Model Fusion through Bayesian Optimization in Language Model Fine-Tuning

Chaeyun Jang[1] **Jungtaek Kim**[2] **Hyungi Lee**[1] **Juho Lee**[1]
[1]KAIST [2]University of Pittsburgh
{jcy9911, lhk2708, juholee}@kaist.ac.kr, jungtaek.kim@pitt.edu

Abstract

Fine-tuning a pretrained model for downstream tasks is a widely-adopted technique, which is known for its adaptability and reliability across various domains. Despite its conceptual simplicity, fine-tuning entails several engineering choices such as the selection of hyperparameters and the determination of checkpoints from an optimization trajectory. To tackle the difficulty of choosing the best model among multiple ones obtained from those choices, one of the effective solutions is model fusion, which combines multiple models on a parameter space. On the other hand, we observe a large discrepancy between loss and actual metric values where a loss is often used to pick out models to fuse. While the loss is generally differentiable and thus easier to optimize, the consideration of metrics is often a preferable goal to improve model performance. In response, we present a novel model fusion technique, optimizing a desired metric as well as a loss using Bayesian Optimization (BO). Moreover, combining the multi-objective BO into model fusion, we devise a bilevel framework, composed of BO models for hyperparameter optimization and model fusion. Experiments across various downstream tasks validate decent performance improvements achieved using our BO-based model fusion method.

1 Introduction

A Natural Language Processing (NLP) domain has significantly been advantaged by pre-trained Transformer-based Masked Language Models (MLMs) such as BERT [8] and RoBERTa [21], and large-scale models like GPT [26] and Llama [34]. Typically, these models are fine-tuned on a supervised downstream dataset for a few epochs. However, this process requires careful tuning of several hyperparameters such as learning rate and weight-decay coefficient. Additionally, selecting the optimal checkpoint for the final model, usually based on validation performance during multiple fine-tuning runs, is crucial, although it does not always ensure optimal generalization on unseen data.

An effective strategy for seeking a high-performing model from multiple candidates is construction of an ensemble of models. However, traditional ensemble methods come with drawbacks, including increased memory usage and time complexity, which scales linearly with the number of models involved. These issues are particularly pertinent for large language models with a large number of parameters. An alternative approach is model fusion, where multiple models are aggregated in the parameter space to produce a single proficient model. One of the simplest forms, known as Stochastic Weight Averaging (SWA) [16], involves taking the average of model parameters obtained during the optimization process. Despite its simplicity, SWA and its variants have proven successful across various tasks, especially in computer vision [16, 23, 5, 25]. A recent advancement in this field is the concept of Model Soups, introduced by Wortsman et al. [37]. In this approach, models from multiple fine-tuning runs with different hyperparameters are weight-averaged in order to create a powerful model that outperforms not only individual models but also ensemble models.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

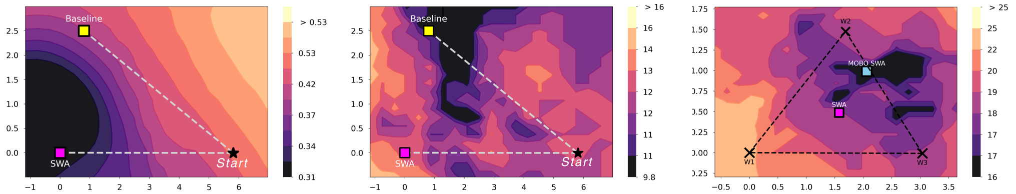

Figure 1: Visualization of a loss landscape over parameters (left) and a metric landscape over parameters (middle) of the RoBERTa-base on the MRPC validation set. *Start* and *End* respectively denote the first and the last members of SWA. Here, the End can be considered as the fine-tuned weight without averaging strategies. Different from the loss landscape, the generalization performance of SWA with regard to the metric landscape suffers from the complex and misaligned surface. Visualization of the metric landscape (right) of the RoBERTa-base on the RTE validation set. w_1, w_2, w_3 indicate the members of the SWA. MOBO-SWA identifies a superior generalization area compared to SWA.

The effectiveness of model fusion has predominantly been explored in the visual domain. For instance, while Model Soups have showcased significant improvements in image classification, they have not demonstrated superiority over individual best models in the NLP tasks [37], which is reaffirmed through our own empirical validation. The mechanism behind the simple averaging methods like SWA lies in their ability to encourage averaged weights to locate on the flatter area near local optima [16, 14]. Consequently, the models fused with simple averaging would be located at the center of such flat minima, and thus exhibit strong generalization properties in terms of the loss function. Unfortunately, for the language models, as we empirically demonstrate, there is a substantial discrepancy between the loss and evaluation metric, so that a flat loss minimum reached by SWA does not necessarily correspond to a flat metric minimum, making a simple averaging method fail to find a good solution.

In this paper we introduce a novel model fusion method, which is dubbed Bilevel-BO-SWA, designed specifically for fine-tuning language models. We start by illustrating that existing approaches are not well-suited for our context. In response to this challenge, we propose constructing a fused model as a weighted combination of individual models, with the goal of maximizing a target metric. Since evaluation metrics in NLP are typically non-differentiable, we employ BO [2, 12], a black-box optimization technique. Our application of BO to this problem is noteworthy for two main reasons:

- **Multi-objective BO**: instead of running BO solely with a single target evaluation metric as an objective, we employ multi-objective BO that considers both metric and loss functions for optimization. Despite the disconnect between loss and metric values, we find that incorporating loss values can still serve as useful guidance, enhancing the efficiency of BO.
- **Bilevel model fusion**: we devise our model fusion process as a bilevel procedure. Here, the outer BO is for optimizing the hyperparameters involved in language model fine-tuning. The inner BO is for the model fusion procedure we propose. The objective of outer BO is to maximize the gain from the inner BO, that is, to find hyperparameters leading to the best fused model with inner BO.

We demonstrate the effectiveness of our methods on several NLP tasks with ROBERTa and the interesting properties of our proposed algorithm through diverse ablation studies.

2 Empirical Analysis on Uniform Weight Averaging

The success of uniform weight averaging (e.g., SWA and Model Soups) in image classification tasks is grounded on the flatness of a loss landscape. Through uniform weight averaging, it is possible to venture into a flat minima on the loss landscape, accordingly, achieving effective generalization performance on a test dataset. This generalization effect is equally observed in the metric landscape, due to the similarity between the loss landscape and the metric landscape in image classification tasks. However, the domain of language modeling, characterized by semantic, morphosyntactic, and pragmatic nuances, necessitates the evaluation of generalization performance across a wide variety of tasks and metrics [9] which are not exactly aligned with a training loss. These metrics often form more complex and less flat surfaces compared to the loss.

The left and middle panels of Figure 1 visually demonstrate that while SWA can reach high generalization performance based on the loss function, it poorly performs with respect to the metric (F1 score) compared to the fine-tuned weights without averaging strategy; refer to Appendix C for detailed numerical assessment comparing the performance of SWA and the naïve fine-tuned model. However, the right panel of Figure 1 shows that even though the naïve uniform averaging of three weight points degrades the metric performance, better points in terms of higher metric values exist in the convex set of the three weight points. The empirical results we observe above, which are caused by the complex and misaligned surface, motivate the need to seek the optimal combination of averaging weights based on the metric. This is in contrast to the previous findings in vision tasks [37] which argue minimal performance difference between the optimized weights and the uniform weights.

3 Model Fusion through Bayesian Optimization

Through this section, we denote our target language model as $\mathcal{M}(\boldsymbol{\theta}(\boldsymbol{\lambda}))$ where $\boldsymbol{\lambda}$ is a hyperparameter vector that is utilized when fine-tuning the model and $\boldsymbol{\theta}(\boldsymbol{\lambda})$ is model parameters trained with $\boldsymbol{\lambda}$. As discussed in the previous sections, our ultimate goal is to find a single proficient model $\mathcal{M}(\bar{\boldsymbol{\theta}}(\boldsymbol{\lambda}))$ by aggregating the last k models $\mathcal{M}(\boldsymbol{\theta}_{T-k+1}(\boldsymbol{\lambda})), \mathcal{M}(\boldsymbol{\theta}_{T-k+2}(\boldsymbol{\lambda})), \ldots, \mathcal{M}(\boldsymbol{\theta}_{T}(\boldsymbol{\lambda}))$ from a single training trajectory with T epochs:

$$\bar{\boldsymbol{\theta}}(\boldsymbol{w}, \boldsymbol{\lambda}) = w_1 \boldsymbol{\theta}_{T-k+1}(\boldsymbol{\lambda}) + w_2 \boldsymbol{\theta}_{T-k+2}(\boldsymbol{\lambda}) + \cdots + w_k \boldsymbol{\theta}_{T}(\boldsymbol{\lambda}), \tag{1}$$

where combination coefficients $w_1, w_2, \ldots w_k \in [0, 1]$ subject to $\sum_{i=1}^{k} w_k = 1$ and $\boldsymbol{w} := [w_1, \ldots, w_k]$. $\boldsymbol{\theta}_i(\boldsymbol{\lambda})$ indicates a parameter checkpoint after completing i^{th} training epoch within a single training trajectory employing the hyperparameters $\boldsymbol{\lambda}$. To measure the performance of $\mathcal{M}(\bar{\boldsymbol{\theta}}(\boldsymbol{w}, \boldsymbol{\lambda}))$, we can use the following performance measures: (i) *loss* and (ii) *metric*. A metric f_{metric} represents the performance measure of our desired task albeit a non-differentiable function, while a loss f_{loss} is generally differentiable but the discrepancy exists. Notably, f_{metric} is a function that produces a task-specific performance for an input model $\mathcal{M}$ on a validation set and $\boldsymbol{w}$.

Our method employs a bilevel optimization approach, where we separately optimize $\boldsymbol{w}$ and $\boldsymbol{\lambda}$. Eventually, this process involves two distinct BO procedures: BO for hyperparameter optimization and multi-objective BO for combination coefficients optimization. Note that we assume that both f_{loss} and f_{metric} are solved by minimizing themselves.

3.1 Multi-Objective Bayesian Optimization for Model Fusion

Optimal combination coefficients $\boldsymbol{w}$ can be selected by considering either the loss or the metric. However, unlike the optimization process of $\boldsymbol{\lambda}$, we take into account both the loss and the metric for the optimization of $\boldsymbol{w}$. Since we simultaneously minimize both f_{loss} and f_{metric}, we adopt Multi-Objective Bayesian Optimization (MOBO) to find a Pareto front which is defined as follows:

$$\mathcal{P} = \left\{ \boldsymbol{w}^\dagger \mid \boldsymbol{w}^\dagger = \operatorname*{arg\,min}_{\boldsymbol{w}} \left(f_{\text{loss}}(\mathcal{M}(\bar{\boldsymbol{\theta}}(\boldsymbol{w}, \boldsymbol{\lambda}))), f_{\text{metric}}(\mathcal{M}(\bar{\boldsymbol{\theta}}(\boldsymbol{w}, \boldsymbol{\lambda}))) \right) \right\}. \tag{2}$$

Instead of the use of random scalarization for solving MOBO, we utilize the expected hypervolume improvement strategy, which is described in [11]. The hypervolume, in this context, is defined as a volume size between $\mathcal{P}$ and a reference point $\boldsymbol{w}_0$. This strategy lets a Pareto front place far from the reference point, such that the Pareto front maximizes the expected hypervolume. To optimize the hypervolume improvement objective, we employ the qNEHVI algorithm [7], which is a recent MOBO, algorithm designed for solving the expected hypervolume improvement problem. The right panel of Figure 1 shows that our MOBO method successfully finds the optimal $\boldsymbol{w}$ that yields the improved performance compared to SWA. Refer to Appendix B for the details of our method.

3.2 Bilevel Bayesian Optimization for Model Fusion

The MOBO-based model fusion described above is based on a learning trajectory constructed from a set of hyperparameters $\boldsymbol{\lambda}$. As $\boldsymbol{\lambda}$ itself has a significant impact on the generalization performance of the local minima reached with it, it is crucial to carefully choose the optimal value of $\boldsymbol{\lambda}$. To this end, we formulate the problem of optimizing $\boldsymbol{\lambda}$ as a bilevel optimization, where the inner objective is the MOBO objective defined in (2). The outer BO is then run with the objective $f_{\text{metric}}(\mathcal{M}(\bar{\boldsymbol{\theta}}(\boldsymbol{w}^\dagger, \boldsymbol{\lambda})))$. For

Table 1: **Results on the GLUE dataset using the RoBERTa-base.** Numerical results in boldface and with an underscore indicate the best and the second-best results in the respective datasets, respectively.

Method	RTE	MRPC	CoLA	STS-B	SST-2	Avg.
Fine-tune	74.94 ± 2.28	91.65 ± 0.65	56.34 ± 2.90	89.86 ± 0.16	94.49 ± 0.04	81.46
SWA [16]	77.19 ± 1.04	91.31 ± 1.82	55.05 ± 3.09	**89.89** ± 0.20	94.49 ± 0.08	81.59
Greedy SWA [37]	76.52 ± 1.31	91.84 ± 0.14	56.47 ± 3.20	89.87 ± 0.18	94.36 ± 0.05	81.81
Learned SWA [37]	77.82 ± 3.58	90.62 ± 1.87	59.02 ± 2.60	89.65 ± 0.09	94.19 ± 0.00	82.26
MOBO-SWA (ours)	77.86 ± 0.28	92.05 ± 1.05	58.20 ± 1.72	89.58 ± 0.12	94.55 ± 0.12	82.45
Bilevel-BO-SWA (ours)	**78.43** ± 0.26	**92.38** ± 0.68	**59.21** ± 3.53	89.86 ± 0.01	**94.97** ± 0.08	**82.97**
Best subset (oracle)	80.66 ± 0.52	92.90 ± 0.22	60.01 ± 1.88	89.93 ± 0.20	95.08 ± 0.06	83.72

this outer BO, we utilize Gaussian process (GP) regression [27] and GP upper confidence bound [33] as a surrogate function and an acquisition function, respectively. Refer to Appendix B for the details of how we construct the BO component. To summarize, we create the best fusion model through a two-step process; first, we go through the outer BO for hyperparameter optimization, and then the inner BO for combination coefficients optimization.

4 Experiments

In this section, we present empirical evidence that demonstrates the effectiveness of Bilevel-BO-SWA in the NLP tasks. As competitors to our method, we test four algorithms, each aimed at finding a single high-performing solution: (i) *Fine-tune*: a straightforward fine-tuning method that selects the best-performing checkpoint based on a specified metric; (ii) *SWA*: an optimization technique that averages the model parameters obtained during a training process; (iii) *Greedy SWA*: a modified version of the SWA algorithm, inspired by the Greedy Soup [37]. After the fine-tuning process, we choose coefficients only if the performance improves after averaging with the previously collected coefficients. (iv) *Learned SWA*: a variant of the SWA algorithm, inspired by the Learned Soup [37]. After fine-tuning, we learn the coefficients considering the loss. In addition, we report the best achievable results: (v) *Best Subset*: the oracle on the *test set* where all possible subsets for the uniform averaging are considered. Our aim is to reach the results of the Best Subset. Moreover, we validate two versions of our method: (i) *MOBO-SWA*: the MOBO optimization with fixed hyperparameters; (ii) *Bilevel-BO-SWA*: our bilevel algorithm to optimize both coefficients and hyperparameters. See Appendix B for the details of the downstream datasets and hyperparameter selections.

Table 1 presents the empirical results obtained on the GLUE dataset using the RoBERTa-base model. Our approaches consistently demonstrate enhanced or equivalent performance across all datasets. Notably, the Bilevel-BO-SWA method achieved significantly improved results for the RTE and MRPC datasets, which are on par with the performance of the Best subset. It is worth highlighting that even our computationally efficient algorithm, MOBO-SWA, exhibits improved performance compared to other baseline methods. These results provide strong empirical support for the efficacy of our proposed techniques in effectively navigating the complexities of the metric landscape. Refer to Appendix C to see additional experiment results on various ablation studies.

5 Conclusion

In this paper, we empirically observed that the well-known uniform averaging algorithms underperform on the NLP tasks due to the discrepancy between the loss and metric landscapes. Then, motivated by the aforementioned observation, we proposed a novel BO-based bilevel algorithm for model fusion. Our method utilizes the MOBO and BO frameworks to seek optimal combination coefficients and hyperparameters, respectively. We validated that our proposed method shows improved performance on the GLUE dataset using the RoBERTa-base model, compared to other baseline methods.

Acknowledgments and Disclosure of Funding

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)). JK acknowledges support from the MDS-Rely Center, which is funded by the National Science Foundation's Industry–University Cooperative Research Center (IUCRC) program through awards EEC-2052662 and EEC-2052776.

References

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2/3):235–256, 2002. 10

[2] E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010. 2, 8

[3] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988. 10

[4] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, 2017. 9

[5] J. Cha, S. Chun, K. Lee, H.-C. Cho, S. Park, Y. Lee, and S. Park. SWAD: domain generalization by seeking flat minima. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021. 1

[6] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer, 2005. 8

[7] S. Daulton, M. Balandat, and E. Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 2187–2200, 2021. 3

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, 2019. 1, 8

[9] J. Dodge, S. Gururangan, D. Card, R. Schwartz, and N. A. Smith. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. 2

[10] B. Dolan, C. Brockett, and C. Quirk. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. 9

[11] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006. 3

[12] R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2023. 2, 8

[13] S. Haghanifar, M. McCourt, B. Cheng, J. Wuenschell, P. Ohodnicki, and P. W. Leu. Creating glasswing butterfly-inspired durable antifogging superomniphobic supertransmissive, superclear nanostructured glass through Bayesian learning and optimization. *Materials Horizons*, 6(8): 1632–1642, 2019. 8

[14] H. He, G. Huang, and Y. Yuan. Asymmetric valleys: Beyond sharp and flat local minima. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019. 2

[15] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the International Conference on Learning and Intelligent Optimization (LION)*, pages 507–523, 2011. 8

[16] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018. 1, 2, 4

[17] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998. 8

[18] J. Kim and S. Choi. On uncertainty estimation by tree-based surrogate models in sequential model-based optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4359–4375, 2022. 8

[19] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017. 8

[20] Y. L. Li, T. G. J. Rudner, and A. G. Wilson. A study of Bayesian neural network surrogates for Bayesian optimization. *arXiv preprint arXiv:2305.20028*, 2023. 8

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 1, 8

[22] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 9

[23] W. Maddox, T. Garipov, P. Izmailov, D. Vetrov, and A. G. Wilson. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019. 1

[24] M. Mosbach, M. Andriushchenko, and D. Klakow. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*, 2020. 8

[25] G. Nam, S. Jang, and J. Lee. Decoupled training for long-tailed classification with stochastic representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 1

[26] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *arXiv preprint arXiv:1801.06146*, 2018. 1, 8

[27] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. 4, 8

[28] B. J. Shields, J. Stevens, J. Li, M. Parasram, F. Damani, J. I. M. Alvarado, J. M. Janey, R. P. Adams, and A. G. Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021. 8

[29] L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017. 9

[30] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, pages 2951–2959, 2012. 8

[31] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013. 8

[32] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 4134–4142, 2016. 8

[33] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1015–1022, 2010. 4, 8

[34] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1, 8

[35] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 8

[36] A. Warstadt, A. Singh, and S. R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. 8

[37] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of The 38th International Conference on Machine Learning (ICML 2022)*, 2022. 1, 2, 3, 4, 8

A Related Work

Fine-Tuning for Pre-trained Language Models. Pre-trained Transformer-based MLMs such as BERT [8] and RoBERTa [21], in addition to auto-regressive language models such as GPT [26] and Llama [34], have had a significant impact on the NLP community recently. The standard procedure for the use of such models typically involves training a pre-trained model for a few epochs on a downstream dataset in a supervised fashion. This process is known as fine-tuning. Despite its straightforward concept, fine-tuning requires the need for various engineering decisions including the selection of hyperparameters and the identification of suitable checkpoints within an optimization trajectory. Addressing the challenge of selecting an optimal model among various models derived from these decisions, one effective strategy is model fusion. This approach combines multiple models within a parameter space to formulate a single proficient model.

Model Fusion for Pre-trained Language Models. The cost of fine-tuning language models is significantly high, rendering the straightforward approach of creating Deep Ensemble (DE) [19] from multiple models and discarding the rest inefficient. On the other hand, a weight averaging method emerges as a more feasible approach for model fusion, mitigating the inference cost while retaining the benefits of ensembles. Notably, SWA employs uniform averaging on a single trajectory, manifesting substantial improvements in generalization within image classification tasks. In the case by Wortsman et al. [37], the models obtained from multiple trajectories are sorted in descending order, and the models are greedily selected for participation in uniform averaging. However, as will be highlighted in § 2, the prior uniform weight averaging methods are found to be inadequate in language models. By combining the discovered cause and the advantages of previous fusion methods, we exhibit a generalization effect by identifying the optimal hyperparameters and a subset of SWA members through a metric-driven BO-based model fusion.

Bayesian Optimization. BO [2, 12] is a promising strategy to optimize a black-box function. In particular, if a target objective is costly in terms of function evaluations, BO is more effective than other existing optimization strategies such as grid search and genetic algorithms. Its efficacy has demonstrated in a wide variety of applications such as hyperparameter optimization [30], nanostructured device design [13], and chemical reaction optimization [28]. Briefly introducing, BO sequentially seeks solution candidates by modeling a surrogate function and maximizing an acquisition function. In the BO community, A GP [27] is often employed as a surrogate function but diverse regression models such as Bayesian neural networks [32, 20] and tree-based models [15, 18] can be used. As a choice of acquisition function, expected improvement [17] and GP upper confidence bound [33] are often considered. See the work [2, 12] for the details of BO.

B Experimental Details

B.1 Datasets

The empirical evaluation utilized several benchmark datasets from the General Language Understanding Evaluation (GLUE) suite [35], each highlighting different aspects of language understanding tasks. For datasets such as RTE, MRPC, CoLA, and STS-B, we split the original development set in half, using one half for validation and the other for testing. For SST-2, 1,000 instances were taken from the training set for validation, while the original development set was used for testing. We specifically chose datasets that are known to be relatively challenging to tune and unstable [24]. Additionally, to evaluate performance on a larger dataset, we included SST-2.

RTE. The Recognizing Textual Entailment task [6] mandates the model to ascertain whether a given hypothesis is entailed or contradicted by a corresponding premise, categorizing it as a binary classification quandary.

CoLA and SST-2. The Corpus of Linguistic Acceptability [36] and the Stanford Sentiment Treebank 2 [31] are single-sentence tasks that necessitate the model to adjudicate linguistic acceptability and sentiment resonance, respectively. CoLA engages in a binary classification paradigm to appraise the grammatical acceptability of a sentence, whereas SST-2 entails binary sentiment classification to discern the sentiment polarity.

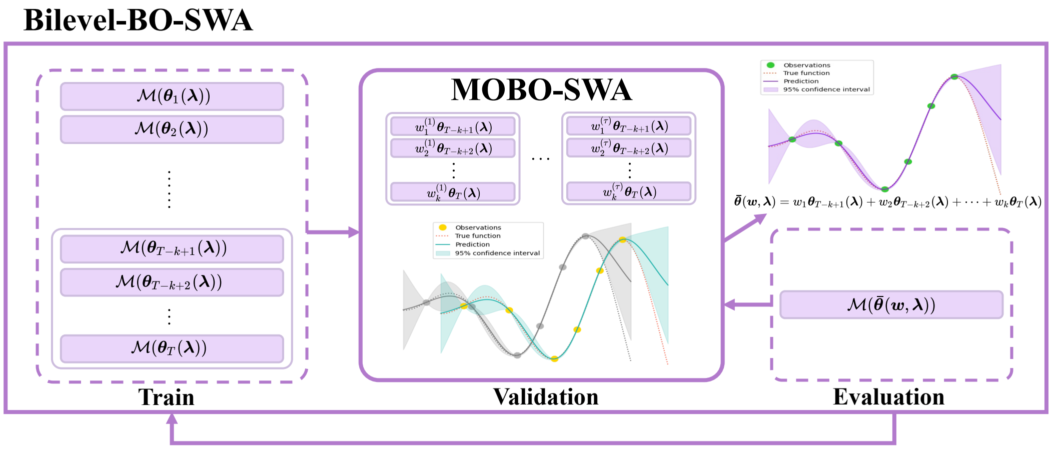
Bilevel-BO-SWA
$\mathcal{M}(\boldsymbol{\theta}_1(\boldsymbol{\lambda}))$
$\mathcal{M}(\boldsymbol{\theta}_2(\boldsymbol{\lambda}))$
$\mathcal{M}(\boldsymbol{\theta}_{T-k+1}(\boldsymbol{\lambda}))$
$\mathcal{M}(\boldsymbol{\theta}_{T-k+2}(\boldsymbol{\lambda}))$
$\mathcal{M}(\boldsymbol{\theta}_T(\boldsymbol{\lambda}))$
Train
MOBO-SWA
$w_1^{(1)}\boldsymbol{\theta}_{T-k+1}(\boldsymbol{\lambda})$
$w_2^{(1)}\boldsymbol{\theta}_{T-k+2}(\boldsymbol{\lambda})$
$w_k^{(1)}\boldsymbol{\theta}_T(\boldsymbol{\lambda})$
$w_1^{(\tau)}\boldsymbol{\theta}_{T-k+1}(\boldsymbol{\lambda})$
$w_2^{(\tau)}\boldsymbol{\theta}_{T-k+2}(\boldsymbol{\lambda})$
$w_k^{(\tau)}\boldsymbol{\theta}_T(\boldsymbol{\lambda})$
Observations
True function
Prediction
95% confidence interval
Validation
Observations
True function
Prediction
95% confidence interval
$\bar{\boldsymbol{\theta}}(\boldsymbol{w},\boldsymbol{\lambda}) = w_1\boldsymbol{\theta}_{T-k+1}(\boldsymbol{\lambda}) + w_2\boldsymbol{\theta}_{T-k+2}(\boldsymbol{\lambda}) + \cdots + w_k\boldsymbol{\theta}_T(\boldsymbol{\lambda})$
$\mathcal{M}(\bar{\boldsymbol{\theta}}(\boldsymbol{w},\boldsymbol{\lambda}))$
Evaluation

Figure 2: **Illustration of the bilevel BO pipeline, named Bilevel-BO-SWA.** It is composed of two components: BO for hyperparameter optimization and multi-objective BO for model fusion via convex combination.

MRPC and STS-B. The Microsoft Research Paraphrase Corpus [10] and the Semantic Textual Similarity Benchmark [4] require the model to quantify semantic equivalence. MRPC orchestrates a binary classification paradigm to determine the paraphrastic nature of sentence pairs, while STS-B mandates scoring the semantic similarity of sentence pairs on a continuous spectrum.

B.2 Experimental Setup

Baselines. In our experimental setup, we choose the pre-trained RoBERTa-base model for fine-tuning masked language models on the GLUE benchmark. In the RoBERTa-base scenario, each task is tuned for 20 epochs, and we select the configuration that exhibits the best metric on the validation dataset. The learning rate is determined through a grid search within [1e-05, 2e-05, 3e-05] , with a designated batch size of 16 for the CoLA, MRPC, RTE, and STS-B tasks, and 32 for the SST-2 task. The learning rate schedule adheres to a linear decay, complemented by a 0.2 warm-up ratio, employing the AdamW optimizer [22]. The outcomes are averaged over a set of seeds, specifically [41, 42, 43, 44]. For SWA, we begin collecting model weights from the point where the baseline converges, specifically from 75% of this point and perform uniform averaging. The SWA scheduler employs a cyclic learning rate scheduler [29] with the optimizer, maintaining the same learning rate. Similar to SWA, Greedy SWA also collects model weights but follows a greedy approach comparable to Model Soups, collecting models only when there is a performance improvement. Learned SWA initially collects models, then mix them using coefficients from a probability simplex. The coefficients are learned by optimizing the loss obtained from this mixed model, selecting the model with the highest validation metric.

Ours. In the case of MOBO SWA, the number of initial points was set equal to the number of weight-averaging members, and iterations were performed five times the amount of the initial points. In the case of the Bilevel-BO-SWA, for the outer BO, the seeds were set to {41, 42, 43, 44}, the learning rate was set within the range of [1e-06 1e-04], weight decay was in the range of [0.0 0.1], and batch sizes were set to [8 32]. A total of 10 iterations were attempted, with each iteration comprising 20 epochs, and 10 SWA members were collected in each iteration. For the inner BO, the same settings as the previous MOBO SWA were used. We present the overall pipeline of Bilevel-BO-SWA as shown in Figure 2.

C Additional Experiments

Discrepency between Loss and Metric. Table 2 again numerically validate that the conventional averaging strategies (i.e. SWA and Model Soup) indeed perform well with the loss function but not with the metric function.

Table 2: **Results on GLUE benchmark for RoBERTa-base.** Evaluation results of SWA and naive fine-tuned model on the RTE, MRPC, SST-2. We used custom validation sets for the evaluation. Here *NLL* is the loss function and *Error rate* is the 1 - *Accuracy* for the RTE and SST-2, and the *F1 score* for the MRPC. The lower value is the better for all the evaluation functions. Please refer to Appendix B to see how we split the custom validation sets.

		Task		
		RTE	MRPC	SST-2
NLL (↓)	Fine-tune	0.97 ± 0.01	0.54 ± 0.02	0.28 ± 0.00
	SWA	**0.87** ± 0.03	**0.53** ± 0.00	**0.22** ± 0.00
Error rate (↓)	Fine-tune	**21.21** ± 0.69	**7.82** ± 0.01	**4.94** ± 0.26
	SWA	21.71 ± 1.47	7.90 ± 0.01	5.16 ± 0.24

Table 3: **Using RoBERTa-base, Performance Analysis of Basic BO on the GLUE Dataset.** When employing BO that focuses solely on a single objective, specifically the metric, it was observed that MOBO-SWA exhibited commendable effectiveness in comparison to BO-SWA, which takes into account both the loss and metric.

Method	RTE	MRPC	CoLA	STS-B	SST-2	Avg.
BO-SWA	77.20 ± 1.97	91.92 ± 0.66	57.56 ± 0.30	**89.63** ± 0.05	94.47 ± 0.15	82.16
MOBO-SWA	**77.86** ± 0.28	**92.05** ± 1.05	**58.20** ± 1.72	89.58 ± 0.12	**94.55** ± 0.12	**82.45**

Table 4: **Comparative Performance Analysis Applying Outer BO on Various Baselines.** The table shows that Bilevel-BO-SWA outperforms other strategies on RTE and MRPC datasets, according to key performance metrics.

	Baseline	SWA	Greedy SWA	Learned SWA	Bilevel-BO-SWA
RTE	77.20	76.68	76.68	78.22	79.60
MRPC	91.41	90.57	90.57	90.03	93.39

Ablation on MOBO and BO. When examining the results presented in Table 3, we assess how our suggested approach, which relies on MOBO, performs in contrast to the method's performance when MOBO is substituted with BO. We utilized a basic BO setting with the Radial Basis Function (RBF) [3] kernel and Upper Confidence Bound (UCB) [1], optimal averaging weights are determined in the validation set based on the metric. This method generally underperforms compared to the MOBO-SWA.

Ablation on the Effectiveness of the Outer BO. Table 4 presents a comparative analysis, focusing on the efficiency of baselines when employing outer BO. The application of outer BO, for hyperparameter optimization, invariably enhances performance across diverse baselines. However, the proposed Bilevel-BO-SWA conspicuously emerges as superior, exhibiting preeminent performance in evaluations across the RTE and MRPC datasets compared to other strategies. The synergy realized through the concurrent application of Bilevel-BO-SWA and outer BO prominently showcases a compelling scenario of cooperative performance enhancement.