Variational Low-Rank Adaptation Using IVON

Anonymous Author(s)
Affiliation
Address
email

Abstract

We use variational learning to improve accuracy and calibration of low-rank adaptation (LoRA) finetuning in large language models. Specifically, we employ the Improved Variational Online Newton (IVON) optimizer, which is a drop-in replacement of AdamW but significantly improves the performance with negligible overhead. We test our method by finetuning a Llama 2 model with 7 billion parameters on a range of commonsense reasoning datasets. Compared to AdamW finetuning, IVON improves accuracy by 2.8% and ECE by 4.6% on average. Our work provides more evidence for the effectiveness of variational learning in large language models. A link to the code will be provided in the final paper.

1 Introduction

Large Language Models (LLMs) exhibit impressive capabilities across a wide range of natural language generation and understanding tasks but adapting them to new data can be hard, because lots of compute and memory is required due to their large number of parameters. While techniques like Low-Rank Adaptation (LoRA) [9] can alleviate this and enable finetuning on a small compute budget, models trained with LoRA still tend to be badly calibrated [23]. Bayesian methods promise to alleviate this, and in fact several Bayesian adaptations of LoRA have been proposed, such as Laplace-LoRA [23], LoRA ensembles [21], or SWAG-LoRA [16]. However, they all require either additional postprocessing steps or multiple training runs.

We present a method that uses variational learning [7, 8] to improve both the accuracy and calibration of LoRA-trained models without training overhead or additional postprocessing steps. The proposed method is called IVON-LoRA and we use the IVON optimizer [18] to estimate a diagonal Gaussian distribution over the low-rank factors that are added in LoRA. Learning the diagonal Gaussian only induces a negligible overhead (around $\approx 1\%$, as discussed in Sec. 3) in terms of training speed when compared to AdamW but provides a posterior from which diverse models can be sampled. We find that IVON-LoRA improves both calibration and accuracy and averaging the predictions of multiple sampled models further significantly improves calibration.

Several recent works consider related approaches to improve language model finetuning. Following a PAC-Bayesian framework, Liu et al. [11] proposes to finetune the full model using perturbed gradient descent. Chen and Garner [2] uses variational learning to estimate parameter importance in adaptive LoRA [24]. However, neither of them has been shown to work for recent billion-scale LLMs. Similar to Liu et al. [11], Zhelnin et al. [25] shows that Gaussian noise injection can improve instruction tuning of LLMs. Different from our work, they finetune on a significantly larger instruction dataset, which is more resilient to bad calibration and overfitting.

We show the effectiveness of our method by finetuning a Llama 2 model with 7 billion parameters on a range of commonsense reasoning tasks. We compare our method to both standard LoRA training with AdamW, and various Bayesian variants of LoRA [16, 23]. Our results show that IVON-LoRA outperforms AdamW significantly in terms of both accuracy and calibration on all tasks. Our method

Submitted to Workshop on Fine-Tuning in Machine Learning (FITML@NeurIPS 2024). Do not distribute.

gives better accuracy than the other Bayesian adaptations of LoRA and provides competitive results in calibration metrics but without requiring training overhead or additional postprocessing.

Altogether, IVON-LoRA is easy to implement and can be used as a plug-in replacement for LoRA training with AdamW that enables better accuracy and calibration without overhead. Our work further shows the effectiveness of variational learning for LLMs.

2 Variational low-Rank adaptation using IVON

The parameters of LLMs contain weight matrices $\mathbf{W} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(D)}\}$, where $\mathbf{W}^{(i)} \in \mathbb{R}^{d_i \times k_i}$. Instead of directly finetuning the weight matrices $\mathbf{W}$, low-rank adaptation (LoRA) [9] learns low-rank increments $\Delta\mathbf{W} = \{\Delta\mathbf{W}^{(1)}, \ldots, \Delta\mathbf{W}^{(D)}\}$. Each increment $\Delta\mathbf{W}^{(i)} = \mathbf{B}^{(i)}\mathbf{A}^{(i)}$ is parametrized by two smaller matrices $\mathbf{B}^{(i)} \in \mathbb{R}^{d_i \times r}$, $\mathbf{A}^{(i)} \in \mathbb{R}^{r \times k_i}$. r is a hyperparameter that determines the rank of the increments $\Delta\mathbf{W}$, and is smaller than d_i and k_i. LoRA initializes all $\mathbf{A}^{(i)}$ as random Gaussian matrices and $\mathbf{B}^{(i)}$ to zero. Given a set of pretrained weight matrices $\mathbf{W}_0$, LoRA optimizes the following loss function,

$$\mathcal{L}(\Delta\mathbf{W}) = \frac{1}{N} \sum_{n=1}^{N} \ell_n(\mathbf{W}_0 + (\alpha/r)\Delta\mathbf{W}), \tag{1}$$

where $\alpha > 0$ is an additional tuning hyperparameter, α/r makes hyperparameter choice consistent when varying the rank r and ℓ_n is the loss for data example n.

We propose a variational learning approach and search for mean-field Gaussian posterior distributions over the low-rank factors $\mathbf{A}$ and $\mathbf{B}$. Denoting $\boldsymbol{\theta} = (\mathbf{A}, \mathbf{B})$, we assume that $q(\boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{m}, \text{diag}(\mathbf{v}))$, meaning that for every scalar entry of $\mathbf{A}$ and $\mathbf{B}$ we learn a scalar mean and variance. This amounts to minimizing the variational objective

$$\mathcal{L}_{\text{variational}}(q) = \lambda \mathbb{E}_{q(\boldsymbol{\theta})}\left[\mathcal{L}(\Delta\mathbf{W})\right] + \mathbb{D}_{\text{KL}}(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta})), \tag{2}$$

over the posterior mean $\mathbf{m}$ and variance $\mathbf{v}$. Here the prior $p(\boldsymbol{\theta}) \sim \mathcal{N}(0, v_0\mathbf{I})$ is chosen as a Gaussian with zero mean and a constant variance v_0, and $\lambda > 0$ is weighting-parameter. Setting $\lambda = N$ one approximates the Bayesian posterior, and larger values target a cold posterior. The variational objective (2) can be optimized using the IVON [18] optimizer,[1] which we employ to learn the variational posterior $q(\boldsymbol{\theta})$ over the low-rank factors $\mathbf{A}$, $\mathbf{B}$. Despite learning a Gaussian mean-field posterior over $\mathbf{A}$ and $\mathbf{B}$, the optimizer has negligible memory overhead when compared to Adam. This is due to the variance playing an analogous role to the second-moment estimate in Adam, we refer the interested reader to Shen et al. [18] for the details.

3 Experiments

To evaluate the effectiveness of the proposed method, we use IVON to finetune pretrained Llama 2 [19] model with 7 billion parameters on six datasets with commonsense reasoning multiple-choice or true/false questions. These six datasets are WinoGrande-Small (WG-S), WinoGrande-Medium (WG-M) [17], ARC-Challenge (ARC-C), ARC-Easy (ARC-E) [4], OpenBookQA (OBQA) [14], and BoolQ [3]. We evaluate the performance of the trained LoRA adapters by calculating the accuracy, expected calibration error (ECE) and negative log-likelihood (NLL) on the test set. We add NLL as another metric for calibration, since ECE is found not always reliable for this purpose [1]. As for baseline methods, we compare the performance of IVON-LoRA adapters with LoRA adapters trained using AdamW. We also consider other methods for improving generalization and calibration, including Monte Carlo Dropout (MC Dropout) [6], Laplace Approximation (LA) [23], Stochastic Weight Averaging (SWA) [10, 16], and SWA-Gaussian (SWAG) [12, 16].

Compared to standard AdamW finetuning which learns a point estimation of the parameters, IVON finetuning learns a *distribution* over the parameters, which allows sampling to obtain an ensemble of models during inference. Considering this, we evaluate two variants of inference with IVON-LoRA adapters: predicting at the mean of the posterior distribution, and predicting at an ensemble of 10

[1] `https://github.com/team-approx-bayes/ivon`

Table 1: Comparison of techniques applied to finetuning/finetuned Llama-2 7B model across commonsense reasoning datasets. Results at the end of training are reported, with subscripts indicating standard error of the mean across 3 runs. We show the relative metric changes achieved by using IVON over AdamW in parentheses, with improvements in blue and degradation in red. The methods marked with * do not require customized pipeline or additional computation during inference.

Metrics	Methods	WG-S	ARC-C	ARC-E	WG-M	OBQA	BoolQ	Average
ACC ↑	AdamW*	$66.5_{0.4}$	$66.7_{0.5}$	$84.9_{0.2}$	$73.5_{0.4}$	$78.9_{0.7}$	$85.8_{0.1}$	76.1
	+ MC Drop	$66.7_{0.4}$	$67.3_{0.5}$	$84.8_{0.4}$	$73.7_{0.2}$	$79.3_{0.5}$	$85.9_{0.2}$	76.3
	+ LA (KFAC)	$66.6_{0.3}$	$66.0_{1.4}$	$84.3_{0.4}$	$73.2_{0.3}$	$78.6_{0.9}$	$85.7_{0.2}$	75.7
	+ LA (diag)	$66.2_{0.3}$	$61.2_{1.9}$	$81.8_{0.5}$	$73.3_{0.3}$	$79.7_{0.8}$	$85.7_{0.2}$	74.7
	+ SWA*	$69.7_{0.6}$	$67.2_{1.3}$	$85.2_{0.1}$	$75.6_{0.2}$	$79.8_{0.5}$	$85.5_{0.1}$	77.2
	+ SWAG	$69.4_{0.6}$	$68.4_{1.3}$	$85.1_{0.2}$	$75.2_{0.4}$	$80.1_{0.1}$	$85.2_{0.2}$	77.2
	IVON@mean*	(+5.6) $\mathbf{72.1}_{0.5}$	(+3.2) $69.9_{0.7}$	(+2.6) $\mathbf{87.5}_{0.6}$	(+3.1) $\mathbf{76.6}_{0.5}$	(+2.0) $80.9_{0.6}$	(+0.3) $86.1_{0.2}$	(+2.8) 78.9
	IVON	(+5.7) $\mathbf{72.2}_{0.5}$	(−0.4) $66.3_{0.6}$	(+0.8) $85.7_{0.3}$	(+2.9) $\mathbf{76.4}_{0.6}$	(+1.5) $80.4_{0.4}$	(−0.1) $85.7_{0.2}$	(+1.7) 77.8
ECE (×100) ↓	AdamW*	$32.8_{0.5}$	$31.4_{0.6}$	$14.5_{0.3}$	$25.3_{0.4}$	$19.1_{0.8}$	$7.6_{0.2}$	21.8
	+ MC Drop	$30.7_{0.3}$	$28.8_{0.7}$	$13.4_{0.4}$	$23.6_{0.1}$	$17.5_{0.6}$	$7.6_{0.3}$	20.2
	+ LA (KFAC)	$\mathbf{5.2}_{2.0}$	$\mathbf{12.4}_{2.5}$	$\mathbf{5.4}_{1.6}$	$11.1_{0.2}$	$5.5_{0.2}$	$3.9_{0.1}$	7.3
	+ LA (diag)	$12.4_{1.2}$	$\mathbf{16.3}_{1.7}$	$24.2_{3.6}$	$\mathbf{5.8}_{0.6}$	$13.1_{1.2}$	$19.7_{0.2}$	15.3
	+ SWA*	$19.7_{0.5}$	$24.6_{0.8}$	$9.8_{0.2}$	$12.3_{1.4}$	$9.7_{0.4}$	$2.4_{0.2}$	13.1
	+ SWAG	$12.9_{1.1}$	$\mathbf{15.6}_{1.1}$	$\mathbf{5.7}_{0.3}$	$8.0_{1.2}$	$5.1_{0.4}$	$\mathbf{1.1}_{0.4}$	8.1
	IVON@mean*	(−5.3) $27.5_{0.4}$	(−5.6) $25.8_{0.4}$	(−4.4) $10.1_{0.4}$	(−2.3) $23.0_{0.5}$	(−7.9) $11.2_{0.5}$	(−2.0) $5.6_{0.1}$	(−4.6) 17.2
	IVON	(−11.0) $21.8_{0.8}$	(−20.7) $\mathbf{10.7}_{0.4}$	(−10.9) $\mathbf{3.6}_{0.6}$	(−3.9) $21.4_{0.5}$	(−15.8) $\mathbf{3.3}_{0.9}$	(−5.0) $2.6_{0.2}$	(−11.2) 10.6
NLL ↓	AdamW*	$4.19_{0.43}$	$3.71_{0.49}$	$1.52_{0.05}$	$2.03_{0.06}$	$1.54_{0.05}$	$0.44_{0.01}$	2.24
	+ MC Drop	$3.75_{0.33}$	$3.25_{0.38}$	$1.36_{0.07}$	$1.85_{0.05}$	$1.40_{0.04}$	$0.43_{0.01}$	2.01
	+ LA (KFAC)	$\mathbf{0.63}_{0.01}$	$\mathbf{0.96}_{0.03}$	$0.49_{0.02}$	$0.76_{0.01}$	$0.68_{0.00}$	$0.37_{0.00}$	0.65
	+ LA (diag)	$0.66_{0.01}$	$\mathbf{1.05}_{0.05}$	$0.70_{0.05}$	$\mathbf{0.57}_{0.01}$	$0.65_{0.01}$	$0.47_{0.00}$	0.68
	+ SWA*	$0.87_{0.02}$	$1.34_{0.06}$	$0.55_{0.00}$	$0.63_{0.04}$	$0.65_{0.02}$	$0.34_{0.00}$	0.73
	+ SWAG	$0.68_{0.02}$	$\mathbf{1.00}_{0.04}$	$0.46_{0.00}$	$\mathbf{0.56}_{0.02}$	$\mathbf{0.55}_{0.02}$	$0.34_{0.00}$	0.60
	IVON@mean*	(−0.69) $3.50_{0.07}$	(−1.74) $1.97_{0.03}$	(−0.83) $0.69_{0.00}$	(+0.40) $2.43_{0.04}$	(−0.88) $0.66_{0.02}$	(−0.08) $0.36_{0.00}$	(−0.64) 1.60
	IVON	(−1.94) $2.25_{0.09}$	(−2.71) $\mathbf{1.00}_{0.03}$	(−1.12) $\mathbf{0.40}_{0.00}$	(+0.13) $2.16_{0.01}$	(−1.00) $\mathbf{0.54}_{0.01}$	(−0.11) $\mathbf{0.33}_{0.00}$	(−1.13) 1.11

samples from the posterior distribution (referred to as IVON@mean and IVON, respectively). For a fair comparison, we use the same number of samples for MC Dropout, SWA and SWAG.

We present the results in Table 1. First, we observe that IVON, as an alternative to AdamW, significantly improves the generalization of LoRA finetuning. When evaluated at the mean, IVON outperforms standard AdamW finetuning and other Bayesian adaptations of LoRA on all datasets in terms of accuracy, often by a large margin. We also observe that IVON exhibits improved calibration compared to AdamW and MC Dropout baselines, as indicated by the lower ECE and NLL values. We want to highlight that these improvements can be achieved with minimal changes to the codebase, that is, replacing the AdamW optimizer with IVON, and without any additional postprocessing steps or noticeable overhead.

Next, we observe that ensembling with samples from IVON's posterior distribution further improves calibration. When evaluate at an ensemble of 10 samples, IVON outperforms all other methods and is comparable to the best-performing LA (with a Kronecker-factored Hessian) and SWAG on ECE and NLL. Notably, IVON achieves this despite using a diagonal Hessian and without an additional pass through the data for computing Hessians at the converged point as in Laplace methods. With this improvement in calibration, IVON still maintains comparable or better accuracy over other methods.

We also notice that by scaling the learned variance during inference (sampling from $\mathcal{N}(\mathbf{m}, \mathrm{diag}(\tau\mathbf{v}))$ with τ being a scaling factor), the ensemble of IVON samples can have different behaviors in terms of accuracy and calibration. In Table 2, we present the results of IVON with different choices of τ during inference. When a smaller τ is used, the ensemble of IVON samples achieves slight improvements both in accuracy and calibration. On the other hand, the ensemble is further improved in calibration at the cost of accuracy when a larger τ is used. This can be useful in tweaking the trade-off between accuracy and calibration to suit the needs of different applications.

Finally, we observe that the overhead induced by IVON is negligible. To investigate this, we profile our training code on an NVIDIA RTX 6000 Ada GPU. In our test run, the forward pass, loss computation, and backward pass of a training step take in total 316.3ms on average. As for the overhead of IVON, the sampling procedure and the optimization step of each training step take 1.8ms and 1.0ms on average, respectively, which is less than 1% of the running time of a training step.

Table 2: Comparison of different variance scaling factor τ used during IVON inference. For $\tau = 0$, it is equivalent to inference at the posterior mean. For $\tau = 1$, it recovers the standard sampling. Results at the end of training are reported. The subscripts indicate standard error of the mean across 3 runs.

Metrics	τ	**WG-S**	**ARC-C**	**ARC-E**	**WG-M**	**OBQA**	**BoolQ**
	0 (IVON@mean)	$72.1_{0.5}$	$69.9_{0.7}$	$\mathbf{87.5}_{0.6}$	$76.6_{0.5}$	$\mathbf{80.9}_{0.6}$	$86.1_{0.2}$
	0.25	$72.3_{0.6}$	$\mathbf{71.1}_{1.1}$	$87.4_{0.6}$	$76.6_{0.5}$	$\mathbf{80.9}_{0.7}$	$\mathbf{86.2}_{0.2}$
	0.5	$\mathbf{72.4}_{0.6}$	$70.1_{1.1}$	$87.2_{0.6}$	$\mathbf{76.7}_{0.5}$	$80.8_{0.8}$	$86.1_{0.2}$
ACC $\uparrow$	0.75	$72.2_{0.7}$	$68.6_{1.0}$	$86.7_{0.4}$	$76.5_{0.4}$	$80.7_{0.4}$	$86.1_{0.1}$
	1 (IVON)	$72.2_{0.5}$	$66.3_{0.6}$	$85.7_{0.3}$	$76.4_{0.6}$	$80.4_{0.4}$	$85.7_{0.2}$
	1.25	$71.7_{0.5}$	$56.0_{0.6}$	$81.5_{0.6}$	$76.3_{0.5}$	$75.6_{0.4}$	$84.8_{0.1}$
	1.5	$70.5_{0.4}$	$30.2_{0.4}$	$48.6_{2.8}$	$76.6_{0.4}$	$53.5_{1.9}$	$82.7_{0.4}$
	0 (IVON@mean)	$27.5_{0.4}$	$25.8_{0.4}$	$10.1_{0.4}$	$23.0_{0.5}$	$11.2_{0.5}$	$5.6_{0.1}$
	0.25	$25.6_{0.5}$	$21.9_{0.7}$	$9.2_{0.4}$	$21.7_{0.5}$	$10.4_{0.8}$	$5.1_{0.2}$
ECE	0.5	$24.3_{0.5}$	$19.3_{0.9}$	$8.1_{0.5}$	$21.6_{0.3}$	$8.8_{0.6}$	$4.7_{0.1}$
($\times 100$) $\downarrow$	0.75	$23.1_{0.7}$	$15.7_{1.0}$	$6.2_{0.3}$	$21.6_{0.3}$	$6.3_{0.8}$	$3.9_{0.3}$
	1 (IVON)	$21.8_{0.8}$	$10.7_{0.4}$	$\mathbf{3.6}_{0.6}$	$21.4_{0.5}$	$\mathbf{3.3}_{0.9}$	$2.6_{0.2}$
	1.25	$20.4_{0.9}$	$\mathbf{7.5}_{0.6}$	$4.9_{0.7}$	$21.3_{0.4}$	$9.2_{0.8}$	$\mathbf{1.1}_{0.3}$
	1.5	$\mathbf{19.2}_{0.6}$	$14.2_{2.0}$	$5.6_{1.9}$	$\mathbf{20.9}_{0.2}$	$15.0_{1.6}$	$2.1_{0.4}$
	0 (IVON@mean)	$3.50_{0.07}$	$1.97_{0.03}$	$0.69_{0.00}$	$2.43_{0.04}$	$0.66_{0.02}$	$0.36_{0.00}$
	0.25	$3.11_{0.05}$	$1.70_{0.03}$	$0.62_{0.01}$	$2.22_{0.01}$	$0.63_{0.02}$	$0.35_{0.00}$
	0.5	$2.87_{0.08}$	$1.41_{0.02}$	$0.53_{0.01}$	$2.22_{0.02}$	$0.59_{0.02}$	$0.35_{0.00}$
NLL $\downarrow$	0.75	$2.52_{0.07}$	$1.17_{0.02}$	$0.46_{0.01}$	$2.20_{0.02}$	$0.54_{0.01}$	$\mathbf{0.34}_{0.00}$
	1 (IVON)	$2.20_{0.06}$	$\mathbf{0.99}_{0.01}$	$\mathbf{0.41}_{0.02}$	$2.17_{0.02}$	$\mathbf{0.53}_{0.01}$	$\mathbf{0.34}_{0.00}$
	1.25	$1.88_{0.08}$	$1.11_{0.01}$	$0.51_{0.02}$	$2.14_{0.03}$	$0.65_{0.01}$	$0.35_{0.00}$
	1.5	$\mathbf{1.60}_{0.10}$	$1.47_{0.01}$	$1.19_{0.05}$	$\mathbf{2.11}_{0.03}$	$1.17_{0.01}$	$0.39_{0.01}$

4 Discussion

Our direct variational learning approach using IVON is surprisingly effective for improving calibration and accuracy in LoRA finetuning. Given the strong results, we hope that this work invigorates research in variational methods for LLMs. Reasons for IVON's success are not fully understood, but one hypothesis is the prevention of overfitting as the finetuning datasets are often comparably small. This may be attributed to the preference for simpler solutions (flatter minima) which is inherent in variational learning [8, 7].

On most of the datasets, ensemble of IVON samples outperforms IVON evaluated at the posterior mean on ECE and NLL, but at the cost of a slight decrease in accuracy. We want to point out this is possibly due to the limited number of samples used in the ensemble. We draw 10 samples for all the ensemble-based methods in our experiments, both to follow the setting in Yang et al. [23] and to keep the computational cost manageable. It is possible that using more IVON samples could further improve the performance of the ensemble, which is reported in Shen et al. [18] on image classification tasks. Nevertheless, the parameter uncertainty obtained by IVON is expected to be useful for several downstream tasks such as sensitivity analysis [15] and model merging [5], which will be explored in future work.

A limitation shared with other Bayesian LoRA methods [23, 16] is that the learned posterior over the increment $\Delta\mathbf{W}$ is non-Gaussian, as it is the product of two Gaussian random variables. Therefore, it cannot be easily merged down into or easily combined with Gaussian uncertainty on the full weights $\mathbf{W}$. A different approach would be to use a variational low-rank correction to correct the mean and variance of a Laplace approximation of the original model. van Niekerk and Rue [20] propose such a low-rank approach in the context of latent Gaussian models, and adapting these ideas to large language models may be an interesting direction for future work.

IVON also has some practical limitations. The method introduces two new hyperparameters over AdamW, which are λ in (2) and the initialization of the posterior variance. This makes tuning IVON a bit more involved than tuning AdamW and the results depend on setting these parameters well. While a good heuristic is to set λ as small as possible while still retaining stable training and setting the posterior initialization in the order of magnitude of the final posterior variance, more principled or automatic ways to set them reliably would be desirable.

References

[1] Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernández. Stop measuring calibration when humans disagree. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1892–1915, 2022.

[2] Haolin Chen and Philip N Garner. A Bayesian interpretation of adaptive low-rank adaptation. *arXiv:2409.10673*, 2024.

[3] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, 2019.

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

[5] Nico Daheim, Thomas Möllenhoff, Edoardo Maria Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. Model merging by uncertainty-based gradient matching. In *International Conference on Learning Representations (ICLR)*, 2024.

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

[7] Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[8] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Conference on Learning Theory (COLT)*, 1993.

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

[10] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[11] Guangliang Liu, Zhiyu Xue, Xitong Zhang, Kristen Marie Johnson, and Rongrong Wang. PAC-tuning: Fine-tuning pretrained language models with PAC-driven perturbed gradient descent. *arXiv:2310.17588*, 2023.

[12] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32, 2019.

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

[14] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, 2018.

[15] Peter Nickl, Lu Xu, Dharmesh Tailor, Thomas Möllenhoff, and Mohammad Emtiyaz Khan. The memory perturbation equation: Understanding model's sensitivity to data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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

[17] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[18] Yuesong Shen, Nico Daheim, Bai Cong, Peter Nickl, Gian Maria Marconi, Bazan Clement Emile Marcel Raoul, Rio Yokota, Iryna Gurevych, Daniel Cremers, Mohammad Emtiyaz Khan, and Thomas Möllenhoff. Variational learning is effective for large deep networks. In *International Conference on Machine Learning*, 2024.

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

[20] Janet van Niekerk and Haavard Rue. Low-rank variational bayes correction to the laplace method. *J. Mach. Learn. Res. (JMLR)*, 25(62):1–25, 2024.

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

[22] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

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

[24] Q Zhang, M Chen, A Bukharin, P He, Y Cheng, W Chen, and T Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2023.

[25] Maxim Zhelnin, Viktor Moskvoretskii, Egor Shvetsov, Egor Venediktov, Mariya Krylova, Aleksandr Zuev, and Evgeny Burnaev. GIFT-SW: Gaussian noise injected fine-tuning of salient weights for LLMs. *arXiv:2408.15300*, 2024.

Table 3: IVON hyperparameters used in experiments.

Hyperparameter	**WG-S**	**ARC-C**	**ARC-E**	**WG-M**	**OBQA**	**BoolQ**
Effective sample size	1×10^7	1×10^6	1×10^6	1×10^8	1×10^6	1×10^7
Hessian initialization	3×10^{-4}	1×10^{-3}	1×10^{-3}	3×10^{-4}	1×10^{-3}	3×10^{-4}
Learning rate			0.03			
Gradient momentum			0.9			
Hessian momentum			$1 - 10^{-5}$			
Clip radius			10^{-3}			

Supplementary Material

Details on experimental setup

Our experimental design is based on Yang et al. [23]. We utilize the PEFT [13] library for LoRA adaptation, and apply LoRA to the query and value weights of the attention layers. Unlike in Yang et al. [23], we do not apply LoRA to the output layer due to numerical instability encountered in some preliminary experiments. The base model is quantized to 8-bit precision, with LoRA weights maintained in 16-bit precision. Finetuning is performed on a single NVIDIA RTX 6000 Ada GPU with a batch size of 4 for 10,000 steps, without gradient accumulation.

To finetune a pretrained language model which predicts the next token in a sequence for solving multiple-choice or true/false questions, we need to wrap the text and the choice of each question with predefined prompt templates to an instruction. We then use the pretrained model to predict the next token of the wrapped instruction, and extract the output logits for the tokens standing for "True"/"False" or "A"/"B"/"C"/"D" choices. For the prompt templates, we use the same ones as in Yang et al. [23]. An example of such a prompt (used for WG-S and WG-M datasets) is as follows:

> Select one of the choices that answers the following question: `{question}` Choices: A. `{option1}`. B `{option2}`. Answer:

Hyperparameters

As for the hyperparameters of LoRA and AdamW finetuning, we use the same settings as in Yang et al. [23], which are also the default settings in Huggingface's Transformers [22] and PEFT [13] library. For LoRA, we set the rank r to 8, α to 16, and the dropout rate to 0.1. For AdamW optimizer, we set the initial learning rate to 5×10^{-5}, weight decay to 0, and use a linear learning rate scheduler which decays the learning rate to 0 at the end of the training.

Working IVON hyperparameters and guidelines for choosing them are discussed in Shen et al. [18]. Still, it is not well understood how to choose them in the context of LoRA finetuning. We empirically find that setting λ as small as possible while still retaining stable training is a good heuristic. To choose the initialization value v_0 of the posterior variance, we track the mean value of the running average of the posterior variance for the first few training steps. We notice that if the mean value changes significantly during the first few steps, then the initialization value is likely too far from a reasonable one. We follow the guideline in Shen et al. [18] and set the learning rate of IVON to 0.03, Hessian momentum to $1 - 10^{-5}$, and clip radius to 10^{-3}. Finally, We summarize the hyperparameters of IVON used in our experiments in Table 3.