Re-implementation of BLoB: Bayesian Low-Rank Adaptation by Backpropagation for Large Language Models

Fredrik Ström
frest@kth.se
KTH

William Rosengren
wrose@kth.se
KTH

Yuusuf Dahlstrand
yuusufd@kth.se
KTH

Abstract

Bayesian Low-Rank Adaptation by Backpropagation (BLoB) is a recently proposed parameter-efficient fine-tuning method that introduces Bayesian uncertainty into Low-Rank Adaptation (LoRA) for large language models. By learning a variational posterior over low-rank adapter parameters during training, BLoB aims to improve calibration and robustness without incurring the cost of full Bayesian fine-tuning. In this work, we first reproduce the original BLoB results on `RoBERTa-base` across a subset of GLUE and SuperGLUE classification tasks, strictly following the authors' released code and hyperparameter configurations. Our reproduction confirms the main findings of the original paper: BLoB achieves competitive accuracy while providing improved uncertainty estimates compared to deterministic baselines, with only minor quantitative deviations attributable to stochastic variation. We then extend BLoB beyond its original scope by applying it to encoder–decoder architectures for abstractive dialogue summarization using `facebook/bart-base` on the DIALOGSUM dataset. In this generative setting, we conduct a systematic hyperparameter sensitivity analysis, revealing a strong dependence on the KL scaling parameter and identifying posterior collapse as a critical failure mode under insufficient regularization. We then analyze how BLoB and LoRA compared across near-deterministic inference. Finally, we analyze Bayesian posterior sampling in BLoB and compare it to LoRA to study the impact of weight-space uncertainty. Our results show that Bayesian sampling enables BLoB to explore a broader space of high-quality summaries, yielding higher expected semantic and lexical overlap with reference summaries compared to LoRA. This improvement arises from weight-space uncertainty rather than decoding stochasticity alone. Overall, our findings validate the reproducibility of BLoB on classification tasks and demonstrate that Bayesian Low-Rank Adaptation provides meaningful benefits in generative settings, while also highlighting practical challenges and sensitivities that must be addressed when deploying the method in practice.

1 Introduction

Large Language Models (LLMs) achieve strong performance across many tasks, but adapting them to downstream applications remains computationally expensive. Parameter-Efficient Fine-Tuning (PEFT) methods such as Low-Rank Adaptation (LoRA) [7] alleviate this cost by freezing the backbone model and training only a small set of additional parameters. However, PEFT methods often produce poorly calibrated models, with fine-tuned LLMs exhibiting overconfidence even when predictions are incorrect [5].

Bayesian methods provide a principled framework for modelling epistemic uncertainty, but most existing approaches do not scale to large language models. Post-hoc techniques such as Laplace-

Project report for DD2610

LoRA [16] approximate uncertainty only around a learned point estimate, while fully Bayesian fine-tuning is computationally infeasible at scale.

To bridge this gap, Wang et al. [15] introduced *Bayesian Low-Rank Adaptation by Backpropagation* (BLoB), which incorporates uncertainty directly into LoRA during training. BLoB achieves improved calibration and robustness on classification benchmarks while retaining the efficiency of PEFT.

However, BLoB is evaluated exclusively on encoder-only classification tasks, leaving its behaviour in generative settings unexplored. This is particularly relevant for text generation, where uncertainty manifests as variability in generated outputs rather than explicit confidence scores.

In this work, we reproduce the original BLoB results on `RoBERTa-base` and extend the method to sequence-to-sequence summarization using `facebook/bart-base`. We further analyze the sensitivity of BLoB to its Bayesian hyperparameters in the generative encoder–decoder setting, and compare Bayesian posterior sampling against LoRA to study how weight uncertainty affects generative performance.

2 Code repository

`https://github.com/Fredrikstrm/BLoB-extension-DLA`

3 Related Work

Parameter-Efficient Fine-Tuning (PEFT). Fine-tuning full Large Language Models (LLMs) is resource-intensive and often impractical. Early PEFT approaches, such as Adapters [6], introduce bottleneck layers within Transformer blocks. More recently, Low-Rank Adaptation (LoRA) [7] has become a standard, injecting trainable low-rank matrices into self-attention layers. While LoRA efficiently recovers performance comparable to full fine-tuning, it typically produces a deterministic point estimate of the weights, which fails to capture the model's epistemic uncertainty.

Bayesian Deep Learning (BDL) in LLMs. BDL offers a principled framework for uncertainty estimation by placing a prior over model weights and computing the posterior. Classic variational inference methods like Bayes by Backprop (BBB) [2] are computationally expensive for modern LLMs due to the doubling of parameters (means and variances). Approximate methods, such as Monte Carlo (MC) Dropout [4] and Deep Ensembles [9], offer alternatives but often incur high inference latency or prohibitive training costs unsuitable for LLM scales.

Bayesian PEFT. Recent work has sought to combine PEFT efficiency with Bayesian uncertainty. A prominent example is *Laplace-LoRA* [16], which applies a post-hoc Laplace approximation to the LoRA weights after standard training. While efficient, it relies on the local curvature of the loss surface at a single mode found by SGD and does not account for the posterior during the optimization trajectory. In contrast, *BLoB* [15] (the subject of this replication) employs variational inference to learn the posterior distribution of LoRA weights directly during training via backpropagation. This allows for a more flexible approximation of the uncertainty manifold. Our work extends this lineage by applying variational LoRA specifically to the encoder-decoder attention mechanisms in generative architectures (BART), a setting explicitly identified as future work in the original BLoB paper.

4 Methods

4.1 Bayesian Low-Rank Adaptation (BLoB): Method Overview

LoRA inserts trainable low-rank adapters into selected linear projections of a frozen backbone model (Figure 1a). The adapted weight is typically written as

$$\mathbf{W}' = \mathbf{W} + \Delta\mathbf{W}, \qquad \Delta\mathbf{W} = \mathbf{BA}, \tag{1}$$

where $\mathbf{W}$ is frozen and $\mathbf{A}, \mathbf{B}$ are learned point estimates.

Laplace-LoRA (Figure 1b) keeps the same point-estimate training, and approximates uncertainty *post hoc* by fitting a Laplace approximation around the learned adapter weights.

BLoB (Figure 1c) instead treats the adapter parameters as random variables throughout training, placing a variational posterior $q_\phi(\mathbf{w})$ over the low-rank parameters $\mathbf{w}$ (i.e., entries of $\mathbf{A}, \mathbf{B}$) and optimizing an ELBO objective:

$$\mathcal{L}_{\text{ELBO}}(\phi) = \mathbb{E}_{q_\phi(\mathbf{w})}[\mathcal{L}_{\text{task}}(\mathbf{w})] + \beta \, \text{KL}(q_\phi(\mathbf{w}) \,\|\, p(\mathbf{w})) \,, \tag{2}$$

where $p(\mathbf{w})$ is a prior and β scales the KL regularization. Sampling from $q_\phi(\mathbf{w})$ is implemented via the reparameterization trick, enabling backpropagation through stochastic adapter draws while keeping the pretrained backbone fixed.

At inference time, BLoB can perform Monte Carlo prediction by drawing $\mathbf{w}^{(s)} \sim q_\phi(\mathbf{w})$ and aggregating outputs across samples, yielding a posterior predictive distribution rather than a single deterministic prediction.

At inference time, BLoB supports sampling-based inference by drawing $\mathbf{w}^{(s)} \sim q_\phi(\mathbf{w})$ and evaluating the model under multiple posterior samples. Aggregating outputs across such samples provides an empirical approximation of the posterior predictive behavior, in contrast to a single fixed-weight prediction.

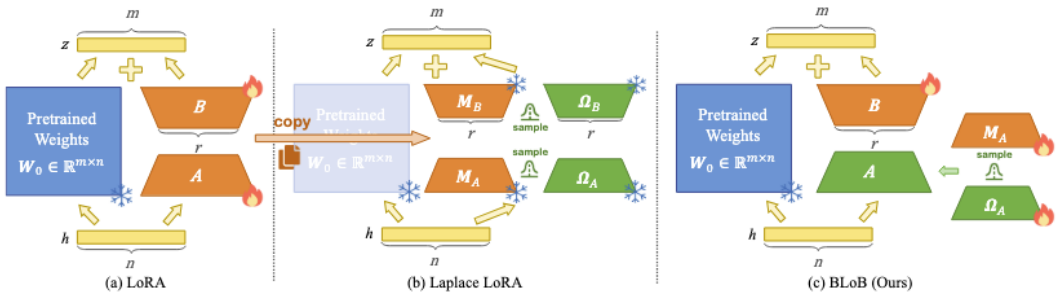


Figure 1: Overview of Bayesian Low-Rank Adaptation by Backpropagation (BLoB) compared to LoRA [7] and Laplace-LoRA [16] (Figure taken from the original BLoB paper [15]).

4.2 Extension to Encoder–Decoder Architectures

We extend the method to *encoder–decoder models* for generative sequence-to-sequence learning by applying BLoB to the `facebook/bart-base` architecture and evaluating it on abstractive summarization.

The Bayesian formulation of BLoB remains unchanged. Low-rank adaptation matrices are introduced via LoRA modules, while all pretrained backbone parameters remain frozen. In contrast to classification, summarization is trained using a conditional sequence likelihood. Given an input dialogue x and target summary $y = (y_1, \ldots, y_T)$, the task loss is the token-level negative log-likelihood

$$\mathcal{L}_{\text{task}}(\mathbf{w}) = -\sum_{t=1}^{T} \log p_{\mathbf{w}}(y_t \mid y_{<t}, x), \tag{3}$$

where $\mathbf{w}$ denotes the low-rank adapter parameters.

As in the original BLoB method, these parameters are modeled by a variational posterior $q_\phi(\mathbf{w})$ and training minimizes the evidence lower bound (ELBO)

$$\mathcal{L}_{\text{ELBO}}(\phi) = \mathbb{E}_{q_\phi(\mathbf{w})}[\mathcal{L}_{\text{task}}(\mathbf{w})] + \beta \, \gamma \, \text{KL}(q_\phi(\mathbf{w}) \,\|\, p(\mathbf{w})) \,, \tag{4}$$

where β controls the overall strength of Bayesian regularization and γ scales the prior variance, following the formulation used in the original BLoB implementation.

BLoB-adapted LoRA modules are inserted into selected projection layers of both the encoder and the decoder. Each posterior sample induces a distinct encoder–decoder model, and sampling multiple realizations at inference approximates the posterior predictive distribution

$$p(y \mid x) \approx \frac{1}{S} \sum_{s=1}^{S} p(y \mid x, \mathbf{w}^{(s)}), \quad \mathbf{w}^{(s)} \sim q_\phi(\mathbf{w}). \tag{5}$$

This extension enables the study of epistemic uncertainty in a generative setting, where posterior uncertainty manifests as variability in the generated text rather than solely in predictive confidence.

Apart from adapting the loss function and model architecture to the sequence-to-sequence setting, all other aspects of BLoB, including frozen backbone parameters, variational training, and reparameterized sampling, are preserved from the original formulation.

4.3 Non-sampling and Bayesian Inference

We consider two inference regimes. In the *non-sampling* setting, a single posterior sample of the weights is used to generate one summary per input. This corresponds either to using a fixed posterior sample or, as discussed in the original BLoB paper, effectively using the posterior mean weights. Importantly, the authors note that using the posterior mean weights, BLoB can yield improved predictions compared to standard LoRA. Non-sampling inference in BLoB is not equivalent to standard LoRA, as the adapter parameters have been optimized under a Bayesian objective rather than as point estimates during training.

In the *Bayesian* setting, multiple posterior samples are drawn from the learned BLoB weight distribution, and a separate summary is generated for each weight sample. This regime is explicitly designed to expose the effect of Bayesian weight uncertainty on model outputs. By generating multiple summaries per input, we enable Monte Carlo estimation of expected performance metrics and facilitate analysis of output variability and uncertainty induced by the posterior.

For comparison, LoRA summaries are generated using a single fixed set of adapter weights, and any variability arises from stochastic decoding. In contrast, BLoB incorporates an additional source of variability through posterior weight sampling. By generating many summaries per dialogue under both approaches, we analyse differences in the resulting score distributions and assess the practical impact of Bayesian weight uncertainty on generative performance.

4.4 Metrics

4.4.1 BERTScore

To measure semantic similarity between generated and reference summaries we use *BERTScore* [17]. Unlike lexical overlap metrics, BERTScore operates in a contextual embedding space and is therefore robust to paraphrasing and stylistic variation. Given a generated summary $\hat{y}_i$ and a reference summary y_i, BERTScore computes precision, recall, and F1 by aligning token embeddings obtained from a pretrained language model using cosine similarity. In the experiments, BERTScore embeddings are computed using the pretrained `microsoft/deberta-xlarge-mnli` model, which has been shown to provide strong semantic representations for natural language inference and summarization tasks. Formal definition of BERTScore is provided in Appendix A.1.

4.4.2 ROUGE

We additionally evaluate summarization quality using the *ROUGE* family of metrics [10], which measure lexical overlap between generated and reference summaries. ROUGE is a standard evaluation protocol for abstractive summarization and provides a complementary perspective to embedding-based metrics such as BERTScore.

Specifically, we report ROUGE-1, ROUGE-2, and ROUGE-L F1 scores, corresponding to unigram overlap, bigram overlap, and longest common subsequence (LCS) overlap, respectively. ROUGE-1 and ROUGE-2 capture surface-level content overlap, while ROUGE-L accounts for sentence-level fluency and ordering by measuring the longest shared subsequence between summaries.

For each generated summary, ROUGE scores are computed against multiple human reference summaries, and the maximum score across references is selected. Final results are obtained by averaging scores across all test instances. All ROUGE scores reported in this work are F1-based. Formal definitions of ROUGE-n and ROUGE-L are provided in Appendix A.2.

5 Data

For classification-based reproduction experiments, we follow the original BLoB paper and evaluate on five natural language understanding tasks drawn from the GLUE and SuperGLUE benchmarks [14, 13]: RTE, MRPC, CoLA, WiC, and BoolQ. These datasets cover a range of linguistic phenomena, including textual entailment, paraphrase detection, grammatical acceptability, word sense disambiguation, and question answering. All datasets are used in their standard train/validation/test splits as provided by the original benchmarks.

For the summarization (seq2seq) task we fine-tune `facebook/bart-base` using the DIALOGSUM dataset [3], which is publicly available on Hugging Face under the identifier `knkarthick/dialogsum`. DIALOGSUM is a large-scale benchmark for abstractive dialogue summarization, consisting of multi-turn dialogues paired with human-written summaries. The dataset contains a total of 13,460 dialogues and is divided into 12,460 training, 500 validation, and 1,500 test instances. The test split comprises 500 unique dialogues, each annotated with three independent human reference summaries, enabling more robust and reliable evaluation. No additional preprocessing was required.

6 Experiments and findings

6.1 Experiment I: Reproduction on GLUE and SuperGLUE Classification Tasks

6.1.1 Experimental Setup

This experiment evaluates the reproducibility of BLoB on classification tasks using the same datasets, model architecture, and evaluation protocol as the original paper. Dataset details are provided in Section 5.

All experiments use the `roberta-base` backbone with Bayesian Low-Rank Adaptation applied to the classification head. Training and evaluation follow the authors' released implementation as closely as possible. In particular, we execute the official bash scripts provided in the repository without modifying hyperparameters or optimization settings.

The training configuration uses AdamW with learning rate 5×10^{-4}, batch size 32, warmup ratio 0.06, and a maximum sequence length of 256. Bayesian regularization is enabled using the authors' default hyperparameters ($\beta = 0.2$, $\gamma = 8$) together with the KL reweighting schedule described in the original codebase. Each experiment is run with three different random seeds to account for stochastic variation.

Following the original study, we evaluate models using Accuracy (ACC), Expected Calibration Error (ECE), and Negative Log-Likelihood (NLL). Accuracy captures predictive performance, while ECE and NLL assess calibration and uncertainty quality, which are central to the claims of the BLoB framework. Lower values indicate better performance for ECE and NLL.

Table 1 reports our reproduced results, and Table 2 lists the corresponding numbers from the original paper for comparison.

6.1.2 Results and Comparison to the Original Paper

From Table 1 our reproduced results closely follow the trends reported in the original paper, Table 2, with only modest quantitative differences. In terms of accuracy, performance is largely comparable and in several cases slightly higher, notably on WiC, CoLA, and BoolQ. Results on RTE and MRPC remain within the reported standard deviations of the original study.

For uncertainty-related metrics, calibration results exhibit a mixed but consistent pattern. Reproduced ECE values are slightly higher on RTE and MRPC, indicating marginally worse calibration, while CoLA shows substantially improved calibration compared to the original paper. WiC yields nearly identical ECE values across both experiments. Similar trends are observed for NLL, where reproduced values are slightly worse on smaller datasets but comparable or marginally improved on larger ones.

Importantly, although absolute metric values differ, the relative behavior across tasks is consistent with the original findings. Datasets with more training examples tend to show more stable calibration, while smaller datasets exhibit higher variance in both accuracy and uncertainty metrics.

Table 1: Reproduced BLoB performance on GLUE tasks (mean ± std over 3 seeds) using RoBERTa-base. Accuracy (ACC) and Expected Calibration Error (ECE) are reported in %, lower is better for ECE and NLL.

	Datasets				
Metric	**RTE**	**MRPC**	**WiC**	**CoLA**	**BoolQ**
ACC	75.81 ± 1.81	87.66 ± 1.44	67.19 ± 2.10	83.32 ± 0.72	78.92 ± 0.29
ECE	10.97 ± 2.45	5.03 ± 1.67	13.02 ± 1.21	3.68 ± 1.14	2.67 ± 0.66
NLL	0.56 ± 0.05	0.32 ± 0.02	0.69 ± 0.01	0.37 ± 0.01	0.46 ± 0.01

Table 2: Original performance on GLUE tasks (mean ± std over 3 seeds) using RoBERTa-base from [15]. Accuracy (ACC) and Expected Calibration Error (ECE) are reported in %, lower is better for ECE and NLL.

	Datasets				
Metric	**RTE**	**MRPC**	**WiC**	**CoLA**	**BoolQ**
ACC	75.45 ± 0.51	88.73 ± 0.35	64.26 ± 1.00	80.89 ± 0.24	75.49 ± 1.60
ECE	8.97 ± 0.98	3.30 ± 0.19	13.03 ± 0.85	7.83 ± 0.27	2.90 ± 0.12
NLL	0.48 ± 0.01	0.26 ± 0.00	0.67 ± 0.01	0.46 ± 0.01	0.51 ± 0.01

6.1.3 Findings

The discrepancies between our reproduced results (Table 1) and those reported in the original paper (Table 2) are not large enough to contradict the original conclusions. Most differences fall within one to two standard deviations and can plausibly be attributed to stochastic variation across random seeds, minor implementation-level differences, and the sensitivity of Bayesian fine-tuning to optimization dynamics.

Overall, this reproduction supports the core claims of the original work: BLoB maintains competitive classification accuracy while providing improved uncertainty estimation compared to non-Bayesian baselines. By strictly adhering to the authors' provided scripts and hyperparameter settings, our results strengthen confidence in the reproducibility of the reported empirical findings.

6.2 Experiment II: Hyperparameter Sensitivity Analysis

6.2.1 Experimental Setup

To study the sensitivity of BLoB to its Bayesian regularization in a generative setting, we conducted a grid search over the two Bayes-specific coefficients β and γ. All experiments fine-tune `facebook/bart-base` using the `blob_summarization` wrapper on the DIALOGSUM dataset. The training configuration is fixed across runs: AdamW with learning rate 5×10^{-4}, batch size 16, warmup ratio 0.06, and maximum input length 512. To keep the sweep computationally feasible, training is limited to 1500 steps, and only a final evaluation is performed.

We vary $\beta \in \{0.1, 0.2, 0.3\}$ and $\gamma \in \{4, 8, 12\}$, yielding 9 configurations. Training uses one posterior sample per update (`bayes-train-n-samples=1`), while evaluation employs Monte Carlo inference with $S = 10$ posterior samples. All runs use a single random seed.

BLoB optimizes the variational objective defined in Section 4.2, Eq. (4), where $\mathbf{w}$ denotes the low-rank adapter parameters. The task loss corresponds to the token-level negative log-likelihood defined in Eq. (5), computed using token-level cross entropy with padding tokens ignored.

6.2.2 Results

Table 3 reports the final KL loss, ELBO, and ROUGE scores for all hyperparameter configurations. For $\gamma = 4$, the KL loss collapses to zero across all values of β, whereas larger values of γ retain a non-zero KL term at the end of training. ELBO values decrease substantially as γ increases, reflecting

stronger Bayesian regularization. ROUGE scores remain similar across configurations, with only minor variation.

Table 3: Hyperparameter sensitivity of BLoB on DIALOGSUM with `facebook/bart-base`. We report final KL loss, ELBO, and ROUGE-1/2/L for different values of the KL weight β and prior scale γ.

β	γ	KL Loss	ELBO	ROUGE-1	ROUGE-2	ROUGE-L
0.1	4	0.0	1200935.81	0.4336	0.1698	0.3524
0.1	8	1357.78	356901.46	0.4336	0.1669	0.3508
0.1	12	690.86	47876.46	0.4324	0.1641	0.3474
0.2	4	0.0	1425236.30	0.4325	0.1699	0.3532
0.2	8	2205.69	577324.49	0.4371	0.1694	0.3542
0.2	12	2679.37	187909.72	0.4341	0.1651	0.3487
0.3	4	0.0	1560386.59	0.4344	0.1709	0.3531
0.3	8	2720.78	711101.24	0.4354	0.1697	0.3530
0.3	12	4300.17	302429.51	0.4360	0.1701	0.3518

6.2.3 Findings

The results show that BLoB's Bayesian behaviour is highly sensitive to the KL scaling parameter γ. As summarized in Table 3, small values ($\gamma = 4$) consistently lead to posterior collapse, where the KL divergence vanishes by the end of training. This behaviour is also visible in the KL trajectories (Figure 3), indicating that posterior sampling becomes ineffective and the model reduces to a deterministic LoRA-like solution.

In contrast, larger values of γ preserve a non-zero KL term throughout training (Table 3, Figure 3), maintaining uncertainty in the low-rank adapter weights. This effect is further reflected in the ELBO curves, where higher γ prevents early collapse and sustains regularization pressure (Figure 4).

Overall, these findings identify γ as the primary control for balancing Bayesian regularization and task performance. While ROUGE scores remain relatively stable across configurations, posterior collapse fundamentally alters the interpretation of uncertainty estimates, highlighting the need for careful hyperparameter tuning when applying BLoB to generative tasks.

6.3 Experiment III: Summarization - non-sampling

6.3.1 Experimental Setup

To evaluate summarization quality, we use beam-search decoding, in order to remove variance from decoding-noise. For a model with parameters θ (LoRA or BLoB), the generated summary for dialogue x_i is

$$\hat{y}_i = f_\theta^\star(x_i), \tag{6}$$

where $f_\theta^\star$ denotes beam search with fixed hyperparameters (beam size $b = 4$, length penalty 1.0, and maximum length 1028). This produces a single summary per dialogue. For BLoB, decoding is performed using the posterior mean of the learned weights.

6.3.2 Evaluation

Each dialogue x_i is associated with a set of reference summaries $\mathcal{Y}_i = \{y_{i,1}, \ldots, y_{i,R}\}$, where $R = 3$. Given a similarity metric $q(\cdot, \cdot) \in [0, 1]$ (ROUGE or BERTScore), the score for a generated summary $\hat{y}_i$ is computed as

$$s_i = \max_{y_{i,r} \in \mathcal{Y}_i} q(\hat{y}_i, y_{i,r}). \tag{7}$$

Final results are obtained by averaging across the test set of $N = 500$ dialogues:

$$\bar{s} = \frac{1}{N} \sum_{i=1}^{N} s_i. \tag{8}$$

We report the mean and standard deviation of s_i across samples. ROUGE metrics are F1-based, and BERTScore corresponds to the F1 variant.

6.3.3 Results

Results are shown in Table 4.

Table 4: Comparison of LoRA and BLoB under deterministic beam-search decoding for dialogue summarization. Scores are averaged over 500 test dialogues using the maximum score across three reference summaries per sample. All metrics report F1 (mean $\pm$ standard deviation).

Model	BERTScore	ROUGE-1	ROUGE-2	ROUGE-L
BLoB	0.459 ± 0.163	0.487 ± 0.141	0.239 ± 0.167	0.409 ± 0.151
LoRA	0.459 ± 0.158	0.482 ± 0.132	0.231 ± 0.152	0.401 ± 0.142

6.3.4 Findings

Under near-deterministic decoding, LoRA and BLoB achieve nearly identical performance across all evaluation metrics. In particular, BERTScore F1 is identical for both models, while ROUGE-1, ROUGE-2, and ROUGE-L differ marginally and within one standard deviation. This indicates that, when posterior uncertainty in BLoB is collapsed to its mean, its summarization quality closely matches that of a point-estimate LoRA model.

These results suggest that any advantages of Bayesian fine-tuning in BLoB do not manifest under deterministic decoding with fixed weights. Consequently, this experiment serves as a controlled baseline, confirming that differences observed in subsequent stochastic or sampling-based experiments cannot be attributed to disparities in deterministic decoding performance.

6.4 Experiment IV: Summarization - Bayesian sampling

6.4.1 Experimental Setup

To analyze the Bayesian behavior of BLoB, we perform inference by sampling adapter weights from the learned variational posterior and generating multiple summaries per dialogue.

Let $q_\phi(A)$ denote the variational posterior over adapter weights. For each dialogue x_i, we draw S samples

$$A^{(s)} \sim q_\phi(A), \qquad s = 1, \ldots, S, \tag{9}$$

construct the corresponding parameters $\theta^{(s)}$, and generate summaries

$$\hat{y}_i^{(s)} = f_{\theta^{(s)}}(x_i). \tag{10}$$

We use $S = 100$ samples for each of the first 100 test dialogues, yielding 10,000 summaries per model.

This procedure constitutes a Monte Carlo approximation of the posterior predictive distribution over summaries. Given three reference summaries $\{y_{i,r}\}_{r=1}^{3}$ and an evaluation metric $q(\cdot, \cdot)$, we compute

$$\mathbf{q}_i^{(s)} = \left(q(\hat{y}_i^{(s)}, y_{i,r})\right)_{r=1}^{3} \tag{11}$$

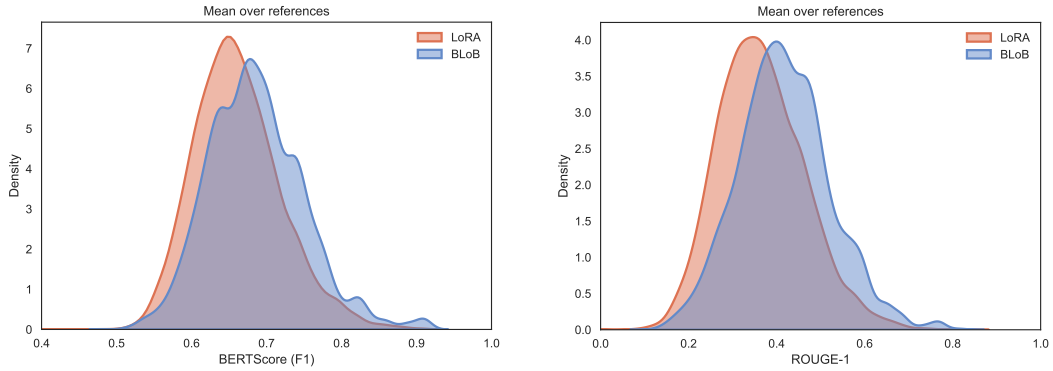
Scores are aggregated across references using

$$\text{mean}(\mathbf{q}_i^{(s)}) = \tfrac{1}{3}\sum_{r=1}^{3} q_{i,r}^{(s)}, \qquad \max(\mathbf{q}_i^{(s)}) = \max_r q_{i,r}^{(s)}, \qquad \min(\mathbf{q}_i^{(s)}) = \min_r q_{i,r}^{(s)}. \tag{12}$$

For each model and metric, this yields 10,000 scores, visualized via kernel density estimates. LoRA variability arises solely from stochastic decoding, whereas BLoB additionally reflects epistemic uncertainty through posterior weight sampling.

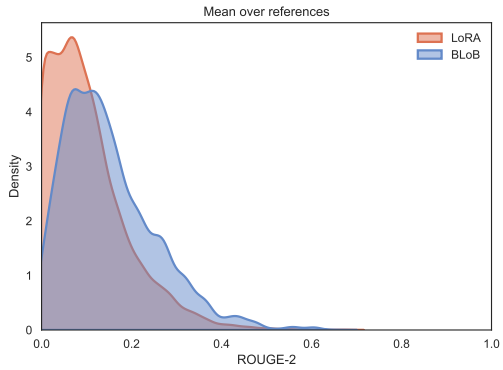
6.4.2 Results

See figure 2

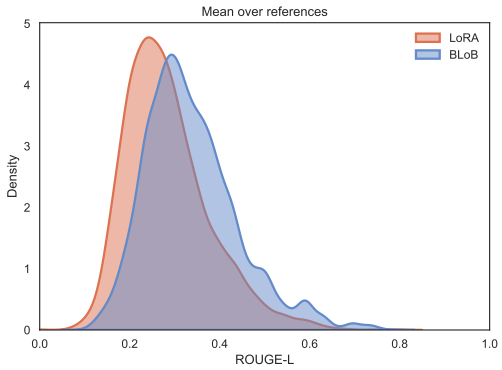


(a) BERTScore (F1)

(b) ROUGE-1



(c) ROUGE-2



(d) ROUGE-L

Figure 2: Experiment IV: Kernel density estimates (KDEs) of per-dialogue scores under Bayesian sampling. We plot the distribution of scores aggregated as the mean over references for each generated sample.

6.4.3 Findings

Figure 2 illustrates the distribution of per-sample scores under Bayesian inference. BLoB generates summaries by sampling adapter weights from the variational posterior, whereas LoRA relies exclusively on stochastic decoding with a fixed set of weights. Across all metrics, BLoB exhibits a consistent rightward shift relative to LoRA (see figures 5,6, 7, 8 in appendix) indicating higher expected performance when sampling from the posterior predictive distribution.

For semantic similarity, as measured by BERTScore (F1), BLoB achieves higher average alignment with the reference summaries. This indicates that posterior sampling does not introduce harmful semantic drift, instead, it allows BLoB to explore multiple semantically plausible summaries. In contrast, LoRA's variability is limited to decoding noise, which primarily induces surface-level paraphrasing and does not substantially alter which information is selected for inclusion.

The effect is more pronounced for ROUGE-based metrics. Because BLoB samples a different set of adapter weights for each generation, it explores a broader region of the summary space. As a result, when generating many summaries per dialogue, BLoB has a higher probability of producing outputs that closely align with the lexical and structural choices of at least one reference. LoRA, by contrast, repeatedly samples from a single fixed model, and decoding noise seem to be insufficient to explore alternative content realizations that better match different references, hence the LoRA systematically shifted to the left.

Interestingly, despite increased exploration, BLoB does not exhibit degraded worst-case behaviour. In contrary, the score distributions indicate that posterior samples are concentrated around high-quality solutions, such that even less favourable samples remain competitive with LoRA. Overall, these results suggest that Bayesian Low-Rank Adaptation enables effective exploration of multiple valid

summaries through weight-space uncertainty, whereas LoRA's variability is largely confined to superficial lexical variation driven by decoding noise.

7 Challenges

Reproducing and extending the BLoB framework involved several practical and methodological challenges. While the paper provides a clear conceptual description of the method, many implementation details were not fully specified and required close inspection of the authors' codebase.

A first issue concerned the provided Bash scripts, which were not directly compatible with macOS environments and required modification to run correctly. In addition, some scripts did not exactly match the experimental settings described in the paper, for instance, defaulting to `RoBERTa-large` in cases where the paper reports results for `RoBERTa-base`. Identifying and resolving such discrepancies required careful cross-referencing between the paper and the code.

The repository itself is highly modular and complex, which, while flexible, made it difficult to trace the exact training and inference behaviour of BLoB adapters. This complexity became particularly apparent when extending the framework to generative summarization tasks, which are not the primary focus of the original work. Adapting BLoB to an encoder–decoder setting required additional care to ensure that output variability stemmed from Bayesian weight sampling rather than stochastic decoding.

Finally, BLoB introduces several additional hyperparameters related to posterior regularization, making comprehensive hyperparameter sensitivity analyses computationally expensive. As a result, such experiments were conducted on reduced datasets and with fewer evaluation steps to remain feasible.

Overall, these challenges highlight the gap between the high-level methodological description in the paper and the practical effort required to reproduce and extend the approach in new settings.

8 Conclusion

In this project, we conducted a comprehensive reproduction and extension study of Bayesian Low-Rank Adaptation by Backpropagation (BLoB). By replicating the original experiments on `RoBERTa-base` classification tasks using the authors' released code and configurations, we confirmed that the reported calibration and robustness benefits of BLoB are reproducible and stable under independent evaluation.

Beyond reproduction, we extended BLoB to a novel setting by applying it to encoder–decoder architectures for abstractive summarization. This extension enabled us to study Bayesian low-rank uncertainty in a generative context, where uncertainty manifests as variability in generated outputs rather than classification confidence scores. Our experiments demonstrate that Bayesian inference through posterior weight sampling allows BLoB to explore a broader space of valid summaries, leading to higher expected ROUGE and BERTScore performance compared to deterministic LoRA under repeated sampling.

Our hyperparameter sweep shows that BLoB can collapse to LoRA-like behaviour under weak KL regularization, underscoring the need for careful tuning.

Overall, our results suggest that Bayesian Low-Rank Adaptation is a promising approach for uncertainty-aware fine-tuning beyond classification tasks. However, practical deployment requires careful consideration of computational overhead, hyperparameter sensitivity, and inference cost, particularly in generative settings. An important direction for future work is the systematic evaluation of out-of-distribution (OOD) generalization for generative tasks, mirroring the extensive OOD analyses conducted in the original paper for classification. Such studies would provide deeper insight into whether Bayesian weight uncertainty translates into improved robustness and uncertainty awareness in open-ended text generation. Additional extensions could explore adaptive KL schedules, more efficient posterior sampling strategies, and applications to other generative tasks such as question answering and long-form text generation.

9 Ethical consideration, societal impact, alignment with UN SDG targets

Environmental Impact. Deep learning research incurs a significant carbon footprint due to computational costs [12, 11]. In this project, we mitigated energy consumption by utilizing parameter-efficient fine-tuning methods (LoRA and BLoB), which require significantly fewer trainable parameters than full fine-tuning. Our experiments were conducted using cloud-based GPU resources via Modal, with a total compute time of approximately 30 GPU-hours across all experiments. While these compute requirements are modest compared to large-scale pre-training of language models, we acknowledge the cumulative environmental impact associated with repeated experimentation and hyperparameter tuning.

Reliability and Safety. The primary societal contribution of this work lies in improving the reliability of LLMs. Standard LLMs often suffer from hallucinations delivered with high confidence [5, 8]. By improving uncertainty calibration via BLoB, we move towards systems that can "know when they do not know." These ethical motivations are shared by the original BLoB framework [15], which aims to improve the reliability and robustness of fine-tuned LLMs without incurring the costs of full Bayesian training. This aligns with UN Sustainable Development Goal (SDG) 9 (Industry, Innovation, and Infrastructure) by fostering more resilient AI infrastructure, and partially with SDG 3 (Good Health and Well-being), as calibrated uncertainty is a prerequisite for deploying AI in high-stakes environments such as medical decision-making.

Risks. However, better uncertainty estimation does not guarantee truthfulness. There is a risk that users may over-rely on calibrated confidence scores, assuming that a high-confidence prediction is factually accurate, whereas it only reflects the model's internal consistency given the training data. Furthermore, variational inference adds complexity to the training pipeline, potentially increasing the barrier to entry for deploying these safer models. Additionally, both the original paper and our experiments rely on standard English-language benchmarks, which may encode societal and cultural biases that are not addressed by uncertainty calibration alone [1].

10 Limitations

The original BLoB paper has several notable limitations.

- **Restricted experimental scope.** BLoB is evaluated on encoder-only architectures and classification tasks. Generative encoder–decoder models are not considered, limiting insight into how Bayesian low-rank uncertainty manifests in realistic text generation settings, where uncertainty also appears as variability in generated outputs.
- **Limited uncertainty evaluation.** Although BLoB is motivated as an uncertainty-aware method, empirical evaluation focuses mainly on predictive performance (accuracy and NLL), with limited analysis of uncertainty decomposition or downstream decision-making under uncertainty.
- **High computational overhead.** Compared to standard LoRA, BLoB introduces substantial additional cost due to posterior sampling and KL computation during both training and inference. This overhead becomes particularly pronounced in generative tasks, where multiple decoding passes are required to approximate the posterior predictive distribution, limiting scalability.
- **Sensitivity to Bayesian hyperparameters.** BLoB relies on several interacting hyperparameters, including the KL weighting coefficients β and γ. Our experiments show that small values of γ can lead to posterior collapse, effectively removing Bayesian behaviour. While these parameters are introduced in the paper, guidance on stable operating regimes and failure modes is limited.
- **Heuristic KL reweighting.** The KL reweighting schedule used during training is adopted as a practical heuristic without theoretical analysis of its convergence properties or robustness, which complicates principled hyperparameter selection.

11 Self Assessment

We believe that our project satisfies the criteria for an **Excellent (A-level)** grade.

At the E–D level, we successfully reproduced the core components of the BLoB framework using the authors' code, carefully addressing reproducibility challenges and maintaining consistency with the original experimental setup. For the D–C–B level, we extended the evaluation by examining additional sources of variation and design choices that were not explored in the original paper.

At the B–A level, we went beyond the scope of the original work by extending BLoB to a novel and more challenging setting: encoder–decoder generative summarization. Within this new context, we conducted a systematic hyperparameter sensitivity analysis, identifying posterior collapse and strong dependence on the KL scaling parameter phenomena not discussed in the original classification-focused study. We further performed large-scale Monte Carlo inference experiments (100 dialogues $\times$ 100 posterior samples), enabling an uncertainty analysis based on output metric score variability rather than classification confidence.

Overall, our project combines faithful reproduction with substantial experimental extensions and critical analysis, and we therefore believe it merits an **A-level** assessment.

References

[1] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, 2021.

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

[3] Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online, August 2021. Association for Computational Linguistics.

[4] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

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

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[8] Ziwei Ji, Nayeon Lee, Rita Frieske, Tianyi Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2023.

[9] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[10] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics, 2004.

[11] David Patterson, Joseph Gonzalez, Urs Hölzle, and Quoc Le. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

[12] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[13] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, 2019.

[14] Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

[15] Yibin Wang, Haizhou Shi, Ligong Han, Dimitris Metaxas, and Hao Wang. Blob: Bayesian low-rank adaptation by backpropagation for large language models, 2025.

[16] Adam X Yang, Maxime Robey, Ju Wang, Hamed Hassani, George Sexton, and Edgar Dobriban. Bayesian low-rank adaptation for large language models. *arXiv preprint arXiv:2308.13111*, 2023.

[17] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

A Appendix: Additional Figures

A.1 BERTScore Metric

Let the reference summary be represented as a sequence of tokens

$$y = (y_1, \ldots, y_m), \tag{13}$$

and the generated summary as

$$\hat{y} = (\hat{y}_1, \ldots, \hat{y}_n), \tag{14}$$

where m and n denote the number of tokens in the reference and generated summaries, respectively. Each token is mapped to a contextual embedding using a pretrained encoder $\phi(\cdot)$.

$$\mathrm{sim}(u, v) = \frac{u^\top v}{\|u\| \, \|v\|} \tag{15}$$

$$P_{\text{BERT}}(\hat{y}, y) = \frac{1}{|\hat{y}|} \sum_{i=1}^{|\hat{y}|} \max_{j \in \{1,\ldots,|y|\}} \mathrm{sim}(\phi(\hat{y}_i), \phi(y_j)) \tag{16}$$

$$R_{\text{BERT}}(\hat{y}, y) = \frac{1}{|y|} \sum_{j=1}^{|y|} \max_{i \in \{1,\ldots,|\hat{y}|\}} \mathrm{sim}(\phi(y_j), \phi(\hat{y}_i)) \tag{17}$$

$$F1_{\text{BERT}}(\hat{y}, y) = \frac{2 \cdot P_{\text{BERT}}(\hat{y}, y) \cdot R_{\text{BERT}}(\hat{y}, y)}{P_{\text{BERT}}(\hat{y}, y) + R_{\text{BERT}}(\hat{y}, y)} \tag{18}$$

A.2 ROUGE Metrics

$$P_n = \frac{\sum_{g \in \mathcal{G}_n(\hat{y})} \min\big(\mathrm{count}(g, \hat{y}),\, \mathrm{count}(g, y)\big)}{\sum_{g \in \mathcal{G}_n(\hat{y})} \mathrm{count}(g, \hat{y})}, \qquad R_n = \frac{\sum_{g \in \mathcal{G}_n(\hat{y})} \min\big(\mathrm{count}(g, \hat{y}),\, \mathrm{count}(g, y)\big)}{\sum_{g \in \mathcal{G}_n(y)} \mathrm{count}(g, y)}, \tag{19}$$

where $\mathcal{G}_n(\cdot)$ denotes the set of n-grams and y and $\hat{y}$ are the reference and generated summaries. The ROUGE-n score is then computed as the harmonic mean:

$$\text{ROUGE-}n = \frac{2 P_n R_n}{P_n + R_n}. \tag{20}$$

ROUGE-L is based on the length of the longest common subsequence between $\hat{y}$ and y. Let $\mathrm{LCS}(\hat{y}, y)$ denote the LCS length. Precision and recall are defined as:

$$P_L = \frac{\mathrm{LCS}(\hat{y}, y)}{|\hat{y}|}, \qquad R_L = \frac{\mathrm{LCS}(\hat{y}, y)}{|y|}, \tag{21}$$

and the ROUGE-L score is given by:

$$\text{ROUGE-}L = \frac{2 P_L R_L}{P_L + R_L}. \tag{22}$$

A.3 Experiment II: Hyperparameter Sensitivity (DIALOGSUM)

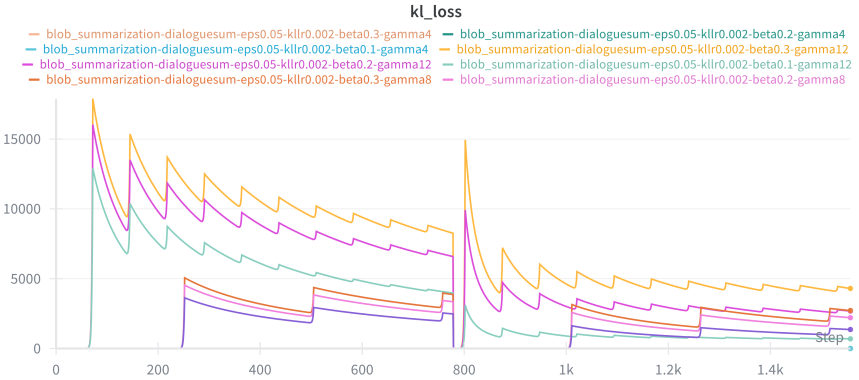


Figure 3: Kullback–Leibler (KL) loss trajectories for the 9 (β, γ) configurations in Experiment II.

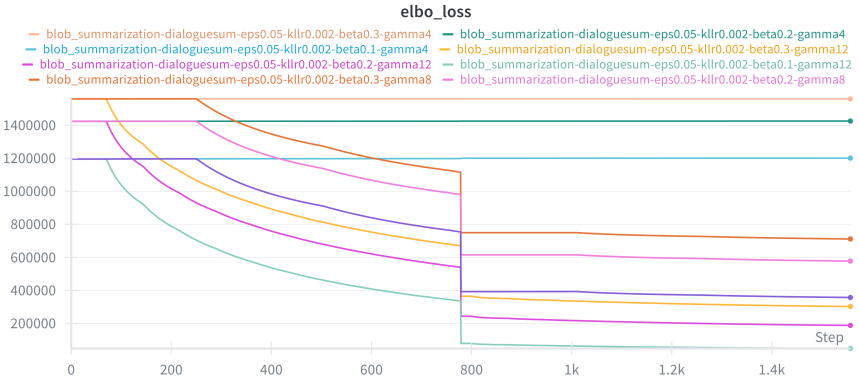
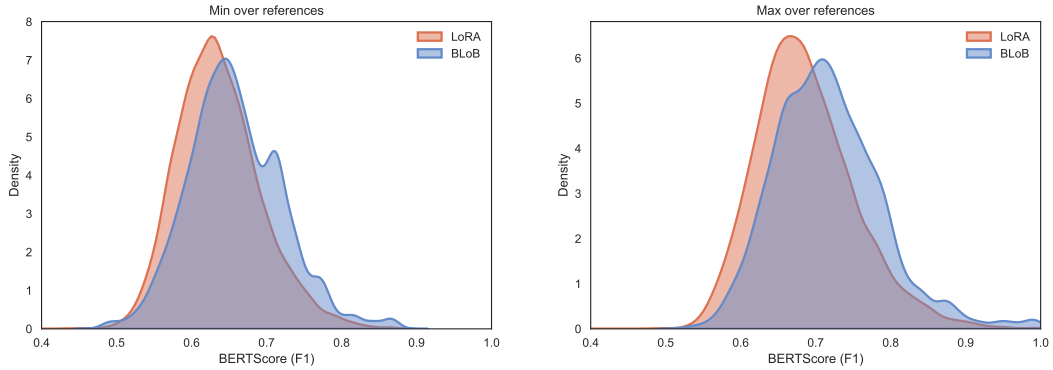


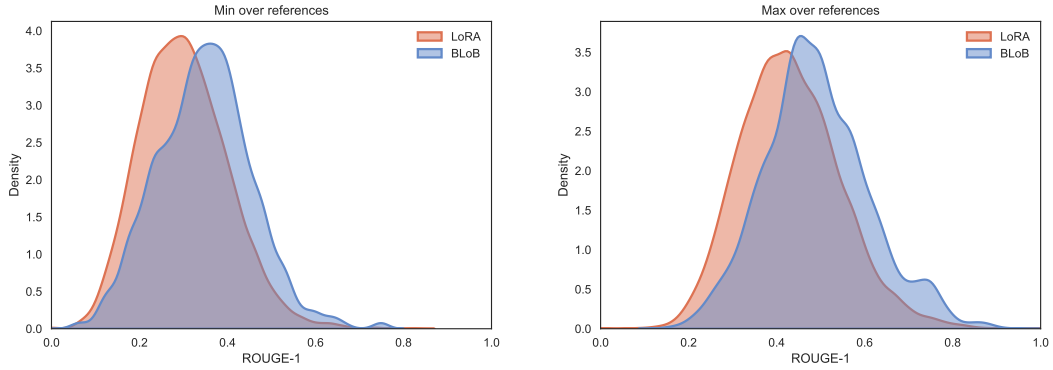
Figure 4: ELBO trajectories for the 9 (β, γ) configurations in Experiment II.

A.4 Experiment IV: Generative Tasks (Summarization) — Bayesian Sampling



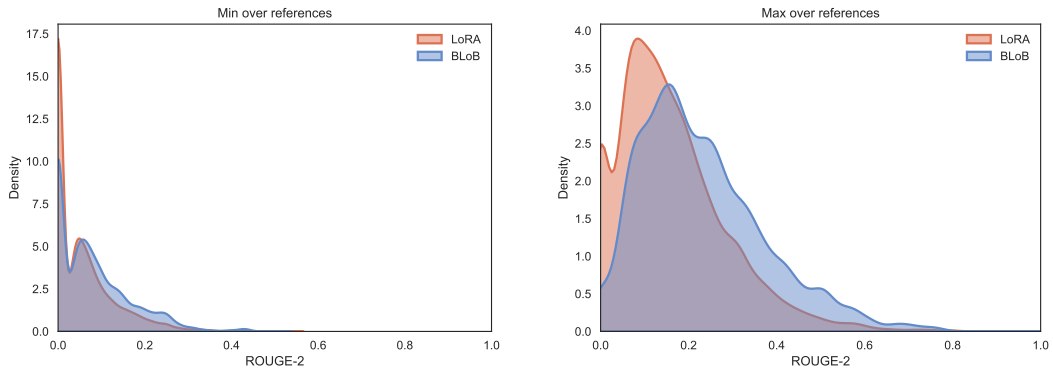
(a) BERTScore (min over references) (b) BERTScore (max over references)

Figure 5: Experiment IV: BERTScore (F1) distributions under Bayesian sampling, aggregated by min/max over references.



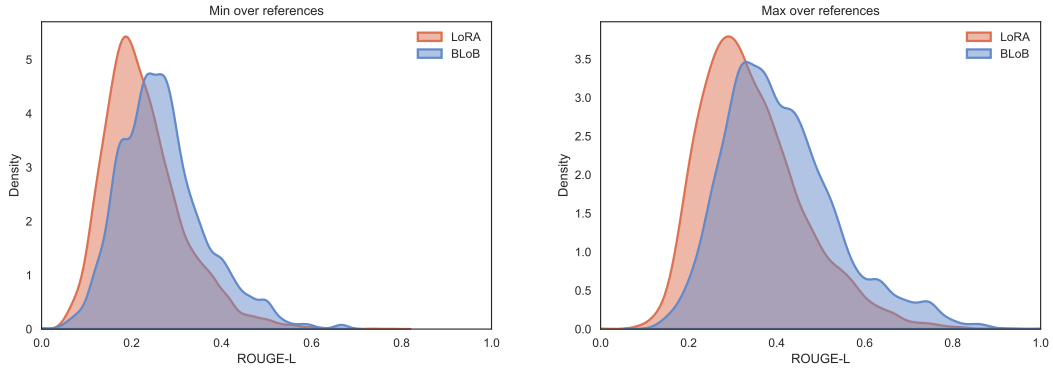
(a) ROUGE-1 (min over references) (b) ROUGE-1 (max over references)

Figure 6: Experiment IV: ROUGE-1 score distributions under Bayesian sampling, aggregated by min/max over references.



(a) ROUGE-2 (min over references) (b) ROUGE-2 (max over references)

Figure 7: Experiment IV: ROUGE-2 score distributions under Bayesian sampling, aggregated by min/max over references.



(a) ROUGE-L (min over references) (b) ROUGE-L (max over references)

Figure 8: Experiment IV: ROUGE-L score distributions under Bayesian sampling, aggregated by min/max over references.