FUNCTIONAL-LEVEL UNCERTAINTY QUANTIFICATION FOR CALIBRATED FINE-TUNING ON LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

From common-sense reasoning to domain-specific tasks, parameter-efficient fine tuning (PEFT) methods for large language models (LLMs) have showcased significant performance improvements on downstream tasks. However, fine-tuned LLMs often struggle with overconfidence in uncertain predictions, particularly due to sparse training data. This overconfidence reflects poor epistemic uncertainty calibration, which arises from limitations in the model's ability to generalize with limited data. Existing PEFT uncertainty quantification methods for LLMs focus on the post fine-tuning stage and thus have limited capability in calibrating epistemic uncertainty. To address these limitations, we propose Functional-Level Uncertainty Quantification for Calibrated Fine-Tuning (UQ4CT), which captures and calibrates functional-level epistemic uncertainty during the fine-tuning stage via a mixture-of-expert framework. We show that UQ4CT reduces Expected Calibration Error (ECE) by more than 25% while maintaining high accuracy across 5 benchmarks. Furthermore, UQ4CT maintains superior ECE performance with high accuracy under distribution shift, showcasing improved generalizability.

1 INTRODUCTION

Large Language Models (LLMs) have revolutionized various domains as general task solvers (Chang et al., 2024). To adapt LLMs for specific downstream tasks or create instruction-following models, fine-tuning have become increasingly important (Houlsby et al., 2019; Hu et al., 2021a; Liu et al., 2022; Ding et al., 2022; 2023). This involves additional training on pre-trained LLMs using a smaller dataset (Zhong et al., 2021; Ren et al., 2022). Through fine-tuning, the model parameters are updated to better adapt to the domain-specific knowledge (Peng et al., 2023). To reduce the computational cost for fine-tuning, Hu et al. (2021a) proposed Low-Rank Adaptation (LoRA), which effectively reduces the parameters required for fine-tuning by introducing low-rank trainable matrices at each layer of the transformer architecture instead of fine-tuning the full model parameters. Li et al. (2024); Wu et al. (2024b) proposed LoRA Mixture-of-Experts (MoE) models which grants better performance while maintaining parameter efficiency.

However, previous studies have shown that fine-tuned LLMs are often overconfident with their predictions (Xiao et al., 2022c; He et al., 2023; Tian et al., 2023; OpenAI, 2023). This resembles poorly calibrated uncertainty (Zhou et al., 2022) due to the sparsity of fine-tuning data. Overconfidence is a crucial problem in safety-related decision making or in fields where data is very limited, such as experimental design, climate science and medical diagnosis (Singhal et al., 2022; Wu et al., 2023a; Lampinen et al., 2023; Li et al., 2022). Thus, methods that enhance uncertainty quantification of fine-tuned LLMs is urgently needed to assure trustworthy predictions for better application.

Established uncertainty quantification methods have been studied in conjunction with the LoRA structure. Monte-Carlo dropout (Gal & Ghahramani, 2016b) interprets dropout in neural networks as approximate Bayesian inference in deep Gaussian processes, allowing uncertainty estimates to be obtained from existing LoRA adapters without modifying them. Checkpoint ensemble (Chen et al., 2017) utilizes predictions from multiple LoRA checkpoints saved during a single fine-tuning process to calibrate uncertainty. Deep ensemble (Lakshminarayanan et al., 2017; Wang et al., 2023; Zhai et al., 2023a) combines the predictions from multiple LoRA adapters for better uncertainty calibration. Laplace-LoRA (Yang et al., 2024a) applies Bayesian inference via Laplace approximation to the LoRA parameters after fine-tuning, resulting in improved calibration and uncertainty estimates.

Although these methods have demonstrated improved uncertainty estimations, they all utilize a fixed set of LoRA parameters fine-tuned over the entire downstream task dataset. The point estimates of parameters have very limited capabilities in capturing epistemic uncertainty, while direct calibration of epistemic uncertainty over the entire LoRA parameter space is an ideal but not practical approach.

Therefore, we propose Functional-Level Uncertainty Quantification for Calibrated Fine-Tuning (UQ4CT) to calibrate the functional-level epistemic uncertainty via the LoRA MoE architecture. We propose a functional perspective on LoRA parameters where we treat the LoRA experts as basis functions and consider the more complex, prompt dependent functions as mixtures of those basis functions. On top of learning the parameters in the LoRA experts, UQ4CT also trains a prompt-dependent LoRA mixture to form a calibrated distribution over the functional space. The LoRA experts capture different functional relationships in the fine-tuning data throughout training, and the MoE routers dynamically select these functional bases conditioned on the input. The selection process models the functional level epistemic uncertainty, and consequently captures the uncertainty in the output space. We calibrate functional level epistemic uncertainty to align with predictive correctness during training time. This significantly improves uncertainty estimations of the model on its predictions without compromising the accuracy. To summarize, our contributions include:

- A novel epistemic uncertainty quantification approach with Mixture-of-Experts architecture during fine-tuning stage to model functional level epistemic uncertainty and align with predictive correctness, which mitigates overconfidence issue and improves generalizability.
- A novel training calibration loss function incorporating predictive correctness to calibrate the prompt-dependent LoRA mixture for better uncertainty estimation.
- More than 25% Expected Calibration Error reduction on 4 common-sense reasoning tasks and 1 domain-specific question answering task, superior ECE performance under distribution shift scenarios on 2 common-sense reasoning tasks and 4 domain-specific question answering tasks without compromising accuracy.

2 PRELIMINARIES

2.1 LOW-RANK ADAPTATION (LORA)

LLMs have numerous large weight matrices to perform matrix multiplication, denoted as $\mathbf{W}_0 \in \mathbb{R}^{n_{\text{out}} \times n_{\text{in}}}$ that maps inputs $\mathbf{x}$ to outputs $\mathbf{h}$. Hu et al. (2021a) proposes LoRA, which fixes $\mathbf{W}_0$ and introduces a low-rank perturbation $\Delta\mathbf{W}$ to the weight matrix:

$$\mathbf{h} = \mathbf{W}_0\mathbf{x} + \Delta\mathbf{W}\mathbf{x} = \mathbf{W}_0\mathbf{x} + \mathbf{B}\mathbf{A}\mathbf{x}. \tag{1}$$

Here, $\Delta\mathbf{W}$ is calculated as the product of two matrices, $\mathbf{B} \in \mathbb{R}^{n_{\text{out}} \times n_{\text{lr}}}$ and $\mathbf{A} \in \mathbb{R}^{n_{\text{lr}} \times n_{\text{in}}}$ where n_{lr} is significantly smaller than n_{in} or n_{out}. For example, we use $n_{\text{lr}} = 32$ while $n_{\text{in}} = n_{\text{out}} = 4096$ for the Llama2-7b model (Touvron et al., 2023c). Therefore, the total number of LoRA parameters for this $\Delta\mathbf{W}$ is $n_{\text{lr}}(n_{\text{in}} + n_{\text{out}})$, which is far smaller than the parameter count of the full matrix, $n_{\text{in}}n_{\text{out}}$. One of the key motivations of incorporating LoRA to fine-tune LLMs is the vast amount of memory cost reduction compared with fine-tuning on the full model. For an LLM with 7 billion parameters, maintaining the average gradient and average squared gradients for optimization multiplies the memory required by a factor of 3 compared to simply loading model weights. LoRA greatly mitigates this memory cost as the tripled memory consumption only applies to LoRA adapters.

2.2 MIXTURE OF EXPERTS (MOE)

LoRA Mixture-of-Experts (Li et al., 2024; Wu et al., 2024b) is a efficient approach to scale the number of parameters while maintaining the same computational bounds. LoRA MoE utilizes the top-k router to assign each token to the LoRA experts (Lepikhin et al., 2020). The router is a linear layer that maps the input hidden state $\mathbf{h}$ to a probability distribution of candidate experts.

Let $\mathbf{h}_i^\ell \in \mathbb{R}^{1\times d}$ $(1 \le i \le s, 1 \le \ell \le L)$ denote the output hidden state of the i-th token at the ℓ-th layer of the LLM, where L is the number of LLM layers and d is the hidden dimension. With $\mathbf{W}_r^\ell$ as

the trainable router weight at layer ℓ, the top-k gate router chooses k experts with highest probability given a hidden state $\mathbf{h}_i^\ell$:

$$R^\ell(\mathbf{h}_i^\ell) = \text{KeepTop-k}(\text{Softmax}(\mathbf{W}_r^\ell \cdot \mathbf{h}_i^\ell)). \tag{2}$$

Finally, we obtain the final MixLoRA prediction with:

$$\text{MixLoRA}(\mathbf{h}^\ell) = \sum_{k=1}^{K} R^\ell(\mathbf{h}^\ell)_k E_k^\ell(\mathbf{h}^\ell), \quad E_k^\ell(\mathbf{h}^\ell) = \mathbf{W}^\ell \cdot \mathbf{h}^\ell + \mathbf{B}_i^\ell \mathbf{A}_i^\ell \cdot \mathbf{h}^\ell \tag{3}$$

where $\mathbf{W}$ is the pretrained weights of the feed-forward network (FFN) layer and $\mathbf{B}_i^\ell \mathbf{A}_i^\ell$ is the i-th LoRA expert.

2.3 Aleatoric and Epistemic Uncertainties

In machine learning models, uncertainty can be categorized into aleatoric (data-wise) and epistemic (model-wise) uncertainty (Hora, 1996; Hüllermeier & Waegeman, 2021). For LLMs, aleatoric uncertainty arises from the inherently ambiguous and context dependent nature of natural languages where a single phrase or sentence can have multiple valid interpretations in different contexts. Epistemic uncertainty is introduced by the model's lack of knowledge due to limited learning capabilities, suboptimal modeling or sparse training data.

Epistemic uncertainty is highly related to several well-known limitations of generative models. For example, it has been observed that when an LLM is pretrained on a diverse range of text data, it is generally well-calibrated, i.e. the predicted probability of the next token generally aligns with what is observed in real text. However, after fine-tuning or alignment with human preferences, the calibration error deteriorates (Zhao et al., 2021; Achiam et al., 2023a). A related phenomenon is forgetting, where the performance of a fine-tuned LLM diminishes on tasks outside the scope of the target downstream task (Lin et al., 2023; Luo et al., 2023).

Motivated by these observations, we explore functional-level epistemic uncertainty in generative models and aim to develop metrics that assess model performance on specific problem instances to fine-tune the parameter mixture.

3 Methodology

The high level goal of UQ4CT is to balance the exploration and exploitation of different LoRA experts during fine-tuning. In particular, we incorporate the functional-level epistemic uncertainty (FEU) to calibrate the prompt-dependent parameter mixture with LoRA MoE.

Assume that our answer a is generated via a mixture of mechanisms or models M, conditioning on the input prompt x. Assume that $e(a)$ is an embedding of a so that least squares distance is a natural distance on the space of $e(a)$. For an expressive enough model class $\mathcal{M}$, and a calibrated distribution $P(M|x)$ over the model class, we can measure the deviation of the generated answer to the "ideal" one $a_* = f_*(x)$ as:

$$\begin{aligned}
&\mathbb{E}_{M\sim P(M|x)}\left[\mathbb{E}_{a\sim P(a|M,x)}\left[\|e(a) - e(a_*)\|^2\right]\right] \\
&= \mathbb{E}_{P(M|x)}\underbrace{\left[\mathbb{E}_{P(a|M,x)}\left[\|e(a) - \mathbb{E}_{P(a|M,x)}[e(a)]\|^2\right]\right]}_{\text{Aleatoric Uncertainty}} + \underbrace{\mathbb{E}_{P(M|x)}\left[\|\mathbb{E}_{P(a|M,x)}[e(a)] - e(a_*)\|^2\right]}_{\text{Epistemic Uncertainty}}.
\end{aligned} \tag{4}$$

Note that we hereby quantify uncertainty as a function of the input prompt x, since the distribution of model M conditions on x. We hence name the task "functional-level uncertainty quantification".

3.1 Functional-level Epistemic Uncertainty

Motivated by the decomposition in Eq. (4) for least squares loss, we may consider a general distance d, and define epistemic uncertainty that characterizes the variation caused by model training procedure. Specifically, we focus on the variation introduced in the model fine-tuning stage of LLMs.

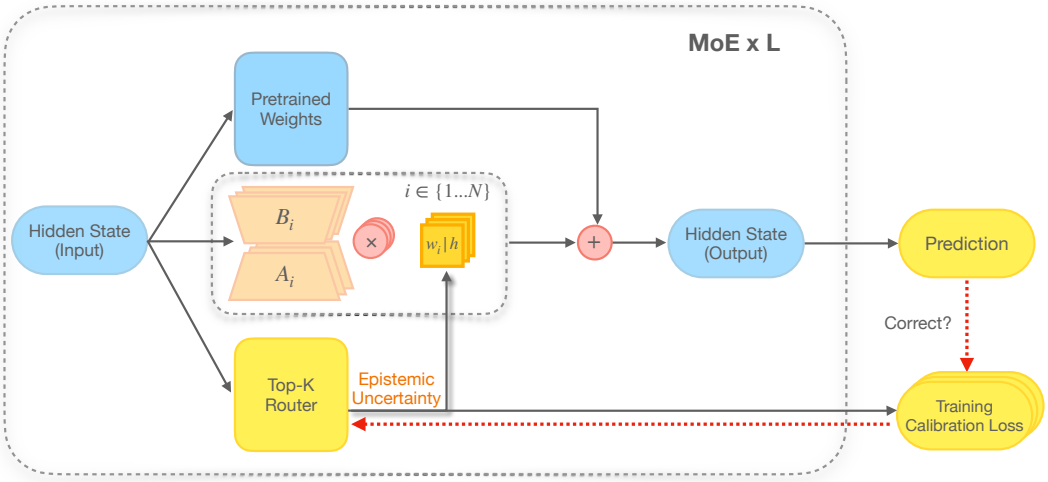

Figure 1: Mixture of Experts (MoE) architecture to capture and calibrate functional-level epistemic uncertainty. Experts $B_{1...N}A_{1...N}$ capture functional relationships in the data throughout fine-tuning, the weights $w_{1...N}|h$ quantify the uncertainty in selecting these functional bases conditioned on the input hidden state h, which is the semantic representation of the input token x. In the UQ4CT workflow, we align this uncertainty with predictive correctness. When the router makes a correct prediction, the loss reinforces this decision, thereby increasing the router confidence in its selection, which aligns with a lower epistemic uncertainty. Conversely, when the router makes an incorrect prediction, the loss penalizes this selection, potentially causing the router to distribute its probabilities more broadly across experts, which is indicative of higher epistemic uncertainty.

Mathematically, given prompt x, we consider the following definition of epistemic uncertainty:

$$\text{Epistemic Uncertainty} = \mathbb{E}_{M\sim P(M|x)}\mathbb{E}_{M'\sim P(M'|x)}\left[\|\mathbb{E}_{P(a|M,x)}[e(a)] - \mathbb{E}_{P(a'|M',x)}[e(a')]\|^2)\right]. \tag{5}$$

Here, a' represents the ground truth output sampled from the ideal MoE model M' conditioned on the prompt x. The epistemic uncertainty measures the least squares distance between $e(a)$ and $e(a')$ from current mixture distribution $P(M|x)$ and the ideal mixture distribution $P(M'|x)$.

3.2 Quantifying FEU with LoRA MoE Framework

We quantify the functional-level epistemic uncertainty (FEU) with the MoE architecture, which is represented by the embedding $e(a)$ in Eq. (5). As shown in Figure 1, the LoRA experts $B_{1...N}A_{1...N}$ capture important functional relationships in the data during fine-tuning. We treat these functions represented by the LoRA experts as basis functions $f_{1...N}$ and define them as follows:

$$f_1, f_2, \ldots, f_N = \{B_1A_1, B_2A_2, \ldots, B_NA_N\}. \tag{6}$$

Conditional on the input prompts, the more complex functional relationships that recursively map inputs to outputs are represented as linear combinations (mixture of experts) of the basis functions.Uncertainty quantification in this functional space reduces to quantifying the uncertainty of the weights over the basis functions. In particular, the weights $w_{1...N}|h$ from the top-k router dynamically selects these basis functions conditioned on the input hidden state h:

$$h = \sum_{i=1}^{N}(w_i|h)\cdot f_i = \sum_{i=1}^{N}(w_i|h)\cdot(B_iA_i). \tag{7}$$

The top-k weights that produce the final output hidden state quantify the functional level epistemic uncertainty in the function selection.

In the mixture of LoRA experts architecture, we follow the routing mechanisms of the MoE layers as in Eq. (2) and (3). Specifically, we employ top-2 gate routers, which chooses the 2 experts with

highest probability given a hidden state $\mathbf{h}_i^\ell$:

$$R^\ell(\mathbf{h}_i^\ell) = \text{KeepTop-2}(\text{Softmax}(\mathbf{W}_r^\ell \cdot \mathbf{h}_i^\ell)). \tag{8}$$

Given an input prompt x with length s, we model functional-level epistemic uncertainty (FEU) by aggregating $R^\ell(\mathbf{h}_i^\ell)$ over both layer dimension and sequence dimension:

$$\text{FEU}(x) = \frac{1}{s}\sum_{i=1}^{s}\left[\frac{1}{L}\sum_{\ell=1}^{L} R^\ell(\mathbf{h}_i^\ell)\right] \tag{9}$$

3.3 Training Calibration Loss

We then calibrate the FEU model of the epistemic uncertainty against predictive accuracy. Specifically for the MoE top-k routers, we design the following calibration loss for training:

$$\mathcal{L}_{\text{cal}} = \left\|\mathbb{1}\{\text{MixLoRA}(x) = y^*\} - \text{FEU}(x)\right\|^2, \tag{10}$$

where the first term is an indicator function of whether the model prediction matches the ground truth y^* given the prompt x. Here, the indicator function resembles $\mathbb{E}_{P(a'|M',x)}[e(a')]$ in Equation 5, where the ground truth y^* is a' and the indicator function maps the predictive correctness to a confidence space $e \in [0, 1]$. We employ a one-hot definition of the ground truth confidence. When the prediction from current mixture model matches the ground truth, the ground truth confidence is 1. Otherwise, when the predictions do not match, the ground truth confidence is 0.

As shown in Figure 1, this term effectively promotes expert exploitation for correct predictions and expert exploration for incorrect predictions by directly calibrating the functional level epistemic uncertainty to align with the predictive correctness. Ideally, when the N LoRA experts together capture all the functional relationships across the data distribution with cross entropy during fine-tuning, our proposed loss $\mathcal{L}_{cal}$ also finds proper mixture of LoRA experts conditioned on the input x by conditionally promoting expert exploitation and exploration. This allows the model to select correct functional relationships regarding x to generate an output that better matches the data distribution, which grants calibrated uncertainty estimations.

Load balancing is a common technique to ensure even exploitation across experts with the MoE architecture (Fedus et al., 2022). We follow the load balancing loss $\mathcal{L}_b$ proposed by (Li et al., 2024) and define our loss function as:

$$\mathcal{L} = \mathbf{CE} + \alpha \cdot \mathcal{L}_b + \beta \cdot \mathcal{L}_{\text{cal}}, \tag{11}$$

where $\mathbf{CE}$ represents cross entropy loss, α and β are the hyperparameters of two auxiliary terms. Details about $\mathcal{L}_b$ can be found in Appendix A.1.

4 Related Work

4.1 Parameter-Efficient Fine Tuning for LLMs

Large Language Models (Brown et al., 2020; Chowdhery et al., 2022; Hoffmann et al., 2022; Touvron et al., 2023a;d) have shown impressive abilities in handling various natural language processing tasks. Building on these advances, instruction fine-tuning(Chung et al., 2022; Iyer et al., 2022; Zheng et al., 2024) has enhanced LLMs' capacity to comprehend and follow human instructions, forming the core of modern conversational AI systems(Wu et al., 2023b; Achiam et al., 2023b). However, as LLMs increase in size, the fine-tuning process demands much more time and memory.

To address these challenges, several strategies have been proposed, including parameter-efficient fine-tuning (PEFT)(Mangrulkar et al., 2022), model distillation(Liu et al., 2023; Xiao et al., 2023), quantization(Frantar et al., 2022; Xiao et al., 2022a), and pruning(Frantar & Alistarh, 2023; Ma et al., 2023). Among these, LoRA(Hu et al., 2021b), which leverages low-rank matrix decomposition of linear layer weights, is a widely adopted PEFT technique that boosts model performance without adding computational costs during inference. For example, VeRA(Kopiczko et al., 2023) introduces learnable scaling vectors to modify shared pairs of frozen random matrices across layers, while FedPara(Hyeon-Woo et al., 2021) focuses on low-rank Hadamard products for federated learning

settings. Tied-Lora(Renduchintala et al., 2023) applies weight tying to further minimize the number of trainable parameters. AdaLoRA(Zhang et al., 2023) uses Singular Value Decomposition (SVD) to prune less important singular values for efficient updates, and DoRA(Liu et al., 2024) separates pre-trained weights into magnitude and direction components, applying LoRA to update the directional component during fine-tuning, thus reducing the number of parameters to be trained.

4.2 Mixture of Experts

The Mixture-of-Experts concept(Jacobs et al., 1991), introduced as early as 1991, presented a novel supervised learning framework where multiple networks (experts) specialize in handling distinct subsets of training data. Modern MoE variants adapt this by modifying the traditional feed-forward sub-layer within transformer blocks, incorporating sparsely activated LoRA experts, which allows for significant expansion in model width without a proportional increase in computational overhead.

Different MoE architectures have since emerged, distinguished by their expert sampling and routing strategies. For example, LLaVA-MoLE(Chen et al., 2024) improves token routing to domain-specific experts within transformer layers, reducing data conflicts and consistently outperforming standard LoRA baselines. Other MoE-based approaches include MoRAL(Yang et al., 2024b), which focuses on efficiently adapting LLMs to new domains and tasks for lifelong learning, and LoRAMoE(Dou et al., 2024), which incorporates LoRAs via a router network to mitigate the issue of world knowledge forgetting. PESC(Wu et al., 2024a) transforms dense models into sparse ones through an MoE structure, lowering computational and GPU memory requirements. MoE-LoRA(Luo et al., 2024) introduces a new parameter-efficient MoE method using Layer-wise Expert Allocation (MoLA) for transformer models, while MoCLE(Gou et al., 2023) activates task-specific model parameters based on instruction clusters. MixLoRA (Li et al., 2024) implements a high-throughput framework for LoRA MoE training and inference process, constructing LoRAs as stochastic experts to reduce computational overhead while expanding model capacity.

Despite the performance improvements these architecture advancements have brought, the overconfidence problem of fine-tuned models is lacking attention (Xiao et al., 2022c; He et al., 2023; Tian et al., 2023; OpenAI, 2023). Enhancing the uncertainty estimation capabilities of these models is fundamental toward more reliable, interpretable and trustworthy applications of LLMs.

4.3 Uncertainty Quantification in LLMs

Uncertainty quantification has garnered substantial attention in various tasks and domains within neural networks(Gal & Ghahramani, 2015; Gal & Ghahramani, 2016a; Malinin & Gales, 2018; Ovadia et al., 2019; Malinin et al., 2021; Lin et al., 2022; Kuhn et al., 2023; Lin et al., 2023). This focus extends to LLMs, where the precise quantification of prediction uncertainty has become a critical area of research(Xiao et al., 2022b; Lin et al., 2022; Mielke et al., 2022; Chen & Mueller, 2023; Duan et al., 2023; Huang et al., 2023). LLMs, particularly in generative tasks, pose unique challenges, especially when it comes to measuring the uncertainty of their outputs(Liu et al., 2019; Malinin & Gales, 2021; Kuhn et al., 2023; Lin et al., 2023). The distinction between aleatoric and epistemic uncertainty was recently examined in the context of LLMs(Hou et al., 2023), though this was approached by ensembling model inputs rather than model instances and did not address fine-tuning tasks specifically.

Existing works have investigated the application of ensembling in fine-tuning LLMs for uncertainty quantification. Gleave & Irving (2022); Sun et al. (2022) focus on uncertainty estimation in full model fine-tuning, while this approach inherently incurs significant memory overhead. Wang et al. (2023); Zhai et al. (2023b); Balabanov & Linander (2024) explore the use of LoRA ensembles for uncertainty estimation in LLMs. Yang et al. (2024a) applies a post-hoc Laplace approximation Mackay (1992) to model LoRA parameters after fine-tuning. BatchEnsemble(Wen et al., 2020), introduces component-specific rank-1 matrices as multiplicative modifications to a base model. Though this method has been applied to LLMs, it has been used in the pre-training phase rather than fine-tuning(Tran et al., 2022). None of these methods provide calibrations on epistemic uncertainty, which is crucial to mitigate overconfidence in the fine-tuning stage given the sparse dataset.

5 EXPERIMENTS

5.1 DATASETS

We include 5 multiple-choice question answering benchmarks to evaluate UQ4CT: OpenBookQA (OBQA, Mihaylov et al. (2018)), ARC-Easy (ARC-E) and ARC-Challenge (ARC-C) from AI2 Reasoning Challenge (Clark et al., 2018), BOOLQ (Clark et al., 2019) and ClimateQA, an expert-annotated domain specific benchmark for climate science. We also use computer science, law, medication and engineering subsets from MMLU dataset(Hendrycks et al., 2020) to evaluate performance under distribution shift. We fine-tune on the publicly available training split and test on the validation split from these benchmarks to evaluate model performance. We report the average model performance over 3 random runs and the standard deviations in the subscript.

5.2 EXPERIMENT SETUP

We implement UQ4CT with PyTorch (Paszke et al., 2019), extending the MixLoRA repository in (Li et al., 2024) and compare the average performance in three random runs and report the mean and standard deviation with following baselines. We use the publicly available LLaMA-2-7B-hf (Touvron et al., 2023c) as our base model. In particular, we apply MixLoRA to query, key, value and output layers, together with the feed-fowrward networks in LLaMA-2-7B-hf (gate layer, down layer and up layer). Details are provided in Appendix A.4

- **LoRA**(Hu et al., 2021a). We use standard LoRA fine-tuning as lower performance bound.
- **Monte Carlo (MC) Dropout**(Gal & Ghahramani, 2016b) keeps dropout on at both training and testing time. By performing multiple forward passes, MC dropout randomly shuts down a portion of model nodes, producing ensemble-alike predictions. To combine LoRA fine-tuning with MC dropout, we apply dropout on the input of the LoRA adapter, following the implementation of Mangrulkar et al. (2022).
- **Deep Ensemble**(Lakshminarayanan et al., 2017) averages the predictions from each ensemble member which have been trained with varying random initialization. We combine deep ensemble with LoRA by fine-tuning 3 randomly initialized LoRAs together and ensembling their output as final predictions.
- **Laplace-LoRA (LA)**(Yang et al., 2024a) applies a post-hoc Laplace approximation on fine-tuned LoRA parameters for better uncertainty estimation.
- **MixLoRA**(Li et al., 2024) incorporates LoRAs via a router network to reduce computational overhead while expanding model capacity. We add this as a baseline to resemble plain LoRA MoE model performance.

Evaluation Metrics. We measure the prediction accuracy on the validation set for all 5 tasks. For uncertainty calibration, we incorporate expected calibration error (ECE, Guo et al. (2017)) with 15 bins, which measures the alignment between predicted probabilities and empirical accuracy. We also investigate model performance under distribution shift to ensure the model has predictable behavior when given data from other domains as this is a crucial component for real-world applications. Specifically, we test models fine-tuned on OBQA dataset with 4 domain-specific MMLU subtask ensembles focusing on different professionalities and ARC-E/C datasets to approximate larger and smaller distribution shifts. Metric details are provided in Appendix A.5.

5.3 RESULTS

We assess the prediction accuracy and uncertainty calibration of models under both in-distribution and distribution shift scenarios. The in-distribution scenario examines the alignment of the fine-tuned model on the target downstream task, while the distribution shift scenario evaluates the generalizability of the model on novel tasks beyond the fine-tuned domain. These two scenarios combined provides a comprehensive assessment of model robustness in real-world applications where it is essential for the model to excel on its primary task while maintaining the ability to effectively handle unforeseen or out-of-distribution inputs.

Table 1: Performance comparison of different methods fine-tuned with LlaMA2-7B across 4 common sense reasoning tasks and a domain-specific task. UQ4CT shows substantial ECE improvements while maintaining high accuracy.

Metrics	Methods	BoolQ	ARC-E	ARC-C	OBQA	ClimateQA
ACC ↑	LoRA	$69.5_{1.93}$	$74.8_{1.39}$	$53.8_{0.6}$	$72.1_{0.87}$	$59.9_{2.13}$
	MC Drop	$66.8_{3.66}$	$76.8_{1.30}$	$50.9_{2.01}$	$74.8_{1.34}$	$58.2_{2.11}$
	Ensemble	$66.2_{3.7}$	$71.2_{1.0}$	$47.5_{0.57}$	$75.5_{1.4}$	$59.6_{6.9}$
	LA	$68.7_{1.32}$	$74.6_{2.11}$	$51.4_{0.83}$	$70.8_{1.24}$	$55.2_{3.29}$
	MixLoRA	$71.5_{1.05}$	$77.7_{2.27}$	$54.3_{1.07}$	$75.5_{2.91}$	$61.6_{1.76}$
	UQ4CT	$73.5_{0.52}$	$76.6_{1.30}$	$52.8_{1.77}$	$77.3_{1.36}$	$63.3_{1.74}$
ECE ↓	LoRA	$11.9_{0.78}$	$11.9_{2.04}$	$19.4_{4.75}$	$10.2_{1.07}$	$14.3_{0.64}$
	MC Drop	$12.2_{0.85}$	$11.9_{1.99}$	$19.8_{4.85}$	$10.9_{0.24}$	$14.3_{0.56}$
	Ensemble	$7.28_{2.3}$	$9.1_{1.49}$	$10.23_{1.39}$	$8.83_{2.35}$	$13.5_{3.29}$
	LA	$17.1_{1.72}$	$16.6_{3.7}$	$18.1_{0.5}$	$17.2_{1.2}$	$12.6_{1.9}$
	MixLoRA	$7.88_{2.09}$	$9.09_{0.81}$	$10.74_{1.07}$	$12.9_{1.99}$	$12.5_{1.32}$
	UQ4CT	$\mathbf{2.3_{0.82}}$	$\mathbf{6.0_{0.2}}$	$\mathbf{6.1_{1.11}}$	$\mathbf{5.0_{1.15}}$	$\mathbf{8.1_{0.52}}$

5.3.1 In-distribution Performance

We first evaluate UQ4CT and baseline models fine-tuned on the 4 common sense reasoning tasks and the climate question answering task under the in-distribution scenario, where models are trained and evaluated on different splits of the same dataset. Note that one of the key advantages of UQ4CT is that uncertainty calibration happens during the fine-tuning stage with little computational overhead, while other UQ methods require repetitive sampling or other post-hoc complexities.

As shown in Table 1, UQ4CT demonstrates notable improvements in uncertainty calibration across a variety of tasks. Across all benchmarks, UQ4CT maintains competitive accuracy (ACC) compared to the baseline methods. For example, on the BoolQ and ClimateQA datasets, UQ4CT achieves accuracy rates of 73.5% and 63.3%, respectively. This empirically demonstrates that UQ4CT is capable of maintaining high accuracy with uncertainty calibration, which assures the gain in UQ performance does not compromise accuracy.

The most substantial performance improvement is observed in the reduction of Expected Calibration Error (ECE). UQ4CT consistently outperforms other methods, reducing ECE by more than 25% on average across the evaluated benchmarks. Unlike other methods where the ECE performance is worsened on the more challenging ARC-C benchmark, UQ4CT achieves an ECE score of 6.1, showcasing the effectiveness of the calibration.

In addition to experiments on LLaMA-2-7B in the main text, we also present additional experiments on fine-tuning Mistral-7B in Appendix A.2 for more comprehensive evaluation of our method. For both LLaMA-2-7B and Mistral-7B models, UQ4CT consistently shows substantial improvements in uncertainty calibration across various tasks. The improvements are critical in applications where the model's confidence must align with its predictive accuracy given limited data, particularly in safety-critical and domain-specific tasks.

5.3.2 Performance Under Distribution Shift

Due to the sparse nature of the fine-tuning data, real world deployment of LLMs often requires the model to be robust to out-of-distribution knowledge (Ouyang et al., 2022; Touvron et al., 2023b;c). Therefore, we evaluate the performance of UQ4CT along with other baseline models fine-tuned on the OBQA dataset under smaller and larger distribution shift scenarios. Similar to the dataset setup in (Yang et al., 2024a), we use ARC-C and ARC-E dataset to simulate smaller distribution shift because the ARC dataset has similar domain focus on general science reasoning, but is generally more challenging and covers a broader range of scientific topics than OBQA. For larger distribution shift, we ensemble the domain-specific MMLU subtasks into 4 benchmarks focusing on different professionalities: Computer Science (CS), Engineering (Eng), Law and Health. These tasks have very broad coverage of the domain task at various knowledge levels ranging from elementary school

Table 2: Performance comparison of different methods fine-tuned on OBQA dataset with LlaMA2-7B across 2 smaller distribution shift (DS) tasks and 4 larger distribution shift tasks. UQ4CT shows substantial ECE improvements while maintaining high accuracy.

		ID	Smaller DS		Larger DS			
Metrics	Methods	OBQA	ARC-C	ARC-E	CS	Eng	Law	Health
ACC ↑	LoRA	$72.1_{0.87}$	$58.6_{1.93}$	$66.5_{3.38}$	$35.5_{2.35}$	$30.8_{1.72}$	$34.9_{1.41}$	$39.1_{1.52}$
	MC Drop	$74.8_{1.34}$	$58.7_{2.07}$	$66.6_{3.30}$	$36.0_{1.69}$	$30.3_{2.25}$	$35.1_{0.86}$	$39.1_{1.35}$
	Ensemble	$75.5_{1.4}$	$57.7_{0.78}$	$69.1_{0.48}$	$36.7_{2.18}$	$30.3_{1.13}$	$35.3_{1.02}$	$39.9_{1.89}$
	LA	$70.8_{1.24}$	$58.7_{0.58}$	$67.9_{0.41}$	$33.7_{1.22}$	$29.6_{1.32}$	$35.4_{0.75}$	$38.5_{1.61}$
	MixLoRA	$75.5_{2.91}$	$58.5_{1.44}$	$69.2_{1.02}$	$35.2_{2.92}$	$30.3_{0.98}$	$35.9_{0.43}$	$40.6_{1.13}$
	UQ4CT	$77.3_{1.36}$	$58.8_{1.06}$	$65.8_{1.31}$	$36.2_{1.24}$	$34.1_{2.31}$	$35.8_{1.01}$	$40.0_{1.24}$
ECE ↓	LoRA	$10.2_{1.07}$	$16.7_{2.28}$	$13.3_{2.48}$	$29.7_{2.69}$	$32.3_{1.85}$	$29.2_{3.08}$	$31.0_{2.13}$
	MC Drop	$10.9_{0.24}$	$16.7_{2.20}$	$13.2_{2.21}$	$23.2_{2.32}$	$31.6_{1.64}$	$28.0_{2.93}$	$25.9_{2.27}$
	Ensemble	$8.83_{2.35}$	$15.1_{1.09}$	$11.1_{0.99}$	$22.4_{1.32}$	$28.5_{2.13}$	$29.0_{1.37}$	$24.5_{0.39}$
	LA	$17.2_{1.2}$	$16.2_{0.5}$	$24.4_{0.42}$	$28.6_{2.61}$	$30.5_{1.43}$	$29.5_{1.83}$	$30.7_{1.2}$
	MixLoRA	$12.9_{1.99}$	$19.0_{1.88}$	$14.5_{2.57}$	$26.4_{3.25}$	$33.7_{1.87}$	$30.3_{2.27}$	$28.3_{1.07}$
	UQ4CT	$\mathbf{5.0_{1.15}}$	$\mathbf{8.9_{3.46}}$	$\mathbf{6.5_{1.85}}$	$\mathbf{19.6_{2.90}}$	$\mathbf{23.1_{1.17}}$	$\mathbf{25.9_{3.43}}$	$\mathbf{21.9_{3.49}}$

to professionals. This domain-specificity demonstrates larger distribution shift from OBQA, which is a general common sense reasoning task. Details of the ensemble is provided in Appendix. A.6.

The distribution shift evaluations are provided in Table 2. UQ4CT provides substantial improvements in terms of ECE while maintains similar accuracy for both smaller and larger distribution shift scenarios. For smaller distribution shifts, UQ4CT shows comparable ECE performance as the in-distribution scenario. For the more challenging larger distribution shifts, UQ4CT still achieves the best ECE performance among all baseline models. Note that UQ4CT also achieves competitive prediction accuracy across all domain-specific tasks. This empirically shows that our proposed alignment of the functional epistemic uncertainty with predictive correctness improves generalizability and mitigating the overconfidence problem on the fine-tuned model.

5.4 Ablation Study

In this section, we conduct ablation studies to investigate the effectiveness of our designed calibration loss, $\mathcal{L}_{cal}$. We first evaluate the incremental weighting performance of the calibration term, which investigates the effectiveness of $\mathcal{L}_{cal}$ at the early stage of fine-tuning. Then, we perform a sensitivity test, where we explore the overall performance impact of $\mathcal{L}_{cal}$. We also conduct an ablation study on the impact of active LoRA experts in Appendix A.3.

5.4.1 Incremental Weighting on Calibration Term

Due to the random initialization of LoRA experts, the predictions during early fine-tuning stage are likely to be incorrect as the model has little knowledge on the functional relationships regarding the data. Thus, it is intuitive to incrementally increase the weight parameter β over the calibration term $\mathcal{L}_{\text{cal}}$ in the training loss for the LoRA experts to learn before calibration. We conduct this study by incrementally increase β from 0 to 1 within 50 gradient steps during the early stage of fine-tuning:

$$\beta = \min\left\{1, \frac{\text{current\_grad\_step}}{50}\right\}. \tag{12}$$

We choose 50 gradient steps from our observation that training loss generally stabilizes after 50 gradient steps, indicating the LoRA experts have learned some functional relationships from data.

As shown in Table 3, the incremental loss has significantly worse ECE performance across all tasks. This demonstrates the advantage of uncertainty calibration even in the early stage. In the beginning, the lack of functional relationships on the training data in LoRA experts lead to high epistemic uncertainty. Thus, UQ4CT encourages exploration over all LoRA experts while UQ4CT_Incremental lacks it due to the small weighting in the beginning.

Table 3: Performance comparison of UQ4CT with and without incremental weighting. Incremental weighting has worse ECE performance while maintains similar accuracy.

Metrics	Methods	BoolQ	ARC-E	ARC-C	OBQA	ClimateQA
ACC ↑	UQ4CT	$73.5_{0.52}$	$76.6_{1.30}$	$52.8_{1.77}$	$77.3_{1.36}$	$63.3_{1.74}$
	UQ4CT_Incremental	$72.0_{0.19}$	$75.4_{0.81}$	$54.6_{0.95}$	$77.6_{0.43}$	$60.2_{3.17}$
ECE ↓	UQ4CT	$\mathbf{2.3_{0.82}}$	$\mathbf{6.0_{0.2}}$	$\mathbf{6.1_{1.11}}$	$\mathbf{5.0_{1.15}}$	$\mathbf{8.1_{0.52}}$
	UQ4CT_Incremental	$4.0_{0.16}$	$9.8_{1.51}$	$13.8_{2.08}$	$10.3_{1.73}$	$12.2_{0.88}$

Table 4: Performance of UQ4CT with varying β value on OBQA dataset. Prediction accuracy and uncertainty alignment increases with β, highlighting the effectiveness of the calibration term.

β	ACC ↑	ECE ↓
0	$75.5_{2.91}$	$12.9_{1.99}$
0.2	$76.0_{0.6}$	$7.47_{0.78}$
0.5	$75.9_{0.31}$	$7.82_{0.93}$
0.8	$76.6_{0.4}$	$5.94_{1.16}$
1	$\mathbf{77.3_{1.36}}$	$\mathbf{5.0_{1.15}}$

5.4.2 Sensitivity Test on Calibration Term

To further understand the effectiveness of the calibration loss, we perform a sensitivity test of the coefficient β in Equation 11. This evaluates how our proposed calibration of parameter mixtures affect the overall model prediction and uncertainty quantification capabilities. We evaluate β values of $0, 0.2, 0.5, 0.8$ and 1, where $\beta = 0$ resembles the original MixLoRA method.

Results in Table 4 demonstrate the effectiveness of the calibration loss. When $\beta = 0$, the model is optimized without calibration on parameter mixtures, resulting in high ECE value. Even with small $\beta = 0.2$ or $\beta = 0.5$, the ECE scores drastically improved compared to no calibration setting. Finally, when $\beta = 1$, the calibration term effectively optimizes the conditional parameter mixtures to generate outputs that fit data distribution well, resulting in lower ECE scores and higher accuracies.

6 Discussion & Conclusion

In this work, we propose Functional-Level Uncertainty Quantification for Calibrated Fine-Tuning (UQ4CT), which addresses the overconfidence issues commonly encountered during fine-tuning of large language models. We present a functional perspective on quantifying epistemic uncertainty in LLMs and utilize it for uncertainty-calibrated fine-tuning. By incorporating functional-level epistemic uncertainty quantification with a mixture-of-experts framework, our proposed uncertainty-calibrated training loss effectively addresses the challenge of overconfidence in fine-tuned LLMs by significantly improving uncertainty calibration while maintaining high accuracy. Our evaluations demonstrate that UQ4CT reduces the Expected Calibration Error by more than 25% without compromising accuracy across a variety of downstream tasks, including common-sense and domain-specific reasoning, under in-distribution and out-of-distribution scenarios.

The limitation of UQ4CT lies in its dependency on predictive correctness. For general language modeling tasks such as chat completion, there lacks a clear metric on whether the response is correct or not. This limits the application of UQ4CT as naively token matching is a poor indicator of semantic correctness due to the ambiguous nature of language. For future work, we are exploring ways to adapt UQ4CT on open-ended problems that lacks a definitive optimization objective.

Reproducibility Statement

We share our experimental details in Appendix A.4, and also provide the code and model weights for running experiments in the supplementary materials to reproduce our model performance results.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023a.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv: 2303.08774*, 2023b.

Oleksandr Balabanov and Hampus Linander. Uncertainty quantification in fine-tuned llms using lora ensembles. *arXiv preprint arXiv:2402.12264*, 2024.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. URL https://api.semanticscholar.org/CorpusID:218971783.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

Hugh Chen, Scott Lundberg, and Su-In Lee. Checkpoint ensembles: Ensemble methods from a single training process. *arXiv preprint arXiv:1710.03282*, 2017.

Jiuhai Chen and Jonas Mueller. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness, 2023.

Shaoxiang Chen, Zequn Jie, and Lin Ma. Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. *arXiv preprint arXiv: 2401.16160*, 2024.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24, 2022. URL https://api.semanticscholar.org/CorpusID:247951931.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416, 2022. URL https://api.semanticscholar.org/CorpusID:253018554.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

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

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. Loramoe: Alleviate world knowledge forgetting in large language models via moe-style plugin. *arXiv*, 2024.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the uncertainty estimation of large language models, 2023.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*, 2022.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. *ArXiv*, abs/2301.00774, 2023. URL https://api.semanticscholar.org/CorpusID:255372747.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ArXiv*, abs/2210.17323, 2022. URL https://api.semanticscholar.org/CorpusID:253237200.

Yarin Gal and Zoubin Ghahramani. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference. *arXiv e-prints*, art. arXiv:1506.02158, June 2015.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016a. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

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

Adam Gleave and Geoffrey Irving. Uncertainty estimation for language reward models, 2022.

Yunhao Gou, Zhili Liu, Kai Chen, Lanqing Hong, Hang Xu, Aoxue Li, Dit-Yan Yeung, James T Kwok, and Yu Zhang. Mixture of cluster-conditional lora experts for vision-language instruction tuning. *arXiv preprint arXiv: 2312.12379*, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Guande He, Jianfei Chen, and Jun Zhu. Preserving pre-trained features helps calibrate fine-tuned language models. In *ICLR*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. Training compute-optimal large language models. *ArXiv*, abs/2203.15556, 2022. URL https://api.semanticscholar.org/CorpusID:247778764.

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

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling, 2023.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021a.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021b. URL https://api.semanticscholar.org/CorpusID:235458009.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. Look Before You Leap: An Exploratory Study of Uncertainty Measurement for Large Language Models, October 2023. URL http://arxiv.org/abs/2307.10236. arXiv:2307.10236 [cs].

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

Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. Fedpara: Low-rank hadamard product for communication-efficient federated learning. *arXiv preprint arXiv: 2108.06098*, 2021.

Srinivas Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Veselin Stoyanov. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *ArXiv*, abs/2212.12017, 2022. URL https://api.semanticscholar.org/CorpusID:255096269.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 1991.

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

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki Markus Asano. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv: 2310.11454*, 2023.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation, April 2023. URL http://arxiv.org/abs/2302.09664. arXiv:2302.09664 [cs].

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

Andrew Kyle Lampinen, Stephanie C. Y. Chan, Ishita Dasgupta, Andrew J. Nam, and Jane X. Wang. Passive learning of active causal strategies in agents and language models, May 2023.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *ICLR*, 2020.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts. *arXiv preprint arXiv:2404.15159*, 2024.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212, 2022.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching Models to Express Their Uncertainty in Words. *arXiv e-prints*, art. arXiv:2205.14334, May 2022. doi: 10.48550/arXiv.2205.14334.

Yong Lin, Lu Tan, Hangyu Lin, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Han Zhao, Yuan Yao, et al. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. *arXiv preprint arXiv:2309.06256*, 2023.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with Confidence: Uncertainty Quantification for Black-box Large Language Models. *arXiv e-prints*, art. arXiv:2305.19187, May 2023. doi: 10.48550/arXiv.2305.19187.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv: 2402.09353*, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zechun Liu, Barlas Oğuz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. *ArXiv*, abs/2305.17888, 2023. URL `https://api.semanticscholar.org/CorpusID:258959117`.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv: 2402.12851*, 2024.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *ArXiv*, abs/2305.11627, 2023. URL `https://api.semanticscholar.org/CorpusID:258823276`.

D. J. C. Mackay. Information-based objective functions for active data selection. *Neural Computation*, 4(2):550–604, 1992.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks, 2018.

Andrey Malinin and Mark Gales. Uncertainty Estimation in Autoregressive Structured Prediction, February 2021. URL `http://arxiv.org/abs/2002.07650`. arXiv:2002.07650 [cs, stat].

Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. Uncertainty in gradient boosting via ensembles, 2021.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022. doi: 10.1162/tacl_a_00494. URL https://aclanthology.org/2022.tacl-1.50.

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

OpenAI. GPT-4 technical report, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Adithya Renduchintala, Tugrul Konuk, and Oleksii Kuchaiev. Tied-lora: Enhacing parameter efficiency of lora with weight tying. *arXiv preprint arXiv: 2311.09578*, 2023.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large Language Models Encode Clinical Knowledge. *arXiv*, 2022.

Meiqi Sun, Wilson Yan, Pieter Abbeel, and Igor Mordatch. Quantifying uncertainty in foundation models via ensembles. In *NeurIPS 2022 Workshop on Robustness in Sequence Modeling*, 2022. URL https://openreview.net/forum?id=LpBlkATV24M.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023a. URL https://api.semanticscholar.org/CorpusID:257219404.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023c.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023d. URL https://api.semanticscholar.org/CorpusID:259950998.

Dustin Tran, Jeremiah Liu, Michael W. Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, Neil Band, Tim G. J. Rudner, Karan Singhal, Zachary Nado, Joost van Amersfoort, Andreas Kirsch, Rodolphe Jenatton, Nithum Thain, Honglin Yuan, Kelly Buchanan, Kevin Murphy, D. Sculley, Yarin Gal, Zoubin Ghahramani, Jasper Snoek, and Balaji Lakshminarayanan. Plex: Towards reliability using pretrained large model extensions, 2022.

Xi Wang, Laurence Aitchison, and Maja Rudolph. Lora ensembles for large language model fine-tuning. *arXiv preprint arXiv:2310.00035*, 2023.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: An alternative approach to efficient ensemble and lifelong learning, 2020.

Haoyuan Wu, Haisheng Zheng, and Bei Yu. Parameter-efficient sparsity crafting from dense to mixture-of-experts for instruction tuning on general tasks. *arXiv preprint arXiv: 2401.02731*, 2024a.

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, May 2023a.

Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136, 2023b.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*, 2024b.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. *ArXiv*, abs/2211.10438, 2022a. URL https://api.semanticscholar.org/CorpusID:253708271.

Guangxuan Xiao, Ji Lin, and Song Han. Offsite-tuning: Transfer learning without full model. *ArXiv*, abs/2302.04870, 2023. URL https://api.semanticscholar.org/CorpusID:256697131.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 7273–7284, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.538. URL https://aclanthology.org/2022.findings-emnlp.538.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In *EMNLP*, 2022c.

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

Shu Yang, Muhammad Asif Ali, Cheng-Long Wang, Lijie Hu, and Di Wang. Moral: Moe augmented lora for llms' lifelong learning. *arXiv preprint arXiv: 2402.11260*, 2024b.

Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles. *arXiv preprint arXiv:2401.00243*, 2023a.

Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles, 2023b.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*, 2023.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *NeurIPS*, 36, 2024.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. *arXiv preprint arXiv:2104.04670*, 2021.

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

Table 5: Performance comparison of different methods fine-tuned with Mistral-7B across 4 common sense reasoning tasks and a domain-specific task. UQ4CT shows significant ECE improvements while maintaining high accuracy.

Metrics	Methods	BoolQ	ARC-E	ARC-C	OBQA	ClimateQA
ACC ↑	LoRA	$70.3_{0.62}$	$84.8_{0.47}$	$70.2_{0.84}$	$82.8_{0.62}$	$72.5_{1.6}$
	MC Drop	$69.6_{1.07}$	$84.6_{0.91}$	$69.6_{0.76}$	$82.6_{0.71}$	$72.5_{1.6}$
	Ensemble	$71.8_{1.29}$	$84.2_{0.66}$	$71.0_{1.41}$	$82.5_{0.6}$	$72.9_{2.88}$
	LA	$70.7_{1.82}$	$82.4_{2.05}$	$68.5_{3.31}$	$82.5_{0.77}$	$71.6_{1.56}$
	MixLoRA	$73.1_{0.38}$	$85.5_{1.27}$	$71.2_{1.75}$	$83.3_{1.14}$	$72.0_{1.69}$
	UQ4CT	$73.6_{0.28}$	$85.9_{0.82}$	$74.4_{0.82}$	$83.7_{1.22}$	$73.2_{1.29}$
ECE ↓	LoRA	$10.17_{0.24}$	$9.46_{1.62}$	$18.42_{1.91}$	$13.3_{0.25}$	$13.72_{2.62}$
	MC Drop	$10.62_{0.51}$	$8.91_{1.35}$	$18.38_{1.66}$	$13.3_{0.31}$	$13.72_{2.61}$
	Ensemble	$8.72_{1.13}$	$8.72_{1.49}$	$17.0_{0.97}$	$9.14_{2.82}$	$12.86_{1.78}$
	LA	$5.33_{2.16}$	$20.3_{5.7}$	$21.27_{4.15}$	$\mathbf{6.41_{3.22}}$	$14.64_{2.21}$
	MixLoRA	$8.81_{1.03}$	$8.16_{0.99}$	$15.51_{3.86}$	$10.53_{1.73}$	$14.05_{3.09}$
	UQ4CT	$\mathbf{3.07_{0.83}}$	$\mathbf{5.7_{0.69}}$	$\mathbf{7.04_{0.58}}$	$\underline{7.92_{1.14}}$	$\mathbf{11.4_{1.14}}$

A Appendix

A.1 Load Balancing Loss

We follow the load balancing loss in (Li et al., 2024). Given N experts indexed by $i = 1$ to N and a batch B with T tokens, the auxiliary loss is computed as:

$$\mathcal{L}_{aux} = a \cdot N \cdot \sum_{i=1}^{N} \mathcal{F}_i \cdot \mathcal{P}_i, \tag{13}$$

where

$$\mathcal{F}_i = \frac{1}{T} \sum_{x \in B} \mathbb{1}\{argmax_k \mathcal{R}(x)_k = i\}, \mathcal{P}_i = \frac{1}{T} \sum_{x \in B} \mathcal{R}(x)_i. \tag{14}$$

Here, $\mathcal{R}(\cdot)$ is the top-k router, $\mathcal{F}_i$ is the fraction of tokens dispatched to expert i and $\mathcal{P}_i$ is the fraction of the router probability allocated for expert i. The final loss is multiplied by the expert count N to keep the loss constant as the number of experts varies, and the constant term a is set to 10^{-2}as a multiplicative coefficient, which is large enough to ensure load balancing while remaining small enough not to overwhelm the primary objective.

A.2 Experimental Results with Mistral-7B

In this section, we present the results using Mistral-7B (Jiang et al., 2023), a different decoder-based LLM backbone. Table 5 shows the results of fine-tuning Mistral-7B on 4 common-sense reasoning tasks and one domain-specific climate question-answering task.

For each of the tasks, UQ4CT effectively calibrates the parameter mixtures, leading to the best ECE performance in 4 out of 5 tasks. This indicates the robustness of UQ4CT across different LLMs.

A.3 Deciding Number of Active Experts

One important aspect of the LoRA MoE architecture is how many experts to activate. Here, we investigate the performance impact of different number of active LoRA experts. We evaluate the model performance with 1 to 5 active experts with 8 in total.

As shown in Table 6, 2 active experts give the optimal performance in terms of accuracy and ECE scores. One expert alone cannot capture complicated functional relationships, while more than 2 experts could potentially introduce redundant functional bases to the model, which deviates the output distribution more from data distribution, thus worsening predictive and calibration performance. Additionally, more active experts lead to a more flattened distribution across experts, which hardens the alignment of parameter mixtures during fine-tuning.

Table 6: Performance comparison of UQ4CT with varying number of experts on OBQA dataset. Top-2 expert selection strategy grants best accuracy and calibration.

Top-K	ACC ↑	ECE ↓
Top-1	$74.8_{0.62}$	$7.69_{1.96}$
Top-2	$\mathbf{77.3_{1.36}}$	$\mathbf{5.0_{1.15}}$
Top-3	$75.2_{0.8}$	$5.8_{0.81}$
Top-4	$75.8_{0.53}$	$7.67_{0.46}$
Top-5	$75.3_{0.5}$	$6.29_{0.61}$

A.4 Training Details

We train our model with total of 8 LoRA experts, and select 2 experts with the highest probability. For each expert, we use $rank = 16$ and $alpha = 32$. We use batch size of 16 to train our model. For climate task, we set the learning rate to $5e-4$ and dropout rate to 0.1 to incorporate the small dataset size. For other tasks, we use $2e-4$ as our learning rate with dropout 0.05. We use AdamW as our optimizer and a cutoff length of 512 for prompts during training.

The experimental setup for single LoRA based models is similar with LoRA ranks set to 80 to accommodate the MoE model size. For the ensemble baseline, we use an ensemble size of 8 with $rank = 16$. For Laplace-LoRA, we follow the Laplace hyperparameters in this Github Repository.

A.5 Expected Calibration Error

Expected calibration error (ECE) is a commonly used metric to asses uncertainty quantification performance. ECE measures the alignment between prediction accuracy and model confidence through regrouping the predicted probabilities into m bins. This method then computes the weighted average of the difference between average accuracy and confidence in each bin:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)|, \tag{15}$$

where $|B_m|$ is the number of evaluated datapoints in bin m, acc and conf is calculated as following:

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i), \tag{16}$$

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} P(\hat{y}_i). \tag{17}$$

A.6 MMLU Distribution Shift Dataset Composition

- **Computer Science (CS)**:
  - College Computer Science
  - Computer Security
  - High School Computer Science
  - Machine Learning
- **Engineering (Eng)**:
  - Electrical Engineering
- **Law**:
  - International Law
  - Jurisprudence
  - Professional Law
- **Health**:

- Anatomy
- Clinical Knowledge
- College Medicine
- Human Aging
- Nutrition
- Professional Medicine
- Virology