{aligned}
\int q^*(a) p(y|x, a) da &= \int p(a) \frac{q^*(a)}{p(a)} p(y|x, a) da \\
&\approx \frac{1}{N} \sum_i^N \frac{q^*(a_i)}{p(a_i)} p(y|x, a_i) \\
&\approx \frac{1}{N} \sum_i^N \frac{q^*(a_i)}{C} p(y|x, a_i)
\end{aligned}
$$

The last line follows as all propmts are assumed to have the same importance a-priori, i.e., $p(a_i) \approx C$ $\forall a_i$.

A.3 Derivation of Importance Sampling Approximation of Variational Inference Objective

We derive here the importance sampling approximation of equation 4. Starting from the variational inference objective function:

$$
\begin{aligned}
&\mathbb{E}_{q(a)}[\log p(\mathbf{y}^*|a, \mathbf{x})] - KL[q(a)|p(a)] \\
&= \int q(a) \log p(\mathbf{y}^*|a, \mathbf{x}) da \\
&- \int q(a) \log \frac{q(a)}{p(a)} da \\
&= \int p(a) \frac{q(a)}{p(a)} \log p(\mathbf{y}^*|a, \mathbf{x}) da \\
&- \int p(a) \frac{q(a)}{p(a)} \log \frac{q(a)}{p(a)} da \\
&\approx \sum_i^N \frac{q(a_i)}{p(a_i)} \log p(\mathbf{y}^*|a_i, \mathbf{x}) \\
&- \sum_i^N \frac{q(a_i)}{p(a_i)} \log \frac{q(a_i)}{p(a_i)} da. \\
&\approx \frac{1}{N} \sum_i^N \frac{q(a_i)}{C} \log p(\mathbf{y}^*|a_i, \mathbf{x}) \\
&- \frac{1}{N} \sum_i^N \frac{q(a_i)}{C} \log \frac{q(a_i)}{C}.
\end{aligned}
$$

The last line follows as all propmts are assumed to have the same importance a-priori, i.e., $p(a_i) \approx C$ $\forall a_i$. As the likelihood estimation of equation 3 only depends on $q(a_i)$, we only need to optimise these discrete values with variational inference, as we only care about the density of our variational

distribution $q(a)$ at these points:

$$\underset{q(a_i),\forall a_i \in \mathbf{a}}{\operatorname{argmax}} \quad \frac{1}{N}\sum_i^N \frac{q(a_i)}{C}\log p(\mathbf{y}^*|a_i,\mathbf{x}) - \frac{1}{N}\sum_i^N \frac{q(a_i)}{C}\log\frac{q(a_i)}{C}.$$

We now re-parametrise the density evaluations to be optimised $q(a_i)$ with the normalised weights $w_i = q(a_i)/NC \in \mathbf{w}$, with $w_i \in [0,1]$ and $\sum w_i = 1$, obtaining the BayesPE optimisation:

$$\underset{\mathbf{w}}{\operatorname{argmax}} \sum_i^N w_i \log p(\mathbf{y}^*|a_i,\mathbf{x}) - \sum_i^N w_i \log w_i$$

A.4 Relation to Bayesian Mixture of Experts

BayesPE can be viewed as a mixture of experts (MoE) with weights obtained through a lower bound approximation of MAP inference with an entropy maximizing prior over the weights. In the Bayesian MoE formulation, a target function $f(x)$, in our case the target predictive distribution $f(x) = p(y|x)$, is built with a finite sum of basis functions $\phi_i(x)$:

$$f(x) = \sum_i^N w_i \phi_i(x).$$

In BayesPE, the basis functions correspond to the predictive probabilities obtained by prompting with different instructions prompts a_i, i.e., $\phi_i(x) = p(y|a_i,x)$, recovering:

$$p(y|x) = \sum_i^N w_i p(y|a_i,x),$$

which is equivalent to equation 5. In Bayesian MoE, the prior over the weights w_i is typically assumed to be a Gaussian. There is generally different ways in which to model the posterior distribution of the weights. Arguably the simplest and more closely related to BayesPE is Bayesian linear regression, where a fixed posterior of weights $p(w|\mathbf{x},\mathbf{y}^*)$ for the linear combination of the basis function ($p(y|a_i,x)$ in our LLM prompting scenario) is learned with the validation data $\mathbf{x}$ and $\mathbf{y}^*$. Under the Gaussian prior and likelihood assumption, the posterior is also Gaussian and tractable to compute through matrix inversion (Mitchell and Beauchamp, 1988).

Under this Bayesian MoE perspective, BayesPE models the weights with a maximal entropy prior $p(\mathbf{w}) = \prod w_i^{-w_i}$ s.t. $\sum w_i = 1$ and $w_i \in [0,1]$, instead of a Gaussian. This results in an intractable posterior and the objective of BayesPE is equivalent to performing a lower bound approximation of MAP inference of this posterior on the validation set of observed data:

$$\begin{aligned}
&\operatorname{argmax}_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{x},\mathbf{y}^*) \\
&\quad \propto \log[p(\mathbf{y}^*|\mathbf{w},\mathbf{x})p(\mathbf{w})] \\
&\quad = \log[\sum_i^N w_i p(\mathbf{y}^*|a_i,\mathbf{x})] + \log[\prod_i^N w_i^{-w_i}] \\
&\quad \geq \sum_i^N w_i \log p(\mathbf{y}^*|a_i,\mathbf{x}) - \sum_i^N w_i \log w_i.
\end{aligned}$$

The last inequality is obtained through Jensen's inequality, meaning that the objective of BayesPE (equation 4) is a lower bound on the MAP objective in the Bayesian MoE formulation. We experimentally compare BayesPE to Bayesian linear regression in Appendix C.4.

There exist more advanced methods to model the weights in Bayesian MoEs that could be applied to prompt ensembles. One interesting possibility is modeling the ensemble as a finite dimensional Gaussian Processes (Ferrari-Trecate et al., 1998; Waterhouse et al., 1995). The MoE is modeled as a Gaussian process with Gaussian prior over the weights $\mathbf{w}$ and hyper-parameters of the kernel are learned with a validation data set. At inference time, the learned kernel can be used to infer the particular weights combination for the current input. Applying this Gaussian process modeling approach for LLM prompt ensemble is an interesting direction for future research. On one hand, we can expect some advantage from the adaptive nature of the weights at inference time through the kernel, as opposed to the fixed weights learned by BayesPE. On the other, having to compute the covariance matrix at inference time implies running the LLM with all prompts for the new example, precluding the possibility to adjust inference budget at test time (as we do with BayesPE in the experiments of figure 4).

A.5 Derivation of Model Error Lower Bound for Prompts Ensembles

We derive here the inequality of equation 6. First, we expand the definition of our LLM of choice,

used as a classifier $p(y|a,x)$ to include a specification of values for its weights θ; $p(y|a,\theta,x)$. We assume no knowledge of the specific values of θ, i.e., black-box scenario, but, as the model was arbitrarily pre-trained with some training data set $\mathcal{D}_T$, we assume these weights to be sampled from the posterior $p(\theta|\mathcal{D}_T)$.

Second, we define optimal weights θ^* which specify an optimal version of the LLM $p(y|a,\theta^*,x)$ which has no reducible error. We can think of this model as a version of the LLM trained with infinite data, leading to vanishing model error. With these definitions, we now formalise the semantic equivalence property of $p(a)$. Specifically, we state:

$$\begin{aligned} &p(y|x,\theta^*,a) = p(y|x,\theta^*,a') \quad \forall a,a' \sim p(a) \\ &\implies p(y|x,\theta^*) = p(y|x,\theta^*,a) \quad \forall a \sim p(a). \end{aligned} \tag{7}$$

This means that we assume the distribution $p(a)$ to capture semantic equivalence, if the optimal model $p(y|x,\theta^*,a)$ is invariant to task instructions prompts a drawn from it. This is because the optimal model $p(y|x,\theta^*,a)$ is assumed to be trained with infinite data, including all possible semantically equivalent inputs for a given output. In the limit of the BayesPE approximation, this property transfers to the learned variational distribution $q^*(a)$, as we sample it only at prompts $a_i \sim p(a)$. With these definitions in place, we can now derive the inequality of equation 6:

$$\begin{aligned} &\mathbb{E}_{q^*(a)}\mathbb{E}_{p(\theta|\mathcal{D}_T)}KL[p(y|x,\theta,a)||p(y|x,\theta^*,a)] = \\ &\int q^*(a)\int p(\theta|\mathcal{D}_T)\int p(y|x,\theta,a)\cdot \\ &\log\frac{p(y|x,\theta,a)}{p(y|x,\theta^*,a)}d\theta da dy = \\ &\int q^*(a)\int p(\theta|\mathcal{D}_T)\cdot \\ &\int p(y|x,\theta,a)\log\frac{p(y|x,\theta,a)p(y|x,\theta)}{p(y|x,\theta^*,a)p(y|x,\theta)}d\theta da dy = \\ &\int p(\theta|\mathcal{D}_T)\int q^*(a)\int p(y|x,\theta,a)\cdot \\ &\log\frac{p(y|x,\theta,a)}{p(y|x,\theta)}da d\theta dy + \\ &\int p(\theta|\mathcal{D}_T)\int q^*(a)\int p(y|x,\theta,a)\cdot \\ &\log\frac{p(y|x,\theta)}{p(y|x,\theta^*,a)}da d\theta dy \end{aligned}$$

Following the condition of equation 7, we can set $p(y|x,\theta^*,a) = p(y|x,\theta^*)$. Replacing this term in the second component on the last line above results in:

$$\begin{aligned} &\int p(\theta|\mathcal{D}_T)\int q^*(a)\int p(y|x,\theta,a)\cdot \\ &\log\frac{p(y|x,\theta,a)}{p(y|x,\theta)}da d\theta dy + \\ &\int p(\theta|\mathcal{D}_T)\int\int q^*(a)p(y|x,\theta,a)da\cdot \\ &\log\frac{p(y|x,\theta)}{p(y|x,\theta^*)}d\theta dy = \\ &\int p(\theta|\mathcal{D}_T)\int q^*(a)\int p(y|x,\theta,a)\cdot \\ &\log\frac{p(y|x,\theta,a)}{p(y|x,\theta)}da d\theta dy + \\ &\int p(\theta|\mathcal{D}_T)\int p(y|x,\theta)\log\frac{p(y|x,\theta)}{p(y|x,\theta^*)}d\theta dy = \\ &\mathbb{E}_{p(a|a_0)}\mathbb{E}_{q(\theta)}KL[p(y|x,\theta,a)||p(y|x,\theta^*,a)] = \\ &\mathbb{E}_{p(\theta|\mathcal{D}_T)}\big[\mathbb{E}_{q^*(a)}KL[p(y|x,\theta,a)||p(y|x,\theta)] \\ &+ KL[p(y|x,\theta)||p(y|x,\theta^*)]\big] \end{aligned}$$

The inequality of equation 6 follows as the second KL divergence is non-negative. The lower bound of equation 6 means that the uncertainty from aggregating output probabilities from different semantically equivalent prompts, as defined in 7, is a lower bound to the expected model error, defined as the KL divergence between the LLM output probability and the output probability of the optimal LLM with no model error (with weights θ^*). This property actually applies to any method ensembling semantically equivalent prompts, including Wightman et al. (2023) and Jiang et al. (2023b). Therefore, this theoretical finding validates prompt ensemble approaches in general as LLM model error approximators. This lower bound generalizes a property which is intuitive to understand with two prompts; If I know the answer to two prompts must be the same (semantic equivalence condition) and I obtain two different answers, at least one of the two answers is wrong. Equation 6 generalizes this to continuous class probabilities and any number of prompts.

A.6 Derivation of Model Error Lower Bound for Bayesian NNs

The lower bound of equation 6 is analogous to the bound on model error we get with Bayesian neural networks. Consider a Bayesian neural network, with posterior weights distribution $p(\theta|\mathcal{D}_T)$ and optimal weights θ^*, for which the resulting model

$p(y|x, \theta^*)$ has no reducible error.

$$\mathbb{E}_{p(\theta|\mathcal{D}_T)} KL[p(y|x,\theta)||p(y|x,\theta^*)] =$$
$$\int p(\theta|\mathcal{D}_T) \int p(y|x,\theta) \log \frac{p(y|x,\theta)}{p(y|x,\theta^*)} d\theta =$$
$$\int p(\theta|\mathcal{D}_T) \int p(y|x,\theta) \log \frac{p(y|x,\theta)p(y|x)}{p(y|x,\theta^*)p(y|x)} d\theta =$$
$$\int p(\theta|\mathcal{D}_T) \int p(y|x,\theta) \log \frac{p(y|x,\theta)}{p(y|x)} d\theta +$$
$$\int p(y|x) \log \frac{p(y|x)}{p(y|x,\theta^*)} d\theta =$$
$$\mathbb{E}_{p(\theta|\mathcal{D}_T)} KL[p(y|x,\theta)||p(y|x)] +$$
$$KL[p(y|x)||p(y|x,\theta^*)]$$

As the second KL divergence in the equation above is non-negative, we can derive the following bound:

$$\mathbb{E}_{p(\theta|\mathcal{D}_T)} KL[p(y|x,\theta)||p(y|x,\theta^*)] \geq \mathbb{E}_{p(\theta|\mathcal{D}_T)} KL[p(y|x,\theta)||p(y|x)].$$

Which is analogous to the bound of equation 6, with the difference that the average divergence is with respect to the change in model's weights θ instead of prompts a.

B Details of Experiments

B.1 Details of Data Sets

For our evaluation, we test on 10 different NLP classification tasks with the respective test sets:

- **Amazon Reviews (Appliances):** Sentiment analysis on Amazon reviews of appliances. Two classes; positive and negative.
- **DBPedia 14:** Topic modeling on a data set of encyclopedia descriptions corresponding to 14 categories.
- **Glue (mrpc):** Binary equivalence judgment on pairs of sentences.
- **Imdb:** Sentiment analysis on a data set of movie reviews on Imdb. Two classes; positive and negative.
- **SNLI:** Entailment classification on a data set of pairs of sentences, where the first sentence implies, contradicts or does not imply the second (3 classes).
- **SST-2:** Sentiment analysis on a Data set of segments from movie reviews. Two classes; positive and negative.
- **TREC:** Topic modeling on a data set of questions labeled with type of information requested. We are using only the coarse labels (6 classes, reduced to 5 for class balance).
- **Yahoo Answers:** Topic modeling on a data set of Questions and answers on Yahoo answers. 10 classes.
- **YouTube:** Spam detection on comments left under YouTube videos. Two classes; Spam and Ham.
- **SMS:** Spam detection on text messages. Two classes; Spam and Ham.

In all our experiments, we balance the test sets we use to evaluate metrics across classes. This is to ensure that all metrics are meaningful measure of performance in each case, as for heavily unbalanced sets accuracy/likelihood based metrics, such as negative log-likelihood, are not meaningful measures of model performance.

B.2 Details of LLMs

We use 5 different open-source LLMs, across different families and sizes. They are listed in Table 4:

B.3 Details of Initial Prompts Generation

For each task, we manually write one prompt instruction, which we then automatically rephrase multiple times with gpt-3.5-turbo. The initial manually defined prompt is not validated on any data and the generated prompts are not manually edited. While better prompts can certainly be defined in more principled engineering ways with validation data sets (e.g. Shin et al. (2020)), we are interested in the optimal exploitation of given prompts and avoiding prompt engineering ensures that our findings are not biased to certain manual fine-tuning choices or particular prompt generation methods. The initial prompts for each data set are as follows:

- **Amazon Reviews:** classify the sentiment of the Amazon review below into one of the following classes:
- **DBPedia 14:** The following is a description of something on Wikipedia. Which of the following categories does the description fall in?
- **Glue (mrpc):** Do the two sentences given below have the same meaning?

Model Name	Huggingface Name	Hardware Used
Falcon-7b-Instruct	tiiuae/falcon-7b-instruct	1 A10 GPU
Falcon-40b-Instruct	tiiuae/falcon-40b-instruct	4 A10 GPUs (bfloat16)
MPT-7b-Instruct	mosaicml/mpt-7b-instruct	1 A10 GPU
MPT-30b-Instruct	mosaicml/mpt-30b-instruct	4 A10 GPUs
Mistral-7b-Instruct	mistralai/Mistral-7B-Instruct-v0.1	1 A10 GPU

Table 4: LLMs with which we tested BayesPE and baselines. We list above the name with which they will appear in result tables, the Huggingface name, i.e., the string to give as argument to AutoModelForCausalLM, and the hardware that was used to run inference.

- **Imdb:** classify the below movie review into one of the following classes:
- **SNLI:** Classify the relation between the hypothesis and thesis below into one of the following classes:
- **SST-2:** classify the sentiment of the movie review extract below into one of the following classes:
- **TREC:** Classify the question below into one of the following type of information classes:
- **Yahoo Answers:** classify the question and answer below into one of the following topics:
- **YouTube:** Is the following Youtube comment spam?
- **SMS:** Is the following text message spam?

In each case, the above is followed by a numbered list of the class words. The initial prompt for each task is then used to formulate a re-phrasing promt to be given to gpt-3.5-turbo. For the first generated prompt, gpt-3.5-turbo is given the prompt "Rephrase the following task instructions such that the meaning remains unchanged. /n {initial prompt}". For all subsequent prompts, the already generated prompts are provided in a list with the prompt 'Provide one more example of rephrased task instructions, such that the meaning is the same as the original task instructions. /n {list of examples} /n Original instructions: {initial prompt}'. For the experiments of section 5, we generate 10 total prompts, from which we draw 5 in each repetition of the experiments. For few-shot, we also draw at random a different set of 5 labeled examples from a held-out set (different from the set we use as validation in the experiments) and append it to each different re-phrasing of the instructions, in order to obtain the final different prompts a_i.

B.4 Details of BayesPE training

the BayesPE weights are trained by performing the maximisation of equation 4, with validation data $\mathcal{D} = \{\mathbf{x}, \mathbf{y}^*\}$ with the LBFGS optimiser with 100 maximum iterations. The learning rate is set to 10^{-3}, however, if the cost becomes NaN or Inf, the optimisation is restarted with halfed learning rate and the process is repeated until succesful convergence. This occured for a few LLM-task combinations in our experiments. Other than this error recovery strategy, the BayesPE optimisation parameters are constant to all experiments. BayesPE optimisation is inexpensive (1-2 seconds on CPU), as it only needs to optimise a few linear weights for class probabilities with 2-15 dimensions.

B.5 Details of Samples Efficiency Experiments

To test performance at varying budget of LLM forward passes, we use Mistral7bInstruct as the base LLM and evaluate performance on three tasks: Amazon Reviews, DBPedia 14 and Yaho Answers. For these experiments, as in the full benchmark, we use a validation set of 50 examples, approximately balanced across classes. This validation set is used to train the weights for BayesPE and select the prompt with the highest validation accuracy for Best. We test with 400 examples, also balanced across classes. We consider a pool of 50 initial prompts. For every repetition of the experiments, we draw a set of $N = 20$ at random from them and use them to optimise the BayesPE weights $\mathbf{w}^*$ and determine the highest accuracy prompt for the Best baseline. The forward passes budget L is then varied between 1 and 10. Best and Standard only operate with a single prompt, therefore they only appear as single points at $L = 1$ in Figure 4. For the baseline Ensemble, L prompts are sampled at random and aggregated to obtain output probabilities. For BayesPE, the prompts corresponding to the top L weights w_i^* are taken and aggregated,

Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Amazon Reviews					
Standard	0.249 ± 0.021	0.972 ± 0.004	0.040 ± 0.006	0.103 ± 0.041	0.067 ± 0.005
Self-Assess	0.708 ± 0.035	0.601 ± 0.123	0.122 ± 0.101	0.257 ± 0.316	0.251 ± 0.008
DBPedia 14					
Standard	0.946 ± 0.074	0.98 ± 0.005	0.051 ± 0.008	0.096 ± 0.034	0.027 ± 0.001
Self-Assess	2.980 ± 0.684	0.502 ± 0.004	0.169 ± 0.296	0.170 ± 0.341	0.075 ± 0.018
Imdb					
Standard	0.290 ± 0.026	0.964 ± 0.0038	0.055 ± 0.010	0.137 ± 0.077	0.083 ± 0.009
Self-Assess	0.616 ± 0.098	0.695 ± 0.197	0.0829 ± 0.094	0.219 ± 0.252	0.220 ± 0.038
Yahoo Answers					
Standard	2.202 ± 0.063	0.6933 ± 0.008	0.161 ± 0.052	0.313 ± 0.037	0.083 ± 0.002
Self-Assess	3.387 ± 1.336	0.495 ± 0.007	0.345 ± 0.416	0.356 ± 0.436	0.123 ± 0.040

Table 5: Probabilistic metrics over different data sets comparing the performance of using the class words probabilities from the LLM directly to compute class probability distribution (Standard) and using the self-assess method (Self-Assess).

weighted by the corresponding weights. The resulting probabilities are re-normalised.

B.6 Details of Labeled Data Efficiency Experiments

To test performance at varying number of labeled validation examples, we use Mistral7bInstruct as the base LLM and evaluate performance on three tasks: Amazon Reviews, DBPedia 14 and Yaho Answers. For each task, we use a test set of 400 examples to evaluate performance and vary the available validation data set size from $M = 0$ to $M = 100$. Similarly to the samples efficiency ablations, for each value of validation set size M, we draw $N = 20$ prompts from a pool of 50 at random and use them on the validation set to choose the prompt with highest accuracy (Best) and to optimise the BayesPE weights $\mathbf{w}^*$. At run time, we use an inference budget of $L = 10$ LLM forward passes and hence 10 prompts. For each method, these are chosen from the total $N = 20$ in the same way as in the samples efficiency experiments (Appendix B.5).

C Additional Results

C.1 Early Experiments with LLM Self-Assessing of Confidence

We initially considered the self-assessing of confidence by the LLM itself as a baseline for uncertainty estimation ad calibration of output probabilities. This method consists in feeding the LLM the original prompt and its original answer and then prompting again to assess if the answer is correct. The probability of the next word confirming correctness is taken as the probability of the class given in the answer, e.g., 'is the above answer correct?' $p(\text{next word} = \text{'yes'}) = p(y = \text{chosen class})$ (Kadavath et al., 2022). In the binary case, the probability distribution is directly derived; if chosen class is 1 $p(y = 1) = p(\text{next word} = \text{'yes'})$ and $p(y = 0) = 1 - p(\text{next word} = \text{'yes'})$. However, in the multi class case, there is not direct mapping. In these cases, we assign the chosen class the confirming word probability ($p(\text{next word} = \text{'yes'})$) and distribute the remaining probability mass uniformly amongst the remaining classes. We tested this approach in initial ablations with Mistral7bInstruct in zero-shot mode with four tasks. Results are shown in table 5.

We found the standard approach of using the LLM probabilities assigned to classes words (Standard in table 5 and rest of the paper) to consistently and substantially outperform this approach of re-evaluating confidence in the given answer (Self-Assess in table 5). As a result, we ruled out this approach as a competitive baseline for BayesPE in all further experiments.

C.2 Additional Efficiency results

Inference Efficiency We show in figure 6 an extended version of the results of figure 4, including zero-shot results. Trends for zero-shot are analogous to those observed in few-shot experiments,

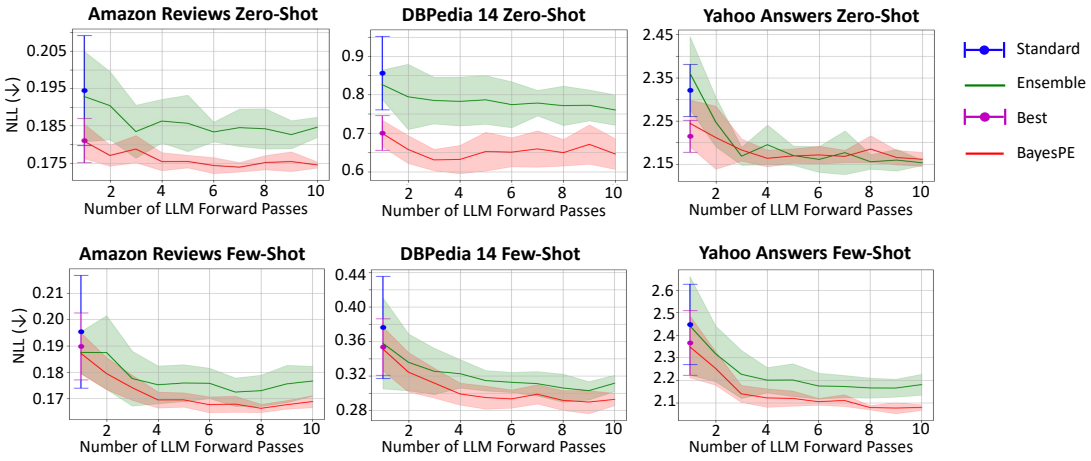

Figure 6: Negative log likelihood (NLL) vs. number of forward passes through the LLM at run-time. Using any prompt (Standard) and running the best performing prompt on validation (Best) can be performed only with a single prompt and hence a single forward pass. Standard ensembling (Ensemble) and our BayesPE approach exploit more prompts as the forward passes budget is increased.

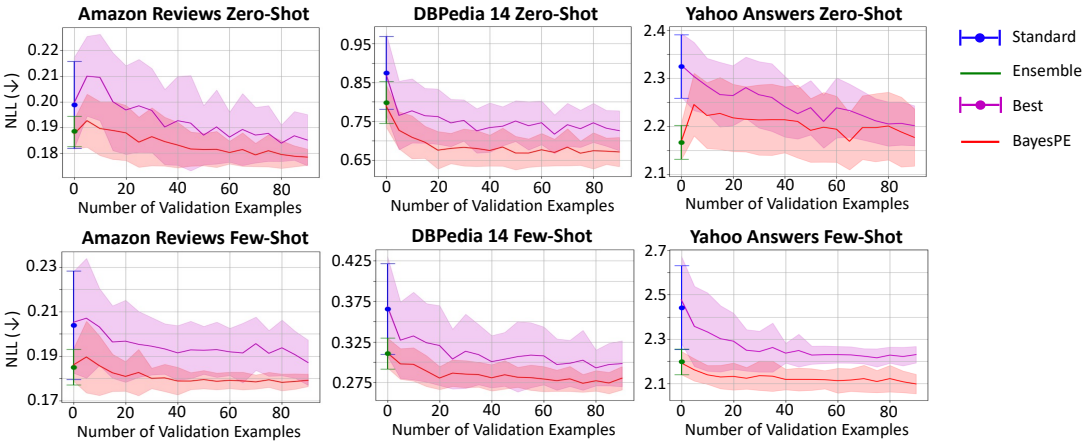

Figure 7: Negative log likelihood (NLL) vs. number of labeled validation examples. Using any prompt (Standard) and aggregating all prompts equally (Ensemble) does not require any validation set and are shown as single points at zero validation examples. Choosing the highest accuracy prompt on validation (Best) and BayesPE benefit from increased number of validation examples.

with BayesPE being consistently better than or competitive with Ensemble for all experimental settings and all inference budgets and providing consistent improvement compared to picking the best prompt on validation. For instance, on DBPedia 14 and a budget of 5 forward passes in zero-shot, BayesPE achieves an NLL of 0.65, while standard ensembling can only achieve an average of 0.77.

Labelled Data Efficiency We show in figure 7 an extended version of the results of figure 5, including zero-shot results. Trends are analogous to those observed in the few-shot results, e.g., for DBPedia 14 in Zero-Shot we report an NLL of 0.67 for BayesPE with 50 validation examples vs. 0.75 for Best and 0.80 for Ensemble. In some cases, most evidently for Yahoo Answers in zero-shot, adding very few validation examples (5-10) initially causes to over-fit and degrade performance (from NLL 2.17 with no validation to 2.25 with five validation examples). However, as the validation data set increases to several tens of examples, this over-fitting is mitigated and the BayesPE NLL matches or improves on that of the Ensemble baseline (e.g. NLL back to 2.17 with 90 validation example for Yahoo Answers in zero-shot).

C.3 Prompt Definition

In all experiments presented above, we have tested the utilization of given prompts independently of any particular engineering, as the set of initial prompts was pre-defined. We investigate here

how different generation strategies for the prompts $a_i \in \mathbf{a}$ affect performance. First, we evaluate performance for different templated prompt augmentation strategies, adopting the different alterations detailed in Jiang et al. (2023b). Second, we test BayesPE in combination with automatic prompt engineering (APE) (Zhou et al., 2022b), which uses the validation set to refine the definition of the initial prompts.

Prompt Augmentation Strategies We adopt the augmentation strategies proposed by Jiang et al. (2023b), adding them one at a time in our experiments to increase prompt variation in each experiment. For these experiments, we use the Yahoo Answers data set, Mistral-7b-Instruct as the base LLM and operate in a few-shot scenarion, providing 5 5 examples in the input prompts. We use 10 prompts to build the ensemble and 50 validation examples. The in-context examples are taken from a separate set of 50 examples. We test with initial prompts defined in the following three ways:

- **Rephrase Only:** We use GPT-3.5-Turbo to rephrase the task instructions (see Appendix B.3), but in-context examples for few shot and order of multiple choices is fixed.
- **Rephrase+Examples:** Instructions are rephrased as described above and the 5 in-context examples for few-shot operations are sampled and ordered at random in each of the 10 prompts.
- **Rephrase+Examples+MC:** Rephrasing and random sampling of the examples are applied as described above and, in addition, multiple choices for the classes presented in the prompts are permuted at random.

The diversity of the prompts icreases with the addition of each included augmentation strategy. We repeat each experiment 10 times, each time drawing at random the 50 validation examples, the in-context examples and the set of 10 re-phrased instructions. Results are shown in table 6. Note that, in these experiments, the Ensemble baseline corresponds to the approach proposed in Jiang et al. (2023b).

In the results of table 6, BayesPE is consistently better or competitive to baselines, providing 48-63% reduction in ECE compared to standard prompting and 4-12% compared to Ensemble, which is the most competitive baseline. We also note that, unlike all competing strategies, BayesPE is less reliant on the prompt definition strategy. For example the ECE is $\sim$ 8% for all prompt definitions, while for ensembles it reduces from 9.2% to 8.4% as prompt diversity is increased. This means that BayesPE is more robust to the initial engineering of the set of prompts in the ensemble, alleviating the burden of designing effective and diverse prompts for calibration.

Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)	F1 (↑)
Rephrase Only						
Standard	2.466	0.670	0.158	0.208	0.087	0.249
Ensemble	2.089	**0.697**	0.092	0.156	**0.080**	0.286
Best	2.098	**0.691**	0.099	0.226	**0.081**	0.284
BayesPE	**2.072**	**0.692**	**0.081**	**0.128**	**0.080**	**0.299**
Rephrase+Examples						
Standard	2.520	0.680	0.213	0.267	0.091	0.251
Ensemble	2.073	**0.696**	0.088	0.151	**0.080**	0.285
Best	2.151	**0.693**	0.119	**0.138**	0.083	0.281
BayesPE	**2.047**	**0.696**	**0.080**	**0.141**	**0.080**	**0.302**
Rephrase+Examples+MC						
Standard	2.499	0.675	0.222	0.322	0.091	0.242
Ensemble	**2.035**	**0.704**	0.084	0.140	**0.080**	0.272
Best	2.233	0.687	0.155	0.215	0.084	0.249
BayesPE	**2.021**	**0.696**	**0.081**	**0.116**	**0.0799**	**0.286**

Table 6: Few-shot classification with Mistral-7B-Instruct on the Yahoo Answers data set and initial prompts generated with different strategies. Metrics values which are best within statistical significance (two-sample p-value>0.05) are printed in bold.

Automatic Prompt Engineering and BayesPE Given a validation set of labeled examples, recent work has proposed different strategies to automatically generate suitable prompts for a given task (Shin et al., 2020; Zhou et al., 2022b; Yang et al., 2023). To test the combination of BayesPE with automatic prompts, we follow Zhou et al. (2022b) and use automatic prompt engineering (APE) to define the initial prompts. We run APE 10 times with the validation set to obtain the initial set of prompts and then apply BayesPE and all baselines, using the same validation set where relevant. For these experiments, we consider the same three tasks as for the efficiency experiments (Amazon Reviews, DBPedia 14 and Yahoo Answers) and use Mistral7BInstruct as the base LLM. We perform experiments in zero-shot format only, as it is not trivial to adapt the APE approach to few-shot scenarios. We run APE (Zhou et al., 2022b) 90 times to obtain an initial pool of prompts. We then select the top 30 based on validation log likelihood. From these, we sample 10 at random and use them for BayesPE, Ensemble and Best. The standard APE baseline is tested by selecting one of the 30 prompts

at random. Prompt sampling is repeated 10 times to obtain error bars and compute statistical significance. In these experiments, the validation set is fixed and is composed of 100 labeled examples for each task. The test set is also fixed and is composed of 400 examples. Table 7 reports results of these experiments.

Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)	F1 (↑)
Amazon Reviews						
APE	0.212	0.978	**0.029**	0.096	0.060	**0.933**
APE+Ensemble	0.192	**0.983**	0.03	0.088	**0.053**	0.932
APE+Best	**0.18**	**0.982**	**0.019**	**0.060**	**0.050**	**0.938**
APE+BayesPE	**0.187**	**0.982**	**0.020**	**0.052**	**0.052**	**0.934**
DBPedia 14						
APE	0.850	0.984	**0.047**	**0.091**	0.028	0.755
APE+Ensemble	0.718	**0.991**	0.057	**0.082**	0.025	0.756
APE+Best	0.756	0.988	**0.039**	0.102	**0.024**	0.763
APE+BayesPE	**0.696**	**0.991**	**0.045**	**0.090**	**0.024**	**0.783**
Yahoo Answers						
APE	2.327	0.667	0.126	0.616	0.088	0.258
APE+Ensemble	2.186	**0.682**	**0.090**	0.984	0.084	**0.292**
APE+Best	2.210	**0.681**	0.110	0.502	0.084	0.259
APE+BayesPE	**2.151**	**0.681**	**0.090**	**0.262**	**0.083**	**0.285**

Table 7: Zero-shot classification with Mistral-7B-Instruct and initial prompts generated with APE (Zhou et al., 2022b). Metrics values which are best within statistical significance (two-sample p-value>0.05) are printed in bold.

Using BayesPE in combination with APE (APE+BayesPE), was found to consistently improve calibration performance over APE alone. In particular, for the Amazon Reviews and Yahoo Answers data sets, we observed substantial reduction of about 30% in ECE and $45-55\%$ in MCE. We also confirm that the deterministic performance, measured by F1-score, is not negatively affected by using BayesPE in combination with APE, remaining competitive or better (difference in F1 from 0 to $+0.03$) with respect to this metric as well. We also observe that BayesPE remains competitive or better than other baselines methods to exploit multiple prompts and validation data.

C.4 Comparison to Bayesian Mixture of Experts

As described in Appendix A.4, BayesPE can be interpreted as a strategy for Bayesian Mixture of Experts (MoE), using the LLM with different prompts as basis functions. We perform some experiments to compare BayesPE to linear Bayesian regression over the weights of the Bayesian MoE. In particular, we preform Ridge regression, where the prior over the weights is a Gaussian with diagonal covariance. To introduce a regularisation which encourages uniform exploitation of the prompts a-priori, we set the means of the prior over the weights to $1/N$ and the diagonal variances to 1. We then infer the posterior using the available validation examples. We compare this approach to BayesPE and the other baselines (Standard, Ensemble and Best). For these experiments, we use the Amazon Reviews, Imdb and Yahoo Answer data sets, an initial ensemble of ten few-shot prompts, designed as described in Appendix B.3, 50 validation examples and Mistral-7b-Instruct as the base LLM. Each experiment is repeated 10 times for statistical significance.

Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)	F1 (↑)
Amazon Reviews						
Standard	0.187	0.981	0.032	0.197	0.054	**0.930**
Ensemble	**0.174**	**0.982**	0.020	0.107	**0.051**	**0.929**
Best	0.187	0.981	**0.010**	**0.032**	0.053	**0.930**
Bayes MoE	**0.173**	**0.982**	0.023	0.089	**0.050**	**0.930**
BayesPE	**0.173**	**0.983**	**0.010**	**0.034**	0.052	**0.930**
Imdb						
Standard	0.196	**0.983**	0.038	0.271	0.058	**0.922**
Ensemble	0.172	**0.985**	0.026	0.203	0.054	**0.925**
Best	0.182	**0.983**	0.030	0.251	0.058	**0.925**
Bayes MoE	0.175	**0.983**	**0.022**	0.235	0.054	0.920
BayesPE	**0.159**	**0.984**	**0.022**	**0.188**	**0.50**	**0.925**
Yahoo Answers						
Standard	2.491	0.711	0.180	0.238	0.086	0.303
Ensemble	2.185	0.710	0.111	0.284	0.081	**0.333**
Best	2.060	0.722	0.094	0.142	0.078	0.331
Bayes MoE	2.059	0.709	0.121	0.205	0.079	0.331
BayesPE	**2.039**	**0.722**	**0.096**	**0.133**	**0.078**	**0.325**

Table 8: Few-shot classification with Mistral-7B-Instruct comparing baselines and BayesPE to a Bayesian MoE approach to infer the weights $\mathbf{w}$ for the ensemble (Bayes MoE). Metrics values which are best within statistical significance (two-sample p-value>0.05) are printed in bold.

As shown in table 8, BayesPE is competitive or better than using linear regression to train the weights $\mathbf{w}$ in a Bayesian MoE formulation, with ECE being equal (Imdb) to $20-56\%$ lower (Amazon Reviews and Yahoo Answers).

C.5 Detailed Results of Benchmark Evaluation

Tables 9-18 report the results of our extensive benchmark evaluations for zero-shot classification in full. Errorbars were obtained by repeating the experiments 10 times, drawing a set of 5 prompts as described in Appendix B.3 each repeat. Tables 19-28 report results for the equivalent few-shot experiments.

C.6 Additional Statistical Significance Results

We report here analogous summary results of our benchmark experiments to those presented in table 3, separated for zero-shot and few shot and then for each individual LLM. Tables 29 and 30 report

summary results for zero-shot and few shot experiments respectively. Tables 31 to 35 reprt summary results for each LLM tested separately. In all tables 29 to 35, numbers in each cell report in how many experiments out of the ones performed each approach was found to be best performing or within statistical significance of the best performing (two-sample p-value> 0.05 over the repetitions of each experiment) for each measured metric. The number in parenthesis reports how many times the method was found to be significantly better than any other (p-value< 0.05).

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	**0.298 ± 0.032**	**0.967 ± 0.004**	**0.09 ± 0.014**	**0.143 ± 0.041**	**0.087 ± 0.012**
Falcon7bInstruct + Ensemble	**0.286 ± 0.014**	**0.969 ± 0.001**	0.102 ± 0.01	**0.155 ± 0.022**	**0.081 ± 0.005**
Falcon7bInstruct + Best	**0.296 ± 0.036**	0.966 ± 0.003	**0.089 ± 0.013**	0.181 ± 0.046	**0.086 ± 0.014**
Falcon7bInstruct + BayesPE	**0.29 ± 0.01**	**0.969 ± 0.001**	0.101 ± 0.007	**0.139 ± 0.01**	**0.083 ± 0.004**
Falcon40bInstruct + Standard	**0.264 ± 0.044**	**0.982 ± 0.002**	**0.105 ± 0.033**	**0.158 ± 0.026**	**0.072 ± 0.016**
Falcon40bInstruct + Ensemble	0.255 ± 0.009	**0.983 ± 0.001**	0.118 ± 0.012	**0.17 ± 0.005**	0.067 ± 0.003
Falcon40bInstruct + Best	**0.266 ± 0.054**	**0.982 ± 0.002**	**0.117 ± 0.036**	**0.166 ± 0.018**	**0.072 ± 0.017**
Falcon40bInstruct + BayesPE	**0.242 ± 0.005**	**0.983 ± 0.001**	**0.105 ± 0.009**	**0.166 ± 0.002**	**0.063 ± 0.002**
MPT7bInstruct + Standard	**0.394 ± 0.045**	0.956 ± 0.008	**0.16 ± 0.016**	**0.18 ± 0.012**	**0.117 ± 0.018**
MPT7bInstruct + Ensemble	0.393 ± 0.01	0.962 ± 0.001	0.193 ± 0.009	0.208 ± 0.01	0.115 ± 0.004
MPT7bInstruct + Best	**0.383 ± 0.028**	**0.963 ± 0.005**	**0.16 ± 0.018**	**0.185 ± 0.009**	**0.112 ± 0.011**
MPT7bInstruct + BayesPE	**0.381 ± 0.006**	**0.964 ± 0.001**	0.178 ± 0.008	0.2 ± 0.004	**0.111 ± 0.003**
MPT30bInstruct + Standard	0.465 ± 0.099	0.955 ± 0.008	0.094 ± 0.043	0.167 ± 0.098	0.142 ± 0.036
MPT30bInstruct + Ensemble	0.407 ± 0.016	**0.96 ± 0.002**	0.06 ± 0.012	**0.101 ± 0.022**	0.124 ± 0.006
MPT30bInstruct + Best	**0.361 ± 0.046**	**0.96 ± 0.002**	**0.047 ± 0.027**	**0.121 ± 0.059**	**0.104 ± 0.015**
MPT30bInstruct + BayesPE	**0.361 ± 0.024**	**0.961 ± 0.001**	**0.04 ± 0.016**	**0.078 ± 0.046**	**0.106 ± 0.009**
Mistral7bInstruct + Standard	**0.247 ± 0.019**	**0.973 ± 0.003**	**0.039 ± 0.008**	**0.088 ± 0.052**	**0.066 ± 0.004**
Mistral7bInstruct + Ensemble	0.246 ± 0.008	0.973 ± 0.001	**0.041 ± 0.004**	**0.115 ± 0.055**	0.067 ± 0.002
Mistral7bInstruct + Best	**0.238 ± 0.014**	**0.974 ± 0.001**	**0.038 ± 0.005**	**0.093 ± 0.052**	**0.064 ± 0.001**
Mistral7bInstruct + BayesPE	**0.235 ± 0.003**	**0.975 ± 0.001**	**0.037 ± 0.005**	**0.093 ± 0.023**	**0.065 ± 0.001**

Table 9: Amazon Reviews zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	2.603 ± 0.305	0.917 ± 0.017	0.184 ± 0.092	0.219 ± 0.114	0.061 ± 0.008
Falcon7bInstruct + Ensemble	2.598 ± 0.106	0.924 ± 0.006	0.182 ± 0.036	0.227 ± 0.02	0.062 ± 0.002
Falcon7bInstruct + Best	2.455 ± 0.153	**0.927 ± 0.004**	0.139 ± 0.02	0.174 ± 0.042	**0.056 ± 0.003**
Falcon7bInstruct + BayesPE	**2.316 ± 0.105**	**0.929 ± 0.003**	**0.114 ± 0.016**	**0.118 ± 0.028**	**0.054 ± 0.002**
Falcon40bInstruct + Standard	2.385 ± 0.029	0.937 ± 0.006	**0.103 ± 0.02**	**0.363 ± 0.066**	0.057 ± 0.001
Falcon40bInstruct + Ensemble	**2.334 ± 0.019**	**0.948 ± 0.004**	0.125 ± 0.024	**0.355 ± 0.031**	**0.056 ± 0.0**
Falcon40bInstruct + Best	**2.324 ± 0.039**	0.939 ± 0.006	**0.08 ± 0.026**	**0.295 ± 0.087**	**0.056 ± 0.001**
Falcon40bInstruct + BayesPE	**2.315 ± 0.025**	0.944 ± 0.004	0.106 ± 0.024	**0.324 ± 0.024**	**0.056 ± 0.001**
MPT7bInstruct + Standard	1.819 ± 0.115	**0.965 ± 0.006**	0.103 ± 0.02	0.211 ± 0.049	0.043 ± 0.002
MPT7bInstruct + Ensemble	1.755 ± 0.036	**0.966 ± 0.002**	**0.07 ± 0.015**	**0.143 ± 0.029**	0.041 ± 0.0
MPT7bInstruct + Best	**1.734 ± 0.06**	0.962 ± 0.002	**0.071 ± 0.008**	**0.171 ± 0.058**	**0.04 ± 0.002**
MPT7bInstruct + BayesPE	**1.709 ± 0.011**	**0.966 ± 0.001**	**0.068 ± 0.01**	**0.147 ± 0.012**	**0.04 ± 0.001**
MPT30bInstruct + Standard	1.638 ± 1.233	0.964 ± 0.025	0.238 ± 0.251	0.32 ± 0.311	0.044 ± 0.032
MPT30bInstruct + Ensemble	1.213 ± 0.198	0.976 ± 0.003	0.202 ± 0.102	0.325 ± 0.16	0.043 ± 0.009
MPT30bInstruct + Best	**0.717 ± 0.069**	0.983 ± 0.001	**0.034 ± 0.012**	**0.096 ± 0.021**	**0.018 ± 0.001**
MPT30bInstruct + BayesPE	**0.667 ± 0.061**	**0.984 ± 0.001**	**0.032 ± 0.008**	**0.104 ± 0.018**	**0.018 ± 0.001**
Mistral7bInstruct + Standard	0.946 ± 0.09	0.979 ± 0.005	**0.06 ± 0.016**	**0.111 ± 0.032**	0.026 ± 0.001
Mistral7bInstruct + Ensemble	0.905 ± 0.029	0.982 ± 0.002	**0.066 ± 0.004**	**0.126 ± 0.021**	0.024 ± 0.001
Mistral7bInstruct + Best	0.89 ± 0.103	**0.984 ± 0.004**	**0.06 ± 0.028**	**0.105 ± 0.043**	**0.024 ± 0.002**
Mistral7bInstruct + BayesPE	**0.808 ± 0.026**	**0.986 ± 0.0**	**0.056 ± 0.025**	**0.126 ± 0.04**	**0.023 ± 0.0**

Table 10: DBPedia 14 zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	**0.853 ± 0.053**	0.272 ± 0.018	0.274 ± 0.015	0.619 ± 0.114	**0.321 ± 0.018**
Falcon7bInstruct + Ensemble	**0.837 ± 0.017**	0.263 ± 0.012	**0.235 ± 0.021**	0.546 ± 0.03	**0.315 ± 0.007**
Falcon7bInstruct + Best	**0.83 ± 0.032**	**0.305 ± 0.034**	**0.22 ± 0.024**	**0.489 ± 0.04**	**0.312 ± 0.013**
Falcon7bInstruct + BayesPE	**0.84 ± 0.007**	0.259 ± 0.018	**0.237 ± 0.021**	0.539 ± 0.028	**0.316 ± 0.003**
Falcon40bInstruct + Standard	**0.886 ± 0.054**	**0.361 ± 0.011**	**0.276 ± 0.023**	**0.484 ± 0.219**	**0.332 ± 0.02**
Falcon40bInstruct + Ensemble	**0.858 ± 0.022**	0.354 ± 0.009	**0.285 ± 0.007**	**0.378 ± 0.018**	**0.322 ± 0.008**
Falcon40bInstruct + Best	**0.862 ± 0.029**	**0.365 ± 0.007**	**0.289 ± 0.025**	**0.374 ± 0.052**	**0.324 ± 0.012**
Falcon40bInstruct + BayesPE	**0.845 ± 0.027**	0.356 ± 0.008	**0.285 ± 0.009**	**0.378 ± 0.022**	**0.318 ± 0.01**
MPT7bInstruct + Standard	**1.435 ± 0.137**	**0.38 ± 0.046**	**0.441 ± 0.016**	**0.539 ± 0.064**	**0.453 ± 0.017**
MPT7bInstruct + Ensemble	1.431 ± 0.03	0.368 ± 0.021	0.443 ± 0.004	**0.523 ± 0.017**	0.455 ± 0.004
MPT7bInstruct + Best	1.438 ± 0.086	**0.386 ± 0.045**	0.443 ± 0.007	**0.51 ± 0.02**	0.455 ± 0.009
MPT7bInstruct + BayesPE	**1.357 ± 0.029**	**0.397 ± 0.006**	**0.433 ± 0.005**	0.64 ± 0.086	**0.444 ± 0.004**
MPT30bInstruct + Standard	1.083 ± 0.076	**0.486 ± 0.02**	0.277 ± 0.036	**0.491 ± 0.029**	0.351 ± 0.016
MPT30bInstruct + Ensemble	**0.973 ± 0.067**	**0.486 ± 0.012**	**0.241 ± 0.028**	**0.482 ± 0.029**	**0.333 ± 0.016**
MPT30bInstruct + Best	1.048 ± 0.03	**0.479 ± 0.017**	0.274 ± 0.012	**0.508 ± 0.03**	0.348 ± 0.009
MPT30bInstruct + BayesPE	**0.961 ± 0.032**	**0.484 ± 0.009**	**0.242 ± 0.013**	**0.48 ± 0.028**	**0.332 ± 0.008**
Mistral7bInstruct + Standard	0.834 ± 0.023	**0.446 ± 0.04**	**0.215 ± 0.017**	**0.379 ± 0.068**	0.306 ± 0.008
Mistral7bInstruct + Ensemble	**0.808 ± 0.019**	**0.438 ± 0.016**	**0.2 ± 0.023**	0.702 ± 0.268	**0.298 ± 0.005**
Mistral7bInstruct + Best	0.827 ± 0.014	**0.437 ± 0.045**	**0.218 ± 0.017**	**0.38 ± 0.065**	0.306 ± 0.007
Mistral7bInstruct + BayesPE	**0.811 ± 0.006**	0.401 ± 0.024	**0.214 ± 0.008**	0.447 ± 0.057	0.303 ± 0.001

Table 11: Glue (mrpc) zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	**0.669 ± 0.179**	0.909 ± 0.014	**0.166 ± 0.133**	**0.266 ± 0.264**	**0.236 ± 0.069**
Falcon7bInstruct + Ensemble	0.714 ± 0.046	**0.927 ± 0.006**	0.262 ± 0.032	0.43 ± 0.052	0.263 ± 0.019
Falcon7bInstruct + Best	**0.566 ± 0.017**	0.904 ± 0.007	**0.085 ± 0.02**	**0.124 ± 0.034**	**0.197 ± 0.008**
Falcon7bInstruct + BayesPE	**0.552 ± 0.017**	0.916 ± 0.004	**0.076 ± 0.026**	**0.12 ± 0.045**	**0.191 ± 0.008**
Falcon40bInstruct + Standard	2.352 ± 0.344	0.923 ± 0.035	0.49 ± 0.008	0.759 ± 0.173	0.484 ± 0.012
Falcon40bInstruct + Ensemble	2.205 ± 0.136	**0.958 ± 0.01**	0.49 ± 0.002	**0.567 ± 0.151**	0.484 ± 0.004
Falcon40bInstruct + Best	2.288 ± 0.341	0.936 ± 0.03	0.489 ± 0.008	**0.688 ± 0.194**	0.482 ± 0.012
Falcon40bInstruct + BayesPE	**1.813 ± 0.126**	0.922 ± 0.041	**0.478 ± 0.005**	0.875 ± 0.008	**0.465 ± 0.006**
MPT7bInstruct + Standard	2.393 ± 0.163	0.764 ± 0.038	0.492 ± 0.004	**0.531 ± 0.114**	0.489 ± 0.006
MPT7bInstruct + Ensemble	2.154 ± 0.109	**0.838 ± 0.019**	0.488 ± 0.002	**0.488 ± 0.002**	0.483 ± 0.003
MPT7bInstruct + Best	2.231 ± 0.364	**0.805 ± 0.054**	**0.486 ± 0.011**	**0.605 ± 0.171**	**0.481 ± 0.016**
MPT7bInstruct + BayesPE	**1.889 ± 0.304**	**0.84 ± 0.036**	**0.477 ± 0.011**	0.754 ± 0.173	**0.467 ± 0.016**
MPT30bInstruct + Standard	1.912 ± 0.269	0.946 ± 0.016	0.478 ± 0.012	**0.821 ± 0.111**	0.461 ± 0.019
MPT30bInstruct + Ensemble	1.734 ± 0.081	**0.961 ± 0.004**	0.474 ± 0.004	**0.859 ± 0.005**	0.455 ± 0.006
MPT30bInstruct + Best	1.967 ± 0.294	0.937 ± 0.012	0.479 ± 0.013	**0.864 ± 0.019**	0.464 ± 0.021
MPT30bInstruct + BayesPE	**1.484 ± 0.146**	**0.961 ± 0.006**	**0.456 ± 0.009**	**0.834 ± 0.008**	**0.425 ± 0.015**
Mistral7bInstruct + Standard	0.289 ± 0.025	0.958 ± 0.01	**0.045 ± 0.019**	**0.108 ± 0.068**	0.086 ± 0.01
Mistral7bInstruct + Ensemble	0.257 ± 0.008	**0.965 ± 0.003**	**0.034 ± 0.009**	0.158 ± 0.051	**0.078 ± 0.003**
Mistral7bInstruct + Best	0.268 ± 0.018	**0.966 ± 0.003**	0.052 ± 0.007	**0.101 ± 0.051**	**0.076 ± 0.002**
Mistral7bInstruct + BayesPE	**0.247 ± 0.01**	**0.966 ± 0.004**	**0.034 ± 0.004**	0.17 ± 0.067	**0.076 ± 0.003**

Table 12: Imdb zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	3.769 ± 0.16	**0.547 ± 0.02**	0.681 ± 0.002	0.681 ± 0.002	0.454 ± 0.002
Falcon7bInstruct + Ensemble	3.808 ± 0.082	**0.561 ± 0.009**	0.682 ± 0.001	0.682 ± 0.001	0.454 ± 0.001
Falcon7bInstruct + Best	3.679 ± 0.162	**0.549 ± 0.017**	**0.68 ± 0.002**	**0.68 ± 0.002**	**0.453 ± 0.002**
Falcon7bInstruct + BayesPE	**3.554 ± 0.051**	0.542 ± 0.01	**0.678 ± 0.001**	**0.678 ± 0.001**	**0.452 ± 0.0**
Falcon40bInstruct + Standard	**1.248 ± 0.192**	**0.468 ± 0.031**	**0.227 ± 0.114**	**0.29 ± 0.163**	**0.255 ± 0.036**
Falcon40bInstruct + Ensemble	1.294 ± 0.061	**0.493 ± 0.018**	0.305 ± 0.047	**0.315 ± 0.048**	0.27 ± 0.014
Falcon40bInstruct + Best	**1.219 ± 0.202**	0.457 ± 0.032	**0.175 ± 0.13**	**0.196 ± 0.173**	**0.248 ± 0.039**
Falcon40bInstruct + BayesPE	**1.185 ± 0.024**	0.456 ± 0.009	**0.2 ± 0.032**	**0.262 ± 0.019**	**0.243 ± 0.006**
MPT7bInstruct + Standard	**1.22 ± 0.064**	**0.499 ± 0.017**	**0.226 ± 0.056**	**0.348 ± 0.178**	**0.251 ± 0.015**
MPT7bInstruct + Ensemble	1.208 ± 0.016	0.491 ± 0.01	0.224 ± 0.016	**0.318 ± 0.147**	0.249 ± 0.004
MPT7bInstruct + Best	**1.208 ± 0.065**	**0.503 ± 0.013**	**0.213 ± 0.059**	**0.328 ± 0.184**	**0.248 ± 0.015**
MPT7bInstruct + BayesPE	**1.19 ± 0.016**	0.489 ± 0.011	**0.209 ± 0.011**	**0.233 ± 0.022**	**0.245 ± 0.004**
MPT30bInstruct + Standard	**1.223 ± 0.079**	0.464 ± 0.015	**0.2 ± 0.069**	**0.505 ± 0.225**	**0.251 ± 0.018**
MPT30bInstruct + Ensemble	**1.17 ± 0.011**	0.462 ± 0.006	**0.152 ± 0.03**	**0.636 ± 0.173**	**0.239 ± 0.003**
MPT30bInstruct + Best	1.196 ± 0.01	**0.477 ± 0.005**	**0.158 ± 0.025**	0.712 ± 0.124	0.244 ± 0.002
MPT30bInstruct + BayesPE	**1.172 ± 0.009**	0.46 ± 0.005	**0.156 ± 0.021**	0.722 ± 0.014	**0.24 ± 0.002**
Mistral7bInstruct + Standard	**1.246 ± 0.177**	**0.517 ± 0.025**	**0.222 ± 0.107**	**0.366 ± 0.2**	**0.254 ± 0.034**
Mistral7bInstruct + Ensemble	1.258 ± 0.063	**0.532 ± 0.011**	0.275 ± 0.056	**0.38 ± 0.115**	0.261 ± 0.015
Mistral7bInstruct + Best	1.413 ± 0.245	**0.524 ± 0.03**	0.334 ± 0.134	**0.476 ± 0.118**	0.288 ± 0.047
Mistral7bInstruct + BayesPE	**1.178 ± 0.017**	**0.527 ± 0.009**	**0.196 ± 0.019**	**0.427 ± 0.235**	**0.242 ± 0.004**

Table 13: SNLI zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	**0.308 ± 0.029**	0.975 ± 0.002	**0.143 ± 0.013**	**0.166 ± 0.011**	**0.083 ± 0.011**
Falcon7bInstruct + Ensemble	**0.328 ± 0.012**	**0.978 ± 0.001**	0.176 ± 0.013	0.197 ± 0.013	**0.088 ± 0.004**
Falcon7bInstruct + Best	**0.309 ± 0.035**	**0.978 ± 0.001**	**0.151 ± 0.006**	0.184 ± 0.004	**0.083 ± 0.015**
Falcon7bInstruct + BayesPE	**0.326 ± 0.013**	**0.978 ± 0.001**	0.176 ± 0.006	0.194 ± 0.008	**0.088 ± 0.005**
Falcon40bInstruct + Standard	0.657 ± 0.118	**0.995 ± 0.002**	0.26 ± 0.07	0.726 ± 0.126	0.235 ± 0.041
Falcon40bInstruct + Ensemble	0.57 ± 0.078	**0.995 ± 0.0**	0.212 ± 0.046	0.719 ± 0.063	0.207 ± 0.026
Falcon40bInstruct + Best	**0.386 ± 0.062**	0.994 ± 0.001	**0.084 ± 0.044**	**0.283 ± 0.196**	**0.134 ± 0.026**
Falcon40bInstruct + BayesPE	**0.387 ± 0.041**	**0.996 ± 0.001**	**0.09 ± 0.03**	**0.274 ± 0.158**	**0.138 ± 0.018**
MPT7bInstruct + Standard	1.473 ± 0.113	0.935 ± 0.014	0.398 ± 0.005	**0.768 ± 0.185**	0.388 ± 0.008
MPT7bInstruct + Ensemble	1.402 ± 0.027	**0.947 ± 0.003**	0.396 ± 0.002	**0.803 ± 0.079**	**0.384 ± 0.003**
MPT7bInstruct + Best	**1.424 ± 0.097**	**0.941 ± 0.013**	**0.396 ± 0.004**	**0.789 ± 0.137**	**0.385 ± 0.007**
MPT7bInstruct + BayesPE	**1.37 ± 0.03**	**0.949 ± 0.004**	**0.394 ± 0.002**	**0.816 ± 0.067**	**0.381 ± 0.003**
MPT30bInstruct + Standard	1.748 ± 0.313	0.979 ± 0.012	0.394 ± 0.014	0.828 ± 0.015	0.377 ± 0.019
MPT30bInstruct + Ensemble	1.605 ± 0.076	**0.989 ± 0.003**	0.394 ± 0.004	0.827 ± 0.011	0.374 ± 0.007
MPT30bInstruct + Best	**1.596 ± 0.296**	0.979 ± 0.009	**0.373 ± 0.025**	**0.768 ± 0.075**	**0.358 ± 0.025**
MPT30bInstruct + BayesPE	**1.427 ± 0.036**	**0.99 ± 0.004**	**0.373 ± 0.012**	0.833 ± 0.009	**0.355 ± 0.005**
Mistral7bInstruct + Standard	**0.159 ± 0.044**	**0.99 ± 0.002**	**0.03 ± 0.01**	**0.192 ± 0.037**	0.047 ± 0.015
Mistral7bInstruct + Ensemble	**0.138 ± 0.012**	**0.991 ± 0.001**	**0.031 ± 0.014**	**0.142 ± 0.085**	**0.04 ± 0.005**
Mistral7bInstruct + Best	**0.127 ± 0.029**	**0.99 ± 0.004**	**0.03 ± 0.01**	0.22 ± 0.013	**0.034 ± 0.009**
Mistral7bInstruct + BayesPE	**0.135 ± 0.009**	**0.991 ± 0.001**	**0.026 ± 0.005**	**0.159 ± 0.072**	**0.038 ± 0.003**

Table 14: SST-2 zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	2.558 ± 0.291	0.67 ± 0.024	0.47 ± 0.108	**0.527 ± 0.084**	0.221 ± 0.026
Falcon7bInstruct + Ensemble	2.395 ± 0.063	**0.704 ± 0.009**	0.444 ± 0.032	**0.557 ± 0.028**	0.209 ± 0.006
Falcon7bInstruct + Best	**2.229 ± 0.022**	0.688 ± 0.012	**0.353 ± 0.019**	**0.538 ± 0.01**	**0.194 ± 0.002**
Falcon7bInstruct + BayesPE	**2.25 ± 0.111**	**0.7 ± 0.022**	**0.383 ± 0.045**	**0.535 ± 0.052**	**0.199 ± 0.007**
Falcon40bInstruct + Standard	**2.227 ± 0.847**	0.658 ± 0.019	**0.32 ± 0.167**	**0.903 ± 0.045**	**0.199 ± 0.041**
Falcon40bInstruct + Ensemble	2.035 ± 0.046	**0.669 ± 0.011**	0.34 ± 0.035	0.924 ± 0.006	0.194 ± 0.005
Falcon40bInstruct + Best	1.924 ± 0.047	0.664 ± 0.005	0.265 ± 0.039	**0.912 ± 0.027**	0.183 ± 0.006
Falcon40bInstruct + BayesPE	**1.883 ± 0.021**	**0.672 ± 0.004**	**0.233 ± 0.022**	**0.872 ± 0.052**	**0.178 ± 0.003**
MPT7bInstruct + Standard	**5.106 ± 0.457**	**0.711 ± 0.015**	**0.77 ± 0.045**	0.802 ± 0.046	**0.307 ± 0.017**
MPT7bInstruct + Ensemble	**4.968 ± 0.14**	**0.717 ± 0.006**	0.777 ± 0.012	0.792 ± 0.027	0.307 ± 0.006
MPT7bInstruct + Best	**5.053 ± 0.559**	**0.714 ± 0.016**	**0.753 ± 0.064**	**0.798 ± 0.051**	**0.302 ± 0.022**
MPT7bInstruct + BayesPE	**4.76 ± 0.264**	**0.717 ± 0.016**	**0.743 ± 0.044**	**0.767 ± 0.02**	**0.295 ± 0.014**
MPT30bInstruct + Standard	2.989 ± 0.091	0.775 ± 0.018	0.575 ± 0.037	0.7 ± 0.034	0.242 ± 0.011
MPT30bInstruct + Ensemble	2.898 ± 0.103	0.787 ± 0.006	0.572 ± 0.023	0.735 ± 0.019	0.242 ± 0.006
MPT30bInstruct + Best	2.912 ± 0.04	**0.794 ± 0.004**	**0.521 ± 0.012**	**0.661 ± 0.008**	**0.226 ± 0.002**
MPT30bInstruct + BayesPE	**2.67 ± 0.169**	**0.794 ± 0.005**	**0.516 ± 0.023**	**0.666 ± 0.02**	**0.222 ± 0.007**
Mistral7bInstruct + Standard	**1.238 ± 0.384**	**0.839 ± 0.078**	0.14 ± 0.074	0.191 ± 0.082	0.123 ± 0.027
Mistral7bInstruct + Ensemble	1.04 ± 0.026	0.872 ± 0.006	**0.076 ± 0.012**	**0.208 ± 0.142**	0.108 ± 0.002
Mistral7bInstruct + Best	1.026 ± 0.02	0.876 ± 0.001	0.118 ± 0.014	0.186 ± 0.026	0.107 ± 0.002
Mistral7bInstruct + BayesPE	**0.983 ± 0.013**	**0.878 ± 0.001**	**0.087 ± 0.013**	**0.121 ± 0.016**	**0.104 ± 0.002**

Table 15: TREC zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	3.081 ± 0.134	**0.577 ± 0.014**	**0.358 ± 0.058**	0.477 ± 0.122	**0.107 ± 0.004**
Falcon7bInstruct + Ensemble	3.083 ± 0.054	**0.584 ± 0.007**	0.378 ± 0.016	0.409 ± 0.015	0.108 ± 0.001
Falcon7bInstruct + Best	3.067 ± 0.097	**0.584 ± 0.008**	0.34 ± 0.002	**0.355 ± 0.05**	0.107 ± 0.001
Falcon7bInstruct + BayesPE	**2.972 ± 0.023**	**0.588 ± 0.007**	**0.333 ± 0.007**	0.544 ± 0.181	**0.105 ± 0.001**
Falcon40bInstruct + Standard	2.599 ± 0.063	**0.6 ± 0.044**	0.145 ± 0.018	**0.543 ± 0.206**	0.094 ± 0.001
Falcon40bInstruct + Ensemble	**2.551 ± 0.025**	**0.607 ± 0.014**	**0.098 ± 0.013**	**0.406 ± 0.208**	**0.093 ± 0.0**
Falcon40bInstruct + Best	2.598 ± 0.027	**0.604 ± 0.021**	0.156 ± 0.013	0.707 ± 0.122	0.094 ± 0.001
Falcon40bInstruct + BayesPE	**2.539 ± 0.018**	**0.602 ± 0.008**	**0.095 ± 0.015**	0.614 ± 0.125	**0.092 ± 0.0**
MPT7bInstruct + Standard	**2.846 ± 0.048**	0.645 ± 0.01	**0.205 ± 0.03**	**0.942 ± 0.026**	**0.095 ± 0.002**
MPT7bInstruct + Ensemble	**2.817 ± 0.014**	**0.655 ± 0.003**	**0.198 ± 0.016**	**0.953 ± 0.007**	**0.095 ± 0.001**
MPT7bInstruct + Best	2.851 ± 0.042	**0.656 ± 0.004**	**0.209 ± 0.023**	**0.947 ± 0.006**	0.095 ± 0.001
MPT7bInstruct + BayesPE	**2.816 ± 0.017**	**0.656 ± 0.002**	**0.208 ± 0.013**	**0.953 ± 0.006**	**0.095 ± 0.001**
MPT30bInstruct + Standard	3.081 ± 0.173	0.642 ± 0.02	0.421 ± 0.072	0.514 ± 0.062	0.11 ± 0.007
MPT30bInstruct + Ensemble	3.077 ± 0.082	**0.656 ± 0.008**	0.453 ± 0.032	0.536 ± 0.043	0.113 ± 0.003
MPT30bInstruct + Best	3.171 ± 0.082	**0.663 ± 0.009**	0.495 ± 0.054	0.585 ± 0.069	0.117 ± 0.005
MPT30bInstruct + BayesPE	**2.888 ± 0.094**	0.631 ± 0.011	**0.346 ± 0.022**	**0.457 ± 0.02**	**0.103 ± 0.002**
Mistral7bInstruct + Standard	2.215 ± 0.061	**0.693 ± 0.008**	0.179 ± 0.047	**0.334 ± 0.055**	0.084 ± 0.002
Mistral7bInstruct + Ensemble	**2.164 ± 0.035**	**0.696 ± 0.001**	0.165 ± 0.025	**0.336 ± 0.025**	0.082 ± 0.001
Mistral7bInstruct + Best	**2.135 ± 0.037**	0.695 ± 0.001	**0.123 ± 0.02**	**0.318 ± 0.024**	**0.081 ± 0.001**
Mistral7bInstruct + BayesPE	**2.154 ± 0.03**	0.694 ± 0.003	**0.139 ± 0.033**	**0.321 ± 0.031**	**0.082 ± 0.001**

Table 16: Yahoo Answers zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	**1.136 ± 0.083**	**0.243 ± 0.08**	**0.358 ± 0.038**	**0.623 ± 0.189**	**0.399 ± 0.017**
Falcon7bInstruct + Ensemble	1.188 ± 0.027	**0.227 ± 0.021**	**0.379 ± 0.01**	**0.741 ± 0.025**	0.411 ± 0.005
Falcon7bInstruct + Best	1.209 ± 0.045	**0.228 ± 0.051**	0.393 ± 0.006	0.77 ± 0.054	0.414 ± 0.008
Falcon7bInstruct + BayesPE	**1.131 ± 0.046**	**0.221 ± 0.03**	**0.367 ± 0.013**	**0.667 ± 0.106**	**0.399 ± 0.01**
Falcon40bInstruct + Standard	0.857 ± 0.026	**0.143 ± 0.029**	**0.314 ± 0.05**	**0.684 ± 0.061**	**0.326 ± 0.01**
Falcon40bInstruct + Ensemble	**0.841 ± 0.006**	**0.131 ± 0.008**	**0.32 ± 0.029**	**0.608 ± 0.171**	**0.321 ± 0.002**
Falcon40bInstruct + Best	0.865 ± 0.016	0.116 ± 0.016	0.367 ± 0.037	**0.698 ± 0.054**	0.331 ± 0.003
Falcon40bInstruct + BayesPE	**0.838 ± 0.005**	**0.127 ± 0.011**	**0.324 ± 0.029**	**0.601 ± 0.16**	**0.32 ± 0.002**
MPT7bInstruct + Standard	0.76 ± 0.053	0.585 ± 0.058	**0.149 ± 0.053**	0.458 ± 0.344	0.272 ± 0.019
MPT7bInstruct + Ensemble	0.752 ± 0.02	0.587 ± 0.02	0.173 ± 0.014	0.474 ± 0.356	0.271 ± 0.007
MPT7bInstruct + Best	**0.712 ± 0.016**	**0.638 ± 0.04**	**0.137 ± 0.032**	**0.127 ± 0.035**	**0.256 ± 0.007**
MPT7bInstruct + BayesPE	**0.722 ± 0.009**	**0.623 ± 0.02**	**0.16 ± 0.01**	**0.24 ± 0.222**	**0.26 ± 0.004**
MPT30bInstruct + Standard	0.684 ± 0.043	**0.722 ± 0.036**	**0.117 ± 0.038**	0.183 ± 0.071	0.232 ± 0.012
MPT30bInstruct + Ensemble	**0.623 ± 0.009**	**0.737 ± 0.011**	**0.089 ± 0.026**	0.118 ± 0.036	**0.214 ± 0.004**
MPT30bInstruct + Best	0.65 ± 0.042	**0.729 ± 0.014**	**0.092 ± 0.013**	**0.149 ± 0.094**	**0.218 ± 0.01**
MPT30bInstruct + BayesPE	**0.616 ± 0.018**	**0.738 ± 0.012**	**0.095 ± 0.02**	**0.087 ± 0.023**	**0.212 ± 0.006**
Mistral7bInstruct + Standard	**0.696 ± 0.045**	**0.614 ± 0.078**	**0.094 ± 0.036**	0.388 ± 0.213	**0.249 ± 0.02**
Mistral7bInstruct + Ensemble	**0.691 ± 0.011**	**0.591 ± 0.023**	**0.078 ± 0.025**	**0.138 ± 0.052**	**0.248 ± 0.005**
Mistral7bInstruct + Best	0.7 ± 0.0	**0.599 ± 0.0**	0.127 ± 0.0	**0.174 ± 0.0**	0.251 ± 0.0
Mistral7bInstruct + BayesPE	**0.69 ± 0.01**	**0.594 ± 0.023**	**0.076 ± 0.024**	**0.148 ± 0.046**	**0.247 ± 0.005**

Table 17: YouTube zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	1.067 ± 0.145	**0.22 ± 0.095**	**0.326 ± 0.079**	**0.624 ± 0.176**	0.384 ± 0.048
Falcon7bInstruct + Ensemble	0.964 ± 0.027	**0.185 ± 0.039**	**0.3 ± 0.018**	**0.565 ± 0.094**	0.357 ± 0.008
Falcon7bInstruct + Best	1.119 ± 0.028	**0.222 ± 0.087**	0.317 ± 0.025	0.696 ± 0.106	0.393 ± 0.003
Falcon7bInstruct + BayesPE	**0.878 ± 0.048**	**0.243 ± 0.102**	**0.269 ± 0.059**	**0.494 ± 0.187**	**0.33 ± 0.022**
Falcon40bInstruct + Standard	0.78 ± 0.034	0.388 ± 0.09	0.171 ± 0.069	**0.381 ± 0.081**	0.289 ± 0.016
Falcon40bInstruct + Ensemble	0.762 ± 0.014	0.378 ± 0.049	0.149 ± 0.041	0.5 ± 0.04	0.281 ± 0.007
Falcon40bInstruct + Best	**0.723 ± 0.009**	**0.485 ± 0.018**	**0.085 ± 0.007**	**0.376 ± 0.069**	**0.263 ± 0.004**
Falcon40bInstruct + BayesPE	0.741 ± 0.011	0.439 ± 0.035	0.11 ± 0.02	0.491 ± 0.065	0.271 ± 0.005
MPT7bInstruct + Standard	1.143 ± 0.156	**0.352 ± 0.085**	0.364 ± 0.043	**0.628 ± 0.218**	0.398 ± 0.026
MPT7bInstruct + Ensemble	1.073 ± 0.055	**0.346 ± 0.025**	0.348 ± 0.015	**0.602 ± 0.181**	0.385 ± 0.013
MPT7bInstruct + Best	**0.967 ± 0.035**	**0.328 ± 0.039**	**0.3 ± 0.017**	**0.476 ± 0.218**	**0.358 ± 0.012**
MPT7bInstruct + BayesPE	**0.988 ± 0.03**	**0.322 ± 0.022**	**0.308 ± 0.013**	**0.525 ± 0.253**	**0.364 ± 0.009**
MPT30bInstruct + Standard	0.55 ± 0.031	0.851 ± 0.011	0.078 ± 0.034	**0.143 ± 0.09**	0.185 ± 0.014
MPT30bInstruct + Ensemble	**0.519 ± 0.015**	**0.863 ± 0.005**	**0.045 ± 0.009**	**0.095 ± 0.027**	0.173 ± 0.007
MPT30bInstruct + Best	0.544 ± 0.021	0.849 ± 0.014	0.081 ± 0.039	0.158 ± 0.075	0.182 ± 0.009
MPT30bInstruct + BayesPE	**0.509 ± 0.005**	**0.861 ± 0.004**	0.066 ± 0.008	**0.098 ± 0.019**	**0.168 ± 0.002**
Mistral7bInstruct + Standard	**0.947 ± 0.03**	**0.296 ± 0.033**	**0.304 ± 0.007**	**0.606 ± 0.057**	**0.353 ± 0.006**
Mistral7bInstruct + Ensemble	0.942 ± 0.013	**0.299 ± 0.017**	0.318 ± 0.018	**0.562 ± 0.028**	0.354 ± 0.004
Mistral7bInstruct + Best	0.95 ± 0.011	**0.304 ± 0.038**	0.322 ± 0.005	0.778 ± 0.153	0.356 ± 0.002
Mistral7bInstruct + BayesPE	**0.926 ± 0.013**	**0.304 ± 0.018**	**0.305 ± 0.009**	**0.606 ± 0.159**	**0.349 ± 0.003**

Table 18: SMS zero-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	**0.222 ± 0.018**	**0.979 ± 0.004**	**0.068 ± 0.011**	**0.108 ± 0.021**	**0.06 ± 0.006**
Falcon7bInstruct + Ensemble	0.22 ± 0.006	**0.981 ± 0.001**	0.081 ± 0.006	**0.124 ± 0.027**	**0.058 ± 0.001**
Falcon7bInstruct + Best	**0.21 ± 0.011**	**0.982 ± 0.004**	**0.071 ± 0.012**	**0.118 ± 0.029**	**0.056 ± 0.005**
Falcon7bInstruct + BayesPE	**0.216 ± 0.005**	**0.981 ± 0.001**	**0.075 ± 0.006**	0.144 ± 0.034	**0.058 ± 0.002**
Falcon40bInstruct + Standard	0.194 ± 0.016	0.986 ± 0.002	**0.055 ± 0.016**	**0.113 ± 0.062**	0.054 ± 0.005
Falcon40bInstruct + Ensemble	0.184 ± 0.007	0.986 ± 0.0	**0.056 ± 0.008**	**0.112 ± 0.037**	0.051 ± 0.002
Falcon40bInstruct + Best	**0.168 ± 0.009**	**0.988 ± 0.001**	**0.049 ± 0.007**	**0.128 ± 0.072**	**0.05 ± 0.002**
Falcon40bInstruct + BayesPE	0.177 ± 0.004	0.987 ± 0.0	**0.05 ± 0.008**	**0.09 ± 0.016**	**0.049 ± 0.001**
MPT7bInstruct + Standard	0.295 ± 0.091	0.984 ± 0.002	0.04 ± 0.027	**0.131 ± 0.067**	0.09 ± 0.032
MPT7bInstruct + Ensemble	0.311 ± 0.03	**0.985 ± 0.0**	**0.021 ± 0.009**	**0.084 ± 0.067**	0.096 ± 0.011
MPT7bInstruct + Best	**0.222 ± 0.023**	0.982 ± 0.001	**0.023 ± 0.009**	**0.121 ± 0.075**	**0.064 ± 0.009**
MPT7bInstruct + BayesPE	**0.26 ± 0.05**	0.984 ± 0.001	0.033 ± 0.011	0.153 ± 0.065	**0.077 ± 0.018**
MPT30bInstruct + Standard	0.356 ± 0.07	0.981 ± 0.003	0.069 ± 0.011	**0.307 ± 0.088**	**0.083 ± 0.018**
MPT30bInstruct + Ensemble	0.294 ± 0.024	**0.986 ± 0.001**	0.061 ± 0.004	**0.29 ± 0.148**	**0.076 ± 0.004**
MPT30bInstruct + Best	**0.292 ± 0.053**	0.982 ± 0.002	**0.056 ± 0.014**	**0.292 ± 0.089**	**0.07 ± 0.01**
MPT30bInstruct + BayesPE	**0.267 ± 0.023**	**0.986 ± 0.001**	**0.056 ± 0.002**	**0.356 ± 0.124**	**0.07 ± 0.003**
Mistral7bInstruct + Standard	0.194 ± 0.006	0.981 ± 0.002	**0.029 ± 0.014**	0.161 ± 0.08	0.057 ± 0.002
Mistral7bInstruct + Ensemble	**0.175 ± 0.002**	0.982 ± 0.0	**0.021 ± 0.008**	**0.112 ± 0.032**	**0.052 ± 0.001**
Mistral7bInstruct + Best	**0.177 ± 0.008**	**0.983 ± 0.0**	0.028 ± 0.001	0.176 ± 0.015	**0.052 ± 0.002**
Mistral7bInstruct + BayesPE	**0.175 ± 0.002**	0.982 ± 0.0	**0.021 ± 0.007**	**0.082 ± 0.034**	**0.051 ± 0.001**

Table 19: Amazon Reviews few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	1.738 ± 0.32	0.973 ± 0.006	0.116 ± 0.046	0.191 ± 0.087	0.044 ± 0.006
Falcon7bInstruct + Ensemble	**1.337 ± 0.185**	**0.984 ± 0.003**	0.114 ± 0.032	0.206 ± 0.089	**0.036 ± 0.002**
Falcon7bInstruct + Best	**1.524 ± 0.306**	0.975 ± 0.007	**0.075 ± 0.018**	**0.106 ± 0.033**	**0.035 ± 0.003**
Falcon7bInstruct + BayesPE	**1.342 ± 0.249**	**0.981 ± 0.006**	**0.067 ± 0.028**	**0.118 ± 0.045**	**0.034 ± 0.002**
Falcon40bInstruct + Standard	0.63 ± 0.05	0.997 ± 0.001	**0.092 ± 0.022**	**0.189 ± 0.046**	0.019 ± 0.002
Falcon40bInstruct + Ensemble	0.59 ± 0.029	**0.998 ± 0.0**	0.122 ± 0.005	0.271 ± 0.018	0.018 ± 0.001
Falcon40bInstruct + Best	**0.511 ± 0.046**	0.998 ± 0.0	**0.09 ± 0.012**	0.251 ± 0.053	**0.016 ± 0.001**
Falcon40bInstruct + BayesPE	**0.511 ± 0.048**	**0.998 ± 0.0**	0.109 ± 0.016	0.294 ± 0.029	**0.016 ± 0.001**
MPT7bInstruct + Standard	0.684 ± 0.154	0.993 ± 0.002	**0.053 ± 0.025**	**0.122 ± 0.09**	0.021 ± 0.005
MPT7bInstruct + Ensemble	**0.484 ± 0.034**	**0.996 ± 0.0**	0.087 ± 0.016	0.224 ± 0.036	**0.016 ± 0.001**
MPT7bInstruct + Best	0.498 ± 0.033	0.995 ± 0.001	**0.04 ± 0.012**	**0.073 ± 0.013**	**0.017 ± 0.001**
MPT7bInstruct + BayesPE	**0.466 ± 0.026**	**0.996 ± 0.001**	0.055 ± 0.012	0.137 ± 0.025	**0.016 ± 0.001**
MPT30bInstruct + Standard	0.47 ± 0.09	0.995 ± 0.001	0.078 ± 0.026	0.204 ± 0.048	0.016 ± 0.003
MPT30bInstruct + Ensemble	**0.314 ± 0.018**	**0.997 ± 0.0**	0.039 ± 0.007	**0.158 ± 0.024**	**0.011 ± 0.001**
MPT30bInstruct + Best	0.445 ± 0.0	0.996 ± 0.0	0.077 ± 0.006	**0.153 ± 0.006**	0.017 ± 0.001
MPT30bInstruct + BayesPE	**0.314 ± 0.023**	0.996 ± 0.0	**0.029 ± 0.009**	**0.144 ± 0.056**	**0.01 ± 0.001**
Mistral7bInstruct + Standard	0.428 ± 0.046	0.995 ± 0.001	0.037 ± 0.006	**0.128 ± 0.048**	0.014 ± 0.002
Mistral7bInstruct + Ensemble	0.379 ± 0.024	**0.995 ± 0.0**	**0.03 ± 0.006**	0.154 ± 0.017	0.012 ± 0.001
Mistral7bInstruct + Best	**0.403 ± 0.079**	0.994 ± 0.001	**0.031 ± 0.01**	**0.114 ± 0.049**	**0.013 ± 0.002**
Mistral7bInstruct + BayesPE	**0.356 ± 0.021**	**0.995 ± 0.0**	**0.032 ± 0.008**	**0.14 ± 0.038**	**0.012 ± 0.001**

Table 20: DBPedia 14 few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	**0.918 ± 0.449**	**0.399 ± 0.054**	**0.179 ± 0.111**	**0.351 ± 0.055**	**0.3 ± 0.058**
Falcon7bInstruct + Ensemble	**0.77 ± 0.047**	0.341 ± 0.024	**0.163 ± 0.034**	0.418 ± 0.112	**0.283 ± 0.017**
Falcon7bInstruct + Best	**0.743 ± 0.024**	**0.412 ± 0.051**	**0.125 ± 0.054**	**0.327 ± 0.05**	**0.273 ± 0.012**
Falcon7bInstruct + BayesPE	**0.757 ± 0.018**	0.333 ± 0.038	0.181 ± 0.051	**0.347 ± 0.132**	**0.281 ± 0.008**
Falcon40bInstruct + Standard	0.997 ± 0.063	**0.224 ± 0.025**	0.344 ± 0.034	**0.683 ± 0.128**	0.372 ± 0.016
Falcon40bInstruct + Ensemble	**0.875 ± 0.021**	0.212 ± 0.009	**0.314 ± 0.018**	0.665 ± 0.045	**0.334 ± 0.009**
Falcon40bInstruct + Best	0.967 ± 0.125	**0.237 ± 0.025**	**0.33 ± 0.038**	0.673 ± 0.098	0.362 ± 0.035
Falcon40bInstruct + BayesPE	0.906 ± 0.014	0.209 ± 0.007	**0.306 ± 0.019**	**0.594 ± 0.047**	0.346 ± 0.006
MPT7bInstruct + Standard	**1.039 ± 0.378**	**0.485 ± 0.078**	0.309 ± 0.098	0.378 ± 0.127	0.355 ± 0.065
MPT7bInstruct + Ensemble	0.945 ± 0.033	**0.511 ± 0.03**	0.318 ± 0.018	0.33 ± 0.013	0.352 ± 0.011
MPT7bInstruct + Best	**0.787 ± 0.066**	**0.461 ± 0.064**	**0.19 ± 0.058**	**0.262 ± 0.031**	**0.292 ± 0.024**
MPT7bInstruct + BayesPE	0.869 ± 0.072	**0.509 ± 0.051**	0.269 ± 0.048	0.301 ± 0.037	0.324 ± 0.025
MPT30bInstruct + Standard	2.236 ± 0.559	0.329 ± 0.034	0.434 ± 0.041	0.651 ± 0.027	0.489 ± 0.037
MPT30bInstruct + Ensemble	**1.27 ± 0.122**	0.328 ± 0.012	**0.368 ± 0.024**	0.678 ± 0.024	**0.42 ± 0.018**
MPT30bInstruct + Best	1.58 ± 0.231	**0.386 ± 0.016**	**0.39 ± 0.045**	**0.609 ± 0.042**	**0.437 ± 0.031**
MPT30bInstruct + BayesPE	**1.253 ± 0.149**	0.348 ± 0.01	**0.354 ± 0.038**	**0.635 ± 0.025**	**0.41 ± 0.025**
Mistral7bInstruct + Standard	1.078 ± 0.058	**0.199 ± 0.013**	0.409 ± 0.044	0.786 ± 0.154	0.407 ± 0.019
Mistral7bInstruct + Ensemble	**0.966 ± 0.04**	0.188 ± 0.003	**0.368 ± 0.019**	**0.61 ± 0.104**	**0.371 ± 0.014**
Mistral7bInstruct + Best	1.164 ± 0.086	**0.19 ± 0.012**	0.418 ± 0.041	0.756 ± 0.048	0.432 ± 0.027
Mistral7bInstruct + BayesPE	**0.95 ± 0.059**	**0.19 ± 0.007**	**0.358 ± 0.016**	**0.689 ± 0.147**	**0.363 ± 0.022**

Table 21: Glue (mrpc) few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	**0.434 ± 0.055**	**0.974 ± 0.011**	**0.102 ± 0.062**	**0.151 ± 0.049**	**0.138 ± 0.022**
Falcon7bInstruct + Ensemble	**0.431 ± 0.015**	**0.977 ± 0.002**	0.215 ± 0.058	0.225 ± 0.048	**0.128 ± 0.008**
Falcon7bInstruct + Best	**0.429 ± 0.063**	0.965 ± 0.012	0.194 ± 0.064	0.219 ± 0.048	**0.131 ± 0.022**
Falcon7bInstruct + BayesPE	**0.429 ± 0.026**	**0.977 ± 0.002**	0.24 ± 0.02	0.245 ± 0.019	**0.127 ± 0.01**
Falcon40bInstruct + Standard	1.515 ± 0.347	0.973 ± 0.003	0.431 ± 0.052	0.796 ± 0.048	0.405 ± 0.053
Falcon40bInstruct + Ensemble	1.351 ± 0.088	**0.98 ± 0.001**	0.434 ± 0.012	0.812 ± 0.017	0.394 ± 0.014
Falcon40bInstruct + Best	**1.136 ± 0.214**	0.975 ± 0.002	**0.373 ± 0.035**	**0.748 ± 0.028**	**0.345 ± 0.033**
Falcon40bInstruct + BayesPE	**1.072 ± 0.05**	0.978 ± 0.002	**0.371 ± 0.018**	**0.758 ± 0.031**	**0.339 ± 0.011**
MPT7bInstruct + Standard	**1.392 ± 0.238**	**0.945 ± 0.017**	**0.447 ± 0.021**	**0.831 ± 0.034**	**0.419 ± 0.03**
MPT7bInstruct + Ensemble	1.487 ± 0.09	**0.955 ± 0.004**	0.46 ± 0.007	**0.852 ± 0.008**	0.437 ± 0.01
MPT7bInstruct + Best	1.577 ± 0.3	**0.944 ± 0.019**	0.46 ± 0.018	**0.802 ± 0.106**	0.437 ± 0.027
MPT7bInstruct + BayesPE	**1.216 ± 0.123**	**0.955 ± 0.006**	**0.432 ± 0.012**	**0.812 ± 0.01**	**0.397 ± 0.016**
MPT30bInstruct + Standard	2.112 ± 0.598	0.966 ± 0.006	0.375 ± 0.078	0.536 ± 0.201	0.374 ± 0.068
MPT30bInstruct + Ensemble	1.425 ± 0.119	0.972 ± 0.001	0.355 ± 0.034	0.677 ± 0.054	0.341 ± 0.02
MPT30bInstruct + Best	**1.356 ± 0.347**	**0.973 ± 0.004**	**0.277 ± 0.071**	**0.342 ± 0.163**	**0.289 ± 0.059**
MPT30bInstruct + BayesPE	**1.203 ± 0.157**	**0.974 ± 0.001**	**0.26 ± 0.056**	**0.333 ± 0.162**	**0.277 ± 0.041**
Mistral7bInstruct + Standard	0.21 ± 0.009	**0.984 ± 0.002**	0.032 ± 0.009	**0.189 ± 0.106**	0.061 ± 0.002
Mistral7bInstruct + Ensemble	**0.167 ± 0.004**	**0.985 ± 0.001**	**0.022 ± 0.006**	**0.223 ± 0.073**	**0.052 ± 0.001**
Mistral7bInstruct + Best	0.193 ± 0.01	0.983 ± 0.001	0.036 ± 0.005	0.288 ± 0.037	0.06 ± 0.001
Mistral7bInstruct + BayesPE	**0.164 ± 0.006**	**0.985 ± 0.001**	**0.025 ± 0.006**	**0.246 ± 0.094**	**0.052 ± 0.002**

Table 22: Imdb few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	3.823 ± 0.148	**0.451 ± 0.01**	0.682 ± 0.002	0.682 ± 0.002	0.454 ± 0.001
Falcon7bInstruct + Ensemble	3.76 ± 0.036	**0.453 ± 0.004**	0.681 ± 0.0	0.681 ± 0.0	0.454 ± 0.0
Falcon7bInstruct + Best	**3.734 ± 0.107**	**0.458 ± 0.011**	**0.68 ± 0.001**	**0.68 ± 0.001**	**0.454 ± 0.001**
Falcon7bInstruct + BayesPE	**3.697 ± 0.016**	**0.452 ± 0.005**	**0.68 ± 0.0**	**0.68 ± 0.0**	**0.453 ± 0.0**
Falcon40bInstruct + Standard	1.575 ± 0.07	**0.483 ± 0.016**	0.452 ± 0.025	0.534 ± 0.084	0.326 ± 0.012
Falcon40bInstruct + Ensemble	1.565 ± 0.036	0.478 ± 0.006	0.451 ± 0.014	0.578 ± 0.103	0.325 ± 0.006
Falcon40bInstruct + Best	1.57 ± 0.071	**0.493 ± 0.013**	0.451 ± 0.026	**0.522 ± 0.093**	0.325 ± 0.012
Falcon40bInstruct + BayesPE	**1.474 ± 0.055**	**0.483 ± 0.014**	**0.411 ± 0.025**	**0.46 ± 0.038**	**0.308 ± 0.011**
MPT7bInstruct + Standard	1.764 ± 0.14	**0.408 ± 0.017**	0.505 ± 0.051	0.829 ± 0.152	0.355 ± 0.022
MPT7bInstruct + Ensemble	1.653 ± 0.055	**0.405 ± 0.006**	0.475 ± 0.019	0.874 ± 0.129	0.34 ± 0.009
MPT7bInstruct + Best	1.682 ± 0.186	**0.414 ± 0.014**	0.48 ± 0.056	**0.667 ± 0.188**	0.342 ± 0.027
MPT7bInstruct + BayesPE	**1.522 ± 0.08**	**0.41 ± 0.004**	**0.423 ± 0.033**	**0.621 ± 0.194**	**0.317 ± 0.014**
MPT30bInstruct + Standard	1.493 ± 0.127	**0.461 ± 0.045**	0.369 ± 0.05	**0.65 ± 0.204**	0.304 ± 0.021
MPT30bInstruct + Ensemble	1.374 ± 0.02	0.458 ± 0.016	**0.337 ± 0.014**	**0.521 ± 0.039**	0.286 ± 0.004
MPT30bInstruct + Best	**1.352 ± 0.052**	**0.476 ± 0.018**	**0.31 ± 0.038**	**0.643 ± 0.183**	**0.28 ± 0.011**
MPT30bInstruct + BayesPE	**1.338 ± 0.025**	**0.469 ± 0.012**	**0.31 ± 0.022**	**0.525 ± 0.061**	**0.278 ± 0.006**
Mistral7bInstruct + Standard	**1.264 ± 0.065**	**0.488 ± 0.023**	**0.245 ± 0.068**	**0.562 ± 0.178**	**0.26 ± 0.015**
Mistral7bInstruct + Ensemble	1.262 ± 0.019	**0.484 ± 0.008**	0.26 ± 0.018	0.528 ± 0.038	**0.261 ± 0.004**
Mistral7bInstruct + Best	**1.224 ± 0.065**	**0.491 ± 0.017**	**0.204 ± 0.07**	**0.455 ± 0.042**	**0.252 ± 0.015**
Mistral7bInstruct + BayesPE	**1.224 ± 0.039**	**0.486 ± 0.008**	**0.219 ± 0.04**	0.568 ± 0.032	**0.252 ± 0.009**

Table 23: SNLI few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	0.414 ± 0.109	0.984 ± 0.001	**0.102 ± 0.046**	**0.223 ± 0.106**	0.129 ± 0.048
Falcon7bInstruct + Ensemble	0.389 ± 0.042	**0.986 ± 0.001**	**0.094 ± 0.025**	**0.148 ± 0.033**	0.121 ± 0.019
Falcon7bInstruct + Best	**0.308 ± 0.035**	0.984 ± 0.0	0.133 ± 0.028	0.198 ± 0.032	**0.084 ± 0.015**
Falcon7bInstruct + BayesPE	**0.295 ± 0.024**	**0.986 ± 0.001**	0.144 ± 0.009	0.197 ± 0.018	**0.078 ± 0.01**
Falcon40bInstruct + Standard	0.869 ± 0.181	**0.994 ± 0.002**	0.307 ± 0.028	0.772 ± 0.029	0.276 ± 0.03
Falcon40bInstruct + Ensemble	0.804 ± 0.063	0.994 ± 0.0	0.3 ± 0.018	0.773 ± 0.03	0.267 ± 0.016
Falcon40bInstruct + Best	**0.693 ± 0.103**	**0.994 ± 0.0**	**0.252 ± 0.064**	**0.684 ± 0.148**	**0.234 ± 0.035**
Falcon40bInstruct + BayesPE	**0.667 ± 0.081**	**0.994 ± 0.0**	**0.233 ± 0.047**	**0.676 ± 0.064**	**0.227 ± 0.028**
MPT7bInstruct + Standard	1.885 ± 0.568	**0.992 ± 0.002**	0.405 ± 0.011	**0.717 ± 0.196**	0.393 ± 0.021
MPT7bInstruct + Ensemble	1.632 ± 0.179	**0.992 ± 0.002**	0.404 ± 0.007	**0.776 ± 0.184**	0.39 ± 0.012
MPT7bInstruct + Best	1.696 ± 0.621	0.99 ± 0.002	0.388 ± 0.028	**0.699 ± 0.186**	0.371 ± 0.04
MPT7bInstruct + BayesPE	**1.137 ± 0.08**	0.99 ± 0.002	**0.362 ± 0.013**	**0.801 ± 0.014**	**0.333 ± 0.014**
MPT30bInstruct + Standard	1.354 ± 0.307	0.989 ± 0.001	0.265 ± 0.063	0.448 ± 0.141	0.266 ± 0.05
MPT30bInstruct + Ensemble	1.144 ± 0.082	**0.992 ± 0.0**	0.253 ± 0.021	0.528 ± 0.098	0.258 ± 0.012
MPT30bInstruct + Best	**0.953 ± 0.225**	0.99 ± 0.001	**0.176 ± 0.031**	**0.244 ± 0.078**	**0.198 ± 0.028**
MPT30bInstruct + BayesPE	**0.892 ± 0.119**	0.991 ± 0.001	**0.173 ± 0.034**	**0.252 ± 0.13**	**0.198 ± 0.029**
Mistral7bInstruct + Standard	0.15 ± 0.034	0.988 ± 0.002	**0.02 ± 0.005**	0.177 ± 0.065	0.04 ± 0.009
Mistral7bInstruct + Ensemble	**0.128 ± 0.008**	0.99 ± 0.001	0.033 ± 0.008	**0.109 ± 0.019**	0.033 ± 0.002
Mistral7bInstruct + Best	**0.112 ± 0.023**	**0.992 ± 0.002**	**0.021 ± 0.005**	0.178 ± 0.066	**0.028 ± 0.007**
Mistral7bInstruct + BayesPE	**0.125 ± 0.008**	**0.991 ± 0.001**	0.039 ± 0.006	0.206 ± 0.067	**0.031 ± 0.002**

Table 24: SST-2 few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	2.583 ± 0.603	0.769 ± 0.03	0.432 ± 0.181	0.619 ± 0.241	0.211 ± 0.039
Falcon7bInstruct + Ensemble	1.881 ± 0.135	**0.798 ± 0.008**	0.313 ± 0.05	0.744 ± 0.229	0.178 ± 0.009
Falcon7bInstruct + Best	1.806 ± 0.229	**0.801 ± 0.008**	**0.134 ± 0.104**	**0.322 ± 0.328**	**0.151 ± 0.015**
Falcon7bInstruct + BayesPE	**1.603 ± 0.166**	0.791 ± 0.004	0.228 ± 0.07	0.704 ± 0.278	**0.162 ± 0.013**
Falcon40bInstruct + Standard	1.405 ± 0.156	0.856 ± 0.025	0.167 ± 0.061	0.235 ± 0.068	0.141 ± 0.016
Falcon40bInstruct + Ensemble	1.371 ± 0.059	0.859 ± 0.007	0.128 ± 0.021	0.304 ± 0.098	0.137 ± 0.006
Falcon40bInstruct + Best	**1.205 ± 0.018**	**0.876 ± 0.014**	**0.09 ± 0.019**	0.14 ± 0.0	**0.118 ± 0.005**
Falcon40bInstruct + BayesPE	**1.199 ± 0.006**	**0.881 ± 0.002**	**0.083 ± 0.013**	**0.134 ± 0.005**	**0.118 ± 0.001**
MPT7bInstruct + Standard	2.136 ± 0.708	0.836 ± 0.027	0.361 ± 0.111	0.535 ± 0.202	0.18 ± 0.042
MPT7bInstruct + Ensemble	1.52 ± 0.139	0.848 ± 0.007	0.268 ± 0.034	0.418 ± 0.117	0.155 ± 0.009
MPT7bInstruct + Best	1.591 ± 0.41	**0.856 ± 0.017**	0.282 ± 0.066	**0.375 ± 0.126**	0.15 ± 0.025
MPT7bInstruct + BayesPE	**1.255 ± 0.117**	**0.868 ± 0.007**	**0.222 ± 0.025**	**0.3 ± 0.016**	**0.131 ± 0.009**
MPT30bInstruct + Standard	2.399 ± 0.564	0.875 ± 0.013	0.382 ± 0.059	0.446 ± 0.078	0.169 ± 0.022
MPT30bInstruct + Ensemble	**1.218 ± 0.09**	**0.892 ± 0.006**	0.177 ± 0.032	0.275 ± 0.044	0.116 ± 0.009
MPT30bInstruct + Best	**1.294 ± 0.223**	**0.891 ± 0.009**	0.178 ± 0.048	**0.238 ± 0.08**	**0.103 ± 0.016**
MPT30bInstruct + BayesPE	**1.155 ± 0.178**	**0.893 ± 0.007**	**0.104 ± 0.056**	**0.178 ± 0.055**	**0.1 ± 0.014**
Mistral7bInstruct + Standard	**0.874 ± 0.116**	**0.907 ± 0.008**	**0.055 ± 0.024**	**0.08 ± 0.032**	0.082 ± 0.007
Mistral7bInstruct + Ensemble	**0.813 ± 0.027**	**0.912 ± 0.004**	**0.048 ± 0.017**	**0.088 ± 0.016**	**0.077 ± 0.002**
Mistral7bInstruct + Best	0.893 ± 0.08	0.906 ± 0.007	**0.043 ± 0.025**	**0.077 ± 0.029**	0.082 ± 0.006
Mistral7bInstruct + BayesPE	**0.802 ± 0.023**	**0.911 ± 0.002**	**0.044 ± 0.018**	**0.086 ± 0.015**	**0.076 ± 0.002**

Table 25: TREC few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	2.717 ± 0.236	**0.665 ± 0.02**	0.208 ± 0.079	0.512 ± 0.278	0.095 ± 0.006
Falcon7bInstruct + Ensemble	2.641 ± 0.067	**0.67 ± 0.01**	**0.125 ± 0.024**	**0.132 ± 0.06**	0.091 ± 0.002
Falcon7bInstruct + Best	**2.509 ± 0.044**	0.67 ± 0.006	**0.148 ± 0.046**	0.555 ± 0.363	**0.09 ± 0.001**
Falcon7bInstruct + BayesPE	**2.517 ± 0.077**	**0.676 ± 0.002**	0.152 ± 0.017	0.627 ± 0.342	**0.089 ± 0.001**
Falcon40bInstruct + Standard	2.601 ± 0.207	0.678 ± 0.01	0.227 ± 0.093	0.341 ± 0.083	0.094 ± 0.005
Falcon40bInstruct + Ensemble	**2.434 ± 0.088**	**0.69 ± 0.004**	0.196 ± 0.029	**0.248 ± 0.067**	0.09 ± 0.002
Falcon40bInstruct + Best	2.653 ± 0.146	0.68 ± 0.008	0.188 ± 0.052	**0.264 ± 0.043**	0.092 ± 0.002
Falcon40bInstruct + BayesPE	**2.343 ± 0.138**	**0.691 ± 0.008**	**0.138 ± 0.042**	**0.252 ± 0.032**	**0.087 ± 0.002**
MPT7bInstruct + Standard	3.457 ± 0.219	0.656 ± 0.013	0.415 ± 0.071	**0.561 ± 0.123**	0.112 ± 0.007
MPT7bInstruct + Ensemble	**3.09 ± 0.066**	**0.678 ± 0.008**	**0.318 ± 0.042**	**0.51 ± 0.08**	**0.102 ± 0.003**
MPT7bInstruct + Best	3.366 ± 0.086	0.652 ± 0.01	0.383 ± 0.032	0.55 ± 0.086	0.108 ± 0.004
MPT7bInstruct + BayesPE	**3.092 ± 0.116**	**0.676 ± 0.008**	**0.325 ± 0.064**	**0.48 ± 0.044**	**0.102 ± 0.005**
MPT30bInstruct + Standard	4.024 ± 0.397	0.687 ± 0.012	0.462 ± 0.055	0.527 ± 0.078	0.11 ± 0.007
MPT30bInstruct + Ensemble	**3.279 ± 0.172**	**0.7 ± 0.008**	**0.328 ± 0.073**	**0.391 ± 0.061**	**0.098 ± 0.004**
MPT30bInstruct + Best	4.512 ± 0.628	0.683 ± 0.017	0.45 ± 0.081	0.494 ± 0.079	0.112 ± 0.01
MPT30bInstruct + BayesPE	**3.223 ± 0.076**	0.69 ± 0.006	**0.271 ± 0.071**	**0.41 ± 0.035**	**0.096 ± 0.004**
Mistral7bInstruct + Standard	2.538 ± 0.4	0.692 ± 0.025	0.186 ± 0.066	0.236 ± 0.086	0.086 ± 0.006
Mistral7bInstruct + Ensemble	2.175 ± 0.052	**0.713 ± 0.004**	**0.101 ± 0.014**	**0.144 ± 0.04**	**0.08 ± 0.001**
Mistral7bInstruct + Best	**2.151 ± 0.046**	**0.709 ± 0.008**	0.119 ± 0.012	**0.157 ± 0.033**	**0.08 ± 0.001**
Mistral7bInstruct + BayesPE	**2.117 ± 0.053**	**0.713 ± 0.006**	**0.105 ± 0.015**	**0.151 ± 0.044**	**0.079 ± 0.001**

Table 26: Yahoo Answers few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	0.995 ± 0.258	0.675 ± 0.056	**0.319 ± 0.108**	0.404 ± 0.143	0.348 ± 0.066
Falcon7bInstruct + Ensemble	0.989 ± 0.086	**0.735 ± 0.02**	0.352 ± 0.032	0.473 ± 0.14	0.361 ± 0.025
Falcon7bInstruct + Best	**0.892 ± 0.279**	0.639 ± 0.104	**0.214 ± 0.167**	**0.275 ± 0.23**	**0.309 ± 0.085**
Falcon7bInstruct + BayesPE	**0.794 ± 0.112**	0.711 ± 0.029	**0.211 ± 0.115**	**0.236 ± 0.129**	**0.291 ± 0.046**
Falcon40bInstruct + Standard	1.245 ± 0.173	**0.155 ± 0.038**	**0.424 ± 0.088**	0.914 ± 0.013	0.451 ± 0.053
Falcon40bInstruct + Ensemble	**1.085 ± 0.051**	0.097 ± 0.01	0.483 ± 0.022	0.903 ± 0.058	**0.417 ± 0.017**
Falcon40bInstruct + Best	**1.105 ± 0.076**	**0.174 ± 0.018**	**0.373 ± 0.04**	**0.803 ± 0.084**	**0.408 ± 0.021**
Falcon40bInstruct + BayesPE	**1.053 ± 0.059**	0.122 ± 0.011	0.427 ± 0.033	**0.888 ± 0.092**	**0.401 ± 0.018**
MPT7bInstruct + Standard	0.874 ± 0.156	**0.452 ± 0.184**	0.224 ± 0.093	0.51 ± 0.228	0.31 ± 0.044
MPT7bInstruct + Ensemble	0.757 ± 0.025	**0.525 ± 0.09**	0.17 ± 0.03	**0.236 ± 0.057**	0.279 ± 0.01
MPT7bInstruct + Best	**0.717 ± 0.077**	**0.578 ± 0.053**	**0.098 ± 0.062**	**0.28 ± 0.115**	**0.257 ± 0.026**
MPT7bInstruct + BayesPE	**0.692 ± 0.017**	**0.574 ± 0.049**	**0.069 ± 0.036**	**0.253 ± 0.126**	**0.249 ± 0.008**
MPT30bInstruct + Standard	1.143 ± 0.189	**0.526 ± 0.127**	0.324 ± 0.07	0.464 ± 0.093	0.36 ± 0.041
MPT30bInstruct + Ensemble	**0.762 ± 0.056**	0.483 ± 0.044	**0.165 ± 0.047**	**0.325 ± 0.132**	**0.278 ± 0.02**
MPT30bInstruct + Best	1.221 ± 0.252	0.388 ± 0.098	0.354 ± 0.108	0.552 ± 0.149	0.391 ± 0.065
MPT30bInstruct + BayesPE	**0.76 ± 0.056**	**0.537 ± 0.055**	**0.158 ± 0.039**	**0.291 ± 0.087**	**0.276 ± 0.019**
Mistral7bInstruct + Standard	0.692 ± 0.017	0.668 ± 0.029	0.092 ± 0.038	**0.511 ± 0.249**	0.239 ± 0.011
Mistral7bInstruct + Ensemble	**0.648 ± 0.019**	**0.711 ± 0.036**	**0.058 ± 0.024**	**0.519 ± 0.413**	**0.227 ± 0.008**
Mistral7bInstruct + Best	**0.661 ± 0.036**	**0.691 ± 0.03**	**0.063 ± 0.005**	**0.354 ± 0.224**	**0.227 ± 0.012**
Mistral7bInstruct + BayesPE	**0.63 ± 0.028**	**0.713 ± 0.028**	**0.074 ± 0.016**	**0.33 ± 0.348**	**0.218 ± 0.011**

Table 27: YouTube few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

Model+Method	NLL (↓)	AUC (↑)	ECE (↓)	MCE (↓)	Brier (↓)
Falcon7bInstruct + Standard	1.056 ± 0.154	**0.596 ± 0.113**	0.35 ± 0.042	**0.607 ± 0.046**	0.366 ± 0.037
Falcon7bInstruct + Ensemble	1.031 ± 0.046	0.601 ± 0.042	0.35 ± 0.014	**0.6 ± 0.016**	0.366 ± 0.011
Falcon7bInstruct + Best	1.14 ± 0.13	0.481 ± 0.13	0.364 ± 0.042	0.661 ± 0.051	0.386 ± 0.028
Falcon7bInstruct + BayesPE	**0.885 ± 0.012**	**0.677 ± 0.045**	**0.299 ± 0.009**	**0.626 ± 0.053**	**0.324 ± 0.003**
Falcon40bInstruct + Standard	1.219 ± 0.255	0.208 ± 0.238	0.438 ± 0.12	0.727 ± 0.216	0.439 ± 0.082
Falcon40bInstruct + Ensemble	**1.003 ± 0.066**	0.085 ± 0.026	0.417 ± 0.068	0.702 ± 0.11	0.386 ± 0.025
Falcon40bInstruct + Best	**0.95 ± 0.085**	**0.415 ± 0.159**	**0.298 ± 0.041**	**0.468 ± 0.122**	**0.347 ± 0.029**
Falcon40bInstruct + BayesPE	**0.985 ± 0.14**	0.196 ± 0.164	**0.362 ± 0.139**	0.68 ± 0.173	**0.374 ± 0.061**
MPT7bInstruct + Standard	1.035 ± 0.149	0.468 ± 0.091	0.33 ± 0.058	0.424 ± 0.111	0.368 ± 0.036
MPT7bInstruct + Ensemble	0.894 ± 0.056	0.529 ± 0.097	0.286 ± 0.024	0.317 ± 0.036	0.332 ± 0.019
MPT7bInstruct + Best	**0.625 ± 0.091**	**0.818 ± 0.134**	**0.064 ± 0.069**	**0.169 ± 0.045**	**0.218 ± 0.039**
MPT7bInstruct + BayesPE	**0.715 ± 0.138**	**0.695 ± 0.186**	**0.118 ± 0.113**	**0.195 ± 0.075**	**0.256 ± 0.057**
MPT30bInstruct + Standard	1.111 ± 0.356	**0.554 ± 0.173**	0.325 ± 0.103	0.444 ± 0.146	0.353 ± 0.074
MPT30bInstruct + Ensemble	0.722 ± 0.035	0.528 ± 0.073	**0.128 ± 0.056**	**0.178 ± 0.104**	0.264 ± 0.017
MPT30bInstruct + Best	0.788 ± 0.1	**0.614 ± 0.133**	0.216 ± 0.042	0.3 ± 0.076	0.276 ± 0.046
MPT30bInstruct + BayesPE	**0.618 ± 0.081**	**0.716 ± 0.156**	**0.092 ± 0.052**	**0.155 ± 0.099**	**0.215 ± 0.038**
Mistral7bInstruct + Standard	0.807 ± 0.072	0.426 ± 0.156	0.199 ± 0.096	0.678 ± 0.288	0.297 ± 0.034
Mistral7bInstruct + Ensemble	0.709 ± 0.03	0.549 ± 0.079	**0.082 ± 0.046**	0.323 ± 0.093	0.257 ± 0.015
Mistral7bInstruct + Best	**0.668 ± 0.036**	**0.639 ± 0.064**	0.078 ± 0.008	**0.159 ± 0.103**	**0.237 ± 0.017**
Mistral7bInstruct + BayesPE	**0.667 ± 0.021**	**0.652 ± 0.041**	**0.06 ± 0.02**	**0.197 ± 0.126**	**0.237 ± 0.01**

Table 28: SMS few-shot classification. Probabilistic metrics evaluated on test set for proposed BayesPE approach and baselines with different LLMs.

NLL	AUC	ECE	MCE	Brier
Standard				
21(0)	26(0)	28(0)	33(1)	21(0)
Ensemble				
17(0)	36(4)	19(1)	31(0)	16(1)
Best				
24(1)	34(3)	32(1)	**38(3)**	31(1)
BayesPE				
49(17)	**37(2)**	**44(7)**	37(5)	**48(9)**

Table 29: Summary of zero-shot results obtained from our extensive benchmark evaluation (50 total experiments).

NLL	AUC	ECE	MCE	Brier
Standard				
7(0)	22(0)	14(1)	20(2)	6(0)
Ensemble				
20(1)	30(6)	20(0)	24(3)	20(1)
Best				
31(1)	2(3)	29(4)	30(2)	34(1)
BayesPE				
47(8)	**35(2)**	**39(9)**	**38(2)**	**48(8)**

Table 30: Summary of few-shot results obtained from our extensive benchmark evaluation (50 total experiments).

NLL	AUC	ECE	MCE	Brier
Standard				
8(0)	11(0)	11(1)	11(2)	9(0)
Ensemble				
6(0)	**16(2)**	6(0)	8(2)	7(0)
Best				
13(0)	11(1)	**13(1)**	11(3)	16(0)
BayesPE				
20(6)	14(2)	**13(3)**	**12(1)**	**20(3)**

Table 31: Summary of results for Falcon7bInstruct (20 total experiments).

NLL	AUC	ECE	MCE	Brier
Standard				
4(0)	10(0)	9(0)	11(1)	5(0)
Ensemble				
8(1)	**11(3)**	5(0)	9(2)	6(1)
Best				
13(2)	**11(3)**	14(1)	16(1)	13(1)
BayesPE				
17(4)	10(0)	**16(4)**	**15(2)**	**18(5)**

Table 32: Summary of results for Falcon40bInstruct (20 total experiments).

NLL	AUC	ECE	MCE	Brier
Standard				
7(0)	10(0)	8(0)	12(0)	6(0)
Ensemble				
4(0)	14(1)	4(0)	12(0)	4(0)
Best				
11(1)	15(0)	14(1)	**19(1)**	13(1)
BayesPE				
19(6)	**17(0)**	**16(4)**	14(1)	**19(3)**

Table 33: Summary of results for MPT7bInstruct (20 total experiments).

NLL	AUC	ECE	MCE	Brier
Standard				
1(0)	5(0)	2(0)	6(0)	2(0)
Ensemble				
9(0)	12(3)	9(1)	11(0)	8(0)
Best				
8(0)	11(2)	11(0)	14(1)	11(0)
BayesPE				
20(4)	**14(1)**	**19(4)**	**18(2)**	**20(4)**

Table 34: Summary of results for MPT30bInstruct (20 total experiments).

NLL	AUC	ECE	MCE	Brier
Standard				
8(0)	12(0)	12(0)	13(0)	5(0)
Ensemble				
10(0)	13(1)	15(0)	15(1)	11(1)
Best				
10(0)	15(1)	10(0)	13(1)	12(0)
BayesPE				
20(5)	**17(1)**	**19(1)**	**16(1)**	**19(2)**

Table 35: Summary of results for Mistral7bInstruct (20 total experiments).