C-LoRA: Contextual Low-Rank Adaptation for Uncertainty Estimation in Large Language Models

Amir Hossein Rahmati[1] **Sanket Jantre**[2] **Weifeng Zhang**[1] **Yucheng Wang**[1]
Byung-Jun Yoon[1,2] **Nathan M. Urban**[2] **Xiaoning Qian**[1,2]

[1]Texas A&M University, College Station, TX [2]Brookhaven National Laboratory, Upton, NY

{amir_hossein_rahmati, weifengzhang, wangyucheng, bjyoon, xqian}@tamu.edu
{sjantre, nurban, byoon, xqian1}@bnl.gov

Abstract

Low-Rank Adaptation (LoRA) offers a cost-effective solution for fine-tuning large language models (LLMs), but it often produces overconfident predictions in data-scarce few-shot settings. To address this issue, several classical statistical learning approaches have been repurposed for scalable uncertainty-aware LoRA fine-tuning. However, these approaches neglect how input characteristics affect the predictive uncertainty estimates. To address this limitation, we propose Contextual Low-Rank Adaptation (**C-LoRA**) as a novel uncertainty-aware and parameter efficient fine-tuning approach, by developing new lightweight LoRA modules contextualized to each input data sample to dynamically adapt uncertainty estimates. Incorporating data-driven contexts into the parameter posteriors, C-LoRA mitigates overfitting, achieves well-calibrated uncertainties, and yields robust predictions. Extensive experiments on LLaMA2-7B models demonstrate that C-LoRA consistently outperforms the state-of-the-art uncertainty-aware LoRA methods in both uncertainty quantification and model generalization. Ablation studies further confirm the critical role of our contextual modules in capturing sample-specific uncertainties. C-LoRA sets a new standard for robust, uncertainty-aware LLM fine-tuning in few-shot regimes. Although our experiments are limited to 7B models, our method is architecture-agnostic and, in principle, applies beyond this scale; studying its scaling to larger models remains an open problem. Our code is available at https://github.com/ahra99/c_lora.

1 Introduction

Large Language Models (LLMs) [1–7] have shown their promising potential in diverse areas [8–18]. Due to their general-purpose language understanding and generation capabilities with human-level performance [1, 2], fine-tuning LLMs to various downstream tasks has drawn significant attention [19–23]. However, when fine-tuned for downstream tasks with limited data, LLMs may hallucinate to produce overconfident results, which becomes a serious concern [24–27]. To address this, reliable estimation of uncertainty has become essential [19, 20, 28].

To enable predictive uncertainty quantification (UQ), probabilistic inference with Bayesian neural network (BNN) has been studied in deep learning [29–41], where neural network weights are treated as random variables with variational inference (VI) approximating the true posterior to provide reliable UQ [42, 43]. Adopting these approaches for LLMs is limited by their prohibitive computational and memory costs compared to conventional methods [19, 20, 22], making full Bayesian fine-tuning of all the parameters of an LLM challenging.

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

Parameter Efficient Fine-Tuning (PEFT) methods, such as Low-rank Adaptation (LoRA), substantially reduce the number of learnable parameters and thereby mitigate the significant computational expenses and excessive memory utilization [21, 44–48]. This efficiency facilitates employing Bayesian methods for UQ in LLMs [19, 20, 49–51]. Most of the recent studies considered a more straightforward solution, like in [49–51] where authors considered ensemble methods. [19] proposed a *post-hoc* Bayesian inference method for the LoRA adapters using Laplace approximation [34]. [20] repurposed a mean-field VI based Bayesian framework for LoRA-based fine-tuning to jointly estimate the LoRA parameters' variational means and variances [20]. However, the uncertainty stemming from data, *aleatoric uncertainty*, has not been considered in the existing methods, leading to poor performance especially when fine-tuning with limited data. This motivates us to focus on incorporating aleatoric uncertainty for a novel parameter efficient fine-tuning approach in this work.

We propose **C**ontextual **L**ow-**R**ank **A**daptation, **C-LoRA**, which enables an explicit consideration of aleatoric uncertainty (data uncertainty). This formulation not only leads to faster training, but also provides a sample-dependent uncertainty estimation for each layer, which leads to beneficial and more favorable uncertainty-aware LLM fine-tuning under small-data scenarios. In particular, we introduce a contextual module for modeling the stochasticity in low-dimensional space dependent on the data, which enables a low-cost contextualized estimation of prediction uncertainty in terms of computation and memory usage. Our contributions can be outlined as follows:

- We propose a new end-to-end Bayesian framework for scalable uncertainty-aware LLM fine-tuning via contextualized LoRA on data in the lower-dimensional space by a flexible contextual module;
- Our framework allows for efficient modeling of aleatoric (data) uncertainty;
- We showcase the superiority of C-LoRA regarding both UQ capabilities and generalizability via extensive experiments across different tasks;
- Through ablation experiments, we demonstrate the significance of our new contextual module on achieving better UQ while offering competitive accuracy with only minor drops on some tasks.

2 Preliminaries

In this paper, vectors and matrices are denoted by bold lowercase and uppercase letters, respectively. Most of our mathematical notations follow the ones adopted in [21], [19], [20], and [39].

2.1 Low-Rank Adaptation (LoRA)

LoRA, a parameter-efficient fine-tuning approach, adapts a pre-trained language model to downstream tasks [21]. It rests on the assumption that the required weight updates have a low intrinsic dimensionality, so LoRA freezes the original weights and instead learns low-rank update matrices. To this end, the modified forward pass becomes:

$$\mathbf{h} = (\mathbf{W}_0 + \Delta\mathbf{W})\mathbf{x} = (\mathbf{W}_0 + \mathbf{BA})\mathbf{x}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^k$ and $\mathbf{h} \in \mathbb{R}^d$ are input and output vectors, respectively, and $\mathbf{W}_0 \in \mathbb{R}^{d\times k}$ represents the frozen pre-trained weights. LoRA inserts two low-rank update factors $\mathbf{B} \in \mathbb{R}^{d\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times k}$ with $r \ll \min(d, k)$. Hence, the number of trainable parameters reduces to $r \times (d + k)$ which is considerably lower than $d\times k$ in the full matrix. This dramatically cuts storage and computational costs while matching full-matrix fine-tuning performance. Here on, we set $k = d$, so that $\mathbf{W}_0 \in \mathbb{R}^{d\times d}$.

2.2 Bayesian Uncertainty Estimation

Let $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ be a dataset of N independent and identically distributed (i.i.d.) observations, where each $\mathbf{x}_i$ is an input sample and y_i is the corresponding output. In the Bayesian paradigm, rather than selecting a single best-fit model parameterization, we maintain a distribution over all plausible model parameters. Specifically, given model parameters $\boldsymbol{\theta}$, Bayesian inference aims to characterize the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ using Bayes' rule: $p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})$, where $p(\mathcal{D}|\boldsymbol{\theta})$ is the likelihood of the observed data under parameters $\boldsymbol{\theta}$, and $p(\boldsymbol{\theta})$ is the prior distribution reflecting beliefs about $\boldsymbol{\theta}$ before seeing any data.

To generate predictions for a new input $\mathbf{x}^*$, Bayesian model averaging, which integrates over the posterior is applied and the intractable integral is approximated using Monte Carlo sampling:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \int p(y^*|\mathbf{x}^*, \boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta} \approx \frac{1}{M}\sum_{m=1}^{M} p(y^*|\mathbf{x}^*, \boldsymbol{\theta}_m), \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D}). \tag{2}$$

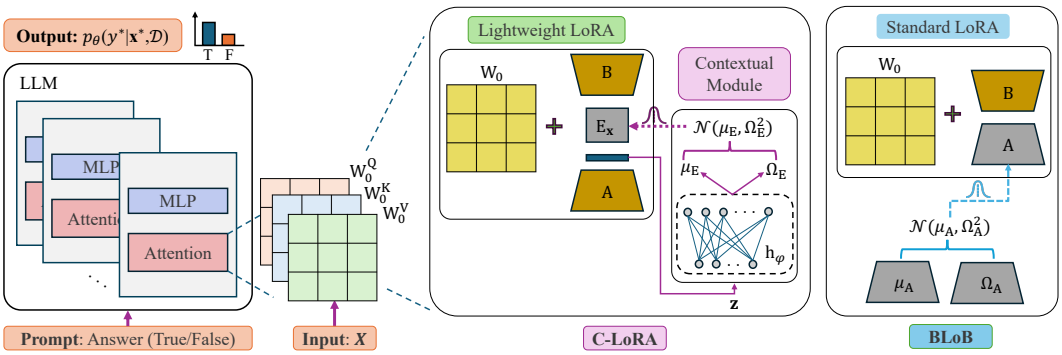

Figure 1: A visual representation of our proposed method, Conextual LoRA **(C-LoRA)** and the Bayesian LoRA by backpropagation **(BLoB)**.

2.3 Bayesian Low-Rank Adaptation

Despite the existence of scalable posterior inference methods, a full Bayesian treatment of LLMs remains computationally prohibitive. By restricting Bayesian updates to the LoRA parameters, a tractable uncertainty quantification scheme can be achieved; however, even Markov chain Monte Carlo (MCMC) over millions of LoRA weights is too costly. As a practical compromise, *Bayesian LoRA by backpropagation (BLoB)* [20] embeds uncertainty estimation directly into fine-tuning via mean-field variational inference on LoRA adapters. More specifically, they keep $\mathbf{B}$ deterministic in (1) and Bayesianize $\mathbf{A}$ with a variational distribution $q(\mathbf{A}) = \mathcal{N}(\mathbf{A}|\boldsymbol{\mu}_\mathbf{A}, \boldsymbol{\Omega}^2_\mathbf{A})$, where $\boldsymbol{\mu}_\mathbf{A}$ and $\boldsymbol{\Omega}^2_\mathbf{A}$ are the variational mean and variance estimates, respectively. To this end, they learn the variational distribution parameters by maximizing the Evidence Lower BOund (ELBO):

$$\mathcal{L}' = \mathbb{E}_q\big[\log p(\mathcal{D}|\mathbf{A},\mathbf{B})\big] \; - \; \mathrm{KL}\big[q(\mathbf{A}) \,\|\, p(\mathbf{A})\big]. \tag{3}$$

Here, the first term measures the expected negative log-likelihood of the data under the variational posterior, while the second term is the Kullback-Leibler (KL) divergence between the variational posterior and the prior, acting as a regularization. Using the reparameterization trick, BLoB jointly updates means and variances providing scalable, predictive uncertainty estimates. A visual representation of BLoB framework is presented in Figure 1.

3 Contextual Low-Rank Adaptation (C-LoRA)

We introduce the formulation of Contextual Low-Rank Adaptation (C-LoRA), which enables scalable, efficient, data-dependent uncertainty quantification at the sample level. First, we introduce a lightweight LoRA factorization to reduce the computational burden of variational inference in Bayesian LoRA. Based on this factorization, we treat the LoRA weights' stochasticity to be data-dependent, and learn the weight distribution parameters via a variational Bayesian objective. A step-by-step description of C-LoRA is presented in Algorithm 1 (Appendix F).

3.1 Lightweight LoRA Factorization

The standard LoRA update in (1) introduces two low-rank matrices whose size scales with the frozen weight dimension d, so that any Bayesian treatment—whether both adapters are stochastic or only $\mathbf{A}$ is (as in BLoB)—incurs a computational cost that grows linearly in d. To break this dependency, we insert a low-dimensional matrix $\mathbf{E} \in \mathbb{R}^{r\times r}$ between $\mathbf{B}$ and $\mathbf{A}$. This modification reduces the stochastic parameter complexity to a constant in d, yielding a modified LoRA factorization as follows

$$\mathbf{h} = (\mathbf{W}_0 + \Delta\mathbf{W})\mathbf{x} = (\mathbf{W}_0 + \mathbf{BEA})\mathbf{x}. \tag{4}$$

Compared to the BLoB mean-field VI framework, this new factorization facilitates a more scalable, lightweight Bayesian LoRA by inferring a distribution over the elements of $\mathbf{E}$ while learning $\mathbf{B}$ and $\mathbf{A}$ deterministically, making data-dependent Bayesianization of LoRA fine-tuning scalable.

3.2 Stochastic LoRA Parameterization with Data Dependence

In conventional Bayesian neural networks, the focus is on epistemic uncertainties introduced by treating model parameters as random variables drawn from a fixed distribution—one that does not depend on the input data. Consequently, while the sampled parameters may vary across data instances, their underlying distribution remains invariant across all samples in the training set. This assumption limits the expressiveness of uncertainty estimates, particularly in few-shot LoRA fine-tuning scenarios. To address this, we propose a data-dependent, or *contextual*, Bayesian fine-tuning paradigm, wherein the distribution of the parameters of low-rank adapters depends on the inputs $\mathbf{x}_i$ for each data sample $(\mathbf{x}_i, y_i)$. Although one could perform Bayesian inference over $\mathbf{B}$ and $\mathbf{A}$ LoRA adapters from (1), this scales linearly with the frozen weight dimension d. Instead, by leveraging the lightweight LoRA factorization from (4), we learn the low-dimensional $\mathbf{E}$ matrix contextually, yielding the weight updates as

$$\Delta\mathbf{W} = \mathbf{B}\mathbf{E}_{\mathbf{x}}\mathbf{A} \tag{5}$$

where $\mathbf{E}_{\mathbf{x}}$ denotes the data-dependent version of $\mathbf{E}$. To be specific, we take an input-dependent Gaussian variational posterior over $\mathbf{E}_{\mathbf{x}_i}$, $q_{\boldsymbol{\phi}}(\mathbf{E}_{\mathbf{x}_i}|\mathbf{x}_i) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{E}}(\mathbf{x}_i), \boldsymbol{\Omega}^2_{\mathbf{E}}(\mathbf{x}_i))$. We learn these distribution parameters via lightweight per-layer auxiliary contextual modules comprising of small neural networks whose parameters across modules are collectively denoted as $\boldsymbol{\varphi}$. Such an input-dependent variational posterior resembles amortized inference in Bayesian modeling where $q_{\boldsymbol{\phi}}$ serves as an inference network that approximates $p(\mathbf{E}_{\mathbf{x}_i}|y_i, \mathbf{x}_i) \propto p(y_i|\mathbf{x}_i, \mathbf{E}_{\mathbf{x}_i})p(\mathbf{E}_{\mathbf{x}_i})$.

Particularly, consider a pre-trained LLM with L layers, each augmented by its own LoRA adapters $\mathbf{B}^l, \mathbf{E}^l_{\mathbf{x}}$, and $\mathbf{A}^l$. Let $\mathbf{x}_i^{l-1}$ denote the output of layer $l-1$. We then model $\mathbf{E}^l_{\mathbf{x}_i}$ autoregressively by conditioning on $\{\mathbf{E}^j_{\mathbf{x}_i}\}_{j<l}$, leading to $q_\phi(\mathbf{E}_{\mathbf{x}_i}|\mathbf{x}_i) = \prod_{l=1}^{L} q_\phi(\mathbf{E}^l_{\mathbf{x}_i}|\mathbf{x}_i^{l-1})$. Especially, in layer l, given $\mathbf{z}^l = \mathbf{A}^l\mathbf{x}^{l-1} \in \mathbb{R}^r$ as input, the corresponding contextual module denoted by h^l_φ produces the parameters $(\boldsymbol{\mu}^l_{\mathbf{E}}, \boldsymbol{\Omega}^l_{\mathbf{E}})$ of the $\mathbf{E}^l_{\mathbf{x}}$ distribution as output. We then draw $\mathbf{E}^l_{\mathbf{x}}$ conditioned on $\boldsymbol{\mu}^l_{\mathbf{E}}$ and $\boldsymbol{\Omega}^l_{\mathbf{E}}$. Finally, multiplying $\mathbf{B}^l$, $\mathbf{E}^l_{\mathbf{x}}$, and $\mathbf{z}^l$ and adding it to $\mathbf{W}^l_0\,\mathbf{x}^{l-1}$ yields the output of the layer l, $\mathbf{x}^l$.

This formulation enables us to focus exclusively on *heteroscedastic uncertainty*—modeling the variability in y_i given $\mathbf{x}_i$—with the epistemic uncertainty modeling available via imposing prior on $\boldsymbol{\varphi}$ or Bayesianizing $\mathbf{A}$ or $\mathbf{B}$ when needed, thereby isolating and highlighting the advantages of data-dependent (heterscedetic) UQ.

Contextual module parameterization. We parameterize each layer's contextual module with a two fully-connected layer neural network whose parameters at layer l are denoted as φ^l. In particular, each of these neural networks have r inputs, C hidden units, and $2 \times r^2$ outputs with the nonlinear ReLU activation function connecting the two fully-connected layers.

3.3 Amortized Inference for Contextual LoRA

In our formulation, the variational distribution is only conditioned on $\mathbf{x}_i$, while y_i contributes through the training objective. We then learn the model parameters, $\boldsymbol{\phi}$, by maximizing the ELBO of $\sum_i \log p(y_i|\mathbf{x}_i) = \sum_i \log \int p(y_i|\mathbf{x}_i, \mathbf{E}_{\mathbf{x}_i})p(\mathbf{E}_{\mathbf{x}_i})d\mathbf{E}_{\mathbf{x}_i}$ given $q_{\boldsymbol{\phi}}(\mathbf{E}_{\mathbf{x}_i}|\mathbf{x}_i)$:

$$\mathcal{L} = \sum_{i=1}^{N} \Big[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{E}_{\mathbf{x}_i}|\mathbf{x}_i)} \log p_{\boldsymbol{\theta}}(y_i|\mathbf{x}_i, \mathbf{E}_{\mathbf{x}_i}) - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{E}_{\mathbf{x}_i}|\mathbf{x}_i) \,\|\, p(\mathbf{E}_{\mathbf{x}_i}))\Big]. \tag{6}$$

Here, $\boldsymbol{\theta}$ denotes the parameters of all $\mathbf{B}$ and $\mathbf{A}$ adapters across layers. Hence, $\boldsymbol{\phi} = \{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ denotes all model parameters. Unlike standard variational inference as in (3), our formulation makes the distribution of $\mathbf{E}_{\mathbf{x}_i}$ input-dependent and replaces the single KL term with a sum of N per-sample KL divergences, whose combined impact scales with N. With the objective defined in (6), we adopt a simple and fixed Gaussian prior $p(\mathbf{E}_{\mathbf{x}})$ for learning $\mathbf{E}_{\mathbf{x}}$ in each layer. The complete learning objective can be equivalently expressed as a summation over all samples: $\mathcal{L} = \sum_{(\mathbf{x},y)\in\mathcal{D}} \mathcal{L}(\mathbf{x}, y)$. Then, we learn the deterministic parameters $\mathbf{B}$ and $\mathbf{A}$ together represented by $\boldsymbol{\theta}$ using the expected negative log-likelihood (NLL) term while excluding the KL term. This ensures that learning of $\boldsymbol{\theta}$ remains supervised, yielding less noisy gradients per sample computed as

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}, y) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{E}_{\mathbf{x}}|\mathbf{x})} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(y|\mathbf{x}, \mathbf{E}_{\mathbf{x}}). \tag{7}$$

We approximate this expectation with a single Monte Carlo draw $\mathbf{E}_{\mathbf{x}} \sim q_{\boldsymbol{\phi}}(\mathbf{E}_{\mathbf{x}}|\mathbf{x})$ for each $\mathbf{x}$ sample.

Next, to update the auxiliary network parameters $\boldsymbol{\varphi}$, which appear in both the NLL and KL terms of (6), we apply the reparameterization trick [52] to handle random sampling of $\mathbf{E}_{\mathbf{x}}$. Concretely, in

each layer, $\mathbf{E}_{\mathbf{x}}^l \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{E}}^l, {\boldsymbol{\Omega}^l}_{\mathbf{E}}^2)$ is reparameterized as $\mathbf{E}_{\mathbf{x}}^l = \boldsymbol{\mu}_{\mathbf{E}}^l + \boldsymbol{\Omega}_{\mathbf{E}}^l \odot \mathcal{E}^l$, where $\mathcal{E}^l \in \mathbb{R}^{r \times r}$ is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Thus, sampling $\mathbf{E}_{\mathbf{x}}^l$ from $q_\phi(\mathbf{E}_{\mathbf{x}}|\mathbf{x})$ is equivalent to evaluating a deterministic mapping $g_\phi(\mathcal{E}, \mathbf{x})$, enabling gradient computation via

$$\nabla_{\boldsymbol{\varphi}} \mathcal{L}(\mathbf{x}, y) = \mathbb{E}_{\mathcal{E}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \left[\nabla_{\boldsymbol{\varphi}} \left(\log p_{\boldsymbol{\theta}}(y|\mathbf{x}, g_{\boldsymbol{\phi}}(\mathcal{E}, \mathbf{x})) - \log \frac{q_{\boldsymbol{\phi}}(g_{\boldsymbol{\phi}}(\mathcal{E}, \mathbf{x})|\mathbf{x})}{p(g_{\boldsymbol{\phi}}(\mathcal{E}, \mathbf{x}))} \right) \right]. \tag{8}$$

3.4 Posterior Predictive Inference and Model Complexity

Posterior predictive inference and uncertainty estimation. Once C-LoRA training has converged, we obtain point estimate of the fine-tuned model by replacing $\mathbf{E}_{\mathbf{x}}$ with its variational mean and computing $p(y|\mathbf{x}, \boldsymbol{\mu}_{\mathbf{E}}(\mathbf{x}))$. As shown in Section 4, we denote these point estimate models by setting M in our model names. Next, to demonstrate the quality of uncertainty estimation, we sample $\mathbf{E}_{\mathbf{x}}$ from its inferred distribution M times and, similar to (2), approximate the posterior predictive distribution using Monte Carlo sampling, yielding $p(y_{\text{test}}|\mathbf{x}_{\text{test}}, \mathcal{D}) = 1/M \sum_{m=1}^{M} p(y|\mathbf{x}, \mathbf{E}_{\mathbf{x}}^m)$, where $\mathbf{E}_{\mathbf{x}}^m \sim q_{\boldsymbol{\phi}}(\mathbf{E}_{\mathbf{x}}|\mathbf{x})$. For example, for these calibrated models with $M = 10$ drawn samples in Section 4, we label them with "M=10" in our model names. When "M=0", the posterior mean is used directly for evaluation.

Reducing contextual module complexity through feature reuse. In layer l, the contextual module is designed to model $q_{\boldsymbol{\phi}}(\mathbf{E}_{\mathbf{x}}^l|\mathbf{x}^{l-1})$ where $\mathbf{x}^l$ is derived from natural language; hence, learning features from scratch becomes both non-trivial and computationally expensive. To mitigate this issue, we follow [39] and take advantage of the main model by feeding $\mathbf{z}^l$ into the contextual module instead. Additional computational cost of $\mathcal{O}(r^4)$, stems from the auxiliary network h_φ^l, whose fully connected layers have size $C = \mathcal{O}(r^2) \ll d$. This overhead is minimal compared to the main model's per-layer operation cost of $\mathcal{O}(d^2)$. As a result, we can efficiently estimate uncertainty at the sample level in a low-dimensional space, sidestepping the costly feature-learning burden within the contextual modules during fine-tuning.

4 Experiments

In this section, we conduct comprehensive experiments to demonstrate the effectiveness of our proposed C-LoRA approach for uncertainty quantification in LLMs on reasoning datasets from various domains. We first specify the experimental settings in Section 4.1, including baselines, fine-tuning setting, and evaluation metrics. We then benchmark C-LoRA against existing uncertainty quantification methods across six reasoning datasets to evaluate overall task performance and uncertainty quality in Section 4.2. Additionally, we test the robustness of each method under distribution shift by fine-tuning on OBQA and evaluating on related OOD datasets in Section 4.3. Lastly, we perform an ablation to demonstrate the role of our auxiliary contextual module in Section 4.4.

4.1 Experimental Settings

Baselines. We provide performance comparisons of **C-LoRA** with well-established cutting-edge baselines that include Deep Ensemble [49, 51, 53], Monte-Carlo dropout (**MCD**) [29], Bayesian LoRA with Backpropagation (**BLoB**) [20], a recent mean-field VI approach applied to LoRA parameters; Laplace-LoRA (**LA**) [19], which uses a Laplace approximation over adapter weights; and *Maximum A Posteriori* (**MAP**), representing a deterministic point-estimate baseline.

LLM fine-tuning settings. Following prior work, we evaluate the performance of **C-LoRA**[1], by fine-tuning **LLaMA2-7B**[6] using the `PEFT` library[54] across six commonsense reasoning benchmarks, including both binary and multiple-choice classification tasks. For each task, the appropriate next-token logits are selected based on the format of the target output, which we include in the Appendix A, and the model is trained to maximize the log-likelihood of the correct answer. Following previous works [20, 21, 28], we apply LoRA to the query, value, and output projection layers using default hyperparameters. All models are fine-tuned with a batch size of 4. Baseline methods are trained for 5000 iterations, while **C-LoRA** is trained for 1500 iterations on all datasets, except for Winogrande-medium (WG-M), which is trained for 2000 iterations. To ensure consistent evaluation, we curate a training-validation split by reserving 80% of the original training set for fine-tuning and holding

[1]Using contextual modules comprising of two fully connected layers with 64 hidden units (C=64) each.

out the remaining 20% as a validation set. This validation split is used for early stopping and checkpointing based on a metric combining validation accuracy and calibration quality (see Appendix F).

Evaluation metrics. We report results using three key evaluation metrics: accuracy (ACC), expected calibration error (ECE), and negative log-likelihood (NLL). While ACC reflects raw predictive performance, ECE and NLL are widely adopted metrics for assessing the quality of uncertainty estimates. ECE measures the alignment between predicted confidence and actual correctness, while NLL evaluates the sharpness and correctness of the model's predicted probability distribution. All reported results are averaged over three independent runs with different random seeds, and we report the mean $\pm$ standard deviation for each metric.

4.2 Evaluation Results under In-distribution Fine-tuning

In this section, we conduct our assessment of fine-tuning performances under the in-distribution scenario for six common-sense reasoning tasks. These tasks encompass: Winogrande-small (`WG-S`), Winogrande-medium (`WG-M`) [55], ARC-Challenge (`ARC-C`), ARC-Easy (`ARC-E`) [56], OpenBookQA (`OBQA`)[57], and `BoolQ` [58]. For all the experiments, the same pre-trained LLaMA2-7B was used as the LLM backbone. For BLoB, we kept the same setting as in [20].

Accuracy and Uncertainty Quantification Table 1 presents the performance of various uncertainty quantification methods applied to LoRA-tuned LLaMA2-7B across six common-sense reasoning datasets. As a complement, Figure 2 in Appendix K visualizes the results with bar plots. In this set of experiments, our proposed method, C-LoRA, as well as the baseline BLoB, are evaluated under both deterministic (M=0) and stochastic (M=10) posterior sampling settings. While C-LoRA, does not achieve the highest accuracy, it maintains competitive performance across all datasets, with accuracies that are typically within 1–2% of the best methods (e.g., `ARC-C`: 67.79%, `OBQA`: 78.26%). Notably, C-LoRA (M=0)—which uses the posterior mean without sampling—performs similarly or slightly better than the sampled version on some tasks (e.g., `ARC-C`: 69.02% vs. 67.79%), suggesting that the contextual posterior mean alone offers a robust deterministic approximation.

The strength of C-LoRA becomes more apparent when examining uncertainty calibration, as measured by expected calibration error (ECE). Here, C-LoRA (M=10) consistently achieves the lowest or second-lowest ECE across all datasets, including `WG-S` (6.86%), `ARC-C` (8.83%), `ARC-E` (4.27%), `WG-M` (3.71%), and `BoolQ` (1.62%). These values are significantly better than those obtained by MAP, MCD, and even Deep Ensemble, indicating that C-LoRA produces confidence estimates that are closely aligned with actual correctness. Even the deterministic variant, C-LoRA (M=0), outperforms BLoB (M=10) in calibration on datasets such as `WG-S` (7.16% vs. 11.23%) and performs comparably on `BoolQ` (1.46% vs. 1.46%). A similar pattern holds for negative log-likelihood (NLL), which reflects the quality of probabilistic predictions. C-LoRA (M=10) achieves among the lowest NLL values in nearly all cases, such as 0.63 on `WG-S`, and 0.88 on `ARC-C`, often outperforming both BLoB (M=10) and Deep Ensemble. Additionally, C-LoRA (M=0) matches or slightly outperforms BLoB (M=10) in NLL on several tasks (e.g., `ARC-C`: 0.89 vs. 0.88), while avoiding the computational overhead of posterior sampling. In summary, while C-LoRA trades off a small amount of accuracy compared to non-Bayesian methods, it delivers substantial improvements in calibration and likelihood, which are essential for uncertainty-aware applications.

Post-hoc Calibration with Temperature Scaling To further evaluate calibration, we apply *post-hoc* temperature scaling [59] to both BLoB and C-LoRA and report the resulting ECE on six common-sense reasoning tasks in Table 2. Bar plots of these results are presented in Figure 3 in Appendix K. Even before temperature scaling, C-LoRA (M=10) achieves stronger calibration than BLoB on nearly every dataset. After applying temperature scaling, C-LoRA (M=0, tmp) achieves the best or second-best ECE in 4 of the 6 datasets, including `WG-S` (4.00%), `ARC-E` (3.86%), `ARC-C` (6.30%), and `WG-M` (2.90%). Notably, it outperforms BLoB (M=10, tmp) in 5 out of the 6 datasets, despite using no posterior sampling. Interestingly, C-LoRA (M=10, tmp) does not always show improvement over its uncalibrated variant. On datasets like `OBQA` and `BoolQ`, temperature scaling can actually worsen ECE (e.g., from 4.00% to 6.27% on `OBQA`). This may be due to the fact that posterior sampling already flattens the predictive distribution, and further scaling can overcorrect, leading to underconfident predictions. Also, temperature scaling minimizes NLL, a proper scoring rule, but it does not directly optimize ECE, which is a binned, non-differentiable, and improper

metric. Consequently, if a model is already reasonably well-calibrated or if its logit distribution exhibits certain local structures, temperature scaling may shift predictions in ways that degrade ECE in specific bins. Similar observations have been reported in prior work [60], where temperature scaling was found to degrade classwise ECE. These results suggest that C-LoRA (M=0) provides an effective balance between calibration quality and inference efficiency.

Table 1: Performance comparison of different methods applied to LoRA on LLAMA2-7B weights accross six common sense reasoning tasks with a validation set split from the training dataset. The best and the second best performances are specified in **boldface** and via underline respectively.

Metric	Method	Datasets					
		WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ
ACC ↑	MAP	69.37±1.04	67.67±1.18	85.20±0.63	74.57±0.73	81.60±0.40	87.68±0.02
	MCD	69.06±1.40	66.66±2.30	85.49±0.74	75.89±0.48	81.46±0.92	87.67±0.08
	Deep Ensemble	68.98±0.97	68.57±2.11	86.24±1.26	77.39±1.08	82.20±0.91	88.07±0.17
	LA	68.18±1.04	64.17±0.97	85.30±0.97	74.15±0.40	77.53±0.80	86.45±0.35
	BLoB (M=0)	69.95±0.95	69.25±0.33	85.79±0.83	75.52±0.36	81.85±0.53	86.63±0.52
	BLoB (M=10)	66.55±0.61	66.66±2.25	84.56±0.20	73.38±0.29	81.44±0.53	86.63±0.50
	C-LoRA (M=0)	67.16±0.27	69.02±1.03	84.74±0.20	72.09±2.70	81.13±1.00	85.84±0.67
	C-LoRA (M=10)	66.21±1.24	67.79±1.27	84.38±0.67	70.48±1.71	78.26±2.61	84.64±0.81
ECE ↓	MAP	29.76±1.08	30.60±1.26	13.49±0.63	23.01±0.44	15.30±0.11	5.93±0.36
	MCD	28.49±1.60	29.60±2.77	12.69±0.60	20.73±0.38	14.34±1.11	5.13±0.25
	Deep Ensemble	28.72±1.46	27.75±1.86	11.87±0.16	18.67±0.29	13.98±1.12	5.24±0.27
	LA	11.41±0.17	30.54±0.70	45.85±2.08	10.80±0.38	35.65±1.14	18.22±0.41
	BLoB (M=0)	21.22±1.67	22.57±1.24	10.13±0.39	12.35±0.86	9.35±1.08	2.90±0.18
	BLoB (M=10)	11.23±1.45	10.77±1.91	4.29±1.08	4.52±0.91	**3.82**±0.96	**1.46**±0.36
	C-LoRA (M=0)	7.16±2.92	12.28±1.01	5.75±1.00	6.07±4.85	5.10±1.36	**1.46**±0.85
	C-LoRA (M=10)	**6.86**±3.99	**8.83**±1.20	**4.27**±1.24	**3.71**±1.30	4.00±0.84	1.62±0.44
NLL ↓	MAP	2.86±0.23	3.07±0.09	1.13±0.10	1.26±0.12	1.04±0.02	0.34±0.00
	MCD	2.50±0.12	2.81±0.25	1.13±0.04	1.16±0.03	1.01±0.07	0.32±0.00
	Deep Ensemble	2.44±0.23	2.20±0.03	0.91±0.05	1.04±0.09	0.87±0.03	0.32±0.00
	LA	**0.62**±0.00	1.17±0.01	0.97±0.05	0.56±0.00	0.98±0.01	0.45±0.00
	BLoB (M=0)	0.90±0.07	1.34±0.05	0.63±0.02	0.61±0.03	0.59±0.02	**0.31**±0.00
	BLoB (M=10)	0.66±0.01	**0.88**±0.03	**0.44**±0.00	**0.54**±0.00	**0.51**±0.01	**0.31**±0.01
	C-LoRA (M=0)	0.64±0.03	0.89±0.09	0.46±0.01	0.58±0.03	0.53±0.01	0.34±0.01
	C-LoRA (M=10)	0.63±0.02	**0.88**±**0.00**	0.48±0.02	0.57±0.03	0.59±0.05	0.35±0.02

Table 2: Performance comparison of uncertainty quantification based on the ECE metric with temperature scaling using validation sets over six common-sense reasoning tasks.

Metric	Method	Datasets					
		WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ
ECE ↓	BLoB (M=0)	21.22±1.67	22.57±1.24	10.13±0.39	12.35±0.86	9.35±1.08	2.90±0.18
	BLoB (M=0, tmp)	13.54±1.19	15.47±0.73	6.16±0.40	5.76±0.40	4.56±0.95	1.89±0.37
	BLoB (M=10)	11.23±1.45	10.77±1.91	4.29±1.08	4.52±0.91	**3.82**±0.96	**1.46**±0.36
	BLoB (M=10, tmp)	5.10±1.47	6.87±1.91	5.38±1.08	3.18±0.62	5.71±0.80	3.40±0.67
	C-LoRA (M=0)	7.16±2.92	12.28±1.01	5.75±1.00	6.07±4.85	5.10±1.36	**1.46**±0.85
	C-LoRA (M=0, tmp)	**4.00**±1.37	6.58±0.76	**3.86**±0.17	**2.90**±0.12	4.90±2.16	1.78±0.56
	C-LoRA (M=10)	6.86±3.99	8.83±1.20	4.27±1.24	3.71±1.30	4.00±0.84	1.62±0.44
	C-LoRA (M=10, tmp)	5.18±1.54	**6.30**±0.87	4.35±1.35	3.87±1.77	6.27±1.01	2.67±0.29

4.3 Robustness Under Distribution Shift

Table 3 presents model performance under both in-distribution (`OBQA`) and out-of-distribution (OOD) settings with smaller (`ARC-E`, `ARC-C`) and larger (`Chem`, `Phy`) distribution shifts. Figure 4 in Appendix K illustrates bar plots of the results. While C-LoRA does not achieve the highest accuracy under shift, it demonstrates strong robustness in uncertainty estimation, especially in calibration (ECE) and likelihood (NLL). In terms of accuracy, Deep Ensemble and BLoB generally maintain higher predictive performance across OOD datasets. However, C-LoRA (M=10) remains competitive, achieving 65.09% accuracy on `ARC-C` and 41.00% on `Chem`—nearly matching the highest-performing methods. While C-LoRA (M=10) sees a slight drop in accuracy under severe shift (e.g., `Phy`: 31.00%),

Table 3: Performance comparison on out-of-distribution datasets. The following results are evaluated using LLaMA2-7B fine-tuned on the OBQA dataset in Table 1.

Metric	Method	Datasets				
		In-Dist.	*Smaller Dist. Shift*		*Larger Dist. Shift*	
		OBQA	ARC-C	ARC-E	Chem	Phy
ACC ↑	MAP	$81.60_{\pm 0.40}$	$67.67_{\pm 1.52}$	$73.88_{\pm 0.27}$	$41.00_{\pm 3.60}$	$33.00_{\pm 5.29}$
	MCD	$81.46_{\pm 0.92}$	$68.69_{\pm 0.85}$	$75.00_{\pm 2.13}$	$39.00_{\pm 3.00}$	$31.00_{\pm 6.24}$
	Deep Ensemble	$82.20_{\pm 0.91}$	$68.01_{\pm 1.28}$	$74.05_{\pm 0.50}$	$42.33_{\pm 3.51}$	$27.66_{\pm 2.51}$
	BLoB (M=0)	$81.85_{\pm 0.53}$	$69.48_{\pm 0.78}$	$76.93_{\pm 1.33}$	$45.33_{\pm 1.15}$	$26.66_{\pm 4.61}$
	BLoB (M=10)	$81.44_{\pm 0.53}$	$67.22_{\pm 1.17}$	$75.11_{\pm 1.25}$	$44.00_{\pm 0.00}$	$33.66_{\pm 2.08}$
	C-LoRA (M=0)	$81.13_{\pm 1.00}$	$66.66_{\pm 0.78}$	$75.99_{\pm 1.63}$	$40.00_{\pm 1.73}$	$28.00_{\pm 1.73}$
	C-LoRA (M=10)	$78.26_{\pm 2.61}$	$65.09_{\pm 4.68}$	$74.29_{\pm 2.90}$	$41.00_{\pm 2.64}$	$31.00_{\pm 1.00}$
ECE ↓	MAP	$15.30_{\pm 0.11}$	$25.54_{\pm 1.25}$	$20.09_{\pm 0.53}$	$29.73_{\pm 0.30}$	$36.22_{\pm 3.60}$
	MCD	$14.34_{\pm 1.11}$	$23.41_{\pm 0.74}$	$18.36_{\pm 1.03}$	$28.67_{\pm 0.77}$	$36.53_{\pm 4.30}$
	Deep Ensemble	$13.98_{\pm 1.12}$	$20.90_{\pm 1.05}$	$16.89_{\pm 0.80}$	$16.10_{\pm 2.22}$	$26.74_{\pm 3.23}$
	BLoB (M=0)	$9.35_{\pm 1.08}$	$15.76_{\pm 1.42}$	$11.70_{\pm 0.62}$	$14.12_{\pm 4.05}$	$27.35_{\pm 4.01}$
	BLoB (M=10)	$3.82_{\pm 0.96}$	$\underline{9.77}_{\pm 0.91}$	$\mathbf{5.74}_{\pm 0.91}$	$\underline{12.63}_{\pm 0.11}$	$\mathbf{17.56}_{\pm 2.81}$
	C-LoRA (M=0)	$5.10_{\pm 1.36}$	$10.08_{\pm 3.30}$	$\underline{6.42}_{\pm 2.48}$	$15.42_{\pm 2.99}$	$24.42_{\pm 6.02}$
	C-LoRA (M=10)	$4.00_{\pm 0.84}$	$\mathbf{8.83}_{\pm 1.44}$	$6.68_{\pm 0.71}$	$\mathbf{12.49}_{\pm 1.18}$	$\underline{18.16}_{\pm 1.52}$
NLL ↓	MAP	$1.04_{\pm 0.02}$	$1.66_{\pm 0.12}$	$1.37_{\pm 0.04}$	$1.81_{\pm 0.03}$	$1.86_{\pm 0.04}$
	MCD	$1.01_{\pm 0.07}$	$1.58_{\pm 0.01}$	$1.24_{\pm 0.03}$	$1.81_{\pm 0.03}$	$1.88_{\pm 0.10}$
	Deep Ensemble	$0.87_{\pm 0.03}$	$1.15_{\pm 0.09}$	$0.94_{\pm 0.06}$	$1.45_{\pm 0.01}$	$1.58_{\pm 0.09}$
	BLoB (M=0)	$0.59_{\pm 0.02}$	$0.95_{\pm 0.05}$	$0.72_{\pm 0.01}$	$1.41_{\pm 0.02}$	$1.57_{\pm 0.03}$
	BLoB (M=10)	$0.51_{\pm 0.01}$	$\mathbf{0.83}_{\pm 0.02}$	$\mathbf{0.63}_{\pm 0.01}$	$\underline{1.35}_{\pm 0.01}$	$\underline{1.46}_{\pm 0.01}$
	C-LoRA (M=0)	$0.53_{\pm 0.01}$	$\underline{0.85}_{\pm 0.04}$	$\underline{0.64}_{\pm 0.04}$	$1.43_{\pm 0.01}$	$1.54_{\pm 0.09}$
	C-LoRA (M=10)	$0.59_{\pm 0.05}$	$0.89_{\pm 0.09}$	$0.67_{\pm 0.02}$	$\mathbf{1.31}_{\pm 0.02}$	$\mathbf{1.42}_{\pm 0.01}$

it significantly outperforms all baselines in ECE and NLL, indicating more reliable and calibrated uncertainty estimates.

For calibration (ECE), C-LoRA (M=10) achieves the lowest error on `Chem` (12.49%) and `ARC-C` (8.83%). This trend also holds for `Phy`, where C-LoRA (M=10) yields 18.16% which is only slightly worse than BLoB and significantly better than all other baselines. This indicates that C-LoRA remains well-calibrated even as the input distribution diverges from training. C-LoRA also excels in negative log-likelihood (NLL), achieving the lowest or second-lowest values across nearly all OOD datasets. On `Chem` and `Phy`, for example, C-LoRA (M=10) obtains NLL of 1.31 and 1.42 respectively, both lower than Deep Ensemble (1.45 and 1.58) and BLoB (1.35 and 1.46) and significantly better than all other baselines. Moreover, it is noteworthy that C-LoRA (M=0), which uses only the posterior mean, also maintains competitive performance. For example, on `ARC-C` and `ARC-E`, C-LoRA (M=0) achieves 10.08% and 6.42% ECE respectively, compared to BLoB (M=10)'s 9.77% and 5.74%. This suggests that C-LoRA's contextual posterior mean already captures meaningful uncertainty, making it attractive for deployment in compute-constrained scenarios.

4.4 Ablation Study: Impact of Contextual Module

We also conduct the ablation study investigating the importance of the contextual module for overall performance. Since in C-LoRA we focus on contextualizing the distribution of matrix $\mathbf{E} \in \mathbb{R}^{r \times r}$, to examine the effect of the contextual module, we compare with the results of fine-tuning the model on different tasks using lightweight factorization introduced in Section 3.1, with mean-field variational inference without the contextual module; we refer to this setting as **FE** in Tables 4 and 5.

Table 4 underscores the importance of the contextual module by comparing FE and C-LoRA across the six common-sense reasoning tasks under in-distribution scenario. C-LoRA offers significant improvements with respect to ECE across almost all tasks for inference-time sample with both M=0 and M=10, except for `BoolQ` where it achieves the second best performance. Also, C-LoRA gives the lowest NLL across all tasks.

Furthermore, we assess the effect of the contextual module on generalization. Table 5 provides the performance comparison of the corresponding models fine-tuned on `OBQA` in Table 4 under similar distribution shifts as in Table 3. C-LoRA outperforms FE across all datasets in both smaller and larger distribution shifts, with respect to ECE. In addition, C-LoRA offers the lowest NLL on all tasks in both

distribution shifts. With respect to accuracy, in all tasks C-LoRA achieves a comparable performance, except for `Chem` where it has a subpar performance; however, this shortcoming is a result of the tradeoff between a generalized model with a proper uncertainty quantification, and an overconfident model with a poor uncertainty estimation. In summary, these results indicate the significance of our auxiliary contextual module, particularly on enhancing the uncertainty quantification and generalization while maintaining competitive predictive performance.

Table 4: Impact of the contextual module on performance. The performance comparison is conducted across six common-sense reasoning tasks with a validation set split from the training dataset.

Metric	Method	Datasets					
		WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ
ACC ↑	FE (M=0)	65.45±1.36	68.35±1.02	85.32±0.39	73.47±2.36	80.46±0.11	84.70±0.45
	FE (M=10)	65.06±1.33	68.20±2.40	85.08±0.36	72.43±1.28	80.93±0.50	84.65±0.52
	C-LoRA (M=0)	67.16±0.27	69.02±1.03	84.74±0.20	72.09±2.70	81.13±1.00	85.84±0.67
	C-LoRA (M=10)	66.21±1.24	67.79±1.27	84.38±0.67	70.48±1.71	78.26±2.61	84.64±0.81
ECE ↓	FE (M=0)	23.76±1.64	27.26±0.90	12.23±0.25	14.90±1.98	11.23±0.35	2.57±0.13
	FE (M=10)	18.12±1.52	19.60±2.42	9.54±0.47	12.21±1.87	8.42±0.50	**1.34**±0.22
	C-LoRA (M=0)	7.16±2.92	12.28±1.01	5.75±1.00	6.07±4.85	5.10±1.36	1.46±0.85
	C-LoRA (M=10)	**6.86**±3.99	**8.83**±1.20	**4.27**±1.24	**3.71**±1.30	**4.00**±0.84	1.62±0.44
NLL ↓	FE (M=0)	0.89±0.07	1.85±0.11	0.85±0.05	0.69±0.03	0.68±0.04	**0.34**±0.00
	FE (M=10)	0.78±0.04	1.35±0.12	0.65±0.02	0.63±0.02	0.60±0.03	**0.34**±0.00
	C-LoRA (M=0)	0.64±0.03	0.89±0.09	**0.46**±0.01	0.58±0.03	**0.53**±0.01	**0.34**±0.01
	C-LoRA (M=10)	**0.63**±0.02	**0.88**±0.00	0.48±0.02	**0.57**±0.03	0.59±0.05	0.35±0.02

Table 5: Impact of the contextual module on generalization. The performance comparison is conducted on out-of-distribution datasets using LLaMA2-7B fine-tuned on OBQA dataset in Table 4.

Metric	Method	Datasets				
		In-Dist.	*Smaller Dist. Shift*		*Larger Dist. Shift*	
		OBQA	ARC-C	ARC-E	Chem	Phy
ACC ↑	FE (M=0)	80.46±0.11	68.91±1.22	74.72±0.63	45.00±1.73	31.66±3.51
	FE (M=10)	80.93±0.50	66.65±1.09	74.76±0.70	46.00±1.73	32.33±1.52
	C-LoRA (M=0)	81.13±1.00	66.66±0.78	75.99±1.63	40.00±1.73	28.00±1.73
	C-LoRA (M=10)	78.26±2.61	65.09±4.68	74.29±2.90	41.00±2.64	31.00±1.00
ECE ↓	FE (M=0)	11.23±0.35	18.44±1.51	14.65±0.49	20.02±2.26	31.75±3.20
	FE (M=10)	8.42±0.50	15.78±1.11	11.69±0.68	17.61±1.25	26.02±1.32
	C-LoRA (M=0)	5.10±1.36	10.08±3.30	**6.42**±2.48	15.42±2.99	24.42±6.02
	C-LoRA (M=10)	4.00±0.84	**8.83**±1.44	6.68±0.71	**12.49**±1.18	**18.16**±1.52
NLL ↓	FE (M=0)	0.68±0.04	1.03±0.01	0.87±0.01	1.53±0.05	1.77±0.03
	FE (M=10)	0.60±0.03	0.95±0.02	0.78±0.01	1.46±0.03	1.67±0.02
	C-LoRA (M=0)	0.53±0.01	**0.85**±0.04	**0.64**±0.04	1.43±0.01	1.54±0.09
	C-LoRA (M=10)	0.59±0.05	0.89±0.09	0.67±0.02	**1.31**±0.02	**1.42**±0.01

5 Related Works

LLMs have demonstrated remarkable successes across a wide range of applications, including code generation, scientific reasoning, and open-domain question answering; however, they are also notorious for *hallucination*—a phenomenon in which the model generates fluent but factually incorrect or unsupported content that is not grounded in the input or external knowledge. To mitigate hallucination, a variety of strategies have been proposed, including Reinforcement Learning from Human Feedback (RLHF) to align model outputs with human preferences [61, 62], and Retrieval-Augmented Generation (RAG) to ground responses in external knowledge [63, 64]. Additionally, probabilistic methods such as BNNs [19, 20] and conformal prediction [65, 66] have been demonstrated to be able to quantify uncertainty and flag unreliable outputs. Recent work also shows the effectiveness of prompt-level interventions [67, 68] by carefully crafting instructions and uncertainty-aware prompts to reduce hallucinated content without altering the model architecture.

Our work connects to previous efforts in BNNs by introducing a Bayesian formulation over the LoRA layers; however, we extend this line of work by using a data-dependent lightweight low-rank

factorization, where the intermediate matrix $\mathbf{E}$ is modeled as random variables, enabling flexible and context-aware uncertainty estimation to mitigate overconfidence and hallucination in LLMs.

6 Conclusion and Discussion

In this work, we introduced C-LoRA, a novel uncertainty-aware parameter-efficient approach for fine-tuning LLMs in an end-to-end Bayesian framework. C-LoRA facilitates modeling the aleatoric uncertainty (data uncertainty) via an efficient and scalable contextual module, without compromising the potential for estimating the model uncertainty. Through extensive empirical studies, we demonstrate the superior performance of our method in achieving outstanding accuracy and strong uncertainty quantification capabilities, with consistent generalization across diverse data distributions. Our method underscores the significance of modeling the aleatoric uncertainty in low-data regimes that can lead to substantial gains in generalization and trustworthiness of LLMs.

While C-LoRA establishes a promising foundation for uncertainty-aware parameter-efficient fine-tuning, several avenues remain open for future exploration. One interesting direction is extending the framework to multi-modal or multi-task settings, where capturing cross-modal or task-dependent uncertainty could further enhance robustness and generalization. Another promising avenue involves integrating C-LoRA with active learning or adaptive data acquisition strategies, leveraging its uncertainty estimates to guide sample-efficient model updates. Moreover, exploring hierarchical or structured priors within the Bayesian formulation may offer richer representations of both epistemic and aleatoric uncertainty.

7 Limitations

In this study, we restricted our experiments to fine-tuning LLaMA2-7B models, which limits our understanding of how C-LoRA scales to larger architectures. Investigating the scaling behavior of C-LoRA on larger models remains an open challenge. Furthermore, evaluating uncertainty in language models remains challenging due to the lack of standardized benchmarks with ground-truth uncertainty labels. While our evaluation—based on held-out test sets and established metrics like ECE and NLL—provides meaningful insight into model calibration and predictive confidence, these metrics offer only an indirect view of uncertainty quality. Developing task-specific benchmarks or human-in-the-loop protocols for evaluating uncertainty in NLP would be a valuable direction. Moreover, while C-LoRA is motivated by data-dependent/aleatoric uncertainty, this work does not provide a theoretical guarantee that it can disentangle aleatoric and epistemic uncertainty, which we leave to future investigation.

We also note that, as shown throughout our experiments, C-LoRA overall provides a more calibrated predictions and better uncertainty estimation at the expense of marginally reduced predictive performance which can be preferable in many high-stakes applications. However, in other domains such as spam detection where accuracy is more critical, this trade-off may not be desirable. Hence we acknowledge that this trade-off is task dependent and should be considered before deploying C-LoRA (and in general any uncertainty-aware method).

Finally, our approach employs amortized variational inference to approximate the posterior distribution over the LoRA adapter parameters, providing a scalable and principled Bayesian treatment suitable for large language models. This variational approximation, based on a Gaussian family and a reverse KL objective, is inherently mode-seeking and may underestimate posterior uncertainty or fail to capture multi-modality in the true posterior. Investigating richer variational families and more comprehensive posterior predictive analyses represents an important direction for future work.

8 Acknowledgment

A.H.R., W.Z., Y.W., and X.N.Q. acknowledge the support from United States National Science Foundation (NSF) grants DMREF-2119103, SHF-2215573, and IIS-2212419. S.J., B.J.Y., N.M.U., and X.N.Q. acknowledge the support from the United States Department of Energy's Office of Science Biological and Environmental Research (BER) program under project B&R# KP1601017 and FWP#CC140. Many of the numerical experiments were conducted using advanced computing resources provided by Texas A&M High Performance Research Computing.

References

[1] Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions, 2024.

[2] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2024.

[3] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. Accessed: 2024-11-15.

[4] Tom B. Brown et al. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[5] OpenAI et al. GPT-4 technical report, 2024.

[6] Hugo Touvron et al. LLaMA 2: Open foundation and fine-tuned chat models, 2023.

[7] Aaron Grattafiori et al. The LLaMA 3 herd of models, 2024.

[8] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21(8):1470–1480, Aug 2024.

[9] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), September 2022.

[10] Karan Singhal, Shekoofeh Azizi, Tao Tu, et al. Publisher correction: Large language models encode clinical knowledge. *Nature*, 620(7973):E19, 2023.

[11] Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Anthony B Costa, Mona G Flores, et al. A large language model for electronic health records. *NPJ Digit. Med.*, 5(1):194, December 2022.

[12] Sanket Jantre, Tianle Wang, Gilchan Park, Kriti Chopra, Nicholas Jeon, Xiaoning Qian, Nathan M Urban, and Byung-Jun Yoon. Uncertainty-aware adaptation of large language models for protein-protein interaction analysis. *arXiv preprint arXiv:2502.06173*, 2025.

[13] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nat. Mach. Intell.*, 6(5):525–535, May 2024.

[14] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science, 2022.

[15] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. FinGPT: Open-source financial large language models, 2023.

[16] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. BloombergGPT: A large language model for finance, 2023.

[17] David Thulke, Yingbo Gao, Petrus Pelser, et al. ClimateGPT: Towards AI synthesizing interdisciplinary research on climate change, 2024.

[18] Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code, 2021.

[19] Adam X. Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[20] Yibin Wang, Haizhou Shi, Ligong Han, Dimitris N. Metaxas, and Hao Wang. BLoB: Bayesian low-rank adaptation by backpropagation for large language models. In *Neural Information Processing Systems (NeurIPS)*, 2024.

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[22] Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning, 2024.

[23] Yaqing Wang, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Awadallah, and Jianfeng Gao. LiST: Lite prompted self-training makes parameter-efficient few-shot learners. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2262–2281. Association for Computational Linguistics, 2022.

[24] Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. Think twice before trusting: Self-detection for large language models through comprehensive answer reflection. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11858–11875, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[25] Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024.

[26] Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in LLMs: Reward calibration in RLHF. In *The Thirteenth International Conference on Learning Representations*, 2025.

[27] Guande He, Jianfei Chen, and Jun Zhu. Preserving pre-trained features helps calibrate fine-tuned language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Cristian Meo, Ksenia Sycheva, Anirudh Goyal, and Justin Dauwels. Bayesian-loRA: LoRA based parameter efficient fine-tuning using optimal quantization levels and rank values trough differentiable Bayesian gates. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ICML 2024)*, 2024.

[29] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

[30] Qingping Zhou, Tengchao Yu, Xiaoqun Zhang, and Jinglai Li. Bayesian inference and uncertainty quantification for medical image reconstruction with Poisson data. *SIAM Journal on Imaging Sciences*, 13(1):29–52, 2020.

[31] Mohammad Hossein Shaker and Eyke Hüllermeier. Ensemble-based uncertainty quantification: Bayesian versus Credal inference. In *Proceedings 31. workshop computational intelligence*. KIT Scientific Publishing, 2021.

[32] Ziyu Wang and Christopher C. Holmes. On uncertainty quantification for near-Bayes optimal algorithms. In *Sixth Symposium on Advances in Approximate Bayesian Inference - Non Archival Track*, 2024.

[33] Ethan Goan and Clinton Fookes. *Bayesian Neural Networks: An Introduction and Survey*, page 45–87. Springer International Publishing, 2020.

[34] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux - effortless Bayesian deep learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20089–20103. Curran Associates, Inc., 2021.

[35] Shujian Zhang, Xinjie Fan, Bo Chen, and Mingyuan Zhou. Bayesian attention belief networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12413–12426. PMLR, 18–24 Jul 2021.

[36] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020.

[37] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5436–5446. PMLR, 13–18 Jul 2020.

[38] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[39] Xinjie Fan, Shujian Zhang, Korawat Tanwisuth, Xiaoning Qian, and Mingyuan Zhou. Contextual dropout: An efficient sample-dependent dropout module. In *International Conference on Learning Representations*, 2021.

[40] Sanket Jantre, Shrijita Bhattacharya, Nathan M Urban, Byung-Jun Yoon, Tapabrata Maiti, Prasanna Balaprakash, and Sandeep Madireddy. Sequential Bayesian neural subnetwork ensembles. *arXiv:2206.00794*, 2022.

[41] Sanket Jantre, Nathan M. Urban, Xiaoning Qian, and Byung-Jun Yoon. Learning active subspaces for effective and scalable uncertainty quantification in deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[42] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe and. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[43] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(40):1303–1347, 2013.

[44] Ning Ding, Yujia Qin, Guang Yang, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, Mar 2023.

[45] Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[46] Zhengxiang Shi and Aldo Lipani. DePT: Decomposed prompt tuning for parameter-efficient fine-tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

[47] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*, 2024.

[48] Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J. Clark, and Mehdi Rezagholizadeh. KronA: Parameter-efficient tuning with Kronecker adapter. In Peyman Passban, Andy Way, and Mehdi Rezagholizadeh, editors, *Enhancing LLM Performance: Efficacy, Fine-Tuning, and Inference Techniques*, pages 49–65. Springer Nature Switzerland, 2025.

[49] Oleksandr Balabanov and Hampus Linander. Uncertainty quantification in fine-tuned LLMs using loRA ensembles. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*, 2025.

[50] Emre Onal, Klemens Flöge, Emma Caldwell, Arsen Sheverdin, and Vincent Fortuin. Gaussian stochastic weight averaging for Bayesian low-rank adaptation of large language models. In *Sixth Symposium on Advances in Approximate Bayesian Inference - Non Archival Track*, 2024.

[51] Xi Wang, Laurence Aitchison, and Maja Rudolph. LoRA ensembles for large language model fine-tuning, 2023.

[52] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.

[53] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[54] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

[55] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: an adversarial Winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021.

[56] 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, 2018.

[57] 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 Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[58] 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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics, 2019.

[59] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017.

[60] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[61] Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. Learning to trust your feelings: Leveraging self-awareness in LLMs for hallucination mitigation. In Wenhao Yu, Weijia Shi, Michihiro Yasunaga, Meng Jiang, Chenguang Zhu, Hannaneh Hajishirzi, Luke Zettlemoyer, and Zhihan Zhang, editors, *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, pages 44–58, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[62] Xueru Wen, Jie Lou, Xinyu Lu, Yuqiu Ji, Xinyan Guan, Yaojie Lu, Hongyu Lin, Ben He, Xianpei Han, Debing Zhang, and Le Sun. On-policy self-alignment with fine-grained knowledge feedback for hallucination mitigation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5215–5231, Vienna, Austria, July 2025. Association for Computational Linguistics.

[63] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[64] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2:1, 2023.

[65] Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. Conformal prediction with large language models for multi-choice question answering. *arXiv preprint arXiv:2305.18404*, 2023.

[66] Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi Jaakkola, and Regina Barzilay. Conformal language modeling. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Representation Learning*, volume 2024, pages 11654–11681, 2024.

[67] Mingjian Jiang, Yangjun Ruan, Sicong Huang, Saifei Liao, Silviu Pitis, Roger Baker Grosse, and Jimmy Ba. Calibrating language models via augmented prompt ensembles. In *Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML)*, 2023.

[68] Yasin Abbasi-Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not to believe your LLM: Iterative prompting for estimating epistemic uncertainty. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[69] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[70] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In *International Conference on Learning Representations*, 2018.

[71] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are Bayesian neural network posteriors really like? In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4629–4640. PMLR, 18–24 Jul 2021.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: As indicated in the abstract and introduction, we introduce a novel data-dependent, uncertainty-aware approach for fine-tuning LLMs.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of the work and the proposed method at the end of the paper in Section 7.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: Our work does not include any theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: In Section 4, we explain the settings of our experiments. Furthermore, in the Appendix, we clearly describe the hyperparameters that we utilized in our experiments.

   Guidelines:

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

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

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

Answer: [No]

Justification: We provide the experiment details in Section 4, and technical details in Section 3. Moreover, additional information is provided in the Appendix. We will release the code for the proposed approach upon acceptance of our paper.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Experiment settings and the hyperparameters are specified in Section 4 and Appendix respectively.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: In all our tables, we report the mean and standard deviations for all metrics over three independent runs, across all tasks.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We have provided the information regarding the computational resources that we used for conducting the experiments in the Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed and adhered to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer:[Yes]

    Justification: The proposed method enables the access to trustworthy fine-tuned LLMs for downstream tasks with reliable UQ which is necessary in many safety critical applications. Further discussion on the societal impacts of our method is provided in the Appendix.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [No]

    Justification: We do not provide safeguards for our proposed methods as we do not see the potential of a high risk of misuse of our uncertainty-aware fine-tuning approach for reliable UQ in LLMs.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have cited papers which provided the pretrained model, as well as the packages and libraries we used. We have also cited the papers for datasets used in this work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [No]

    Justification: We will release the GitHub repository implementing our C-LoRA method with proper instructions upon the acceptance of our paper.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: Our study does not involve research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [No]

    Justification: In this work, LLM was used only for editing and formatting purposes.

    Guidelines:

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

A Dataset Details

A brief summary of the prompt templates used for fine-tuning LLaMA2 on common-sense reasoning tasks is provided in Table 6. More details on the size of training datasets after applying an 80% train-validation split, and the number of labels are gathered in Table 7.

Table 6: Prompt templates for fine-tuning on common-sense reasoning tasks

task	**prompt**
Winogrande (WG-S/WG-M)	Select one of the choices that answers the following question: {question} Choices: A. {option1}. B. {option2}. Answer:
ARC (ARC-C/ARC-E), Openbook QA (OBQA), MMLU	Select one of the choices that answers the following question: {question} Choices: A. {choice1}. B. {choice2}, C. {choice3}. D. {choice4}. Answer:
BoolQ	Answer the question with only True or False: {question} Context: {passage}

Table 7: Size of the training dataset after train-validation split and number of labels in each dataset

	WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ
Training dataset size	512	892	1.8K	2.05k	3.97k	1.99k
Number of labels	2	5	5	2	4	2

B Temperature Scaling

Temperature scaling is a commonly adopted method to convert the predicted probabilities to more calibrated values [59, 69]. To do so, it employs a positive constant scaler, T, called the temperature parameter, to *soften* the softmax values making the distribution less peaky. That is, given the logit vector $\mathbf{z}$, the new confidence can be expressed as,

$$\mathbf{q} = \sigma(\mathbf{z}/T), \tag{9}$$

where σ is the softmax operator. It has been empirically shown in [59] that such temperature scaling leads to lower Expected Calibration Error (ECE) on classification tasks. Here, T can be estimated via optimizing the Negative Log-Likelihood (NLL) on the validation set. Since T does not change the maximum of the softmax function, the prediction is unaffected; therefore, it does not change the model's accuracy while enhancing model's calibration.

C Uncertainty Estimation Metrics

Uncertainty quantification is commonly assessed by NLL and ECE in the literature. The summation of the negative expected log probability of predicting the correct label is calculated for NLL. That is, for the model P_θ, and a dataset of size N, NLL is computed as,

$$\text{NLL} = \frac{1}{N}\sum_{i=1}^{N} -\log P_\theta(y_i), \tag{10}$$

where y_i denotes the correct label. This metric promotes the model to assign a higher probability to the correct predictions. For an overconfident model in an incorrect prediction, the probability of correct answer decreases, which leads to an increase in NLL. ECE on the other hand, estimates how well a model is calibrated by assessing how close the model's confidence is to its accuracy. Specifically, by binning the predictions according to the confidence levels, this metric is calculated by the weighted average of the absolute difference in each bin, that is,

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)|, \tag{11}$$

where acc(B_m) and conf(B_m) indicate the average accuracy and confidence in each bin, B_m,

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i), \quad \text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} P(\hat{y}_i), \tag{12}$$

in which $|B_m|$ denotes the number of examples in bin m.

D Hyperparameters

Table 8 summarizes the hyperparameters for LoRA fine-tuning of LLaMA2-7B, selected based on prior work [19–21] and default values provided in PEFT library [54]. Following [20], we optimize the KL term with SGD and a linear learning-rate scheduler.

Table 8: Hyperparameters of LoRA fine-tuning of LlaMA2-7B

Hyper-parameter	**Value**
Optimizer	AdamW
LR Scheduler	Linear
Learning Rate	$1e$-4
Batch Size	4
Max Sequence Length	300
LoRA α	16
LoRA r	8

E On the Importance of Flexibility

To study the impact of complexity we examine the performance of FE with the case where the matrix $\mathbf{E}$ is a diagonal matrix with r random variables. We refer to this as **DE**.

Table 9 summarizes the performance of DE and FE with M=0 and M=10 under in-distribution scenario for the six common-sense reasoning tasks. These models are ordered from top to bottom for each metric, such that each successive model is more flexible than the previous one. From Table 9, it is clear that as the flexibility increases, the performance generally improves. Specifically, in almost all tasks, FE provides a better uncertainty quantification with respect to DE, with a lower ECE and NLL. Although FE does not always deliver the highest accuracy, it shows a comparable performance across all tasks. Particularly, it provides the best accuracy in `ARC-C` under both deterministic (M=0) and stochastic (M=10) settings. It can be concluded from Table 9 that a more flexible model yields more reliable uncertainty quantification, with no significant loss in predictive accuracy.

To assess the effect of model complexity on generalization, we further compared DE and FE under out-of-distribution scenario, using the models fine-tuned on `OBQA` from Table 9. As Table 10 indicates, FE consistently delivers superior uncertainty quantification over all tasks under both smaller and larger distribution shifts. It also delivers the best accuracy in almost all tasks.

These findings from Tables 9 and 10 confirm that richer stochastic structures enhance both generalization and uncertainty estimation, motivating our focus on contextualizing the full matrix $\mathbf{E}$.

F Checkpoint Metric

Given our goal of achieving a well-calibrated model while maintaining competitive accuracy relative to the states of the art (SOTAs), we devise a metric that incorporates both accuracy (ACC) and expected calibration error (ECE). Specifically, we define:

$$\mathcal{C} = (1 - \text{ACC}_{\text{val}}) \cdot \text{ECE}_{\text{val}},$$

with ACC_{val} and ECE_{val} representing the validation accuracy and expected calibration error, respectively. During training, the model is evaluated using this criterion every 100 steps. We summarize the training algorithm of C-LoRA in Algorithm 1.

Table 9: Impact of different levels of flexibility on performance. Performance comparison is conducted across six common-sense reasoning tasks with a validation set split from the training dataset. The best and the second best performances are specified in **boldface** and via underline respectively.

Metric	Method	Datasets					
		WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ
ACC ↑	DE (M=0)	65.98±2.46	66.77±1.28	85.61±0.36	75.19±0.05	81.80±1.05	87.47±0.03
	DE (M=10)	66.09±1.50	66.88±1.47	85.71±0.64	75.19±0.16	82.06±0.41	87.44±0.07
	FE (M=0)	65.45±1.36	68.35±1.02	85.32±0.39	73.47±2.36	80.46±0.11	84.70±0.45
	FE (M=10)	65.06±1.33	68.20±2.40	85.08±0.36	72.43±1.28	80.93±0.50	84.65±0.52
ECE ↓	DE (M=0)	30.99±1.39	29.63±1.58	13.17±0.30	21.91±0.49	14.19±0.37	4.67±0.17
	DE (M=10)	27.00±0.62	26.26±1.17	11.79±0.16	20.21±0.09	13.24±0.34	4.41±0.02
	FE (M=0)	23.76±1.64	27.26±0.90	12.23±0.25	14.90±1.98	11.23±0.35	2.57±0.13
	FE (M=10)	**18.12**±1.52	**19.60**±2.42	**9.54**±0.47	**12.21**±1.87	**8.42**±0.50	**1.34**±0.22
NLL ↓	DE (M=0)	2.14±0.00	2.83±0.26	1.17±0.08	1.18±0.06	0.91±0.02	**0.32**±0.00
	DE (M=10)	1.58±0.05	2.21±0.06	1.05±0.07	1.05±0.05	0.85±0.02	**0.32**±0.00
	FE (M=0)	0.89±0.07	1.85±0.11	0.85±0.05	0.69±0.03	0.68±0.04	0.34±0.00
	FE (M=10)	**0.78**±0.04	**1.35**±0.12	**0.65**±0.02	**0.63**±0.02	**0.60**±0.03	0.34±0.00

Table 10: Impact of different levels of complexity on generalization. Performance comparison is conducted on out-of-distribution datasets. The following results are evaluated using LLaMA2-7B fine-tuned on the OBQA dataset. The best and the second best performances are specified in **boldface** and via *underline* respectively.

Metric	Method	Datasets				
		In-Dist.	*Smaller Dist. Shift*		*Larger Dist. Shift*	
		OBQA	ARC-C	ARC-E	Chem	Phy
ACC ↑	DE (M=0)	81.80±1.05	68.46±1.92	74.75±0.79	44.33±2.08	31.00±4.35
	DE (M=10)	82.06±0.41	67.90±0.67	74.80±1.07	43.33±0.57	30.66±4.51
	FE (M=0)	80.46±0.11	68.91±1.22	74.72±0.63	45.00±1.73	31.66±3.51
	FE (M=10)	80.93±0.50	66.65±1.09	74.76±0.70	46.00±1.73	32.33±1.52
ECE ↓	DE (M=0)	14.19±0.37	23.51±2.07	17.74±0.64	23.37±3.97	34.33±2.34
	DE (M=10)	13.24±0.34	22.31±0.53	16.81±1.17	21.90±1.90	35.14±5.68
	FE (M=0)	11.23±0.35	18.44±1.51	14.65±0.49	20.02±2.26	31.75±3.20
	FE (M=10)	8.42±0.50	**15.78**±1.11	**11.69**±0.68	**17.61**±1.25	**26.02**±1.32
NLL ↓	DE (M=0)	0.91±0.02	1.35±0.08	1.08±0.03	1.64±0.06	1.86±0.05
	DE (M=10)	0.85±0.02	1.28±0.05	1.04±0.02	1.62±0.06	1.82±0.05
	FE (M=0)	0.68±0.04	1.03±0.01	0.87±0.01	1.53±0.05	1.77±0.03
	FE (M=10)	0.60±0.03	**0.95**±0.02	**0.78**±0.01	**1.46**±0.03	**1.67**±0.02

G Flipout

To speed up the sampling, we apply the Flipout technique–originally introduced in [70] and also adopted by [51]–in C-LoRA to the low-rank matrix $\mathbf{E}$. In particular, having two randomly sampled flipping vectors $s = \{-1, +1\}^r$ and $t = \{-1, +1\}^r$, and considering $\mathbf{b}_i$ to be the i-th input in a mini-batch, the output after flipout is:

$$\mathbf{o_i} = \mathbf{W}\mathbf{b}_i = \mathbf{W_0}\mathbf{b}_i + \mathbf{B}\mathbf{E}\mathbf{A}\mathbf{b}_i = \mathbf{W_0}\mathbf{b}_i + \mathbf{B}(\boldsymbol{\mu}_\mathbf{E} + (\mathcal{E} \circ \boldsymbol{\Omega}_\mathbf{E}) \circ (t_i s_i^\top))\mathbf{A}\mathbf{b}_i, \quad \mathcal{E} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

H Discussion on Results of Laplace Approximation

For experiments involving fine-tuning via Laplace approximation (LA), we used the publicly released code provided by the authors of [19]. However, the results that we attempted to reproduce—reported in Table 1 in our main paper—are substantially worse than those reported in the original paper [19]. This discrepancy is especially pronounced for the `OBQA` and `BoolQ` datasets. Our findings are

Algorithm 1 Contextual Low-rank Adaptation **(C-LoRA)**

Require: Dataset split: $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{validation}}$, $\mathcal{D}_{\text{test}}$
Require: contextual module h_φ, deterministic parameters $\boldsymbol{\theta}$, number of iterations T, evaluation frequency f_{eval}, learning rate η, checkpoint metric b

$b \leftarrow \infty$
for $t \leftarrow 0$ to T **do**
  $\mathbf{x}, y \sim \mathcal{D}_{\text{train}}$
  $\mathbf{x}^0 \leftarrow \mathbf{x}$
  for $l \leftarrow 1$ to L **do**
    $\boldsymbol{\mu}_{\mathbf{E}}^l, \boldsymbol{\Omega}_{\mathbf{E}}^l \leftarrow h_\varphi^l(\mathbf{x}^{l-1})$
    $\mathcal{E}^l \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{E}_{\mathbf{x}}^l \leftarrow \text{Flipout}(\mathcal{E}^l)$
  end for
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{x}, y)$ ▷ Eq. 7
  $\varphi \leftarrow \varphi - \eta \nabla_{\varphi} \mathcal{L}(\mathbf{x}, y)$ ▷ Eq. 8
  if $t \bmod f_{\text{eval}} = 0$ **then**
    Compute validation accuracy and ECE
    $\tilde{b} \leftarrow (1 - \text{ACC}_{\text{val}}) \cdot \text{ECE}_{\text{val}}$
    if $\tilde{b} < b$ **then**
      $b \leftarrow \tilde{b}$
      Save $\boldsymbol{\theta}$, φ ▷ Checkpoint best model
      Record performance on $\mathcal{D}_{\text{test}}$
    end if
  end if
end for

consistent with those reported by the authors of BLoB [20], which suggests that the degradation in performance may stem from sub-optimal MAP solutions that negatively impact the quality of the Laplace approximation-based LoRA fine-tuning.

Additionally, in our experiments, we observed that LA is notably memory-intensive, requiring hardware with higher memory capacity than BLoB and our C-LoRA implementations. This might make the practical applicability of LA more challenging considering the scalability compared to other competing methods.

I Discussion on the choice of prior

In this work, we adopt a fixed Gaussian prior which, when paired with a variational Gaussian posterior, enables a closed-form KL divergence and further facilitates performance comparison across baseline methods. However, due to Monte Carlo approximation of the ELBO, alternative richer distributions could also be considered as priors. While richer priors may offer empirical benefits in specific settings, they can introduce additional computational and design challenges. Moreover, due to the highly expressive contextual modules, despite the simplicity, fixed Gaussian prior mainly act as a light regularizer rather than as a strict inductive bottleneck. Prior work [71] has also shown that Bayesian Model Averaging (BMA) is robust to the choice of prior, with posterior predictive behavior remaining similar across different prior families.

J Empirical Study of Contextualized Variance

To further illustrate that our model captures *input-dependent* uncertainty, we analyze the predictive variance behavior of the LM head for three OBQA questions, each sharing the same correct label (option "C"). For each question, we extract the predicted variance matrix from the final layer, compute the mean variance across tokens, and report the ℓ_2 norm of the resulting vector to summarize overall uncertainty.

Table 11: **Example questions and corresponding variance magnitudes.** Despite sharing the same label, the model assigns distinct variances to each question, reflecting input-dependent uncertainty.

Question	Variance ℓ_2 norm
Q1: What would light bounce off of when it hits it?	3.7661
Q2: A mother births what?	3.6792
Q3: What happens when animals in hot environments are active?	3.9443

Despite sharing the same label, the model assigns distinct variances to each question, reflecting differences in ambiguity and specificity of the input. This confirms that C-LoRA modulates its predictive distribution based on input semantics, not merely on class labels.

We further examined whether the learned variance meaningfully affects the model's representations. Specifically, we computed the ℓ_2 norm of the difference between the last-token embedding at $M = 0$ and the mean embedding across $M = 10$ stochastic samples. For the three representative examples (Q1–Q3), the differences ranged from 6.18 to 7.25, while the embedding norms themselves were approximately 120–122, corresponding to a relative change of about 5–6%. In embedding space, this level of perturbation is non-trivial and indicates meaningful input-dependent variability introduced by the contextualized variance modules.

We extended this analysis to a broader set of examples, comparing 8 correct and 8 incorrect predictions. For each, we computed the ℓ_2 norm of the difference between the deterministic embedding ($M = 0$) and the mean of embeddings from $M = 10$ samples, then averaged across each group. Correct predictions had an average difference of 6.7 ± 1.09, while incorrect predictions showed higher variability at 8.45 ± 2.66. This systematic increase in embedding perturbation for incorrect predictions suggests that C-LoRA's learned variance corresponds to greater uncertainty for ambiguous or difficult inputs, further supporting its interpretation as modeling input-specific uncertainty.

K Visualization of Results

For a visual performance comparison, we present the results under in-distribution scenario, after temperature scaling, and the results under out-of-distribution scenario, reported in Tables 1, 2, and 3, using bar plots.

Figure 3 presents bar plots summarizing the results reported in Table 1. As discussed in Section 4.2, all methods exhibit similar predictive performance overall, while C-LoRA and BLoB ($M = 10$) achieve superior uncertainty quantification, as measured by ECE and NLL. Figure 3 also illustrates the effect of temperature scaling on calibration, corresponding to Table 2. As noted in Section 4.2, temperature scaling substantially improves calibration. Finally, Figure 4 visualizes the out-of-distribution results from Table 3 via bar plots. Consistent with in-distribution results, performance degrades as the distribution shifts; however, predictive performance remains largely similar across methods, with C-LoRA and BLoB demonstrating more robust uncertainty quantification according to ECE and NLL. As we mentioned earlier, it worth noting that, although C-LoRA offers similar predictive performance overall, it can underperform on certain datasets in exchange for better uncertainty quantification.

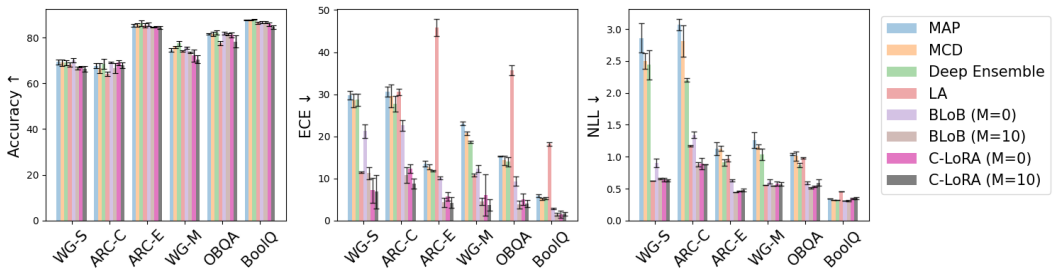

Figure 2: Visualization of the results in Table 1.

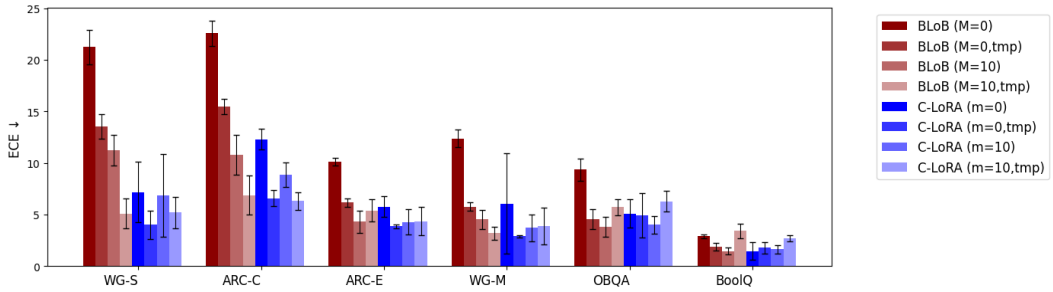

Figure 3: Visualization of the results in Table 2.

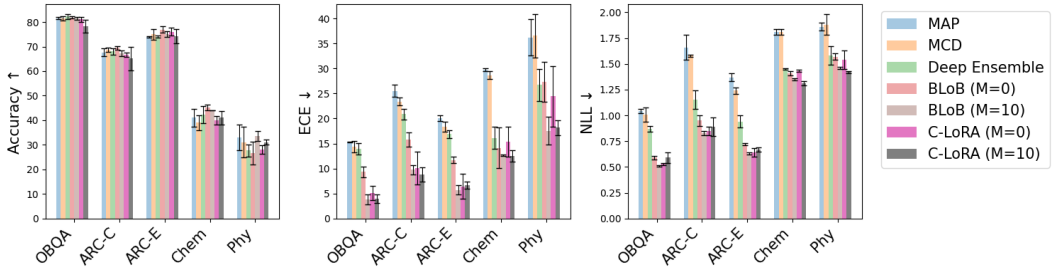

Figure 4: Visualization of the results in Table 3.

L Broader Impacts

Our uncertainty-aware fine-tuning framework integrating uncertainty quantification in LLMs could enhance the safety, reliability, and trustworthiness of AI systems deployed in high-stakes settings such as medical diagnosis, atmospheric modeling, and autonomous navigation, where unrecognized model errors can have major consequences. Moreover, as our approach explicitly models aleatoric or data uncertainty, it is particularly well-suited for low-resource tasks and rare-event prediction, providing access to robust LLM-based AI tools in fields ranging from global health to societal governance and policy-making.

We anticipate minimal additional computational overhead compared to standard fine-tuning, and because our framework is compatible with existing Bayesian extensions for epistemic uncertainty, it offers a clear path toward unified uncertainty quantification without introducing undue complexity. We do not foresee any negative societal impacts beyond those already inherent to large-scale model deployment, and we believe that equipping models with calibrated confidence measures is an essential step toward more ethical, accountable, and human-centered AI.

M Additional Information

In our contextual module, we set C, the number of hidden units, to 64 in order to ensure sufficient expressiveness for learning meaningful features. Also, to have a robust performance, the variance output, $\mathbf{\Omega_E}$, uses a sigmoid activation function; however, we do not use any activation function for the mean $\boldsymbol{\mu}_{\mathbf{E}}$. All the experiments for almost all methods were conducted using 1 NVIDIA A100 GPU with 40 GB memory except for BLoB, for which we used 2 NVIDIA A100 with 40 GB memory. Also for LA we used NVIDIA L40S GPU with 48 GB memory due to its memory demands.