Bayesian Adaptation Gym: A Benchmark for the Bayesian Low-Rank Adaptation of Multi-Modal Language Models

Colin Samplawski[1] **Ramneet Kaur**[1] **Manoj Acharya**[1] **Anirban Roy**[1] **Adam D. Cobb**[1]

[1]Neuro-Symbolic Computing and Intelligence Research Group, Computer Science Laboratory, SRI International

Abstract

Large multi-modal language models are increasingly deployed in high-stakes domains, making well-calibrated uncertainty essential. Traditional Bayesian methods approximate posteriors over all model weights, which becomes intractable for modern large models. For this reason, recent work instead considers Bayesian low-rank adaptation to enable tractable posterior approximation. Due to a lack of a standardized benchmark to evaluate these approaches, it remains unclear where these methods provide meaningful benefits. To fill this gap, we introduce Bayesian Adaptation Gym (BAG), a benchmark for the Bayesian adaptation of multi-modal language models. BAG provides reference implementations of classic Bayesian baselines and state-of-the-art adaptation methods, along with a multi-modal dataset and task suite designed to probe calibration, robustness under distribution shift, and decision-making under uncertainty via active learning. Using BAG, we conduct and report extensive experiments across model sizes, datasets, and tasks to highlight the successes and failures of current Bayesian adaptation approaches. To enable further research, BAG is fully open source: `https://github.com/SRI-CSL/BayesAdapt`.

1 INTRODUCTION

The uncertainty quantification (UQ) of large, multi-modal language models has become an increasing priority of the research community due to widespread use of these models in many high-stakes domains, such as healthcare [Poon et al., 2025], scientific research [Chen et al., 2025], or even everyday use by the general public. UQ offers a promising way to ensure well-calibrated and trustworthy outputs from these models which are known to hallucinate and output unreliable information [Kalai et al., 2025].

A broad family of approaches has emerged in the space of UQ over large models. The most popular solutions are "black box" approaches, such as self consistency-based approaches [Wang et al., 2023] or verbalized confidence [Xuan et al., 2025], which consider only the uncertainty in the output space of the model. While often effective, they are incapable of probing the uncertainty in the parameters of the model itself. For this, we require Bayesian methods [Wang and Yeung, 2020]. Here, uncertainty quantification is performed by directly approximating a distribution over the weights of the model $P(\mathbf{W}|\mathcal{D})$ where $\mathcal{D}$ is a training (or fine-tuning) dataset, and $\mathbf{W}$ are the model parameters. We can then estimate a model's output uncertainty by marginalizing over this parameter posterior distribution via a Bayesian model average.

The fundamental challenge is then finding a high quality approximation of the parameter posterior $P(\mathbf{W}|\mathcal{D})$. It is well known that this quickly becomes intractable as the dimensionality of $\mathbf{W}$ grows, making the huge parameter sizes of modern LLMs especially challenging. However, an emerging body of work has focused instead on the task of Bayesian adaptation [Yang et al., 2024, Wang et al., 2024, Shi et al., 2025, Samplawski et al., 2025, Pham et al., 2025]. In this setting, we model the posterior distribution not over the full parameters $\mathbf{W}$, but over low-rank adaptation (LoRA) parameters learned during fine-tuning. Due to the low-rank nature of these parameters, approximating this LoRA posterior remains tractable.

The current benchmarking protocol for the Bayesian adaptation of LLMs used throughout prior work originates from the Laplace LoRA work of Yang et al. [2024]. Under this setup, models are fine-tuned and tested on multiple-choice commonsense reasoning and trivia datasets. Classification accuracy, along with uncertainty metrics such as expected calibration error (ECE) and negative log likelihood (NLL) are used to judge the performance of different techniques.

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:5919–5966.

This protocol has become the standard evaluation scheme used in essentially all subsequent work on this problem [Wang et al., 2024, Shi et al., 2025, Samplawski et al., 2025, Pham et al., 2025].

A major limitation of this protocol is that for most of the datasets considered, there is little "headroom" for fine-tuning. By this we mean that there is a small performance difference before and after fine-tuning (see Table 1). This can happen either because the training set provides limited transferable signal, and/or because the base model already has strong performance due to pretraining exposure (implicit or explicit). This low headroom often makes the Bayesian adaptation process less of a fine-tuning and more of a very expensive and complex in-distribution calibration process. In Section 4.2 we show that a simple temperature scaling baseline [Guo et al., 2017] is often competitive in this evaluation setup. Additionally, prior tasks are limited to text-based datasets and unimodal LLMs, ignoring current multi-modal use cases.

Beyond dataset issues, prior work provides limited discussion of resource usage, despite Bayesian adaptation methods often incurring substantial additional cost. Moreover, evaluations are typically restricted to a narrow band of backbone sizes, most commonly 7-8B parameter LLMs, leaving it unclear how these methods scale in memory and latency with other model sizes. These problems are further intensified by the lack of a modular framework with which to build and evaluate methods.

To fill these gaps, we introduce Bayesian Adaptation Gym (BAG), a benchmark for the Bayesian adaptation of multi-modal language models. We design BAG along the following axes: first, we prioritize multi-modal tasks with meaningful adaptation headroom, where fine-tuning offers clear gains over the pretrained model. Second, we measure uncertainty quality and robustness for both in-distribution data and across a broader set of out-of-distribution shifts than in prior work. As such, we are able to highlight clearer differences in uncertainty performance across methods. Third, we include the decision-centric evaluation of active learning, where uncertainty directly affects which data are acquired and thus impacts label efficiency. Fourth, we treat resource utilization as a first-class metric by standardizing reporting of inference-time resource usage across a range of backbone sizes.

Finally, BAG is released as a modular and extensible framework with reference implementations of classic Bayesian deep learning and recent Bayesian adaptation methods, enabling controlled and reproducible comparisons under a unified training and evaluation pipeline, while also making it straightforward to add novel datasets and adaptation techniques.

The contributions of our work are as follows:

- We introduce Bayesian Adaptation Gym (BAG) a modular and extensible Python framework to benchmark Bayesian adaptation of VLMs (Section 3).
- We include reference implementations of classic and state-of-the-art techniques for the problem of Bayesian adaptation.
- BAG includes a dataset suite made up new multi-modal tasks, as well as new protocols for out-of-distribution evaluation and active learning.
- Using BAG, we perform and present an extensive set of experiments, where we explore performance across different model sizes, datasets, and tasks.
- To enable further research, BAG is fully open source: `https://github.com/SRI-CSL/BayesAdapt`

2 PRIOR WORK AND PRELIMINARIES

Prior work on uncertainty quantification of recent models has attracted a large literature of techniques to apply to this problem. There remains limited work which aims to rigorously compare these approaches. However, what does exist [Vashurin et al., 2025] does not consider Bayesian methods. On this front, there have been a number of prior works which rigorously benchmark Bayesian techniques [Band et al., 2021, Wilson et al., 2021, Vadera et al., 2022]. However, these prior benchmarks originate from the pre-LLM era and are primarily concerned with the posterior approximation of image classification CNNs trained from scratch, leaving the increasingly important setting of Bayesian adaptation of pretrained VLMs without a standardized evaluation protocol.

2.1 LOW-RANK ADAPTATION

First introduced by Hu et al. [2022], low-rank adaptation (LoRA) has become a common technique for the tractable fine-tuning of large models. Let $\mathbf{W}_0 \in \mathbb{R}^{n\times d}$ be the pretrained weights a single linear layer, where d is the embedding dimension and n is the output size. When fine-tuning with LoRA, the pretrained weights $\mathbf{W}_0$ are kept fixed and instead we learn a new pair of low-rank matrices $\mathbf{A} \in \mathbb{R}^{r\times d}$ and $\mathbf{B} \in \mathbb{R}^{n\times r}$, for LoRA rank $r \ll \min(n, d)$. We can then compute the forward pass as the addition of linear operations on a batch of B input features $\mathbf{x} \in \mathbb{R}^{B\times d}$:

$$\mathbf{h} = \mathbf{x}\mathbf{W}_0^T + \mathbf{x}(\mathbf{B}\mathbf{A})^T \tag{1}$$

The low-rank nature of $\mathbf{A}$ and $\mathbf{B}$ lead to considerable memory savings during fine-tuning compared to full weight updates.

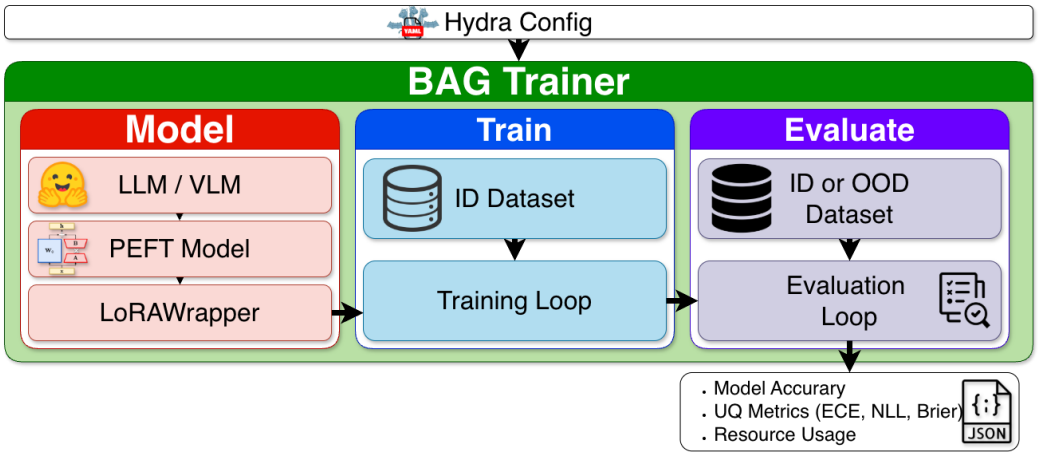


Figure 1: Overview of the BAG `Trainer` object. Every box represents a component in the framework which can be modified or extended.

2.2 BAYESIAN LOW-RANK ADAPTATION

Bayesian low-rank adaptation replaces the point-estimate adapter of a typical LoRA fine-tuning with a distribution over adapter parameters. This treats $\mathbf{A}$ and/or $\mathbf{B}$ as random variables with a prior and posterior inferred using a fine-tuning dataset $\mathcal{D}$. More concretely, for frozen pretrained base weights $\mathbf{W}_0$ and LoRA parameters $\Delta := (\mathbf{A}, \mathbf{B})$, the Bayesian predictive distribution marginalizes over the adapter posterior:

$$P(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) = \int P(\mathbf{y} \mid \mathbf{x}, \mathbf{W}_0, \Delta)\, P(\Delta \mid \mathcal{D})\, \mathrm{d}\Delta, \quad (2)$$

which is approximated in practice using a tractable posterior approximation. Since Δ lives in a low-dimensional subspace determined by the LoRA rank r, this posterior approximation and inference is far more tractable than full Bayesian inference over $\mathbf{W}_0$, while still capturing parameter uncertainty in the full weight space. In practice, we approximate this intractable integral by taking the Bayesian model average over N Monte Carlo samples from the posterior approximation.

3 BAYESIAN ADAPTATION GYM

Bayesian Adaptation Gym (BAG) is a benchmark and programming framework for evaluating Bayesian low-rank adaptation of multi-modal language models. A core design goal is to make comparisons between methods controlled and reproducible by standardizing (i) how Bayesian adapters are inserted into the model, (ii) how evaluation is performed, (iii) how datasets and tasks are formatted, and (iv) how evaluation metrics (including resource costs) are measured and reported. BAG is also designed with an eye toward future research in this emerging area. Its modular and extensible nature makes it straightforward to test new VLM variants from HuggingFace (usually just one line change in the config file), new datasets, and new Bayesian adaptation techniques.

Loosely inspired by PyTorch Lightning [Falcon, 2019], the high level structure of BAG is built around a monolithic `Trainer` object with modular training and evaluation loops (Figure 1). The `Trainer` and its behavior are completely defined by a `hydra` configuration file [Yadan, 2019]. The `Trainer` includes the boilerplate code needed to load any HuggingFace VLM and insert a LoRA adapter around every desired layer. A key abstraction of BAG is then a set of implemented `lorawrappers` which provide a modular way to define the logic of each Bayesian adaptation technique. BAG then defines a universal dataset output format for collation, allowing the easy integration of HuggingFace or custom datasets into the training and evaluation pipelines. The `Trainer` and `lorawrappers` implementations are both highly modular and extensible, allowing the straightforward modification of nearly every component either by modifying `hydra` configuration or extending to new subclasses. Through the use of the `ray` plugin [Moritz et al., 2018] to `hydra`, BAG easily supports performing massive parallel sweeps on multi-GPU nodes.

3.1 IMPLEMENTED METHODS

BAG offers implementations of standard Bayesian methods as well as recent state-of-the-art approaches designed specifically for the problem of Bayesian low-rank adaptation. We briefly describe them here, with more details for each method provided in Appendix Section B.

MLE and MAP: A fundamental baseline is the standard LoRA fine-tuning process itself, where the low-rank matrices are learned by gradient-based optimization without explicitly modeling a weight posterior. We refer to this as the maximum likelihood estimate (MLE). Following prior

work, we also consider a Maximum A Posteriori (MAP) variant by adding weight decay during LoRA fine-tuning.

We then group the Bayesian methods into two categories: post-hoc methods, which require access to a pretrained adapter and approximate the posterior after training, and online methods, which minimize training loss and learn a posterior approximation end-to-end during fine-tuning.

Laplace: The Laplace-LoRA approach of Yang et al. [2024] constructs a local Gaussian approximation of the posterior around a trained adapter using second-order curvature, with additional structure and approximations to keep the Hessian computations tractable.

TFB: Training-Free Bayesianization (TFB) Shi et al. [2025] converts a trained MLE adapter into a Bayesian one by restricting the posterior to a low-rank isotropic Gaussian over the adapter weights. Rather than optimizing variational parameters, TFB selects a single global noise scale σ_q via a simple binary search which seeks to maximize uncertainty while keeping classification performance on an in-distribution "anchor" dataset within a tolerance ϵ.

TempScale: Although not a Bayesian method, we also include temperature scaling [Guo et al., 2017] as a simple calibration baseline. Using gradient optimization on a validation set, we learn a single scalar T which rescales the probabilities of a pretrained MLE adapter.

Deep Ensembles: Deep ensembles [Lakshminarayanan et al., 2017] train N independent LoRA adapters using different random seeds. In principle, this method samples weights from the true posterior, but comes at the significant cost of multiplying training and storage cost by N.

MCDropout: The Monte Carlo (MC) dropout approach of Gal and Ghahramani [2016] reinterprets standard dropout as a form of approximate variational inference, leading to an implicit posterior approximation via Bernoulli masking. At test time, dropout remains enabled and N stochastic forward passes are used to approximate the predictive distribution.

BLoB: The Bayesian LoRA by Backprop (BLoB) approach of Wang et al. [2024] extends Bayes by Backprop [Blundell et al., 2015] to LoRA by asserting a variational diagonal Gaussian posterior over the LoRA $\mathbf{A}$ matrix, learning both a mean and variance parameter for each weight in $\mathbf{A}$. Using the reparameterization trick, we can draw stochastic weight samples during training to optimize the evidence lower bound (ELBO) [Jordan et al., 1999], jointly estimating the adapter weights and their uncertainty during fine-tuning.

ScalaBL: Similar to BLoB, the scalable Bayesian low-rank adaptation (ScalaBL) approach of Samplawski et al. [2025] performs stochastic variational inference in an r-dimensional subspace (where r is the LoRA rank), then repurposes the $\mathbf{A}$ and $\mathbf{B}$ matrices to map low-dimensional samples into the full-weight space. This reduces the number of additional variational parameters compared to BLoB while still remaining broadly performant.

3.2 DATASETS

In BAG, we carry forward all text-only datasets typically used in prior work, namely: Winogrande [Sakaguchi et al., 2021], ARC-Easy/Challenge [Clark et al., 2018], OpenBookQA [Mihaylov et al., 2018], BoolQ [Clark et al., 2019], and MMLU [Hendrycks et al., 2021].

We also incorporate a new text-only dataset derived from the Reasoning Gym benchmark of Stojanovski et al. [2025]. Specifically, we include the **Circuit Logic** task, where the model must determine the output truth value of a randomly generated Boolean circuit displayed in Unicode, given input assignments for each variable. Because each circuit can be represented equivalently as a logical formula, the task naturally supports controlled distribution shifts (e.g., training on one representation and evaluating on the other). This provides a useful stress test for out-of-distribution generalization and for whether uncertainty estimates increase appropriately when the representation changes while the underlying reasoning problem remains the same. Further details are provided in Appendix Section C.5.

In an effort to move beyond a unimodal evaluation setup, we additionally add the following vision-based datasets: **SLAKE** [Liu et al., 2021] is a medical visual question answering dataset built around radiology images, with questions that require both understanding the image (e.g., anatomy/abnormalities) and applying clinical knowledge. Further details are provided in Appendix Section C.6.

MathVerse [Zhang et al., 2024] is a mathematical reasoning benchmark where each problem pairs a natural language question with a visual input (e.g., geometric diagrams, plots, tables, etc.) to test whether a model can reason mathematically about what it sees. Further details are provided in Appendix Section C.7.

MMStar [Chen et al., 2024] is a dataset of hand-selected image-question pairs which test a model's visual perception and understanding. It is designed specifically such that there is minimal leakage in pretrained models and that the image content must be understood to correctly answer the question. Further details are provided in Appendix Section C.8.

A limitation of these datasets is their relatively small size. To address this, we introduce **SymbolicRegressionQA** (SRQA), a novel multi-modal dataset based on the task of symbolic regression. In symbolic regression, the goal is to recover a symbolic equation $y = f(x)$ for an unknown function f, given sampled numerical input–output pairs (x, y).

We leverage the data generation code of Meidani et al. [2024], which can be used to draw an effectively unlimited

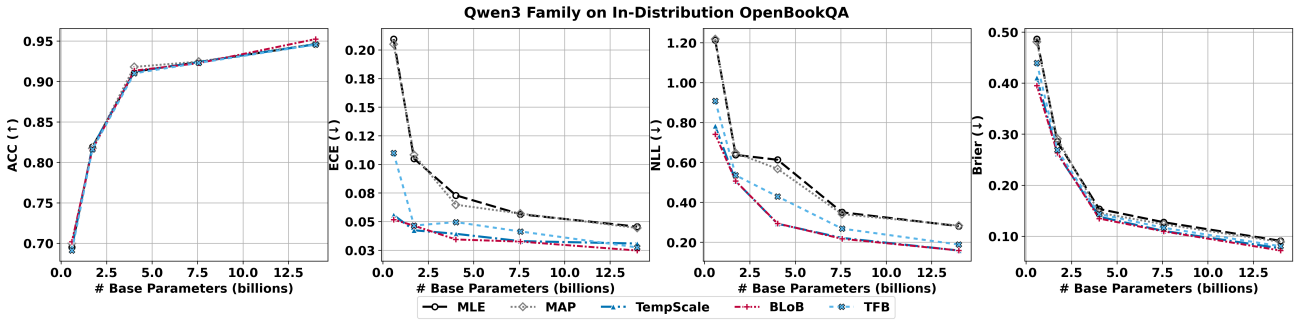


Figure 2: Test of model calibration using OpenBookQA dataset from prior work across a range of model sizes in the Qwen3 family. We see that simple temperature scaling is competitive baseline on this evaluation setup (which is often used in prior work).

number of samples of symbolic expressions together with corresponding numerical input–output data. We configure the generator to produce only one-dimensional functions and render each sampled function as a plot using its associated (x, y) samples. For every function, we additionally sample three distractor expressions, yielding a four-way multiple-choice question in which the model must identify the correct expression given the plot image. This construction has several appealing properties: it supports near-unlimited data samples, its difficulty is easily tuned (e.g., by adjusting the data generator parameters or increasing the number of answer choices), and it is unlikely to overlap with typical pretraining corpora. Further details are provided in Appendix Section C.9.

3.3 IMPLEMENTATION DETAILS AND HYPERPARAMETERS

BAG began as a fork of the `bayesian-peft` library introduced by Wang et al. [2024], which provided the implementations for BLoB and TFB. The original code for Laplace and ScalaBL was integrated directly into the framework. For all experiments, we use the recent Qwen3 [Yang et al., 2025] and Qwen3-VL [Bai et al., 2025] family of models. We also provide some experiments using the Gemma 3 VLM in Appendix Section D.4. For the text-only models we consider sizes from 0.6 billion parameters to 14 billion base parameters and for the VLMs we use the instruction-tuned versions with 2, 4, and 8 billion base parameters. Following prior work, we apply the LoRA adapters to the query and value projection linear layers in each attention layer of the models as well as in the final classification output layer using a LoRA rank of $r = 8$. The effect of this hyperparameter is explored in Appendix Section D.3. Although beyond the scope of this paper, using BAG, is it straightforward to change perform efficient last layer Bayesianization [Harrison et al., 2024] by modifying the `hydra` configuration. Unless stated otherwise, we train all methods for 5000 steps using the AdamW optimizer using a batch size of 4. For all methods which require multiple samples, we use a Bayesian model average of size $N = 10$. All approach-specific hyperparameters were copied over from the original implementation. For all experiments, we report the mean over 4 random seeds. Most experiments were performed on a single node with 8 RTX 6000 GPUs each with 48 GB of memory, with all experiments only using a single GPU at a time. For some experiments with large memory requirements we used a single A100 GPU with 80GB of memory.

4 EXPERIMENTS

4.1 DATASET HEADROOM

Many datasets used in prior work exhibit low headroom, meaning fine-tuning yields only marginal gains over the pretrained model. Table 1 quantifies headroom by comparing pretrained versus finetuned version across all methods. We observe essentially no headroom on ARC-Challenge (with similar phenomena observed on ARC-Easy, OpenBookQA, and BoolQ), while Winogrande and SLAKE retain moderate headroom. In contrast, datasets newly introduced in BAG, such as Circuit Logic and SRQA, show large improvements after adaptation, providing a cleaner testbed for adaptation methods. Finally, MathVerse (and MMStar) remain challenging in a different way: performance is mediocre both before and after fine-tuning, consistent with a setting where the pretrained model lacks prior knowledge and the available training signal is weak.

We see in the case of low headroom datasets, the non-Bayesian methods (MLE and MAP) see no improvement in uncertainty-based metrics, while Bayesian methods (and temperature scaling) are able to calibrate these already accurate logits. For the high headroom datasets of Circuit Logic and SRQA, we see the non-Bayesian methods are often well-calibrated. However, this is largely an effect of large training set (10K generated samples each). The effect of

Table 1: In-distribution results for datasets with different amount of headroom using Qwen3-8B and Qwen3B-VL-Instruct models. Underlined datasets were not considered by prior work.

Metric	Method	ARC-Challenge	Winogrande (s)	Circuit Logic	SLAKE	SRQA	MathVerse
ACC (↑)	Pretrained	0.910	0.511	0.387	0.789	0.673	0.502
	MLE	$0.912_{\pm 0.003}$	$0.760_{\pm 0.005}$	$0.842_{\pm 0.003}$	$0.892_{\pm 0.008}$	$0.966_{\pm 0.003}$	$0.513_{\pm 0.031}$
	MAP	$0.910_{\pm 0.003}$	$0.764_{\pm 0.006}$	$0.840_{\pm 0.006}$	$0.901_{\pm 0.006}$	$0.964_{\pm 0.005}$	$0.527_{\pm 0.031}$
	TempScale	$0.912_{\pm 0.003}$	$0.760_{\pm 0.005}$	$0.842_{\pm 0.003}$	$0.892_{\pm 0.008}$	$0.965_{\pm 0.002}$	$0.511_{\pm 0.030}$
	MCDropout	$0.913_{\pm 0.003}$	$0.759_{\pm 0.003}$	$0.842_{\pm 0.004}$	$0.887_{\pm 0.011}$	$0.961_{\pm 0.000}$	$0.532_{\pm 0.016}$
	Ensemble	$0.917_{\pm 0.002}$	$0.760_{\pm 0.006}$	$0.824_{\pm 0.002}$	$0.893_{\pm 0.005}$	$0.952_{\pm 0.003}$	$0.571_{\pm 0.018}$
	Laplace	$0.879_{\pm 0.009}$	$0.760_{\pm 0.004}$	$0.840_{\pm 0.000}$	$0.891_{\pm 0.009}$	$0.965_{\pm 0.002}$	$0.512_{\pm 0.025}$
	BLoB	$0.917_{\pm 0.002}$	$0.771_{\pm 0.010}$	$0.843_{\pm 0.006}$	$0.886_{\pm 0.008}$	$0.962_{\pm 0.004}$	$0.569_{\pm 0.025}$
	ScalaBL	$0.922_{\pm 0.003}$	$0.761_{\pm 0.004}$	$0.825_{\pm 0.011}$	$0.877_{\pm 0.005}$	$0.948_{\pm 0.002}$	$0.584_{\pm 0.018}$
	TFB	$0.911_{\pm 0.002}$	$0.761_{\pm 0.003}$	$0.830_{\pm 0.005}$	$0.885_{\pm 0.013}$	$0.967_{\pm 0.001}$	$0.503_{\pm 0.035}$
ECE (↓)	Pretrained	0.072	0.431	0.574	0.161	0.251	0.310
	MLE	$0.079_{\pm 0.004}$	$0.227_{\pm 0.005}$	$0.027_{\pm 0.007}$	$0.100_{\pm 0.007}$	$0.016_{\pm 0.001}$	$0.439_{\pm 0.027}$
	MAP	$0.082_{\pm 0.004}$	$0.222_{\pm 0.008}$	$0.020_{\pm 0.002}$	$0.093_{\pm 0.011}$	$0.015_{\pm 0.004}$	$0.427_{\pm 0.027}$
	TempScale	$0.028_{\pm 0.004}$	$0.126_{\pm 0.004}$	$0.018_{\pm 0.006}$	$0.032_{\pm 0.007}$	$0.014_{\pm 0.004}$	$0.097_{\pm 0.014}$
	MCDropout	$0.079_{\pm 0.002}$	$0.225_{\pm 0.005}$	$0.023_{\pm 0.004}$	$0.106_{\pm 0.009}$	$0.016_{\pm 0.001}$	$0.407_{\pm 0.016}$
	Ensemble	$0.065_{\pm 0.003}$	$0.198_{\pm 0.010}$	$0.034_{\pm 0.003}$	$0.082_{\pm 0.003}$	$0.014_{\pm 0.002}$	$0.222_{\pm 0.011}$
	Laplace	$0.429_{\pm 0.012}$	$0.189_{\pm 0.005}$	$0.055_{\pm 0.009}$	$0.109_{\pm 0.020}$	$0.015_{\pm 0.003}$	$0.069_{\pm 0.032}$
	BLoB	$0.030_{\pm 0.003}$	$0.108_{\pm 0.009}$	$0.032_{\pm 0.007}$	$0.049_{\pm 0.006}$	$0.011_{\pm 0.002}$	$0.171_{\pm 0.019}$
	ScalaBL	$0.036_{\pm 0.003}$	$0.125_{\pm 0.004}$	$0.058_{\pm 0.014}$	$0.037_{\pm 0.004}$	$0.026_{\pm 0.003}$	$0.153_{\pm 0.018}$
	TFB	$0.065_{\pm 0.003}$	$0.199_{\pm 0.002}$	$0.076_{\pm 0.001}$	$0.081_{\pm 0.007}$	$0.018_{\pm 0.006}$	$0.364_{\pm 0.029}$
NLL (↓)	Pretrained	0.947	1.656	2.007	0.778	1.918	1.749
	MLE	$0.833_{\pm 0.068}$	$2.614_{\pm 0.408}$	$0.294_{\pm 0.004}$	$0.980_{\pm 0.083}$	$0.092_{\pm 0.004}$	$4.757_{\pm 0.529}$
	MAP	$0.919_{\pm 0.148}$	$2.351_{\pm 0.102}$	$0.295_{\pm 0.004}$	$0.954_{\pm 0.162}$	$0.096_{\pm 0.004}$	$4.417_{\pm 0.366}$
	TempScale	$0.277_{\pm 0.005}$	$0.583_{\pm 0.009}$	$0.292_{\pm 0.005}$	$0.270_{\pm 0.007}$	$0.090_{\pm 0.004}$	$1.136_{\pm 0.061}$
	MCDropout	$0.839_{\pm 0.056}$	$2.373_{\pm 0.109}$	$0.295_{\pm 0.004}$	$1.061_{\pm 0.081}$	$0.103_{\pm 0.000}$	$4.265_{\pm 0.039}$
	Ensemble	$0.640_{\pm 0.041}$	$1.803_{\pm 0.155}$	$0.314_{\pm 0.004}$	$0.716_{\pm 0.052}$	$0.132_{\pm 0.005}$	$1.849_{\pm 0.049}$
	Laplace	$0.853_{\pm 0.035}$	$0.605_{\pm 0.005}$	$0.318_{\pm 0.016}$	$0.320_{\pm 0.014}$	$0.093_{\pm 0.005}$	$1.107_{\pm 0.040}$
	BLoB	$0.258_{\pm 0.009}$	$0.613_{\pm 0.022}$	$0.311_{\pm 0.002}$	$0.276_{\pm 0.009}$	$0.102_{\pm 0.003}$	$1.218_{\pm 0.058}$
	ScalaBL	$0.264_{\pm 0.007}$	$0.648_{\pm 0.013}$	$0.356_{\pm 0.009}$	$0.265_{\pm 0.008}$	$0.141_{\pm 0.002}$	$1.072_{\pm 0.042}$
	TFB	$0.578_{\pm 0.030}$	$1.867_{\pm 0.301}$	$0.373_{\pm 0.003}$	$0.582_{\pm 0.054}$	$0.091_{\pm 0.004}$	$3.180_{\pm 0.410}$

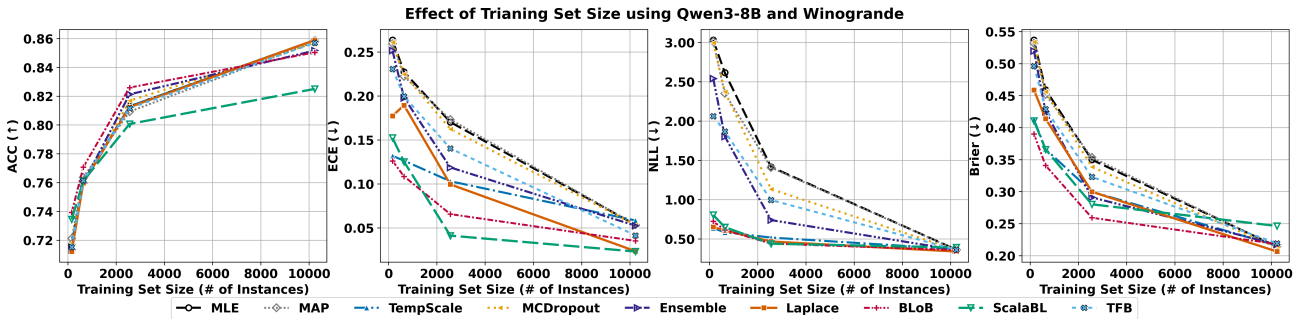


Figure 3: Effect of training set size using the Winograde dataset. We see that as the size of training set increases all methods converge to similar performance.

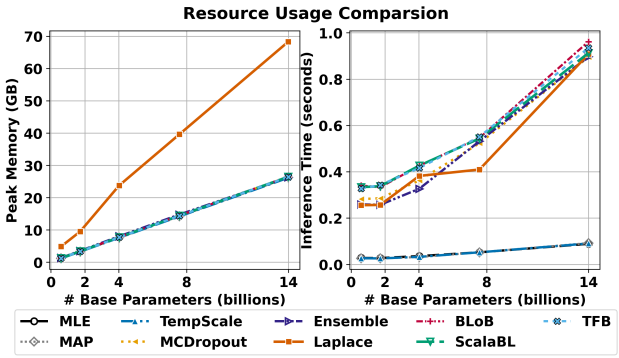


Figure 4: Inference time latency and peak memory usage for Qwen3 family using a batch size of 4 on Winogrande test set.

dataset size is further discussed below. These datasets are still well-suited for limited data experiments such as active learning, which we discuss in Section 4.6. We see for MathVerse, which is a small high headroom dataset, no method is able to significantly improve accuracy, but the Bayesian methods are generally better calibrated, while non-Bayesian methods actually achieving worse calibration.

4.2 IN-DISTRIBUTION CALIBRATION

We first discuss the evaluation protocol typically used in prior work. In Figure 2 we show the in-distribution test performance for the Qwen3 model family. For visual clarity, we focus on the comparison between the recent state-of-the-art methods of BLoB and TFB against a simple MLE baseline, with and without temperature scaling. We first note that the accuracy of all approaches is within a tight bound of each other. This is due to the fact that the accuracy achievable on a dataset like OpenBookQA is highly determined by the model's base knowledge. We see that across all model sizes, the simple temperature-scaled MLE baseline has very competitive performance with BLoB and TFB across all UQ metrics. This highlights one of the fundamental shortcomings of the evaluation protocol of prior work, since this evaluation protocol is unable to distinguish between a complex Bayesian adaptation method and a simple baseline. We further notice that as the base model gets larger, the MLE gets more accurate and better calibrated.

4.3 EFFECT OF TRAINING SET SIZE

Winogrande [Sakaguchi et al., 2021] is a binary commonsense reasoning dataset commonly used in prior work. It comes in a range of training set sizes from extra small (xs) to large (l), with prior work using small (s) and medium (m). These training sets are nested subsets of each other (e.g. all the instances in xs are in s, etc.), with a shared test set used for all sizes. In BAG, we support the full range of sizes and use it as a test of the effect of training set on the adaptation process. We adapt a Qwen3-8B model on each size of Winogrande and report results for all adaptation methods in Figure 3.

We observe the sensible behavior where the Bayesian techniques achieve better performance on the uncertainty metrics compared to an MLE or MAP point-estimate when the training set is small. However, as the training set grows, all methods converge to similar performance across metrics and methods, Bayesian or otherwise. This suggests that Bayesian adaptation methods are most useful in small training set regimes.

4.4 RESOURCE USAGE

In Figure 4 we provide the latency (seconds) and peak memory usage (GB) during model inference for a range of Qwen3 base models (0.6B to 14B parameters) using a batch size of 4 on the Winogrande test set. On the memory axis, we see that nearly all approaches have the same peak memory usage. This isn't surprising, as each sample from the posterior is computed sequentially, giving them all the same memory usage as a single MLE adapter. However, we can see the extreme memory demands of the Laplace approach, which scales poorly with base model size compared to all other methods.

In terms of latency, the approaches can be divided into two categories: single-shot approaches (MLE, MAP, and TempScale) and sampling-based approaches (all other approaches except Laplace). Naturally, these single-shot approaches, which need to compute only a single forward pass result in much lower latency. A major downside of these sampling-based approaches is that a forward pass of the entire network is computed for each sample ($N = 10$ for all approaches) in order to compute the Bayesian model average. We see they are all much slower than MLE, but none is significantly faster than the others. We also see that even though Laplace is not a sampling-based approach, it still has worse latency than the MLE.

We further note that in general the training time of experiments using BAG is generally fast. A typical training run for a single seed takes approximately 5 to 20 minutes of wall-clock time, depending on the particular choice of method, model size, and dataset. In general, the low-rank weight updates of LoRA enable efficient training runs compared to the full parameter training of earlier Bayesian deep learning work.

Finally, we note that since we do not consider applying the adaptation within the vision encoder of the VLMs, the resource use of VLMs is only an additional constant factor compared to the LLM results shown here. We show additional latency results for VLMs in Appendix Figure 17.

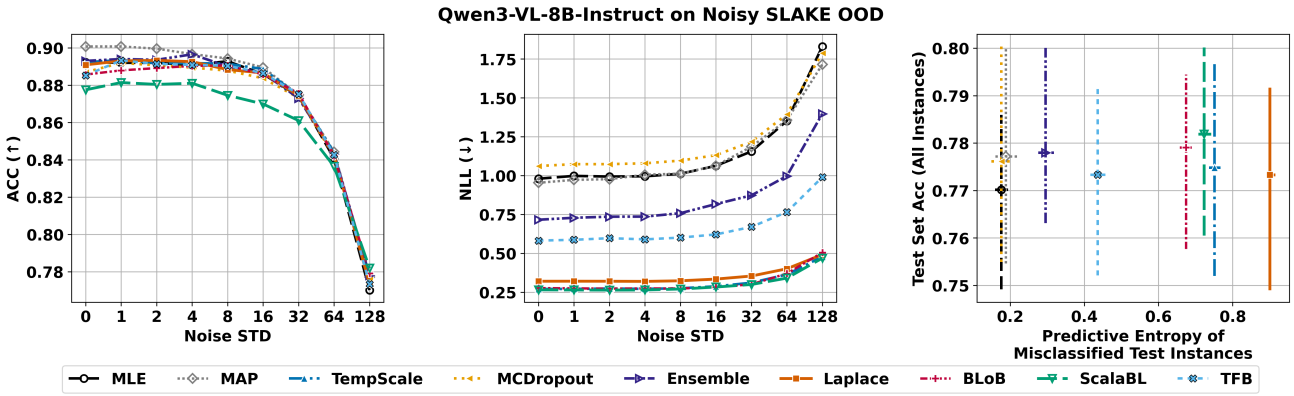


Figure 5: Evaluating approaches on SLAKE test images with Gaussian noise. Left and center plots show the degradation of performance in accuracy and NLL as noise increases, respectively. In the right most plot, we compare predictive entropy of *misclassified* points against full test set accuracy using the highest level of noise ($\sigma = 128$). Due to the noise in the evaluation process 10 evaluation runs we used for each of the 4 training seeds. Plots show the mean of those 40 points. Error bars show the IQR range.

4.5 ROBUSTNESS EXPERIMENTS

A key motivation of Bayesian approaches is to build models which more robustly respond to out-of-distribution (OOD) data encountered at test time. Prior work considered the OOD experiment of training on the OpenBookQA dataset and testing on either ARC-Easy/Challenge or MMLU. For similar reasons discussed above, we find this to be a poor test of the model's OOD performance, as most of these datasets have low headroom and the final performance is closely tied to the pretrained knowledge of the model. However, BAG still includes comparisons using prior datasets, where we produce a set of experiments using MMLU in Appendix Section D.1.1.

In contrast to prior work, in BAG we instead consider more meaningful OOD scenarios. We consider using the Circuit Logic dataset as a test of the model's ability to handle changes in input representation for the same underlying logical reasoning problem. Results for this set of experiments are shown in Appendix Section D.1.2.

SLAKE with Noise BAG extends to multi-modal datasets and enables out-of-distribution evaluations that are not possible in the text-only settings of prior work. In particular, we test robustness to test time visual corruption by adding Gaussian noise to the input image. We perform this experiment on the SLAKE dataset, since robustness to acquisition noise and artifacts especially important in medical settings. Concretely, we perturb grayscale pixel intensities with noise sampled from $\mathcal{N}(0, \sigma)$ (in $[0, 255]$ pixel units) and evaluate performance across increasing noise levels $\sigma \in [1, 2, 4, \ldots, 128]$ (see Appendix Figure 7). In Figure 5 we display results for Qwen3-VL-8B-Instruct on this experiment.

In the left hand and center figures we display how each method in BAG responds to increasing noise in terms of accuracy and NLL. As expected, across all approaches we see a strong decline in performance as the noise increases. In the right most figure we plot the average predictive entropy of misclassified test instances versus the full test set accuracy for the highest noise level ($\sigma = 128$). We would expect a Bayesian method to respond to noisy input by expressing higher uncertainty in its predictions. We see that approaches such as MLE and MAP maintain low entropy even in the face of high input noise, while Bayesian methods such as BLoB, ScalaBL, TFB, and Laplace have much higher uncertainty in this setting. However, we do again see competitive performance by the simple temperature scaling baseline.

4.6 ACTIVE LEARNING

Next we turn our attention to the problem of active learning. In this setting, we benchmark the performance of a model's predictive uncertainty by using it to guide the training data selection process and reduce the burden of labeling instances. We begin by taking the full training set to be the unlabeled data pool. We randomly select $k = 10$ instances as the initial training set. Then, for T iterations we perform the following loop: train an adaptation approach from scratch (i.e. cold start) using the current training set, use this trained model to compute an acquisition function a for each instance in the pool, acquire the label for the top $k = 10$ instances ranked by the acquisition function, and add them to the training set. For the sake of visualization, we also evaluate the model on a held aside test set after each iteration.

For non-sampling approaches such as MLE, MAP, and temperature scaling, we use the model's predictive entropy as

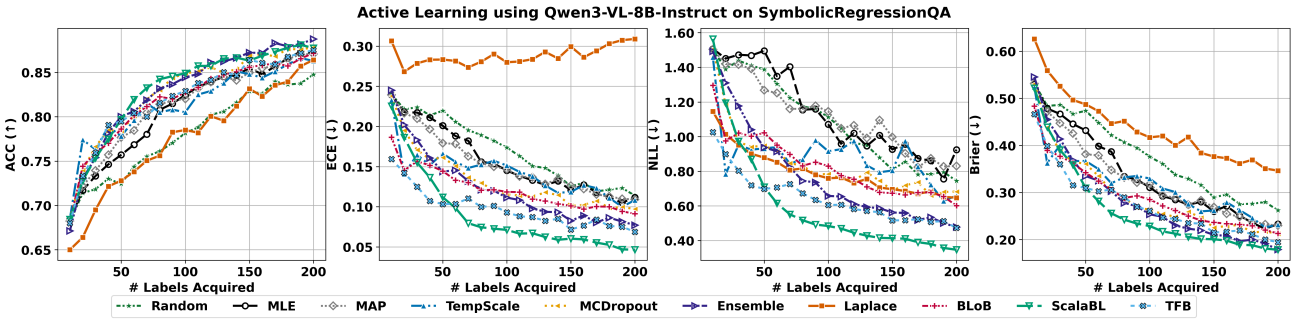


Figure 6: Active learning experiment on the SymbolicRegressionQA dataset using Qwen3-VL-8B-Instruct.

the acquisition function. We also consider a random policy baseline where new instances are chosen at random for MLE training. For the Bayesian adaptation approaches, we use Bayesian Active Learning by Disagreement (BALD) [Houlsby et al., 2011]. Further details on our active learning setup are discussed in Appendix Section B.6. The results for this experiment using the SymbolicRegressionQA dataset are shown in Figure 6.

We first note, the MLE with a random selection policy performs worse than nearly every method and metric considered. This shows that, even without uncertainty-driven selection, using simple predictive entropy leads to better data selection quality.

In contrast to the evaluation setup of prior work our active learning setup shows a stronger and more definitive use case for Bayesian adaptation approaches. We see that in terms of NLL, all the Bayesian approaches (except Laplace) achieve stronger performance. We see especially strong performance using ScalaBL, which we hypothesize is due to its low-rank subspace providing a useful regularization when the training set is small.

An interesting feature of this active learning experiment is that it provides a different rank ordering of the Bayesian approaches compared to the one provided by the evaluation setup of prior work. In particular, we see that BLoB tends to perform comparably with MCDropout, a noticeable departure from the in-distribution calibration experiments of prior work. Beyond ScalaBL, we also see that Deep Ensembles and TFB achieve strong performance on this task. We also notice the worse performance of Laplace, which we suspect is caused by a low-quality Hessian approximation due to the small training set sizes used in active learning.

5 DISCUSSION

Our experimentation via BAG has led to the following observations. If the interest is primarily in having a well-calibrated classifier, then collecting a validation set and performing temperature scaling is likely more straightforward than using Bayesian adaptation. We find that Bayesian methods mostly shine when applied to domains where the base model has limited knowledge and the amount of fine-tuning dataset is low. We find Bayesian methods express greater uncertainty for out-of-distribution test cases than point-wise estimates. Our results on active learning suggest that Bayesian adaptation may be useful for tasks like data curation, active testing, and decision-problems more broadly.

Overall, the approaches of BLoB, TFB, and ScalaBL remain the most promising methods. We find that samples from these approximations often outperform deep ensembles while requiring less training time. We find that the Laplace LoRA approach has inconsistent performance across datasets and is continually held back by its high memory demands, which reduces its scalability to larger models.

6 CONCLUSION

In this work we introduced Bayesian Adaptation Gym (BAG), a modular and extensible framework for the development and benchmarking of low-rank Bayesian adaptation of multi-modal language models. BAG includes implementations of traditional Bayesian techniques and state-of-the-art adaptation approaches. BAG also introduces a larger set of datasets and tasks to tackle an existing gap in the literature, where the prior work used data with little headroom for improvement, and did not provide enough distinguishing power to determine the utility of these approaches. We also include VLMs in the benchmark to evaluate Bayesian methods on multi-modal tasks for the first time. Finally, with the inclusion of the active learning task, which requires decision-making under uncertainty, we show that Bayesian adaptation approaches do provide benefits when data is sparse and better calibration is necessary. We open-source BAG and aim to include more approaches and tasks as this area of research continues to expand.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Neil Band, Tim GJ Rudner, Qixuan Feng, Angelos Filos, Zachary Nado, Michael W Dusenberry, Ghassen Jerfel, Dustin Tran, and Yarin Gal. Benchmarking bayesian deep learning on diabetic retinopathy detection tasks. In *NeurIPS*, 2021.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *ICML*, 2015.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, 2024.

Qiguang Chen, Mingda Yang, Libo Qin, Jinhao Liu, Zheng Yan, Jiannan Guan, Dengyun Peng, Yiyan Ji, Hanjing Li, Mengkang Hu, et al. Ai4research: A survey of artificial intelligence for scientific research. *arXiv preprint arXiv:2507.01903*, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL-HLT*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

William Falcon. Pytorch lightning, March 2019. URL `https://github.com/Lightning-AI/lightning`.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with mmlu? In *NAACL-HLT*, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

James Harrison, John Willes, and Jasper Snoek. Variational bayesian last layers. In *ICLR*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37: 183–233, 1999.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *2021 IEEE 18th international symposium on biomedical imaging (ISBI)*, 2021.

Kazem Meidani, Parshin Shojaee, Chandan K Reddy, and Amir Barati Farimani. Snip: Bridging mathematical symbolic and numeric realms with unified pre-training. In *ICLR*, 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, pages 2381–2391, 2018.

Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: a distributed framework for emerging ai applications. In *Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation*, page 561–577, 2018.

Ngoc-Quan Pham, Tuan Truong, Quyen Tran, Tan Nguyen, Dinh Phung, and Trung Le. Promoting ensemble diversity with interactive bayesian distributional robustness for fine-tuning foundation models. In *ICML*, 2025.

Eric G Poon, Christy Harris Lemak, Juan C Rojas, Janet Guptill, and David Classen. Adoption of artificial intelligence in healthcare: survey of health system priorities, successes, and challenges. *Journal of the American Medical Informatics Association: JAMIA*, 32(7):1093, 2025.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Colin Samplawski, Adam D Cobb, Manoj Acharya, Ramneet Kaur, and Susmit Jha. Scalable bayesian low-rank adaptation of large language models via stochastic variational subspace inference. In *UAI*, 2025.

Haizhou Shi, Yibin Wang, Ligong Han, Huan Zhang, and Hao Wang. Training-free bayesianization for low-rank adapters of large language models. In *NeurIPS*, 2025.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. In *NeurIPS*, 2025.

Meet Vadera, Jinyang Li, Adam Cobb, Brian Jalaian, Tarek Abdelzaher, and Benjamin Marlin. Ursabench: A system for comprehensive benchmarking of bayesian deep neural network models and inference methods. In *MLSys*, 2022.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, et al. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248, 2025.

Hao Wang and Dit-Yan Yeung. A survey on bayesian deep learning. *ACM Computing Surveys*, 53(5), 2020.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.

Yibin Wang, Haizhou Shi, Ligong Han, Dimitris Metaxas, and Hao Wang. Blob: Bayesian low-rank adaptation by backpropagation for large language models. In *NeurIPS*, 2024.

Andrew Gordon Wilson, Pavel Izmailov, Matthew D Hoffman, Yarin Gal, Yingzhen Li, Melanie F Pradier, Sharad Vikram, Andrew Foong, Sanae Lotfi, and Sebastian Farquhar. Evaluating approximate inference in bayesian deep learning. In *NeurIPS*, 2021.

Weihao Xuan, Qingcheng Zeng, Heli Qi, Junjue Wang, and Naoto Yokoya. Seeing is believing, but how much? a comprehensive analysis of verbalized calibration in vision-language models. In *EMNLP*, 2025.

Omry Yadan. Hydra - a framework for elegantly configuring complex applications, 2019. URL `https://github.com/facebookresearch/hydra`.

Adam X Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *ICLR*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *ECCV*, 2024.

Bayesian Adaptation Gym: A Benchmark for the Bayesian Low-Rank Adaptation of Multi-Modal Language Models (Supplementary Material)

Colin Samplawski[1] **Ramneet Kaur**[1] **Manoj Acharya**[1] **Anirban Roy**[1] **Adam D. Cobb**[1]

[1]Neuro-Symbolic Computing and Intelligence Research Group, Computer Science Laboratory, SRI International

A A CALL FOR COMMUNITY ENGAGEMENT

We envision BAG as an evolving benchmark that incorporates new methods and datasets as they become available. We hope that this can be achieved via robust community engagement. To that end, the current code on GitHub contains a guide on working with BAG to add new adaptation methods, datasets, and VLMs.

B FURTHER METHOD DETAILS

In this section we provide further details about each implemented method in BAG.

B.1 MLE AND MAP

As discussed above, MLE and MAP represent a simple LoRA fine-tuning with and without weight decay, respectively. For the MAP approach, we use a weight decay of 0.01.

B.2 DEEP ENSEMBLES

For deep ensembles [Lakshminarayanan et al., 2017], we train a set of $N = 10$ MLE adapters in parallel. At test time, we average the softmax probabilities across the ensemble members.

B.3 MC DROPOUT

In the MCDropout approach of Gal and Ghahramani [2016] we train in a similar fashion to the MLE approach, but set the LoRA dropout rate to be 0.1. Then at test time we perform $N = 10$ forward passes of the model with dropout turned on and compute the average softmax probabilities across these samples.

B.4 LAPLACE LORA

The Laplace LoRA approach of Yang et al. [2024] is a post-hoc Bayesian adaptation method which starts from a standard fine-tuning LoRA checkpoint θ_{MLE}. An isotropic Gaussian prior $p(\theta) = \mathcal{N}(0, \lambda^{-1} I)$ is placed over the LoRA parameters and the posterior is approximated by a Laplace Gaussian:

$$p(\theta \mid \mathcal{D}) \approx \mathcal{N}(\theta_{\mathrm{MLE}}, \Sigma), \qquad \Sigma = (F + \lambda I)^{-1},$$

where F is the Fisher curvature, in a KFAC Kronecker-factorized form.

For a test input $\mathbf{x}^*$, the model is linearized around θ_{MLE}:

$$f_\theta(\mathbf{x}^*) \approx f_{\theta_{\text{MLE}}}(\mathbf{x}^*) + J_{x^*}(\theta - \theta_{\text{MLE}}),$$

which induces an approximate Gaussian distribution over logits

$$f_\theta(\mathbf{x}^*) \sim \mathcal{N}\big(f_{\theta_{\text{MLE}}}(\mathbf{x}^*),\ \Lambda\big), \qquad \Lambda = J_{\mathbf{x}^*}\Sigma J_{\mathbf{x}^*}^T.$$

Predictive uncertainty is obtained via Bayesian model averaging by sampling $\tilde{f} = f_{\theta_{\text{MLE}}} + L\xi$ where $LL^T = \Lambda$ and $\xi \sim \mathcal{N}(0, I)$.

B.5 STOCHASTIC VARIATIONAL APPROACHES

B.5.1 BLoB

The state-of-the-art approach of BLoB [Wang et al., 2024] is based on stochastic variational inference over the LoRA parameters $\mathbf{A}$. More specifically, they follow the Bayes by Backprop approach introduced by Blundell et al. [2015]. That is, they learn a variational approximation $q_\theta(\mathbf{A})$ which is taken to be a Gaussian distribution with mean $\mathbf{A}_\mu$ and variance $\mathbf{A}_\sigma$. Using the reparameterization trick [Kingma and Welling, 2013], samples can be projected from this low rank distribution into the full weight space:

$$\mathbf{W}_t = \mathbf{W}_0 + \mathbf{B}(\mathbf{A}_\mu + \mathbf{A}_\sigma \cdot \boldsymbol{\epsilon}_t) \tag{3}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, 1)$.

This process remains fully differentiable, allowing all the parameters ($\mathbf{B}$, $\mathbf{A}_\mu$, and $\mathbf{A}_\sigma$) to be learned end-to-end using stochastic gradient methods. The training loss for these approaches is then the standard evidence lower bound (ELBO) [Jordan et al., 1999].

$$\mathcal{L}_t = \log P(\mathcal{D}_t|\mathbf{W}_t) - \beta D_{KL}(q_\theta(\mathbf{A})||P(\mathbf{A})) \tag{4}$$

where $P(\mathbf{A})$ is a standard Gaussian prior. During training we use a single sample from $q_\theta(\mathbf{A})$ and at test time we use the average predictive distribution across $N = 10$ samples.

B.5.2 ScalaBL

The ScalaBL approach of Samplawski et al. [2025] follows the stochastic variational inference approach of BLoB, but instead performs inference in a r-dimensional subspace (where r is the LoRA rank). That is, we learn a variational approximation over an r-dimensional vector $\mathbf{s}$ as a diagonal Gaussian distribution:

$$q_{\boldsymbol{\theta}}(\mathbf{s}) = \mathcal{N}(\mathbf{s}|\mathbf{s}_\mu, \text{diag}(\mathbf{s}_\sigma)) \tag{5}$$

with mean and variance parameters $\boldsymbol{\theta} = [\mathbf{s}_\mu, \mathbf{s}_\sigma]$. Like BLoB the reparameterization trick is used to generate weight full weight samples:

$$\mathbf{W}_t = \mathbf{W}_0 + \mathbf{B}\,\text{diag}(\mathbf{s}_\mu + \mathbf{s}_\sigma \cdot \boldsymbol{\epsilon}_t)\mathbf{A} \tag{6}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, 1)$. This is again optimized using the ELBO [Jordan et al., 1999]:

$$\mathcal{L}_t = \log P(\mathcal{D}_t|\mathbf{W}_t) - \beta D_{KL}(q_\theta(\mathbf{s})||P(\mathbf{s})) \tag{7}$$

ScalaBL enjoys greater parameter efficiency compared to BLoB as it only needs to learn r variance parameters compared to needing to learn the full $\mathbf{A}_\sigma$ matrix which is typically $r \times d$, where d is the embedding dimension of the model. Like with BLoB, we use a single posterior sample during training and $N = 10$ samples during test.

Algorithm 1 Active Learning Loop

Require: Unlabeled pool $\mathcal{U}$
Require: batch size k, downsample size M, number of iterations T
Require: Model initialization distribution $\pi(\theta)$
Require: Acquisition function $a(\mathbf{x}; \theta)$, oracle ORACLELABEL$(\cdot)$

$\mathcal{L} \leftarrow$ RANDOMSAMPLE$\big(\mathcal{U}, k\big)$
$\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{L}$
for $t = 1$ **to** T **do**
  $\theta_0 \sim \pi(\theta)$ ▷ Cold start parameters
  $\theta^* \leftarrow$ TRAIN$(\mathcal{L}, \theta_0)$
  $\mathcal{S} \leftarrow$ RANDOMSAMPLE$\big(\mathcal{U}, M\big)$ ▷ Downsample the pool for acquisition computation
  $\mathcal{Q} \leftarrow$ TOPK$\big(a(\mathbf{x}, \theta^*)$ FOR $\mathbf{x} \in S\big)$ ▷ Add k instances with highest acquisition function
  $\mathbf{y}_\mathcal{Q} \leftarrow$ ORACLELABEL$(\mathcal{Q})$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{(\mathbf{x}, y) : \mathbf{x} \in \mathcal{Q}, y \in \mathbf{y}_\mathcal{Q}\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{Q}$
end for
return $\mathcal{L}$

B.5.3 Training-Free Bayesianization

The Training-Free Bayesianization (TFB) approach of Shi et al. [2025] begins with a trained LoRA MLE adapter $\mathbf{A}$ and $\mathbf{B}$. Then a singular value decomposition (SVD) is performed on the learned $\mathbf{B}$ matrix:

$$\mathbf{U}\text{diag}(\mathbf{s})\mathbf{V}^T = \mathbf{B} \tag{8}$$

where $\mathbf{s} \in \mathbb{R}^r$ is the vector of singular values and $\mathbf{U} \in \mathbb{R}^{n \times r}$, $\mathbf{V} \in \mathbb{R}^{r \times r}$ are the left and right singular vectors, respectively. Using this they construct an equivalent LoRA adapter with $\mathbf{A}' = \mathbf{V}^T\mathbf{A}$ and $\mathbf{B}' = \mathbf{U}\text{diag}(\mathbf{s})$. They then construct a Gaussian variational approximation with mean $\mathbf{V}^T\mathbf{A}$ and diagonal covariance matrix given by $\text{diag}(\sigma_q/\mathbf{s})$ repeated d times.

Here σ_q is a global variance parameter that is shared across all adapted layers. Rather than using stochastic training, a simple binary search is used. At every iteration the NLL on an "anchor" dataset is computed. The goal of the binary search is to find a maximal σ_q that keeps the ratio between the original NLL (i.e. when $\sigma_q = 0$) and current NLL within a tolerance threshold ϵ.

We use the training set used for fine-tuning as the anchor dataset. Following Shi et al. [2025] we set the starting range of the binary search to $\sigma_q \in [0.001, 0.2]$, and use a ratio tolerance of $\epsilon = 0.003$.

B.6 ACTIVE LEARNING

In BAG we apply recent Bayesian adaptation approaches to the problem of active learning for the first time. The active learning loop that we used is shown in Algorithm 1. For each training run within the loop, we train for 1000 steps and start from randomly initialized adaptation parameters each time (i.e. cold start). We find that the main bottleneck in this loop is computing the acquisition function a on each element in the unlabeled pool, which in practice is the training set of one of the datasets in BAG. For this reason we randomly downsample $M = 1000$ points at every iteration and select the top $k = 10$ points only from that subset.

For point-wise approaches, such as MLE or MAP, we use the predictive entropy of the softmax probabilities output by the model as the acquisition function. For the Bayesian approaches which compute a softmax probabilties for N samples from the posterior, we use Bayesian Active Learning by Disagreement (BALD) [Houlsby et al., 2011]:

$$a_{\text{BALD}}(\mathbf{x}, \theta) \;=\; H\big[p(y \mid \mathbf{x}, \mathcal{L})\big] \;-\; \mathbb{E}_{p(\theta \mid \mathcal{L})}\big[\, H\big[p(y \mid \mathbf{x}, \theta)\big]\,\big]\,. \tag{9}$$

BALD scores a candidate point $\mathbf{x}$ by the mutual information between its unknown label y and the current model parameters θ given the current labeled data $\mathcal{L}$. It prefers points whose predictive distribution is globally uncertain (first term) and for which different posterior samples of θ disagree strongly (second term).

B.7 METRICS

In BAG, we report four common metrics for evaluating predictive performance and uncertainty quantification: accuracy, expected calibration error (ECE), negative log-likelihood (NLL), and Brier score [Brier, 1950]. For a test set $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ and a probabilistic classifier $P_{\boldsymbol{\theta}}(y \mid \mathbf{x})$, these metrics are defined as follows:

Accuracy: Accuracy measures the fraction of correct predictions:

$$\text{ACC} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\left[\arg\max_c P_{\boldsymbol{\theta}}(y = c \mid \mathbf{x}_n) = y_n\right]. \tag{10}$$

Expected calibration error: ECE measures the mismatch between confidence and empirical accuracy by binning predictions according to their confidence $\hat{p}_n := \max_c P_{\boldsymbol{\theta}}(y = c \mid \mathbf{x}_n)$. Let B_k be the set of indices whose confidences fall into bin $k \in \{1, \ldots, K\}$:

$$\text{ECE} = \sum_{k=1}^{K} \frac{|B_k|}{N} \left|\text{ACC}(B_k) - \text{conf}(B_k)\right|, \tag{11}$$

where $\text{ACC}(B_k)$ is the accuracy of the model on the instances in the kth bin and

$$\text{conf}(B_k) = \frac{1}{|B_k|} \sum_{n \in B_k} \hat{p}_n \tag{12}$$

Following prior work [Yang et al., 2024, Wang et al., 2024, Samplawski et al., 2025], we use $K = 15$ bins in all experiments.

Negative log-likelihood: NLL is the average negative log-probability assigned to the true label:

$$\text{NLL} = -\frac{1}{N} \sum_{n=1}^{N} \log P_{\boldsymbol{\theta}}(y_n \mid \mathbf{x}_n). \tag{13}$$

Brier score: The Brier score measures the mean squared error between the predicted probabilities and the one-hot label vector. Let $\mathbf{p}_{\boldsymbol{\theta}}(\mathbf{x}_n) \in [0, 1]^C$ denote the predicted class-probability vector over C classes and let $\mathbf{e}_{y_n}$ be the one-hot vector for label y_n:

$$\text{Brier score} = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{p}_{\boldsymbol{\theta}}(\mathbf{x}_n) - \mathbf{e}_{y_n}\|_2^2. \tag{14}$$

C ADDITIONAL DATASET DETAILS

C.1 WINOGRANDE

Winogrande [Sakaguchi et al., 2021] is a dataset for evaluating commonsense reasoning and co-reference resolution via fill-in-the-blank sentence-completion problems. The training set of Winogrande is available in 6 sizes: extra-small (xs), small (s), medium (m), large (l), and extra-large (xl). These training sets are nested subsets of each other (e.g. all the instances in xs are in s). There is then one test set that is used for all sizes. The dataset has no prior defined validation split, which we require for the temperature scaling method. We build a shared validation split using the instances are that are unique to the xl training set.

C.1.1 Prompt Format by Example

```
For the sentence given below, select the answer that best fills in the
blank (_) from the given choices.

Ian volunteered to eat Dennis's menudo after already having a bowl because _
despised eating intestine.

Choices:
A) Ian
B) Dennis
```

C.1.2 Dataset Statistics

Size	# Instances			# Classes
	Train	Validation	Test	
xs	160	30164	1267	2
s	640			
m	2558			
l	10234			

Table 2: Winogrande statistics

C.2 ARC

The ARC (AI2 Reasoning Challenge) dataset [Clark et al., 2018] is a multiple-choice science question dataset split into two difficulties: Easy and Challenge.

C.2.1 Prompt Format by Example

```
Answer the multiple choice question below.
Output the letter of your choice only.

Which land form is the result of the constructive force of a glacier?
Choices:
A) valleys carved by a moving glacier
B) piles of rocks deposited by a melting glacier
C) grooves created in a granite surface by a glacier
D) bedrock hills roughened by the passing of a glacier
```

C.2.2 Dataset Statistics

Difficulty	# Instances			# Classes
	Train	Validation	Test	
Easy	2251	570	2376	4
Challenge	1119	299	1172	

Table 3: ARC statistics

C.3 OPENBOOKQA

The OpenBookQA (OBQA) dataset [Mihaylov et al., 2018] is a multiple-choice elementary school science question dataset.

C.3.1 Prompt Format by Example

```
Answer the multiple choice question below.
Output the letter of your choice only.

The sun is responsible for
Choices:
A) puppies learning new tricks
B) children growing up and getting old
C) flowers wilting in a vase
D) plants sprouting, blooming and wilting
```

C.3.2 Dataset Statistics

# Instances			# Classes
Train	**Validation**	**Test**	
4957	500	500	4

Table 4: OpenBookQA statistics

C.4 BOOLQ

BoolQ [Clark et al., 2019] is a yes/no question answering dataset built from Google search queries paired with Wikipedia passages. Each example asks whether the passage entails the answer to the question, making it a benchmark for reading comprehension and natural language reasoning. In early experiments we found that BoolQ tended to have low headroom and was expensive to run due to the long text passages. For these reasons we do not report results on this dataset in this paper. However, the dataset is still fully supported in BAG.

C.4.1 Prompt Format by Example

```
Read the passage below and answer the question with the words 'true' or 'false'.

Passage: Windows Movie Maker (formerly known as Windows Live Movie Maker in
Windows 7) is a discontinued video editing software by Microsoft.
It is a part of Windows Essentials software suite and offers the ability to
create and edit videos as well as to publish them on OneDrive, Facebook,
Vimeo, YouTube, and Flickr.

Question: is windows movie maker part of windows essentials?
```

C.4.2 Dataset Statistics

C.5 CIRCUIT LOGIC

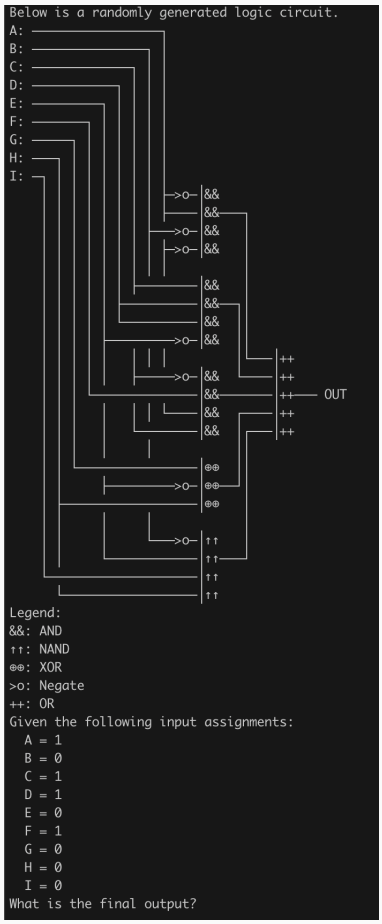
We adapt the Circuit Logic task from the recently purposed Reasoning Gym of Stojanovski et al. [2025]. The model must determine the output truth value of a randomly generated Boolean circuit displayed in unicode, given input assignments for

# Instances			# Classes
Train	Validation	Test	
9427	0	3270	2

Table 5: BoolQ statistics

each variable variables. Because each circuit can be represented equivalently as a logical formula, we have the ability to test a model's ability to generalize to different input representations for the same problem (see example below).

C.5.1 Prompt Format by Example




```
Below is a randomly generated logic expression.
(A>o&&A&&B>o&&A>o)++(C&&D&&D&&E>o)++(C>o&&F&&A&&C)++(G⊕⊕E>o⊕⊕H)++(B>o↑↑E↑↑I↑↑H)
Legend:
&&: AND
↑↑: NAND
⊕⊕: XOR
>o: Negate
++: OR
Given the following input assignments:
  A = 1
  B = 0
  C = 1
  D = 1
  E = 0
  F = 1
  G = 0
  H = 0
  I = 0
What is the final output?
```

C.5.2 Dataset Statistics

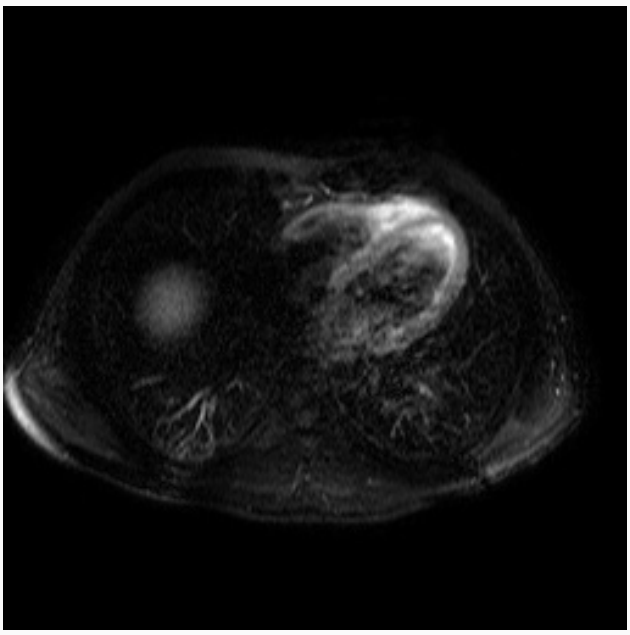
# Instances			# Classes
Train	Validation	Test	
10000	1000	1000	2

Table 6: Circuit Logic statistics

C.6 SLAKE

The SLAKE dataset of Liu et al. [2021] tests a model's ability to understand radiology images in the medical domain. We use the subset of the full dataset which consists of Yes/No questions in English.

C.6.1 Prompt Format by Example



```
Answer the following question as Yes or No only.
Does the picture contain liver?
```

C.6.2 Examples with Noise

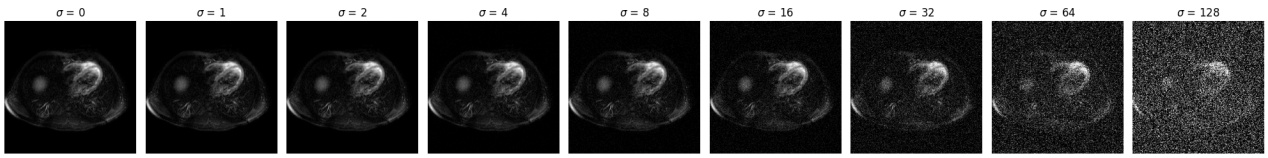


Figure 7: SLAKE image with increasing amounts of Gaussian pixel noise, $\sigma \in [0, 1, 2, 4, \ldots, 128]$

.

C.6.3 Dataset Statistics

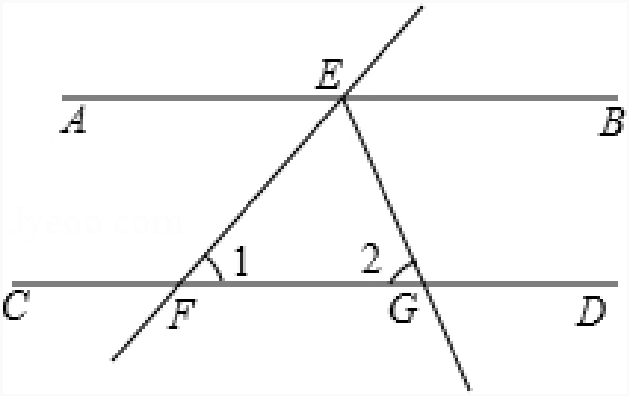
# Instances			# Classes
Train	**Validation**	**Test**	
1681	355	358	2

Table 7: SLAKE statistics

C.7 MATHVERSE

The MathVerse dataset of Zhang et al. [2024] tests a model's ability to answer mathematical reasoning problems which can only be answered correctly by understanding the associated input image. We randomly generate a train/validation/test split for use for this dataset.

C.7.1 Prompt Format by Example




```
Answer the multiple choice question below.
Output the letter of your choice only.

As shown in the figure, angle 1 = 50.0, then angle 2 is equal to ()

Choices:
A:50°
B:60°
C:65°
D:90°
```

C.7.2 Dataset Statistics

# Instances			# Classes
Train	Validation	Test	
1465	315	315	4

Table 8: MathVerse statistics

C.8 MMSTAR

The MMStar dataset of Chen et al. [2024] tests a model's visual understanding and reasoning abilities using data cases where the model must understand the input image in order to correctly answer the question. We randomly generate a train/validation/test split for use for this dataset.

C.8.1 Prompt Format by Example


```
Answer the multiple choice question below.
Output the letter of your choice only.

What is the position of the blue car in the image?
Options:
A: parked on the sidewalk
B: driving on the road
C: parked on the grass
D: parked on the road
```

C.8.2 Dataset Statistics

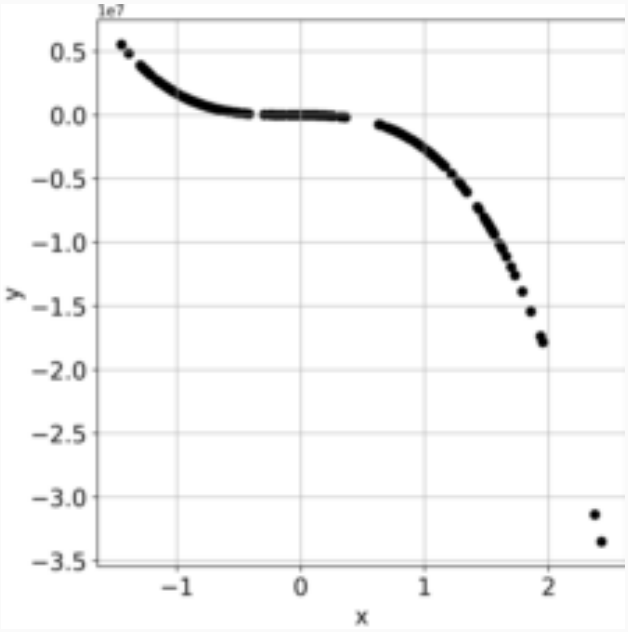
# Instances			# Classes
Train	Validation	Test	
1465	225	225	4

Table 9: MMStar statistics

C.9 SYMBOLICREGRESSIONQA

In BAG we introduce a novel dataset that we call SymbolicRegressionQA (SRQA). Using the data generation code of Meidani et al. [2024], we can generate a near finite supply of symbolic functions f along with numerical data (x, y) such that $x = f(x)$. We configure the data generator to generate only 1 dimensional functions so that we can render them as plots (see example below). We then sample 3 additional expressions to create a 4-way multiple choice question. The model is then tasked with picking which of the four options generated the plot.

C.9.1 Prompt Format by Example




```
For the provided image of a plot, which of following formulas best describes
the relationship between the variables?

Output the letter of your choice only.

Choices:
A) y = 5.75 - 0.85*x
B) y = -42.2124*exp(8.29*x) + 1.35*Abs(8.48*x - 0.693) - 0.697
C) y = -2137999.7563*(x + 0.0751)**3 + 62.6*sin(0.54*x + 0.072) + 0.524
D) y = 74.5*x - 1.68 + 39.2/(0.4401*(0.0785*x + 1)**2*cos(5.4*x - 0.163) - 5.86
```

C.9.2 Dataset Statistics

# Instances			# Classes
Train	Validation	Test	
10000	1000	1000	4

Table 10: SymbolicRegressionQA statistics

D ADDITIONAL RESULTS

In this section we provide a number of tables and figures which display additional experiment results.

D.1 IN-DISTRIBUTION RESULTS

In the Tables 11 through 15 we display in-distribution results using the datasets of prior work: ARC-Easy, ARC-Challenge, and OpenBookQA (obqa) across all model sizes. This is followed by results on the Winogrande dataset for all sizes in Tables 16 through 20. Then we show in-distribution performance for our image-based datasets: SLAKE, MMStar, MathVerse, and SymbolicRegressionQA (SRQA) in Tables 21 through 23.

D.1.1 OOD Results: OBQA -> MMLU

We next consider out-of-distribution (ODD) experiments. We start with OOD experiments similar to prior work where we first train an adapter on the OpenBookQA dataset and then test on various topics from the MMLU dataset [Hendrycks et al., 2021]. We note that in contrast to prior work we use the MMLU-Redux2.0 dataset which fixes a number of known issues in the original MMLU dataset [Gema et al., 2025]. The results of these experiments are displayed in Tables 24 through 28.

D.1.2 OOD: Circuit Logic Representations

Next we present results on the out-of-distribution experiments using the Circuit Logic dataset. We train on a model using the circuit representation and then evaluate using a test of examples in the expression representation, and vice versa. The results for these experiments are displayed in Tables 29 through 32. We see that across all approaches the model is able to generalize well from circuit representation to expression representation, but not the other way around.

D.2 EFFECT OF HYPERPARAMETERS

We next explore the effect of various hyperparameters. BAG's integration with `hydra` and the `ray` distributed computation runtime makes it simple to sweep over different hyperparameters just by modifying a bash script. Furthermore, its integration with the `ray` ecosystem enables straightforward application of more advanced hyperparameter tuning algorithms, such as Bayesian optimization, which is an exciting direction for future work to select critical hyperparameters such as the LoRA rank r or model size.

D.3 EFFECT OF LORA RANK r

In Figures 8 through 14 the effect of the LoRA rank r on the performance for a subset of the in-distribution, OOD, and active learning experiments. This is a key hyperparameter as it controls the general expressiveness of the fine-tuning process, as well as the dimensionality of the weight posterior for the Bayesian approaches. In general we notice the same high level trends regardless of rank. We see that the recent SoTA VI based approaches (BLoB and ScalaBL) strongly outperform simple baselines such as MLE or MCDropout. We find that the LoRA rank also has only a minor impact on the performance across in-distribution, OOD, and active learning results. Naturally, as the rank increases the capability of the fine-tuning process also increases. We suspect that for very large ranks this could lead to an overfitting effect. State of the art VI approaches can respond to this by increasing the weight of the KL term in the ELBO.

D.4 EFFECT OF MODEL FAMILY

In this section we present results where compare the Qwen3 model used in all prior experiments with Google's Gemma3 VLM. In Tables 33 and 34 we display in-distribution results for the Winogrande Small and SLAKE datasets, respectively. In Figure 15, we present results OOD results using noisy SLAKE and Gemma3. In general we notice a similar outcome to the rank ablation above, with VI approaches out performing baselines. We see across most datasets that Gemma3-4B performs slightly worse than its Qwen3 analogue, which we attribute to Gemma being a weaker model overall compared to Qwen, even before fine-tuning.

D.5 EFFECT OF MONTE CARLO SAMPLES

In Figures 16 and 17 we present results studying the effect of the number of Monte Carlo samples used to construct the Bayesian ensemble for the variational methods, for an in-domain setting (Winogrande Small) and OOD (noisy SLAKE with highest level of noise). We consider ensemble sizes in $[1, 2, 5, 10, 20]$. For the state of the art variational methods we see improvement across all metrics as the number of samples increase, with diminishing returns after 10 samples. We see that TFB in general performs worse than the truly online variational approaches, which we suspect is due to the more limited posterior approximation that it performs, with a shared variance parameter across all adapted layers.

These figures also present results showing how latency increases with the size of the ensemble. We see a strictly linear relationship, which isn't surprising as the ensemble members are computed sequentially. We note that there is a fundamental trade-off between latency and memory when computing these Bayesian ensembles. In the results presented in this paper we

take the point in the trade-off space that minimizes memory and maximizes latency. However, it is straightforward to batch the computation of multiple LoRA adapters. This leads to increased memory demands but saves on latency.

Table 11: In-Distribution Performance Comparison for Qwen3-0.6B

Metric	Method	ARC-Easy	ARC-Challenge	obqa
ACC (↑)	MLE	$0.807_{\pm 0.004}$	$0.601_{\pm 0.004}$	$0.695_{\pm 0.004}$
	MAP	$0.803_{\pm 0.002}$	$0.612_{\pm 0.009}$	$0.700_{\pm 0.008}$
	TempScale	$0.807_{\pm 0.004}$	$0.601_{\pm 0.004}$	$0.695_{\pm 0.004}$
	MCDropout	$0.806_{\pm 0.005}$	$0.608_{\pm 0.009}$	$0.700_{\pm 0.009}$
	Ensemble	$0.817_{\pm 0.005}$	$0.623_{\pm 0.005}$	$0.712_{\pm 0.008}$
	Laplace	$0.809_{\pm 0.003}$	$0.583_{\pm 0.011}$	$0.696_{\pm 0.003}$
	BLoB	$0.826_{\pm 0.003}$	$0.633_{\pm 0.008}$	$0.702_{\pm 0.010}$
	ScalaBL	$0.818_{\pm 0.003}$	$0.628_{\pm 0.007}$	$0.695_{\pm 0.008}$
	TFB	$0.807_{\pm 0.004}$	$0.604_{\pm 0.011}$	$0.691_{\pm 0.012}$
ECE (↓)	MLE	$0.168_{\pm 0.005}$	$0.363_{\pm 0.007}$	$0.209_{\pm 0.007}$
	MAP	$0.172_{\pm 0.006}$	$0.356_{\pm 0.004}$	$0.205_{\pm 0.004}$
	TempScale	$0.044_{\pm 0.005}$	$0.084_{\pm 0.009}$	$0.056_{\pm 0.016}$
	MCDropout	$0.163_{\pm 0.002}$	$0.348_{\pm 0.006}$	$0.197_{\pm 0.007}$
	Ensemble	$0.118_{\pm 0.005}$	$0.245_{\pm 0.008}$	$0.175_{\pm 0.008}$
	Laplace	$0.050_{\pm 0.016}$	$0.057_{\pm 0.004}$	$0.122_{\pm 0.010}$
	BLoB	$0.033_{\pm 0.007}$	$0.045_{\pm 0.009}$	$0.052_{\pm 0.009}$
	ScalaBL	$0.035_{\pm 0.004}$	$0.122_{\pm 0.005}$	$0.046_{\pm 0.011}$
	TFB	$0.088_{\pm 0.006}$	$0.224_{\pm 0.011}$	$0.110_{\pm 0.008}$
NLL (↓)	MLE	$1.424_{\pm 0.163}$	$3.964_{\pm 0.196}$	$1.213_{\pm 0.045}$
	MAP	$1.544_{\pm 0.201}$	$3.924_{\pm 0.098}$	$1.218_{\pm 0.044}$
	TempScale	$0.582_{\pm 0.007}$	$1.060_{\pm 0.023}$	$0.782_{\pm 0.012}$
	MCDropout	$1.401_{\pm 0.100}$	$3.877_{\pm 0.185}$	$1.155_{\pm 0.039}$
	Ensemble	$1.041_{\pm 0.049}$	$2.526_{\pm 0.083}$	$1.063_{\pm 0.010}$
	Laplace	$0.557_{\pm 0.011}$	$1.034_{\pm 0.026}$	$0.925_{\pm 0.031}$
	BLoB	$0.477_{\pm 0.005}$	$0.902_{\pm 0.010}$	$0.741_{\pm 0.009}$
	ScalaBL	$0.501_{\pm 0.006}$	$1.015_{\pm 0.008}$	$0.785_{\pm 0.010}$
	TFB	$0.792_{\pm 0.060}$	$2.031_{\pm 0.084}$	$0.909_{\pm 0.042}$
Brier (↓)	MLE	$0.351_{\pm 0.007}$	$0.749_{\pm 0.016}$	$0.486_{\pm 0.004}$
	MAP	$0.361_{\pm 0.009}$	$0.732_{\pm 0.008}$	$0.481_{\pm 0.005}$
	TempScale	$0.285_{\pm 0.004}$	$0.544_{\pm 0.012}$	$0.411_{\pm 0.005}$
	MCDropout	$0.348_{\pm 0.005}$	$0.724_{\pm 0.013}$	$0.474_{\pm 0.009}$
	Ensemble	$0.298_{\pm 0.003}$	$0.607_{\pm 0.009}$	$0.447_{\pm 0.007}$
	Laplace	$0.281_{\pm 0.002}$	$0.536_{\pm 0.009}$	$0.430_{\pm 0.009}$
	BLoB	$0.250_{\pm 0.003}$	$0.479_{\pm 0.004}$	$0.395_{\pm 0.004}$
	ScalaBL	$0.258_{\pm 0.002}$	$0.514_{\pm 0.005}$	$0.423_{\pm 0.006}$
	TFB	$0.292_{\pm 0.003}$	$0.609_{\pm 0.013}$	$0.439_{\pm 0.013}$

Table 12: In-Distribution Performance Comparison for Qwen3-1.7B

Metric	Method	ARC-Easy	ARC-Challenge	obqa
ACC (↑)	MLE	$0.905_{\pm 0.003}$	$0.780_{\pm 0.005}$	$0.819_{\pm 0.012}$
	MAP	$0.904_{\pm 0.005}$	$0.778_{\pm 0.008}$	$0.818_{\pm 0.010}$
	TempScale	$0.905_{\pm 0.003}$	$0.780_{\pm 0.005}$	$0.819_{\pm 0.012}$
	MCDropout	$0.907_{\pm 0.004}$	$0.778_{\pm 0.007}$	$0.818_{\pm 0.009}$
	Ensemble	$0.909_{\pm 0.003}$	$0.789_{\pm 0.006}$	$0.815_{\pm 0.007}$
	Laplace	$0.900_{\pm 0.003}$	$0.771_{\pm 0.005}$	$0.823_{\pm 0.011}$
	BLoB	$0.917_{\pm 0.003}$	$0.791_{\pm 0.004}$	$0.816_{\pm 0.007}$
	ScalaBL	$0.915_{\pm 0.003}$	$0.788_{\pm 0.003}$	$0.802_{\pm 0.008}$
	TFB	$0.904_{\pm 0.003}$	$0.779_{\pm 0.006}$	$0.816_{\pm 0.010}$
ECE (↓)	MLE	$0.084_{\pm 0.003}$	$0.200_{\pm 0.004}$	$0.105_{\pm 0.009}$
	MAP	$0.085_{\pm 0.003}$	$0.203_{\pm 0.009}$	$0.108_{\pm 0.010}$
	TempScale	$0.027_{\pm 0.004}$	$0.064_{\pm 0.010}$	$0.042_{\pm 0.007}$
	MCDropout	$0.082_{\pm 0.005}$	$0.200_{\pm 0.007}$	$0.104_{\pm 0.010}$
	Ensemble	$0.068_{\pm 0.004}$	$0.147_{\pm 0.008}$	$0.100_{\pm 0.004}$
	Laplace	$0.159_{\pm 0.029}$	$0.183_{\pm 0.041}$	$0.049_{\pm 0.009}$
	BLoB	$0.023_{\pm 0.003}$	$0.072_{\pm 0.004}$	$0.046_{\pm 0.010}$
	ScalaBL	$0.024_{\pm 0.002}$	$0.072_{\pm 0.004}$	$0.044_{\pm 0.007}$
	TFB	$0.059_{\pm 0.004}$	$0.145_{\pm 0.006}$	$0.047_{\pm 0.009}$
NLL (↓)	MLE	$0.829_{\pm 0.023}$	$1.973_{\pm 0.112}$	$0.637_{\pm 0.017}$
	MAP	$0.838_{\pm 0.033}$	$2.018_{\pm 0.046}$	$0.647_{\pm 0.032}$
	TempScale	$0.303_{\pm 0.012}$	$0.658_{\pm 0.014}$	$0.506_{\pm 0.007}$
	MCDropout	$0.832_{\pm 0.033}$	$1.958_{\pm 0.125}$	$0.628_{\pm 0.024}$
	Ensemble	$0.591_{\pm 0.034}$	$1.423_{\pm 0.041}$	$0.616_{\pm 0.007}$
	Laplace	$0.395_{\pm 0.038}$	$0.737_{\pm 0.039}$	$0.516_{\pm 0.014}$
	BLoB	$0.252_{\pm 0.006}$	$0.634_{\pm 0.012}$	$0.506_{\pm 0.008}$
	ScalaBL	$0.259_{\pm 0.004}$	$0.603_{\pm 0.009}$	$0.541_{\pm 0.009}$
	TFB	$0.499_{\pm 0.023}$	$1.232_{\pm 0.082}$	$0.535_{\pm 0.017}$
Brier (↓)	MLE	$0.175_{\pm 0.006}$	$0.413_{\pm 0.009}$	$0.286_{\pm 0.007}$
	MAP	$0.178_{\pm 0.005}$	$0.418_{\pm 0.017}$	$0.291_{\pm 0.014}$
	TempScale	$0.148_{\pm 0.005}$	$0.325_{\pm 0.007}$	$0.261_{\pm 0.004}$
	MCDropout	$0.172_{\pm 0.008}$	$0.413_{\pm 0.011}$	$0.282_{\pm 0.010}$
	Ensemble	$0.153_{\pm 0.004}$	$0.355_{\pm 0.008}$	$0.279_{\pm 0.005}$
	Laplace	$0.190_{\pm 0.015}$	$0.378_{\pm 0.021}$	$0.261_{\pm 0.004}$
	BLoB	$0.125_{\pm 0.003}$	$0.303_{\pm 0.004}$	$0.264_{\pm 0.005}$
	ScalaBL	$0.131_{\pm 0.002}$	$0.308_{\pm 0.004}$	$0.284_{\pm 0.004}$
	TFB	$0.154_{\pm 0.007}$	$0.360_{\pm 0.007}$	$0.269_{\pm 0.005}$

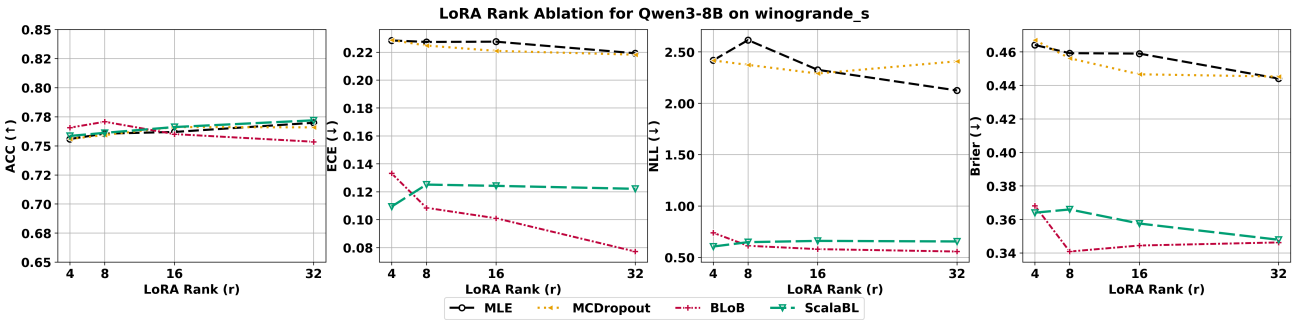


Figure 8: Ablation of LoRA rank on Winogrande-Small using Qwen3-8B.

Table 13: In-Distribution Performance Comparison for Qwen3-4B

Metric	Method	ARC-Easy	ARC-Challenge	obqa
ACC (↑)	MLE	$0.954_{\pm 0.003}$	$0.887_{\pm 0.002}$	$0.912_{\pm 0.014}$
	MAP	$0.954_{\pm 0.002}$	$0.885_{\pm 0.006}$	$0.918_{\pm 0.006}$
	TempScale	$0.954_{\pm 0.003}$	$0.887_{\pm 0.002}$	$0.912_{\pm 0.014}$
	MCDropout	$0.953_{\pm 0.002}$	$0.885_{\pm 0.002}$	$0.916_{\pm 0.007}$
	Ensemble	$0.956_{\pm 0.001}$	$0.891_{\pm 0.004}$	$0.914_{\pm 0.001}$
	Laplace	$0.947_{\pm 0.004}$	$0.855_{\pm 0.010}$	$0.916_{\pm 0.009}$
	BLoB	$0.960_{\pm 0.002}$	$0.888_{\pm 0.004}$	$0.913_{\pm 0.005}$
	ScalaBL	$0.955_{\pm 0.001}$	$0.887_{\pm 0.004}$	$0.907_{\pm 0.005}$
	TFB	$0.956_{\pm 0.001}$	$0.884_{\pm 0.002}$	$0.910_{\pm 0.018}$
ECE (↓)	MLE	$0.039_{\pm 0.003}$	$0.100_{\pm 0.004}$	$0.073_{\pm 0.016}$
	MAP	$0.038_{\pm 0.001}$	$0.104_{\pm 0.004}$	$0.065_{\pm 0.009}$
	TempScale	$0.018_{\pm 0.004}$	$0.026_{\pm 0.008}$	$0.039_{\pm 0.005}$
	MCDropout	$0.040_{\pm 0.002}$	$0.100_{\pm 0.002}$	$0.067_{\pm 0.005}$
	Ensemble	$0.032_{\pm 0.001}$	$0.076_{\pm 0.002}$	$0.062_{\pm 0.003}$
	Laplace	$0.317_{\pm 0.017}$	$0.319_{\pm 0.046}$	$0.087_{\pm 0.015}$
	BLoB	$0.009_{\pm 0.002}$	$0.031_{\pm 0.004}$	$0.034_{\pm 0.005}$
	ScalaBL	$0.021_{\pm 0.001}$	$0.056_{\pm 0.004}$	$0.028_{\pm 0.006}$
	TFB	$0.025_{\pm 0.002}$	$0.071_{\pm 0.002}$	$0.050_{\pm 0.014}$
NLL (↓)	MLE	$0.315_{\pm 0.029}$	$1.002_{\pm 0.084}$	$0.613_{\pm 0.131}$
	MAP	$0.329_{\pm 0.045}$	$1.082_{\pm 0.106}$	$0.568_{\pm 0.065}$
	TempScale	$0.144_{\pm 0.010}$	$0.373_{\pm 0.010}$	$0.292_{\pm 0.012}$
	MCDropout	$0.317_{\pm 0.019}$	$0.975_{\pm 0.021}$	$0.549_{\pm 0.014}$
	Ensemble	$0.263_{\pm 0.018}$	$0.738_{\pm 0.011}$	$0.463_{\pm 0.060}$
	Laplace	$0.494_{\pm 0.030}$	$0.717_{\pm 0.089}$	$0.312_{\pm 0.018}$
	BLoB	$0.114_{\pm 0.004}$	$0.351_{\pm 0.012}$	$0.294_{\pm 0.009}$
	ScalaBL	$0.144_{\pm 0.005}$	$0.416_{\pm 0.016}$	$0.285_{\pm 0.006}$
	TFB	$0.193_{\pm 0.016}$	$0.627_{\pm 0.036}$	$0.429_{\pm 0.061}$
Brier (↓)	MLE	$0.082_{\pm 0.005}$	$0.208_{\pm 0.005}$	$0.153_{\pm 0.025}$
	MAP	$0.081_{\pm 0.002}$	$0.217_{\pm 0.009}$	$0.146_{\pm 0.012}$
	TempScale	$0.070_{\pm 0.004}$	$0.175_{\pm 0.003}$	$0.138_{\pm 0.008}$
	MCDropout	$0.083_{\pm 0.003}$	$0.210_{\pm 0.003}$	$0.145_{\pm 0.009}$
	Ensemble	$0.075_{\pm 0.001}$	$0.184_{\pm 0.002}$	$0.143_{\pm 0.002}$
	Laplace	$0.215_{\pm 0.016}$	$0.355_{\pm 0.047}$	$0.155_{\pm 0.006}$
	BLoB	$0.058_{\pm 0.002}$	$0.167_{\pm 0.004}$	$0.135_{\pm 0.004}$
	ScalaBL	$0.069_{\pm 0.001}$	$0.175_{\pm 0.004}$	$0.139_{\pm 0.003}$
	TFB	$0.071_{\pm 0.003}$	$0.189_{\pm 0.003}$	$0.143_{\pm 0.017}$

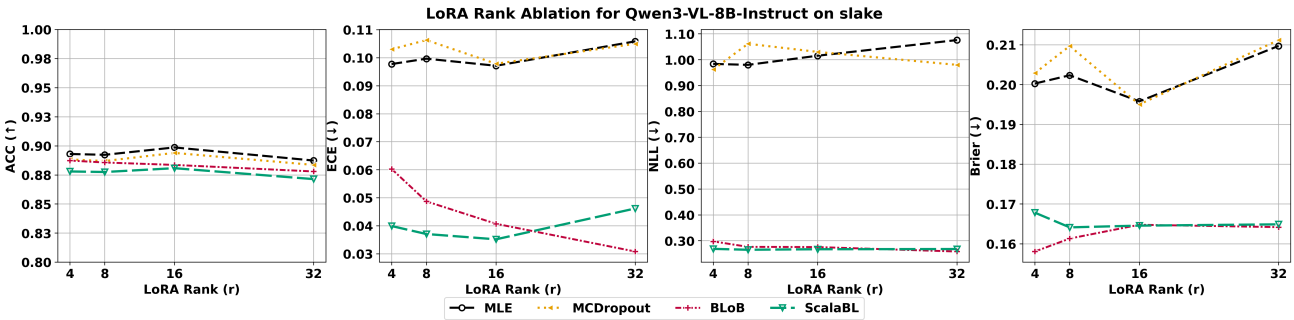


Figure 9: Ablation of LoRA rank on SLAKE using Qwen3-VL-8B-Instruct.

Table 14: In-Distribution Performance Comparison for Qwen3-8B

Metric	Method	ARC-Easy	ARC-Challenge	obqa
ACC (↑)	MLE	$0.966_{\pm 0.002}$	$0.912_{\pm 0.003}$	$0.924_{\pm 0.008}$
	MAP	$0.967_{\pm 0.004}$	$0.910_{\pm 0.003}$	$0.925_{\pm 0.003}$
	TempScale	$0.966_{\pm 0.002}$	$0.912_{\pm 0.003}$	$0.924_{\pm 0.008}$
	MCDropout	$0.965_{\pm 0.002}$	$0.913_{\pm 0.003}$	$0.924_{\pm 0.002}$
	Ensemble	$0.968_{\pm 0.002}$	$0.917_{\pm 0.002}$	$0.920_{\pm 0.004}$
	Laplace	$0.963_{\pm 0.005}$	$0.879_{\pm 0.009}$	$0.924_{\pm 0.004}$
	BLoB	$0.973_{\pm 0.002}$	$0.917_{\pm 0.002}$	$0.923_{\pm 0.006}$
	ScalaBL	$0.971_{\pm 0.002}$	$0.922_{\pm 0.003}$	$0.919_{\pm 0.004}$
	TFB	$0.967_{\pm 0.004}$	$0.911_{\pm 0.002}$	$0.923_{\pm 0.005}$
ECE (↓)	MLE	$0.028_{\pm 0.003}$	$0.079_{\pm 0.004}$	$0.056_{\pm 0.007}$
	MAP	$0.028_{\pm 0.004}$	$0.082_{\pm 0.004}$	$0.057_{\pm 0.001}$
	TempScale	$0.013_{\pm 0.005}$	$0.028_{\pm 0.004}$	$0.033_{\pm 0.012}$
	MCDropout	$0.029_{\pm 0.003}$	$0.079_{\pm 0.002}$	$0.060_{\pm 0.002}$
	Ensemble	$0.023_{\pm 0.001}$	$0.065_{\pm 0.003}$	$0.057_{\pm 0.007}$
	Laplace	$0.351_{\pm 0.052}$	$0.429_{\pm 0.012}$	$0.071_{\pm 0.015}$
	BLoB	$0.010_{\pm 0.002}$	$0.030_{\pm 0.003}$	$0.033_{\pm 0.007}$
	ScalaBL	$0.013_{\pm 0.001}$	$0.036_{\pm 0.003}$	$0.026_{\pm 0.004}$
	TFB	$0.018_{\pm 0.006}$	$0.065_{\pm 0.003}$	$0.041_{\pm 0.004}$
NLL (↓)	MLE	$0.226_{\pm 0.049}$	$0.833_{\pm 0.068}$	$0.350_{\pm 0.034}$
	MAP	$0.260_{\pm 0.063}$	$0.919_{\pm 0.148}$	$0.341_{\pm 0.026}$
	TempScale	$0.105_{\pm 0.008}$	$0.277_{\pm 0.005}$	$0.221_{\pm 0.013}$
	MCDropout	$0.262_{\pm 0.059}$	$0.839_{\pm 0.056}$	$0.361_{\pm 0.020}$
	Ensemble	$0.200_{\pm 0.024}$	$0.640_{\pm 0.041}$	$0.285_{\pm 0.022}$
	Laplace	$0.517_{\pm 0.077}$	$0.853_{\pm 0.035}$	$0.245_{\pm 0.015}$
	BLoB	$0.090_{\pm 0.005}$	$0.258_{\pm 0.009}$	$0.217_{\pm 0.013}$
	ScalaBL	$0.104_{\pm 0.003}$	$0.264_{\pm 0.007}$	$0.210_{\pm 0.004}$
	TFB	$0.155_{\pm 0.041}$	$0.578_{\pm 0.030}$	$0.268_{\pm 0.023}$
Brier (↓)	MLE	$0.058_{\pm 0.007}$	$0.164_{\pm 0.007}$	$0.127_{\pm 0.010}$
	MAP	$0.059_{\pm 0.007}$	$0.167_{\pm 0.005}$	$0.123_{\pm 0.006}$
	TempScale	$0.051_{\pm 0.004}$	$0.131_{\pm 0.002}$	$0.110_{\pm 0.006}$
	MCDropout	$0.060_{\pm 0.004}$	$0.162_{\pm 0.003}$	$0.129_{\pm 0.003}$
	Ensemble	$0.054_{\pm 0.002}$	$0.145_{\pm 0.002}$	$0.122_{\pm 0.006}$
	Laplace	$0.221_{\pm 0.042}$	$0.423_{\pm 0.021}$	$0.121_{\pm 0.005}$
	BLoB	$0.042_{\pm 0.001}$	$0.122_{\pm 0.002}$	$0.109_{\pm 0.006}$
	ScalaBL	$0.045_{\pm 0.002}$	$0.119_{\pm 0.003}$	$0.111_{\pm 0.003}$
	TFB	$0.052_{\pm 0.006}$	$0.146_{\pm 0.005}$	$0.116_{\pm 0.008}$

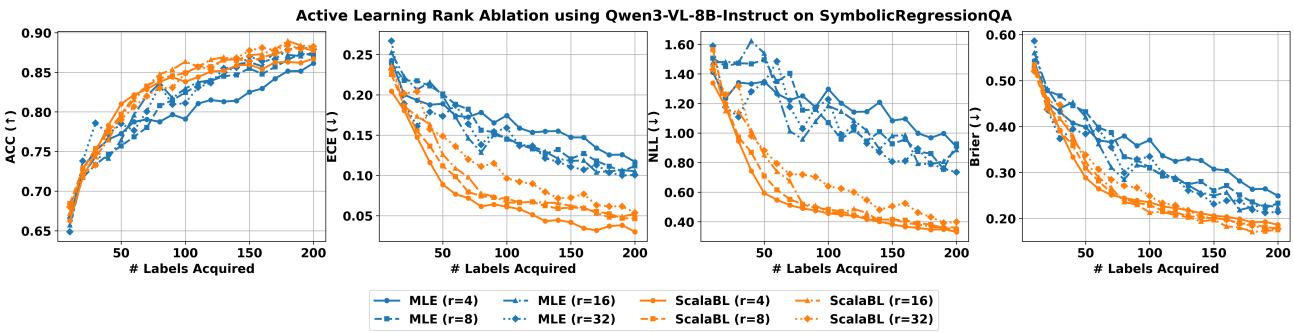


Figure 10: Ablation of LoRA rank on SRQA Active Learning using Qwen3-VL-8B-Instruct.

Table 15: In-Distribution Performance Comparison for Qwen3-14B

Metric	Method	ARC-Easy	ARC-Challenge	obqa
ACC (↑)	MLE	$0.980_{\pm 0.001}$	$0.935_{\pm 0.002}$	$0.946_{\pm 0.005}$
	MAP	$0.979_{\pm 0.001}$	$0.935_{\pm 0.002}$	$0.946_{\pm 0.007}$
	TempScale	$0.980_{\pm 0.001}$	$0.935_{\pm 0.002}$	$0.946_{\pm 0.005}$
	MCDropout	$0.980_{\pm 0.001}$	$0.935_{\pm 0.002}$	$0.948_{\pm 0.002}$
	Ensemble	$0.979_{\pm 0.001}$	$0.935_{\pm 0.002}$	$0.957_{\pm 0.003}$
	Laplace	$0.972_{\pm 0.003}$	$0.901_{\pm 0.011}$	$0.947_{\pm 0.005}$
	BLoB	$0.979_{\pm 0.001}$	$0.936_{\pm 0.003}$	$0.952_{\pm 0.005}$
	ScalaBL	$0.980_{\pm 0.001}$	$0.936_{\pm 0.002}$	$0.952_{\pm 0.004}$
	TFB	$0.979_{\pm 0.001}$	$0.935_{\pm 0.003}$	$0.946_{\pm 0.007}$
ECE (↓)	MLE	$0.017_{\pm 0.001}$	$0.057_{\pm 0.002}$	$0.046_{\pm 0.003}$
	MAP	$0.017_{\pm 0.001}$	$0.057_{\pm 0.003}$	$0.044_{\pm 0.006}$
	TempScale	$0.010_{\pm 0.002}$	$0.019_{\pm 0.005}$	$0.031_{\pm 0.003}$
	MCDropout	$0.017_{\pm 0.001}$	$0.057_{\pm 0.002}$	$0.041_{\pm 0.004}$
	Ensemble	$0.012_{\pm 0.001}$	$0.049_{\pm 0.003}$	$0.023_{\pm 0.003}$
	Laplace	$0.408_{\pm 0.017}$	$0.457_{\pm 0.016}$	$0.152_{\pm 0.014}$
	BLoB	$0.009_{\pm 0.001}$	$0.034_{\pm 0.003}$	$0.025_{\pm 0.005}$
	ScalaBL	$0.012_{\pm 0.001}$	$0.039_{\pm 0.003}$	$0.018_{\pm 0.005}$
	TFB	$0.011_{\pm 0.001}$	$0.045_{\pm 0.004}$	$0.027_{\pm 0.003}$
NLL (↓)	MLE	$0.150_{\pm 0.012}$	$0.544_{\pm 0.012}$	$0.281_{\pm 0.031}$
	MAP	$0.155_{\pm 0.012}$	$0.543_{\pm 0.021}$	$0.283_{\pm 0.040}$
	TempScale	$0.074_{\pm 0.004}$	$0.219_{\pm 0.009}$	$0.158_{\pm 0.010}$
	MCDropout	$0.149_{\pm 0.011}$	$0.535_{\pm 0.014}$	$0.264_{\pm 0.025}$
	Ensemble	$0.082_{\pm 0.003}$	$0.328_{\pm 0.004}$	$0.131_{\pm 0.009}$
	Laplace	$0.592_{\pm 0.036}$	$0.851_{\pm 0.011}$	$0.276_{\pm 0.019}$
	BLoB	$0.080_{\pm 0.005}$	$0.261_{\pm 0.013}$	$0.159_{\pm 0.015}$
	ScalaBL	$0.089_{\pm 0.007}$	$0.269_{\pm 0.007}$	$0.134_{\pm 0.004}$
	TFB	$0.104_{\pm 0.006}$	$0.374_{\pm 0.009}$	$0.188_{\pm 0.015}$
Brier (↓)	MLE	$0.036_{\pm 0.002}$	$0.119_{\pm 0.004}$	$0.091_{\pm 0.009}$
	MAP	$0.037_{\pm 0.002}$	$0.118_{\pm 0.004}$	$0.089_{\pm 0.007}$
	TempScale	$0.033_{\pm 0.002}$	$0.103_{\pm 0.003}$	$0.076_{\pm 0.006}$
	MCDropout	$0.036_{\pm 0.002}$	$0.118_{\pm 0.003}$	$0.086_{\pm 0.006}$
	Ensemble	$0.034_{\pm 0.001}$	$0.110_{\pm 0.003}$	$0.064_{\pm 0.005}$
	Laplace	$0.263_{\pm 0.025}$	$0.424_{\pm 0.006}$	$0.120_{\pm 0.009}$
	BLoB	$0.032_{\pm 0.001}$	$0.100_{\pm 0.003}$	$0.072_{\pm 0.006}$
	ScalaBL	$0.034_{\pm 0.001}$	$0.104_{\pm 0.002}$	$0.069_{\pm 0.002}$
	TFB	$0.033_{\pm 0.002}$	$0.111_{\pm 0.005}$	$0.080_{\pm 0.005}$

Table 16: In-Distribution Performance Comparison for Qwen3-0.6B

Metric	Method	winogrande_xs	winogrande_s	winogrande_m	winogrande_l
ACC (↑)	MLE	$0.555_{\pm 0.016}$	$0.574_{\pm 0.011}$	$0.576_{\pm 0.002}$	$0.588_{\pm 0.008}$
	MAP	$0.543_{\pm 0.015}$	$0.574_{\pm 0.010}$	$0.575_{\pm 0.004}$	$0.586_{\pm 0.012}$
	TempScale	$0.555_{\pm 0.016}$	$0.574_{\pm 0.011}$	$0.576_{\pm 0.002}$	$0.588_{\pm 0.008}$
	MCDropout	$0.555_{\pm 0.008}$	$0.575_{\pm 0.007}$	$0.576_{\pm 0.004}$	$0.590_{\pm 0.004}$
	Ensemble	$0.557_{\pm 0.005}$	$0.573_{\pm 0.002}$	$0.586_{\pm 0.008}$	$0.599_{\pm 0.003}$
	Laplace	$0.554_{\pm 0.012}$	$0.574_{\pm 0.009}$	$0.576_{\pm 0.008}$	$0.589_{\pm 0.004}$
	BLoB	$0.502_{\pm 0.013}$	$0.501_{\pm 0.012}$	$0.528_{\pm 0.032}$	$0.581_{\pm 0.009}$
	ScalaBL	$0.541_{\pm 0.026}$	$0.549_{\pm 0.032}$	$0.515_{\pm 0.021}$	$0.554_{\pm 0.022}$
	TFB	$0.554_{\pm 0.017}$	$0.575_{\pm 0.010}$	$0.562_{\pm 0.013}$	$0.569_{\pm 0.025}$
ECE (↓)	MLE	$0.407_{\pm 0.009}$	$0.401_{\pm 0.011}$	$0.164_{\pm 0.089}$	$0.088_{\pm 0.036}$
	MAP	$0.425_{\pm 0.014}$	$0.399_{\pm 0.014}$	$0.150_{\pm 0.070}$	$0.085_{\pm 0.033}$
	TempScale	$0.085_{\pm 0.017}$	$0.135_{\pm 0.009}$	$0.101_{\pm 0.017}$	$0.130_{\pm 0.035}$
	MCDropout	$0.404_{\pm 0.005}$	$0.384_{\pm 0.006}$	$0.138_{\pm 0.056}$	$0.080_{\pm 0.032}$
	Ensemble	$0.287_{\pm 0.012}$	$0.283_{\pm 0.003}$	$0.141_{\pm 0.039}$	$0.077_{\pm 0.011}$
	Laplace	$0.026_{\pm 0.014}$	$0.063_{\pm 0.017}$	$0.104_{\pm 0.048}$	$0.073_{\pm 0.031}$
	BLoB	$0.020_{\pm 0.010}$	$0.016_{\pm 0.009}$	$0.022_{\pm 0.008}$	$0.055_{\pm 0.010}$
	ScalaBL	$0.102_{\pm 0.042}$	$0.038_{\pm 0.009}$	$0.024_{\pm 0.010}$	$0.031_{\pm 0.013}$
	TFB	$0.280_{\pm 0.018}$	$0.285_{\pm 0.009}$	$0.116_{\pm 0.049}$	$0.069_{\pm 0.004}$
NLL (↓)	MLE	$3.828_{\pm 0.481}$	$3.978_{\pm 0.438}$	$0.984_{\pm 0.418}$	$0.700_{\pm 0.016}$
	MAP	$4.075_{\pm 0.225}$	$3.820_{\pm 0.497}$	$0.884_{\pm 0.250}$	$0.697_{\pm 0.015}$
	TempScale	$0.704_{\pm 0.007}$	$0.726_{\pm 0.005}$	$0.735_{\pm 0.009}$	$0.750_{\pm 0.030}$
	MCDropout	$3.712_{\pm 0.268}$	$3.349_{\pm 0.163}$	$0.875_{\pm 0.215}$	$0.695_{\pm 0.012}$
	Ensemble	$1.805_{\pm 0.108}$	$1.818_{\pm 0.022}$	$0.807_{\pm 0.080}$	$0.683_{\pm 0.004}$
	Laplace	$0.687_{\pm 0.002}$	$0.708_{\pm 0.018}$	$0.774_{\pm 0.130}$	$0.688_{\pm 0.009}$
	BLoB	$0.694_{\pm 0.001}$	$0.694_{\pm 0.001}$	$0.687_{\pm 0.007}$	$0.676_{\pm 0.002}$
	ScalaBL	$0.734_{\pm 0.027}$	$0.689_{\pm 0.005}$	$0.692_{\pm 0.004}$	$0.682_{\pm 0.006}$
	TFB	$1.606_{\pm 0.218}$	$1.810_{\pm 0.186}$	$0.774_{\pm 0.118}$	$0.692_{\pm 0.010}$
Brier (↓)	MLE	$0.837_{\pm 0.019}$	$0.810_{\pm 0.018}$	$0.566_{\pm 0.077}$	$0.493_{\pm 0.008}$
	MAP	$0.864_{\pm 0.025}$	$0.810_{\pm 0.022}$	$0.552_{\pm 0.058}$	$0.492_{\pm 0.007}$
	TempScale	$0.509_{\pm 0.007}$	$0.523_{\pm 0.004}$	$0.516_{\pm 0.009}$	$0.517_{\pm 0.013}$
	MCDropout	$0.830_{\pm 0.011}$	$0.789_{\pm 0.010}$	$0.551_{\pm 0.051}$	$0.491_{\pm 0.006}$
	Ensemble	$0.680_{\pm 0.006}$	$0.668_{\pm 0.004}$	$0.539_{\pm 0.028}$	$0.482_{\pm 0.001}$
	Laplace	$0.494_{\pm 0.002}$	$0.501_{\pm 0.009}$	$0.516_{\pm 0.033}$	$0.487_{\pm 0.005}$
	BLoB	$0.501_{\pm 0.001}$	$0.501_{\pm 0.001}$	$0.494_{\pm 0.007}$	$0.482_{\pm 0.002}$
	ScalaBL	$0.525_{\pm 0.016}$	$0.495_{\pm 0.005}$	$0.499_{\pm 0.004}$	$0.489_{\pm 0.007}$
	TFB	$0.684_{\pm 0.019}$	$0.666_{\pm 0.012}$	$0.527_{\pm 0.031}$	$0.495_{\pm 0.011}$

Table 17: In-Distribution Performance Comparison for Qwen3-1.7B

Metric	Method	winogrande_xs	winogrande_s	winogrande_m	winogrande_l
ACC (↑)	MLE	$0.578_{\pm 0.003}$	$0.597_{\pm 0.005}$	$0.623_{\pm 0.002}$	$0.657_{\pm 0.005}$
	MAP	$0.588_{\pm 0.009}$	$0.602_{\pm 0.008}$	$0.623_{\pm 0.006}$	$0.657_{\pm 0.005}$
	TempScale	$0.578_{\pm 0.003}$	$0.597_{\pm 0.005}$	$0.623_{\pm 0.002}$	$0.657_{\pm 0.005}$
	MCDropout	$0.584_{\pm 0.009}$	$0.599_{\pm 0.005}$	$0.627_{\pm 0.010}$	$0.659_{\pm 0.003}$
	Ensemble	$0.585_{\pm 0.006}$	$0.604_{\pm 0.010}$	$0.630_{\pm 0.005}$	$0.659_{\pm 0.007}$
	Laplace	$0.580_{\pm 0.004}$	$0.597_{\pm 0.007}$	$0.622_{\pm 0.003}$	$0.660_{\pm 0.005}$
	BLoB	$0.524_{\pm 0.050}$	$0.534_{\pm 0.047}$	$0.570_{\pm 0.046}$	$0.640_{\pm 0.009}$
	ScalaBL	$0.519_{\pm 0.021}$	$0.510_{\pm 0.011}$	$0.502_{\pm 0.010}$	$0.580_{\pm 0.005}$
	TFB	$0.575_{\pm 0.005}$	$0.602_{\pm 0.002}$	$0.625_{\pm 0.002}$	$0.625_{\pm 0.010}$
ECE (↓)	MLE	$0.398_{\pm 0.004}$	$0.381_{\pm 0.007}$	$0.285_{\pm 0.010}$	$0.082_{\pm 0.011}$
	MAP	$0.389_{\pm 0.013}$	$0.378_{\pm 0.008}$	$0.284_{\pm 0.011}$	$0.081_{\pm 0.011}$
	TempScale	$0.146_{\pm 0.006}$	$0.162_{\pm 0.009}$	$0.131_{\pm 0.002}$	$0.116_{\pm 0.006}$
	MCDropout	$0.388_{\pm 0.012}$	$0.375_{\pm 0.005}$	$0.273_{\pm 0.015}$	$0.082_{\pm 0.010}$
	Ensemble	$0.325_{\pm 0.013}$	$0.296_{\pm 0.004}$	$0.262_{\pm 0.004}$	$0.075_{\pm 0.004}$
	Laplace	$0.038_{\pm 0.009}$	$0.030_{\pm 0.006}$	$0.156_{\pm 0.005}$	$0.062_{\pm 0.009}$
	BLoB	$0.051_{\pm 0.051}$	$0.038_{\pm 0.036}$	$0.036_{\pm 0.023}$	$0.068_{\pm 0.006}$
	ScalaBL	$0.056_{\pm 0.035}$	$0.036_{\pm 0.008}$	$0.035_{\pm 0.009}$	$0.055_{\pm 0.009}$
	TFB	$0.323_{\pm 0.011}$	$0.311_{\pm 0.008}$	$0.206_{\pm 0.010}$	$0.042_{\pm 0.003}$
NLL (↓)	MLE	$4.515_{\pm 0.303}$	$4.537_{\pm 0.156}$	$1.645_{\pm 0.161}$	$0.650_{\pm 0.010}$
	MAP	$4.442_{\pm 0.321}$	$4.422_{\pm 0.169}$	$1.593_{\pm 0.171}$	$0.649_{\pm 0.009}$
	TempScale	$0.738_{\pm 0.002}$	$0.746_{\pm 0.007}$	$0.721_{\pm 0.008}$	$0.694_{\pm 0.006}$
	MCDropout	$4.365_{\pm 0.287}$	$4.272_{\pm 0.215}$	$1.457_{\pm 0.140}$	$0.649_{\pm 0.009}$
	Ensemble	$2.891_{\pm 0.165}$	$2.533_{\pm 0.102}$	$1.444_{\pm 0.057}$	$0.642_{\pm 0.003}$
	Laplace	$0.678_{\pm 0.001}$	$0.668_{\pm 0.002}$	$0.876_{\pm 0.052}$	$0.634_{\pm 0.007}$
	BLoB	$0.703_{\pm 0.017}$	$0.691_{\pm 0.004}$	$0.675_{\pm 0.015}$	$0.638_{\pm 0.004}$
	ScalaBL	$0.705_{\pm 0.015}$	$0.696_{\pm 0.002}$	$0.697_{\pm 0.002}$	$0.678_{\pm 0.003}$
	TFB	$2.577_{\pm 0.240}$	$2.631_{\pm 0.080}$	$1.032_{\pm 0.069}$	$0.647_{\pm 0.004}$
Brier (↓)	MLE	$0.806_{\pm 0.005}$	$0.772_{\pm 0.011}$	$0.636_{\pm 0.011}$	$0.444_{\pm 0.005}$
	MAP	$0.789_{\pm 0.021}$	$0.765_{\pm 0.013}$	$0.631_{\pm 0.015}$	$0.444_{\pm 0.005}$
	TempScale	$0.530_{\pm 0.001}$	$0.530_{\pm 0.005}$	$0.497_{\pm 0.003}$	$0.460_{\pm 0.004}$
	MCDropout	$0.790_{\pm 0.019}$	$0.764_{\pm 0.007}$	$0.617_{\pm 0.015}$	$0.445_{\pm 0.005}$
	Ensemble	$0.706_{\pm 0.018}$	$0.656_{\pm 0.007}$	$0.601_{\pm 0.009}$	$0.441_{\pm 0.001}$
	Laplace	$0.485_{\pm 0.001}$	$0.474_{\pm 0.001}$	$0.514_{\pm 0.008}$	$0.437_{\pm 0.004}$
	BLoB	$0.503_{\pm 0.004}$	$0.496_{\pm 0.008}$	$0.482_{\pm 0.015}$	$0.446_{\pm 0.003}$
	ScalaBL	$0.509_{\pm 0.010}$	$0.503_{\pm 0.002}$	$0.504_{\pm 0.002}$	$0.485_{\pm 0.003}$
	TFB	$0.707_{\pm 0.018}$	$0.684_{\pm 0.008}$	$0.555_{\pm 0.009}$	$0.453_{\pm 0.004}$

Table 18: In-Distribution Performance Comparison for Qwen3-4B

Metric	Method	winogrande_xs	winogrande_s	winogrande_m	winogrande_l
ACC (↑)	MLE	$0.685_{\pm 0.004}$	$0.711_{\pm 0.004}$	$0.756_{\pm 0.009}$	$0.807_{\pm 0.006}$
	MAP	$0.687_{\pm 0.007}$	$0.711_{\pm 0.013}$	$0.762_{\pm 0.005}$	$0.805_{\pm 0.005}$
	TempScale	$0.685_{\pm 0.004}$	$0.711_{\pm 0.004}$	$0.756_{\pm 0.009}$	$0.807_{\pm 0.006}$
	MCDropout	$0.685_{\pm 0.008}$	$0.709_{\pm 0.004}$	$0.760_{\pm 0.008}$	$0.809_{\pm 0.003}$
	Ensemble	$0.680_{\pm 0.002}$	$0.721_{\pm 0.008}$	$0.778_{\pm 0.008}$	$0.814_{\pm 0.003}$
	Laplace	$0.680_{\pm 0.005}$	$0.712_{\pm 0.006}$	$0.756_{\pm 0.008}$	$0.805_{\pm 0.007}$
	BLoB	$0.695_{\pm 0.006}$	$0.727_{\pm 0.005}$	$0.769_{\pm 0.008}$	$0.800_{\pm 0.005}$
	ScalaBL	$0.683_{\pm 0.006}$	$0.716_{\pm 0.007}$	$0.740_{\pm 0.006}$	$0.767_{\pm 0.005}$
	TFB	$0.683_{\pm 0.004}$	$0.712_{\pm 0.007}$	$0.752_{\pm 0.009}$	$0.789_{\pm 0.005}$
ECE (↓)	MLE	$0.294_{\pm 0.005}$	$0.268_{\pm 0.004}$	$0.210_{\pm 0.014}$	$0.076_{\pm 0.005}$
	MAP	$0.288_{\pm 0.005}$	$0.272_{\pm 0.011}$	$0.206_{\pm 0.005}$	$0.078_{\pm 0.006}$
	TempScale	$0.136_{\pm 0.002}$	$0.138_{\pm 0.008}$	$0.125_{\pm 0.008}$	$0.083_{\pm 0.005}$
	MCDropout	$0.290_{\pm 0.008}$	$0.270_{\pm 0.003}$	$0.208_{\pm 0.011}$	$0.077_{\pm 0.004}$
	Ensemble	$0.269_{\pm 0.004}$	$0.210_{\pm 0.005}$	$0.141_{\pm 0.008}$	$0.062_{\pm 0.006}$
	Laplace	$0.140_{\pm 0.004}$	$0.136_{\pm 0.005}$	$0.035_{\pm 0.011}$	$0.039_{\pm 0.006}$
	BLoB	$0.134_{\pm 0.006}$	$0.099_{\pm 0.011}$	$0.052_{\pm 0.005}$	$0.046_{\pm 0.006}$
	ScalaBL	$0.175_{\pm 0.008}$	$0.110_{\pm 0.005}$	$0.037_{\pm 0.004}$	$0.031_{\pm 0.004}$
	TFB	$0.234_{\pm 0.004}$	$0.220_{\pm 0.004}$	$0.127_{\pm 0.021}$	$0.051_{\pm 0.012}$
NLL (↓)	MLE	$3.433_{\pm 0.072}$	$2.789_{\pm 0.202}$	$1.243_{\pm 0.169}$	$0.465_{\pm 0.012}$
	MAP	$3.262_{\pm 0.147}$	$2.995_{\pm 0.013}$	$1.260_{\pm 0.110}$	$0.467_{\pm 0.012}$
	TempScale	$0.679_{\pm 0.005}$	$0.647_{\pm 0.009}$	$0.608_{\pm 0.024}$	$0.475_{\pm 0.011}$
	MCDropout	$3.323_{\pm 0.183}$	$2.773_{\pm 0.330}$	$1.232_{\pm 0.100}$	$0.464_{\pm 0.008}$
	Ensemble	$2.860_{\pm 0.034}$	$1.806_{\pm 0.060}$	$0.838_{\pm 0.042}$	$0.436_{\pm 0.008}$
	Laplace	$0.653_{\pm 0.002}$	$0.615_{\pm 0.004}$	$0.515_{\pm 0.009}$	$0.430_{\pm 0.011}$
	BLoB	$0.718_{\pm 0.025}$	$0.591_{\pm 0.021}$	$0.482_{\pm 0.008}$	$0.442_{\pm 0.003}$
	ScalaBL	$0.891_{\pm 0.034}$	$0.621_{\pm 0.010}$	$0.528_{\pm 0.005}$	$0.491_{\pm 0.003}$
	TFB	$1.785_{\pm 0.064}$	$1.630_{\pm 0.085}$	$0.695_{\pm 0.083}$	$0.471_{\pm 0.012}$
Brier (↓)	MLE	$0.600_{\pm 0.007}$	$0.548_{\pm 0.006}$	$0.443_{\pm 0.023}$	$0.284_{\pm 0.005}$
	MAP	$0.592_{\pm 0.010}$	$0.552_{\pm 0.023}$	$0.432_{\pm 0.014}$	$0.285_{\pm 0.006}$
	TempScale	$0.454_{\pm 0.002}$	$0.427_{\pm 0.005}$	$0.376_{\pm 0.014}$	$0.287_{\pm 0.007}$
	MCDropout	$0.595_{\pm 0.013}$	$0.553_{\pm 0.005}$	$0.437_{\pm 0.017}$	$0.284_{\pm 0.004}$
	Ensemble	$0.570_{\pm 0.011}$	$0.481_{\pm 0.004}$	$0.358_{\pm 0.012}$	$0.270_{\pm 0.005}$
	Laplace	$0.460_{\pm 0.002}$	$0.424_{\pm 0.004}$	$0.338_{\pm 0.009}$	$0.272_{\pm 0.005}$
	BLoB	$0.430_{\pm 0.007}$	$0.378_{\pm 0.007}$	$0.315_{\pm 0.006}$	$0.284_{\pm 0.003}$
	ScalaBL	$0.477_{\pm 0.008}$	$0.403_{\pm 0.004}$	$0.352_{\pm 0.004}$	$0.322_{\pm 0.003}$
	TFB	$0.534_{\pm 0.003}$	$0.495_{\pm 0.002}$	$0.378_{\pm 0.019}$	$0.304_{\pm 0.009}$

Table 19: In-Distribution Performance Comparison for Qwen3-8B

Metric	Method	winogrande_xs	winogrande_s	winogrande_m	winogrande_l
ACC (↑)	MLE	$0.717_{\pm 0.007}$	$0.760_{\pm 0.005}$	$0.813_{\pm 0.004}$	$0.859_{\pm 0.004}$
	MAP	$0.721_{\pm 0.005}$	$0.764_{\pm 0.006}$	$0.809_{\pm 0.002}$	$0.858_{\pm 0.004}$
	TempScale	$0.717_{\pm 0.007}$	$0.760_{\pm 0.005}$	$0.813_{\pm 0.004}$	$0.859_{\pm 0.004}$
	MCDropout	$0.718_{\pm 0.004}$	$0.759_{\pm 0.003}$	$0.817_{\pm 0.002}$	$0.857_{\pm 0.004}$
	Ensemble	$0.713_{\pm 0.001}$	$0.760_{\pm 0.006}$	$0.821_{\pm 0.006}$	$0.852_{\pm 0.007}$
	Laplace	$0.712_{\pm 0.006}$	$0.760_{\pm 0.004}$	$0.812_{\pm 0.005}$	$0.859_{\pm 0.004}$
	BLoB	$0.739_{\pm 0.006}$	$0.771_{\pm 0.010}$	$0.826_{\pm 0.004}$	$0.850_{\pm 0.009}$
	ScalaBL	$0.734_{\pm 0.004}$	$0.761_{\pm 0.004}$	$0.801_{\pm 0.005}$	$0.825_{\pm 0.005}$
	TFB	$0.715_{\pm 0.007}$	$0.761_{\pm 0.003}$	$0.812_{\pm 0.006}$	$0.857_{\pm 0.002}$
ECE (↓)	MLE	$0.264_{\pm 0.005}$	$0.227_{\pm 0.005}$	$0.170_{\pm 0.008}$	$0.053_{\pm 0.007}$
	MAP	$0.259_{\pm 0.005}$	$0.222_{\pm 0.008}$	$0.173_{\pm 0.006}$	$0.054_{\pm 0.007}$
	TempScale	$0.132_{\pm 0.008}$	$0.126_{\pm 0.004}$	$0.103_{\pm 0.004}$	$0.058_{\pm 0.005}$
	MCDropout	$0.261_{\pm 0.004}$	$0.225_{\pm 0.005}$	$0.162_{\pm 0.004}$	$0.053_{\pm 0.007}$
	Ensemble	$0.252_{\pm 0.006}$	$0.198_{\pm 0.010}$	$0.119_{\pm 0.007}$	$0.052_{\pm 0.007}$
	Laplace	$0.177_{\pm 0.006}$	$0.189_{\pm 0.005}$	$0.099_{\pm 0.069}$	$0.023_{\pm 0.009}$
	BLoB	$0.126_{\pm 0.006}$	$0.108_{\pm 0.009}$	$0.066_{\pm 0.006}$	$0.035_{\pm 0.006}$
	ScalaBL	$0.152_{\pm 0.004}$	$0.125_{\pm 0.004}$	$0.041_{\pm 0.006}$	$0.023_{\pm 0.005}$
	TFB	$0.231_{\pm 0.005}$	$0.199_{\pm 0.002}$	$0.140_{\pm 0.012}$	$0.041_{\pm 0.013}$
NLL (↓)	MLE	$3.030_{\pm 0.110}$	$2.614_{\pm 0.408}$	$1.415_{\pm 0.548}$	$0.362_{\pm 0.012}$
	MAP	$3.009_{\pm 0.150}$	$2.351_{\pm 0.102}$	$1.406_{\pm 0.529}$	$0.361_{\pm 0.012}$
	TempScale	$0.636_{\pm 0.017}$	$0.583_{\pm 0.009}$	$0.516_{\pm 0.016}$	$0.369_{\pm 0.009}$
	MCDropout	$2.994_{\pm 0.086}$	$2.373_{\pm 0.109}$	$1.132_{\pm 0.184}$	$0.359_{\pm 0.010}$
	Ensemble	$2.540_{\pm 0.336}$	$1.803_{\pm 0.155}$	$0.738_{\pm 0.052}$	$0.358_{\pm 0.012}$
	Laplace	$0.652_{\pm 0.002}$	$0.605_{\pm 0.005}$	$0.466_{\pm 0.053}$	$0.338_{\pm 0.008}$
	BLoB	$0.722_{\pm 0.022}$	$0.613_{\pm 0.022}$	$0.439_{\pm 0.008}$	$0.356_{\pm 0.010}$
	ScalaBL	$0.803_{\pm 0.015}$	$0.648_{\pm 0.013}$	$0.438_{\pm 0.004}$	$0.390_{\pm 0.005}$
	TFB	$2.060_{\pm 0.055}$	$1.867_{\pm 0.301}$	$0.993_{\pm 0.353}$	$0.361_{\pm 0.004}$
Brier (↓)	MLE	$0.537_{\pm 0.011}$	$0.459_{\pm 0.007}$	$0.350_{\pm 0.013}$	$0.214_{\pm 0.006}$
	MAP	$0.532_{\pm 0.008}$	$0.452_{\pm 0.012}$	$0.353_{\pm 0.005}$	$0.214_{\pm 0.006}$
	TempScale	$0.413_{\pm 0.010}$	$0.366_{\pm 0.004}$	$0.300_{\pm 0.008}$	$0.216_{\pm 0.006}$
	MCDropout	$0.534_{\pm 0.009}$	$0.456_{\pm 0.009}$	$0.339_{\pm 0.008}$	$0.212_{\pm 0.006}$
	Ensemble	$0.520_{\pm 0.013}$	$0.425_{\pm 0.012}$	$0.291_{\pm 0.012}$	$0.217_{\pm 0.008}$
	Laplace	$0.459_{\pm 0.002}$	$0.414_{\pm 0.005}$	$0.300_{\pm 0.038}$	$0.206_{\pm 0.006}$
	BLoB	$0.390_{\pm 0.006}$	$0.341_{\pm 0.010}$	$0.259_{\pm 0.004}$	$0.218_{\pm 0.007}$
	ScalaBL	$0.411_{\pm 0.004}$	$0.366_{\pm 0.003}$	$0.280_{\pm 0.003}$	$0.246_{\pm 0.004}$
	TFB	$0.496_{\pm 0.011}$	$0.429_{\pm 0.003}$	$0.323_{\pm 0.011}$	$0.219_{\pm 0.002}$

Table 20: In-Distribution Performance Comparison for Qwen3-14B

Metric	Method	winogrande_xs	winogrande_s	winogrande_m	winogrande_l
ACC (↑)	MLE	$0.741_{\pm 0.002}$	$0.786_{\pm 0.004}$	$0.847_{\pm 0.007}$	$0.885_{\pm 0.005}$
	MAP	$0.744_{\pm 0.006}$	$0.787_{\pm 0.006}$	$0.848_{\pm 0.006}$	$0.884_{\pm 0.001}$
	TempScale	$0.741_{\pm 0.002}$	$0.786_{\pm 0.004}$	$0.847_{\pm 0.007}$	$0.885_{\pm 0.005}$
	MCDropout	$0.746_{\pm 0.006}$	$0.785_{\pm 0.009}$	$0.846_{\pm 0.004}$	$0.884_{\pm 0.002}$
	Ensemble	$0.748_{\pm 0.007}$	$0.794_{\pm 0.003}$	$0.856_{\pm 0.006}$	$0.867_{\pm 0.004}$
	Laplace	$0.741_{\pm 0.004}$	$0.785_{\pm 0.004}$	$0.846_{\pm 0.007}$	$0.887_{\pm 0.003}$
	BLoB	$0.766_{\pm 0.007}$	$0.799_{\pm 0.008}$	$0.847_{\pm 0.006}$	$0.883_{\pm 0.003}$
	ScalaBL	$0.755_{\pm 0.005}$	$0.790_{\pm 0.006}$	$0.830_{\pm 0.006}$	$0.861_{\pm 0.004}$
	TFB	$0.743_{\pm 0.003}$	$0.786_{\pm 0.003}$	$0.842_{\pm 0.010}$	$0.872_{\pm 0.004}$
ECE (↓)	MLE	$0.241_{\pm 0.003}$	$0.201_{\pm 0.006}$	$0.139_{\pm 0.009}$	$0.047_{\pm 0.006}$
	MAP	$0.240_{\pm 0.005}$	$0.198_{\pm 0.006}$	$0.140_{\pm 0.003}$	$0.046_{\pm 0.004}$
	TempScale	$0.124_{\pm 0.003}$	$0.108_{\pm 0.008}$	$0.090_{\pm 0.005}$	$0.055_{\pm 0.006}$
	MCDropout	$0.238_{\pm 0.005}$	$0.201_{\pm 0.010}$	$0.140_{\pm 0.004}$	$0.044_{\pm 0.003}$
	Ensemble	$0.209_{\pm 0.010}$	$0.160_{\pm 0.002}$	$0.044_{\pm 0.004}$	$0.023_{\pm 0.017}$
	Laplace	$0.182_{\pm 0.006}$	$0.186_{\pm 0.004}$	$0.110_{\pm 0.018}$	$0.023_{\pm 0.004}$
	BLoB	$0.119_{\pm 0.007}$	$0.098_{\pm 0.007}$	$0.070_{\pm 0.005}$	$0.028_{\pm 0.004}$
	ScalaBL	$0.147_{\pm 0.004}$	$0.102_{\pm 0.006}$	$0.031_{\pm 0.004}$	$0.018_{\pm 0.004}$
	TFB	$0.207_{\pm 0.004}$	$0.167_{\pm 0.010}$	$0.084_{\pm 0.025}$	$0.073_{\pm 0.012}$
NLL (↓)	MLE	$2.529_{\pm 0.169}$	$2.070_{\pm 0.225}$	$1.096_{\pm 0.200}$	$0.308_{\pm 0.003}$
	MAP	$2.565_{\pm 0.198}$	$2.127_{\pm 0.201}$	$1.136_{\pm 0.108}$	$0.312_{\pm 0.010}$
	TempScale	$0.582_{\pm 0.006}$	$0.531_{\pm 0.014}$	$0.491_{\pm 0.023}$	$0.317_{\pm 0.001}$
	MCDropout	$2.526_{\pm 0.118}$	$2.198_{\pm 0.189}$	$1.118_{\pm 0.126}$	$0.305_{\pm 0.007}$
	Ensemble	$1.715_{\pm 0.145}$	$0.951_{\pm 0.074}$	$0.360_{\pm 0.007}$	$0.327_{\pm 0.016}$
	Laplace	$0.618_{\pm 0.002}$	$0.571_{\pm 0.006}$	$0.425_{\pm 0.021}$	$0.287_{\pm 0.005}$
	BLoB	$0.710_{\pm 0.022}$	$0.576_{\pm 0.014}$	$0.416_{\pm 0.013}$	$0.292_{\pm 0.003}$
	ScalaBL	$0.783_{\pm 0.027}$	$0.562_{\pm 0.016}$	$0.391_{\pm 0.006}$	$0.333_{\pm 0.003}$
	TFB	$1.560_{\pm 0.119}$	$1.266_{\pm 0.184}$	$0.613_{\pm 0.121}$	$0.346_{\pm 0.013}$
Brier (↓)	MLE	$0.489_{\pm 0.008}$	$0.407_{\pm 0.012}$	$0.288_{\pm 0.016}$	$0.179_{\pm 0.002}$
	MAP	$0.487_{\pm 0.008}$	$0.404_{\pm 0.010}$	$0.286_{\pm 0.004}$	$0.182_{\pm 0.004}$
	TempScale	$0.376_{\pm 0.005}$	$0.331_{\pm 0.010}$	$0.260_{\pm 0.011}$	$0.181_{\pm 0.002}$
	MCDropout	$0.484_{\pm 0.010}$	$0.409_{\pm 0.017}$	$0.290_{\pm 0.007}$	$0.179_{\pm 0.003}$
	Ensemble	$0.443_{\pm 0.017}$	$0.355_{\pm 0.002}$	$0.219_{\pm 0.005}$	$0.198_{\pm 0.008}$
	Laplace	$0.426_{\pm 0.002}$	$0.381_{\pm 0.005}$	$0.259_{\pm 0.016}$	$0.171_{\pm 0.003}$
	BLoB	$0.356_{\pm 0.006}$	$0.306_{\pm 0.009}$	$0.235_{\pm 0.006}$	$0.173_{\pm 0.002}$
	ScalaBL	$0.381_{\pm 0.004}$	$0.321_{\pm 0.006}$	$0.245_{\pm 0.005}$	$0.203_{\pm 0.003}$
	TFB	$0.442_{\pm 0.007}$	$0.372_{\pm 0.012}$	$0.260_{\pm 0.015}$	$0.204_{\pm 0.009}$

Table 21: In-Distribution Performance Comparison for Qwen3-VL-2B-Instruct

Metric	Method	slake	mmstar	MathVerse	srqa
ACC (↑)	MLE	$0.873_{\pm 0.004}$	$0.558_{\pm 0.037}$	$0.383_{\pm 0.026}$	$0.949_{\pm 0.003}$
	MAP	$0.874_{\pm 0.001}$	$0.560_{\pm 0.022}$	$0.395_{\pm 0.024}$	$0.949_{\pm 0.002}$
	TempScale	$0.873_{\pm 0.004}$	$0.561_{\pm 0.033}$	$0.380_{\pm 0.022}$	$0.949_{\pm 0.003}$
	MCDropout	$0.878_{\pm 0.007}$	$0.575_{\pm 0.016}$	$0.383_{\pm 0.033}$	$0.947_{\pm 0.003}$
	Ensemble	$0.876_{\pm 0.009}$	$0.586_{\pm 0.012}$	$0.417_{\pm 0.004}$	$0.925_{\pm 0.004}$
	Laplace	$0.873_{\pm 0.002}$	$0.562_{\pm 0.013}$	$0.379_{\pm 0.043}$	$0.948_{\pm 0.004}$
	BLoB	$0.869_{\pm 0.006}$	$0.540_{\pm 0.015}$	$0.425_{\pm 0.027}$	$0.942_{\pm 0.003}$
	ScalaBL	$0.853_{\pm 0.013}$	$0.568_{\pm 0.019}$	$0.423_{\pm 0.042}$	$0.919_{\pm 0.002}$
	TFB	$0.873_{\pm 0.004}$	$0.561_{\pm 0.023}$	$0.390_{\pm 0.020}$	$0.944_{\pm 0.002}$
ECE (↓)	MLE	$0.117_{\pm 0.002}$	$0.403_{\pm 0.030}$	$0.571_{\pm 0.027}$	$0.016_{\pm 0.003}$
	MAP	$0.119_{\pm 0.004}$	$0.401_{\pm 0.028}$	$0.554_{\pm 0.026}$	$0.019_{\pm 0.004}$
	TempScale	$0.053_{\pm 0.008}$	$0.073_{\pm 0.013}$	$0.077_{\pm 0.028}$	$0.014_{\pm 0.003}$
	MCDropout	$0.115_{\pm 0.006}$	$0.387_{\pm 0.018}$	$0.558_{\pm 0.037}$	$0.016_{\pm 0.003}$
	Ensemble	$0.106_{\pm 0.006}$	$0.270_{\pm 0.004}$	$0.349_{\pm 0.012}$	$0.017_{\pm 0.005}$
	Laplace	$0.038_{\pm 0.004}$	$0.103_{\pm 0.038}$	$0.161_{\pm 0.058}$	$0.015_{\pm 0.001}$
	BLoB	$0.041_{\pm 0.008}$	$0.171_{\pm 0.017}$	$0.192_{\pm 0.023}$	$0.018_{\pm 0.003}$
	ScalaBL	$0.037_{\pm 0.008}$	$0.112_{\pm 0.027}$	$0.164_{\pm 0.018}$	$0.042_{\pm 0.005}$
	TFB	$0.105_{\pm 0.004}$	$0.310_{\pm 0.023}$	$0.433_{\pm 0.027}$	$0.027_{\pm 0.005}$
NLL (↓)	MLE	$1.073_{\pm 0.043}$	$5.391_{\pm 0.867}$	$7.403_{\pm 0.772}$	$0.132_{\pm 0.006}$
	MAP	$1.154_{\pm 0.095}$	$4.926_{\pm 0.390}$	$7.102_{\pm 0.576}$	$0.137_{\pm 0.005}$
	TempScale	$0.295_{\pm 0.008}$	$1.063_{\pm 0.027}$	$1.310_{\pm 0.016}$	$0.130_{\pm 0.005}$
	MCDropout	$1.129_{\pm 0.075}$	$5.248_{\pm 0.366}$	$8.331_{\pm 1.515}$	$0.134_{\pm 0.007}$
	Ensemble	$0.820_{\pm 0.082}$	$2.650_{\pm 0.077}$	$3.772_{\pm 0.252}$	$0.192_{\pm 0.003}$
	Laplace	$0.278_{\pm 0.007}$	$1.065_{\pm 0.037}$	$1.616_{\pm 0.299}$	$0.130_{\pm 0.004}$
	BLoB	$0.276_{\pm 0.009}$	$1.226_{\pm 0.028}$	$1.511_{\pm 0.043}$	$0.149_{\pm 0.003}$
	ScalaBL	$0.307_{\pm 0.036}$	$1.035_{\pm 0.033}$	$1.361_{\pm 0.050}$	$0.226_{\pm 0.003}$
	TFB	$0.644_{\pm 0.059}$	$3.317_{\pm 0.505}$	$4.679_{\pm 0.457}$	$0.150_{\pm 0.005}$
Brier (↓)	MLE	$0.235_{\pm 0.002}$	$0.823_{\pm 0.050}$	$1.165_{\pm 0.047}$	$0.075_{\pm 0.004}$
	MAP	$0.235_{\pm 0.005}$	$0.814_{\pm 0.037}$	$1.139_{\pm 0.037}$	$0.077_{\pm 0.003}$
	TempScale	$0.182_{\pm 0.004}$	$0.570_{\pm 0.018}$	$0.711_{\pm 0.011}$	$0.074_{\pm 0.003}$
	MCDropout	$0.229_{\pm 0.011}$	$0.796_{\pm 0.033}$	$1.152_{\pm 0.068}$	$0.077_{\pm 0.004}$
	Ensemble	$0.216_{\pm 0.012}$	$0.660_{\pm 0.010}$	$0.885_{\pm 0.013}$	$0.108_{\pm 0.002}$
	Laplace	$0.170_{\pm 0.003}$	$0.565_{\pm 0.006}$	$0.758_{\pm 0.045}$	$0.074_{\pm 0.002}$
	BLoB	$0.175_{\pm 0.006}$	$0.609_{\pm 0.011}$	$0.755_{\pm 0.025}$	$0.085_{\pm 0.002}$
	ScalaBL	$0.194_{\pm 0.021}$	$0.551_{\pm 0.020}$	$0.723_{\pm 0.013}$	$0.123_{\pm 0.001}$
	TFB	$0.215_{\pm 0.002}$	$0.722_{\pm 0.031}$	$0.998_{\pm 0.042}$	$0.081_{\pm 0.002}$

Table 22: In-Distribution Performance Comparison for Qwen3-VL-4B-Instruct

Metric	Method	slake	mmstar	MathVerse	srqa
ACC (↑)	MLE	$0.894_{\pm 0.005}$	$0.609_{\pm 0.021}$	$0.512_{\pm 0.026}$	$0.962_{\pm 0.003}$
	MAP	$0.892_{\pm 0.008}$	$0.611_{\pm 0.028}$	$0.492_{\pm 0.009}$	$0.963_{\pm 0.004}$
	TempScale	$0.894_{\pm 0.005}$	$0.608_{\pm 0.015}$	$0.516_{\pm 0.028}$	$0.962_{\pm 0.003}$
	MCDropout	$0.899_{\pm 0.004}$	$0.630_{\pm 0.023}$	$0.518_{\pm 0.017}$	$0.964_{\pm 0.003}$
	Ensemble	$0.899_{\pm 0.011}$	$0.621_{\pm 0.016}$	$0.548_{\pm 0.017}$	$0.948_{\pm 0.004}$
	Laplace	$0.894_{\pm 0.005}$	$0.580_{\pm 0.006}$	$0.532_{\pm 0.036}$	$0.962_{\pm 0.005}$
	BLoB	$0.900_{\pm 0.006}$	$0.608_{\pm 0.018}$	$0.521_{\pm 0.023}$	$0.953_{\pm 0.005}$
	ScalaBL	$0.891_{\pm 0.007}$	$0.637_{\pm 0.016}$	$0.548_{\pm 0.018}$	$0.947_{\pm 0.002}$
	TFB	$0.894_{\pm 0.006}$	$0.610_{\pm 0.007}$	$0.511_{\pm 0.024}$	$0.960_{\pm 0.002}$
ECE (↓)	MLE	$0.100_{\pm 0.004}$	$0.354_{\pm 0.026}$	$0.405_{\pm 0.030}$	$0.014_{\pm 0.002}$
	MAP	$0.103_{\pm 0.008}$	$0.353_{\pm 0.031}$	$0.442_{\pm 0.009}$	$0.014_{\pm 0.003}$
	TempScale	$0.042_{\pm 0.010}$	$0.084_{\pm 0.022}$	$0.101_{\pm 0.014}$	$0.012_{\pm 0.002}$
	MCDropout	$0.095_{\pm 0.004}$	$0.336_{\pm 0.013}$	$0.405_{\pm 0.023}$	$0.013_{\pm 0.003}$
	Ensemble	$0.084_{\pm 0.008}$	$0.240_{\pm 0.010}$	$0.224_{\pm 0.023}$	$0.014_{\pm 0.005}$
	Laplace	$0.086_{\pm 0.012}$	$0.131_{\pm 0.021}$	$0.109_{\pm 0.014}$	$0.016_{\pm 0.006}$
	BLoB	$0.032_{\pm 0.008}$	$0.138_{\pm 0.018}$	$0.134_{\pm 0.024}$	$0.016_{\pm 0.003}$
	ScalaBL	$0.035_{\pm 0.008}$	$0.125_{\pm 0.018}$	$0.120_{\pm 0.022}$	$0.031_{\pm 0.004}$
	TFB	$0.085_{\pm 0.004}$	$0.280_{\pm 0.016}$	$0.287_{\pm 0.028}$	$0.019_{\pm 0.003}$
NLL (↓)	MLE	$0.866_{\pm 0.047}$	$4.183_{\pm 0.424}$	$3.352_{\pm 0.296}$	$0.099_{\pm 0.001}$
	MAP	$0.962_{\pm 0.065}$	$4.332_{\pm 0.321}$	$3.933_{\pm 0.382}$	$0.095_{\pm 0.006}$
	TempScale	$0.260_{\pm 0.008}$	$1.010_{\pm 0.007}$	$1.097_{\pm 0.027}$	$0.096_{\pm 0.001}$
	MCDropout	$0.856_{\pm 0.147}$	$4.079_{\pm 0.225}$	$3.592_{\pm 0.172}$	$0.093_{\pm 0.003}$
	Ensemble	$0.652_{\pm 0.032}$	$2.186_{\pm 0.152}$	$1.631_{\pm 0.103}$	$0.131_{\pm 0.004}$
	Laplace	$0.285_{\pm 0.010}$	$1.045_{\pm 0.020}$	$1.084_{\pm 0.032}$	$0.097_{\pm 0.002}$
	BLoB	$0.240_{\pm 0.007}$	$1.126_{\pm 0.035}$	$1.141_{\pm 0.054}$	$0.110_{\pm 0.005}$
	ScalaBL	$0.258_{\pm 0.010}$	$0.984_{\pm 0.015}$	$1.015_{\pm 0.039}$	$0.147_{\pm 0.005}$
	TFB	$0.551_{\pm 0.028}$	$2.787_{\pm 0.289}$	$2.078_{\pm 0.232}$	$0.107_{\pm 0.006}$
Brier (↓)	MLE	$0.197_{\pm 0.010}$	$0.726_{\pm 0.037}$	$0.861_{\pm 0.038}$	$0.055_{\pm 0.003}$
	MAP	$0.201_{\pm 0.016}$	$0.722_{\pm 0.054}$	$0.911_{\pm 0.010}$	$0.054_{\pm 0.004}$
	TempScale	$0.156_{\pm 0.004}$	$0.534_{\pm 0.005}$	$0.592_{\pm 0.015}$	$0.054_{\pm 0.002}$
	MCDropout	$0.189_{\pm 0.007}$	$0.692_{\pm 0.034}$	$0.855_{\pm 0.044}$	$0.052_{\pm 0.003}$
	Ensemble	$0.175_{\pm 0.007}$	$0.601_{\pm 0.009}$	$0.661_{\pm 0.027}$	$0.074_{\pm 0.004}$
	Laplace	$0.169_{\pm 0.005}$	$0.560_{\pm 0.010}$	$0.596_{\pm 0.016}$	$0.055_{\pm 0.002}$
	BLoB	$0.149_{\pm 0.004}$	$0.539_{\pm 0.015}$	$0.602_{\pm 0.023}$	$0.063_{\pm 0.003}$
	ScalaBL	$0.161_{\pm 0.004}$	$0.500_{\pm 0.009}$	$0.557_{\pm 0.018}$	$0.080_{\pm 0.002}$
	TFB	$0.181_{\pm 0.006}$	$0.646_{\pm 0.029}$	$0.726_{\pm 0.029}$	$0.059_{\pm 0.002}$

Table 23: In-Distribution Performance Comparison for Qwen3-VL-8B-Instruct

Metric	Method	slake	mmstar	MathVerse	srqa
ACC (↑)	MLE	$0.892_{\pm 0.008}$	$0.656_{\pm 0.020}$	$0.513_{\pm 0.031}$	$0.966_{\pm 0.003}$
	MAP	$0.901_{\pm 0.006}$	$0.632_{\pm 0.035}$	$0.527_{\pm 0.031}$	$0.964_{\pm 0.005}$
	TempScale	$0.892_{\pm 0.008}$	$0.653_{\pm 0.025}$	$0.511_{\pm 0.030}$	$0.965_{\pm 0.002}$
	MCDropout	$0.887_{\pm 0.011}$	$0.644_{\pm 0.011}$	$0.532_{\pm 0.016}$	$0.961_{\pm 0.000}$
	Ensemble	$0.893_{\pm 0.005}$	$0.653_{\pm 0.016}$	$0.571_{\pm 0.018}$	$0.952_{\pm 0.003}$
	Laplace	$0.891_{\pm 0.009}$	$0.610_{\pm 0.025}$	$0.512_{\pm 0.025}$	$0.965_{\pm 0.002}$
	BLoB	$0.886_{\pm 0.008}$	$0.638_{\pm 0.017}$	$0.569_{\pm 0.025}$	$0.962_{\pm 0.004}$
	ScalaBL	$0.877_{\pm 0.005}$	$0.655_{\pm 0.019}$	$0.584_{\pm 0.018}$	$0.948_{\pm 0.002}$
	TFB	$0.885_{\pm 0.013}$	$0.660_{\pm 0.019}$	$0.503_{\pm 0.035}$	$0.967_{\pm 0.001}$
ECE (↓)	MLE	$0.100_{\pm 0.007}$	$0.316_{\pm 0.022}$	$0.439_{\pm 0.027}$	$0.016_{\pm 0.001}$
	MAP	$0.093_{\pm 0.011}$	$0.337_{\pm 0.034}$	$0.427_{\pm 0.027}$	$0.015_{\pm 0.004}$
	TempScale	$0.032_{\pm 0.007}$	$0.085_{\pm 0.011}$	$0.097_{\pm 0.014}$	$0.014_{\pm 0.004}$
	MCDropout	$0.106_{\pm 0.009}$	$0.325_{\pm 0.014}$	$0.407_{\pm 0.016}$	$0.016_{\pm 0.001}$
	Ensemble	$0.082_{\pm 0.003}$	$0.210_{\pm 0.012}$	$0.222_{\pm 0.011}$	$0.014_{\pm 0.002}$
	Laplace	$0.109_{\pm 0.020}$	$0.161_{\pm 0.024}$	$0.069_{\pm 0.032}$	$0.015_{\pm 0.003}$
	BLoB	$0.049_{\pm 0.006}$	$0.175_{\pm 0.015}$	$0.171_{\pm 0.019}$	$0.011_{\pm 0.002}$
	ScalaBL	$0.037_{\pm 0.004}$	$0.125_{\pm 0.016}$	$0.153_{\pm 0.018}$	$0.026_{\pm 0.003}$
	TFB	$0.081_{\pm 0.007}$	$0.264_{\pm 0.022}$	$0.364_{\pm 0.029}$	$0.018_{\pm 0.006}$
NLL (↓)	MLE	$0.980_{\pm 0.083}$	$4.291_{\pm 0.395}$	$4.757_{\pm 0.529}$	$0.092_{\pm 0.004}$
	MAP	$0.954_{\pm 0.162}$	$4.447_{\pm 0.395}$	$4.417_{\pm 0.366}$	$0.096_{\pm 0.004}$
	TempScale	$0.270_{\pm 0.007}$	$0.962_{\pm 0.021}$	$1.136_{\pm 0.061}$	$0.090_{\pm 0.004}$
	MCDropout	$1.061_{\pm 0.081}$	$4.175_{\pm 0.342}$	$4.265_{\pm 0.039}$	$0.103_{\pm 0.000}$
	Ensemble	$0.716_{\pm 0.052}$	$1.707_{\pm 0.032}$	$1.849_{\pm 0.049}$	$0.132_{\pm 0.005}$
	Laplace	$0.320_{\pm 0.014}$	$1.027_{\pm 0.004}$	$1.107_{\pm 0.040}$	$0.093_{\pm 0.005}$
	BLoB	$0.276_{\pm 0.009}$	$1.246_{\pm 0.060}$	$1.218_{\pm 0.058}$	$0.102_{\pm 0.003}$
	ScalaBL	$0.265_{\pm 0.008}$	$0.964_{\pm 0.025}$	$1.072_{\pm 0.042}$	$0.141_{\pm 0.002}$
	TFB	$0.582_{\pm 0.054}$	$3.146_{\pm 0.343}$	$3.180_{\pm 0.410}$	$0.091_{\pm 0.004}$
Brier (↓)	MLE	$0.202_{\pm 0.015}$	$0.649_{\pm 0.042}$	$0.907_{\pm 0.050}$	$0.050_{\pm 0.003}$
	MAP	$0.184_{\pm 0.016}$	$0.680_{\pm 0.060}$	$0.871_{\pm 0.054}$	$0.053_{\pm 0.003}$
	TempScale	$0.161_{\pm 0.007}$	$0.502_{\pm 0.011}$	$0.615_{\pm 0.035}$	$0.050_{\pm 0.003}$
	MCDropout	$0.210_{\pm 0.019}$	$0.665_{\pm 0.028}$	$0.856_{\pm 0.024}$	$0.058_{\pm 0.000}$
	Ensemble	$0.175_{\pm 0.005}$	$0.533_{\pm 0.015}$	$0.639_{\pm 0.006}$	$0.073_{\pm 0.004}$
	Laplace	$0.190_{\pm 0.010}$	$0.552_{\pm 0.002}$	$0.607_{\pm 0.017}$	$0.051_{\pm 0.002}$
	BLoB	$0.161_{\pm 0.004}$	$0.524_{\pm 0.018}$	$0.598_{\pm 0.017}$	$0.057_{\pm 0.002}$
	ScalaBL	$0.164_{\pm 0.003}$	$0.484_{\pm 0.018}$	$0.563_{\pm 0.018}$	$0.077_{\pm 0.001}$
	TFB	$0.180_{\pm 0.014}$	$0.597_{\pm 0.029}$	$0.811_{\pm 0.048}$	$0.052_{\pm 0.002}$

Table 24: OOD Performance Comparison for Qwen3-0.6B

Metric	Method	Train on obqa					
		In-dist	OOD (test set)				
			MMLU-Bio	MMLU-CS	MMLU-Chem	MMLU-Math	MMLU-Physics
ACC (↑)	MLE	$0.695_{\pm 0.004}$	$0.487_{\pm 0.037}$	$0.423_{\pm 0.025}$	$0.460_{\pm 0.026}$	$0.348_{\pm 0.017}$	$0.272_{\pm 0.035}$
	MAP	$0.700_{\pm 0.008}$	$0.505_{\pm 0.027}$	$0.420_{\pm 0.026}$	$0.443_{\pm 0.048}$	$0.351_{\pm 0.035}$	$0.288_{\pm 0.039}$
	TempScale	$0.695_{\pm 0.004}$	$0.487_{\pm 0.037}$	$0.423_{\pm 0.025}$	$0.460_{\pm 0.026}$	$0.348_{\pm 0.017}$	$0.272_{\pm 0.035}$
	MCDropout	$0.700_{\pm 0.009}$	$0.513_{\pm 0.030}$	$0.412_{\pm 0.000}$	$0.460_{\pm 0.046}$	$0.356_{\pm 0.046}$	$0.280_{\pm 0.039}$
	Ensemble	$0.712_{\pm 0.008}$	$0.526_{\pm 0.018}$	$0.436_{\pm 0.020}$	$0.470_{\pm 0.023}$	$0.384_{\pm 0.014}$	$0.275_{\pm 0.026}$
	Laplace	$0.696_{\pm 0.003}$	$0.492_{\pm 0.041}$	$0.412_{\pm 0.015}$	$0.460_{\pm 0.023}$	$0.364_{\pm 0.008}$	$0.293_{\pm 0.033}$
	BLoB	$0.702_{\pm 0.010}$	$0.503_{\pm 0.030}$	$0.379_{\pm 0.040}$	$0.470_{\pm 0.020}$	$0.374_{\pm 0.008}$	$0.315_{\pm 0.019}$
	ScalaBL	$0.695_{\pm 0.008}$	$0.508_{\pm 0.017}$	$0.384_{\pm 0.018}$	$0.460_{\pm 0.017}$	$0.414_{\pm 0.012}$	$0.263_{\pm 0.033}$
	TFB	$0.691_{\pm 0.012}$	$0.490_{\pm 0.043}$	$0.392_{\pm 0.000}$	$0.460_{\pm 0.032}$	$0.371_{\pm 0.017}$	$0.295_{\pm 0.017}$
ECE (↓)	MLE	$0.209_{\pm 0.007}$	$0.337_{\pm 0.042}$	$0.276_{\pm 0.060}$	$0.265_{\pm 0.002}$	$0.271_{\pm 0.033}$	$0.379_{\pm 0.028}$
	MAP	$0.205_{\pm 0.004}$	$0.334_{\pm 0.013}$	$0.272_{\pm 0.044}$	$0.290_{\pm 0.028}$	$0.285_{\pm 0.043}$	$0.383_{\pm 0.022}$
	TempScale	$0.056_{\pm 0.016}$	$0.144_{\pm 0.028}$	$0.097_{\pm 0.027}$	$0.152_{\pm 0.040}$	$0.122_{\pm 0.048}$	$0.193_{\pm 0.036}$
	MCDropout	$0.197_{\pm 0.007}$	$0.305_{\pm 0.017}$	$0.273_{\pm 0.021}$	$0.257_{\pm 0.041}$	$0.269_{\pm 0.043}$	$0.364_{\pm 0.047}$
	Ensemble	$0.175_{\pm 0.008}$	$0.263_{\pm 0.009}$	$0.237_{\pm 0.018}$	$0.227_{\pm 0.022}$	$0.221_{\pm 0.018}$	$0.352_{\pm 0.014}$
	Laplace	$0.122_{\pm 0.010}$	$0.206_{\pm 0.045}$	$0.205_{\pm 0.031}$	$0.184_{\pm 0.018}$	$0.228_{\pm 0.033}$	$0.287_{\pm 0.029}$
	BLoB	$0.052_{\pm 0.009}$	$0.126_{\pm 0.022}$	$0.150_{\pm 0.040}$	$0.118_{\pm 0.021}$	$0.129_{\pm 0.039}$	$0.161_{\pm 0.018}$
	ScalaBL	$0.046_{\pm 0.011}$	$0.131_{\pm 0.024}$	$0.130_{\pm 0.008}$	$0.139_{\pm 0.034}$	$0.104_{\pm 0.031}$	$0.234_{\pm 0.027}$
	TFB	$0.110_{\pm 0.008}$	$0.212_{\pm 0.019}$	$0.210_{\pm 0.045}$	$0.180_{\pm 0.019}$	$0.193_{\pm 0.022}$	$0.263_{\pm 0.013}$
NLL (↓)	MLE	$1.213_{\pm 0.045}$	$1.782_{\pm 0.108}$	$1.556_{\pm 0.173}$	$1.650_{\pm 0.077}$	$1.629_{\pm 0.084}$	$1.979_{\pm 0.049}$
	MAP	$1.218_{\pm 0.044}$	$1.778_{\pm 0.156}$	$1.555_{\pm 0.099}$	$1.677_{\pm 0.117}$	$1.647_{\pm 0.066}$	$2.019_{\pm 0.031}$
	TempScale	$0.782_{\pm 0.012}$	$1.146_{\pm 0.039}$	$1.221_{\pm 0.034}$	$1.212_{\pm 0.034}$	$1.323_{\pm 0.022}$	$1.407_{\pm 0.021}$
	MCDropout	$1.155_{\pm 0.039}$	$1.644_{\pm 0.134}$	$1.542_{\pm 0.124}$	$1.582_{\pm 0.102}$	$1.627_{\pm 0.050}$	$1.944_{\pm 0.031}$
	Ensemble	$1.063_{\pm 0.010}$	$1.537_{\pm 0.044}$	$1.439_{\pm 0.041}$	$1.441_{\pm 0.040}$	$1.529_{\pm 0.016}$	$1.785_{\pm 0.065}$
	Laplace	$0.925_{\pm 0.031}$	$1.346_{\pm 0.074}$	$1.369_{\pm 0.112}$	$1.404_{\pm 0.059}$	$1.492_{\pm 0.052}$	$1.689_{\pm 0.047}$
	BLoB	$0.741_{\pm 0.009}$	$1.136_{\pm 0.011}$	$1.270_{\pm 0.037}$	$1.126_{\pm 0.026}$	$1.321_{\pm 0.016}$	$1.376_{\pm 0.023}$
	ScalaBL	$0.785_{\pm 0.010}$	$1.120_{\pm 0.027}$	$1.294_{\pm 0.018}$	$1.149_{\pm 0.025}$	$1.292_{\pm 0.012}$	$1.417_{\pm 0.023}$
	TFB	$0.909_{\pm 0.042}$	$1.326_{\pm 0.067}$	$1.361_{\pm 0.123}$	$1.415_{\pm 0.057}$	$1.433_{\pm 0.045}$	$1.636_{\pm 0.026}$
Brier (↓)	MLE	$0.486_{\pm 0.004}$	$0.762_{\pm 0.043}$	$0.780_{\pm 0.049}$	$0.733_{\pm 0.026}$	$0.824_{\pm 0.032}$	$0.940_{\pm 0.020}$
	MAP	$0.481_{\pm 0.005}$	$0.739_{\pm 0.041}$	$0.784_{\pm 0.036}$	$0.743_{\pm 0.054}$	$0.830_{\pm 0.025}$	$0.945_{\pm 0.020}$
	TempScale	$0.411_{\pm 0.005}$	$0.627_{\pm 0.018}$	$0.670_{\pm 0.017}$	$0.650_{\pm 0.020}$	$0.723_{\pm 0.014}$	$0.766_{\pm 0.012}$
	MCDropout	$0.474_{\pm 0.009}$	$0.713_{\pm 0.038}$	$0.784_{\pm 0.024}$	$0.725_{\pm 0.048}$	$0.828_{\pm 0.021}$	$0.923_{\pm 0.028}$
	Ensemble	$0.447_{\pm 0.007}$	$0.695_{\pm 0.022}$	$0.766_{\pm 0.019}$	$0.683_{\pm 0.014}$	$0.783_{\pm 0.008}$	$0.883_{\pm 0.025}$
	Laplace	$0.430_{\pm 0.009}$	$0.659_{\pm 0.034}$	$0.721_{\pm 0.033}$	$0.680_{\pm 0.023}$	$0.782_{\pm 0.024}$	$0.842_{\pm 0.020}$
	BLoB	$0.395_{\pm 0.004}$	$0.618_{\pm 0.007}$	$0.692_{\pm 0.024}$	$0.612_{\pm 0.017}$	$0.716_{\pm 0.010}$	$0.733_{\pm 0.009}$
	ScalaBL	$0.423_{\pm 0.006}$	$0.607_{\pm 0.013}$	$0.707_{\pm 0.007}$	$0.624_{\pm 0.011}$	$0.707_{\pm 0.007}$	$0.763_{\pm 0.010}$
	TFB	$0.439_{\pm 0.013}$	$0.663_{\pm 0.030}$	$0.729_{\pm 0.042}$	$0.684_{\pm 0.007}$	$0.766_{\pm 0.019}$	$0.819_{\pm 0.012}$

Table 25: OOD Performance Comparison for Qwen3-1.7B

Metric	Method	Train on obqa					
		In-dist	OOD (test set)				
			MMLU-Bio	MMLU-CS	MMLU-Chem	MMLU-Math	MMLU-Physics
ACC (↑)	MLE	$0.819_{\pm 0.012}$	$0.564_{\pm 0.010}$	$0.505_{\pm 0.012}$	$0.527_{\pm 0.017}$	$0.404_{\pm 0.030}$	$0.453_{\pm 0.048}$
	MAP	$0.818_{\pm 0.010}$	$0.582_{\pm 0.022}$	$0.508_{\pm 0.015}$	$0.533_{\pm 0.029}$	$0.394_{\pm 0.030}$	$0.447_{\pm 0.036}$
	TempScale	$0.819_{\pm 0.012}$	$0.564_{\pm 0.010}$	$0.505_{\pm 0.012}$	$0.527_{\pm 0.017}$	$0.404_{\pm 0.030}$	$0.453_{\pm 0.048}$
	MCDropout	$0.818_{\pm 0.009}$	$0.574_{\pm 0.021}$	$0.490_{\pm 0.018}$	$0.540_{\pm 0.023}$	$0.386_{\pm 0.021}$	$0.472_{\pm 0.022}$
	Ensemble	$0.815_{\pm 0.007}$	$0.566_{\pm 0.018}$	$0.505_{\pm 0.012}$	$0.537_{\pm 0.033}$	$0.364_{\pm 0.034}$	$0.470_{\pm 0.036}$
	Laplace	$0.823_{\pm 0.011}$	$0.561_{\pm 0.008}$	$0.495_{\pm 0.025}$	$0.537_{\pm 0.013}$	$0.399_{\pm 0.027}$	$0.450_{\pm 0.032}$
	BLoB	$0.816_{\pm 0.007}$	$0.597_{\pm 0.013}$	$0.518_{\pm 0.013}$	$0.587_{\pm 0.022}$	$0.386_{\pm 0.038}$	$0.395_{\pm 0.033}$
	ScalaBL	$0.802_{\pm 0.008}$	$0.622_{\pm 0.026}$	$0.521_{\pm 0.006}$	$0.547_{\pm 0.019}$	$0.396_{\pm 0.021}$	$0.453_{\pm 0.022}$
	TFB	$0.816_{\pm 0.010}$	$0.554_{\pm 0.010}$	$0.482_{\pm 0.020}$	$0.543_{\pm 0.013}$	$0.364_{\pm 0.036}$	$0.405_{\pm 0.010}$
ECE (↓)	MLE	$0.105_{\pm 0.009}$	$0.280_{\pm 0.031}$	$0.229_{\pm 0.018}$	$0.210_{\pm 0.012}$	$0.212_{\pm 0.021}$	$0.256_{\pm 0.024}$
	MAP	$0.108_{\pm 0.010}$	$0.261_{\pm 0.024}$	$0.233_{\pm 0.022}$	$0.220_{\pm 0.017}$	$0.215_{\pm 0.041}$	$0.211_{\pm 0.019}$
	TempScale	$0.042_{\pm 0.007}$	$0.156_{\pm 0.016}$	$0.119_{\pm 0.008}$	$0.127_{\pm 0.032}$	$0.109_{\pm 0.052}$	$0.183_{\pm 0.047}$
	MCDropout	$0.104_{\pm 0.010}$	$0.266_{\pm 0.025}$	$0.240_{\pm 0.019}$	$0.184_{\pm 0.013}$	$0.189_{\pm 0.014}$	$0.234_{\pm 0.051}$
	Ensemble	$0.100_{\pm 0.004}$	$0.263_{\pm 0.017}$	$0.244_{\pm 0.016}$	$0.188_{\pm 0.042}$	$0.232_{\pm 0.031}$	$0.236_{\pm 0.029}$
	Laplace	$0.049_{\pm 0.009}$	$0.189_{\pm 0.015}$	$0.163_{\pm 0.019}$	$0.154_{\pm 0.028}$	$0.160_{\pm 0.021}$	$0.203_{\pm 0.023}$
	BLoB	$0.046_{\pm 0.010}$	$0.163_{\pm 0.024}$	$0.158_{\pm 0.014}$	$0.136_{\pm 0.028}$	$0.173_{\pm 0.031}$	$0.204_{\pm 0.033}$
	ScalaBL	$0.044_{\pm 0.007}$	$0.113_{\pm 0.033}$	$0.130_{\pm 0.025}$	$0.132_{\pm 0.026}$	$0.130_{\pm 0.017}$	$0.211_{\pm 0.021}$
	TFB	$0.047_{\pm 0.009}$	$0.191_{\pm 0.031}$	$0.132_{\pm 0.019}$	$0.134_{\pm 0.025}$	$0.148_{\pm 0.034}$	$0.183_{\pm 0.032}$
NLL (↓)	MLE	$0.637_{\pm 0.017}$	$1.395_{\pm 0.132}$	$1.328_{\pm 0.018}$	$1.143_{\pm 0.037}$	$1.411_{\pm 0.039}$	$1.645_{\pm 0.015}$
	MAP	$0.647_{\pm 0.032}$	$1.392_{\pm 0.094}$	$1.317_{\pm 0.019}$	$1.148_{\pm 0.041}$	$1.437_{\pm 0.056}$	$1.610_{\pm 0.041}$
	TempScale	$0.506_{\pm 0.007}$	$0.994_{\pm 0.022}$	$1.115_{\pm 0.012}$	$1.002_{\pm 0.014}$	$1.293_{\pm 0.016}$	$1.314_{\pm 0.020}$
	MCDropout	$0.628_{\pm 0.024}$	$1.364_{\pm 0.099}$	$1.307_{\pm 0.039}$	$1.123_{\pm 0.046}$	$1.420_{\pm 0.035}$	$1.607_{\pm 0.047}$
	Ensemble	$0.616_{\pm 0.007}$	$1.371_{\pm 0.052}$	$1.304_{\pm 0.040}$	$1.114_{\pm 0.055}$	$1.422_{\pm 0.036}$	$1.571_{\pm 0.030}$
	Laplace	$0.516_{\pm 0.014}$	$1.108_{\pm 0.056}$	$1.188_{\pm 0.008}$	$1.058_{\pm 0.013}$	$1.338_{\pm 0.016}$	$1.438_{\pm 0.023}$
	BLoB	$0.506_{\pm 0.008}$	$0.990_{\pm 0.024}$	$1.130_{\pm 0.021}$	$0.962_{\pm 0.011}$	$1.344_{\pm 0.013}$	$1.316_{\pm 0.031}$
	ScalaBL	$0.541_{\pm 0.009}$	$0.931_{\pm 0.033}$	$1.074_{\pm 0.015}$	$0.991_{\pm 0.033}$	$1.311_{\pm 0.014}$	$1.295_{\pm 0.037}$
	TFB	$0.535_{\pm 0.017}$	$1.150_{\pm 0.059}$	$1.191_{\pm 0.023}$	$1.059_{\pm 0.021}$	$1.352_{\pm 0.050}$	$1.466_{\pm 0.030}$
Brier (↓)	MLE	$0.286_{\pm 0.007}$	$0.638_{\pm 0.020}$	$0.687_{\pm 0.009}$	$0.596_{\pm 0.021}$	$0.763_{\pm 0.024}$	$0.758_{\pm 0.018}$
	MAP	$0.291_{\pm 0.014}$	$0.632_{\pm 0.012}$	$0.682_{\pm 0.020}$	$0.595_{\pm 0.027}$	$0.774_{\pm 0.028}$	$0.743_{\pm 0.020}$
	TempScale	$0.261_{\pm 0.004}$	$0.529_{\pm 0.005}$	$0.609_{\pm 0.009}$	$0.537_{\pm 0.011}$	$0.707_{\pm 0.010}$	$0.697_{\pm 0.013}$
	MCDropout	$0.282_{\pm 0.010}$	$0.628_{\pm 0.026}$	$0.682_{\pm 0.013}$	$0.584_{\pm 0.026}$	$0.767_{\pm 0.020}$	$0.744_{\pm 0.036}$
	Ensemble	$0.279_{\pm 0.005}$	$0.625_{\pm 0.004}$	$0.678_{\pm 0.016}$	$0.571_{\pm 0.015}$	$0.776_{\pm 0.016}$	$0.729_{\pm 0.014}$
	Laplace	$0.261_{\pm 0.004}$	$0.547_{\pm 0.014}$	$0.640_{\pm 0.005}$	$0.563_{\pm 0.009}$	$0.732_{\pm 0.012}$	$0.724_{\pm 0.015}$
	BLoB	$0.264_{\pm 0.005}$	$0.520_{\pm 0.007}$	$0.603_{\pm 0.004}$	$0.516_{\pm 0.015}$	$0.729_{\pm 0.007}$	$0.695_{\pm 0.012}$
	ScalaBL	$0.284_{\pm 0.004}$	$0.489_{\pm 0.017}$	$0.588_{\pm 0.012}$	$0.531_{\pm 0.012}$	$0.713_{\pm 0.003}$	$0.703_{\pm 0.016}$
	TFB	$0.269_{\pm 0.005}$	$0.567_{\pm 0.013}$	$0.634_{\pm 0.019}$	$0.566_{\pm 0.009}$	$0.743_{\pm 0.026}$	$0.730_{\pm 0.013}$

Table 26: OOD Performance Comparison for Qwen3-4B

Metric	Method	Train on obqa					
		In-dist	OOD (test set)				
			MMLU-Bio	MMLU-CS	MMLU-Chem	MMLU-Math	MMLU-Physics
ACC (↑)	MLE	$0.912_{\pm 0.014}$	$0.842_{\pm 0.026}$	$0.647_{\pm 0.015}$	$0.663_{\pm 0.030}$	$0.515_{\pm 0.036}$	$0.597_{\pm 0.017}$
	MAP	$0.918_{\pm 0.006}$	$0.865_{\pm 0.015}$	$0.637_{\pm 0.023}$	$0.640_{\pm 0.029}$	$0.503_{\pm 0.036}$	$0.610_{\pm 0.008}$
	TempScale	$0.912_{\pm 0.014}$	$0.842_{\pm 0.026}$	$0.647_{\pm 0.015}$	$0.663_{\pm 0.030}$	$0.515_{\pm 0.036}$	$0.597_{\pm 0.017}$
	MCDropout	$0.916_{\pm 0.007}$	$0.862_{\pm 0.018}$	$0.634_{\pm 0.013}$	$0.653_{\pm 0.036}$	$0.513_{\pm 0.022}$	$0.620_{\pm 0.022}$
	Ensemble	$0.914_{\pm 0.001}$	$0.857_{\pm 0.008}$	$0.629_{\pm 0.008}$	$0.680_{\pm 0.019}$	$0.497_{\pm 0.028}$	$0.598_{\pm 0.013}$
	Laplace	$0.916_{\pm 0.009}$	$0.842_{\pm 0.021}$	$0.626_{\pm 0.010}$	$0.657_{\pm 0.017}$	$0.513_{\pm 0.017}$	$0.610_{\pm 0.014}$
	BLoB	$0.913_{\pm 0.005}$	$0.865_{\pm 0.021}$	$0.642_{\pm 0.013}$	$0.627_{\pm 0.033}$	$0.510_{\pm 0.037}$	$0.577_{\pm 0.015}$
	ScalaBL	$0.907_{\pm 0.005}$	$0.827_{\pm 0.025}$	$0.657_{\pm 0.015}$	$0.680_{\pm 0.011}$	$0.513_{\pm 0.013}$	$0.618_{\pm 0.015}$
	TFB	$0.910_{\pm 0.018}$	$0.837_{\pm 0.028}$	$0.644_{\pm 0.013}$	$0.677_{\pm 0.037}$	$0.503_{\pm 0.050}$	$0.595_{\pm 0.017}$
ECE (↓)	MLE	$0.073_{\pm 0.016}$	$0.122_{\pm 0.026}$	$0.275_{\pm 0.008}$	$0.241_{\pm 0.024}$	$0.360_{\pm 0.022}$	$0.285_{\pm 0.007}$
	MAP	$0.065_{\pm 0.009}$	$0.105_{\pm 0.011}$	$0.295_{\pm 0.011}$	$0.235_{\pm 0.026}$	$0.361_{\pm 0.038}$	$0.273_{\pm 0.010}$
	TempScale	$0.039_{\pm 0.005}$	$0.081_{\pm 0.013}$	$0.112_{\pm 0.031}$	$0.110_{\pm 0.012}$	$0.155_{\pm 0.023}$	$0.124_{\pm 0.010}$
	MCDropout	$0.067_{\pm 0.005}$	$0.112_{\pm 0.018}$	$0.286_{\pm 0.009}$	$0.239_{\pm 0.019}$	$0.343_{\pm 0.021}$	$0.277_{\pm 0.019}$
	Ensemble	$0.062_{\pm 0.003}$	$0.095_{\pm 0.006}$	$0.256_{\pm 0.009}$	$0.190_{\pm 0.028}$	$0.341_{\pm 0.032}$	$0.276_{\pm 0.012}$
	Laplace	$0.087_{\pm 0.015}$	$0.104_{\pm 0.021}$	$0.103_{\pm 0.025}$	$0.128_{\pm 0.011}$	$0.156_{\pm 0.047}$	$0.108_{\pm 0.056}$
	BLoB	$0.034_{\pm 0.005}$	$0.083_{\pm 0.025}$	$0.164_{\pm 0.010}$	$0.148_{\pm 0.024}$	$0.173_{\pm 0.024}$	$0.171_{\pm 0.004}$
	ScalaBL	$0.028_{\pm 0.006}$	$0.066_{\pm 0.016}$	$0.137_{\pm 0.032}$	$0.114_{\pm 0.016}$	$0.138_{\pm 0.002}$	$0.124_{\pm 0.023}$
	TFB	$0.050_{\pm 0.014}$	$0.091_{\pm 0.017}$	$0.220_{\pm 0.032}$	$0.156_{\pm 0.021}$	$0.265_{\pm 0.045}$	$0.205_{\pm 0.013}$
NLL (↓)	MLE	$0.613_{\pm 0.131}$	$0.689_{\pm 0.111}$	$2.049_{\pm 0.290}$	$1.305_{\pm 0.106}$	$2.261_{\pm 0.319}$	$1.720_{\pm 0.180}$
	MAP	$0.568_{\pm 0.065}$	$0.614_{\pm 0.086}$	$2.005_{\pm 0.235}$	$1.281_{\pm 0.104}$	$2.189_{\pm 0.126}$	$1.679_{\pm 0.105}$
	TempScale	$0.292_{\pm 0.012}$	$0.397_{\pm 0.022}$	$0.907_{\pm 0.027}$	$0.760_{\pm 0.033}$	$1.111_{\pm 0.031}$	$0.893_{\pm 0.019}$
	MCDropout	$0.549_{\pm 0.014}$	$0.610_{\pm 0.050}$	$1.923_{\pm 0.073}$	$1.220_{\pm 0.129}$	$2.079_{\pm 0.145}$	$1.627_{\pm 0.075}$
	Ensemble	$0.463_{\pm 0.060}$	$0.532_{\pm 0.045}$	$1.616_{\pm 0.084}$	$1.082_{\pm 0.043}$	$1.829_{\pm 0.139}$	$1.418_{\pm 0.026}$
	Laplace	$0.312_{\pm 0.018}$	$0.435_{\pm 0.017}$	$0.936_{\pm 0.037}$	$0.819_{\pm 0.035}$	$1.116_{\pm 0.024}$	$0.985_{\pm 0.035}$
	BLoB	$0.294_{\pm 0.009}$	$0.406_{\pm 0.035}$	$1.069_{\pm 0.035}$	$0.836_{\pm 0.034}$	$1.148_{\pm 0.023}$	$1.021_{\pm 0.015}$
	ScalaBL	$0.285_{\pm 0.006}$	$0.399_{\pm 0.013}$	$0.944_{\pm 0.023}$	$0.801_{\pm 0.032}$	$1.078_{\pm 0.010}$	$0.934_{\pm 0.032}$
	TFB	$0.429_{\pm 0.061}$	$0.501_{\pm 0.054}$	$1.453_{\pm 0.112}$	$1.010_{\pm 0.099}$	$1.587_{\pm 0.109}$	$1.203_{\pm 0.039}$
Brier (↓)	MLE	$0.153_{\pm 0.025}$	$0.248_{\pm 0.038}$	$0.593_{\pm 0.006}$	$0.517_{\pm 0.043}$	$0.779_{\pm 0.036}$	$0.619_{\pm 0.006}$
	MAP	$0.146_{\pm 0.012}$	$0.222_{\pm 0.017}$	$0.602_{\pm 0.007}$	$0.529_{\pm 0.026}$	$0.776_{\pm 0.053}$	$0.619_{\pm 0.020}$
	TempScale	$0.138_{\pm 0.008}$	$0.210_{\pm 0.018}$	$0.470_{\pm 0.008}$	$0.407_{\pm 0.022}$	$0.602_{\pm 0.019}$	$0.488_{\pm 0.010}$
	MCDropout	$0.145_{\pm 0.009}$	$0.225_{\pm 0.024}$	$0.601_{\pm 0.014}$	$0.518_{\pm 0.035}$	$0.767_{\pm 0.026}$	$0.623_{\pm 0.014}$
	Ensemble	$0.143_{\pm 0.002}$	$0.221_{\pm 0.010}$	$0.565_{\pm 0.012}$	$0.466_{\pm 0.022}$	$0.743_{\pm 0.039}$	$0.609_{\pm 0.018}$
	Laplace	$0.155_{\pm 0.006}$	$0.240_{\pm 0.008}$	$0.486_{\pm 0.010}$	$0.422_{\pm 0.007}$	$0.615_{\pm 0.010}$	$0.518_{\pm 0.015}$
	BLoB	$0.135_{\pm 0.004}$	$0.210_{\pm 0.018}$	$0.506_{\pm 0.005}$	$0.444_{\pm 0.024}$	$0.613_{\pm 0.018}$	$0.535_{\pm 0.010}$
	ScalaBL	$0.139_{\pm 0.003}$	$0.221_{\pm 0.009}$	$0.473_{\pm 0.004}$	$0.407_{\pm 0.014}$	$0.602_{\pm 0.003}$	$0.499_{\pm 0.014}$
	TFB	$0.143_{\pm 0.017}$	$0.233_{\pm 0.020}$	$0.537_{\pm 0.016}$	$0.459_{\pm 0.039}$	$0.689_{\pm 0.047}$	$0.562_{\pm 0.022}$

Table 27: OOD Performance Comparison for Qwen3-8B

Metric	Method	Train on obqa					
		In-dist	OOD (test set)				
			MMLU-Bio	MMLU-CS	MMLU-Chem	MMLU-Math	MMLU-Physics
ACC (↑)	MLE	$0.924_{\pm 0.008}$	$0.834_{\pm 0.021}$	$0.678_{\pm 0.023}$	$0.713_{\pm 0.008}$	$0.581_{\pm 0.017}$	$0.573_{\pm 0.021}$
	MAP	$0.925_{\pm 0.003}$	$0.832_{\pm 0.031}$	$0.673_{\pm 0.023}$	$0.720_{\pm 0.029}$	$0.568_{\pm 0.010}$	$0.575_{\pm 0.025}$
	TempScale	$0.924_{\pm 0.008}$	$0.834_{\pm 0.021}$	$0.678_{\pm 0.023}$	$0.713_{\pm 0.008}$	$0.581_{\pm 0.017}$	$0.573_{\pm 0.021}$
	MCDropout	$0.924_{\pm 0.002}$	$0.849_{\pm 0.013}$	$0.675_{\pm 0.013}$	$0.713_{\pm 0.013}$	$0.598_{\pm 0.028}$	$0.557_{\pm 0.029}$
	Ensemble	$0.920_{\pm 0.004}$	$0.849_{\pm 0.015}$	$0.688_{\pm 0.018}$	$0.717_{\pm 0.013}$	$0.581_{\pm 0.024}$	$0.575_{\pm 0.030}$
	Laplace	$0.924_{\pm 0.004}$	$0.867_{\pm 0.013}$	$0.701_{\pm 0.021}$	$0.680_{\pm 0.014}$	$0.535_{\pm 0.025}$	$0.560_{\pm 0.021}$
	BLoB	$0.923_{\pm 0.006}$	$0.888_{\pm 0.012}$	$0.668_{\pm 0.013}$	$0.683_{\pm 0.007}$	$0.601_{\pm 0.031}$	$0.575_{\pm 0.013}$
	ScalaBL	$0.919_{\pm 0.004}$	$0.865_{\pm 0.015}$	$0.680_{\pm 0.019}$	$0.723_{\pm 0.027}$	$0.598_{\pm 0.010}$	$0.565_{\pm 0.006}$
	TFB	$0.923_{\pm 0.005}$	$0.849_{\pm 0.015}$	$0.665_{\pm 0.038}$	$0.693_{\pm 0.015}$	$0.604_{\pm 0.035}$	$0.573_{\pm 0.021}$
ECE (↓)	MLE	$0.056_{\pm 0.007}$	$0.116_{\pm 0.012}$	$0.224_{\pm 0.009}$	$0.182_{\pm 0.011}$	$0.256_{\pm 0.024}$	$0.310_{\pm 0.017}$
	MAP	$0.057_{\pm 0.001}$	$0.118_{\pm 0.026}$	$0.208_{\pm 0.016}$	$0.187_{\pm 0.023}$	$0.259_{\pm 0.011}$	$0.302_{\pm 0.021}$
	TempScale	$0.033_{\pm 0.012}$	$0.082_{\pm 0.011}$	$0.131_{\pm 0.011}$	$0.116_{\pm 0.027}$	$0.134_{\pm 0.002}$	$0.169_{\pm 0.027}$
	MCDropout	$0.060_{\pm 0.002}$	$0.122_{\pm 0.012}$	$0.216_{\pm 0.019}$	$0.178_{\pm 0.008}$	$0.261_{\pm 0.025}$	$0.319_{\pm 0.038}$
	Ensemble	$0.057_{\pm 0.007}$	$0.092_{\pm 0.012}$	$0.189_{\pm 0.025}$	$0.164_{\pm 0.029}$	$0.232_{\pm 0.047}$	$0.282_{\pm 0.043}$
	Laplace	$0.071_{\pm 0.015}$	$0.119_{\pm 0.031}$	$0.098_{\pm 0.014}$	$0.098_{\pm 0.007}$	$0.136_{\pm 0.023}$	$0.121_{\pm 0.056}$
	BLoB	$0.033_{\pm 0.007}$	$0.050_{\pm 0.004}$	$0.166_{\pm 0.020}$	$0.139_{\pm 0.005}$	$0.168_{\pm 0.015}$	$0.223_{\pm 0.016}$
	ScalaBL	$0.026_{\pm 0.004}$	$0.078_{\pm 0.012}$	$0.145_{\pm 0.011}$	$0.112_{\pm 0.012}$	$0.153_{\pm 0.012}$	$0.219_{\pm 0.009}$
	TFB	$0.041_{\pm 0.004}$	$0.080_{\pm 0.005}$	$0.193_{\pm 0.026}$	$0.151_{\pm 0.028}$	$0.179_{\pm 0.018}$	$0.238_{\pm 0.024}$
NLL (↓)	MLE	$0.350_{\pm 0.034}$	$0.558_{\pm 0.037}$	$1.230_{\pm 0.064}$	$1.063_{\pm 0.064}$	$1.638_{\pm 0.135}$	$1.692_{\pm 0.082}$
	MAP	$0.341_{\pm 0.026}$	$0.540_{\pm 0.035}$	$1.177_{\pm 0.044}$	$1.006_{\pm 0.015}$	$1.583_{\pm 0.103}$	$1.653_{\pm 0.034}$
	TempScale	$0.221_{\pm 0.013}$	$0.387_{\pm 0.011}$	$0.764_{\pm 0.021}$	$0.710_{\pm 0.039}$	$1.028_{\pm 0.028}$	$0.965_{\pm 0.046}$
	MCDropout	$0.361_{\pm 0.020}$	$0.566_{\pm 0.050}$	$1.189_{\pm 0.094}$	$1.078_{\pm 0.035}$	$1.655_{\pm 0.149}$	$1.747_{\pm 0.064}$
	Ensemble	$0.285_{\pm 0.022}$	$0.438_{\pm 0.079}$	$1.091_{\pm 0.138}$	$0.985_{\pm 0.094}$	$1.419_{\pm 0.226}$	$1.466_{\pm 0.202}$
	Laplace	$0.245_{\pm 0.015}$	$0.412_{\pm 0.046}$	$0.844_{\pm 0.067}$	$0.784_{\pm 0.045}$	$1.042_{\pm 0.069}$	$0.992_{\pm 0.084}$
	BLoB	$0.217_{\pm 0.013}$	$0.336_{\pm 0.029}$	$0.949_{\pm 0.057}$	$0.852_{\pm 0.045}$	$1.115_{\pm 0.026}$	$1.215_{\pm 0.056}$
	ScalaBL	$0.210_{\pm 0.004}$	$0.304_{\pm 0.010}$	$0.861_{\pm 0.013}$	$0.734_{\pm 0.023}$	$1.053_{\pm 0.005}$	$1.039_{\pm 0.012}$
	TFB	$0.268_{\pm 0.023}$	$0.422_{\pm 0.038}$	$0.984_{\pm 0.075}$	$0.921_{\pm 0.049}$	$1.321_{\pm 0.116}$	$1.291_{\pm 0.008}$
Brier (↓)	MLE	$0.127_{\pm 0.010}$	$0.246_{\pm 0.016}$	$0.507_{\pm 0.019}$	$0.440_{\pm 0.018}$	$0.639_{\pm 0.023}$	$0.689_{\pm 0.027}$
	MAP	$0.123_{\pm 0.006}$	$0.251_{\pm 0.029}$	$0.490_{\pm 0.020}$	$0.417_{\pm 0.021}$	$0.638_{\pm 0.022}$	$0.681_{\pm 0.019}$
	TempScale	$0.110_{\pm 0.006}$	$0.206_{\pm 0.008}$	$0.420_{\pm 0.012}$	$0.368_{\pm 0.016}$	$0.550_{\pm 0.014}$	$0.532_{\pm 0.024}$
	MCDropout	$0.129_{\pm 0.003}$	$0.250_{\pm 0.020}$	$0.497_{\pm 0.031}$	$0.439_{\pm 0.017}$	$0.647_{\pm 0.027}$	$0.707_{\pm 0.032}$
	Ensemble	$0.122_{\pm 0.006}$	$0.205_{\pm 0.022}$	$0.475_{\pm 0.009}$	$0.425_{\pm 0.015}$	$0.616_{\pm 0.033}$	$0.651_{\pm 0.046}$
	Laplace	$0.121_{\pm 0.005}$	$0.223_{\pm 0.012}$	$0.431_{\pm 0.032}$	$0.403_{\pm 0.025}$	$0.564_{\pm 0.044}$	$0.542_{\pm 0.069}$
	BLoB	$0.109_{\pm 0.006}$	$0.170_{\pm 0.014}$	$0.450_{\pm 0.018}$	$0.410_{\pm 0.017}$	$0.551_{\pm 0.013}$	$0.598_{\pm 0.012}$
	ScalaBL	$0.111_{\pm 0.003}$	$0.169_{\pm 0.005}$	$0.432_{\pm 0.005}$	$0.371_{\pm 0.011}$	$0.538_{\pm 0.012}$	$0.548_{\pm 0.005}$
	TFB	$0.116_{\pm 0.008}$	$0.217_{\pm 0.006}$	$0.468_{\pm 0.018}$	$0.421_{\pm 0.019}$	$0.593_{\pm 0.027}$	$0.604_{\pm 0.016}$

Table 28: OOD Performance Comparison for Qwen3-14B

Metric	Method	Train on obqa					
		In-dist	OOD (test set)				
			MMLU-Bio	MMLU-CS	MMLU-Chem	MMLU-Math	MMLU-Physics
ACC (↑)	MLE	$0.946_{\pm 0.005}$	$0.878_{\pm 0.012}$	$0.740_{\pm 0.018}$	$0.687_{\pm 0.013}$	$0.682_{\pm 0.024}$	$0.665_{\pm 0.013}$
	MAP	$0.946_{\pm 0.007}$	$0.878_{\pm 0.014}$	$0.747_{\pm 0.020}$	$0.693_{\pm 0.022}$	$0.689_{\pm 0.024}$	$0.680_{\pm 0.008}$
	TempScale	$0.946_{\pm 0.005}$	$0.878_{\pm 0.012}$	$0.740_{\pm 0.018}$	$0.687_{\pm 0.013}$	$0.682_{\pm 0.024}$	$0.665_{\pm 0.013}$
	MCDropout	$0.948_{\pm 0.002}$	$0.885_{\pm 0.010}$	$0.745_{\pm 0.024}$	$0.693_{\pm 0.011}$	$0.694_{\pm 0.019}$	$0.677_{\pm 0.022}$
	Ensemble	$0.957_{\pm 0.003}$	$0.857_{\pm 0.016}$	$0.711_{\pm 0.023}$	$0.680_{\pm 0.012}$	$0.667_{\pm 0.018}$	$0.650_{\pm 0.017}$
	Laplace	$0.947_{\pm 0.005}$	$0.898_{\pm 0.014}$	$0.732_{\pm 0.026}$	$0.667_{\pm 0.016}$	$0.677_{\pm 0.017}$	$0.680_{\pm 0.018}$
	BLoB	$0.952_{\pm 0.005}$	$0.883_{\pm 0.013}$	$0.750_{\pm 0.013}$	$0.660_{\pm 0.013}$	$0.682_{\pm 0.021}$	$0.665_{\pm 0.019}$
	ScalaBL	$0.952_{\pm 0.004}$	$0.890_{\pm 0.013}$	$0.765_{\pm 0.010}$	$0.693_{\pm 0.019}$	$0.669_{\pm 0.015}$	$0.668_{\pm 0.019}$
	TFB	$0.946_{\pm 0.007}$	$0.880_{\pm 0.005}$	$0.745_{\pm 0.034}$	$0.697_{\pm 0.017}$	$0.687_{\pm 0.030}$	$0.660_{\pm 0.014}$
ECE (↓)	MLE	$0.046_{\pm 0.003}$	$0.104_{\pm 0.007}$	$0.183_{\pm 0.016}$	$0.244_{\pm 0.014}$	$0.234_{\pm 0.025}$	$0.274_{\pm 0.011}$
	MAP	$0.044_{\pm 0.006}$	$0.104_{\pm 0.009}$	$0.188_{\pm 0.015}$	$0.238_{\pm 0.022}$	$0.247_{\pm 0.021}$	$0.272_{\pm 0.027}$
	TempScale	$0.031_{\pm 0.003}$	$0.067_{\pm 0.006}$	$0.086_{\pm 0.022}$	$0.137_{\pm 0.018}$	$0.119_{\pm 0.033}$	$0.138_{\pm 0.016}$
	MCDropout	$0.041_{\pm 0.004}$	$0.099_{\pm 0.007}$	$0.188_{\pm 0.017}$	$0.232_{\pm 0.016}$	$0.231_{\pm 0.015}$	$0.262_{\pm 0.025}$
	Ensemble	$0.023_{\pm 0.003}$	$0.080_{\pm 0.006}$	$0.111_{\pm 0.016}$	$0.164_{\pm 0.012}$	$0.104_{\pm 0.011}$	$0.182_{\pm 0.021}$
	Laplace	$0.152_{\pm 0.014}$	$0.159_{\pm 0.009}$	$0.137_{\pm 0.021}$	$0.136_{\pm 0.011}$	$0.148_{\pm 0.013}$	$0.139_{\pm 0.026}$
	BLoB	$0.025_{\pm 0.005}$	$0.076_{\pm 0.012}$	$0.154_{\pm 0.017}$	$0.225_{\pm 0.009}$	$0.179_{\pm 0.012}$	$0.219_{\pm 0.019}$
	ScalaBL	$0.018_{\pm 0.005}$	$0.056_{\pm 0.010}$	$0.114_{\pm 0.025}$	$0.141_{\pm 0.011}$	$0.133_{\pm 0.012}$	$0.197_{\pm 0.015}$
	TFB	$0.027_{\pm 0.003}$	$0.071_{\pm 0.006}$	$0.147_{\pm 0.012}$	$0.178_{\pm 0.021}$	$0.161_{\pm 0.023}$	$0.204_{\pm 0.017}$
NLL (↓)	MLE	$0.281_{\pm 0.031}$	$0.555_{\pm 0.019}$	$1.167_{\pm 0.028}$	$1.477_{\pm 0.071}$	$1.519_{\pm 0.124}$	$1.556_{\pm 0.176}$
	MAP	$0.283_{\pm 0.040}$	$0.593_{\pm 0.061}$	$1.195_{\pm 0.075}$	$1.503_{\pm 0.052}$	$1.565_{\pm 0.029}$	$1.586_{\pm 0.125}$
	TempScale	$0.158_{\pm 0.010}$	$0.297_{\pm 0.015}$	$0.626_{\pm 0.015}$	$0.725_{\pm 0.032}$	$0.832_{\pm 0.018}$	$0.750_{\pm 0.043}$
	MCDropout	$0.264_{\pm 0.025}$	$0.544_{\pm 0.049}$	$1.170_{\pm 0.053}$	$1.466_{\pm 0.025}$	$1.555_{\pm 0.067}$	$1.525_{\pm 0.159}$
	Ensemble	$0.131_{\pm 0.009}$	$0.261_{\pm 0.064}$	$0.701_{\pm 0.039}$	$0.736_{\pm 0.052}$	$0.827_{\pm 0.012}$	$0.866_{\pm 0.028}$
	Laplace	$0.276_{\pm 0.019}$	$0.386_{\pm 0.019}$	$0.736_{\pm 0.014}$	$0.736_{\pm 0.021}$	$0.863_{\pm 0.015}$	$0.764_{\pm 0.041}$
	BLoB	$0.159_{\pm 0.015}$	$0.306_{\pm 0.044}$	$0.840_{\pm 0.022}$	$1.082_{\pm 0.041}$	$1.131_{\pm 0.006}$	$1.108_{\pm 0.099}$
	ScalaBL	$0.134_{\pm 0.004}$	$0.236_{\pm 0.015}$	$0.692_{\pm 0.014}$	$0.753_{\pm 0.043}$	$0.909_{\pm 0.013}$	$0.885_{\pm 0.022}$
	TFB	$0.188_{\pm 0.015}$	$0.391_{\pm 0.037}$	$0.839_{\pm 0.086}$	$1.009_{\pm 0.047}$	$1.151_{\pm 0.105}$	$1.121_{\pm 0.138}$
Brier (↓)	MLE	$0.091_{\pm 0.009}$	$0.203_{\pm 0.014}$	$0.404_{\pm 0.015}$	$0.508_{\pm 0.021}$	$0.522_{\pm 0.034}$	$0.557_{\pm 0.027}$
	MAP	$0.089_{\pm 0.007}$	$0.210_{\pm 0.015}$	$0.398_{\pm 0.017}$	$0.496_{\pm 0.029}$	$0.524_{\pm 0.025}$	$0.551_{\pm 0.027}$
	TempScale	$0.076_{\pm 0.006}$	$0.161_{\pm 0.009}$	$0.330_{\pm 0.010}$	$0.393_{\pm 0.019}$	$0.437_{\pm 0.011}$	$0.420_{\pm 0.024}$
	MCDropout	$0.086_{\pm 0.006}$	$0.200_{\pm 0.014}$	$0.405_{\pm 0.020}$	$0.498_{\pm 0.017}$	$0.515_{\pm 0.018}$	$0.540_{\pm 0.041}$
	Ensemble	$0.064_{\pm 0.005}$	$0.159_{\pm 0.016}$	$0.361_{\pm 0.022}$	$0.404_{\pm 0.014}$	$0.417_{\pm 0.023}$	$0.472_{\pm 0.031}$
	Laplace	$0.120_{\pm 0.009}$	$0.184_{\pm 0.017}$	$0.381_{\pm 0.025}$	$0.385_{\pm 0.015}$	$0.473_{\pm 0.028}$	$0.408_{\pm 0.056}$
	BLoB	$0.072_{\pm 0.006}$	$0.163_{\pm 0.019}$	$0.376_{\pm 0.014}$	$0.489_{\pm 0.007}$	$0.477_{\pm 0.010}$	$0.497_{\pm 0.025}$
	ScalaBL	$0.069_{\pm 0.002}$	$0.136_{\pm 0.009}$	$0.345_{\pm 0.004}$	$0.396_{\pm 0.012}$	$0.452_{\pm 0.008}$	$0.470_{\pm 0.016}$
	TFB	$0.080_{\pm 0.005}$	$0.182_{\pm 0.009}$	$0.360_{\pm 0.015}$	$0.439_{\pm 0.022}$	$0.468_{\pm 0.027}$	$0.484_{\pm 0.029}$

Table 29: OOD Performance Comparison for Qwen3-0.6B

Metric	Method	Train on circuit_logic		Train on expression_logic	
		In-dist	expression_logic (OOD)	In-dist	circuit_logic (OOD)
ACC (↑)	MLE	$0.841_{\pm 0.001}$	$0.831_{\pm 0.004}$	$0.848_{\pm 0.011}$	$0.625_{\pm 0.018}$
	MAP	$0.843_{\pm 0.005}$	$0.832_{\pm 0.004}$	$0.847_{\pm 0.006}$	$0.623_{\pm 0.015}$
	TempScale	$0.841_{\pm 0.001}$	$0.831_{\pm 0.004}$	$0.848_{\pm 0.011}$	$0.625_{\pm 0.018}$
	MCDropout	$0.843_{\pm 0.008}$	$0.831_{\pm 0.005}$	$0.845_{\pm 0.011}$	$0.628_{\pm 0.025}$
	Ensemble	$0.845_{\pm 0.003}$	$0.830_{\pm 0.003}$	$0.848_{\pm 0.008}$	$0.619_{\pm 0.003}$
	Laplace	$0.840_{\pm 0.007}$	$0.830_{\pm 0.003}$	$0.845_{\pm 0.008}$	$0.626_{\pm 0.016}$
	BLoB	$0.828_{\pm 0.009}$	$0.829_{\pm 0.004}$	$0.831_{\pm 0.006}$	$0.632_{\pm 0.034}$
	ScalaBL	$0.826_{\pm 0.010}$	$0.827_{\pm 0.003}$	$0.834_{\pm 0.009}$	$0.620_{\pm 0.005}$
	TFB	$0.824_{\pm 0.009}$	$0.761_{\pm 0.015}$	$0.826_{\pm 0.016}$	$0.602_{\pm 0.032}$
ECE (↓)	MLE	$0.024_{\pm 0.006}$	$0.092_{\pm 0.017}$	$0.031_{\pm 0.009}$	$0.126_{\pm 0.025}$
	MAP	$0.020_{\pm 0.002}$	$0.097_{\pm 0.009}$	$0.026_{\pm 0.005}$	$0.115_{\pm 0.018}$
	TempScale	$0.020_{\pm 0.003}$	$0.087_{\pm 0.023}$	$0.030_{\pm 0.006}$	$0.123_{\pm 0.033}$
	MCDropout	$0.022_{\pm 0.005}$	$0.092_{\pm 0.016}$	$0.030_{\pm 0.008}$	$0.108_{\pm 0.037}$
	Ensemble	$0.026_{\pm 0.004}$	$0.095_{\pm 0.006}$	$0.030_{\pm 0.004}$	$0.135_{\pm 0.012}$
	Laplace	$0.040_{\pm 0.005}$	$0.134_{\pm 0.014}$	$0.060_{\pm 0.009}$	$0.131_{\pm 0.013}$
	BLoB	$0.028_{\pm 0.008}$	$0.086_{\pm 0.026}$	$0.028_{\pm 0.008}$	$0.079_{\pm 0.028}$
	ScalaBL	$0.037_{\pm 0.003}$	$0.095_{\pm 0.009}$	$0.036_{\pm 0.004}$	$0.128_{\pm 0.009}$
	TFB	$0.084_{\pm 0.008}$	$0.102_{\pm 0.015}$	$0.069_{\pm 0.009}$	$0.033_{\pm 0.011}$
NLL (↓)	MLE	$0.298_{\pm 0.003}$	$0.402_{\pm 0.014}$	$0.288_{\pm 0.003}$	$0.621_{\pm 0.013}$
	MAP	$0.298_{\pm 0.002}$	$0.402_{\pm 0.019}$	$0.289_{\pm 0.004}$	$0.628_{\pm 0.012}$
	TempScale	$0.297_{\pm 0.003}$	$0.392_{\pm 0.016}$	$0.287_{\pm 0.003}$	$0.622_{\pm 0.019}$
	MCDropout	$0.299_{\pm 0.003}$	$0.403_{\pm 0.023}$	$0.289_{\pm 0.002}$	$0.640_{\pm 0.026}$
	Ensemble	$0.297_{\pm 0.002}$	$0.381_{\pm 0.004}$	$0.290_{\pm 0.002}$	$0.617_{\pm 0.008}$
	Laplace	$0.311_{\pm 0.003}$	$0.441_{\pm 0.012}$	$0.318_{\pm 0.007}$	$0.626_{\pm 0.008}$
	BLoB	$0.324_{\pm 0.008}$	$0.380_{\pm 0.023}$	$0.305_{\pm 0.002}$	$0.637_{\pm 0.021}$
	ScalaBL	$0.342_{\pm 0.005}$	$0.389_{\pm 0.002}$	$0.320_{\pm 0.003}$	$0.616_{\pm 0.010}$
	TFB	$0.381_{\pm 0.007}$	$0.535_{\pm 0.009}$	$0.366_{\pm 0.023}$	$0.666_{\pm 0.013}$
Brier (↓)	MLE	$0.195_{\pm 0.002}$	$0.247_{\pm 0.011}$	$0.191_{\pm 0.002}$	$0.434_{\pm 0.015}$
	MAP	$0.195_{\pm 0.002}$	$0.246_{\pm 0.016}$	$0.191_{\pm 0.002}$	$0.440_{\pm 0.012}$
	TempScale	$0.195_{\pm 0.002}$	$0.241_{\pm 0.012}$	$0.191_{\pm 0.002}$	$0.435_{\pm 0.021}$
	MCDropout	$0.196_{\pm 0.002}$	$0.249_{\pm 0.019}$	$0.191_{\pm 0.001}$	$0.449_{\pm 0.025}$
	Ensemble	$0.194_{\pm 0.001}$	$0.231_{\pm 0.002}$	$0.191_{\pm 0.001}$	$0.429_{\pm 0.007}$
	Laplace	$0.198_{\pm 0.001}$	$0.272_{\pm 0.011}$	$0.198_{\pm 0.002}$	$0.437_{\pm 0.009}$
	BLoB	$0.205_{\pm 0.003}$	$0.232_{\pm 0.017}$	$0.199_{\pm 0.001}$	$0.446_{\pm 0.020}$
	ScalaBL	$0.211_{\pm 0.002}$	$0.235_{\pm 0.001}$	$0.203_{\pm 0.002}$	$0.430_{\pm 0.011}$
	TFB	$0.238_{\pm 0.004}$	$0.353_{\pm 0.009}$	$0.230_{\pm 0.017}$	$0.473_{\pm 0.012}$

Table 30: OOD Performance Comparison for Qwen3-1.7B

Metric	Method	Train on circuit_logic		Train on expression_logic	
		In-dist	expression_logic (OOD)	In-dist	circuit_logic (OOD)
ACC (↑)	MLE	$0.838_{\pm 0.004}$	$0.784_{\pm 0.038}$	$0.846_{\pm 0.002}$	$0.606_{\pm 0.007}$
	MAP	$0.848_{\pm 0.006}$	$0.746_{\pm 0.099}$	$0.846_{\pm 0.004}$	$0.608_{\pm 0.011}$
	TempScale	$0.838_{\pm 0.004}$	$0.784_{\pm 0.038}$	$0.846_{\pm 0.002}$	$0.606_{\pm 0.007}$
	MCDropout	$0.842_{\pm 0.010}$	$0.791_{\pm 0.037}$	$0.847_{\pm 0.006}$	$0.613_{\pm 0.009}$
	Ensemble	$0.829_{\pm 0.003}$	$0.758_{\pm 0.093}$	$0.842_{\pm 0.005}$	$0.610_{\pm 0.004}$
	Laplace	$0.841_{\pm 0.005}$	$0.783_{\pm 0.042}$	$0.846_{\pm 0.004}$	$0.606_{\pm 0.006}$
	BLoB	$0.827_{\pm 0.008}$	$0.664_{\pm 0.044}$	$0.830_{\pm 0.005}$	$0.613_{\pm 0.001}$
	ScalaBL	$0.830_{\pm 0.007}$	$0.713_{\pm 0.084}$	$0.830_{\pm 0.006}$	$0.613_{\pm 0.001}$
	TFB	$0.829_{\pm 0.010}$	$0.671_{\pm 0.039}$	$0.830_{\pm 0.007}$	$0.578_{\pm 0.029}$
ECE (↓)	MLE	$0.020_{\pm 0.002}$	$0.143_{\pm 0.037}$	$0.025_{\pm 0.006}$	$0.090_{\pm 0.053}$
	MAP	$0.027_{\pm 0.003}$	$0.128_{\pm 0.035}$	$0.029_{\pm 0.008}$	$0.073_{\pm 0.031}$
	TempScale	$0.014_{\pm 0.004}$	$0.134_{\pm 0.032}$	$0.023_{\pm 0.003}$	$0.101_{\pm 0.049}$
	MCDropout	$0.021_{\pm 0.004}$	$0.145_{\pm 0.036}$	$0.027_{\pm 0.006}$	$0.083_{\pm 0.027}$
	Ensemble	$0.045_{\pm 0.007}$	$0.165_{\pm 0.034}$	$0.027_{\pm 0.006}$	$0.110_{\pm 0.058}$
	Laplace	$0.038_{\pm 0.005}$	$0.169_{\pm 0.037}$	$0.065_{\pm 0.007}$	$0.090_{\pm 0.025}$
	BLoB	$0.029_{\pm 0.003}$	$0.149_{\pm 0.047}$	$0.033_{\pm 0.007}$	$0.104_{\pm 0.058}$
	ScalaBL	$0.034_{\pm 0.009}$	$0.114_{\pm 0.057}$	$0.037_{\pm 0.006}$	$0.141_{\pm 0.016}$
	TFB	$0.072_{\pm 0.006}$	$0.061_{\pm 0.021}$	$0.067_{\pm 0.005}$	$0.049_{\pm 0.007}$
NLL (↓)	MLE	$0.294_{\pm 0.001}$	$0.518_{\pm 0.026}$	$0.284_{\pm 0.003}$	$0.655_{\pm 0.012}$
	MAP	$0.292_{\pm 0.001}$	$0.544_{\pm 0.051}$	$0.284_{\pm 0.003}$	$0.650_{\pm 0.013}$
	TempScale	$0.293_{\pm 0.001}$	$0.508_{\pm 0.025}$	$0.282_{\pm 0.003}$	$0.661_{\pm 0.024}$
	MCDropout	$0.293_{\pm 0.001}$	$0.526_{\pm 0.024}$	$0.285_{\pm 0.002}$	$0.648_{\pm 0.011}$
	Ensemble	$0.325_{\pm 0.005}$	$0.531_{\pm 0.070}$	$0.287_{\pm 0.002}$	$0.651_{\pm 0.009}$
	Laplace	$0.304_{\pm 0.001}$	$0.544_{\pm 0.032}$	$0.320_{\pm 0.007}$	$0.648_{\pm 0.009}$
	BLoB	$0.314_{\pm 0.005}$	$0.555_{\pm 0.033}$	$0.304_{\pm 0.003}$	$0.671_{\pm 0.018}$
	ScalaBL	$0.343_{\pm 0.002}$	$0.560_{\pm 0.033}$	$0.322_{\pm 0.002}$	$0.643_{\pm 0.003}$
	TFB	$0.364_{\pm 0.004}$	$0.614_{\pm 0.029}$	$0.355_{\pm 0.005}$	$0.681_{\pm 0.017}$
Brier (↓)	MLE	$0.194_{\pm 0.001}$	$0.338_{\pm 0.021}$	$0.189_{\pm 0.002}$	$0.463_{\pm 0.012}$
	MAP	$0.193_{\pm 0.001}$	$0.361_{\pm 0.049}$	$0.189_{\pm 0.002}$	$0.459_{\pm 0.012}$
	TempScale	$0.194_{\pm 0.001}$	$0.330_{\pm 0.020}$	$0.190_{\pm 0.002}$	$0.467_{\pm 0.020}$
	MCDropout	$0.193_{\pm 0.001}$	$0.344_{\pm 0.022}$	$0.190_{\pm 0.001}$	$0.457_{\pm 0.011}$
	Ensemble	$0.207_{\pm 0.004}$	$0.351_{\pm 0.061}$	$0.190_{\pm 0.001}$	$0.461_{\pm 0.007}$
	Laplace	$0.196_{\pm 0.001}$	$0.359_{\pm 0.027}$	$0.198_{\pm 0.003}$	$0.457_{\pm 0.006}$
	BLoB	$0.202_{\pm 0.001}$	$0.380_{\pm 0.034}$	$0.199_{\pm 0.001}$	$0.478_{\pm 0.015}$
	ScalaBL	$0.211_{\pm 0.001}$	$0.377_{\pm 0.033}$	$0.203_{\pm 0.001}$	$0.455_{\pm 0.003}$
	TFB	$0.227_{\pm 0.003}$	$0.425_{\pm 0.027}$	$0.223_{\pm 0.002}$	$0.487_{\pm 0.016}$

Table 31: OOD Performance Comparison for Qwen3-4B

Metric	Method	Train on circuit_logic		Train on expression_logic	
		In-dist	expression_logic (OOD)	In-dist	circuit_logic (OOD)
ACC (↑)	MLE	$0.848_{\pm 0.009}$	$0.660_{\pm 0.069}$	$0.841_{\pm 0.008}$	$0.613_{\pm 0.000}$
	MAP	$0.847_{\pm 0.004}$	$0.659_{\pm 0.061}$	$0.840_{\pm 0.005}$	$0.613_{\pm 0.000}$
	TempScale	$0.848_{\pm 0.009}$	$0.660_{\pm 0.069}$	$0.841_{\pm 0.008}$	$0.613_{\pm 0.000}$
	MCDropout	$0.850_{\pm 0.009}$	$0.641_{\pm 0.023}$	$0.841_{\pm 0.003}$	$0.613_{\pm 0.000}$
	Ensemble	$0.835_{\pm 0.008}$	$0.781_{\pm 0.087}$	$0.842_{\pm 0.002}$	$0.613_{\pm 0.001}$
	Laplace	$0.854_{\pm 0.001}$	$0.658_{\pm 0.067}$	$0.843_{\pm 0.012}$	$0.613_{\pm 0.000}$
	BLoB	$0.827_{\pm 0.006}$	$0.803_{\pm 0.034}$	$0.831_{\pm 0.003}$	$0.625_{\pm 0.023}$
	ScalaBL	$0.830_{\pm 0.009}$	$0.784_{\pm 0.067}$	$0.830_{\pm 0.009}$	$0.614_{\pm 0.001}$
	TFB	$0.817_{\pm 0.012}$	$0.626_{\pm 0.034}$	$0.809_{\pm 0.011}$	$0.635_{\pm 0.008}$
ECE (↓)	MLE	$0.028_{\pm 0.009}$	$0.182_{\pm 0.057}$	$0.025_{\pm 0.008}$	$0.266_{\pm 0.017}$
	MAP	$0.024_{\pm 0.005}$	$0.181_{\pm 0.054}$	$0.025_{\pm 0.005}$	$0.267_{\pm 0.017}$
	TempScale	$0.021_{\pm 0.006}$	$0.179_{\pm 0.064}$	$0.024_{\pm 0.008}$	$0.293_{\pm 0.019}$
	MCDropout	$0.033_{\pm 0.004}$	$0.203_{\pm 0.031}$	$0.024_{\pm 0.004}$	$0.257_{\pm 0.012}$
	Ensemble	$0.034_{\pm 0.006}$	$0.159_{\pm 0.027}$	$0.026_{\pm 0.005}$	$0.249_{\pm 0.035}$
	Laplace	$0.043_{\pm 0.010}$	$0.183_{\pm 0.053}$	$0.054_{\pm 0.016}$	$0.238_{\pm 0.014}$
	BLoB	$0.027_{\pm 0.005}$	$0.124_{\pm 0.061}$	$0.023_{\pm 0.006}$	$0.214_{\pm 0.069}$
	ScalaBL	$0.033_{\pm 0.006}$	$0.126_{\pm 0.031}$	$0.035_{\pm 0.008}$	$0.222_{\pm 0.032}$
	TFB	$0.088_{\pm 0.010}$	$0.035_{\pm 0.012}$	$0.086_{\pm 0.009}$	$0.064_{\pm 0.015}$
NLL (↓)	MLE	$0.293_{\pm 0.002}$	$0.554_{\pm 0.035}$	$0.288_{\pm 0.002}$	$0.773_{\pm 0.042}$
	MAP	$0.293_{\pm 0.001}$	$0.543_{\pm 0.042}$	$0.288_{\pm 0.002}$	$0.774_{\pm 0.045}$
	TempScale	$0.291_{\pm 0.002}$	$0.557_{\pm 0.042}$	$0.286_{\pm 0.001}$	$0.848_{\pm 0.065}$
	MCDropout	$0.293_{\pm 0.001}$	$0.552_{\pm 0.027}$	$0.288_{\pm 0.001}$	$0.750_{\pm 0.034}$
	Ensemble	$0.319_{\pm 0.005}$	$0.488_{\pm 0.048}$	$0.288_{\pm 0.002}$	$0.705_{\pm 0.095}$
	Laplace	$0.296_{\pm 0.002}$	$0.557_{\pm 0.032}$	$0.309_{\pm 0.005}$	$0.703_{\pm 0.002}$
	BLoB	$0.316_{\pm 0.005}$	$0.470_{\pm 0.028}$	$0.302_{\pm 0.001}$	$0.676_{\pm 0.146}$
	ScalaBL	$0.340_{\pm 0.008}$	$0.477_{\pm 0.027}$	$0.315_{\pm 0.005}$	$0.674_{\pm 0.065}$
	TFB	$0.411_{\pm 0.004}$	$0.640_{\pm 0.022}$	$0.413_{\pm 0.010}$	$0.650_{\pm 0.007}$
Brier (↓)	MLE	$0.193_{\pm 0.001}$	$0.383_{\pm 0.034}$	$0.191_{\pm 0.001}$	$0.546_{\pm 0.025}$
	MAP	$0.193_{\pm 0.001}$	$0.372_{\pm 0.038}$	$0.191_{\pm 0.001}$	$0.544_{\pm 0.026}$
	TempScale	$0.193_{\pm 0.001}$	$0.389_{\pm 0.042}$	$0.191_{\pm 0.001}$	$0.581_{\pm 0.032}$
	MCDropout	$0.192_{\pm 0.001}$	$0.379_{\pm 0.025}$	$0.191_{\pm 0.001}$	$0.532_{\pm 0.021}$
	Ensemble	$0.202_{\pm 0.001}$	$0.314_{\pm 0.041}$	$0.192_{\pm 0.001}$	$0.500_{\pm 0.062}$
	Laplace	$0.193_{\pm 0.001}$	$0.381_{\pm 0.030}$	$0.196_{\pm 0.003}$	$0.505_{\pm 0.002}$
	BLoB	$0.202_{\pm 0.002}$	$0.302_{\pm 0.020}$	$0.198_{\pm 0.001}$	$0.473_{\pm 0.104}$
	ScalaBL	$0.210_{\pm 0.002}$	$0.307_{\pm 0.027}$	$0.201_{\pm 0.003}$	$0.480_{\pm 0.048}$
	TFB	$0.258_{\pm 0.005}$	$0.450_{\pm 0.020}$	$0.261_{\pm 0.008}$	$0.456_{\pm 0.006}$

Table 32: OOD Performance Comparison for Qwen3-8B

Metric	Method	Train on circuit_logic		Train on expression_logic	
		In-dist	expression_logic (OOD)	In-dist	circuit_logic (OOD)
ACC (↑)	MLE	$0.842_{\pm 0.003}$	$0.823_{\pm 0.010}$	$0.840_{\pm 0.008}$	$0.645_{\pm 0.029}$
	MAP	$0.840_{\pm 0.006}$	$0.826_{\pm 0.007}$	$0.837_{\pm 0.008}$	$0.668_{\pm 0.058}$
	TempScale	$0.842_{\pm 0.003}$	$0.823_{\pm 0.010}$	$0.840_{\pm 0.008}$	$0.645_{\pm 0.029}$
	MCDropout	$0.842_{\pm 0.004}$	$0.806_{\pm 0.022}$	$0.838_{\pm 0.003}$	$0.669_{\pm 0.063}$
	Ensemble	$0.824_{\pm 0.002}$	$0.825_{\pm 0.011}$	$0.843_{\pm 0.002}$	$0.652_{\pm 0.006}$
	Laplace	$0.840_{\pm 0.000}$	$0.824_{\pm 0.010}$	$0.840_{\pm 0.002}$	$0.641_{\pm 0.001}$
	BLoB	$0.843_{\pm 0.006}$	$0.830_{\pm 0.004}$	$0.833_{\pm 0.007}$	$0.740_{\pm 0.033}$
	ScalaBL	$0.825_{\pm 0.011}$	$0.826_{\pm 0.009}$	$0.822_{\pm 0.012}$	$0.705_{\pm 0.070}$
	TFB	$0.830_{\pm 0.005}$	$0.766_{\pm 0.025}$	$0.825_{\pm 0.002}$	$0.589_{\pm 0.026}$
ECE (↓)	MLE	$0.027_{\pm 0.007}$	$0.119_{\pm 0.017}$	$0.023_{\pm 0.008}$	$0.108_{\pm 0.050}$
	MAP	$0.020_{\pm 0.002}$	$0.124_{\pm 0.019}$	$0.024_{\pm 0.007}$	$0.107_{\pm 0.029}$
	TempScale	$0.018_{\pm 0.006}$	$0.100_{\pm 0.017}$	$0.020_{\pm 0.003}$	$0.111_{\pm 0.067}$
	MCDropout	$0.023_{\pm 0.004}$	$0.120_{\pm 0.028}$	$0.023_{\pm 0.003}$	$0.109_{\pm 0.026}$
	Ensemble	$0.034_{\pm 0.003}$	$0.114_{\pm 0.009}$	$0.021_{\pm 0.006}$	$0.098_{\pm 0.000}$
	Laplace	$0.055_{\pm 0.009}$	$0.154_{\pm 0.039}$	$0.053_{\pm 0.009}$	$0.092_{\pm 0.027}$
	BLoB	$0.032_{\pm 0.007}$	$0.111_{\pm 0.013}$	$0.030_{\pm 0.006}$	$0.145_{\pm 0.034}$
	ScalaBL	$0.058_{\pm 0.014}$	$0.138_{\pm 0.011}$	$0.046_{\pm 0.007}$	$0.156_{\pm 0.044}$
	TFB	$0.076_{\pm 0.001}$	$0.119_{\pm 0.014}$	$0.070_{\pm 0.016}$	$0.036_{\pm 0.014}$
NLL (↓)	MLE	$0.294_{\pm 0.004}$	$0.430_{\pm 0.033}$	$0.291_{\pm 0.005}$	$0.624_{\pm 0.033}$
	MAP	$0.295_{\pm 0.004}$	$0.435_{\pm 0.048}$	$0.293_{\pm 0.003}$	$0.602_{\pm 0.031}$
	TempScale	$0.292_{\pm 0.005}$	$0.412_{\pm 0.033}$	$0.289_{\pm 0.005}$	$0.625_{\pm 0.049}$
	MCDropout	$0.295_{\pm 0.004}$	$0.477_{\pm 0.034}$	$0.291_{\pm 0.003}$	$0.614_{\pm 0.040}$
	Ensemble	$0.314_{\pm 0.004}$	$0.395_{\pm 0.008}$	$0.291_{\pm 0.000}$	$0.585_{\pm 0.002}$
	Laplace	$0.318_{\pm 0.016}$	$0.470_{\pm 0.052}$	$0.314_{\pm 0.003}$	$0.637_{\pm 0.017}$
	BLoB	$0.311_{\pm 0.002}$	$0.395_{\pm 0.016}$	$0.310_{\pm 0.004}$	$0.583_{\pm 0.024}$
	ScalaBL	$0.356_{\pm 0.009}$	$0.463_{\pm 0.038}$	$0.323_{\pm 0.005}$	$0.625_{\pm 0.018}$
	TFB	$0.373_{\pm 0.003}$	$0.538_{\pm 0.024}$	$0.360_{\pm 0.007}$	$0.668_{\pm 0.013}$
Brier (↓)	MLE	$0.194_{\pm 0.002}$	$0.266_{\pm 0.026}$	$0.193_{\pm 0.002}$	$0.435_{\pm 0.032}$
	MAP	$0.193_{\pm 0.002}$	$0.271_{\pm 0.039}$	$0.194_{\pm 0.001}$	$0.416_{\pm 0.030}$
	TempScale	$0.194_{\pm 0.001}$	$0.254_{\pm 0.024}$	$0.193_{\pm 0.002}$	$0.435_{\pm 0.043}$
	MCDropout	$0.194_{\pm 0.002}$	$0.306_{\pm 0.029}$	$0.193_{\pm 0.001}$	$0.426_{\pm 0.037}$
	Ensemble	$0.203_{\pm 0.000}$	$0.236_{\pm 0.005}$	$0.193_{\pm 0.001}$	$0.401_{\pm 0.003}$
	Laplace	$0.199_{\pm 0.006}$	$0.294_{\pm 0.041}$	$0.198_{\pm 0.001}$	$0.445_{\pm 0.016}$
	BLoB	$0.200_{\pm 0.001}$	$0.238_{\pm 0.010}$	$0.200_{\pm 0.001}$	$0.398_{\pm 0.019}$
	ScalaBL	$0.217_{\pm 0.005}$	$0.289_{\pm 0.034}$	$0.204_{\pm 0.002}$	$0.434_{\pm 0.018}$
	TFB	$0.233_{\pm 0.002}$	$0.356_{\pm 0.021}$	$0.227_{\pm 0.005}$	$0.475_{\pm 0.013}$

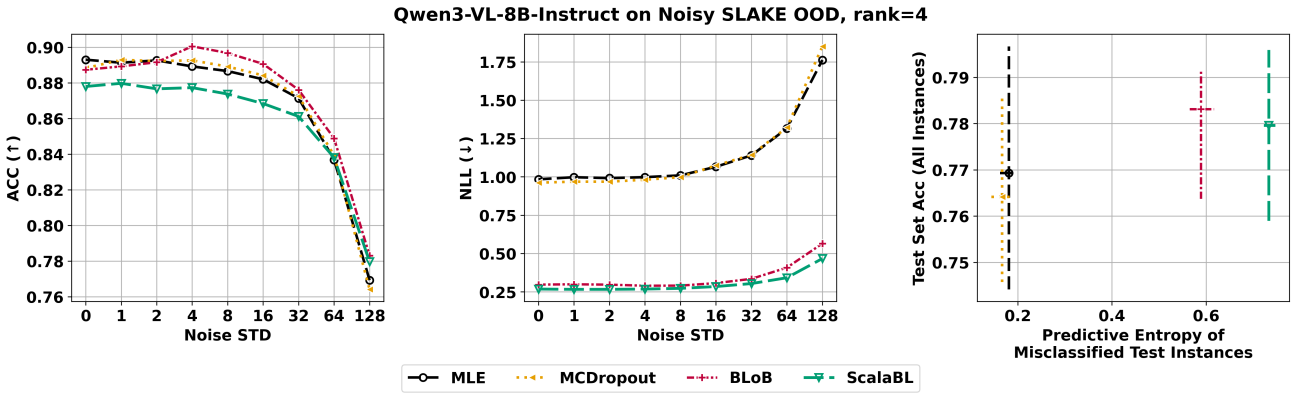


Figure 11: Effect of Noise on SLAKE dataset using Qwen3-VL-8B-Instruct with $r = 4$

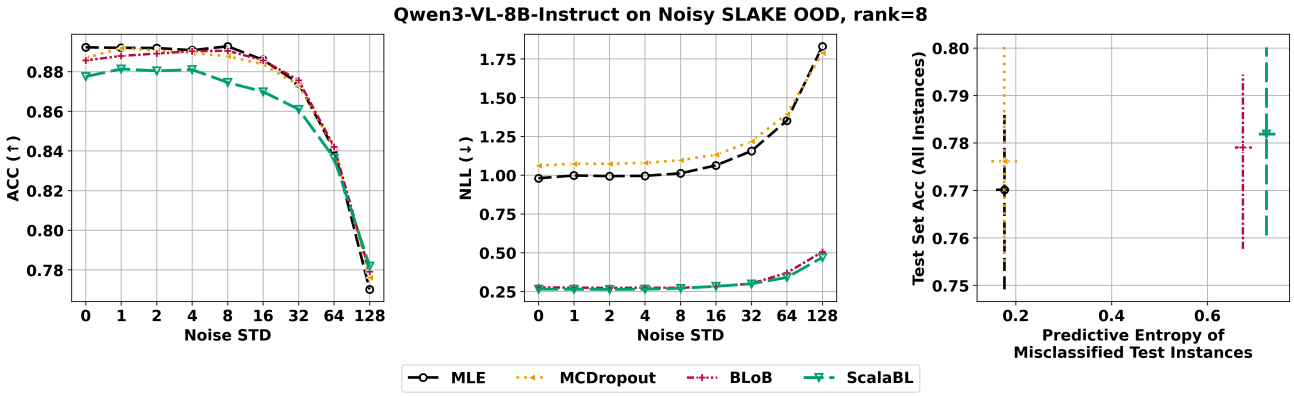


Figure 12: Effect of Noise on SLAKE dataset using Qwen3-VL-8B-Instruct with $r = 8$

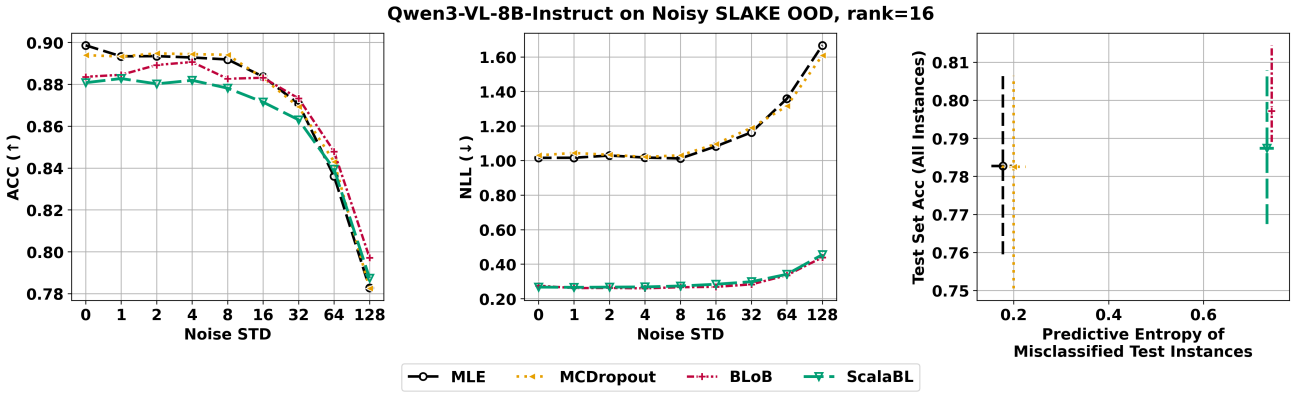


Figure 13: Effect of Noise on SLAKE dataset using Qwen3-VL-8B-Instruct with $r = 16$

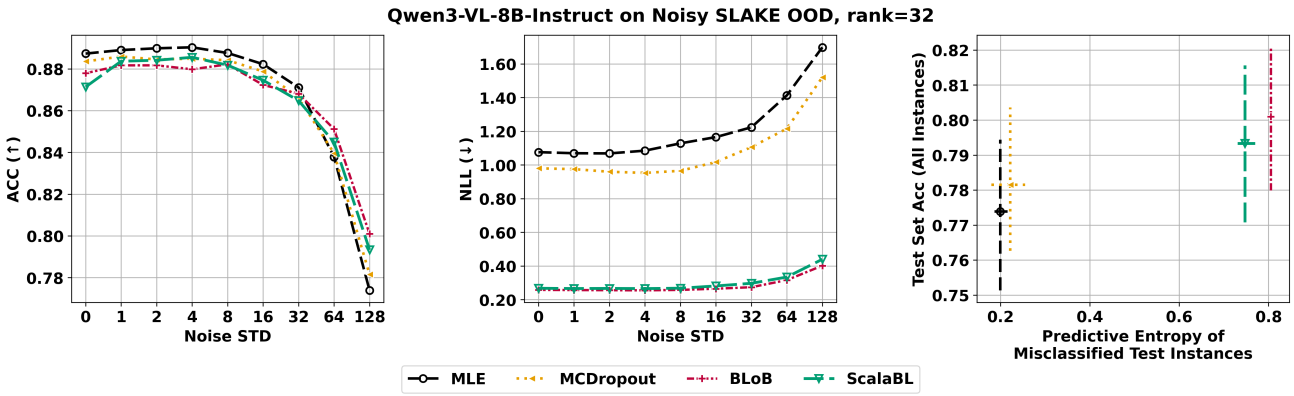


Figure 14: Effect of Noise on SLAKE dataset using Qwen3-VL-8B-Instruct with $r = 32$

Table 33: Gemma versus Qwen on Winogrande Small In-Distribution

Metric	Method	gemma-3-4b-it	Qwen3-4B
ACC (↑)	MLE	$0.681_{\pm 0.002}$	$0.711_{\pm 0.004}$
	MCDropout	$0.684_{\pm 0.003}$	$0.709_{\pm 0.004}$
	BLoB	$0.682_{\pm 0.005}$	$0.727_{\pm 0.005}$
	ScalaBL	$0.663_{\pm 0.003}$	$0.716_{\pm 0.007}$
ECE (↓)	MLE	$0.311_{\pm 0.003}$	$0.268_{\pm 0.004}$
	MCDropout	$0.298_{\pm 0.003}$	$0.270_{\pm 0.003}$
	BLoB	$0.110_{\pm 0.011}$	$0.099_{\pm 0.011}$
	ScalaBL	$0.085_{\pm 0.003}$	$0.110_{\pm 0.005}$
NLL (↓)	MLE	$4.182_{\pm 1.204}$	$2.789_{\pm 0.202}$
	MCDropout	$3.562_{\pm 0.871}$	$2.773_{\pm 0.330}$
	BLoB	$0.636_{\pm 0.015}$	$0.591_{\pm 0.021}$
	ScalaBL	$0.633_{\pm 0.003}$	$0.621_{\pm 0.010}$

Table 34: Gemma versus Qwen on SLAKE In-Distribution

Metric	Method	gemma-3-4b-it	Qwen3-VL-4B-Instruct
ACC (↑)	MLE	$0.875_{\pm 0.006}$	$0.894_{\pm 0.005}$
	MCDropout	$0.865_{\pm 0.017}$	$0.899_{\pm 0.004}$
	BLoB	$0.875_{\pm 0.011}$	$0.900_{\pm 0.006}$
	ScalaBL	$0.840_{\pm 0.005}$	$0.891_{\pm 0.007}$
ECE (↓)	MLE	$0.116_{\pm 0.006}$	$0.100_{\pm 0.004}$
	MCDropout	$0.123_{\pm 0.012}$	$0.095_{\pm 0.004}$
	BLoB	$0.035_{\pm 0.006}$	$0.032_{\pm 0.008}$
	ScalaBL	$0.045_{\pm 0.010}$	$0.035_{\pm 0.008}$
NLL (↓)	MLE	$0.742_{\pm 0.054}$	$0.866_{\pm 0.047}$
	MCDropout	$0.675_{\pm 0.034}$	$0.856_{\pm 0.147}$
	BLoB	$0.292_{\pm 0.003}$	$0.240_{\pm 0.007}$
	ScalaBL	$0.350_{\pm 0.004}$	$0.258_{\pm 0.010}$

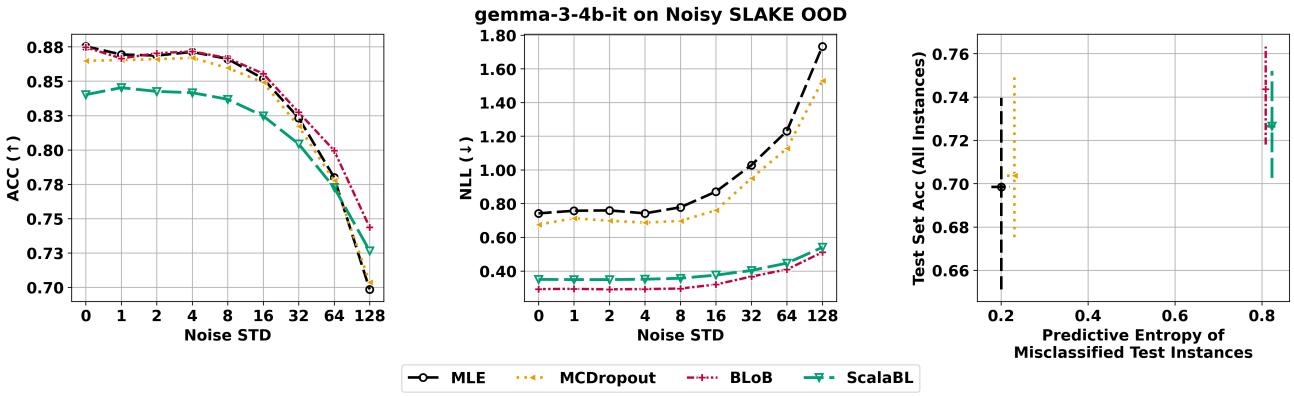


Figure 15: Effect of Noise on SLAKE dataset using Gemma-4B

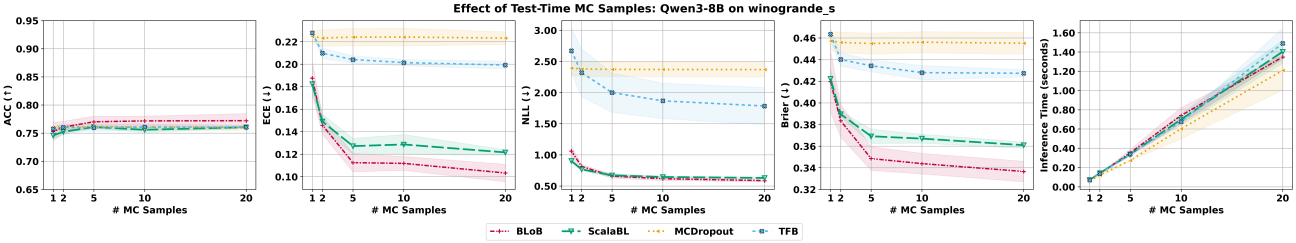


Figure 16: Effect of Number of Test Time Monte Carlo Samples on Winogrande-Small

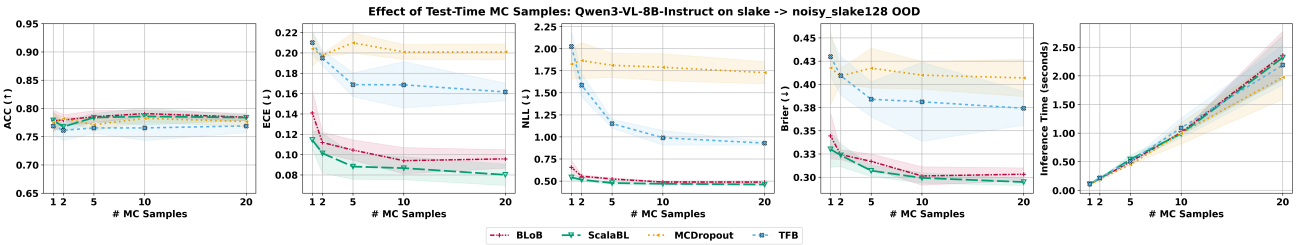


Figure 17: Effect of Number of Test Time Monte Carlo Samples on Noisy SLAKE OOD (Highest Noise Level)