

# A Bayesian Interpretation of Adaptive Low-Rank Adaptation

Haolin Chen

*Idiap Research Institute*, Martigny, Switzerland  
*École Polytechnique Fédérale de Lausanne*, Lausanne, Switzerland  
haolin.chen@idiap.ch

Philip N. Garner

*Idiap Research Institute*  
Martigny, Switzerland  
phil.garner@idiap.ch

**Abstract**—Motivated by the sensitivity-based importance score of the adaptive low-rank adaptation (AdaLoRA), we utilize more theoretically supported metrics, including the signal-to-noise ratio (SNR), along with the Improved Variational Online Newton (IVON) optimizer, for adaptive parameter budget allocation. The resulting Bayesian counterpart not only has matched or surpassed the performance of using the sensitivity-based importance metric but is also a faster alternative to AdaLoRA with Adam. Our theoretical analysis reveals a significant connection between the two metrics, providing a Bayesian perspective on the efficacy of sensitivity as an importance score. Furthermore, our findings suggest that the magnitude, rather than the variance, is the primary indicator of the importance of parameters.

**Index Terms**—variational inference, low-rank adaptation, adaptive budget allocation, importance score.

## I. INTRODUCTION

In the context of the adaptation of large-scale pre-trained models, it has long been of interest to fine-tune the model in a parameter-efficient manner. Parameter-efficient fine-tuning (PEFT) techniques [1] typically optimize a small subset of the model parameters that are either original or additional ones while leaving the rest unchanged. The low-rank adaptation (LoRA) [2] is one of the most efficient and flexible PEFT techniques. Based on the assumption that the change of weights during fine-tuning has a low intrinsic rank, LoRA performs adaptation by optimizing the low-rank approximation of the change of the original weight matrices. Nevertheless, LoRA has limitations as it pre-defines an identical rank for all target weight matrices and therefore ignores the varying importance of weights across modules and layers. This is problematic as adding more trainable parameters to important weights contributes to better performance, however by contrast, doing so to less important weights yields marginal improvements or even inferior outcomes [3].

In light of the limitation, there arises a natural question of how to allocate trainable parameters to different modules according to their importance to maximize the fine-tuning performance. To this end, a variety of techniques for LoRA has been proposed to address the problem, the most representative one of which is AdaLoRA [3]. AdaLoRA parameterizes the delta weight mimicking the singular value decomposition (SVD) to enable dynamic adjustment of the rank: it identifies

the importance of each SVD triplet in the entire model by a sensitivity-based metric and gradually prunes less important triplets during fine-tuning to reach the parameter budget. It has been demonstrated that AdaLoRA can effectively improve the model performance and parameter efficiency compared to LoRA.

Motivated by AdaLoRA, we are primarily interested in the importance scoring mechanism as it can be generically applied to PEFT for parameter selection. The sensitivity-based importance metric is originally based on the heuristic that the importance of parameters can be quantified by the error induced by removing them, which in turn can be approximated by the square of the gradient-weight product [4], [5]. Meanwhile, there are importance metrics with strong theoretical support, many of which originate from Bayesian neural networks (BNNs). A widely recognized metric is the signal-to-noise ratio (SNR) [6]–[8], commonly used in BNN pruning and compression. The interpretation is straightforward: a low SNR makes the neuron’s output too noisy to be useful, while a high SNR indicates valuable, low-noise output. The SNR could be a drop-in replacement for the sensitivity-based importance score in AdaLoRA, allowing the pruning of SVD triplets with low SNRs during fine-tuning for dynamic rank adjustment.

The calculation of SNR requires knowledge of the variance of the parameters, typically assuming they follow a Gaussian distribution; this is closely related to variational inference (VI). VI tackles the optimization task of neural networks by approximating complex posterior distributions of the parameters; this involves selecting a simpler, parameterized distribution and minimizing the Kullback-Leibler (KL) divergence between this distribution and the true posterior. Recent advances in VI [9] have shown not only superior performance in model calibration and predictive uncertainty compared to traditional optimizers like Adam [10], but also high efficiency and effectiveness in large-scale networks.

In this study, we leverage Bayesian importance metrics alongside the Improved Variational Online Newton (IVON) optimizer, a recent advance in VI, to develop a Bayesian counterpart to AdaLoRA, utilizing SNR as the importance score. By comparing its performance with the sensitivity-based importance metric on the GLUE benchmark [11], we demonstrate that the Bayesian approach not only achieves comparable or superior performance but also offers a 10% speed-up over

the original AdaLoRA with Adam. A closer examination of the underlying theory reveals a strong connection between these two metrics, providing a Bayesian interpretation of the sensitivity as an importance score. Additionally, our findings indicate that the magnitude, rather than the variance, is the primary indicator of the importance of parameters. The source code is available.<sup>1</sup>

## II. ADAPTIVE BUDGET ALLOCATION

### A. Overview

The techniques that enable adaptively allocating trainable parameters across different modules and layers generally fall into two categories: importance scoring-based methods and regularization-based methods. For importance scoring-based methods, the key is to find a proper importance metric and prune less important components accordingly. Whilst some work [12], [13] adopts AdaLoRA’s sensitivity-based approach, other heuristic metrics, such as the magnitude of the weight [14] and the accumulated gradient [15], have also been explored. Among regularization-based approaches, diff pruning [16] is representative: it applies  $L_0$  regularization to the delta weight (which shares the same dimensions as the pre-trained weights) and prunes it element-wise according to the magnitude. Similarly but based on LoRA, SoRA [17] introduces a gating unit in between the two LoRA matrices and applies  $L_1$  regularization to the gate to zero out unimportant ranks. However, regularization-based approaches cannot guarantee to achieve target parameter budgets since they depend on unpredictable sparsity regularizations controlled by sparsity-promoting priors and threshold values, and therefore often require onerous hyperparameter tuning.

### B. Revisiting AdaLoRA

AdaLoRA has the following main components.

1) *SVD-based adaptation*: AdaLoRA parameterizes the delta weight in the form of singular value decomposition:  $W = W_0 + \Delta W = W_0 + P\Lambda Q$ , where  $P$  and  $Q$  are singular vectors and the diagonal matrix  $\Lambda$  contains singular values. To avoid the intensive computational cost of SVD, a penalty  $R(P, Q) = \|P^\top P - I\|_F^2 + \|Q^\top Q - I\|_F^2$  is added to the loss to enforce the orthogonality of  $P$  and  $Q$  so that every rank is independent of each other. During adaptation, only the singular values are masked out while the singular vectors are maintained so that dropped triplets can be reactivated later.

2) *Sensitivity-based importance scoring*: The sensitivity is defined as the magnitude of the gradient-weight product:  $I(\theta) = |\theta \nabla_\theta \ell|$ , where  $\theta$  is a trainable parameter. The authors of AdaLoRA argue that the sensitivity itself is too variable and uncertain to be estimated due to the stochasticity of training and therefore propose to use sensitivity smoothing and uncertainty quantification:  $\bar{I}(\theta) = \beta_1 \bar{I}^{t-1}(\theta) + (1 - \beta_1) I^t(\theta)$ ,  $\bar{U}(\theta) = \beta_2 \bar{U}^{t-1}(\theta) + (1 - \beta_2) |I^t(\theta) - \bar{I}^t(\theta)|$ , where  $\bar{I}^t$  is the smoothed sensitivity by exponential moving average and  $\bar{U}^t$  is the uncertainty quantification of  $I$ . The final importance

score is  $s^t(\theta) = \bar{I}^t(\theta) \cdot \bar{U}^t(\theta)$ . The authors compared its performance with the magnitude of singular values and the sensitivity without smoothing and found the proposed metric performed the best.

3) *Global budget scheduler*: The global budget is defined as the total rank of all delta weights in the model. AdaLoRA starts from an initial budget  $b^0$  that is slightly higher (usually 1.5 times) than the target budget  $b^T$ , warms up the training for  $t_i$  steps, and gradually decreases the budget  $b^t$  to reach  $b^T$  following a cubic schedule. After this, the budget distribution is fixed until training finishes after  $t_f$  steps.

### C. Bayesian Importance Scores

In this work, we focus on theoretically supported importance metrics that originate from Bayesian neural networks (BNN). BNNs model weights as probability distributions, enabling the network to quantify uncertainties in its predictions. The most commonly used distribution is the Gaussian distribution, therefore the model is parameterized by two sets of parameters: the mean  $\mu$  and the standard deviation  $\sigma$  (or the variance  $\sigma^2$ , we also refer to  $\sigma$  as variance for the sake of simplicity).

1)  $\text{SNR}(\theta) = |\mu|/\sigma$ : The signal-to-noise ratio (SNR) [6]–[8] is a commonly used importance metric in BNN that considers both the magnitude and the variance (also the uncertainty) of the weights. It has a simple interpretation: a low SNR results in a neuron’s output being too noisy to be useful, while a high SNR signifies meaningful output with minimal noise. It has been utilized in both in-training and post-training pruning of BNNs [6], [18].

2)  $\text{SNR}(|\theta|)$ : Li et al. [18] argue that the random sampling of weights before each forward pass of BNN needs to be considered. Instead of using  $|\mu|$  which is equal to  $|\mathbb{E}_q \theta|$  (where  $q$  is the posterior distribution of parameters), it is more appropriate to use  $\mathbb{E}_q |\theta|$  in the SNR. The resulting metric is:

$$\text{SNR}_q(|\theta|) = \frac{\mu(2\Phi(\frac{\mu}{\sigma})-1) + \frac{2\sigma}{\sqrt{2\pi}} \exp(-\frac{\mu^2}{2\sigma^2})}{\sqrt{\sigma^2 + \mu^2 - [\mu(2\Phi(\frac{\mu}{\sigma})-1) + \frac{2\sigma}{\sqrt{2\pi}} \exp(-\frac{\mu^2}{2\sigma^2})]^2}}, \text{ where}$$

$\Phi(x) := \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp(-\frac{y^2}{2}) dy$  is the cumulative distribution function. It has been shown the new metric outperforms the standard SNR in training sparse BNNs [18].

3)  $|\mu|$  and  $1/\sigma$ : We want to identify the key component in the SNR that reflects the importance of parameters. The absolute value of the mean, or the magnitude, is a straightforward metric that directly impacts the neuron’s output. This metric is widely used in neural network pruning, commonly known as magnitude pruning [19]. Another choice is to use the variance alone as an importance metric. The intuition is that parameters with a low variance have less uncertainty, and therefore are more important.

## III. VARIATIONAL INFERENCE

The calculation of SNR requires approximating a Gaussian distribution over parameters, which is exactly the objective of variational inference (VI). In contrast with traditional deep learning methods that estimate parameters by minimizing the empirical risk  $\ell(\theta)$  with gradient descent, variational methods

<sup>1</sup><https://github.com/idiap/vilora>

estimate a posterior distribution  $q(\theta)$  over parameters by minimizing  $\mathcal{L}(q) = \mathbb{E}_{q(\theta)}[\ell(\theta)] + \mathbb{D}_{KL}(q(\theta)||p(\theta))$ , where  $p(\theta)$  is the prior.

The optimization of  $\mathcal{L}(q)$  is fundamentally different from minimizing  $\ell(\theta)$  using gradient descent. For example, the expectation term requires sampling of  $\theta$  before each forward pass, and the number of parameters of  $q$  is doubled for the commonly used Gaussian distribution with a diagonal covariance. Early approaches [6], [7] that optimize  $q(\theta)$  using gradient descent have failed to scale up on modern architectures. Recent natural gradient-based methods [20] have shown promising results using an Adam-like form; however, they still underperform Adam and have higher computational costs.

The Improved Variational Online Newton (IVON) [9] is a recent VI optimizer that matches the performance of Adam at a comparable computational cost. Its key innovations include bypassing the expensive per-example gradient square computation through a reparameterization trick and incorporating several practical techniques to enhance performance. IVON stands out as the first VI optimizer proven to be both effective and efficient for training large networks, while still delivering the benefits of VI. In our experiments, we utilize IVON to estimate the variance of parameters, enabling the use of SNR as an importance metric.

## IV. EXPERIMENTS

### A. Models and Datasets

We compare the fine-tuning performance of AdaLoRA using different importance scores on DeBERTaV3-base [21]. The experiments are conducted on the General Language Understanding Evaluation (GLUE) benchmark [11], which includes four natural language inference tasks, three similarity and paraphrase tasks, and two single-sentence classification tasks.

### B. Implementation Details

We base our code on the text classification examples of the Hugging Face Transformers library [22] and the Parameter-Efficient Fine-Tuning (PEFT) library [23]. For IVON, we use the official implementation<sup>2</sup>. We compare the methods under two budget configurations where the target rank is set to 2 and 4 respectively, resulting in the total trainable parameters being 0.3M and 0.6M (of 86M). Full fine-tuning and LoRA applied to all modules are also added as baselines.

### C. Training and Evaluation

Our experiments are based on the official hyperparameters of AdaLoRA<sup>3</sup> which are optimal when training with Adam. For IVON, the learning rate is set to 0.5 for MRPC and RTE and 0.4 for the rest. Same as Adam, a warm-up stage and the linear decay learning rate schedule are adopted. We found that IVON generally converges slower than Adam at the beginning of training, therefore requiring a much higher learning rate during warm-up for good results especially on small datasets.

<sup>2</sup><https://github.com/team-approx-bayes/ivon>

<sup>3</sup><https://github.com/QingruZhang/AdaLoRA>

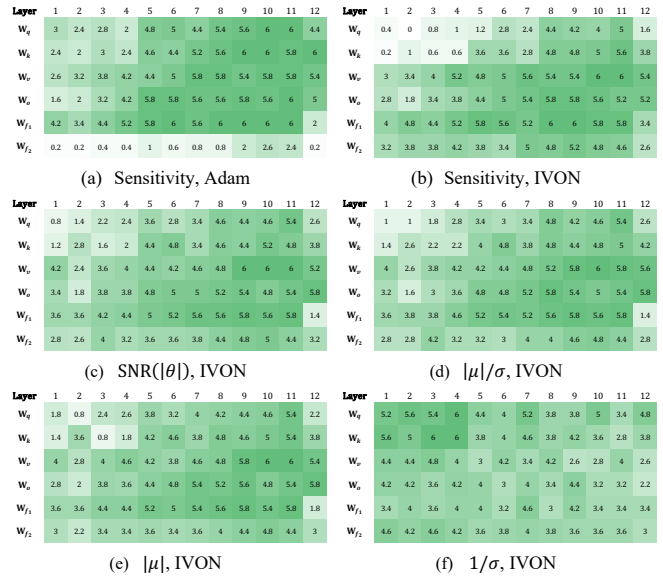


Fig. 1. Comparison of rank distributions after fine-tuning DeBERTaV3-base on MNLI, with deeper colors indicating higher ranks. Results are averaged across five runs with different random seeds.  $W_q$ ,  $W_k$ ,  $W_v$ ,  $W_o$ : weights of the query, key, value, output layers of attention;  $W_{f_1}$ ,  $W_{f_2}$ : weights of the feed-forward layers.

As a result, for COLA, STS-B, MRPC, and RTE, we use a higher learning rate of 2.0 in the warm-up stage and return to the normal learning rate afterwards. For evaluation, we use the best-performing model on the validation set. The results are averaged across five runs with different random seeds.

## V. RESULTS AND ANALYSES

The main results are shown in Table I. For MNLI, the “matched” validation set was used for evaluation. Note that we sort the tasks according to dataset sizes and divide them into two groups since we notice that IVON needs extra tricks to ensure good results on small datasets. In general, all PEFT methods outperform full fine-tuning, and AdaLoRA outperforms LoRA. Switching the optimizer from Adam to IVON results in comparable performance, demonstrating that IVON is capable of state-of-the-art performance in PEFT. We further elaborate our findings from the following perspectives.

### A. Comparison of Importance Scores

Both  $\text{SNR}(\theta)$  and  $\text{SNR}(|\theta|)$  outperform the sensitivity when using IVON, and at least one of the SNR metrics outperforms or ties with the original AdaLoRA with Adam. However, there is no clear winner between the two SNR metrics. This could be explained by the fact that the sparsity level in the AdaLoRA case is not high (only 1/3 of the initial ranks are pruned), and that it is the SVD triplet that is pruned as a parameter group, thus the performance difference between the two metrics is not properly reflected in such a setting. Interestingly, the magnitude outperforms the sensitivity and one of the SNR metrics especially on small datasets. The metric was not experimented in [3]. On the one hand, this

TABLE I  
MAIN RESULTS. THE NUMBER IN MODEL NAMES REFERS TO THE TARGET RANK. THE **BEST** AND THE SECOND BEST RESULTS ARE MARKED.

Model	Optimizer	Criterion	Group 1				Group 2				Group 1	Group 2	All
			MNLI 393k Acc(m)	QQP 364k Acc	QNLI 108k Acc	SST-2 67k Acc	COLA 8.5k Mcc	STS-B 7k Corr	MRPC 3.7k Acc	RTE 2.5k Acc			
Full FT	Adam	None	89.89	92.50	94.03	95.73	69.77	91.06	89.75	84.84	93.04	83.86	88.45
LoRA <sub>2</sub>	Adam	$r = 2$	89.92	<b>91.70</b>	93.97	95.30	69.07	90.89	90.15	87.29	92.72	84.35	88.54
AdaLoRA <sub>2</sub>	Adam	Sensitivity	90.40	91.66	<b>94.49</b>	<u>95.67</u>	<u>70.78</u>	91.47	90.39	87.00	<u>93.05</u>	84.91	88.98
AdaLoRA <sub>2</sub>	IVON	Sensitivity	90.44	91.69	94.36	<u>95.55</u>	69.65	91.89	90.25	87.94	93.01	84.93	88.97
AdaLoRA <sub>2</sub>	IVON	SNR( $ \theta $ )	90.44	91.68	94.40	<b>95.80</b>	70.28	<b>92.04</b>	90.10	88.16	<b>93.08</b>	85.15	89.11
AdaLoRA <sub>2</sub>	IVON	$ \mu /\sigma$	<b>90.46</b>	<b>91.70</b>	<u>94.33</u>	95.62	70.63	91.90	<b>90.83</b>	<b>88.38</b>	93.03	<b>85.43</b>	<b>89.23</b>
AdaLoRA <sub>2</sub>	IVON	$ \mu $	90.42	<b>91.70</b>	94.33	95.57	<b>70.91</b>	91.99	90.64	87.87	93.01	<u>85.35</u>	89.18
AdaLoRA <sub>2</sub>	IVON	$1/\sigma$	90.37	91.32	94.30	95.62	69.35	91.91	90.59	<u>88.16</u>	92.90	<u>85.00</u>	88.95
LoRA <sub>4</sub>	Adam	$r = 4$	89.68	<b>92.03</b>	94.13	95.32	69.58	90.69	<b>90.25</b>	87.08	92.79	84.40	88.59
AdaLoRA <sub>4</sub>	Adam	Sensitivity	90.52	91.91	<b>94.56</b>	95.78	<b>69.85</b>	91.68	<b>90.25</b>	88.16	<b>93.19</b>	84.98	<b>89.09</b>
AdaLoRA <sub>4</sub>	IVON	Sensitivity	90.54	91.74	94.45	<b>95.80</b>	69.41	<b>92.03</b>	89.90	88.09	93.13	84.86	89.00
AdaLoRA <sub>4</sub>	IVON	SNR( $ \theta $ )	90.59	91.78	94.43	95.67	69.83	91.97	89.95	<b>88.52</b>	93.11	<b>85.07</b>	<b>89.09</b>
AdaLoRA <sub>4</sub>	IVON	$ \mu /\sigma$	<b>90.60</b>	91.77	<u>94.49</u>	95.69	<u>69.32</u>	<u>92.00</u>	89.95	<u>88.30</u>	<u>93.14</u>	84.89	89.01
AdaLoRA <sub>4</sub>	IVON	$ \mu $	90.56	91.77	94.40	95.55	69.72	91.98	<b>90.25</b>	88.23	<u>93.07</u>	<u>85.05</u>	89.06
AdaLoRA <sub>4</sub>	IVON	$1/\sigma$	90.48	91.22	94.32	95.62	69.58	91.93	<u>90.20</u>	87.51	92.91	<u>84.80</u>	<u>88.86</u>

demonstrates the effectiveness of magnitude pruning; on the other hand, this is probably because the sensitivity or the variance needs more iterations to be estimated accurately given their smoothing nature. Using the variance alone performs the worst among all metrics, however, it still outperforms LoRA with a fixed rank, indicating that the uncertainty of parameters does correlate with the importance.

### B. Visualizing Final Rank Distributions

Figure 1 shows the final rank distributions of different methods after fine-tuning the model on MNLI. An obvious difference between Adam and IVON using the sensitivity can be observed comparing (a) and (b), indicating a distinction between the training dynamics of the two optimizers. The distributions of the two SNR metrics (c, d) and the magnitude (e) resemble that of the sensitivity with IVON, which corroborates with quantified results. Unlike the magnitude (e), the variance (f) shows an evenly-distributed pattern. This confirms that the magnitude plays a determining role in reflecting the importance of parameters.

### C. Speed

The variance of the parameter is inferred inherently in IVON, thus the SNR does not require the extra computation of the weight-gradient product of the sensitivity during fine-tuning. On an NVIDIA H100, using the SNR with IVON brings a 10% speed up compared to using the sensitivity with Adam, despite the IVON itself being 2% slower than Adam with other conditions kept the same.

### D. A Bayesian Interpretation of Sensitivity

The similarity in performance and the rank distribution between the sensitivity and the SNR suggests a close relationship between them. A closer examination of the underlying theory reveals that sensitivity is, in fact, aligned with the principles of SNR. Specifically, in IVON, the standard

deviation  $\sigma$  is calculated as  $\sigma = 1/\sqrt{\lambda(\mathbf{h} + \delta)}$ , where  $\mathbf{h}$  is the diagonal Hessian,  $\lambda$  is the effective sample size, and  $\delta$  is a weight decay term. Notably,  $\mathbf{h}$  can be approximated by the expected square gradient on the training data [24],  $\mathbf{h} \approx \mathbb{E}_{\mathcal{D}}[(\nabla_{\theta}\ell)^2]$ , also known as the diagonal of the expected Fisher information matrix (FIM). Consequently, the inverse of the standard deviation,  $1/\sigma$ , in the context of SNR, is akin to the root mean square of the gradient  $\sqrt{\mathbb{E}_{\mathcal{D}}[(\nabla_{\theta}\ell)^2]}$ , and therefore analogous to the magnitude of the gradient  $|\nabla_{\theta}\ell|$ . This implies that the sensitivity  $|\theta\nabla_{\theta}\ell|$  has the component  $|\nabla_{\theta}\ell|$  acting as an uncertainty measure analogous to  $1/\sigma$  in SNR, thereby providing a Bayesian interpretation of the sensitivity as an importance metric. These findings resonate with the comment in [5] that the sensitivity has connections with the FIM. Note that both methods adopt exponential moving average smoothing to compute the global value of the corresponding metric during training. The main difference is that the smoothing is applied to the magnitude of the gradient-weight product in AdaLoRA, while the SNR is computed using the global Hessian tracked by IVON.

## VI. CONCLUSIONS

In this study, we developed a Bayesian alternative to AdaLoRA, leveraging the signal-to-noise ratio as the importance score with the IVON optimizer. By comparing the performance of different importance metrics, we demonstrated that this Bayesian approach not only matched or surpassed the performance of using the sensitivity-based importance metric on the GLUE benchmark, but was also a faster alternative to the original AdaLoRA with Adam. The theoretical analysis uncovered a significant link between these two metrics, offering a Bayesian perspective on the efficacy of the heuristic sensitivity-based metric as an importance score. Furthermore, our results suggested that the magnitude, rather than the variance, served as the key indicator of the importance of parameters.

## REFERENCES

- [1] N. Ding *et al.*, “Parameter-efficient fine-tuning of large-scale pre-trained language models,” *Nat. Mac. Intell.*, vol. 5, no. 3, pp. 220–235, 2023.
- [2] E. J. Hu *et al.*, “Lora: Low-rank adaptation of large language models,” in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.
- [3] Q. Zhang *et al.*, “Adaptive budget allocation for parameter-efficient fine-tuning,” in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.
- [4] L. Theis, I. Korshunova, A. Tejani, and F. Huszár, “Faster gaze prediction with dense networks and fisher pruning,” *CoRR*, vol. abs/1801.05787, 2018.
- [5] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, “Importance estimation for neural network pruning,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. IEEE, 2019, pp. 11 264–11 272.
- [6] A. Graves, “Practical variational inference for neural networks,” in *Annual Conference on Neural Information Processing Systems 2011. 12-14 December 2011, Granada, Spain*, 2011, pp. 2348–2356.
- [7] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural network,” in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, vol. 37. JMLR.org, 2015, pp. 1613–1622.
- [8] K. Neklyudov, D. Molchanov, A. Ashukha, and D. P. Vetrov, “Structured bayesian pruning via log-normal multiplicative noise,” in *Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 6775–6784.
- [9] Y. Shen *et al.*, “Variational learning is effective for large deep networks,” in *Proceedings of the 41st International Conference on Machine Learning*, vol. 235. PMLR, 21–27 Jul 2024, pp. 44 665–44 686.
- [10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*, 2015.
- [11] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.
- [12] F. Zhang, L. Li, J. Chen, Z. Jiang, B. Wang, and Y. Qian, “Incelora: Incremental parameter allocation method for parameter-efficient fine-tuning,” *CoRR*, vol. abs/2308.12043, 2023.
- [13] H. Wang, T. Liu, T. Zhao, and J. Gao, “Roselora: Row and column-wise sparse low-rank adaptation of pre-trained language model for knowledge editing and fine-tuning,” *CoRR*, vol. abs/2406.10777, 2024.
- [14] Y. Mao, K. Huang, C. Guan, G. Bao, F. Mo, and J. Xu, “Dora: Enhancing parameter-efficient fine-tuning with dynamic rank distribution,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 2024, pp. 11 662–11 675.
- [15] M. Nikdan, S. Tabesh, and D. Alistarh, “Rosa: Accurate parameter-efficient fine-tuning via robust adaptation,” *CoRR*, vol. abs/2401.04679, 2024.
- [16] D. Guo, A. M. Rush, and Y. Kim, “Parameter-efficient transfer learning with diff pruning,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 2021, pp. 4884–4896.
- [17] N. Ding *et al.*, “Sparse low-rank adaptation of pre-trained language models,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 2023, pp. 4133–4145.
- [18] J. Li, Z. Miao, Q. Qiu, and R. Zhang, “Training bayesian neural networks with sparse subspace variational inference,” in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.
- [19] S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural network,” in *Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 1135–1143.
- [20] K. Osawa *et al.*, “Practical deep learning with bayesian principles,” in *Annual Conference on Neural Information Processing Systems 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019, pp. 4289–4301.
- [21] P. He, J. Gao, and W. Chen, “Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing,” in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.
- [22] T. Wolf *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*, 2020, pp. 38–45.
- [23] S. Mangrulkar *et al.*, “Pefit: State-of-the-art parameter-efficient fine-tuning methods,” <https://github.com/huggingface/peft>, 2022.
- [24] J. Kirkpatrick *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.