# Variational Low-Rank Adaptation Using IVON

**Bai Cong**[1,2] **Nico Daheim**[4] **Yuesong Shen**[3,5] **Daniel Cremers**[3,5] **Rio Yokota**[1]
**Mohammad Emtiyaz Khan**[2] **Thomas Möllenhoff**[2]

[1]Institute of Science Tokyo [2]RIKEN Center for AI Project [3]Technical University of Munich
[4]Technical University of Darmstadt [5]Munich Center for Machine Learning

## Abstract

We show that variational learning can significantly improve the accuracy and calibration of Low-Rank Adaptation (LoRA) without a substantial increase in the cost. We replace AdamW by the Improved Variational Online Newton (IVON) algorithm to finetune large language models. For Llama-2 with 7 billion parameters, IVON improves the accuracy over AdamW by $2.8\%$ and expected calibration error by $4.6\%$. The accuracy is also better than the other Bayesian alternatives, yet the cost is lower and the implementation is easier. Our work provides additional evidence for the effectiveness of IVON for large language models. The code is available at `https://github.com/team-approx-bayes/ivon-lora`.

## 1 Introduction

Bayesian methods are expected to improve the accuracy and calibration performance of Large Language Models (LLMs) on downstream tasks, but they rarely succeed at such massive scale and, even when they do, often there is a substantial cost to pay. This is certainly true for finetuning with Low-Rank Adaptation [10], where many Bayesian variants have recently been proposed but they all require additional computations compared to standard finetuning practices. For example, the SWAG-LoRA method [17] needs additional computation to obtain weight mean and a non-trivial approximation of the full covariance matrix. LoRA ensemble [22] requires multiple LoRA weights to be trained. Methods such as Laplace-LoRA [24] require an additional pass through the data to compute a Hessian or Fisher approximation. It is then natural to ask whether it is ever possible to use Bayes to improve LoRA without such overheads and increase in the costs.

Here, we show that the variational (Bayesian) learning can significantly improve both the accuracy and calibration of LoRA finetuning without a substantial increase in the cost. Our proposal is to simply replace the standard optimizers like AdamW by a variational learning algorithm called the Improved Variational Online Newton (IVON) algorithm [19]. IVON uses a nearly identical implementation as AdamW and the swap requires just a few lines of code change. The main advantage of IVON is that its scale vector, used for the learning rate adaptation, also yields an estimate of posterior variance for free. The only minor overhead is due to sampling from the posterior but we show that this cost is negligible in practice (approximately $1\%$ of the total training time). We achieve significant improvements in performance when finetuning the Llama-2 model with 7 billion parameters on a range of commonsense reasoning tasks: accuracy increases by $2.8\%$ while expected calibration error (ECE) decreases by $4.6\%$. The accuracy is also better than the other Bayesian alternatives, yet the cost is much lower and the implementation is easier. Our work provides additional evidence for the work of Shen et al. [19], showing effectiveness of IVON for large deep networks.

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

## 2 Variational low-rank adaptation using IVON

We will introduce our approach that we call IVON-LoRA. The idea is simple: we replace the standard AdamW optimizer by IVON which optimizes a variational-Bayesian objective. In other words, we switch the standard objective used by AdamW to a variational one. More formally, let us denote the AdamW objective by $\ell(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is the vector containing all the entries of LoRA's low-rank parameters (often denoted by $\mathbf{A}$ and $\mathbf{B}$). The variational learning minimizes a different objective where an expectation of $\ell(\boldsymbol{\theta})$ over a distribution $q(\boldsymbol{\theta})$ is used (shown on the right),

$$\min_{\boldsymbol{\theta}} \ \ell(\boldsymbol{\theta}) \quad \text{versus} \quad \min_{q(\boldsymbol{\theta})} \ \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ell(\boldsymbol{\theta})\right] + \frac{1}{\lambda}\,\mathbb{D}_{\text{KL}}[q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta})]. \tag{1}$$

IVON uses a Gaussian $q(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{m}, \text{diag}(\mathbf{v}))$. The mean $\mathbf{m}$ plays a similar role to $\boldsymbol{\theta}$ obtained by AdamW, while the posterior variance vector $\mathbf{v}$ is an additional quantity. The prior $p(\boldsymbol{\theta}) = \mathcal{N}(0, v_0\mathbf{I})$ is a zero mean isotropic Gaussian with a scalar variance $v_0$. A scalar weighting parameter $\lambda$ is used to take care of the data size $N$. This is because $\ell(\boldsymbol{\theta})$ is often an *average* over the whole dataset. Therefore, when using $\lambda = N$, we target the posterior distribution while with larger values we go towards a "colder" posterior [26, 8].

Despite such differences in the objectives, the implementation of IVON is nearly identical to AdamW which makes the replacement easy and can be done by just a few lines of code change. The key point is that estimation of $\mathbf{v}$ is automatically done through the scale vector that adapts the learning rate. Therefore, posterior variances are obtained for free. The only additional step is to sample $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{m}, \text{diag}(\mathbf{v}))$ to evaluate the expectation in Eq. 1, but its overhead can be reduced by using one Monte-Carlo sample per iteration. For the details, we refer to the original IVON paper by Shen et al. [19]. Overall, IVON is a promising alternative to the existing Bayesian approaches that require additional overheads due to either post-processing or extra training runs.

## 3 Experiments

To evaluate the effectiveness of the proposed method, we use IVON to finetune a pretrained Llama-2 model with 7 billion parameters [20] on six datasets with commonsense reasoning multiple-choice or true/false questions. These six datasets are WinoGrande-Small (WG-S), WinoGrande-Medium (WG-M) [18], ARC-Challenge (ARC-C), ARC-Easy (ARC-E) [4], OpenBookQA (OBQA) [15], and BoolQ [3]. We evaluate the performance of the trained LoRA adapters by calculating the accuracy and Expected Calibration Error (ECE) on the test set. We also use test Negative Log-Likelihood (NLL) and Brier score because ECE can be sometimes unreliable [1]. As for the baseline methods, we compare the performance of IVON-LoRA adapters with LoRA adapters trained using AdamW. We also consider other Bayesian alternatives, including Monte Carlo Dropout (MC Dropout) [6], Laplace Approximation (LA) [24], Stochastic Weight Averaging (SWA) [11, 17], and SWA-Gaussian (SWAG) [13, 17].

IVON is evaluated in two ways at test time: first, by using the prediction just at the mean $\mathbf{m}$, and second, by using an averaged prediction over 10 samples from the posterior distribution. The two methods are referred to as 'IVON@mean' and 'IVON', respectively. For a fair comparison, we use the same number of samples for MC Dropout and SWAG.

The results are summarized 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 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, as indicated by the lower ECE, NLL and Brier values.

Next, we observe that ensembling with samples from IVON's posterior distribution further improves calibration. When evaluate at an ensemble of 10 samples drawn from the posterior distribution, IVON outperforms all other methods and is comparable to the best-performing LA (with a Kronecker-factored Hessian) and SWAG on ECE, NLL and Brier. 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.

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 |
| Brier ↓ | AdamW* | $0.66_{0.01}$ | $0.63_{0.01}$ | $0.29_{0.00}$ | $0.51_{0.01}$ | $0.39_{0.01}$ | $0.23_{0.00}$ | 0.45 |
| | + MC Drop | $0.63_{0.01}$ | $0.59_{0.01}$ | $0.28_{0.01}$ | $0.49_{0.00}$ | $0.37_{0.01}$ | $0.22_{0.00}$ | 0.43 |
| | + LA (KFAC) | $\mathbf{0.44}_{0.01}$ | $0.50_{0.02}$ | $0.24_{0.00}$ | $0.40_{0.00}$ | $0.31_{0.01}$ | $\mathbf{0.21}_{0.00}$ | 0.35 |
| | + LA (diag) | $0.47_{0.01}$ | $0.57_{0.03}$ | $0.35_{0.03}$ | $0.38_{0.01}$ | $0.33_{0.01}$ | $0.29_{0.00}$ | 0.40 |
| | + SWA* | $0.48_{0.00}$ | $0.55_{0.01}$ | $0.25_{0.00}$ | $\mathbf{0.37}_{0.01}$ | $0.31_{0.01}$ | $\mathbf{0.21}_{0.00}$ | 0.36 |
| | + SWAG | $\mathbf{0.43}_{0.01}$ | $\mathbf{0.48}_{0.01}$ | $0.23_{0.00}$ | $\mathbf{0.36}_{0.01}$ | $\mathbf{0.28}_{0.01}$ | $\mathbf{0.21}_{0.00}$ | 0.33 |
| | IVON@mean* | (−0.11) $0.55_{0.01}$ | (−0.09) $0.54_{0.01}$ | (−0.07) $\mathbf{0.22}_{0.01}$ | (−0.05) $0.46_{0.01}$ | (−0.09) $0.30_{0.01}$ | (−0.02) $\mathbf{0.21}_{0.00}$ | (−0.05) 0.40 |
| | IVON | (−0.18) $0.48_{0.01}$ | (−0.16) $\mathbf{0.47}_{0.01}$ | (−0.08) $\mathbf{0.21}_{0.01}$ | (−0.07) $0.44_{0.01}$ | (−0.11) $\mathbf{0.28}_{0.01}$ | (−0.02) $\mathbf{0.21}_{0.00}$ | (−0.10) 0.35 |

It is also possible to interpolate between IVON@mean and IVON to achieve the best of both. Specifically, at test time, we can scale $\mathbf{v}$ by a scalar $\tau > 0$, that is, we predict using parameters sampled from $\mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}, \operatorname{diag}(\tau \mathbf{v}))$. For $\tau = 0$, we get IVON@mean and, for $\tau = 1$, we get IVON. Increasing $\tau$ then allows us to gradually explore the neighboring solutions around the mean and take advantage of the diversity to improve calibration with a graceful loss of the accuracy. This is shown in Fig. 1 where we see that, as $\tau$ increases, the error increases (accuracy decreases) but the NLL decreases (calibration improves). In practice, this simple scaling technique is useful to get a desirable trade-off between accuracy and calibration for specific applications.

Finally, we observe that the per-step time overhead of IVON is negligible compared to AdamW. 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. The overall training speed of IVON and AdamW are similar as shown in Figure 2.

## 4 Discussion

Our direct variational learning approach using IVON effectively improves 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 [9, 7].

In a broader context, several recent works consider related approaches to improve language model finetuning. Following a PAC-Bayesian framework, Liu et al. [12] proposes to finetune the full model using perturbed gradient descent. Chen and Garner [2] uses variational learning to estimate parameter

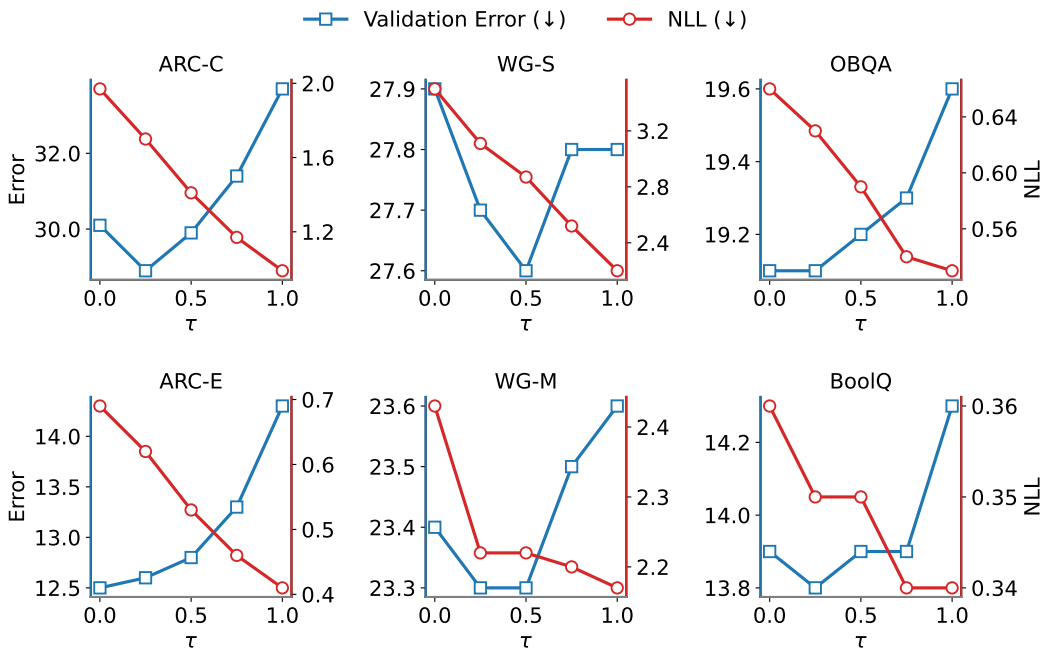


Figure 1: Interpolation between 'IVON@mean' and 'IVON' enables us to trade-off accuracy for better calibration at test time. Essentially, we use $\mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}, \operatorname{diag}(\tau \mathbf{v}))$ with a scalar $\tau \in [0, 1]$. For $\tau = 0$, we recover IVON@mean (leftmost marker) and, for $\tau = 1$, we recover IVON (rightmost marker). Generally, as $\tau$ is increased, the error increases while the NLL decreases. The trend is consistent across datasets (with a few minor exceptions). Metrics are averaged over 3 runs.

importance in adaptive LoRA (AdaLoRA) [25]. However, neither of them has been shown to work for recent billion-scale LLMs. Similar to Liu et al. [12], Zhelnin et al. [27] 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. Nevertheless, these methods still demonstrate the potential of Bayesian methods and noise injection in improving LLM finetuning.

On most of the datasets, ensemble of IVON samples outperforms IVON evaluated at the posterior mean on ECE, NLL and Brier but at the cost of a slight decrease in accuracy. This is perhaps due to the limited number of samples used in the ensemble. In our experiments, we draw 10 samples for all the ensemble-based methods, both to follow the setting in Yang et al. [24] 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. [19] on image classification tasks. Nevertheless, the parameter uncertainty obtained by IVON is expected to be useful for several downstream tasks such as sensitivity analysis [16] and model merging [5], which will be explored in future work.

A limitation shared with other Bayesian LoRA methods [24, 17] is that the learned posterior over the increment low-rank parameters is non-Gaussian because it is a product of two Gaussian random variables. If this is indeed a problem, a workaround could 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 [21] 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, namely the weighting parameter $\lambda$ and the initialization of the posterior variance $\mathbf{v}$. This makes tuning IVON a bit more involved than tuning AdamW and the results depend on setting these parameters well. A good heuristic is to set $\lambda$ as small as possible while still retaining stable training and setting the posterior initialization in the order of magnitude of the final posterior variance. The heuristic works well in practice but the method could still benefit from more principled and automatic ways to set the hyperparameters reliably.

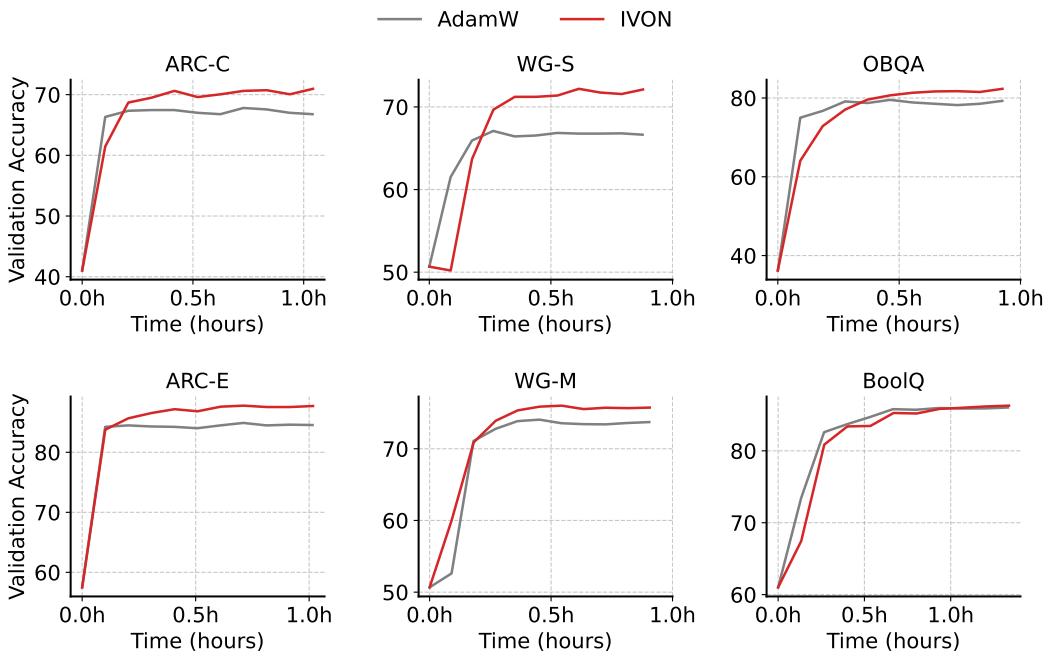


Figure 2: The training speeds of IVON and AdamW are similar. We plot validation accuracies of the two methods versus time in hours. Results are averaged over 3 runs.

## Acknowledgments and Disclosure of Funding

This work is supported by JST CREST Grant Number JPMJCR2112. This research work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. Y. Shen and D. Cremers are supported by the Munich Center for Machine Learning (MCML) and the ERC Advanced Grant SIMULACRON.

## References

[1] Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernández. Stop measuring calibration when humans disagree. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 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 *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 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: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] Peter Grünwald. The safe Bayesian - learning the learning rate via the mixability gap. In *Algorithmic Learning Theory (ALT)*, 2012.

[9] 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.

[10] 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 (ICLR)*, 2022.

[11] P Izmailov, AG Wilson, D Podoprikhin, D Vetrov, and T Garipov. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

[12] Guangliang Liu, Zhiyu Xue, Xitong Zhang, Kristen Johnson, and Rongrong Wang. PAC-tuning: Fine-tuning pre-trained language models with PAC-driven perturbed gradient descent. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[13] 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 (NeurIPS)*, 2019.

[14] 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.

[15] 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 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[16] 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.

[17] 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 *Symposium on Advances in Approximate Bayesian Inference (AABI)*, 2024.

[18] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 2021.

[19] 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 (ICML)*, 2024.

[20] 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:2307.09288*, 2023.

[21] Janet van Niekerk and Haavard Rue. Low-rank variational bayes correction to the Laplace method. *J. Mach. Learn. Res. (JMLR)*, 2024.

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

[23] 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 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

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

[25] 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.

[26] Tong Zhang. From $\varepsilon$-entropy to KL-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, pages 2180–2210, 2006.

[27] 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 2: IVON hyperparameters used in experiments.

| **Hyperparameter** | **WG-S** | **ARC-C** | **ARC-E** | **WG-M** | **OBQA** | **BoolQ** |
|---|---|---|---|---|---|---|
| Effective sample size | $1 \times 10^7$ | $1 \times 10^6$ | $1 \times 10^6$ | $1 \times 10^8$ | $1 \times 10^6$ | $1 \times 10^7$ |
| Hessian initialization | $3 \times 10^{-4}$ | $1 \times 10^{-3}$ | $1 \times 10^{-3}$ | $3 \times 10^{-4}$ | $1 \times 10^{-3}$ | $3 \times 10^{-4}$ |
| 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. [24]. We utilize the PEFT [14] library for LoRA adaptation, and apply LoRA to the query and value weights of the attention layers. Unlike in Yang et al. [24], 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. [24]. 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. [24], which are also the default settings in Huggingface's Transformers [23] and PEFT [14] library. For LoRA, we set the rank $r$ to 8, $\alpha$ to 16, and the dropout rate to 0.1. For AdamW optimizer, we set the initial learning rate to $5 \times 10^{-5}$, 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. [19]. Still, it is not well understood how to choose them in the context of LoRA finetuning. We empirically find that setting $\lambda$ 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. [19] 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 2.