TRAINING-FREE BAYESIANIZATION FOR LOW-RANK ADAPTERS OF LARGE LANGUAGE MODELS

Haizhou Shi[*1] **Yibin Wang**[*2] **Ligong Han**[3] **Huan Zhang**[2] **Hao Wang**[1]

ABSTRACT

Estimating the uncertainty of responses of Large Language Models (LLMs) remains a critical challenge. While recent Bayesian methods have demonstrated effectiveness in quantifying uncertainty through low-rank weight updates, they typically require complex fine-tuning or post-training procedures. In this paper, we propose **T**raining-**F**ree **B**ayesianization (`TFB`), a novel framework that efficiently transforms existing off-the-shelf trained low-rank adapters into Bayesian ones without additional training. `TFB` systematically searches for the maximally acceptable level of variance in the weight posterior, constrained within a family of low-rank isotropic Gaussian distributions. We theoretically demonstrate that under mild conditions, this search process is equivalent to KL-regularized variational optimization, a generalized form of variational inference. Through comprehensive experiments, we show that `TFB` achieves superior uncertainty estimation and generalization compared to existing methods while eliminating the need for complex training procedures.

1 INTRODUCTION

Despite recent advances in Large Language Models (LLMs) showing great capacity for generating responsive answers to human instructions (Biderman et al., 2023; Wei et al., 2022; 2021; Min et al., 2022; Chowdhery et al., 2023; Anil et al., 2023; Touvron et al., 2023a;b; Radford et al., 2019; Brown et al., 2020; Achiam et al., 2023; OpenAI, 2022), the reliability of such large models remains a critical concern (Wang et al., 2024b;a), as untruthful yet confident answers could cause significant damage to individuals and society (Gupta et al., 2024; Nikitin et al., 2024; Yadkori et al., 2024; Kapoor et al., 2024). The accurate estimation of uncertainty in LLMs has thus emerged as an urgent challenge. Current approaches mainly follow two paths: one focuses on directly asking the model to elicit its internal internal (verbalized) uncertainty (Xiong et al., 2023; Tian et al., 2023; Kapoor et al., 2024), while the other employs complex fine-tuning techniques (Yang et al., 2023; Wang et al., 2024c).

Both approaches suffer from inherent limitations. Verbalized uncertainty, while simple to implement, remains controversial in terms of its empirical reliability and ***theoretical soundness*** (Kadavath et al., 2022; Kuhn et al., 2023). On the other hand, low-rank adapters (LoRA (Hu et al., 2022)), which offer a parameter-efficient way to adapt LLMs by adding a small set of low-rank weight matrices, have emerged as a promising direction for fine-tuning models. However, while LoRA efficiently adapts large models to new tasks, it does not itself provide a mechanism for principled uncertainty estimation. In response, recent Bayesianization attempts, such as BLoB (Wang et al., 2024c), integrate Bayesian methods with LoRA, but they still require complex training procedures and sophisticated hyperparameter tuning, ***limiting their practicality***. These constraints motivate the research question:

*Can we "Bayesianize" an LLM's low-rank adapter in a way that is both **theoretically sound** and **empirically simple**?*

In this paper, we diverge from conventional fine-tuning and post-training approaches. Instead, we develop a Training-Free Bayesianization (`TFB`) technique applicable to *any* given low-rank LLM adapter. `TFB` constrains the family of full-weight approximate posteriors produced by LoRA adapters to low-rank isotropic Gaussian distributions. Given a trained LoRA adapter, it systematically searches

[*]Equal Contribution. [1]Rutgers University. [2]University of Illinois Urbana-Champaign (UIUC). [3]MIT-IBM Watson AI Lab. Correspondence to: Haizhou Shi <haizhou.shi@rutgers.edu>, Hao Wang <hw488@cs.rutgers.edu>.

for the maximally acceptable variance of the variational distribution of the weight posterior, without the need for complex fine-tuning procedures. `TFB`'s search range and stopping criteria can be determined using *any* in-distribution "anchor dataset," e.g., a small subset of the training dataset. Note that (i) this eliminates the need for an additional calibration or validation dataset; (ii) this flexibility extends to both supervised and unsupervised data, even regardless of whether it was used in the original LoRA training. Despite its simplicity, we theoretically demonstrate that, `TFB`'s process of finding the maximal variance of the low-rank isotropic Gaussian posterior is equivalent to generalized variational inference, under mild conditions.

We verify `TFB`'s effectiveness through extensive empirical evaluation across various settings, datasets, LLM backbones, LoRA weights, and LoRA variants. Our comprehensive experiments demonstrate that this novel training-free Bayesianization framework consistently achieves superior generalization and more accurate uncertainty estimation. To summarize, the main contributions of this paper are:

- We propose Training-Free Bayesianization (`TFB`), the first framework to transform trained LoRA adapters into Bayesian ones without fine-tuning or gradient estimation.
- We establish theoretical connections between `TFB` and generalized variational inference, proving their equivalence under mild conditions.
- We develop an efficient implementation of `TFB` requiring only an anchor dataset for search, making it widely applicable across different application scenarios.
- Through comprehensive experiments, we demonstrate that `TFB` consistently improves uncertainty estimation for off-the-shelf LoRA adapters, and overall surpasses the state-of-the-art counterparts of Bayesian LoRA.

2 Related Work

LLM Uncertainty Estimation. To estimate the uncertainty of LLMs, the models themselves are often employed to generate and evaluate their own uncertainty (Lin et al., 2022; Kadavath et al., 2022). However, such approaches typically rely on task-specific labels and require additional training. Semantic entropy (Kuhn et al., 2023) leverages the invariance of language stemming from shared meanings to estimate uncertainty, while mutual information is used to compute a lower bound on model uncertainty by sampling from the model's output distribution (Yadkori et al., 2024). Despite their contributions, these methods fail to accurately capture true model uncertainty, as they do not model the probability distribution over the LLM parameters (Hüllermeier & Waegeman, 2021; Abdar et al., 2021; Gawlikowski et al., 2023).

Bayesian Low-Rank Adaptation. The Bayesian framework provides a powerful approach for capturing and estimating uncertainty during fine-tuning by defining prior distributions and approximating posterior distributions over the model parameters (Neal, 2012; Hernández-Lobato & Adams, 2015; Gal & Ghahramani, 2016; Wang & Yeung, 2016). Recent research has explored combining Bayesian methods with LoRA to mitigate the additional computational overhead associated with modeling parameter distributions across the entire parameter space. Yang et al. (2023) applies a Kronecker-factorized Laplace approximation to fine-tuned LoRA parameters. More recently, BLoB (Wang et al., 2024c) advances the field by simultaneously estimating both the mean and covariance of LLM parameters within a single fine-tuning stage. Our proposed training-free Bayesianization represents a significant departure from these existing methods. Unlike approaches that require re-training (Gal & Ghahramani, 2016; Wang et al., 2023; Balabanov & Linander, 2024; Wang et al., 2024c) or rely on continued training and gradient estimation (Yang et al., 2023), our method achieves uncertainty estimation without any additional training steps, substantially improving the simplicity and efficiency for Bayesian learning of LLMs.

3 Methodology

This section presents our Training-Free Bayesianization (`TFB`), covering problem setup (Sec. 3.1), our low-rank Gaussian posterior formulation (Sec. 3.2), and a novel approach for converting deterministic weights to probabilistic distributions without training (Sec. 3.3). The complete algorithmic implementation is provided (Sec. 3.4), with theoretical foundations addressed in a separate section (Sec. 4).

Notation. Scalars, vectors, and matrices are denoted by lowercase letters, lowercase boldface letters, and uppercase boldface letters, respectively. For a matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{m\times n}$, we use $\text{vec}(\boldsymbol{X}) = [\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \cdots, \boldsymbol{x}_n^\top]^\top \in \mathbb{R}^{(mn)\times 1}$ to denote vectorization. $\otimes$ and $\circ$ denote the Kronecker and element-wise product, respectively. We use $\mathbf{0}_n \in \mathbb{R}^{n\times n}$ to denote a zero matrix.

3.1 Preliminaries

Low-Rank Adaptation (LoRA). Given a pre-trained neural network layer with weight matrix $\boldsymbol{W}_0$, Low-Rank Adaptation (LoRA) (Hu et al., 2022) confines weight updates to a low-rank subspace during fine-tuning, expressing the update as $\Delta\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$, where $\Delta\boldsymbol{W} \in \mathbb{R}^{m\times n}$, $\boldsymbol{B} \in \mathbb{R}^{m\times r}$, and $\boldsymbol{A} \in \mathbb{R}^{r\times n}$. For input $\boldsymbol{h}$ and output $\boldsymbol{z}$ of the LoRA layer, the forward pass computation is given by:

$$\boldsymbol{z} = \boldsymbol{W}_0\boldsymbol{h} + \Delta\boldsymbol{W}\boldsymbol{h} = \boldsymbol{W}_0\boldsymbol{h} + \boldsymbol{B}\boldsymbol{A}\boldsymbol{h}. \tag{1}$$

LoRA Bayesianization with Low-Rank Gaussian Distribution. BLoB (Wang et al., 2024c), a pioneering work in low-rank Bayesianization for LLMs, empirically demonstrates that modeling $\boldsymbol{A}$'s elements with independent Gaussian variables suffices for effective uncertainty estimation in LoRA. Specifically, the probability density of each element of $\boldsymbol{A}$ follows $q(A_{ij}) = \mathcal{N}(A_{ij}|M_{ij}, \Omega_{ij}^2), \forall i \in [r], \forall j \in [n]$, where matrices $\boldsymbol{M}$ and $\boldsymbol{\Omega}$, sharing the dimensions of $\boldsymbol{A}$, represent the mean and standard deviation of the random variable $\boldsymbol{A}$, respectively. This formulation is equivalent to approximating the Bayesianized low-rank adapter's posterior in the full-weight space of $\boldsymbol{W}$ with a low-rank degenerate distribution, i.e.,

$$q(\text{vec}(\boldsymbol{W})|\boldsymbol{B}, \boldsymbol{\theta}) = \mathcal{N}(\text{vec}(\boldsymbol{W})|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q), \tag{2}$$

where $\boldsymbol{\theta} = \{\boldsymbol{M}, \boldsymbol{\Omega}\}$ denotes the set of parameters modeling $\boldsymbol{A}$'s variational distribution, $\boldsymbol{\mu}_q = \text{vec}(\boldsymbol{W}_0 + \boldsymbol{B}\boldsymbol{M})$ is its mean, and $\boldsymbol{\Sigma}_q = [\boldsymbol{I}_n \otimes \boldsymbol{B}][\text{diag}(\text{vec}(\boldsymbol{\Omega})^2)][\boldsymbol{I}_n \otimes \boldsymbol{B}^\top]$ is its covariance. In this paper, we adopt a similar approach for modeling the variational distribution of the weight posterior, focusing exclusively on Bayesianizing the weight update matrix $\boldsymbol{A}$.

3.2 TFB's Variational Low-Rank Isotropic Gaussians

Variational Distribution Family. In TFB, we constrain the variational distributions of the weight posterior to a more compact family of Gaussians than BLoB: specifically, we employ full-space isotropic Gaussian distributions projected onto the low-rank space:

$$q(\text{vec}(\boldsymbol{W})|\boldsymbol{B}, \boldsymbol{\theta}) = \mathcal{N}(\text{vec}(\boldsymbol{W})|\boldsymbol{\mu}_q, \text{proj}(\sigma_q^2\boldsymbol{I})), \tag{3}$$

where $\boldsymbol{\mu}_q$ is defined as in Eqn. 2. Here, $\sigma_q^2\boldsymbol{I} \in \mathbb{R}^{mn\times mn}$ represents a full-rank isotropic covariance matrix with standard deviation σ_q, and $\text{proj}(\cdot)$ denotes a linear projection operator that maps the full-space covariance matrix onto the low-rank space (see Proposition D.1.1 for details). [1]

TFB as Generalized Variational Inference. The choice of low-rank isotropic Gaussian approximate posteriors serves both *theoretical* and *empirical* purposes: it provides a single-parameter family that enables converting the generalized variational inference into a variance maximization problem (more details in Sec. 3.3 and Theorem 4.2), and empirically outperforms alternative distribution families (Sec. 5.3). Below, we present a practically efficient implementation for Bayesianizing LoRA under the constraint specified in Eqn. 3, with detailed theoretical analysis provided in Theorem 4.1.

TFB in Practice. Consider a LoRA layer with weight updates $\boldsymbol{B} \in \mathbb{R}^{m\times r}, \boldsymbol{A} \in \mathbb{R}^{r\times n}$ and a standard deviation scale $\sigma_q > 0$. We begin by computing the compact Singular Value Decomposition (SVD) (Klema & Laub, 1980) of $\boldsymbol{B}$:

$$\boldsymbol{B} = \boldsymbol{U}\,\text{diag}(\boldsymbol{d})\boldsymbol{V}^\top, \tag{4}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times r}$ are orthonormal matrices, and $\boldsymbol{d} = [d_1, d_2, \cdots, d_r]^\top$ is the vector consisting of singular values with all positive entries[2]. We then transform the original weight matrices $\{\boldsymbol{B}, \boldsymbol{A}\}$ into an equivalent pair

$$\{\boldsymbol{B}' = \boldsymbol{U}, \boldsymbol{A}' = \boldsymbol{V}^\top\boldsymbol{A}\}, \tag{5}$$

maintaining the equality $\Delta\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A} = \boldsymbol{B}'\boldsymbol{A}'$. Following BLoB's Asymmetric Bayesianization scheme, we define the variational distribution for $\boldsymbol{A}'$ using the mean matrix $\boldsymbol{M} = \boldsymbol{A}'$ and the standard deviation matrix $\boldsymbol{\Omega} \in \mathbb{R}^{r\times n}$, such that

$$q(A'_{ij}) = \mathcal{N}(A'_{ij}|M_{ij}, \Omega_{ij}^2), \forall i \in [r], \forall j \in [n]. \tag{6}$$

[1] $\text{proj}(\cdot)$ only depends on the rank r of the trained LoRA.

[2] By stating $\boldsymbol{d} \succ \mathbf{0}$, we assume $\boldsymbol{B}$ has the full column rank r, which usually holds for LLM adaptation.

Unlike BLoB, our $\boldsymbol{\Omega}$ is not freely parameterized but instead derived from projecting the full-space matrix $\sigma_q \boldsymbol{I}$ onto the low-rank weight space:

$$\Omega_{ij} = \sigma_q/d_i, \quad \forall i \in [r], \forall j \in [n], \tag{7}$$

where $\boldsymbol{d}$ is defined in Eqn. 4. This solution can be expressed compactly as $\boldsymbol{\Omega} = [\sigma_q/\boldsymbol{d}, \cdots, \sigma_q/\boldsymbol{d}]$, comprising n repeated vectors. To summarize, our `TFB`

- **takes as input** a trained LoRA matrix pair $\{\boldsymbol{B} = \boldsymbol{U}\operatorname{diag}(\boldsymbol{d})\boldsymbol{V}^\top, \boldsymbol{A}\}$ and a predetermined standard deviation σ_q, and
- **outputs** a "Bayesianized" LoRA adapter $\{\boldsymbol{B}', \boldsymbol{A}'\}$, where $\boldsymbol{B}' = \boldsymbol{U}\operatorname{diag}(\boldsymbol{d})$, and $\boldsymbol{A}'$ becomes a distribution $q(\boldsymbol{A}') = \prod_{i\in[r],j\in[n]} \mathcal{N}(A'_{ij}|M_{ij}, \Omega^2_{ij})$, with $\boldsymbol{M} = \boldsymbol{V}^\top \boldsymbol{A}$, and $\boldsymbol{\Omega} = [\sigma_q/\boldsymbol{d}, \cdots, \sigma_q/\boldsymbol{d}]$.

Note that the formulation in Eqn. 7 significantly improves memory efficiency during inference, reducing the storage for standard deviation parameters from $O(rn)$ to $O(r)$. While alternative parameterization approaches are possible, they must be capable of generating the low-rank isotropic Gaussian noises as demonstrated in Theorem 4.1. We have selected the current method to ensure maximum compatibility with existing implementations (Wang et al., 2024c). In `TFB`, we use a single σ_q shared across all LoRA layers.

3.3 Training-Free Bayesianization (`TFB`)

The previous section presents a straightforward Bayesianization scheme **for a predetermined value of σ_q.** In this section, we describe a practical method for determining σ_q.

A General Bayesianization Framework. Consider an in-distribution "anchor" dataset $\mathcal{D}$, an associated evaluation metric l, and a performance change tolerance ϵ. `TFB` determines σ_q by solving a constrained optimization problem:

$$\begin{aligned} \max \quad & \sigma_q \\ \textit{s.t.} \quad & |l(\mathcal{D}|\boldsymbol{B}', \boldsymbol{M}, \boldsymbol{\Omega}(\sigma_q)) - l(\mathcal{D}|\boldsymbol{B}, \boldsymbol{A})| \le \epsilon, \end{aligned} \tag{8}$$

where $l(\mathcal{D}|\boldsymbol{B}, \boldsymbol{A})$ and $l(\mathcal{D}|\boldsymbol{B}', \boldsymbol{M}, \boldsymbol{\Omega}(\sigma_q)) = \mathbb{E}_{\boldsymbol{E}\sim\mathcal{N}(0,\boldsymbol{\Omega}^2)}[l(\mathcal{D}|\boldsymbol{B}', \boldsymbol{M} + \boldsymbol{E})]$ denote the pre- and post-Bayesianization performance, respectively. This optimization maximizes the noise scale σ_q applied to model weights $\boldsymbol{M}$ while ensuring that the resulting performance change remains within an acceptable threshold ϵ.

Anchor Dataset $\mathcal{D}$ and Evaluation Metric l. Our *general* `TFB` framework accommodates various choices of anchor dataset $\mathcal{D}$ and evaluation metric l based on practical requirements. Below, we consider two key scenarios (with N being slightly overloaded in its notation).

For supervised dataset $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$: The Negative-Log Likelihood (NLL) serves as a natural evaluation metric: $l_{\text{nll}}(\mathcal{D}) = -\frac{1}{N}\sum_{n=1}^N \log P_{\boldsymbol{\theta}}(y_n|\boldsymbol{x}_n)$, as it theoretically corresponds to minimizing the KL-regularized variational objective (more details in Sec. 4). The anchor dataset $\mathcal{D}$ can be either the original training set used for the LoRA model or an independent calibration dataset, as commonly employed in calibration-based methods (Guo et al., 2017; Zhao et al., 2021). Alternative evaluation metrics such as accuracy or F1 score are also readily applicable. In our experimental setup, to ensure fair comparisons across uncertainty estimation baselines, we use the original training data as $\mathcal{D}$ (maintaining the same information access as baselines) and employ NLL as the evaluation metric. Additional results with accuracy as l can be found in Appendix F.2.

For unsupervised dataset $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$: One can generate pseudo-labels $\widehat{y}$ using the model before Bayesianization, effectively converting the problem to the supervised case with $\mathcal{D} = \{\boldsymbol{x}_n, \widehat{y}_n\}_{n=1}^N$. Hence our `TFB` offers substantially more flexibility compared to pure calibration methods, which typically rely on a labeled unseen calibration dataset. As a general framework, `TFB` also supports alternative evaluation metrics and statistical measures specifically designed for unsupervised data.

Performance Change Tolerance ϵ. The selection of performance change tolerance ϵ is critical in `TFB`. While our experiments demonstrate that a fixed $\epsilon = 0.3\%$ for NLL or $\epsilon = 1\%$ for accuracy can achieve effective uncertainty estimation across various datasets and LoRA checkpoints, an adaptive ϵ can further improve the performance of `TFB`.

Many factors can affect the setting of ϵ, among which the most important is the property of the given LoRA checkpoint. For instance, an overfitted LoRA can typically accommodate a larger tolerance ϵ

when using the training dataset (or its subset) as the anchor dataset. Additional characteristics of the data, model, and adaptation tasks can inform the choice of ϵ as well.

3.4 TFB: FINAL ALGORITHM

Final TFB Algorithm: Automatically Determining σ_q. Our final algorithm, presented in Algorithm 1, employs a binary search strategy to determine the optimal σ_q^* within an initial range $[\sigma_{q_{\min}}, \sigma_{q_{\max}}]$. After identifying the optimal σ_q^*, we Bayesianize all LoRA layers using this value.

Prediction. After Bayesianizing the LLM, for prediction, we sample multiple times from the weight posterior and average the output:

$$\begin{aligned} P_{\boldsymbol{\theta}}(y|\boldsymbol{x}) &= \mathbb{E}_{q(\boldsymbol{W}|\boldsymbol{\theta})}[P(y|\boldsymbol{x}, \boldsymbol{W})] \\ &\approx \tfrac{1}{N}\sum_{n=1}^{N} P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W}_n), \quad \boldsymbol{W}_n \sim q(\boldsymbol{W}|\boldsymbol{\theta}), \end{aligned} \tag{9}$$

where $q(\boldsymbol{W}|\boldsymbol{\theta})$ denotes the variational distribution defined in Eqn. 3, and we set the number of test-time samples to $N = 10$, following BLoB's protocol (Wang et al., 2024c).

Remark on TFB's Efficiency. While TFB with binary search is efficient in terms of both time and memory (Appendix F.1), and yields near-optimal solution of σ_q[3], more efficient parallel searching technique can be applied in practice. For instance, in Appendix F.7, we conduct a grid search across 8 different σ_q values in parallel, construct an approximate function $\widehat{\sigma}_q(p)$ through piecewise linear interpolation of the observed performance, and estimate $\sigma_q^* \approx \widehat{\sigma}_q(p_0 - \epsilon)$, where p_0 denotes the model's performance before TFB.

4 THEORETICAL ANALYSIS

In this section, we discuss our theoretical analysis, with complete proofs provided in Appendix D. First, we demonstrate that our TFB's Bayesianization scheme, defined in Equations 4, 5, and 7, projects a full-rank isotropic Gaussian distribution onto the low-rank space. We then prove that Eqn. 8 is equivalent to generalized variational inference for LLMs' weights under specific, achievable conditions, offering solid theoretical grounding for TFB.

Assumption 4.1. *The evaluation metric $l_{\mathcal{D}} : \mathbb{R}_+ \to \mathbb{R}_+$ is the Negative Log Likelihood (NLL) evaluated on the data distribution $\mathcal{D}$ for the variational standard deviation σ_q:*

$$l_{\mathcal{D}}(\sigma_q) = -\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}, \boldsymbol{W}\sim q(\cdot|\sigma_q)}[\log P(y|\boldsymbol{x}, \boldsymbol{W})]. \tag{10}$$

Furthermore, we assume $l_{\mathcal{D}}$ is locally convex, i.e., there exists $\epsilon_0 > 0$ such that $l''_{\mathcal{D}}(\sigma_q) > 0$, for all $\sigma_q \in [0, \epsilon_0)$.

Remark. *The local convexity of the loss function is not an unrealistic assumption (Milne, 2019). For instance, a local minimum $\boldsymbol{W}_0$ of a twice-differentiable loss function l will imply the local convexity around $\boldsymbol{W}_0$, which has been widely assumed in Laplace Approximation (Tierney & Kadane, 1986; Bishop, 2006).*

Theorem 4.1 (Equivalent Variational Distribution of the Full Weight $\boldsymbol{W}$ in TFB). *With the pre-trained weight matrix $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$, the low-rank weight update matrix $\{\boldsymbol{B}' \in \mathbb{R}^{m\times r}, \boldsymbol{A}' \in \mathbb{R}^{r\times n}\}$ transformed from the given matrices $\{\boldsymbol{B}, \boldsymbol{A}\}$ following Eqn. 4 and 5, suppose that the variational distribution of $\boldsymbol{A}'$ is Gaussian $q(\boldsymbol{A}'|\boldsymbol{\theta}) = \prod_{ij} \mathcal{N}(A_{ij}|M_{ij}, \Omega_{ij}^2)$, where $\boldsymbol{M} = [M_{ij} = A'_{ij}] \in \mathbb{R}^{r\times n}$ is its mean and $\boldsymbol{\Omega} = [\Omega_{ij}] \in \mathbb{R}^{r\times n}$ is the standard deviation calculated as in Eqn. 7. The equivalent variational distribution $q(\mathrm{vec}(\boldsymbol{W})|\sigma_q)$ defined on the full weight $\boldsymbol{W}$ is*

$$\begin{aligned} q(\mathrm{vec}(\boldsymbol{W})|\sigma_q) &= \mathcal{N}(\mathrm{vec}(\boldsymbol{W})|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q), \\ \textit{where} \quad \boldsymbol{\mu}_q &= \mathrm{vec}(\boldsymbol{W}_0 + \boldsymbol{B}'\boldsymbol{M}), \\ \boldsymbol{\Sigma}_q &= \sigma_q^2 \boldsymbol{I}_n \otimes \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{0}_{m-r} \end{bmatrix}. \end{aligned} \tag{11}$$

[3]While traditional search algorithms require monotonicity within the search range to guarantee optimal solutions, empirically a near-optimal σ_q is sufficient for effective uncertainty estimation.

Theorem 4.1 establishes that for any given σ_q, our algorithm for regrouping $\boldsymbol{B}, \boldsymbol{A}$ and computing the standard deviation matrix $\boldsymbol{\Omega}$ successfully constrains the corresponding full-weight variational distributions to the family of low-rank isotropic Gaussian distributions. This lays the foundation for the equivalence between our TFB and generalized variational inference to approximate the posterior distribution of LLM parameters (details in Theorem 4.2).

While alternative families of Gaussians parameterized by a single scale σ_q are possible, our empirical results demonstrate that our proposed approach achieves superior performance (Sec. 5.3).

Theorem 4.2 (TFB as Generalized Variational Inference). *Suppose the evaluation metric $l_{\mathcal{D}}(\sigma_q)$ defined following Assumption 4.1 is locally convex within the range of $\sigma_q \in [0, \epsilon_0)$. Suppose the approximate distribution of $\boldsymbol{W}$ given σ_q is defined following Theorem 4.1. Suppose we have the prior distribution $P(\text{vec}(\boldsymbol{W})) = \mathcal{N}(\text{vec}(\boldsymbol{W})|\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$, where $\boldsymbol{\mu}_p = \boldsymbol{\mu}_q = \text{vec}(\boldsymbol{W}_0 + \boldsymbol{B}'\boldsymbol{M})$, and $\boldsymbol{\Sigma}_p = \sigma_p^2 \boldsymbol{I}$ with $\sigma_p > \epsilon_0$. Then $\exists \widetilde{\epsilon} > 0$, s.t. the following two optimizations*

(i) Generalized Variational Inference (Blundell et al., 2015; Higgins et al., 2017; Khan et al., 2018)

$$\min_{\sigma_q} \quad l_{\mathcal{D}}(\sigma_q) + \lambda\, \text{KL}[q(\boldsymbol{W}|\sigma_q) \parallel P(\boldsymbol{W})], \tag{12}$$

and (ii) Training-Free Bayesianization (TFB)

$$\begin{aligned} \max \quad & \sigma_q \\ \textit{s.t.} \quad & l_{\mathcal{D}}(\sigma_q) \leq \widetilde{\epsilon}, \end{aligned} \tag{13}$$

are equivalent, where λ is the regularization coefficient of the KL-divergence.

This theorem provides the theoretical foundation for TFB. It demonstrates that under specific conditions – namely, local convexity within $[0, \epsilon_0)$ and prior standard deviation $\sigma_p > \epsilon_0$ – maximizing the scale σ_q of the standard deviation matrix is equivalent to performing generalized variational inference, which approximates the posterior distribution of LLM parameters. Notably, when $\lambda = 1/|\mathcal{D}|$ is set to the reciprocal of the dataset size, it reduces to Variational Inference (VI).

5 Experiments

We evaluate TFB through comprehensive experiments. Refer to Appendix F for more results.

5.1 Settings

Models, Datasets, and Evaluation. We use the latest open-source `Meta-Llama-3.1-8B` as our primary LLM backbone while also providing additional results on other recent LLM architectures in Appendix F.4, including `llama-2-7b-hf`, `Meta-Llama-3-8B`, and `Mistral-7B-v0.3` from the Llama (Dubey et al., 2024) and Mistral (Jiang et al., 2023) families.

For in-distribution experiments, we evaluate model performance on six commonsense reasoning tasks: Winogrande-Small (**WG-S**) and Winogrande-Medium (**WG-M**) (Sakaguchi et al., 2021), ARC-Challenge (**ARC-C**) and ARC-Easy (**ARC-E**) (Clark et al., 2018), Open Book Question Answering (**OBQA**) (Mihaylov et al., 2018), and BoolQ (Clark et al., 2019). Furthermore, we use models fine-tuned on OBQA (Mihaylov et al., 2018) to evaluate their generalization ability on out-of-distribution datasets: college-level chemistry (**Chem**) and physics (**Phy**) subsets of MMLU (Hendrycks et al., 2021). Label spaces and prompt templates are detailed in Appendix E.1.

To assess uncertainty estimation, we measure Expected Calibration Error (**ECE** (Naeini et al., 2015)) and Negative Log-Likelihood (**NLL**) on the test dataset. We also report Accuracy (**ACC**) to ensure models maintain strong performance while improving calibration. Additional evaluation details are provided in Appendix E.2.

Baselines. We compare TFB with state-of-the-art uncertainty estimation methods for LoRA-adapted LLMs, including ensemble-based method: Deep Ensemble (**ENS**) (Lakshminarayanan et al., 2017; Balabanov & Linander, 2024; Wang et al., 2023), variational inference methods: Monte-Carlo Dropout (**MCD**) (Gal & Ghahramani, 2016), Bayesian LoRA by Backprop (**BLoB**) (Wang et al., 2024c), and post-training method: Laplace-LoRA (**LAP**) (Yang et al., 2023). For reference, we also include two standard PEFT baselines: Maximum Likelihood Estimation (**MLE**) (Hu et al., 2022) and Maximum A Posteriori (**MAP**). All baselines are implemented following the protocols established in BLoB, detailed in Appendix E.4.

Table 1: **Performance of different methods applied to LoRA on Llama3.1-8B pre-trained weights,** where Accuracy (**ACC**) and Expected Calibration Error (**ECE**) are reported in percentages. **"TF?"** denotes whether a method is **T**raining-**F**ree. The evaluation is done across six common-sense reasoning tasks with a shared hyper-parameter setting after fine-tuning of 5 epochs. We use $N = 10$ samples during inference in all sampling-based methods including **BLoB Wang et al. (2024c)** and **`TFB`**. Rows with shading indicate training-free Bayesianization methods that use a pre-trained LoRA as their mean. `TFB` adopts a shared configuration for all the experiments: the anchor dataset $\mathcal{D}$ is set to a randomly sampled subset of the original training set without labels; the performance evaluation metric l is set to the NLL loss; and the performance drop tolerance ϵ is set to the relative performance change of 0.3%. "↑" and "↓" indicate that higher and lower values are preferred, respectively. **Boldface** and <u>underlining</u> denote the best and the second-best performance, respectively.

Metric	Method	TF?	In-Distribution Datasets						Out-of-Distribution Datasets (OBQA→X)			
									Small Shift		*Large Shift*	
			WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	ARC-C	ARC-E	Chem	Phy
ACC (↑)	MCD	✗	<u>78.03±0.61</u>	81.64±1.79	91.37±0.38	83.18±0.84	87.20±1.02	89.93±0.16	81.42±1.38	87.27±0.84	47.92±2.25	**46.53±0.49**
	ENS	✗	**78.82±0.52**	82.55±0.42	**91.84±0.36**	**83.99±0.74**	87.37±0.67	**90.50±0.14**	79.62±0.57	86.56±0.60	**49.65±3.22**	44.44±1.96
	LAP	BP	76.05±0.92	79.95±0.42	90.73±0.08	82.83±0.85	87.90±0.20	89.36±0.52	81.08±1.20	87.21±1.20	<u>48.26±3.93</u>	<u>46.18±1.30</u>
	MLE	-	77.87±0.54	81.08±0.48	91.67±0.36	82.30±0.53	87.90±0.87	89.58±0.26	81.48±2.41	86.83±0.87	45.83±0.85	42.36±1.77
	+ `TFB` (Ours)	✓	77.44±0.30	82.53±1.00	91.33±0.37	82.53±0.56	**88.53±0.57**	89.75±0.25	79.76±1.24	85.52±0.56	44.33±4.03	37.00±2.16
	MAP	-	76.90±0.97	81.08±2.48	91.61±0.44	82.59±0.28	85.73±0.19	90.09±0.28	79.98±0.87	86.58±0.79	43.40±4.98	38.54±3.40
	+ `TFB` (Ours)	✓	76.43±0.72	<u>82.80±1.42</u>	91.39±0.37	82.64±0.58	86.00±0.16	89.96±0.18	80.61±1.24	86.30±0.89	45.33±2.87	35.67±4.11
	BLoB	✗	76.45±0.37	82.32±1.15	91.14±0.54	82.01±0.56	87.57±0.21	89.65±0.15	79.75±0.43	87.13±0.00	42.71±3.71	44.79±6.64
	BLoB-Mean	✗	77.72±0.12	82.60±0.60	91.64±0.55	<u>83.92±0.48</u>	<u>88.00±0.80</u>	89.86±0.05	<u>82.06±1.15</u>	**88.54±0.31**	39.93±5.20	39.93±4.02
	+ `TFB` (Ours)	✓	77.81±0.36	**83.33±0.19**	<u>91.76±0.48</u>	83.81±0.39	87.80±0.16	<u>90.11±0.28</u>	**82.93±1.54**	<u>87.64±0.51</u>	39.67±7.32	37.33±6.65
ECE (↓)	MCD	✗	16.13±0.54	13.69±1.11	6.73±0.71	13.05±0.99	9.76±0.71	7.95±0.17	13.63±1.18	9.27±0.60	30.91±3.57	33.08±1.40
	ENS	✗	14.72±0.17	13.45±1.19	6.59±0.45	11.17±0.92	8.17±0.86	7.35±0.55	11.37±1.82	7.21±1.13	18.92±6.03	26.80±3.23
	LAP	BP	**4.18±0.11**	9.26±3.08	5.27±0.51	**3.50±0.78**	8.93±0.34	**1.93±0.22**	7.83±1.49	7.80±1.99	**14.49±0.57**	<u>13.17±2.14</u>
	MLE	-	17.02±0.46	16.35±0.68	7.00±0.53	13.83±0.65	9.77±0.81	8.69±0.21	14.45±2.19	10.78±0.50	32.46±2.60	38.41±4.44
	+ `TFB` (Ours)	✓	12.98±0.37	11.63±0.68	5.14±0.14	10.01±0.70	7.20±0.47	7.39±0.26	<u>6.54±0.53</u>	5.69±1.64	<u>14.63±1.46</u>	19.68±3.27
	MAP	-	18.71±0.74	15.77±1.60	6.62±0.64	14.26±0.92	12.19±0.55	8.40±0.25	16.46±0.44	11.36±0.58	34.79±3.76	38.50±2.18
	+ `TFB` (Ours)	✓	14.95±0.65	11.27±2.53	5.76±0.63	10.97±1.19	9.70±0.69	6.86±0.31	13.25±0.95	9.22±0.91	27.21±2.62	35.91±4.12
	BLoB	✗	9.93±0.22	**5.41±1.17**	<u>2.70±0.87</u>	4.28±0.64	<u>2.91±0.92</u>	<u>2.58±0.25</u>	**5.61±0.40**	**2.48±0.43**	16.67±0.87	**12.78±4.18**
	BLoB-Mean	✗	15.43±0.15	12.41±1.52	4.91±0.28	9.37±1.33	6.44±0.15	6.26±0.29	11.22±0.38	6.34±0.71	26.65±3.06	25.40±5.40
	+ `TFB` (Ours)	✓	<u>8.16±0.48</u>	<u>6.48±0.36</u>	**2.44±0.50**	<u>3.83±0.43</u>	**2.67±0.18**	3.10±0.59	6.69±1.63	<u>3.61±0.87</u>	18.45±6.75	20.53±6.27
NLL (↓)	MCD	✗	0.83±0.01	0.99±0.10	0.45±0.06	0.64±0.03	0.62±0.08	0.49±0.01	1.03±0.02	0.61±0.03	1.91±0.18	2.02±0.15
	ENS	✗	0.75±0.02	0.80±0.11	0.38±0.03	0.55±0.02	0.45±0.05	0.42±0.05	0.72±0.07	<u>0.44±0.03</u>	1.40±0.18	1.50±0.13
	LAP	BP	<u>0.56±0.00</u>	1.18±0.02	1.04±0.01	0.51±0.00	0.94±0.00	0.43±0.00	1.17±0.01	1.11±0.00	**1.27±0.01**	**1.28±0.00**
	MLE	-	0.88±0.04	1.20±0.11	0.46±0.04	0.68±0.01	0.61±0.06	0.52±0.01	1.07±0.06	0.72±0.06	1.91±0.16	2.25±0.21
	+ `TFB` (Ours)	✓	0.68±0.03	0.85±0.02	0.33±0.03	0.53±0.01	0.46±0.04	0.42±0.00	0.66±0.02	<u>0.44±0.01</u>	1.39±0.11	1.49±0.05
	MAP	-	0.99±0.07	1.12±0.23	0.46±0.03	0.74±0.07	0.79±0.02	0.52±0.01	1.19±0.04	0.83±0.06	1.97±0.13	2.32±0.10
	+ `TFB` (Ours)	✓	0.77±0.05	0.80±0.15	0.38±0.03	0.57±0.05	0.61±0.03	0.40±0.01	0.96±0.08	0.66±0.06	1.69±0.16	2.12±0.08
	BLoB	✗	0.58±0.00	**0.51±0.03**	**0.23±0.01**	<u>0.43±0.01</u>	<u>0.34±0.01</u>	**0.26±0.01**	<u>0.56±0.02</u>	**0.35±0.02**	<u>1.34±0.04</u>	<u>1.35±0.10</u>
	BLoB-Mean	✗	0.74±0.02	0.73±0.04	<u>0.29±0.03</u>	0.47±0.03	0.37±0.02	0.32±0.02	0.67±0.07	0.39±0.03	1.53±0.13	1.54±0.15
	+ `TFB` (Ours)	✓	**0.55±0.01**	<u>0.53±0.04</u>	**0.23±0.02**	**0.40±0.01**	**0.33±0.02**	<u>0.27±0.01</u>	**0.52±0.05**	**0.35±0.02**	1.36±0.13	1.46±0.11

`TFB` Implementation. `TFB` can be directly applied to trained LoRA adapters without additional training. As indicated by the **"TF?"** column in Table 1, `TFB` is **T**raining-**F**ree and requires only LLM inference (✓), while the other methods need full retraining (✗) or gradient estimation with Backpropagation (**BP**). We evaluate `TFB` on three off-the-shelf LoRA checkpoints: **MLE**, **MAP**, and the mean component of **BLoB** (obtained by discarding BLoB's standard deviation matrix $\mathbf{\Omega}$). More details are included in Appendix E.4.

5.2 `TFB` Improves Accuracy and Uncertainty Estimation

Table 1 shows results on comprehensive metrics (ACC, ECE, and NLL) for various methods applied to LoRA on Llama3.1-8B pre-trained weights. More empirical results on Llama2-7B can be found in Appendix F.7.

In-Distribution Results. The addition of `TFB` maintains competitive accuracy while substantially improving model calibration across in-distribution datasets. For ECE, `TFB` yields notable improvements when applied to different base methods: MLE+`TFB` reduces ECE to 5.14% on ARC-E (from 7.00%); similarly MAP+`TFB` and BLoB-Mean+`TFB` reduce ECE to 9.70% on OBQA (from 12.19%) and 3.83% on WG-M (from 9.37%), respectively. For NLL, `TFB` consistently produces better-calibrated predictions, with BLoB-Mean+`TFB` achieving strong performance across datasets: 0.23 on ARC-E (from 0.29), 0.33 on OBQA (from 0.37), and 0.27 on BoolQ (from 0.32). These improvements in both ECE and NLL demonstrate `TFB`'s effectiveness in enhancing model calibration while preserving accuracy on in-distribution tasks.

Out-of-Distribution Results. For out-of-distribution datasets, which represent a more challenging evaluation scenario, `TFB` continues to show benefits, though the performance gaps are generally smaller. In both Small Shift and Large Shift scenarios, `TFB`-enhanced methods maintain relatively

Table 2: **Performance of `TFB` with different variational distribution families applied to the mean of BLoB on Llama3.1-8B pre-trained weights. FR:** Full-rank isotropic Gaussian noises are applied to ΔW; **C-STD:** Standard deviation matrix $\mathbf{\Omega} = [\Omega_{ij} = \sigma_q]$ is constant. The evaluation protocol strictly follows Table 1. **"Rk."**: Average ranking of each method when compared to all other approaches on in-distribution datasets.

Metric	Method	In-Distribution Datasets							Out-of-Distribution Datasets (OBQA→X)			
									Small Shift		*Large Shift*	
		WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	Rk. (↓)	ARC-C	ARC-E	Chem	Phy
ACC (↑)	BLoB-Mean	77.72±0.12	82.60±0.60	91.64±0.55	**83.92±0.48**	88.00±0.80	89.86±0.05	2.50	82.06±1.15	**88.54±0.31**	39.93±5.20	39.93±4.02
	+ TFB (FR)	75.57±0.25	83.20±0.65	91.58±0.67	82.19±1.09	**88.73±0.41**	89.46±0.17	2.83	81.33±0.82	88.06±0.75	42.00±2.16	**41.33±5.44**
	+ TFB (C-STD)	76.35±0.08	83.20±0.33	91.33±0.70	81.79±0.51	88.20±0.57	89.65±0.08	3.00	81.73±0.68	88.18±0.65	**43.00±1.41**	39.33±3.86
	+ TFB (Final)	**77.81±0.36**	**83.33±0.19**	**91.76±0.48**	83.81±0.39	87.80±0.16	**90.11±0.28**	**1.67**	**82.93±1.54**	87.64±0.51	39.67±7.32	37.33±6.65
ECE (↓)	BLoB-Mean	15.43±0.15	12.41±1.52	4.91±0.28	9.37±1.33	6.44±0.15	6.26±0.29	4.00	11.22±0.38	6.34±0.71	26.65±3.06	25.40±5.40
	+ TFB (FR)	10.42±0.29	7.45±0.88	**2.01±1.03**	4.36±0.68	3.70±1.04	3.62±0.10	2.67	7.19±1.40	3.29±1.03	**17.78±1.01**	**19.14±4.01**
	+ TFB (C-STD)	9.23±0.20	**5.98±0.32**	2.94±0.67	3.86±0.45	3.17±0.21	**2.82±0.62**	1.83	6.89±0.89	**2.76±0.88**	18.27±2.52	19.45±3.46
	+ TFB (Final)	**8.16±0.48**	6.48±0.36	2.44±0.50	**3.83±0.43**	**2.67±0.18**	3.10±0.59	**1.50**	**6.69±1.63**	3.61±0.87	18.45±6.75	20.53±6.27
NLL (↓)	BLoB-Mean	0.74±0.02	0.73±0.04	0.29±0.03	0.47±0.03	0.37±0.02	0.32±0.02	3.67	0.67±0.07	0.39±0.03	1.53±0.13	1.54±0.15
	+ TFB (FR)	0.60±0.01	0.53±0.03	0.23±0.02	0.43±0.01	**0.33±0.02**	0.27±0.01	2.00	0.57±0.04	0.34±0.02	**1.34±0.07**	1.42±0.09
	+ TFB (C-STD)	0.57±0.01	**0.51±0.02**	**0.22±0.01**	0.43±0.01	**0.33±0.01**	**0.26±0.01**	**1.33**	0.56±0.04	**0.33±0.02**	**1.34±0.08**	**1.41±0.09**
	+ TFB (Final)	**0.55±0.01**	0.53±0.04	0.23±0.02	**0.40±0.01**	**0.33±0.02**	0.27±0.01	1.50	**0.52±0.05**	0.35±0.02	1.36±0.13	1.46±0.11

strong performance, particularly in the Small Shift cases (ARC-C and ARC-E). However, there's a noticeable performance drop in the Large Shift scenarios (Chem and Phy), which is expected given the significant domain difference. Even in these challenging cases, `TFB`-enhanced methods tend to maintain better calibration (lower ECE scores) compared to their base counterparts, suggesting improved reliability in out-of-distribution settings.

5.3 `TFB` Beyond the Low-Rank Isotropic Gaussians

In this section, we consider two simple `TFB` variants with other families of Gaussians for modeling the variational distributions of $\boldsymbol{W}$: (i) Full-Rank Isotropic Gaussian (**FR**, $\mathbf{\Sigma}_q = \sigma_q^2 \boldsymbol{I}$), and (ii) Constant Low-Rank Standard Deviation (**C-STD**, $\mathbf{\Omega} = [\Omega_{ij} = \sigma_q]$). Similar to our final `TFB`, both distributions are controlled by a single σ_q parameter and fit the maximal variance search in Eqn. 8. For fair comparison, we adopt the same optimal σ_q^* search protocol of `TFB` as described in Sec. 5.1. Table 2 shows the performances of `TFB` and its variants applied to the mean of BLoB (more in Table 7 of Appendix F.3).

These results show that our final `TFB` outperforms both variants **FR** and **C-STD** across multiple metrics on in-distribution datasets, with notable improvements in accuracy (e.g., 83.13% on OBQA) and calibration (ECE reduced by up to 16.74%). While these two simple variants show better NLL scores, these improvements come at the cost of significantly degraded overall performance, making them impractical for real-world applications. Although our final `TFB` maintains strong performance on datasets with smaller distributional shifts, its advantages diminish on datasets with larger shifts in the domains of Physics and Chemistry.

Advantages of Final `TFB`'s Variational Low-Rank Isotropic Gaussians. Compared to *`TFB` (FR)* and *`TFB` (C-STD)*, *`TFB` (Final)* offers additional advantages. It is computationally more efficient than **FR** with noise complexity of $O(rn)$ versus $O(mn)$. Furthermore, unlike **C-STD** whose variational distributions vary with different but equivalent LoRA matrix pairs (see Appendix F.3 for details), *`TFB` (Final)* produces consistent Bayesianization for all equivalent LoRAs satisfying $\boldsymbol{BA} = \Delta \boldsymbol{W}$.

6 Conclusion

In this paper, we introduce Training-Free Bayesianization (`TFB`), a novel framework that transforms trained LoRA adapters into Bayesian ones without additional training. By systematically searching for the maximally acceptable variance in the weight posterior within a family of low-rank isotropic Gaussian distributions, `TFB` provides a practical solution to uncertainty estimation in LLMs. Our theoretical analysis shows that `TFB`'s variance maximization process is equivalent to generalized variational inference under mild conditions. Our empirical results verify its superior performance across various settings and model configurations. Our framework's simplicity and effectiveness, requiring only an anchor dataset for search, makes it widely applicable across different domains. As LLMs continue to evolve, `TFB` represents a significant step toward more reliable and uncertainty-aware AI systems, paving the way for future research in adaptive and trustworthy machine learning.

REFERENCES

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

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

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Bishop, C. M. Pattern recognition and machine learning. *Springer google schola*, 2:1122–1128, 2006.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Burstein, J., Doran, C., and Solorio, T. (eds.), *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)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Fan, X., Zhang, S., Chen, B., and Zhou, M. Bayesian attention modules. *Advances in Neural Information Processing Systems*, 33:16362–16376, 2020.

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

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Gupta, N., Narasimhan, H., Jitkrittum, W., Rawat, A. S., Menon, A. K., and Kumar, S. Language model cascades: Token-level uncertainty and beyond. *arXiv preprint arXiv:2404.10136*, 2024.

Gustafsson, F. K., Danelljan, M., and Schon, T. B. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 318–319, 2020.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Hernández-Lobato, J. M. and Adams, R. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pp. 1861–1869. PMLR, 2015.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hüllermeier, E. and Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., Ganguli, D., Hernandez, D., Jacobson, J., Kernion, J., Kravec, S., Lovitt, L., Ndousse, K., Olsson, C., Ringer, S., Amodei, D., Brown, T., Clark, J., Joseph, N., Mann, B., McCandlish, S., Olah, C., and Kaplan, J. Language models (mostly) know what they know, 2022.

Kapoor, S., Gruver, N., Roberts, M., Collins, K., Pal, A., Bhatt, U., Weller, A., Dooley, S., Goldblum, M., and Wilson, A. G. Large language models must be taught to know what they don't know. *arXiv preprint arXiv:2406.08391*, 2024.

Karush, W. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*, 1939.

Khan, M., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. Fast and scalable bayesian deep learning by weight-perturbation in adam. In *International conference on machine learning*, pp. 2611–2620. PMLR, 2018.

Kjeldsen, T. H. A contextualized historical analysis of the kuhn–tucker theorem in nonlinear programming: the impact of world war ii. *Historia mathematica*, 27(4):331–361, 2000.

Klema, V. and Laub, A. The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, 25(2):164–176, 1980.

Kopiczko, D. J., Blankevoort, T., and Asano, Y. M. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

Krishnan, R., Esposito, P., and Subedar, M. Bayesian-torch: Bayesian neural network layers for uncertainty estimation. https://github.com/IntelLabs/bayesian-torch, January 2022. URL https://doi.org/10.5281/zenodo.5908307.

Kuhn, H. W. and Tucker, A. W. Nonlinear programming. In *Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492. Berkeley: University of California Press, 1951.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Lin, S., Hilton, J., and Evans, O. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022.

Meng, F., Wang, Z., and Zhang, M. Pissa: Principal singular values and singular vectors adaptation of large language models. *arXiv preprint arXiv:2404.02948*, 2024.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260.

Milne, T. Piecewise strong convexity of neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/b33128cb0089003ddfb5199e1b679652-Paper.pdf.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Neal, R. M. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Nikitin, A., Kossen, J., Gal, Y., and Marttinen, P. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *arXiv preprint arXiv:2405.20003*, 2024.

OpenAI. Introducing chatgpt. [online]. available: https://openai.com/blog/chatgpt. 2022.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, aug 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., Finn, C., and Manning, C. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330.

Tierney, L. and Kadane, J. B. Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, 81(393):82–86, 1986.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Wang, H. and Yeung, D.-Y. Towards bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3395–3408, 2016.

Wang, H., Tan, S., Hong, Z., Zhang, D., and Wang, H. Variational language concepts for interpreting foundation language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8645–8671, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.505. URL https://aclanthology.org/2024.findings-emnlp.505/.

Wang, H., Tan, S., and Wang, H. Probabilistic conceptual explainers: trustworthy conceptual explanations for vision foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024b.

Wang, X., Aitchison, L., and Rudolph, M. Lora ensembles for large language model fine-tuning, 2023.

Wang, Y., Shi, H., Han, L., Metaxas, D., and Wang, H. Blob: Bayesian low-rank adaptation by backpropagation for large language models. *arXiv preprint arXiv:2406.11675*, 2024c.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

Yadkori, Y. A., Kuzborskij, I., György, A., and Szepesvári, C. To believe or not to believe your llm. *arXiv preprint arXiv:2406.02543*, 2024.

Yang, A. X., Robeyns, M., Wang, X., and Aitchison, L. Bayesian low-rank adaptation for large language models. *arXiv preprint arXiv:2308.13111*, 2023.

Zhang, Q., Chen, M., Bukharin, A., Karampatziakis, N., He, P., Cheng, Y., Chen, W., and Zhao, T. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

Zhang, S., Fan, X., Chen, B., and Zhou, M. Bayesian attention belief networks. In *International Conference on Machine Learning*, pp. 12413–12426. PMLR, 2021.

Zhao, Z., Wallace, E., Feng, S., Klein, D., and Singh, S. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. PMLR, 2021.

APPENDIX

In Appendix A, we presents the full algorithmic description of our proposed TFB. In Appendix B, we discuss the limitations of TFB. Next, in Appendix C, we present a more detailed introduction to recent advances of Bayesian Low-Rank Adaptation. In Appendix D, we provide detailed proofs for all theorems presented in the main paper. In Appendix E, we describe our experimental methodology. Finally, in Appendix F, we present additional empirical results, including the full results of experiments with different searching strategies for optimal σ_q^*, variational distribution family, LLM backbones, and LoRA-like PEFTs.

A ALGORITHM

Algorithm 1 **T**raining-**F**ree **B**ayesianization (**TFB**)

input $\mathcal{D}$: Anchor Dataset;
$\{\boldsymbol{B}, \boldsymbol{A}\}$: Low-Rank Component;
l: Model Evaluation Metric;
ϵ: Performance Change Tolerance;
$[\sigma_{q_{\min}}, \sigma_{q_{\max}}]$: search range of σ_q.

Evaluate the original performance: $p_0 \leftarrow l(\mathcal{D}|\boldsymbol{B}, \boldsymbol{A})$.
Singular Value Decomposition on $\boldsymbol{B}$:
  $\boldsymbol{U}, \mathrm{diag}(\boldsymbol{d}), \boldsymbol{V} \leftarrow \mathrm{SVD}(\boldsymbol{B})$. ▷ Eqn. 4.
Get an equivalent pair of the low-rank component:
  $\boldsymbol{B}' \leftarrow \boldsymbol{U}\,\mathrm{diag}(\boldsymbol{d})$; $\boldsymbol{A}' \leftarrow \boldsymbol{V}^\top \boldsymbol{A}$. ▷ Eqn. 5.
while σ_q not converged **do**
  $\sigma_q \leftarrow (\sigma_{q_{\max}} + \sigma_{q_{\min}})/2$.
  Calculate the standard deviation matrix $\boldsymbol{\Omega}$ for $\boldsymbol{A}'$:
    $\Omega_{ij} = \sigma_q / d_i$. ▷ Eqn. 7.
  Evaluate the performance:
    $p \leftarrow l(\mathcal{D}|\boldsymbol{B}', \boldsymbol{A}', \boldsymbol{\Omega})$.
  if $|p - p_0| < \epsilon$ **then**
    $\sigma_{q_{\min}} \leftarrow \sigma_q$.
  else
    $\sigma_{q_{\max}} \leftarrow \sigma_q$.
  end if
end while
output $\{\boldsymbol{B}', \boldsymbol{A}', \boldsymbol{\Omega}\}$: Bayesianized Low-Rank Adapter.

B LIMITATIONS

TFB is subject to several limitations. First, our approach relies on the availability of an anchor dataset for determining search range and stopping criteria. Although this dataset doesn't require supervision or prior use in LoRA training, its quality and representativeness could impact the effectiveness of uncertainty estimation. Second, by constraining the family of full-weight posteriors to low-rank isotropic Gaussian distributions, TFB may not capture more complex uncertainty patterns that could be present in the data. While this constraint enables our training-free approach, it represents a trade-off between computational efficiency and model expressiveness. Finally, while we have demonstrated the effectiveness of TFB in various settings, its performance in more complex generation tasks requires further investigation. Future work could explore extending the framework to handle more sophisticated language generation scenarios and broader applications.

C RELATED WORK

Bayesian Low-Rank Adaptation. The Bayesian framework provides a powerful approach for capturing and estimating uncertainty by defining prior distributions and approximating posterior distributions over the parameter space (Neal, 2012; Hernández-Lobato & Adams, 2015; Gal & Ghahramani, 2016; Wang & Yeung, 2016; Gustafsson et al., 2020). However, modeling parameter distributions across the entire parameter space during fine-tuning introduces significant computational overhead (Fan et al., 2020; Zhang et al., 2021). To address this challenge, recent research has explored combining Bayesian methods with Parameter-Efficient Fine-Tuning (PEFT) techniques to improve the efficiency of uncertainty estimation. Several notable approaches have emerged in this direction. Wang et al. (2023) and Balabanov & Linander (2024) demonstrate improved performance by training multiple LoRA modules and ensemble their predictions during inference. Taking a different approach, Yang et al. (2023) applies a Kronecker-factorized Laplace approximation to fine-tuned LoRA parameters. More recently, BLoB (Wang et al., 2024c) advances the field by simultaneously estimating both the mean and covariance of LLM parameters within a single fine-tuning stage, leading to substantial performance improvements. Our proposed training-free Bayesianization represents a significant departure from these existing methods. Unlike approaches that require re-training (Gal & Ghahramani, 2016; Wang et al., 2023; Balabanov & Linander, 2024; Wang et al., 2024c) or rely on continued training and gradient estimation (Yang et al., 2023), our method achieves uncertainty estimation without any additional training steps, substantially improving the simplicity and efficiency for Bayesian learning of LLMs.

D Proof of Theorems

Lemma D.1. *With the pre-trained weight matrix $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$ and the low-rank weight update matrix $\boldsymbol{B} \in \mathbb{R}^{m\times r}$, suppose that the variational distribution of the other low-rank update matrix $\boldsymbol{A} \in \mathbb{R}^{r\times n}$ is Gaussian with $q(\boldsymbol{A}|\boldsymbol{\theta} = \{\boldsymbol{M}, \boldsymbol{\Omega}\}) = \prod_{ij} \mathcal{N}(A_{ij}|M_{ij}, \Omega_{ij}^2)$, where $\boldsymbol{M} = [M_{ij}] \in \mathbb{R}^{r\times n}$ and $\boldsymbol{\Omega} = [\Omega_{ij}] \in \mathbb{R}^{r\times n}$ are its mean and standard deviation, respectively. The equivalent variational distribution defined on the full weight matrix $\boldsymbol{W}$ is given by*

$$\begin{aligned} q(\operatorname{vec}(\boldsymbol{W})|\boldsymbol{B}, \boldsymbol{\theta}) &= \mathcal{N}(\operatorname{vec}(\boldsymbol{W})|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q), \\ \textit{where} \quad \boldsymbol{\mu}_q &= \operatorname{vec}(\boldsymbol{W}_0 + \boldsymbol{B}\boldsymbol{M}), \\ \boldsymbol{\Sigma}_q &= [\boldsymbol{I}_n \otimes \boldsymbol{B}][\operatorname{diag}(\operatorname{vec}(\boldsymbol{\Omega})^2)][\boldsymbol{I}_n \otimes \boldsymbol{B}^\top]. \end{aligned} \tag{14}$$

Theorem 4.1 (Equivalent Variational Distribution of the Full Weight $\boldsymbol{W}$ in `TFB`). *With the pre-trained weight matrix $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$, the low-rank weight update matrix $\{\boldsymbol{B}' \in \mathbb{R}^{m\times r}, \boldsymbol{A}' \in \mathbb{R}^{r\times n}\}$ transformed from the given matrices $\{\boldsymbol{B}, \boldsymbol{A}\}$ following Eqn. 4 and 5, suppose that the variational distribution of $\boldsymbol{A}'$ is Gaussian $q(\boldsymbol{A}'|\boldsymbol{\theta}) = \prod_{ij} \mathcal{N}(A_{ij}|M_{ij}, \Omega_{ij}^2)$, where $\boldsymbol{M} = [M_{ij} = A'_{ij}] \in \mathbb{R}^{r\times n}$ is its mean and $\boldsymbol{\Omega} = [\Omega_{ij}] \in \mathbb{R}^{r\times n}$ is the standard deviation calculated as in Eqn. 7. The equivalent variational distribution $q(\operatorname{vec}(\boldsymbol{W})|\sigma_q)$ defined on the full weight matrix $\boldsymbol{W}$ is*

$$\begin{aligned} q(\operatorname{vec}(\boldsymbol{W})|\sigma_q) &= \mathcal{N}(\operatorname{vec}(\boldsymbol{W})|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q), \\ \textit{where} \quad \boldsymbol{\mu}_q &= \operatorname{vec}(\boldsymbol{W}_0 + \boldsymbol{B}'\boldsymbol{M}), \\ \boldsymbol{\Sigma}_q &= \sigma_q^2 \boldsymbol{I}_n \otimes \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{0}_{m-r} \end{bmatrix}. \end{aligned} \tag{15}$$

Proof. We have the following lemma from BLoB that calculates the covariance matrix of a given low-rank Bayesianization scheme $\{\boldsymbol{B}, \boldsymbol{A}, \boldsymbol{\Omega}\}$ (Wang et al., 2024c).

Based on the assumption outlined in Eqn. 4, 5, and 7, we have the following properties about $\boldsymbol{B}'$, $\boldsymbol{M}$, and $\boldsymbol{\Omega}$ of `TFB`:

$$\boldsymbol{B}' = \boldsymbol{U} \operatorname{diag}(\boldsymbol{d}), \tag{16}$$

$$\boldsymbol{\Omega} = [1/\boldsymbol{d}, \cdots, 1/\boldsymbol{d}], \tag{17}$$

$$\text{where } \boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_r, \boldsymbol{U}\boldsymbol{U}^\top = \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{0}_{m-r} \end{bmatrix}. \tag{18}$$

It now can be easily shown that the covariance matrix of `TFB` is:

$$\begin{aligned} \boldsymbol{\Sigma}_q &= [\boldsymbol{I}_n \otimes \boldsymbol{B}'][\operatorname{diag}(\operatorname{vec}(\boldsymbol{\Omega})^2)][\boldsymbol{I}_n \otimes \boldsymbol{B}'^\top] & (19) \\ &= [\boldsymbol{I}_n \otimes \boldsymbol{B}'][\boldsymbol{I}_n \otimes \operatorname{diag}(1/\boldsymbol{d})^2][\boldsymbol{I}_n \otimes \boldsymbol{B}'^\top] & (20) \\ &= \boldsymbol{I}_n \otimes [\boldsymbol{B}' \operatorname{diag}(\sigma_q/\boldsymbol{d})^2 \boldsymbol{B}'^\top] & (21) \\ &= \boldsymbol{I}_n \otimes [\boldsymbol{U} \operatorname{diag}(\boldsymbol{d}) \operatorname{diag}(\sigma_q/\boldsymbol{d})^2 \operatorname{diag}(\boldsymbol{d})^\top \boldsymbol{U}^\top] & (22) \\ &= \sigma_q^2 \boldsymbol{I}_n \otimes \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{0}_{m-r} \end{bmatrix}, & (23) \end{aligned}$$

which proves that $q(\operatorname{vec}(\boldsymbol{W}))$ is a low-rank isotropic Gaussian distribution. ◻

Proposition D.1.1. *The function* $\mathrm{proj}(\cdot)$ *defined in Eqn. 3 projects the full-dimensional isotropic Gaussian to the low-rank subspace of LoRA. It can be formulated as*

$$\mathrm{proj}(\sigma_q^2 \boldsymbol{I}_{mn}) = \boldsymbol{P}(\sigma_q^2 \boldsymbol{I}_{mn}), \tag{24}$$

$$\textit{where} \quad \boldsymbol{P} = \boldsymbol{I}_n \otimes \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{0}_{m-r} \end{bmatrix}. \tag{25}$$

Proof. By Theorem 4.1, we have

$$\boldsymbol{P}\boldsymbol{I}_{mn} = \boldsymbol{I}_n \otimes \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{0}_{m-r} \end{bmatrix}. \tag{26}$$

Hence it is trivial to have $\boldsymbol{P} = \boldsymbol{I}_n \otimes \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{0}_{m-r} \end{bmatrix}$. □

Theorem 4.2 (`TFB` as Generalized Variational Inference). *Suppose the evaluation metric* $l_{\mathcal{D}}(\sigma_q)$ *defined following Assumption 4.1 is locally convex within the range of* $0 < \sigma_q < \epsilon_0$. *Suppose the approximate distribution of* $\boldsymbol{W}$ *given* σ_q *is defined following Theorem 4.2. Suppose we have the prior distribution* $P(\mathrm{vec}(\boldsymbol{W})) = \mathcal{N}(\mathrm{vec}(\boldsymbol{W})|\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$, *where* $\boldsymbol{\mu}_p = \boldsymbol{\mu}_q = \mathrm{vec}(\boldsymbol{W}_0 + \boldsymbol{B}'\boldsymbol{M})$, *and* $\boldsymbol{\Sigma}_p = \sigma_p^2 \boldsymbol{I}$ *with* $\sigma_p > \epsilon_0$. *Then* $\exists \widetilde{\epsilon} > 0$, *s.t. the two following optimization problems*

(i) Generalized Variational Inference (Blundell et al., 2015; Higgins et al., 2017; Khan et al., 2018)

$$\min_{\sigma_q} \quad l_{\mathcal{D}}(\sigma_q) + \lambda\, \mathrm{KL}[q(\boldsymbol{W}|\sigma_q) \parallel P(\boldsymbol{W})], \tag{27}$$

and (ii) Training-Free Bayesianization (`TFB`*)*

$$\begin{aligned} \max \quad & \sigma_q \\ \textit{s.t.} \quad & l_{\mathcal{D}}(\sigma_q) \leq \widetilde{\epsilon}, \end{aligned} \tag{28}$$

are equivalent, where λ *is the coefficient of the KL-divergence regularization.*

Proof. First we prove the KL divergence term is convex w.r.t. σ_q. For two Gaussian distributions q and p whose covariance matrices $\boldsymbol{\Sigma}_q \in \mathbb{R}^{d\times d}$ and $\boldsymbol{\Sigma}_p \in \mathbb{R}^{d\times d}$ are both full-rank, with their means as $\boldsymbol{\mu}_q \in \mathbb{R}^d$ and $\boldsymbol{\mu}_p \in \mathbb{R}^d$, we have their KL-divergence as

$$\mathrm{KL}[q\|p] = \frac{1}{2}\left[\log\frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}_q|} - d + \mathrm{tr}(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}_q) + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^\top \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)\right]. \tag{29}$$

For `TFB`, to avoid unbounded KL divergence, we project the original assumed Gaussian prior P into the same low-rank sub-space of the posterior q. We summarize the prior and variational distribution of the posterior as follows:

$$\begin{aligned} q(\mathrm{vec}(\boldsymbol{W})|\sigma_q) &= \mathcal{N}\left(\mathrm{vec}(\boldsymbol{W})|\boldsymbol{\mu}_q = \mathrm{vec}(\boldsymbol{W}_0 + \boldsymbol{B}'\boldsymbol{M}), \boldsymbol{\Sigma}_q = \sigma_q^2 \boldsymbol{I}_n \otimes \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{0}_{m-r} \end{bmatrix}\right), \\ P(\mathrm{vec}(\boldsymbol{W})|\sigma_p) &= \mathcal{N}\left(\mathrm{vec}(\boldsymbol{W})|\boldsymbol{\mu}_p = \mathrm{vec}(\boldsymbol{W}_0 + \boldsymbol{B}'\boldsymbol{M}), \boldsymbol{\Sigma}_p = \sigma_p^2 \boldsymbol{I}_n \otimes \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{0}_{m-r} \end{bmatrix}\right). \end{aligned} \tag{30}$$

Substituting Eqn. 30 back into Eqn. 29, we have

$$\mathrm{KL}[q(\mathrm{vec}(\boldsymbol{W})|\sigma_q)\|P(\mathrm{vec}(\boldsymbol{W}|\sigma_p))] = \frac{nr}{2}\left[\log(\sigma_p^2) - 1 + \left\{-\log(\sigma_q^2) + \frac{\sigma_q^2}{\sigma_p^2}\right\}\right], \tag{31}$$

which is convex w.r.t. σ_q and the global minimum of KL is achieved when $\sigma_q = \sigma_p$.

With $\sigma_q \leq \epsilon_0$, the convexity of two terms (KL and $l_{\mathcal{D}}$) holds. Hence we show by the Karush–Kuhn–Tucker theorem (Kjeldsen, 2000; Karush, 1939; Kuhn & Tucker, 1951) that, for any given λ there exists $\widetilde{\epsilon}$ such that the following two optimization problems are equivalent:

1. Minimization of generalized variational inference in the **Lagrange-form optimization**

$$\min_{\sigma_q} \quad \mathrm{KL}[q(\mathrm{vec}(\boldsymbol{W})|\sigma_q) \parallel P(\mathrm{vec}(\boldsymbol{W})|\sigma_p)] + \frac{1}{\lambda} l_{\mathcal{D}}(\sigma_q); \tag{32}$$

2. The **constrained-form optimization** corresponding to Eqn. 32

$$\begin{aligned} \min \quad & \mathrm{KL}[q(\mathrm{vec}(\boldsymbol{W})|\sigma_q) \parallel P(\mathrm{vec}(\boldsymbol{W})|\sigma_p)] \\ \textit{s.t.} \quad & l_{\mathcal{D}}(\sigma_q) \leq \widetilde{\epsilon}. \end{aligned} \tag{33}$$

Since the KL term is monotonically decreasing when $\sigma_q \in [0, \sigma_p)$, and due to the fact that $\sigma_p > \epsilon_0$, the optimization in Eqn. 33 is equivalent to our final Training-Free Bayesianization (TFB):

$$\begin{aligned} \max \quad & \sigma_q \\ \textit{s.t.} \quad & l_{\mathcal{D}}(\sigma_q) \leq \widetilde{\epsilon}. \end{aligned} \tag{34}$$

□

E Implementation Details

E.1 Datasets

We provide details of the datasets used in this work, as shown in Table 3. The combined dataset consisting of the six commonsense reasoning tasks contains the label set of "[A, B, C, D, E, True, False]".

Table 3: Dataset Statistics. The size of the Anchor Set $\mathcal{D}$ is used in Table 1, 2 and 10.

	WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	Combined
Size of Label Space	2	5	5	2	4	2	7
Size of Training Set	640	1,119	2,251	2,258	4,957	9,427	20,652
Size of Anchor Set $\mathcal{D}$	500 (78%)	500 (45%)	500 (22%)	500 (22%)	500 (10%)	500 (5%)	500 (2%)
Size of Test Set	1,267	299	570	1,267	500	3,270	7,173

E.2 Evaluation Metrics

Negative Log-Likelihood (**NLL**) and Expected Calibration Error (**ECE** (Naeini et al., 2015)) are key metrics for uncertainty estimation. For a model $P_{\boldsymbol{\theta}}$ and test dataset $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, **NLL** penalizes models that assign low probabilities to correct labels, and is defined as:

$$\mathrm{NLL} = \tfrac{1}{N} \sum_{n=1}^{N} - \log P_{\boldsymbol{\theta}}(y_n). \tag{35}$$

ECE measures the alignment between model confidence and accuracy by binning predictions:

$$\mathrm{ECE} = \sum_{m=1}^{M} \tfrac{|B_m|}{n} \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right|, \tag{36}$$

where $\mathrm{acc}(B_m) = {}^1\!/_{|B_m|} \sum_{i \in B_m} \mathbb{1}(\widehat{y}_i = y_i)$ is the average accuracy and $\mathrm{conf}(B_m) = {}^1\!/_{|B_m|} \sum_{i \in B_m} P(\widehat{y}_i)$ is the average confidence in bin B_m. We use bin size $|B_m| = 15$ throughout this paper.

E.3 Searched σ_q of TFB

We report the searched σ_q^* using Algorithm 1 in Table 4, where the reported values are the mean values of three random seeds.

Table 4: Searched σ_q^* of TFB using Algorithm 1.

Base Model	WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ
MLE	0.004500	0.003917	0.004500	0.004354	0.003771	0.004063
MAP	0.004500	0.003479	0.003188	0.004208	0.003917	0.005083
BLoB-Mean	0.005813	0.005229	0.005229	0.006250	0.006250	0.005958

E.4 Bayesianization (Training)

Shared Configuration. We report the mean and standard deviation of all experimental results calculated over three random seeds. For all training processes in our experiments, we employ the AdamW optimizer. The learning rate follows a linear decay schedule with a warmup ratio of 0.06 and a maximum value of $2e-4$. The batch size is set to 4, and the maximum sentence length is limited to 300 tokens. The LoRA configuration includes LoRA $\alpha = 16$ and LoRA $r = 8$. PiSSA (Meng et al., 2024) follows the exact same configuration as the LoRA's. For VeRA (Kopiczko et al., 2023), due to its characteristic of shared weights across different layers which enables higher-rank setting with the same memory efficiency, we set its rank to $r = 256$ and learning rate to $5e-3$ for the MLE training on the combined dataset.

Baseline Configuration. The baseline configuration mainly follows BLoB (Wang et al., 2024c). MLE follows the standard LoRA implementation. For MAP, we implement it with a weight decay rate of $1e-5$. MCD consists of an ensemble of 10 LoRAs with a dropout rate of $p = 0.1$. For ENS, we fine-tune 3 LoRAs independently and combine them by averaging their logits during evaluation. We implement LAP and apply it to the MAP checkpoints. For BBB and BLoB, we use the default settings from Bayesian-Torch library (Krishnan et al., 2022), applying Bayesianization only to the $\boldsymbol{A}$ matrix. During training, the number of samples is set to $K = 1$ for both BBB and BLoB. At test time, we use $N = 10$ samples, matching the configuration of `TFB`.

`TFB` Configuration. We randomly sample unlabeled training data points to construct the anchor dataset $\mathcal{D} = \{\boldsymbol{x}_i, \widehat{y}_i\}_{i\in[M]}$ where $\widehat{y}_i$ is the pseudo-label generated by the given LoRA adapter before Bayesianization; the anchor dataset size $M = 500$ is fixed for all the datasets. We use NLL as the metric l and set the performance change tolerance ϵ to 0.3% of relative performance change for all the datasets. To determine the optimal σ_q^*, we perform a 5-step binary search with the initial range of $[0.001, 0.015]$ using Algorithm 1. Similar to the other baseline methods, the final results of `TFB` are reported as averages across three random seeds using σ_q^*.

F Additional Experimental Results

We present additional experimental results in this section. Due to space constraints (and large table size), we defer several detailed tables to the end of this section rather than presenting them alongside the corresponding analyses.

F.1 Computational Complexity Analysis

We compare the computational efficiency of `TFB` and BLoB during the process of Bayesianization in Table 5. We also report the computational cost of the standard LoRA fine-tuning as reference. All three methods are evaluated on the configurations detailed in Appendix E.4. For LoRA and BLoB, the evaluation of running time and maximum GPU memory is based on fine-tuning for 5 epochs. `TFB` uses a fixed number of 500 training examples to search for σ_q^* across all datasets, and performs binary search for at most 5 rounds (sequentially).

Table 5: **A comparison of running time and maximum GPU memory cost between `TFB` and BLoB during the process of Bayesianizatioin.** The subscripts in the table calculate the relative cost of a method compared to that of LoRA, a non-Bayesian baseline method. **RED** and **GREEN** represent **worse** and **better** efficiency, respectivley.

Method	Batch Size	Datasets											
		WG-S		ARC-C		ARC-E		WG-M		OBQA		BoolQ	
		Time (s)	Mem. (MB)	Time (s)	Mem. (MB)	Time (s)	Mem. (MB)	Time (s)	Mem. (MB)	Time (s)	Mem. (MB)	Time (s)	Mem. (MB)
LoRA	4	338	12,894	632	19,762	1,238	18,640	1,339	13,164	2,692	17,208	6,489	29,450
BLoB	4	371 (1.10x)	13,194 (1.02x)	685 (1.08x)	21,736 (1.10x)	1,360 (1.10x)	20,700 (1.11x)	1,476 (1.10x)	13,194 (1.00x)	3,257 (1.21x)	18,046 (1.05x)	7,251 (1.12x)	30,578 (1.04x)
`TFB` (Ours)	4	1,203 (3.56x)	10,372 (0.80x)	1,257 (1.99x)	11,966 (0.61x)	1,246 (1.01x)	11,202 (0.60x)	1,237 (0.92x)	10,344 (0.79x)	1,238 (0.46x)	10,376 (0.60x)	1,452 (0.22x)	16,340 (0.55x)
`TFB` (Ours)	8	628 (1.86x)	10,666 (0.83x)	731 (1.16x)	15,286 (0.77x)	702 (0.57x)	12,598 (0.68x)	634 (0.47x)	10,662 (0.81x)	642 (0.24x)	12,116 (0.70x)	1,015 (0.16x)	22,146 (0.75x)
`TFB` (Ours)	12	446 (1.31x)	12,064 (0.93x)	599 (0.94x)	18,204 (0.92x)	540 (0.43x)	14,310 (0.76x)	441 (0.32x)	11,370 (0.86x)	487 (0.18x)	13,410 (0.77x)	908 (0.13x)	25,220 (0.85x)

As shown in the table, for a small dataset, e.g., WG-S with ∼600 training examples, `TFB` can have a higher cost of time, especially when `TFB` is performed under the same batch size as the two baselines: almost 3x slower than BLoB. However, for a large dataset, e.g., BoolQ with ∼10,000 training examples, `TFB` Bayesianization process is almost 5x faster than BLoB using only half of the GPU memory. As `TFB` does not require gradient estimation, which significantly reduces the GPU memory, `TFB` can be further accelerated by increasing the batch size. When the batch size is increased from 4 to 12, `TFB` has a lower time and lower peak GPU memory usage compared to BLoB for almost all the datasets. `TFB` even has significantly better efficiency than standard LoRA fine-tuning, thanks to its training-free property.

Note that our efficiency analysis for `TFB` encompasses the complete parameter search process, while the reported metrics for other methods only include their final successful runs. This means the baseline measurements exclude significant hidden costs, particularly the computational resources required for hyperparameter tuning to determine optimal fine-tuning configurations. *Therefore this comparison inherently favors the baseline method BLoB.* Nevertheless, the fact that `TFB` still demonstrates superior efficiency in both time and memory consumption, even under these conditions that advantage the baselines, further underscores the benefits of its "training-free" approach. The results above also implies `TFB`'s flexibility, as it can adapt to limited-resource environments by trading time for reduced memory usage or vice versa, depending on operational requirements.

F.2 `TFB` BEYOND THE NLL METRIC

We report the additional results of `TFB` when using Accuracy (ACC) as the evaluation metrics l in Table 6. Comparing the two evaluation metrics (l=ACC vs l=NLL) in Table 6, we observe comparable performance across all datasets. In some cases, accuracy-based evaluation (l=ACC) even yields slightly better results. For instance, BLoB-Mean+`TFB` achieves lower ECE on several datasets when using l=ACC. However, we adopt NLL as the primary evaluation metric in Table 1 since it better aligns with our theoretical framework in Theorem 4.2.

F.3 `TFB` BEYOND THE LOW-RANK ISOTROPIC GAUSSIANS

In Sec. 5.3, we compare `TFB` with two alternative Gaussian distribution families that are controlled by a single parameter σ_q:

- Full-Rank Isotropic Gaussian (**FR**): given σ_q, the FR's variational distribution of the weight matrix $q(\text{vec}(\boldsymbol{W})) = \mathcal{N}(\text{vec}(\boldsymbol{W})|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ where $\boldsymbol{\mu}_q = \boldsymbol{W}_0 + \boldsymbol{BA}$ (same as `TFB`) and $\boldsymbol{\Sigma}_q = \sigma_q^2 \boldsymbol{I}_{mn}$ is full-rank.
- Constant Standard Deviation Matrix (**C-STD**): given σ_q, the C-STD's variational distribution of the weight matrix $q(\text{vec}(\boldsymbol{W})) = \mathcal{N}(\text{vec}(\boldsymbol{W})|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ where $\boldsymbol{\mu}_q = \boldsymbol{W}_0 + \boldsymbol{BA}$ (same as `TFB`) and $\boldsymbol{\Sigma}_q = \sigma_q^2 \boldsymbol{I}_n \otimes [\boldsymbol{BB}^\top]$.

C-STD's covariance matrix $\boldsymbol{\Sigma}_q$ is derived through Lemma D.1:

$$\begin{aligned}
\boldsymbol{\Sigma}_q &= [\boldsymbol{I}_n \otimes \boldsymbol{B}][\text{diag}(\text{vec}(\boldsymbol{\Omega})^2)][\boldsymbol{I}_n \otimes \boldsymbol{B}^\top] && (37)\\
&= [\boldsymbol{I}_n \otimes \boldsymbol{B}][\sigma_q^2 \boldsymbol{I}_{rn}][\boldsymbol{I}_n \otimes \boldsymbol{B}^\top] && (38)\\
&= \sigma_q^2 [\boldsymbol{I}_n \otimes \boldsymbol{B}][\boldsymbol{I}_n \otimes \boldsymbol{B}^\top] && (39)\\
&= \sigma_q^2 \boldsymbol{I}_n \otimes [\boldsymbol{BB}^\top]. && (40)
\end{aligned}$$

This depends on $\boldsymbol{B}$ and thus varies for equivalent LoRA parameterizations $\boldsymbol{B}, \boldsymbol{A}$ of the same $\Delta\boldsymbol{W}$.

We report the additional results comparing `TFB` with other approximate families Gaussians (FR and C-STD as discussed in Sec. 5.3) when using Accuracy as the evaluation metrics l in Table 7. When the evaluation metric is set to Accuracy, the advantage of `TFB` becomes more significant compared to the results shown in Table 2. `TFB` with low-rank isotropic Gaussian as the variational distribution demonstrates superior calibration performance compared to both **FR** and **C-STD** variants while maintaining competitive accuracy. For ECE, `TFB` achieves better results across most datasets, with notable improvements on in-distribution tasks: 8.78% on WG-S (vs. 12.06% for **FR** and 11.61% for **C-STD**) and 1.28% on BoolQ (vs. 3.26% for **FR** and 2.65% for **C-STD**). Similarly for NLL, `TFB` consistently outperforms or matches the baseline variants, particularly on WG-S (0.55 vs. 0.63 for **FR** and 0.61 for **C-STD**) while preserving comparable accuracy scores. These results suggest that `TFB`'s approach to variance modeling is more effective than both full-rank isotropic and constant standard deviation alternatives.

F.4 `TFB` BEYOND THE LLAMA3.1-8B BACKBONE

We report the detailed performance of `TFB` applied to various LLM backbones in Table 8. While the baseline MLE is typically trained for 2 epochs (shown with each backbone name), we

Table 6: **Performance of different methods applied to LoRA on Llama3.1-8B pre-trained weights,** where Accuracy (**ACC**) and Expected Calibration Error (**ECE**) are reported in percentages. **"TF?"** denotes whether a method is **T**raining-**F**ree. The evaluation is done across six common-sense reasoning tasks with a shared hyper-parameter setting after fine-tuning of 5 epochs . We sample $N = 10$ during inference in all sampling-based methods including **BLoB Wang et al. (2024c)** and **`TFB`**. Rows with shading indicate training-free Bayesianization methods that use a pre-trained LoRA as their mean. For `TFB`, we randomly sample a subset of the training data without labels as the anchor dataset $\mathcal{D}$. For accuracy-based evaluation (l=ACC), we set the performance drop tolerance to $\epsilon = 1\%$. For NLL loss (l=NLL), we use the same settings as in Table 1. "↑" and "↓" indicate that higher and lower values are preferred, respectively. **Boldface** and underlining denote the best and the second-best performance, respectively.

Metric	Method	TF?	In-Distribution Datasets						Out-of-Distribution Datasets (OBQA→X)			
									Small Shift		*Large Shift*	
			WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	ARC-C	ARC-E	Chem	Phy
ACC (↑)	MCD	✗	$\underline{78.03\pm0.61}$	81.64±1.79	91.37±0.38	83.18±0.84	87.20±1.02	89.93±0.16	81.42±1.38	87.27±0.84	47.92±2.25	**46.53±0.49**
	ENS	✗	**78.82±0.52**	82.55±0.42	**91.84±0.36**	**83.99±0.74**	87.37±0.67	**90.50±0.14**	79.62±0.57	86.56±0.60	**49.65±3.22**	44.44±1.96
	LAP	BP	76.05±0.92	79.95±0.42	90.73±0.08	82.83±0.85	87.90±0.20	89.36±0.52	81.08±1.20	87.21±1.20	$\underline{48.26\pm3.93}$	$\underline{46.18\pm1.30}$
	MLE	-	77.87±0.54	81.08±0.48	91.67±0.36	82.30±0.53	87.90±0.87	89.58±0.26	81.48±2.41	86.83±0.87	45.83±0.85	42.36±1.77
	+ `TFB` (l=ACC)	✓	76.48±0.20	81.47±0.50	90.85±0.23*	81.55±0.33	86.00±0.43	89.05±0.20	78.13±2.95	85.52±0.56	44.33±4.03	37.00±2.16
	+ `TFB` (l=NLL)	✓	77.44±0.30	82.53±1.00	91.33±0.37	82.53±0.56	**88.53±0.57**	89.75±0.25	79.76±1.24	85.52±0.56	44.33±4.03	37.00±2.16
	MAP	-	76.90±0.97	81.08±2.48	91.61±0.44	82.59±0.28	85.73±0.19	90.09±0.28	79.98±0.87	86.58±0.79	43.40±4.98	38.54±3.40
	+ `TFB` (l=ACC)	✓	74.80±0.66	82.27±1.00	91.15±0.75*	81.25±0.72	84.87±0.34	89.58±0.06	78.42±2.31	84.00±0.65	40.33±1.89	37.67±4.64
	+ `TFB` (l=NLL)	✓	76.43±0.72	82.80±1.42	91.39±0.37	82.64±0.58	86.00±0.16	89.96±0.18	80.61±1.24	86.30±0.89	45.33±2.87	35.67±4.11
	BLoB	✗	76.45±0.37	82.32±1.15	91.14±0.54	82.01±0.56	87.57±0.21	89.65±0.15	79.75±0.43	87.13±0.00	42.71±3.71	44.79±6.64
	BLoB-Mean	✗	77.72±0.12	82.60±0.60	91.64±0.55	$\underline{83.92\pm0.48}$	88.00±0.80	89.86±0.05	82.06±1.15	**88.54±0.31**	39.93±5.20	39.93±4.02
	+ `TFB` (l=ACC)	✓	77.76±0.07	$\underline{82.93\pm0.19}$	91.58±0.48*	83.84±0.75	$\underline{88.13\pm0.34}$	89.99±0.18	$\underline{82.40\pm1.96}$	87.45±0.26	39.67±7.32	37.67±6.85
	+ `TFB` (l=NLL)	✓	77.81±0.36	**83.33±0.19**	$\underline{91.76\pm0.48}$	83.81±0.39	87.80±0.16	$\underline{90.11\pm0.28}$	**82.93±1.54**	$\underline{87.64\pm0.51}$	39.67±7.32	37.33±6.65
ECE (↓)	MCD	✗	16.13±0.54	13.69±1.11	6.73±0.71	13.05±0.99	9.76±0.71	7.95±0.17	13.63±1.18	9.27±0.60	30.91±3.57	33.08±1.40
	ENS	✗	14.72±0.17	13.45±1.19	6.59±0.45	11.17±0.92	8.17±0.86	7.35±0.55	11.37±1.82	7.21±1.13	18.92±6.03	26.80±3.23
	LAP	BP	**4.18±0.11**	9.26±3.08	5.27±0.51	3.50±0.78	8.93±0.34	$\underline{1.93\pm0.22}$	7.83±1.49	7.80±1.99	**14.49±0.57**	$\underline{13.17\pm2.14}$
	MLE	-	17.02±0.46	16.35±0.68	7.00±0.53	13.83±0.65	9.77±0.81	8.69±0.21	14.45±2.19	10.78±0.50	32.46±2.60	38.41±4.44
	+ `TFB` (l=ACC)	✓	$\underline{6.71\pm0.67}$	6.14±1.18	3.42±1.44*	**3.45±0.17**	6.02±1.30	3.84±0.47	6.81±1.48	5.69±1.64	$\underline{14.63\pm1.46}$	19.68±3.27
	+ `TFB` (l=NLL)	✓	12.98±0.37	11.63±0.68	5.14±0.14	10.01±0.70	7.20±0.47	7.39±0.26	6.54±0.53	5.69±1.64	$\underline{14.63\pm1.46}$	19.68±3.27
	MAP	-	18.71±0.74	15.77±1.60	6.62±0.64	14.26±0.92	12.19±0.55	8.40±0.25	16.46±0.44	11.36±0.58	34.79±3.76	38.50±2.18
	+ `TFB` (l=ACC)	✓	8.22±0.89	7.54±0.93	4.44±1.83*	$\underline{3.82\pm0.45}$	5.22±0.09	3.89±0.15	6.11±0.60	6.07±0.69	15.74±1.99	19.64±2.92
	+ `TFB` (l=NLL)	✓	14.95±0.65	11.27±2.53	5.76±0.63	10.97±1.19	9.70±0.69	6.86±0.31	13.25±0.95	9.22±0.91	27.21±2.62	35.91±4.12
	BLoB	✗	9.93±0.22	$\underline{5.41\pm1.17}$	$\underline{2.70\pm0.87}$	4.28±0.64	$\underline{2.91\pm0.92}$	2.58±0.25	**5.61±0.40**	**2.48±0.43**	16.67±0.87	**12.78±4.18**
	BLoB-Mean	✗	15.43±0.15	12.41±1.52	4.91±0.28	9.37±1.33	6.44±0.15	6.26±0.29	11.22±0.38	6.34±0.71	26.65±3.06	25.40±5.40
	+ `TFB` (l=ACC)	✓	8.78±1.38	**4.97±0.20**	2.90±0.71*	5.11±0.80	3.09±0.03	**1.28±0.43**	$\underline{5.66\pm1.02}$	$\underline{3.53\pm0.32}$	18.59±7.26	18.07±5.49
	+ `TFB` (l=NLL)	✓	8.16±0.48	6.48±0.36	**2.44±0.50**	3.83±0.43	**2.67±0.18**	3.10±0.59	6.69±1.63	3.61±0.87	18.45±6.75	20.53±6.27
NLL (↓)	MCD	✗	0.83±0.01	0.99±0.10	0.45±0.06	0.64±0.03	0.62±0.08	0.49±0.01	1.03±0.02	0.61±0.03	1.91±0.18	2.02±0.15
	ENS	✗	0.75±0.02	0.80±0.11	0.38±0.03	0.55±0.02	0.45±0.05	0.42±0.05	0.72±0.07	$\underline{0.44\pm0.03}$	1.40±0.18	1.50±0.13
	LAP	BP	$\underline{0.56\pm0.00}$	1.18±0.02	1.04±0.01	0.51±0.00	0.94±0.00	0.43±0.00	1.17±0.01	1.11±0.00	**1.27±0.01**	**1.28±0.00**
	MLE	-	0.88±0.04	1.20±0.11	0.46±0.04	0.68±0.01	0.61±0.06	0.52±0.01	1.07±0.06	0.72±0.06	1.91±0.16	2.25±0.21
	+ `TFB` (l=ACC)	✓	$\underline{0.56\pm0.03}$	0.62±0.04	0.29±0.03*	0.45±0.01	0.43±0.01	0.31±0.01	0.67±0.05	$\underline{0.44\pm0.01}$	1.39±0.11	1.49±0.05
	+ `TFB` (l=NLL)	✓	0.68±0.03	0.85±0.02	0.33±0.03	0.53±0.01	0.46±0.04	0.42±0.00	0.66±0.02	$\underline{0.44\pm0.01}$	1.39±0.11	1.49±0.05
	MAP	-	0.99±0.07	1.12±0.23	0.46±0.03	0.74±0.07	0.79±0.02	0.52±0.01	1.19±0.04	0.83±0.06	1.97±0.13	2.32±0.10
	+ `TFB` (l=ACC)	✓	0.59±0.01	0.57±0.03	0.30±0.01*	0.45±0.01	0.47±0.03	0.30±0.00	0.70±0.07	0.51±0.02	1.37±0.09	1.54±0.06
	+ `TFB` (l=NLL)	✓	0.77±0.05	0.80±0.15	0.38±0.03	0.57±0.05	0.61±0.03	0.40±0.01	0.96±0.08	0.66±0.06	1.69±0.16	2.12±0.08
	BLoB	✗	0.58±0.00	**0.51±0.03**	**0.23±0.01**	0.43±0.01	$\underline{0.34\pm0.01}$	**0.26±0.01**	0.56±0.02	**0.35±0.02**	$\underline{1.34\pm0.04}$	$\underline{1.35\pm0.10}$
	BLoB-Mean	✗	0.74±0.02	0.73±0.04	0.29±0.03	0.47±0.03	0.37±0.02	0.32±0.02	0.67±0.07	0.39±0.03	1.53±0.13	1.54±0.15
	+ `TFB` (l=ACC)	✓	**0.55±0.02**	$\underline{0.53\pm0.02}$	$\underline{0.24\pm0.01}^*$	$\underline{0.41\pm0.01}$	$\underline{0.34\pm0.01}$	$\underline{0.27\pm0.01}$	**0.51±0.04**	**0.35±0.02**	1.36±0.13	1.44±0.10
	+ `TFB` (l=NLL)	✓	**0.55±0.01**	$\underline{0.53\pm0.04}$	**0.23±0.02**	**0.40±0.01**	**0.33±0.02**	$\underline{0.27\pm0.01}$	$\underline{0.52\pm0.05}$	**0.35±0.02**	1.36±0.13	1.46±0.11

also report results with reduced training (1 epoch) for comparison. Although training with fewer steps (early stopping) can effectively reduce model overconfidence, it typically leads to performance degradation.

The results demonstrate that `TFB` consistently improves model calibration across different backbones while maintaining competitive accuracy. Specifically, for Llama2-7B, `TFB` reduces the ECE from 4.50% to 1.24% on the combined dataset while preserving the accuracy (81.32% vs 81.41%). Similar improvements are observed with Llama3-8B, Llama3.1-8B, and Mistral-7B-v0.3, where `TFB` achieves better calibration than both the full training and early stopping baselines without sacrificing performance, suggesting its effectiveness as a general approach for enhancing LLM calibration.

F.5 `TFB` BEYOND THE NAIVE LORA

We report the detailed performance of `TFB` applied to various LoRA variants in Table 9. The baseline models are trained for 2 epochs using pre-trained Llama3.1-8B on the concatenated dataset of six commonsense reasoning tasks. Specifically, we consider the two LoRA variants:

- VeRA (Kopiczko et al., 2023): Uses shared low-rank matrices $\boldsymbol{B}$ and $\boldsymbol{A}$ across layers, with layer-specific trainable scalar vector $\boldsymbol{d}$ and bias vector $\boldsymbol{b}$. Concretely, the parameterization

Table 7: **Performance of `TFB` with different posterior families applied to the mean of BLoB on Llama3.1-8B pre-trained weights. FR:** Full-rank isotropic Gaussian noises are applied to $\Delta \boldsymbol{W}$; **C-STD:** Standard deviation matrix $\boldsymbol{\Omega} = [\Omega_{ij} = \sigma_q]$ is constant. The evaluation protocol strictly follows Table 1. "↑" and "↓" indicate that higher and lower values are preferred, respectively. **Boldface** and underlining denote the best and the second-best performance, respectively (only for `TFB` variants).

Metric	Method	In-Distribution Datasets						Out-of-Distribution Datasets (OBQA→X)			
								Small Shift		*Large Shift*	
		WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	ARC-C	ARC-E	Chem	Phy
ACC (↑)	BLoB-Mean	77.72±0.12	82.60±0.60	91.64±0.55	**83.92±0.48**	88.00±0.80	89.86±0.05	82.06±1.15	**88.54±0.31**	39.93±5.20	39.93±4.02
	+ `TFB` (FR)	77.68±0.35	82.53±1.36	**91.88±0.09***	83.36±0.82	87.67±1.06	89.53±0.33	79.87±2.07	87.39±0.23	**44.00±2.83**	**41.33±3.86**
	+ `TFB` (C-STD)	77.71±0.40	82.80±0.98	91.64±0.54*	83.31±0.75	**88.33±0.50**	89.71±0.16	80.53±0.50	87.15±1.09	43.33±0.94	39.33±3.68
	+ `TFB` (Final)	**77.76±0.07**	**82.93±0.19**	91.58±0.48*	83.84±0.75	88.13±0.34	**89.99±0.18**	**82.40±1.96**	87.45±0.26	39.67±7.32	37.67±6.85
ECE (↓)	BLoB-Mean	15.43±0.15	12.41±1.52	4.91±0.28	9.37±1.33	6.44±0.15	6.26±0.29	11.22±0.38	6.34±0.71	26.65±3.06	25.40±5.40
	+ `TFB` (FR)	12.06±0.59	6.48±0.89	3.25±0.60*	7.83±1.27	2.87±0.29	3.26±0.58	6.14±0.83	3.24±0.95	**13.45±3.69**	17.44±3.51
	+ `TFB` (C-STD)	11.61±0.44	5.43±0.74	3.78±0.39*	7.39±0.77	4.19±0.60	2.65±0.56	**5.23±1.04**	**2.79±0.22**	15.23±0.93	**15.56±3.46**
	+ `TFB` (Final)	**8.78±1.38**	**4.97±0.20**	**2.90±0.71***	**5.11±0.80**	**3.09±0.03**	**1.28±0.43**	5.66±1.02	3.53±0.32	18.59±7.26	18.07±5.49
NLL (↓)	BLoB-Mean	0.74±0.02	0.73±0.04	0.29±0.03	0.47±0.03	0.37±0.02	0.32±0.02	0.67±0.07	0.39±0.03	1.53±0.13	1.54±0.15
	+ `TFB` (FR)	0.63±0.02	**0.53±0.01**	**0.24±0.01***	0.44±0.02	0.35±0.03	0.27±0.01	0.59±0.03	0.36±0.02	**1.31±0.04**	1.41±0.07
	+ `TFB` (C-STD)	0.61±0.00	**0.53±0.02**	0.25±0.02*	0.44±0.02	**0.34±0.02**	**0.26±0.01**	0.57±0.03	0.36±0.02	**1.31±0.05**	1.40±0.06
	+ `TFB` (Final)	**0.55±0.02**	**0.53±0.02**	**0.24±0.01***	**0.41±0.01**	**0.34±0.01**	0.27±0.01	**0.51±0.04**	**0.35±0.02**	1.36±0.13	1.44±0.10

of VeRA's updated weight matrix $\boldsymbol{W}$ is modeled as:

$$\boldsymbol{W} = \boldsymbol{W}_0 + \Delta\boldsymbol{W} = \boldsymbol{W}_0 + [\text{diag}(\boldsymbol{b})]\boldsymbol{B}[\text{diag}(\boldsymbol{d})]\boldsymbol{A}. \tag{41}$$

Hence after the fine-tuning of VeRA, we can easily regroup the weight matrices into $\{\boldsymbol{B}' = [\text{diag}(\boldsymbol{b})]\boldsymbol{B}[\text{diag}(\boldsymbol{d})], \boldsymbol{A}' = \boldsymbol{A}\}$, and apply the `TFB` Bayesianization scheme illustrated in Algorithm 1.

- PiSSA (Meng et al., 2024): Employs an alternative initialization scheme while maintaining LoRA's parameterization and training procedure. Hence the `TFB` process for PiSSA is trivial.

The results in Table 9 show that `TFB` consistently improves calibration across different LoRA variants while preserving model performance. Notably, when applied to the standard LoRA, `TFB` significantly reduces the ECE from 4.74% to 1.05% on the combined dataset with minimal impact on accuracy (86.45% vs 86.70%). Similar improvements are observed with VeRA and PiSSA variants, where `TFB` achieves better calibration (reducing ECE to 1.44% and 1.17% respectively) while maintaining comparable accuracy levels. These results demonstrate that `TFB` can effectively enhance model calibration across different LoRA architectures without compromising their performance.

F.6 Improving the Inference-Time Efficiency of `TFB`

We report the detailed performance of last-layer `TFB` (`LL TFB`) in Table 10. As indicated in the table, with only $N = 10$ samples, last-layer Bayesianization provides a less effective uncertainty estimation compared to full-model Bayesianization. However, increasing the number of samples to $N = 100$ significantly enhances the posterior estimation, allowing last-layer Bayesianization to achieve better accuracy. This improvement further allows it to outperform the full-model Bayesianization in terms of NLL across most datasets.

F.7 Additional Results on Llama2-7B

We report the detailed performance of `TFB` applied to the Llama2-7B pre-trained weights in Table 11. The performance change tolerance ϵ is set adaptively to either 1% or 0.5%, depending on the checkpoint's overfitting characteristics. To determine the optimal σ_q^*, we conduct parallel experiments with eight values of $\sigma_q \in [0.01, 0.015, 0.02, 0.025, 0.03, 0.035, 0.04, 0.05]$ using a single random seed. We construct an approximate function $\widehat{\sigma}_q(p)$ through piecewise linear interpolation of the observed performance and estimate $\sigma_q^* \approx \widehat{\sigma}_q(p_0 - \epsilon)$. Similar to other baseline methods, the final results of `TFB` are reported as averages across three random seeds using σ_q^*.

In-Distribution (IND) Results. We observed several key patterns from the IND Datasets results. For example, the MLE baseline shows relatively strong accuracy but suffers from high ECE values (e.g., 29.83% on WG-S), indicating significant overconfidence. This aligns with the common challenge of LLM overconfidence during conventional fine-tuning.

Table 8: **`TFB` Performances with various LLM backbones (Touvron et al., 2023a;b; Dubey et al., 2024; Jiang et al., 2023),** where Accuracy (**ACC**) and Expected Calibration Error (**ECE**) are reported in percentages. The MLE training for each different backbone is conducted for **2 epochs** on the concatenated dataset of six commonsense reasoning tasks, with a shared hyperparameter setting; "Fewer Epochs" represents training for **1 epoch**. We use $N = 10$ samples for `TFB` during inference and rows with shading indicate training-free Bayesianization methods that use a pre-trained LoRA as their mean. "↑" and "↓" indicate that higher and lower values are preferred, respectively. **Boldface** and underlining denote the best and the second-best performance, respectively.

Metric	Method	Datasets						
		WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	Combined
ACC (↑)	Llama2-7B	**72.30**±0.90	73.24±1.34	**87.66**±0.81	72.30±0.90	83.27±1.53	**87.84**±0.57	**81.41**±0.64
	+ Fewer Epochs	63.85±3.68	69.23±0.33	86.73±0.97	63.85±3.68	79.67±1.55	86.08±0.23	76.88±1.11
	+ `TFB` (Ours)	72.03±0.88	**74.36**±1.58	87.31±1.14	**72.85**±0.96	**83.73**±0.70	87.44±0.34	81.32±0.51
	Llama3-8B	**81.45**±0.00	**84.95**±1.53	92.63±0.93	**81.45**±0.00	**88.20**±0.53	**90.19**±0.13	**86.93**±0.09
	+ Fewer Epochs	79.08±1.18	82.72±0.39	92.22±0.83	79.08±1.18	86.07±0.90	79.94±14.97	82.01±5.76
	+ `TFB` (Ours)	81.19±0.51	84.73±1.68	**92.98**±0.80	81.11±0.55	87.73±0.64	89.75±0.10	86.61±0.20
	Llama3.1-8B	**81.24**±0.05	82.72±0.19	**92.11**±1.05	**81.24**±0.05	**87.80**±2.03	**90.20**±0.11	**86.70**±0.08
	+ Fewer Epochs	78.11±0.12	**83.95**±1.00	91.17±1.17	78.11±0.12	85.33±0.90	89.38±0.35	84.96±0.22
	+ `TFB` (Ours)	80.66±0.70	82.50±0.84	91.93±1.05	81.22±0.83	87.73±1.29	89.96±0.23	86.45±0.33
	Mistral-7B-v0.3	**82.45**±0.82	**84.28**±1.53	90.94±0.27	**82.45**±0.82	87.73±0.31	**89.71**±0.48	**86.88**±0.51
	+ Fewer Epochs	79.72±0.00	83.95±0.33	**91.58**±0.63	79.72±0.00	87.53±0.31	89.20±0.20	85.71±0.11
	+ `TFB` (Ours)	81.74±0.43	84.06±1.68	90.99±0.73	81.74±0.75	**87.93**±0.42	89.71±0.32	86.64±0.28
ECE (↓)	Llama2-7B	9.17±0.74	9.37±1.27	**2.65**±0.16	9.17±0.74	5.54±0.66	**1.59**±0.49	4.50±0.37
	+ Fewer Epochs	**4.83**±1.17	**5.67**±0.92	4.46±0.23	**4.83**±1.17	4.41±0.83	6.90±1.73	2.00±0.34
	+ `TFB` (Ours)	5.44±0.80	6.06±1.54	3.83±0.74	5.50±1.55	**3.87**±1.15	2.51±0.35	**1.24**±0.22
	Llama3-8B	8.49±0.14	6.76±1.77	**2.57**±0.84	8.49±0.14	3.84±0.37	**1.88**±1.18	4.28±0.54
	+ Fewer Epochs	4.45±0.32	**4.99**±2.00	2.83±0.58	4.45±0.32	**3.14**±0.13	2.71±0.25	1.79±1.16
	+ `TFB` (Ours)	**3.47**±0.74	5.58±0.58	4.34±1.59	**4.07**±0.28	3.79±0.90	3.49±1.42	**1.64**±0.64
	Llama3.1-8B	8.58±0.56	8.58±0.29	**2.92**±0.92	8.58±0.56	3.85±1.18	**2.32**±0.27	4.74±0.28
	+ Fewer Epochs	4.76±0.91	**4.23**±0.95	3.11±0.76	4.76±0.91	3.99±0.93	3.02±0.59	1.45±0.38
	+ `TFB` (Ours)	**4.45**±0.36	4.34±1.29	2.97±0.26	**4.56**±0.68	**3.55**±0.55	3.16±0.45	**1.05**±0.06
	Mistral-7B-v0.3	8.02±1.68	6.98±1.18	4.12±0.13	8.02±1.68	5.99±0.48	3.17±0.55	5.05±0.88
	+ Fewer Epochs	5.72±2.01	4.74±1.31	**2.52**±0.79	5.72±2.01	3.50±0.75	1.70±0.47	2.47±1.09
	+ `TFB` (Ours)	**4.47**±2.00	**4.72**±0.83	2.62±0.20	**4.01**±1.08	4.10±0.26	**0.97**±0.18	**1.68**±0.53
NLL (↓)	Llama2-7B	0.58±0.01	0.69±0.03	**0.35**±0.00	0.58±0.01	0.48±0.03	**0.30**±0.00	**0.43**±0.00
	+ Fewer Epochs	0.64±0.03	0.78±0.01	0.39±0.01	0.64±0.03	0.56±0.02	0.36±0.01	0.50±0.01
	+ `TFB` (Ours)	**0.56**±0.01	**0.68**±0.02	**0.35**±0.02	**0.57**±0.01	**0.46**±0.03	0.31±0.00	**0.43**±0.00
	Llama3-8B	0.48±0.01	0.47±0.03	**0.22**±0.01	0.48±0.01	**0.35**±0.01	**0.25**±0.00	**0.34**±0.00
	+ Fewer Epochs	0.46±0.01	0.48±0.01	**0.22**±0.02	0.46±0.01	0.37±0.02	0.41±0.20	0.40±0.08
	+ `TFB` (Ours)	**0.44**±0.01	**0.45**±0.02	0.23±0.00	**0.44**±0.01	**0.35**±0.01	0.27±0.01	**0.34**±0.00
	Llama3.1-8B	0.48±0.01	0.53±0.01	0.24±0.03	0.48±0.01	**0.33**±0.03	**0.25**±0.00	0.35±0.00
	+ Fewer Epochs	0.48±0.00	**0.45**±0.00	**0.23**±0.01	0.48±0.00	0.37±0.01	0.27±0.00	0.36±0.00
	+ `TFB` (Ours)	**0.44**±0.01	0.46±0.00	**0.23**±0.02	**0.44**±0.01	**0.33**±0.02	0.27±0.00	**0.34**±0.00
	Mistral-7B-v0.3	0.46±0.04	0.47±0.02	0.28±0.01	0.46±0.04	0.36±0.03	**0.26**±0.01	0.35±0.02
	+ Fewer Epochs	0.47±0.02	0.46±0.01	**0.25**±0.01	0.47±0.02	**0.35**±0.01	**0.26**±0.00	0.35±0.01
	+ `TFB` (Ours)	**0.42**±0.02	**0.43**±0.02	0.26±0.01	**0.42**±0.02	**0.33**±0.01	**0.26**±0.01	**0.33**±0.01

`TFB` applied to BLoB-Mean demonstrates strong overall performance across the IND datasets, achieving the highest accuracy on several datasets (69.94% on WG-S, 70.72% on ARC-C, and 86.74% on ARC-E). More importantly, it achieves this while maintaining lower ECE values compared to methods like MCD and ENS, suggesting better calibrated predictions. The method also shows strong NLL performance, with values consistently among the lowest across datasets (0.62 for WG-S, 0.86 for ARC-C).

In summary, `TFB` consistently enhances the performance of baseline methods (MLE, MAP, and BLoB-Mean) across different evaluation scenarios, with notable improvements in both accuracy and calibration metrics. The improvements are particularly evident in the significant ECE reductions (e.g., from 29.83% to 16.26% for MLE on WG-S) while maintaining or improving accuracy, with the most substantial gains observed when `TFB` is combined with BLoB-Mean, achieving both the highest accuracy and lowest ECE values across most datasets.

Out-of-Distribution (OOD) Results. The OOD evaluation reveals interesting patterns across both smaller and larger distribution shifts. For smaller shifts (ARC-C and ARC-E), BLoB-Mean with `TFB` maintains strong performance, achieving 70.38% and 80.16% accuracy respectively, while keeping ECE values low (12.28% and 8.07%). This suggests robust generalization under moderate distribution shifts.

For larger shifts (Chem and Phy datasets), we see a more significant performance degradation across all methods, as expected. However, BLoB-Mean with `TFB` still maintains competitive

Table 9: **Performance of `TFB` when applied to variants of LoRAs (Hu et al., 2022; Zhang et al., 2023; Kopiczko et al., 2023; Meng et al., 2024),** where Accuracy (**ACC**) and Expected Calibration Error (**ECE**) are reported in percentages. The MLE training for each LoRA variant is conducted with pre-trained Llama3.1-8B model for **2 epochs** on the concatenated dataset of six commonsense reasoning tasks, with a shared hyperparameter setting. We set the number of samples to $N = 10$ for `TFB` during inference and rows with shading indicate training-free Bayesianization methods that use a pre-trained LoRA as their mean. "↑" and "↓" indicate that higher and lower values are preferred, respectively. **Boldface** denotes the best performance.

Metric	Method	Datasets						
		WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	Combined
ACC (↑)	LoRA	**81.24±0.05**	**82.72±0.19**	**92.11±1.05**	**81.24±0.05**	**87.80±2.03**	**90.20±0.11**	**86.70±0.08**
	+ `TFB` (Ours)	80.66±0.70	82.50±0.84	91.93±1.05	81.22±0.83	87.73±1.29	89.96±0.23	86.45±0.33
	VeRA	**78.24±1.03**	**82.39±2.55**	**90.47±1.17**	**78.24±1.03**	**86.13±0.23**	**89.27±0.27**	**84.93±0.50**
	+ `TFB` (Ours)	76.82±0.97	81.27±2.34	90.35±0.91	77.03±1.04	86.07±0.64	88.99±0.32	84.28±0.48
	PiSSA	**81.45±1.45**	**83.95±1.77**	92.22±0.54	**81.45±1.45**	**88.40±0.69**	**90.09±0.11**	**86.83±0.51**
	+ `TFB` (Ours)	80.77±1.42	82.94±1.21	**92.40±0.66**	81.32±0.78	88.13±0.42	90.01±0.23	86.61±0.43
ECE (↓)	LoRA	8.58±0.56	8.58±0.29	**2.92±0.92**	8.58±0.56	3.85±1.18	**2.32±0.27**	4.74±0.28
	+ `TFB` (Ours)	**4.45±0.36**	**4.34±1.29**	2.97±0.26	**4.56±0.68**	**3.55±0.55**	3.16±0.45	**1.05±0.06**
	VeRA	9.54±0.47	7.26±2.62	3.72±0.86	9.54±0.47	5.41±0.78	2.28±0.40	5.11±0.55
	+ `TFB` (Ours)	**5.03±0.92**	**5.92±1.53**	**2.80±0.57**	**5.09±0.87**	**3.31±0.84**	**1.78±0.40**	**1.44±0.44**
	PiSSA	7.36±0.40	8.12±1.28	2.83±1.09	7.36±0.40	3.73±1.07	2.59±0.30	4.26±0.14
	+ `TFB` (Ours)	**4.59±0.63**	**4.97±0.63**	**2.71±0.65**	**4.37±0.32**	**2.96±0.16**	**1.41±0.64**	**1.17±0.22**
NLL (↓)	LoRA	0.48±0.01	0.53±0.01	0.24±0.03	0.48±0.01	**0.33±0.03**	**0.25±0.00**	0.35±0.00
	+ `TFB` (Ours)	**0.44±0.01**	**0.46±0.00**	**0.23±0.02**	**0.44±0.01**	**0.33±0.02**	0.27±0.00	**0.34±0.00**
	VeRA	0.54±0.01	0.53±0.05	0.29±0.03	0.54±0.01	0.41±0.03	**0.27±0.01**	0.39±0.01
	+ `TFB` (Ours)	**0.51±0.01**	**0.51±0.02**	**0.27±0.02**	**0.50±0.01**	**0.39±0.02**	0.28±0.01	**0.38±0.01**
	PiSSA	0.47±0.01	0.49±0.02	**0.23±0.02**	0.47±0.01	**0.32±0.03**	**0.26±0.00**	0.35±0.00
	+ `TFB` (Ours)	**0.44±0.01**	**0.46±0.02**	**0.23±0.01**	**0.44±0.01**	**0.32±0.02**	**0.26±0.00**	**0.33±0.00**

performance, achieving 42.67% accuracy on Chem and 30.67% on Phy, while maintaining reasonable calibration metrics. The method's NLL values (1.35 and 1.46 respectively) remain competitive with other approaches, indicating relatively well-calibrated uncertainty estimates even under substantial distribution shifts.

Notable is the consistently strong performance of the BLoB variants (both w/ and w/o `TFB`) across different metrics and datasets, suggesting that this approach offers a robust framework for both in-distribution and out-of-distribution scenarios. The results demonstrate that the method successfully balances the trade-off between accuracy and calibration, particularly evident in the out-of-distribution scenarios where maintaining both aspects becomes more challenging.

Table 10: **Performance of Last-Layer `TFB` (`LL TFB`) applied to LoRA on Llama3.1-8B pre-trained weights,** where Accuracy (**ACC**) and Expected Calibration Error (**ECE**) are reported in percentages. The evaluation is done across six common-sense reasoning tasks with a shared hyper-parameter setting after 5 epochs. We sample N times during inference in the sampling-based methods. Rows with shading indicate training-free Bayesianization methods that use a pre-trained LoRA as their mean. "↑" and "↓" indicate that higher and lower values are preferred, respectively. **Boldface** and underlining denote the best and the second-best performance, respectively.

Metric	Method	#Sample (N)	Datasets					
			WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ
ACC (↑)	MLE	-	**77.87±0.54**	81.08±0.48	91.67±0.36	82.30±0.53	87.90±0.87	89.58±0.26
	+ TFB	10	77.44±0.30	82.53±1.00	91.33±0.37	82.53±0.56	**88.53±0.57**	89.75±0.25
	+ LL TFB	10	76.96±0.46	82.00±0.40	90.97±0.34	82.67±0.49	87.80±1.07	89.62±0.12
	+ LL TFB	100	77.39±0.32	82.13±0.82	91.33±0.37	82.61±0.55	87.80±0.91	89.66±0.28
	MAP	-	76.90±0.97	81.08±2.48	91.61±0.44	82.59±0.28	85.73±0.19	90.09±0.28
	+ TFB (Ours)	10	76.43±0.72	82.80±1.42	91.39±0.37	82.64±0.58	86.00±0.16	89.96±0.18
	+ LL TFB	10	76.35±0.89	83.07±1.97	91.15±0.52	82.27±0.53	85.27±0.19	90.09±0.20
	+ LL TFB	100	76.72±0.77	83.07±2.12	91.15±0.60	82.53±0.33	85.60±0.16	90.02±0.14
	BLoB	10	76.45±0.37	82.32±1.15	91.14±0.54	82.01±0.56	87.57±0.21	89.65±0.15
	BLoB-Mean	-	77.72±0.12	82.60±0.60	91.64±0.55	**83.92±0.48**	88.00±0.80	89.86±0.05
	+ TFB (Ours)	10	77.81±0.36	**83.33±0.19**	**91.76±0.48**	83.81±0.39	87.80±0.16	**90.11±0.28**
	+ LL TFB	10	77.57±1.02	82.80±0.33	91.45±0.54	83.23±0.57	88.33±0.09	89.85±0.13
	+ LL TFB	100	77.60±0.62	**83.33±0.82**	91.39±0.60	83.63±0.62	87.60±0.43	90.03±0.03
ECE (↓)	MLE	-	17.02±0.46	16.35±0.68	7.00±0.53	13.83±0.65	9.77±0.81	8.69±0.21
	+ TFB (Ours)	10	12.98±0.37	11.63±0.68	5.14±0.14	10.01±0.70	7.20±0.47	7.39±0.26
	+ LL TFB	10	14.42±0.41	13.86±0.45	6.92±0.62	10.32±0.90	8.56±0.96	7.52±0.12
	+ LL TFB	100	13.45±0.30	13.17±0.62	6.84±0.67	10.76±0.88	8.68±0.60	7.46±0.10
	MAP	-	18.71±0.74	15.77±1.60	6.62±0.64	14.26±0.92	12.19±0.55	8.40±0.25
	+ TFB (Ours)	10	14.95±0.65	11.27±2.53	5.76±0.63	10.97±1.19	9.70±0.69	6.86±0.31
	+ LL TFB	10	16.03±0.64	12.72±1.33	6.54±0.68	12.06±1.09	11.36±0.34	7.51±0.23
	+ LL TFB	100	15.56±0.97	12.84±2.17	6.38±0.66	11.80±1.14	11.22±0.38	7.30±0.41
	BLoB	10	9.93±0.22	**5.41±1.17**	2.70±0.87	4.28±0.64	2.91±0.92	2.58±0.25
	BLoB-Mean	-	15.43±0.15	12.41±1.52	4.91±0.28	9.37±1.33	6.44±0.15	6.26±0.29
	+ TFB (Ours)	10	**8.16±0.48**	6.48±0.36	**2.44±0.50**	3.83±0.43	**2.67±0.18**	3.10±0.59
	+ LL TFB	10	9.68±0.70	7.20±0.91	3.01±0.66	3.94±0.78	3.33±0.93	2.96±0.30
	+ LL TFB	100	8.88±0.32	6.47±1.55	2.84±0.50	**3.40±0.82**	3.70±0.27	**2.51±0.46**
NLL (↓)	MLE	-	0.88±0.04	1.20±0.11	0.46±0.04	0.68±0.01	0.61±0.06	0.52±0.01
	+ TFB (Ours)	10	0.68±0.03	0.85±0.02	0.33±0.03	0.53±0.01	0.46±0.04	0.42±0.00
	+ LL TFB	10	0.70±0.02	0.96±0.12	0.41±0.06	0.53±0.02	0.50±0.06	0.42±0.01
	+ LL TFB	100	0.66±0.02	0.84±0.08	0.39±0.07	0.53±0.02	0.49±0.05	0.40±0.00
	MAP	-	0.99±0.07	1.12±0.23	0.46±0.03	0.74±0.07	0.79±0.02	0.52±0.01
	+ TFB (Ours)	10	0.77±0.05	0.80±0.15	0.38±0.03	0.57±0.05	0.61±0.03	0.40±0.01
	+ LL TFB	10	0.80±0.07	0.88±0.19	0.43±0.02	0.60±0.05	0.65±0.01	0.43±0.02
	+ LL TFB	100	0.77±0.06	0.86±0.18	0.41±0.02	0.57±0.04	0.63±0.02	0.40±0.03
	BLoB	10	0.58±0.00	**0.51±0.03**	**0.23±0.01**	0.43±0.01	0.34±0.01	**0.26±0.01**
	BLoB-Mean	-	0.74±0.02	0.73±0.04	0.29±0.03	0.47±0.03	0.37±0.02	0.32±0.02
	+ TFB	10	0.55±0.01	0.53±0.04	**0.23±0.02**	0.40±0.01	0.33±0.02	0.27±0.01
	+ LL TFB	10	0.56±0.02	0.60±0.05	0.26±0.02	0.41±0.01	0.33±0.01	0.27±0.01
	+ LL TFB	100	**0.53±0.01**	0.54±0.04	0.24±0.01	**0.39±0.01**	**0.31±0.01**	**0.26±0.01**

Table 11: **Performance of different methods applied to LoRA on Llama2-7B pre-trained weights,** where Accuracy (**ACC**) and Expected Calibration Error (**ECE**) are reported in percentages. **"TF?"** denotes whether a method is **T**raining-**F**ree. The evaluation is done across six common-sense reasoning tasks with a shared hyper-parameter setting after 5,000 gradient steps. We sample $N = 10$ during inference in all sampling-based methods including **BLoB** Wang et al. (2024c) and **`TFB`**. Rows with shading indicate training-free Bayesianization methods that use a pre-trained LoRA as their mean. For `TFB`, the anchor dataset $\mathcal{D}$ is set to a randomly sampled subset of the original training set, the performance evaluation metric l is set to accuracy, and the performance drop tolerance is set adaptively to 1% or 0.5% based on whether the given mean overfits. "↑" and "↓" indicate that higher and lower values are preferred, respectively. **Boldface** and underlining denote the best and the second-best performance, respectively.

Metric	Method	TF?	In-Distribution Datasets						Out-of-Distribution Datasets (OBQA→X)			
									Small Shift		*Large Shift*	
			WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	ARC-C	ARC-E	Chem	Phy
ACC (↑)	MCD	✗	69.46±0.62	68.69±1.30	86.21±0.46	**76.45±0.04**	81.72±0.10	87.29±0.13	69.03±0.70	76.00±1.58	42.71±0.01	29.17±4.54
	ENS	✗	69.57±0.66	66.20±2.01	84.40±0.81	75.32±0.21	81.38±0.91	87.09±0.11	67.34±0.70	75.18±2.03	$\underline{43.75\pm1.04}$	30.56±2.62
	BBB	✗	56.54±7.87	68.13±1.27	85.86±0.74	73.63±2.44	82.06±0.59	87.21±0.22	67.25±1.18	75.83±0.75	42.36±0.49	30.21±2.25
	LAP	BP	69.20±1.50	66.78±0.69	80.05±0.22	75.55±0.36	82.12±0.67	86.95±0.09	69.14±1.15	74.94±0.96	44.10±1.30	$\underline{31.60\pm0.49}$
	MLE	-	68.99±0.58	69.10±2.84	85.65±0.92	74.53±0.66	81.52±0.25	86.53±0.28	66.20±0.87	75.12±0.85	40.62±2.25	28.82±1.30
	+ `TFB` (Ours)	✓	69.83±1.02	68.13±1.03	86.21±0.90	$\underline{75.95\pm0.34}$	82.80±0.35	$\underline{87.66\pm0.35}$	$\underline{69.93\pm2.11}$	$\underline{78.87\pm1.06}$	34.67±3.51	31.00±2.00
	MAP	-	68.62±0.71	67.59±0.40	$\underline{86.55\pm0.55}$	75.61±0.71	81.38±0.65	86.50±0.41	69.59±0.33	75.47±0.73	**44.79±0.00**	28.47±1.20
	+ `TFB` (Ours)	✓	69.17±1.08	67.68±1.73	85.86±0.37	75.87±0.40	$\underline{83.07\pm0.61}$	**87.74±0.23**	69.37±2.54	78.76±0.87	34.33±5.51	31.00±1.00
	BLoB	✗	68.80±0.53	67.59±0.43	86.37±0.34	73.26±1.36	81.99±1.48	86.58±0.18	67.71±1.13	76.37±0.80	**44.79±1.47**	$\underline{31.60\pm2.73}$
	BLoB-Mean	✗	**72.15±0.17**	$\underline{69.56\pm0.91}$	86.31±0.37	75.47±1.36	82.53±0.74	86.69±0.08	$\underline{69.93\pm1.20}$	76.88±0.41	41.67±2.25	**31.94±1.77**
	+ `TFB` (Ours)	✓	$\underline{69.94\pm1.68}$	**70.72±2.25**	**86.74±0.97**	73.13±2.38	**83.13±0.76**	86.36±0.26	**70.38±1.03**	**80.16±0.71**	42.67±1.15	30.67±1.53
ECE (↓)	MCD	✗	27.98±0.44	27.53±0.80	12.20±0.56	19.55±0.47	13.10±0.11	3.46±0.16	19.54±0.33	15.32±1.16	17.9±0.63	29.53±4.20
	ENS	✗	28.52±0.55	29.16±2.37	12.57±0.58	20.86±0.43	15.34±0.27	9.61±0.24	$\underline{7.59\pm1.43}$	6.44±0.83	12.04±4.57	17.52±1.28
	BBB	✗	21.81±12.95	26.23±1.47	12.28±0.58	15.76±4.71	11.38±1.07	3.74±0.10	19.90±0.66	13.41±0.85	15.67±1.23	26.10±4.76
	LAP	BP	**4.15±1.12**	16.25±2.61	33.29±0.57	7.40±0.27	8.70±1.77	$\underline{1.30\pm0.33}$	**5.84±0.64**	8.51±1.06	10.76±3.41	**13.91±0.90**
	MLE	-	29.83±0.58	29.00±1.97	13.12±1.39	20.62±0.74	12.55±0.46	3.18±0.09	22.20±0.39	16.47±0.86	21.72±0.30	29.60±1.29
	+ `TFB` (Ours)	✓	16.26±0.36	6.93±1.43	5.82±0.87	8.78±0.84	4.60±0.62	2.30±0.50	8.47±2.04	**4.64±0.75**	15.87±5.17	16.77±4.10
	MAP	-	29.76±0.87	29.42±0.68	12.07±0.55	23.07±0.14	13.26±0.82	3.16±0.23	19.31±1.46	15.68±0.51	17.55±1.95	30.25±2.18
	+ `TFB` (Ours)	✓	11.72±0.56	6.07±1.89	6.99±0.96	5.21±0.86	$\underline{3.82\pm0.60}$	2.65±0.30	8.39±0.75	$\underline{4.86\pm1.03}$	16.11±3.22	$\underline{16.35\pm2.94}$
	BLoB	✗	8.98±0.58	10.81±1.29	4.54±0.90	**3.98±1.04**	**3.64±0.54**	**1.24±0.33**	9.55±0.40	5.48±1.27	$\underline{9.77\pm1.35}$	18.29±1.35
	BLoB-Mean	✗	18.63±0.31	22.51±0.93	9.64±0.60	11.58±1.24	8.65±0.98	2.88±0.07	14.00±1.02	10.70±0.39	15.05±0.77	22.90±2.27
	+ `TFB` (Ours)	✓	$\underline{6.33\pm1.04}$	**5.77±0.32**	**3.03±0.43**	$\underline{4.07\pm1.65}$	5.94±0.46	5.37±0.44	12.28±1.24	8.07±1.01	12.36±1.73	22.02±0.30
NLL (↓)	MCD	✗	2.79±0.53	2.67±0.15	1.00±0.14	1.02±0.03	0.77±0.03	0.31±0.00	1.08±0.01	0.88±0.03	1.59±0.07	1.67±0.05
	ENS	✗	2.71±0.08	2.46±0.22	0.82±0.03	1.25±0.03	1.06±0.04	0.57±0.02	0.86±0.01	0.69±0.03	**1.28±0.00**	$\underline{1.39\pm0.03}$
	BBB	✗	1.40±0.55	2.23±0.04	0.91±0.06	0.84±0.15	0.66±0.05	0.31±0.00	1.06±0.01	0.79±0.02	1.49±0.05	1.62±0.06
	LAP	BP	**0.60±0.00**	1.03±0.04	0.88±0.00	0.57±0.01	0.52±0.01	0.31±0.00	0.81±0.00	0.70±0.02	$\underline{1.35\pm0.03}$	**1.36±0.01**
	MLE	-	3.17±0.37	2.85±0.27	1.17±0.13	0.95±0.07	0.73±0.03	0.32±0.00	1.16±0.00	0.92±0.03	1.56±0.06	1.66±0.05
	+ `TFB` (Ours)	✓	0.86±0.06	0.98±0.02	0.48±0.04	0.59±0.01	0.54±0.02	0.30±0.00	0.87±0.03	0.70±0.05	1.46±0.03	1.43±0.05
	MAP	-	2.46±0.34	2.66±0.11	0.90±0.05	1.62±0.29	0.75±0.01	0.33±0.00	1.10±0.07	0.93±0.04	1.55±0.06	1.65±0.03
	+ `TFB` (Ours)	✓	0.72±0.03	0.96±0.03	0.50±0.04	$\underline{0.55\pm0.01}$	0.53±0.02	**0.30±0.00**	0.87±0.02	0.71±0.05	1.46±0.02	1.42±0.05
	BLoB	✗	0.63±0.01	**0.84±0.00**	**0.41±0.02**	**0.54±0.01**	**0.49±0.01**	$\underline{0.31\pm0.00}$	**0.83±0.01**	**0.60±0.01**	1.38±0.01	1.46±0.02
	BLoB-Mean	✗	0.79±0.01	1.27±0.02	0.57±0.03	0.60±0.03	0.56±0.00	0.32±0.01	0.89±0.02	0.67±0.02	1.44±0.00	1.53±0.02
	+ `TFB` (Ours)	✓	$\underline{0.62\pm0.03}$	$\underline{0.86\pm0.01}$	$\underline{0.42\pm0.03}$	0.56±0.03	$\underline{0.50\pm0.01}$	0.34±0.00	$\underline{0.84\pm0.03}$	0.61±0.01	$\underline{1.35\pm0.01}$	1.46±0.06