Technical Appendix

A. Hyperparamters

RoBERTa-large experiments The memory budget for all the RoBERTa-large experiments is a single RTX 4090 GPU with 24GB memory. The number of layers optimized by ZO and FO optimizers are both 12. The ratio of trainable parameters belonging to the FO optimizer is 50% for LoHO-Adam$_{intra}$. The number of training steps is 20,000 for LoHO-SGD$_{inter}$ and 10,000 steps for LoHO-Adam$_{inter}$. We evaluated the model on the validation set every 10% of the total training steps, saving the best validation checkpoint to evaluate on the test set. Following the MeZO work (Malladi et al. 2024), the random seeds used in our experiments are $\{42, 21, 13, 87, 100\}$. Other hyperparamters for the RoBERTa-large experiments are shown in Table 1.

Table 1: Hyperparamters for RoBERTa-large experiments. HP denotes hyperparameter, and lr denotes learning rate.

Method	HP	SST-2	RTE	MNLI	SNLI
LoHO-SGD$_{inter}$	batch size	64			
	lr of ZO	1e-6			
	lr of FO	2e-5			
	ϵ	1e-3			
LoHO-Adam$_{inter}$	batch size	64			
	lr of ZO	1e-12			
	lr of FO	1e-5	1e-5	5e-5	5e-5
	ϵ	1e-3			
LoHO-Adam$_{intra}$	batch size	64			
	lr of ZO	1e-12			
	lr of FO	5e-5			
	ϵ	1e-3			

OPT experiments The memory budget for the OPT-13B and OPT-30B experiments is a single A800 GPU with 80GB memory. As the average sequence lengths of the datasets are different, their maximum numbers of layers that can be optimized by the FO optimizer are also different. We evaluated the model on the validation set every 10% of the total training steps, saving the best validation checkpoint to evaluate on the test set. The ratio of trainable parameters belonging to the FO optimizer is 50% and 20% for LoHO-Adam$_{intra}$ on OPT-13B and OPT-30B respectively. We present other hyperparameters for the OPT-13B experiments and OPT-30B experiments in Table 2 and Table 3 respectively.

B. More Results of Convergence Rate Comparison

Here, we present the convergence rate comparison on the other two datasets to further support the effectiveness of our method, as shown in Figure 1. For better presentation, we only show the losses of the first 50,000 steps of MeZO, although MeZO is actually fine-tuned with 100,000 steps. The results consistently show the faster convergence rate and shorter training time of our methods compared with the MeZO method. In addition, the convergence rate of LoHO-Adam$_{inter}$ is faster than LoHO-SGD$_{inter}$.

Table 2: Hyperparamters for OPT-13B experiments.

Method	HP	RTE	CB	BoolQ	WIC	MultiRC
LoHO-SGD$_{inter}$	batch size	16				
	lr of ZO	1e-7				
	lr of FO	1e-4	1e-4	1e-4	1e-4	3e-3
	ϵ	1e-3				
	#FO layers	24	12	6	24	2
	training steps	10,000	12,000	20,000	10,000	10,000
LoHO-Adam$_{inter}$	batch size	16				
	lr of ZO	1e-12				
	lr of FO	5e-5	1e-5	1e-5	5e-7	5e-5
	ϵ	1e-3				
	#FO layers	20	10	6	20	2
	training steps	1000	1000	500	1000	500
LoHO-Adam$_{intra}$	batch size	16				
	lr of ZO	1e-12				
	lr of FO	5e-5	1e-5	1e-5	1e-6	1e-5
	ϵ	1e-3				
	#FO layers	20	10	6	20	2
	training steps	1000	1000	500	1000	500

Table 3: Hyperparamters for OPT-30B experiments.

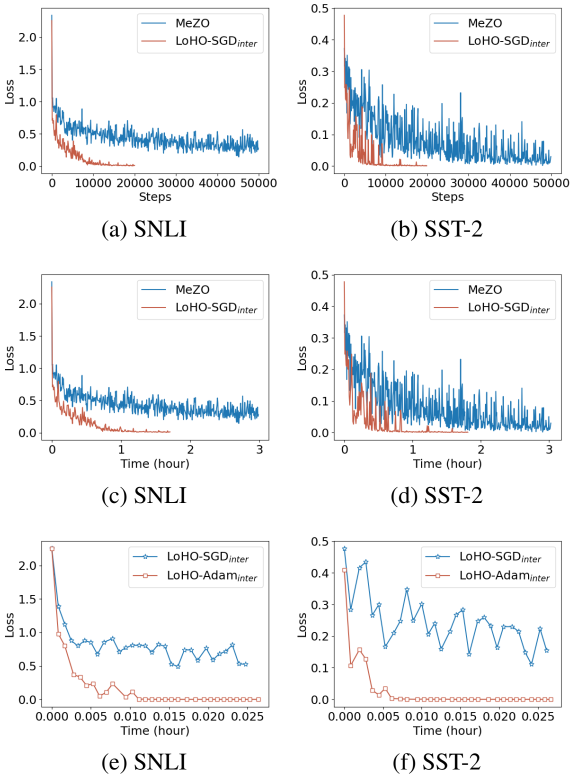
Method	HP	RTE	WIC	BoolQ
LoHO-SGD$_{inter}$	batch size	8		
	lr of ZO	1e-7	1e-12	1e-7
	lr of FO	1e-3	3e-4	1e-7
	ϵ	1e-3		
	#FO layers	8	12	1
	training steps	15,000	10,000	20,000
LoHO-Adam$_{inter}$	batch size	8		
	lr of ZO	1e-12	1e-12	1e-7
	lr of FO	2e-4	1e-6	1e-7
	ϵ	1e-3		
	#FO layers	4	4	1
	training steps	1000	1000	20,000
LoHO-Adam$_{intra}$	batch size	8		
	lr of ZO	1e-12	1e-12	1e-7
	lr of FO	3e-4	5e-6	1e-7
	ϵ	1e-3		
	#FO layers	4	4	1
	training steps	1000	1000	10,000

C. Related Work on Second-order Optimization

Second-order optimization methods use second-order information to guide the gradient descent. For example, Schaul, Zhang, and LeCun (2013) proposed to employ the diagonal Hessian matrix to adaptively adjust the learning rate of SGD during training. Pascanu and Bengio (2013) utilize the second-order information and the manifold information to extend the natural gradient descent. Different from using a diagonal approximation, they use a truncated Newton approach or inverting the metric matrix. AdaHessian (Yao et al. 2021) utilizes an estimated diagonal of the Hessian matrix, which is combined with spatial averaging and momentum to precondition the gradient vector. Our proposed techniques in this paper can be extended to those optimizers. Our future work aims to study the hybrids between different order optimizers.

D. Performance of LoHO with PEFT

The proposed LoHO can work with both full fine-tuning and parameter-efficient fine-tuning (PEFT) methods. Here, we present the performance comparison of MeZO and LoHO-



(a) SNLI (b) SST-2

(c) SNLI (d) SST-2

(e) SNLI (f) SST-2

Figure 1: Comparison of convergence rate on the SNLI and SST-2 datasets using RoBERTa-large as the backbone.

Adam$_{inter}$ combining with LoRA (Hu et al. 2022) using OPT-13B as the backbone. In LoRA, the backbone model is frozen, and only the newly added low-rank modules are trainable. The experimental results are presented in Table 4. We can see that LoHO-Adam$_{inter}$ (LoRA) outperforms MeZO (LoRA) on most datasets. In addition to LoRA, our LoHO can work with other PEFT methods such as prefix-tuning (Li and Liang 2021). We leave this in the future work.

Table 4: Performance comparison between MeZO (LoRA) and LoHO-Adam$_{inter}$ (LoRA) on OPT-13B.

Method	RTE	CB	BoolQ	WIC	MultiRC
MeZO (LoRA)	67.9	66.1	73.8	59.7	61.5
LoHO-Adam$_{inter}$ (LoRA)	**75.5**	**78.6**	67.4	**66.0**	**66.6**

References

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. 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*, 4582–4597. Association for Computational Linguistics.

Malladi, S.; Gao, T.; Nichani, E.; Damian, A.; Lee, J. D.; Chen, D.; and Arora, S. 2024. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36.

Pascanu, R.; and Bengio, Y. 2013. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*.

Schaul, T.; Zhang, S.; and LeCun, Y. 2013. No more pesky learning rates. In *International conference on machine learning*, 343–351. PMLR.

Yao, Z.; Gholami, A.; Shen, S.; Mustafa, M.; Keutzer, K.; and Mahoney, M. 2021. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, 10665–10673.