A LAPLACE PREDICTIVE POSTERIOR APPROXIMATIONS

In this section, we review several methods for obtaining the predictive posterior of Laplace approximation.

A.1 MONTE CARLO SAMPLING

As descibed in main text, we can obtain a closed-form Gaussian posterior on output logits,

$$f_{\boldsymbol{\theta}}(\mathbf{x}_*) \sim \mathcal{N}\left(f_{\boldsymbol{\theta}_{\text{MAP}}}(\mathbf{x}_*), \boldsymbol{\Lambda}\right), \tag{15}$$

where

$$\boldsymbol{\Lambda} = (\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|^T_{\boldsymbol{\theta}_{\text{MAP}}})\boldsymbol{\Sigma}(\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|_{\boldsymbol{\theta}_{\text{MAP}}}). \tag{16}$$

To obtain samples of $f_{\boldsymbol{\theta}}(\mathbf{x}_*)$, we can decompose the covariance using the Cholesky factorization $\boldsymbol{\Lambda} = \mathbf{L}\mathbf{L}^T$ with

$$\tilde{f}_{\boldsymbol{\theta}}(\mathbf{x}_*) = f_{\boldsymbol{\theta}_{\text{MAP}}}(\mathbf{x}_*) + \mathbf{L}\boldsymbol{\xi}, \tag{17}$$

where $\boldsymbol{\xi}$ is a vector of IID standard normal random variables. We can compute the Bayesian model average by computing the average probabilities (passing the sampled logits through softmax function) under the Gaussian random noise from $\boldsymbol{\xi}$.

A.2 PROBIT APPROXIMATION

A closed-form approximation of the predictive posterior for classification can be obtained by integrating out the posterior over weights with a generalized probit approximation (Lu et al., 2020; Daxberger et al., 2021a) of the likelihood,

$$p(y_*|\mathbf{x}_*, \mathcal{D}) \approx \text{Categorical}\left(y_*, \text{softmax}\left(\frac{f_{\boldsymbol{\theta}_{\text{MAP}}}(\mathbf{x}_*)}{\sqrt{1 + \frac{\pi}{8}\operatorname{diag}(\boldsymbol{\Lambda})}}\right)\right), \tag{18}$$

where

$$\boldsymbol{\Lambda} = (\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|^T_{\boldsymbol{\theta}_{\text{MAP}}})\boldsymbol{\Sigma}(\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|_{\boldsymbol{\theta}_{\text{MAP}}}). \tag{19}$$

Although probit approximation provably preserves decision boundary in binary sigmoid classification (Kristiadi et al., 2020), it does hold for softmax multiclass classification.

A.3 LAPLACE BRIDGE

The Laplace bridge (MacKay, 1998; Daxberger et al., 2021a) maps a Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to a Dirichlet distribution $\mathcal{D}(\boldsymbol{\alpha})$ over classes with

$$\boldsymbol{\alpha}_i = \frac{1}{\boldsymbol{\Sigma}_{ii}}\left(1 - \frac{2}{C} + \frac{\exp(\boldsymbol{\mu}_i)}{C^2}\sum_j \exp(-\boldsymbol{\mu}_j)\right), \tag{20}$$

where C denotes the number of classes. Similar to the generalized probit approximation, it also ignores the covariance terms and only considers the diagonal of the covariance $\boldsymbol{\Sigma}_{ii}$.

A.4 EMPIRICAL COMPARISON

Figure 3 shows a comparison of these approximations in fine-tuning Llama2-7B and applying full Laplace-LoRA post-hoc. Specifically, we considered Monte Carlo sampling using the full covariance (MC joint), Monte Carlo sampling only using the diagonal covariance (MC indep), generalized probit approximation (probit), and Laplace bridge (bridge). MC joint consistently achieves highest accuracy and among the best NLL, while bridge is often the worst, probit and MC indep can sometimes give suboptimal performance. This is likely due to bridge, probit and MC indep are all approximations that ignored the covariances between logits, whereas MC joint faithfully approximates the true predictive posterior.

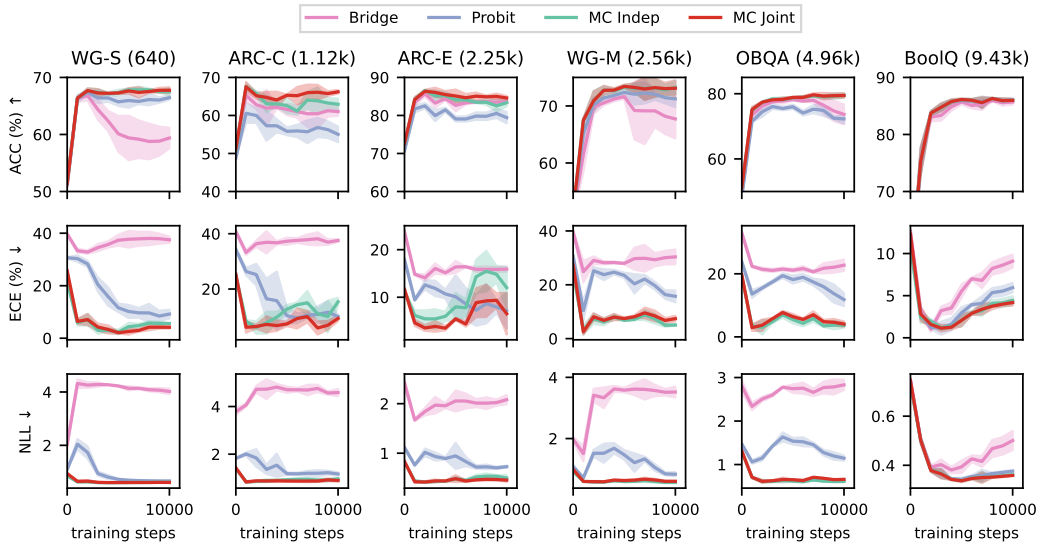

Figure 3: Fine-tuning of LlaMA2-7B across six common sense reasoning tasks, comparing different Laplace predictive posterior approximations: Laplace bridge approximation (bridge), generalized probit approximation (probit), Monte Carlo sampling using the diagonal covariance (MC indep), and Monte Carlo sampling using the full covariance (MC joint).

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

Table 5: Prompt templates for fine-tuning LlaMA-7B on common sense reasoning tasks.

B Prompt templates for LLaMA2 common sense reasoning tasks

We present our prompt templates used to fine-tune LlaMA2 on common sense reasoning tasks in Table 5. For ARC datasets, although the majority of quesstions have four choices, there are a tiny amount of questions with three or five choices which we remove for consistency. In ARC-C, there are 1/1119 with three choices and 1/1119 with five choices in the training set; 3/299 with three choices and 1/299 with five choices in the evaluation set. In ARC-E, there are 6/2251 with three choices and 4/2251 with five choices in the trainng set; 1/570 with three choices and 2/570 with five choices in the evaluation set.

C Hyperparameters

We follow the default hyperparameters from Huggingface's Transformer (Wolf et al., 2020) and PEFT (Mangrulkar et al., 2022) libraries. Hyperparameters used for fine-tuning RoBERTa-base and RoBERTa-large with LoRA are shown in Table 6, those for fine-tuning LlaMA-7B are shown in Table 7. Note the only differences between the two settings are a smaller batch size to fit into our GPU memory, and a longer max sequence length to account for prompt length.

Hyperparameter	Value
LoRA r	8
LoRA α	16
Dropout Probability	0.1
Weight Decay	0
Learning Rate	5×10^{-5}
Learning Rate Scheduler	Linear
Batch Size	32
Max Sequence Length	256

Table 6: Hyperparameters used in fine-tuning RoBERTa-base and RoBERTa-large with LoRA.

Hyperparameter	Value
LoRA r	8
LoRA α	16
Dropout Probability	0.1
Weight Decay	0
Learning Rate	5×10^{-5}
Learning Rate Scheduler	Linear
Batch Size	4
Max Sequence Length	300

Table 7: Hyperparameters used in fine-tuning LlaMA-7B with LoRA.

D METRICS FOR UNCERTAINTY QUANTIFICATION

There are two commonly used metrics for measuring uncertainty quantification in neural networks: negative log-likelihood (NLL) and expected calibration error (ECE). NLL computes the sum of negative expected log probability of predicting the true label,

$$\text{NLL} = \sum_{i=1}^{N} -\log P(\hat{y}_i = y_i), \tag{21}$$

where $P(\hat{y}_i)$ is the model's output distribution, y_i is the true label. NLL is also equivalent to cross-entropy between the one-hot data distribution and the model's output distribution. NLL encourages the model to give high probability to correct answers. If the model is overconfident in a wrong answer, then the probability of giving the right answer would be low which raises NLL. On the other hand, ECE measures the alignment between model's confidence and accuracy, by binning the highest predicted probabilities and compute a weighted average between the difference of average accuracy and confidence in each bin,

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)|, \tag{22}$$

where $\text{acc}(B_m)$ and $\text{conf}(B_m)$ are the average accuracy and average confidence in each bin,

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i), \qquad \text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} P(\hat{y}_i), \tag{23}$$

and $|B_m|$ is the number of examples in bin m. However, expected calibration error cannot be optimized directly like negative log-likelihood, as a completely random model will have the same accuracy and confidence for each datapoint, thus achieving zero ECE (Ashukha et al., 2020).

E OPTIMIZING LAPLACE PRIOR PRECISION

In this section, we present how we optimize the Laplace prior precision. When there is only a training set available with no validation set (such as in Figure 1, 4, 5 and Table 1, 8, 9), we can use the

Algorithm 1 Optimize Laplace prior precision using the training set model evidence

Initialize prior precision λ, learning rate η, optimization steps M
Obtain θ_{MAP} from a fine-tuned checkpoint, pre-compute $\log \mathrm{P}\left(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}\right)$ and Fisher $\mathbf{F}$
for $step = 1, ..., M$ **do**
  Compute posterior covariance $\boldsymbol{\Sigma} = \mathbf{F} + \lambda \mathbf{I}$
  Calculate $\mathcal{L}(\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}) = \log \mathrm{P}\left(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}\right) + \log \mathrm{P}\left(\boldsymbol{\theta}\right) = \log \mathrm{P}\left(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}\right) + \lambda ||\boldsymbol{\theta}_{\text{MAP}}||_2^2$
  Perform a gradient step with respect to λ to maximize log model evidence (Eq.13):
    $\lambda \leftarrow \lambda + \eta \nabla_\lambda \big(- \mathcal{L}(\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}_{\text{MAP}}) + \frac{1}{2} \log |\boldsymbol{\Sigma}| \big)$
end for

Algorithm 2 Optimize Laplace prior precision using validation log-likelihood

Initialize prior precision λ, learning rate η, batch size b, validation set $(\mathbf{X}, \mathbf{y})$
Obtain $\boldsymbol{\theta}_{\text{MAP}}$ from a fine-tuned checkpoint, pre-compute mean $f_{\boldsymbol{\theta}_{\text{MAP}}}$, Jacobian $\mathbf{J} = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{X})|_{\boldsymbol{\theta}_{\text{MAP}}}$, Fisher $\mathbf{F}$
for $step = 1, ..., M$ **do**
  Randomly sample a batch of validation data $\mathbf{X}_b, \mathbf{y}_b$ with corresponding Jacobian $\mathbf{J}_b$
  Compute posterior covariance $\boldsymbol{\Sigma} = \mathbf{F} + \lambda \mathbf{I}$
  Calculate batch logits covariance $\boldsymbol{\Lambda}_b = \mathbf{J}_b^T \boldsymbol{\Sigma} \mathbf{J}_b$ and Cholesky $\boldsymbol{\Lambda}_b = \mathbf{L}_b \mathbf{L}_b^T$
  Obtain batch Bayesian model average $\tilde{f}_{\boldsymbol{\theta}}(\mathbf{X}_b) = f_{\boldsymbol{\theta}_{\text{MAP}}}(\mathbf{X}_b) + \mathbf{L}_b \boldsymbol{\xi}$
  Evaluate validation likelihood $\mathrm{P}\left(\mathbf{y}_b|\mathbf{X}_b, \boldsymbol{\theta}\right) = \text{Categorical}\big(\mathbf{y}_b; \text{softmax}(\tilde{f}_{\boldsymbol{\theta}}(\mathbf{X}_b))\big)$
  Perform a gradient step with respect to λ to maximize mini-batch validation log-likelihood:
    $\lambda \leftarrow \lambda + \eta \nabla_\lambda \log \mathrm{P}\left(\mathbf{y}_b|\mathbf{X}_b, \boldsymbol{\theta}\right)$
end for

Laplace model evidence in Equation 13 to optimize prior precision. Our algorithm is presented in Algorithm 1, and we chose $\eta = 0.1$, and $M = 100$.

When we introduce a validation set by splitting the training set (such as in Figure 8, 6, 7 and Table 2, 10, 11), we can use the validation log-likelihood to optimize the Laplace prior precision. For memory and computational efficiency, we precompute the mean $f_{\boldsymbol{\theta}_{\text{MAP}}}$ and the Jacobian $\mathbf{J} = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{X})|_{\boldsymbol{\theta}_{\text{MAP}}}$, then perform mini-batch gradient descent on λ (reparametrization in Bayesian model average allows gradient flowing through) as detailed in Algorithm 2. We chose $\eta = 0.1$, $M = 1000$, and $b = 4$.

F Additional Experiments

F.1 Fine-tuning RoBERTa for text classification

In this section, we present additional results of fine-tuning RoBERTa-base (Fig. 4) and RoBERTa-large (Fig. 5) (Liu et al., 2019) with LoRA on text classification tasks from GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a). Results for RoBERTa-base are shown in Figure 4 and Table 8, and results for RoBERTa-large are shown in Figure 5 and Table 9. Surprisingly, checkpoint ensemble and Monte-Carlo (MC) dropout exhibit distinct behavior on RoBERTa models compared to LlaMA2-7B. Checkpoint ensemble often performs much worse than the Maximum a Posteriori (MAP) estimation in terms of Expected Calibration Error (ECE) and Negative Log-Likelihood (NLL), while MC dropout often offers much more improvement on RoBERTa models. We suspect this difference arises due to an extra hidden penultimate layer with an additional dropout layer in front in the default RoBERTa fine-tuning setup; whereas, in LlaMA2-7B fine-tuning, we have LoRA weights and dropout on LoRA layers at the end. However, the gain from MC dropout diminishes on RoBERTa-large compared to RoBERTa-base. On the other hand, Laplace-LoRA (LA) consistently delivers substantial gains on any models we have tested (RoBERTa-base, RoBERTa-large, and LlaMA2-7B), demonstrating the robustness of Laplace-LoRA. Moreover, the Last-Layer Laplace-LoRA (LLLA) offers modest improvements as in LlaMA2-7B when optimized by Laplace model evidence, underscoring the significance of performing Bayesian inference on all LoRA weights.

Similarly, we also conduct experiments by splitting the training set into a 80% training set and a 20% validation set, then tune temperature and Laplace prior precision on the validation set. The results are

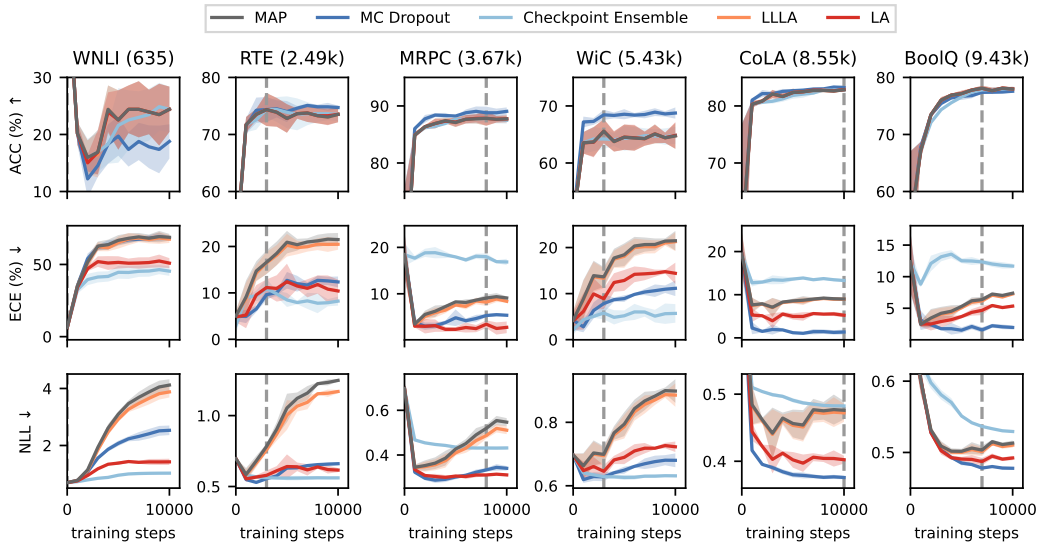

Figure 4: Fine-tuning of RoBERTa-base across six GLUE and SuperGLUE tasks (presented column-wise, with number of training examples in brackets), evaluated on the test set every 1000 gradient steps. The vertical dashed line gives the number of training steps with optimal MAP performance. Note that RoBERTa-base seems to fail on WNLI, but RoBERTa-large succeeds (Fig. 5).

Table 8: Comparison of different post-hoc methods applied to the fine-tuned RoBERTa-base across six GLUE and SuperGLUE tasks. Results are evaluated at the early stopping point of 5000 gradient steps.

Methods	Metrics	WNLI	RTE	MRPC	WiC	CoLA	BoolQ
ACC ↑	MAP	$22.5_{4.6}$	$72.8_{2.2}$	$87.1_{0.7}$	$64.5_{2.0}$	$82.4_{0.6}$	$77.2_{0.4}$
	MC Drop	$19.7_{3.0}$	$74.0_{1.6}$	$88.2_{0.0}$	$68.5_{0.8}$	$82.7_{0.1}$	$76.7_{0.6}$
	Ckpt Ens	$21.6_{5.8}$	$73.9_{2.5}$	$87.7_{0.6}$	$64.3_{1.9}$	$81.8_{0.5}$	$76.6_{0.2}$
	LLLA	$22.5_{4.6}$	$72.8_{2.2}$	$87.1_{0.7}$	$64.5_{2.0}$	$82.4_{0.6}$	$77.2_{0.4}$
	LA	$22.5_{4.6}$	$72.8_{2.2}$	$87.2_{0.7}$	$64.5_{2.0}$	$82.4_{0.6}$	$77.2_{0.5}$
ECE ↓	MAP	$66.7_{3.2}$	$20.9_{2.1}$	$8.3_{0.1}$	$18.4_{2.2}$	$8.6_{0.5}$	$5.3_{0.2}$
	MC Drop	$65.6_{1.8}$	$12.7_{1.6}$	$4.8_{1.6}$	$8.9_{0.6}$	$\mathbf{1.2_{0.1}}$	$\mathbf{1.5_{0.4}}$
	Ckpt Ens	$\mathbf{44.6_{3.9}}$	$\mathbf{8.5_{1.3}}$	$17.9_{0.1}$	$\mathbf{5.2_{1.0}}$	$13.6_{0.3}$	$12.5_{0.3}$
	LLLA	$65.2_{3.7}$	$20.3_{2.2}$	$7.6_{0.3}$	$18.1_{2.2}$	$8.5_{0.4}$	$5.1_{0.1}$
	LA	$51.4_{3.8}$	$12.4_{3.4}$	$\mathbf{2.3_{0.4}}$	$12.9_{2.0}$	$5.0_{0.6}$	$3.7_{0.2}$
NLL ↓	MAP	$3.10_{0.09}$	$1.05_{0.09}$	$0.43_{0.01}$	$0.79_{0.03}$	$0.46_{0.02}$	$0.50_{0.00}$
	MC Drop	$2.07_{0.08}$	$0.63_{0.01}$	$\mathbf{0.30_{0.01}}$	$0.65_{0.01}$	$\mathbf{0.38_{0.00}}$	$\mathbf{0.49_{0.00}}$
	Ckpt Ens	$\mathbf{0.96_{0.02}}$	$\mathbf{0.56_{0.01}}$	$0.44_{0.00}$	$\mathbf{0.63_{0.01}}$	$0.49_{0.00}$	$0.55_{0.00}$
	LLLA	$2.92_{0.08}$	$1.00_{0.09}$	$0.41_{0.01}$	$0.79_{0.03}$	$0.45_{0.02}$	$0.50_{0.00}$
	LA	$1.38_{0.06}$	$0.64_{0.08}$	$\mathbf{0.30_{0.00}}$	$0.69_{0.02}$	$0.40_{0.01}$	$\mathbf{0.49_{0.00}}$

shown in Figure 6 and Table 10 for RoBERTa-base, and Figure 7 and Table 11 for RoBERTa-large. Again, full Laplace-LoRA LA offers the most improvements most of the time, and is the most robust method overall.

F.2 Fine-tuning LLaMA2-7B for common sense reasoning

Figure 8 displays the results of tuning temperature scaling and Laplace prior precision on a held-out validation set split from the training set. LLLA is missing a few zero checkpoint evaluation results due to Cholesky errors. Comparing Figure 8 to Figure 1 in the main text, it is evident that splitting the training set into a smaller training set (80% data) and a validation set (20% data) slightly impact the accuracy of fine-tuned model.

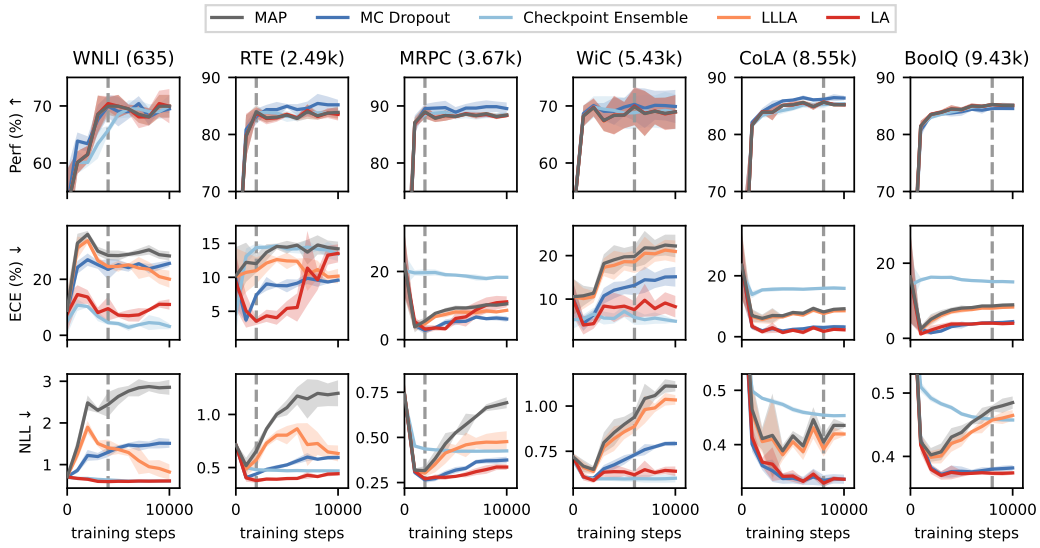

Figure 5: Fine-tuning RoBERTa-large across the six GLUE and SuperGLUE tasks in Fig. 4. The vertical dashed line gives the number of training steps with optimal MAP performance.

Table 9: Comparison of different post-hoc methods applied to the fine-tuned RoBERTa-large across six GLUE and SuperGLUE tasks. Results are evaluated at the early stopping point of 5000 gradient steps.

Methods	Metrics	WNLI	RTE	MRPC	WiC	CoLA	BoolQ
ACC ↑	MAP	$70.0_{1.8}$	$83.4_{0.3}$	$88.2_{0.3}$	$68.4_{2.0}$	$85.2_{0.5}$	$84.4_{0.4}$
	MC Drop	$69.0_{2.3}$	$85.0_{0.3}$	$88.9_{0.8}$	$69.8_{1.4}$	$86.0_{0.1}$	$84.0_{0.6}$
	Ckpt Ens	$68.5_{0.7}$	$83.2_{0.9}$	$88.2_{0.5}$	$68.8_{2.4}$	$84.6_{0.4}$	$84.3_{0.5}$
	LLLA	$70.0_{1.8}$	$83.5_{0.3}$	$88.2_{0.3}$	$68.4_{2.0}$	$85.2_{0.5}$	$84.3_{0.3}$
	LA	$70.0_{1.8}$	$83.5_{0.3}$	$88.2_{0.3}$	$68.4_{2.0}$	$85.2_{0.5}$	$84.4_{0.4}$
ECE ↓	MAP	$28.5_{1.6}$	$14.4_{0.8}$	$9.3_{0.6}$	$19.7_{1.8}$	$8.0_{0.4}$	$8.0_{0.6}$
	MC Drop	$25.2_{1.6}$	$8.8_{0.9}$	$\mathbf{5.3_{1.0}}$	$12.6_{1.3}$	$2.4_{0.3}$	$\mathbf{3.6_{0.9}}$
	Ckpt Ens	$\mathbf{4.2_{0.7}}$	$14.2_{0.5}$	$18.9_{0.2}$	$\mathbf{7.2_{1.2}}$	$15.7_{0.5}$	$15.6_{0.3}$
	LLLA	$24.6_{1.7}$	$12.4_{1.1}$	$8.1_{0.6}$	$18.6_{1.8}$	$7.6_{0.3}$	$7.5_{0.7}$
	LA	$7.2_{0.7}$	$\mathbf{5.4_{1.0}}$	$6.2_{1.5}$	$8.4_{2.1}$	$\mathbf{2.1_{0.3}}$	$3.8_{0.5}$
NLL ↓	MAP	$2.64_{0.18}$	$1.06_{0.06}$	$0.53_{0.01}$	$0.90_{0.04}$	$0.42_{0.01}$	$0.44_{0.01}$
	MC Drop	$1.40_{0.06}$	$0.53_{0.01}$	$0.32_{0.02}$	$0.70_{0.02}$	$\mathbf{0.34_{0.01}}$	$0.38_{0.01}$
	Ckpt Ens	$0.62_{0.01}$	$0.47_{0.00}$	$0.42_{0.00}$	$\mathbf{0.60_{0.01}}$	$0.46_{0.00}$	$0.47_{0.00}$
	LLLA	$1.36_{0.10}$	$0.80_{0.08}$	$0.44_{0.02}$	$0.86_{0.04}$	$0.41_{0.01}$	$0.43_{0.01}$
	LA	$\mathbf{0.60_{0.02}}$	$\mathbf{0.40_{0.01}}$	$\mathbf{0.28_{0.00}}$	$0.64_{0.02}$	$\mathbf{0.34_{0.01}}$	$\mathbf{0.37_{0.00}}$

F.3 Diagonal Laplace approximation

F.3.1 RoBERTa

In this section, we present the results for Laplace-LoRA utilizing a diagonal approximation of the Hessian. This approach is generally not found to be as effective as the Kronecker-factored Approximate Curvature (K-FAC) (Daxberger et al., 2021a) that approximates the block diagonal Hessian. The results on RoBERTa-base and RoBERTa-large are shown in Figure 9 Table 12, and Figure 10 Table 13. LLLA still offers a tiny advantage over Maximum a Posteriori (MAP) in ECE and NLL, however, LA show mixed performance across the datasets.

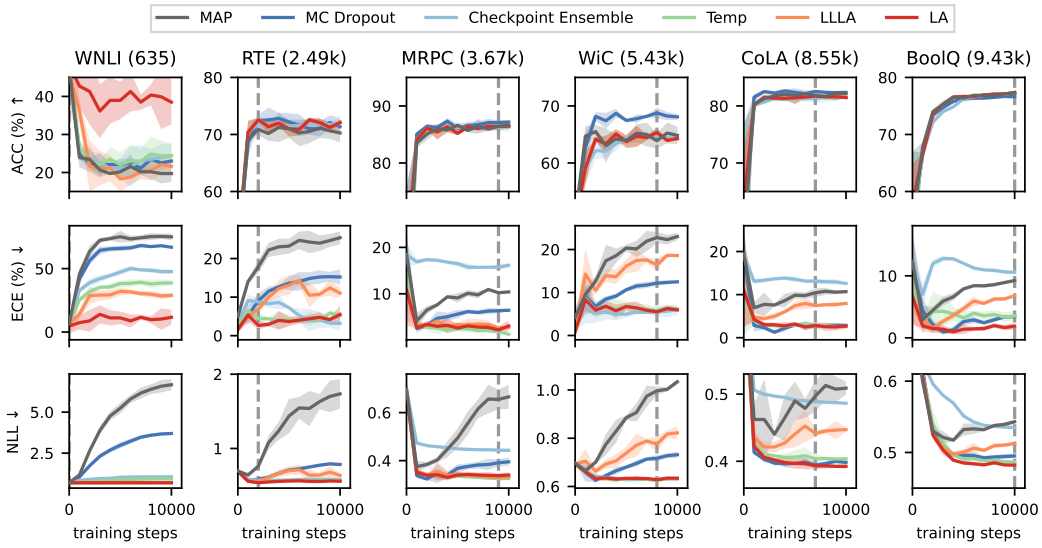

Figure 6: Fine-tuning RoBERTa-base across the six GLUE and SuperGLUE tasks. The vertical dashed line gives the checkpoint with optimal MAP performance on a held-out validation set.

Table 10: Comparison of different post-hoc methods applied to the fine-tuned RoBERTa-base across six GLUE and SuperGLUE tasks, with a validation set split from the training set used for tuning temperature and Laplace prior precision. Results are evaluated at the best MAP performance checkpoint observed on the validation set.

Methods	Metrics	WNLI	RTE	MRPC	WiC	CoLA	BoolQ
ACC ↑	MAP	$47.9_{6.0}$	$70.9_{0.9}$	$86.4_{0.6}$	$63.9_{1.2}$	$81.8_{0.3}$	$77.2_{0.2}$
	MC Drop	$47.9_{6.0}$	$72.4_{0.9}$	$87.1_{0.3}$	$68.8_{0.9}$	$82.6_{0.1}$	$76.6_{0.4}$
	Ckpt Ens	$47.9_{6.0}$	$71.8_{0.9}$	$86.3_{0.5}$	$64.7_{0.3}$	$81.4_{0.7}$	$77.2_{0.1}$
	Temp	$47.9_{6.0}$	$72.6_{0.3}$	$86.5_{0.3}$	$65.4_{0.8}$	$81.8_{0.8}$	$77.3_{0.2}$
	LLLA	$46.0_{10.0}$	$72.6_{0.3}$	$86.4_{0.4}$	$65.3_{0.7}$	$81.8_{0.7}$	$77.4_{0.2}$
	LA	$48.4_{7.7}$	$72.6_{0.3}$	$86.4_{0.3}$	$65.4_{0.6}$	$81.7_{0.7}$	$77.4_{0.2}$
ECE ↓	MAP	$6.3_{4.3}$	$17.8_{1.4}$	$10.2_{0.2}$	$22.8_{1.1}$	$10.4_{0.9}$	$9.2_{0.1}$
	MC Drop	$6.4_{4.4}$	$9.0_{1.7}$	$6.3_{0.4}$	$12.2_{0.5}$	$\mathbf{2.8_{0.3}}$	$3.4_{0.1}$
	Ckpt Ens	$6.2_{2.3}$	$8.8_{0.9}$	$15.7_{0.6}$	$5.5_{1.0}$	$13.1_{0.5}$	$10.6_{0.1}$
	Temp	$5.5_{1.7}$	$4.5_{1.3}$	$\mathbf{1.9_{0.3}}$	$5.8_{0.9}$	$2.9_{0.6}$	$3.5_{0.7}$
	LLLA	$6.8_{7.8}$	$6.9_{0.7}$	$2.6_{0.4}$	$16.4_{1.4}$	$7.4_{0.3}$	$6.8_{0.2}$
	LA	$\mathbf{4.7_{5.2}}$	$\mathbf{2.7_{1.0}}$	$2.2_{0.7}$	$\mathbf{5.5_{0.5}}$	$\mathbf{2.8_{0.6}}$	$\mathbf{1.9_{0.5}}$
NLL ↓	MAP	$0.70_{0.01}$	$0.76_{0.03}$	$0.66_{0.04}$	$1.00_{0.02}$	$0.50_{0.02}$	$0.54_{0.00}$
	MC Drop	$0.70_{0.01}$	$0.58_{0.03}$	$0.39_{0.01}$	$0.72_{0.01}$	$0.39_{0.00}$	$0.50_{0.00}$
	Ckpt Ens	$\mathbf{0.69_{0.00}}$	$0.57_{0.00}$	$0.44_{0.00}$	$0.63_{0.00}$	$0.49_{0.00}$	$0.53_{0.00}$
	Temp	$\mathbf{0.69_{0.00}}$	$\mathbf{0.54_{0.00}}$	$\mathbf{0.32_{0.00}}$	$\mathbf{0.62_{0.01}}$	$0.40_{0.01}$	$0.49_{0.00}$
	LLLA	$\mathbf{0.69_{0.00}}$	$0.56_{0.01}$	$0.33_{0.00}$	$0.78_{0.03}$	$0.44_{0.00}$	$0.51_{0.00}$
	LA	$\mathbf{0.69_{0.00}}$	$\mathbf{0.54_{0.00}}$	$0.34_{0.01}$	$\mathbf{0.62_{0.01}}$	$\mathbf{0.39_{0.00}}$	$\mathbf{0.48_{0.00}}$

F.3.2 LLAMA2-7B

On LlaMA2-7B, both LLLA and LA show mixed performance and do not offer consistent improvements as shown in Figure 11 and Table 14. Specifically, the accuracy of diagonal LA is much worse than MAP on ARC datasets, and ECE is worse than MAP on ARC-E, OBQA and BoolQ.

On the other hand, when we split a validation set from the training set and tune the Laplace prior precision using the validation log-likelihood, the results are much better, as shown in Figure 12 and Table 15.

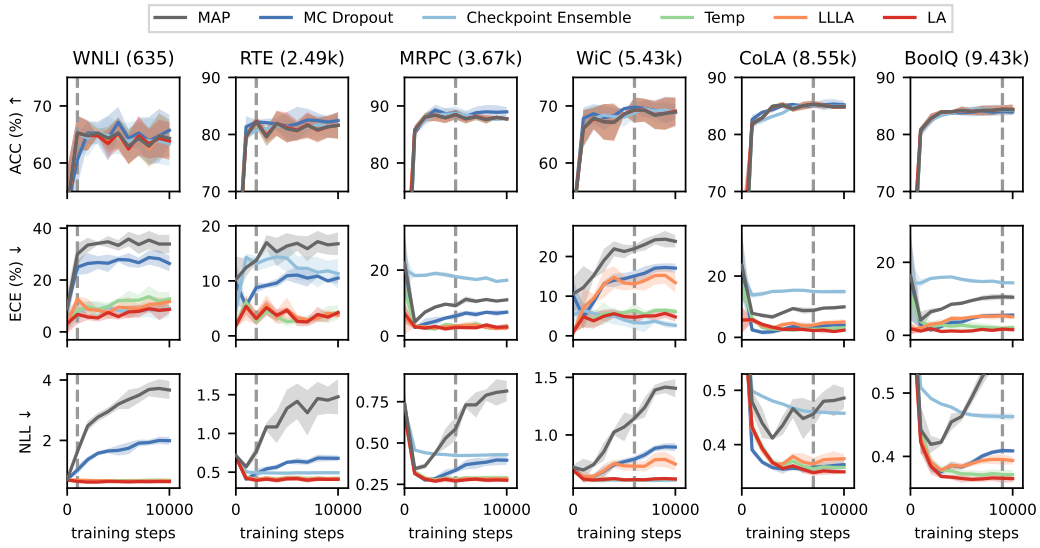

Figure 7: Fine-tuning RoBERTa-large across the six GLUE and SuperGLUE tasks. The vertical dashed line gives the checkpoint with optimal MAP performance on a held-out validation set.

Table 11: Comparison of different post-hoc methods applied to the fine-tuned RoBERTa-large across six GLUE and SuperGLUE tasks, with a validation set split from the training set used for tuning temperature and Laplace prior precision. Results are evaluated at the best MAP performance checkpoint observed on the validation set.

Methods	Metrics	WNLI	RTE	MRPC	WiC	CoLA	BoolQ
ACC ↑	MAP	$65.3_{2.7}$	$82.2_{0.9}$	$88.5_{0.7}$	$69.3_{1.9}$	$85.3_{0.6}$	$84.4_{0.6}$
	MC Drop	$60.6_{4.0}$	$82.2_{0.3}$	$88.7_{0.7}$	$69.8_{0.8}$	$85.1_{0.3}$	$84.0_{0.6}$
	Ckpt Ens	$65.3_{2.7}$	$80.7_{0.3}$	$88.6_{1.1}$	$68.8_{1.0}$	$85.2_{0.4}$	$84.3_{0.8}$
	Temp	$65.3_{2.7}$	$82.2_{0.9}$	$88.5_{0.7}$	$69.3_{1.9}$	$85.3_{0.6}$	$84.4_{0.6}$
	LLLA	$65.3_{2.7}$	$82.2_{0.9}$	$88.5_{0.7}$	$69.3_{1.9}$	$85.3_{0.6}$	$84.4_{0.6}$
	LA	$65.3_{2.7}$	$82.3_{0.8}$	$88.5_{0.7}$	$69.3_{2.0}$	$85.3_{0.5}$	$84.4_{0.6}$
ECE ↓	MAP	$30.0_{3.5}$	$13.7_{0.4}$	$9.2_{0.7}$	$22.0_{0.8}$	$8.8_{0.8}$	$10.5_{0.6}$
	MC Drop	$24.9_{2.6}$	$8.8_{0.8}$	$6.0_{0.9}$	$15.1_{1.1}$	$3.3_{0.8}$	$5.5_{0.4}$
	Ckpt Ens	$7.7_{3.0}$	$13.1_{1.2}$	$18.0_{0.5}$	$\mathbf{3.3_{1.6}}$	$15.0_{0.3}$	$14.4_{0.1}$
	Temp	$8.1_{2.2}$	$\mathbf{2.6_{0.7}}$	$3.1_{0.4}$	$6.4_{1.3}$	$2.7_{0.8}$	$1.9_{0.7}$
	LLLA	$12.7_{0.9}$	$3.3_{1.0}$	$3.0_{0.5}$	$13.1_{1.2}$	$4.0_{0.4}$	$5.3_{0.8}$
	LA	$\mathbf{6.8_{3.3}}$	$3.1_{1.1}$	$\mathbf{2.6_{0.7}}$	$4.7_{1.7}$	$\mathbf{2.2_{0.2}}$	$\mathbf{1.7_{0.4}}$
NLL ↓	MAP	$1.63_{0.09}$	$0.77_{0.07}$	$0.59_{0.06}$	$1.14_{0.02}$	$0.46_{0.02}$	$0.56_{0.01}$
	MC Drop	$1.02_{0.13}$	$0.48_{0.06}$	$0.33_{0.01}$	$0.79_{0.02}$	$0.36_{0.01}$	$0.41_{0.00}$
	Ckpt Ens	$0.65_{0.02}$	$0.49_{0.01}$	$0.42_{0.00}$	$\mathbf{0.61_{0.01}}$	$0.46_{0.00}$	$0.46_{0.00}$
	Temp	$0.65_{0.01}$	$0.41_{0.01}$	$0.28_{0.02}$	$\mathbf{0.61_{0.01}}s$	$0.36_{0.00}$	$0.37_{0.01}$
	LLLA	$0.69_{0.02}$	$0.40_{0.02}$	$\mathbf{0.27_{0.01}}$	$0.73_{0.03}$	$0.36_{0.01}$	$0.40_{0.01}$
	LA	$\mathbf{0.64_{0.00}}$	$\mathbf{0.39_{0.01}}$	$\mathbf{0.27_{0.01}}$	$\mathbf{0.61_{0.01}}$	$\mathbf{0.35_{0.01}}$	$\mathbf{0.37_{0.01}}$

G Out-of-distribution evaluation

Here we present the specific far OOD MMLU datasets we used for evaluations in Table 3 and Table 4. The specific task splits we selected for each subject are shown in Table 16 assigned by Hendrycks et al. (2020).

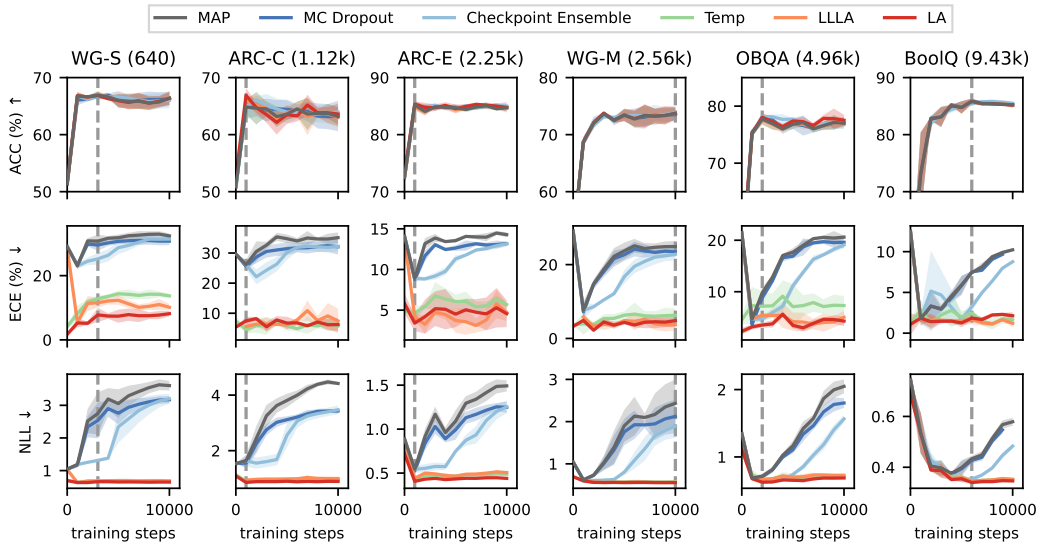

Figure 8: Fine-tuning of LlaMA2-7B across six common sense reasoning tasks (presented column-wise, with number of training examples in brackets), evaluated on the test set every 1000 gradient steps. The vertical dashed line gives the checkpoint with optimal MAP performance on a held-out validation set.

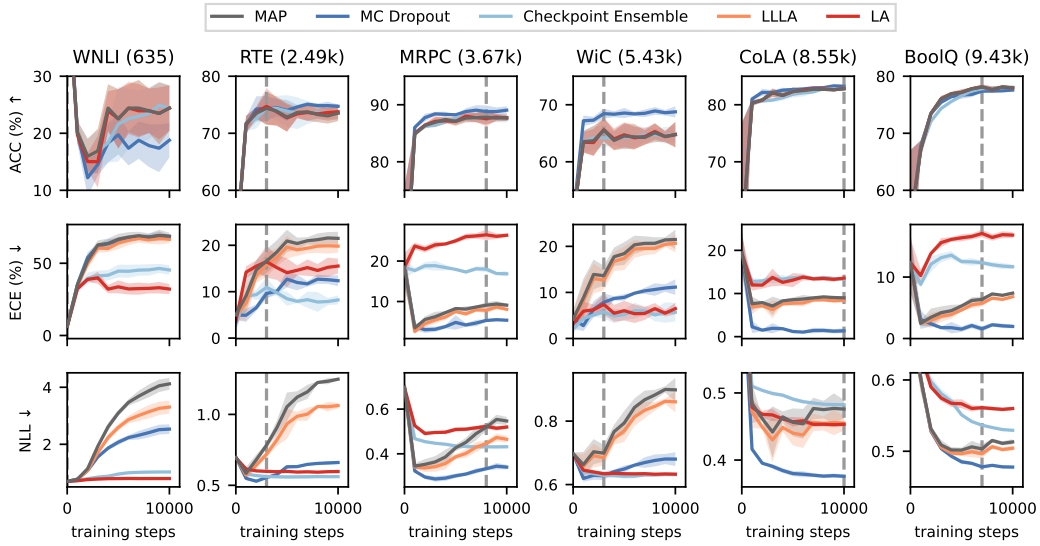

Figure 9: Fine-tuning of RoBERTa-base across six GLUE and SuperGLUE tasks (presented column-wise, with number of training examples in brackets), evaluated on the test set every 1000 gradient steps. The vertical dashed line gives the number of training steps with optimal MAP performance. LA and LLLA using diagonal Fisher approximation.

Table 12: Comparison of different post-hoc methods applied to the fine-tuned RoBERTa-base across six common GLUE and SuperGLUE tasks. Results are evaluated at the early stopping point of 5000 gradient steps. LA and LLLA using diagonal Fisher approximation.

Methods	Metrics	WNLI	RTE	MRPC	WiC	CoLA	BoolQ
ACC ↑	MAP	$22.5_{4.6}$	$72.8_{2.2}$	$87.1_{0.7}$	$64.5_{2.0}$	$82.4_{0.6}$	$77.2_{0.4}$
	MC Drop	$19.7_{3.0}$	$74.0_{1.6}$	$88.2_{0.0}$	$68.5_{0.8}$	$82.7_{0.1}$	$76.7_{0.6}$
	Ckpt Ens	$21.6_{5.8}$	$73.9_{2.5}$	$87.7_{0.6}$	$64.3_{1.9}$	$81.8_{0.5}$	$76.6_{0.2}$
	LLLA	$22.5_{4.6}$	$72.8_{2.2}$	$87.1_{0.7}$	$64.5_{2.0}$	$82.4_{0.6}$	$77.2_{0.4}$
	LA	$22.5_{4.6}$	$72.7_{2.1}$	$87.1_{0.5}$	$64.7_{1.9}$	$82.3_{0.7}$	$77.2_{0.5}$
ECE ↓	MAP	$66.7_{3.2}$	$20.9_{2.1}$	$8.3_{0.1}$	$18.4_{2.2}$	$8.6_{0.5}$	$5.3_{0.2}$
	MC Drop	$65.6_{1.8}$	$12.7_{1.6}$	$\mathbf{4.8_{1.6}}$	$8.9_{0.6}$	$\mathbf{1.2_{0.1}}$	$\mathbf{1.5_{0.4}}$
	Ckpt Ens	$44.6_{3.9}$	$\mathbf{8.5_{1.3}}$	$17.9_{0.1}$	$\mathbf{5.2_{1.0}}$	$13.6_{0.3}$	$12.5_{0.3}$
	LLLA	$64.0_{3.7}$	$19.6_{2.1}$	$7.5_{0.2}$	$17.4_{2.3}$	$7.7_{0.5}$	$4.6_{0.1}$
	LA	$\mathbf{33.7_{4.0}}$	$14.1_{2.8}$	$25.0_{0.5}$	$6.0_{1.9}$	$13.1_{1.4}$	$16.4_{0.5}$
NLL ↓	MAP	$3.10_{0.09}$	$1.05_{0.09}$	$0.43_{0.01}$	$0.79_{0.03}$	$0.46_{0.02}$	$0.50_{0.00}$
	MC Drop	$2.07_{0.08}$	$0.63_{0.01}$	$\mathbf{0.30_{0.01}}$	$0.65_{0.01}$	$\mathbf{0.38_{0.00}}$	$\mathbf{0.49_{0.00}}$
	Ckpt Ens	$0.96_{0.02}$	$\mathbf{0.56_{0.01}}$	$0.44_{0.00}$	$\mathbf{0.63_{0.01}}$	$0.49_{0.00}$	$0.55_{0.00}$
	LLLA	$2.59_{0.10}$	$0.93_{0.09}$	$0.39_{0.01}$	$0.77_{0.03}$	$0.44_{0.02}$	$0.50_{0.00}$
	LA	$\mathbf{0.80_{0.00}}$	$0.60_{0.01}$	$0.51_{0.01}$	$\mathbf{0.63_{0.00}}$	$0.46_{0.01}$	$0.57_{0.01}$

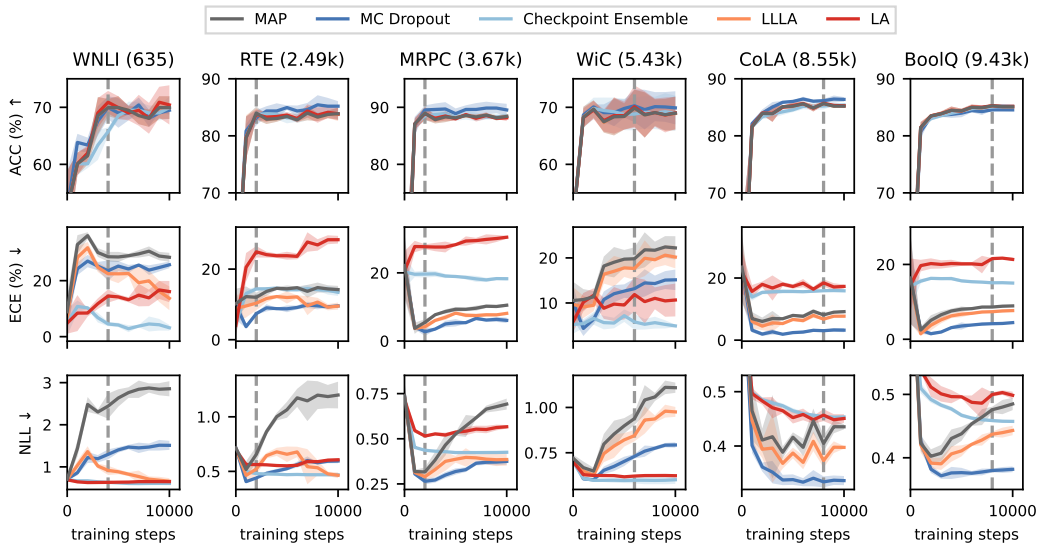

Figure 10: Fine-tuning of RoBERTa-large across six GLUE and SuperGLUE tasks (presented column-wise, with number of training examples in brackets), evaluated on the test set every 1000 gradient steps. The vertical dashed line gives the number of training steps with optimal MAP performance. LA and LLLA using diagonal Fisher approximation.

Table 13: Comparison of different post-hoc methods applied to the fine-tuned RoBERTa-large across six common GLUE and SuperGLUE tasks. Results are evaluated at the early stopping point of 5000 gradient steps. LA and LLLA using diagonal Fisher approximation.

Methods	Metrics	WNLI	RTE	MRPC	WiC	CoLA	BoolQ
ACC ↑	MAP	$70.0_{1.8}$	$83.4_{0.3}$	$88.2_{0.3}$	$68.4_{2.0}$	$85.2_{0.5}$	$84.4_{0.4}$
	MC Drop	$69.0_{2.3}$	$85.0_{0.3}$	$88.9_{0.8}$	$69.8_{1.4}$	$86.0_{0.1}$	$84.0_{0.6}$
	Ckpt Ens	$68.5_{0.7}$	$83.2_{0.9}$	$88.2_{0.5}$	$68.8_{2.4}$	$84.6_{0.4}$	$84.3_{0.5}$
	LLLA	$70.0_{1.8}$	$83.5_{0.3}$	$88.2_{0.3}$	$68.4_{2.0}$	$85.2_{0.5}$	$84.4_{0.4}$
	LA	$70.0_{1.8}$	$83.6_{0.5}$	$88.0_{0.3}$	$68.5_{2.2}$	$85.3_{0.6}$	$84.4_{0.4}$
ECE ↓	MAP	$28.5_{1.6}$	$14.4_{0.8}$	$9.3_{0.6}$	$19.7_{1.8}$	$8.0_{0.4}$	$8.0_{0.6}$
	MC Drop	$25.2_{1.6}$	$\mathbf{8.8_{0.9}}$	$\mathbf{5.3_{1.0}}$	$12.6_{1.3}$	$\mathbf{2.4_{0.3}}$	$\mathbf{3.6_{0.9}}$
	Ckpt Ens	$\mathbf{4.2_{0.7}}$	$14.2_{0.5}$	$18.9_{0.2}$	$\mathbf{7.2_{1.2}}$	$15.7_{0.5}$	$15.6_{0.3}$
	LLLA	$22.4_{3.0}$	$12.0_{1.5}$	$8.2_{0.3}$	$17.8_{1.8}$	$6.7_{0.4}$	$6.7_{0.7}$
	LA	$14.2_{0.8}$	$23.8_{1.0}$	$28.3_{0.2}$	$9.5_{2.8}$	$17.3_{0.6}$	$20.2_{0.5}$
NLL ↓	MAP	$2.64_{0.18}$	$1.06_{0.06}$	$0.53_{0.01}$	$0.90_{0.04}$	$0.42_{0.01}$	$0.44_{0.01}$
	MC Drop	$1.40_{0.06}$	$0.53_{0.01}$	$\mathbf{0.32_{0.02}}$	$0.70_{0.02}$	$\mathbf{0.34_{0.01}}$	$\mathbf{0.38_{0.01}}$
	Ckpt Ens	$\mathbf{0.62_{0.01}}$	$\mathbf{0.47_{0.00}}$	$0.42_{0.00}$	$\mathbf{0.60_{0.01}}$	$0.46_{0.00}$	$0.47_{0.00}$
	LLLA	$0.88_{0.05}$	$0.66_{0.06}$	$0.39_{0.02}$	$0.82_{0.03}$	$0.39_{0.01}$	$0.41_{0.01}$
	LA	$0.64_{0.01}$	$0.55_{0.01}$	$0.54_{0.01}$	$0.62_{0.00}$	$0.45_{0.01}$	$0.50_{0.00}$

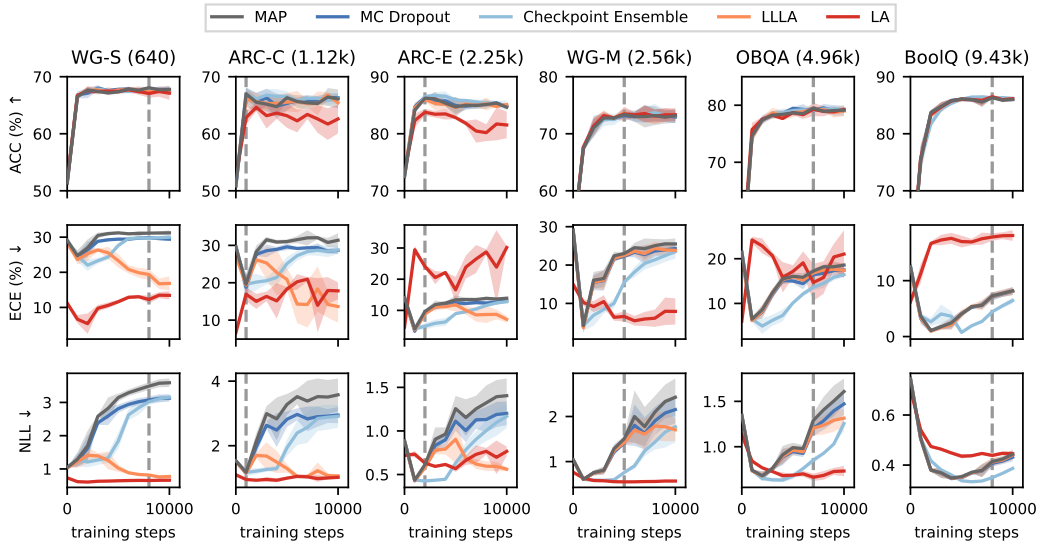

Figure 11: Fine-tuning of LlaMA2-7B across six common sense reasoning tasks (presented column-wise, with number of training examples in brackets), evaluated on the test set every 1000 gradient steps. Laplace (diagonal) prior precision is tuned using model evidence.

Table 14: Comparison of different post-hoc methods applied to the fine-tuned LlaMA2-7B across six common sense reasoning tasks. Results are evaluated at the early stopping point of 5000 gradient steps.

Methods	Metrics	WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ
ACC ↑	MAP	$67.4_{0.3}$	$66.3_{0.6}$	$84.7_{1.5}$	$73.4_{0.4}$	$78.7_{0.4}$	$86.1_{0.2}$
	MC Drop	$67.8_{0.1}$	$65.3_{1.0}$	$85.0_{1.3}$	$73.2_{0.5}$	$79.5_{0.2}$	$86.0_{0.3}$
	Ckpt Ens	$67.4_{0.3}$	$66.6_{0.8}$	$85.9_{0.2}$	$73.5_{0.7}$	$79.1_{0.2}$	$86.0_{0.3}$
	LLLA	$67.7_{0.3}$	$65.5_{1.4}$	$84.6_{1.2}$	$73.6_{0.7}$	$78.7_{0.7}$	$86.0_{0.3}$
	LA	$67.4_{0.4}$	$63.2_{1.9}$	$82.8_{0.2}$	$73.5_{1.1}$	$78.7_{0.4}$	$86.1_{0.3}$
ECE ↓	MAP	$31.2_{0.3}$	$31.0_{0.5}$	$13.4_{1.3}$	$23.0_{0.1}$	$16.1_{0.6}$	$4.0_{0.5}$
	MC Drop	$29.4_{0.3}$	$29.6_{0.8}$	$12.4_{1.2}$	$22.2_{0.2}$	$15.0_{0.4}$	$4.1_{0.4}$
	Ckpt Ens	$27.6_{0.6}$	$25.1_{1.6}$	$\mathbf{9.0_{0.3}}$	$15.5_{0.2}$	$\mathbf{10.1_{0.3}}$	$\mathbf{0.7_{0.2}}$
	LLLA	$23.3_{1.7}$	$20.9_{2.6}$	$11.8_{1.9}$	$22.5_{0.6}$	$15.9_{0.3}$	$4.1_{0.6}$
	LA	$\mathbf{11.6_{0.5}}$	$\mathbf{18.3_{0.5}}$	$16.6_{3.7}$	$\mathbf{6.6_{1.2}}$	$17.2_{1.2}$	$17.0_{0.9}$
NLL ↓	MAP	$3.15_{0.10}$	$3.28_{0.29}$	$1.26_{0.13}$	$1.51_{0.05}$	$0.99_{0.05}$	$0.35_{0.01}$
	MC Drop	$2.81_{0.11}$	$2.82_{0.21}$	$1.11_{0.10}$	$1.41_{0.03}$	$0.95_{0.04}$	$0.35_{0.01}$
	Ckpt Ens	$1.84_{0.22}$	$1.93_{0.22}$	$0.63_{0.01}$	$0.76_{0.01}$	$\mathbf{0.68_{0.03}}$	$\mathbf{0.34_{0.00}}$
	LLLA	$1.02_{0.10}$	$1.30_{0.11}$	$0.90_{0.24}$	$1.42_{0.05}$	$0.96_{0.05}$	$0.35_{0.01}$
	LA	$\mathbf{0.64_{0.01}}$	$\mathbf{1.00_{0.04}}$	$\mathbf{0.57_{0.03}}$	$\mathbf{0.55_{0.01}}$	$0.69_{0.01}$	$0.44_{0.01}$

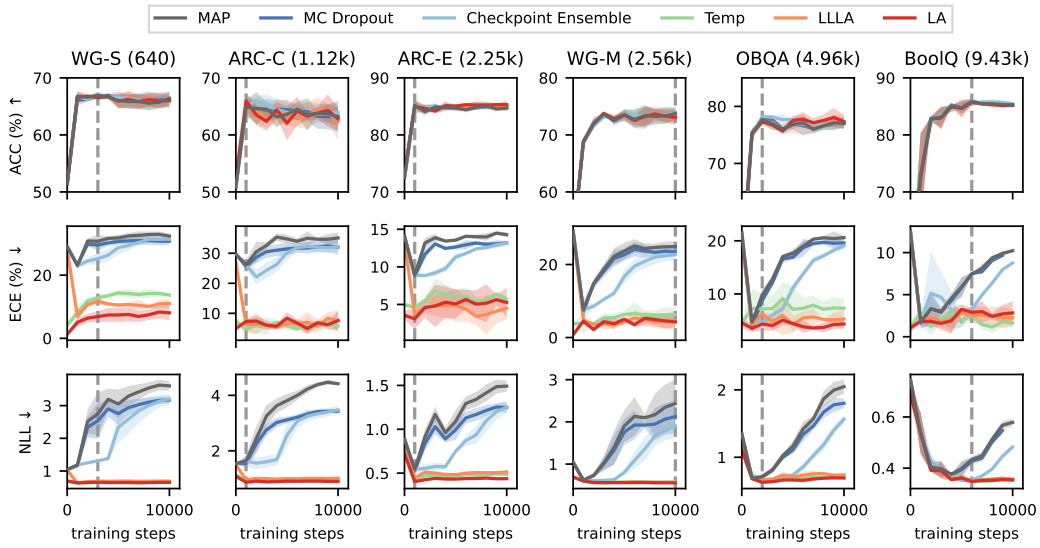

Figure 12: Fine-tuning of LlaMA2-7B across six common sense reasoning tasks (presented column-wise, with number of training examples in brackets), evaluated on the test set every 1000 gradient steps. The vertical dashed line gives the checkpoint with optimal MAP performance on a held-out validation set. Temperature scaling and Laplace (diagonal) prior precision are tuned on the validation set.

Table 15: Comparison of different post-hoc methods applied to the fine-tuned LlaMA2-7B across six common sense reasoning tasks, with a validation set split from the training set used for tuning temperature and Laplace prior precision. Results are evaluated at the best MAP performance checkpoint observed on the validation set.

Methods	Metrics	WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ
ACC ↑	MAP	$67.0_{0.6}$	$64.9_{1.1}$	$85.2_{0.6}$	$73.7_{0.9}$	$77.7_{0.8}$	$85.8_{0.4}$
	MC Drop	$66.7_{0.3}$	$64.9_{1.9}$	$85.1_{0.5}$	$73.5_{0.9}$	$77.7_{0.2}$	$85.9_{0.4}$
	Ckpt Ens	$66.7_{0.3}$	$64.9_{1.1}$	$85.2_{0.6}$	$73.8_{1.0}$	$78.2_{0.2}$	$85.4_{0.3}$
	Temp	$67.0_{0.6}$	$64.9_{1.1}$	$85.2_{0.6}$	$73.7_{0.9}$	$77.7_{0.8}$	$85.8_{0.4}$
	LLLA	$66.7_{0.3}$	$64.4_{1.0}$	$85.1_{0.8}$	$73.1_{1.2}$	$77.2_{0.3}$	$85.7_{0.4}$
	LA	$66.5_{0.3}$	$66.0_{1.2}$	$85.0_{1.2}$	$73.1_{1.1}$	$77.3_{0.3}$	$85.7_{0.5}$
ECE ↓	MAP	$30.8_{1.8}$	$26.1_{1.4}$	$8.9_{0.3}$	$24.9_{1.3}$	$9.8_{1.0}$	$7.4_{0.1}$
	MC Drop	$29.5_{1.6}$	$25.6_{0.7}$	$8.8_{0.6}$	$23.5_{1.2}$	$8.8_{0.8}$	$7.5_{0.1}$
	Ckpt Ens	$25.2_{1.6}$	$26.1_{1.4}$	$8.9_{0.3}$	$22.8_{1.4}$	$4.7_{0.5}$	$3.2_{0.5}$
	Temp	$12.8_{0.9}$	$\mathbf{4.6_{1.0}}$	$4.7_{0.8}$	$6.3_{1.6}$	$7.2_{2.6}$	$\mathbf{2.5_{0.3}}$
	LLLA	$11.8_{1.1}$	$6.0_{1.8}$	$3.7_{0.5}$	$4.5_{2.2}$	$6.2_{0.5}$	$2.6_{0.9}$
	LA	$\mathbf{6.9_{1.5}}$	$7.3_{0.6}$	$\mathbf{3.1_{1.2}}$	$\mathbf{4.3_{1.3}}$	$\mathbf{4.3_{1.1}}$	$2.9_{0.5}$
NLL ↓	MAP	$2.75_{0.57}$	$1.64_{0.19}$	$0.54_{0.03}$	$2.43_{0.50}$	$0.71_{0.03}$	$0.43_{0.01}$
	MC Drop	$2.54_{0.49}$	$1.55_{0.16}$	$0.52_{0.04}$	$2.12_{0.35}$	$0.71_{0.04}$	$0.43_{0.01}$
	Ckpt Ens	$1.31_{0.04}$	$1.64_{0.18}$	$0.54_{0.03}$	$1.89_{0.24}$	$0.65_{0.02}$	$\mathbf{0.35_{0.01}}$
	Temp	$0.68_{0.01}$	$0.90_{0.01}$	$0.43_{0.02}$	$0.58_{0.01}$	$0.67_{0.02}$	$\mathbf{0.35_{0.00}}$
	LLLA	$0.68_{0.02}$	$0.95_{0.03}$	$0.44_{0.01}$	$0.57_{0.01}$	$0.66_{0.02}$	$\mathbf{0.35_{0.00}}$
	LA	$\mathbf{0.66_{0.02}}$	$\mathbf{0.86_{0.03}}$	$\mathbf{0.40_{0.02}}$	$\mathbf{0.55_{0.01}}$	$\mathbf{0.63_{0.01}}$	$\mathbf{0.35_{0.01}}$

Subject	**Task**
Computer Science (CS)	college computer science computer security high school computer science machine learning
Engineering (Eng)	electrical engineering
Law	international law jurisprudence professional law
Health	anatomy clinical knowledge college medicine human aging nutrition professional medicine virology

Table 16: Far OOD MMLU subjects and tasks.