Investigating Robustness of Parameter Allocation Strategies Across Scales: A Case Study on the Mix-And-Match Adapter

Peiyang Wen pw1179@nyu.edu
Zhihao Su zs1512@nyu.edu
Yancheng He ych357@nyu.edu
Lake Wang yxw209@nyu.edu

Abstract

Mixture-of-Adapters (MoA), a variant of Parameter Efficient Fine-Tuning (PEFT), has demonstrated notable performance by combining multiple PEFT methods within each transformer layer. However, the optimal parameter allocation ratios between the PEFT components of MoA and the robustness of these ratios across different tunable parameter budgets remain under-explored. In this paper, we conducted a comprehensive ablation study on the Mix-And-Match (MAM) Adapter, a state-of-the-art MoA approach, to investigate whether the default parameter allocation ratio employed by the MAM Apdater is indeed optimal for its default tunable parameter budget. Furthermore, we examined the robustness of the optimal allocation ratio across different parameter budgets. Our experiments revealed that the default allocation ratio is suboptimal and that the optimal ratio varies across tasks and parameter budgets. These findings highlight the importance of task-specific and budget-aware optimization of parameter allocation in MoA.

1 Introduction

Parameter efficient fine tuning (PEFT) has emerged as a powerful technique for efficiently adapting pretrained language models (PLMs) to specific downstream tasks by fine-tuning only a minimal number of additional parameters. Notable PEFT methods include Adapter tuning (Houlsby et al., 2019), Prefix-tuning (Li and Liang, 2021), LoRA (Hu et al., 2021), and BitFit (Zaken et al., 2021). These methods typically inject a small set of tunable parameter modules into the attention and/or feed-forward sub-layers of transformer blocks, while keeping the original PLM parameters frozen during the fine-tuning phase.

Recent studies on Mixture-of-Adapters (MoA), which involve merging multiple state-of-the-art PEFT methods (e.g., Adapter, LoRA, Prefix-tuning), have demonstrated significant performance gains on various downstream tasks compared to using a single PEFT method for fine-tuning (He et al., 2021; Chen et al., 2023; Hu et al., 2023). For instance, He et al. (2021) introduced the Mix-And-Match adapter (MAM Adapter) as the top-performing PEFT architecture that surpassed state-of-the-art single PEFT methods (e.g., Adapter, LoRA, Prefix-tuning, BitFit, Prompt-tuning) on multiple NLP benchmarks and achieved comparable results to full fine-tuning by adding only 0.5% of pretrained parameters. The MAM Adapter incorporates a Prefix-Tuning module in the attention block and a Scaled Parallel Adapter with more allocated parameters in the feed-forward network (FFN) block, effectively combining the strengths of both PEFT methods.

Despite the promising results of the MAM Adapter, the parameter allocation strategies between its PEFT components remain under-explored. He et al. (2021) claimed that optimal performance was observed when allocating more parameters to the parallel adapter module at their specific tunable parameter budget (6.7% of pretrained parameters). However, they did not provide comprehensive evidence of the performance impact of different *parameter allocation ratios*[1], or investigate the generalizability of their findings across various *tunable parameter budgets*[2]. We thus hypothesize that the default parameter allocation ratio of the MAM Adapter is optimal only for the default tunable parameter budget they examined, not for other parameter budgets.

In our work, we performed a comprehensive ablation study on the MAM Adapter to determine if the default parameter allocation ratio is indeed

[1]Controls the allocation of tunable parameters between two PEFT components of the MAM Adapter: Prefix-Tuning and Parallel Adapter. Calculated as the ratio of prefix-tuning params to total tunable params. More details in section 3.

[2]Total tunable parameters for the PEFT method. Measured as the percentage of tunable parameters against all model parameters. More details in section 3.

optimal for the default tunable parameter budget. Additionally, we investigated whether the optimal parameter allocation ratio for the default tunable parameter budget can be robustly generalized to other parameter budgets, thus examining the robustness of parameter allocation strategy across scales. By comparing the experiment results on an extensive set of allocation ratios and tunable budgets, we observed that the default allocation ratio employed by the MAM Adapter is suboptimal and that the optimal ratio varies across tasks and parameter budgets, thus confirming our hypothesis that the default parameter allocation strategy could not be well generalized across different scales. These findings highlight the importance of task-specific and budget-aware optimization of parameter allocation in MAM adapters.

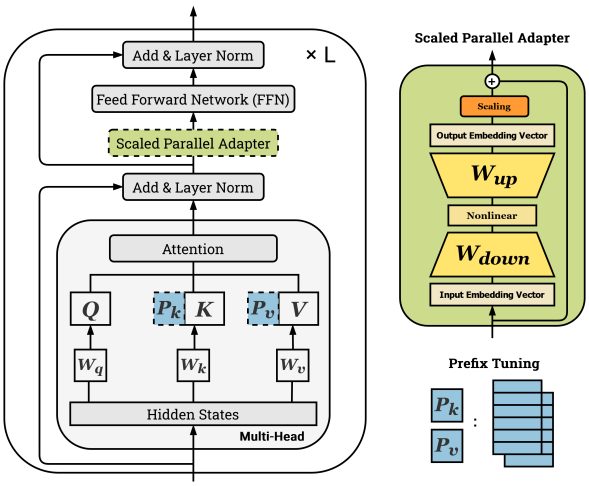

Figure 1: Illustration of the transformer architecture with MAM Adapter inserted. We use blocks with dashed borderlines to represent the two added modules by the MAM Adapter.

2 Related Work

Parameter-efficient fine-tuning (PEFT) methods have gained significant attention in recent years due to their ability to adapt pre-trained language models (PLMs) to downstream tasks with minimal additional parameters. Houlsby et al. (2019) introduced adapter modules, which allow efficient transfer learning by adding only a small number of tunable parameters to each task while maintaining near state-of-the-art performance on benchmarks like GLUE (Wang et al., 2018). Other notable PEFT methods include LoRA (Hu et al., 2021), which adapts large language models by injecting low-rank matrices into the feed-forward network (FFN) block to approximate parameter updates, and prefix-tuning (Li and Liang, 2021), which prepends l tunable prefix tokens to the input or hidden layers in the attention block for adapting PLMs while keeping the original model parameters unchanged.

Building upon these individual PEFT methods, recent studies have explored the potential of combining multiple PEFT techniques to further improve performance. He et al. (He et al., 2021) introduced the Mix-And-Match adapter (MAM Adapter), a top-performing PEFT architecture that combines Prefix-tuning in the attention block and a scaled parallel Adapter in the FFN block, allocating more parameters to the parallel Adapter module to achieve optimal performance. Similarly, Chen et al. (Chen et al., 2023) developed the S4 method, which leverages spindle layer grouping, uniform parameter allocation, and strategic tuning across all model layers. Hu et al. (Hu et al., 2023) presented LLM-Adapter, a framework that integrates adapters into large language models (LLMs) for adapter-based PEFT methods across diverse tasks.

Despite these advancements, the optimal parameter allocation strategies between different PEFT components remain under-explored. Our work addresses this gap by investigating the optimal parameter allocation ratios and their robustness across different tunable parameter budgets in the context of the MAM Adapter, contributing to a deeper understanding of Mixture-of-Adapters PEFT architectures and their performance across NLP downstream tasks.

3 Mix-And-Match Adapter

Here and in Figure 1, we present the MAM Adapter, our target Mixture-of-Adapters (MoA) architecture that integrates a Prefix-tuning module in the attention block and a Scaled Parallel Adapter module in the feed-forward network (FFN) block. The details of each module are elaborated below:

Prefix-Tuning module prepends a sequence of learnable prefix vectors to the keys and values of the multi-head attention mechanism at every layer of the pretrained model. The prefix vectors are the only parameters that are fine-tuned for the downstream task, while the pretrained model remains fixed. In the default MAM Adapter setting, the Prefix-tuning module uses a small bottleneck dimension (e.g., 30) to modify the multi-head attention outputs at each layer.

- Input: The query vector $\mathbf{x} \in \mathbf{R}^d$.

RTE	**Text:** *Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44, according to the Christopher Reeve Foundation.* **Hypothesis:** *Christopher Reeve had an accident.* **Entailment:** `False`
BoolQ	**Passage:** *Barq's – Barq's is an American soft drink. Its brand of root beer is notable for having caffeine. Barq's, created by Edward Barq and bottled since the turn of the 20th century, is owned by the Barq family but bottled by the Coca-Cola Company. It was known as Barq's Famous Olde Tyme Root Beer until 2012.* **Question:** *is barq's root beer a pepsi product* **Answer:** `No`
COPA	**Premise:** *My body cast a shadow over the grass.* **Question:** *What's the CAUSE for this?* **Alternative 1:** *The sun was rising.* **Alternative 2:** *The grass was cut.* **Correct Alternative:** `1`

Table 1: Validation set examples from the three target tasks in SuperGLUE (Wang et al., 2019). **Bold** text represents part of the example format for each task. Text in *italics* is part of the model input. Text in a `Miriam Fixed` font represents the expected model output.

- Output: The modified head attention output is updated to $\mathbf{h} \leftarrow (1-\lambda)\mathbf{h} + \lambda\Delta\mathbf{h}$, where $\Delta\mathbf{h} := \text{softmax}(\mathbf{x}\mathbf{W}_q\mathbf{P}_k{}^T)\mathbf{P}_v$ and λ is a gating variable (He et al., 2021).

Scaled Parallel Adapter module combines the adapter architecture with the scaling and parallel insertion techniques of LoRA. A pair of down and up projection matrices are used to modify the hidden representations in the feed-forward network (FFN) sub-layers. The modification is added to the original hidden representation in parallel, scaled by a learnable scalar parameter. In the default MAM Adapter setting, more parameters are allocated to this module, using a larger bottleneck dimension (e.g., 512) to modify the FFN outputs.

- Input: The FFN input representation $\mathbf{h} \in R^d$.
- Output: The modified FFN output is updated to $\mathbf{h} \leftarrow \mathbf{h} + s \cdot \Delta\mathbf{h}$, where $\Delta\mathbf{h} := \text{ReLU}(\mathbf{h}\mathbf{W}_{down})\mathbf{W}_{up}$ and s is a scaling factor (He et al., 2021).

In addition, we define the *parameter allocation ratio* as the percentage of total tunable parameters allocated to the Prefix-Tuning module. The *tunable parameter budget* is defined as the percentage of pretrained parameters. In the default MAM Adapter setting examined based on BERT-base-uncased model, the Prefix-Tuning module uses a small bottleneck dimension of l=30, while the Scaled Parallel Adapter uses a larger dimension of r=512. This corresponds to allocating around 6% (= *parameter allocation ratio*) of the MAM Adapter's tunable parameters to the Prefix-Tuning module and the remaining 94% to the Scaled Parallel Adapter module. The *tunable parameter budget* in this default setting is equivalent to 9.15% of the pretrained model's parameters.

4 Experimental Setup

Datasets: We evaluate the performance of the MAM Adapter on three downstream tasks from the SuperGLUE benchmark (Wang et al., 2019), as shown in Table 1: (1) RTE (Recognizing Textual Entailment) is a binary entailment benchmark that requires the model to predict whether a hypothesis is entailed by a premise; (2) BoolQ (Boolean Questions) (Clark et al., 2019) is a question answering benchmark where the model must determine if the answer to a yes/no question is contained within a given passage; and (3) COPA (Choice of Plausible Alternatives) (Roemmele et al., 2011) is a causal reasoning benchmark in which the model is given a premise and must choose the most plausible alternative that either causes or is caused by the premise.

Experiment 1: In experiment 1, we analyze the effectiveness of the default parameter allocation ratio for the MAM adapter using the BERT-base-uncased model. This evaluation is focused on the default tunable parameter budget set at 9.15% of the total model parameters, exploring whether the advocated 6% (Prefix-Tuning) versus 94% (Parallel Adapter) split is indeed optimal. This evaluation is conducted across three benchmarks from the SuperGLUE suite—RTE, BoolQ, and COPA. We systematically explore a series of parameter allocation ratios: $r = \{0, 0.06, 0.5, 0.9, 1.0\}$, where $r = 0$ entirely omits the prefix tuning module, and $r = 1$ excludes the parallel adapter, to determine their individual contributions to the model's performance.

This experiment aims to verify whether the default allocation ratio previously suggested in the literature is optimal or if different ratios could improve outcomes, thereby deepening our understanding of the MAM Adapter's adaptability.

Experiment 2: In experiment 2, we assess the robustness of the optimal allocation ratio identified in Experiment 1 (tunable parameter budget = 9.15%) by comparing it to the optimal allocation ratios determined for two additional tunable parameter budgets: 1% and 5%. As in Experiment 1, three benchmarks (RTE, COPA, and BoolQ) are used, and the BERT-base-uncased model is employed as the pre-trained model for fine-tuning. This comparison aims to establish whether the optimal ratio remains consistent across different budget sizes or if the allocation strategy requires adjustments based on the magnitude of the tunable parameter budgets, thus investigating the robustness of the parameter allocation strategy across various scales.

5 Results

Param Budget	Allocation Ratio	RTE	BoolQ	COPA
1%	0%	64.6	71.8	**65.0**
1%	6%*	64.6	73.3	63.0
1%	50%	63.9	73.5	64.0
1%	90%	**65.3**	**73.5**	60.0
1%	100%	63.9	72.9	**65.0**
5%	0%	65.7	**73.4**	**69.0**
5%	6%*	64.6	72.4	63.0
5%	50%	**69.0**	72.8	64.0
5%	90%	62.1	71.0	66.0
5%	100%	63.5	70.0	61.0
9.15%*	0%	**65.7**	**72.9**	66.0
9.15%*	6%*	65.3	71.9	67.0
9.15%*	50%	65.3	71.8	**68.0**
9.15%*	90%	63.5	71.6	66.0
9.15%*	100%	63.9	71.5	64.0

Table 2: Accuracy on the validation set of RTE, BoolQ, and COPA. * indicates the default parameter configuration employed by the MAM Adapter.

Table 2 shows the results for each *tunable parameter budget* and *parameter allocation ratio* combination we examined. For each combination, we extensively fine-tuned the model parameters against validation losses to identify the optimal accuracy result, including batch size, learning rate, epoch number, warm-up ratio, label smoothing, weight decay, and max gradient norm. The scaling factor is set as fixed at 1. For each parameter budget, we highlighted the highest accuracy of every tasks including RTE, BoolQ and COPA.

Regarding Experiment 1, we aimed to examine whether the default allocation ratio of 6% employed by the MAM Adapter is indeed optimal under the default parameter budget of 9.15%. From Figure 2, the results showed that none of the highest accuracy of all three cases occurs when the allocation ratio equals 6%, indicating that the default allocation ratio is not the optimal ratio for the RTE, BoolQ and COPA task using BERT-base-uncased model. However, the general downward trend of accuracy corresponds to the observations from the previous literature of He et al. (He et al., 2021) which states that allocating more parameters to feed-forward network will give better results.

In Experiment 2, we tested whether the optimal allocation ratio remains robust under various tunable parameter budgets by running the models under two additional parameter budgets. As shown in Figure 3, under a 1% tunable parameter budget, the results show that none of the highest accuracy of all three cases occurs at the respective optimal parameter allocation ratio under default tunable parameter budget. Meanwhile, under a 5% tunable parameter budget, only BoolQ achieves the highest accuracy at 0% parameter allocation ratio that corresponds to its case under default tunable parameter budget.

The instability of the optimal allocation ratio across various tunable parameter budgets demonstrated that, in our experiment setting, the parameter allocation ratio is not robust across scales, and the choice of allocation ratio has a strong impact on the final fine-tuning performance. However, we can see from Figure 3 that the accuracy fluctuation at default tunable parameter budget is the lowest, compared with 1% and 5% tunable parameter budgets. Such a small variance indicates that increasing tunable parameter budgets will make the model less sensitive to the parameter allocation ratio.

6 Conclusions

Targeting on examining the robustness of Mixture-of-Adapters, we conducted a comprehensive ablation analysis on the MAM Adapter, evaluating the reported optimal allocation ratio of 6% by (He et al., 2021), and examining the robustness of optimal parameter allocation ratios across different parameter budget levels. The BERT-base-uncased model is employed as the target model and we used three SuperGLUE benchmarks for evaluation: RTE, BoolQ, and COPA with an extensive set of allocation ratios and tunable budgets.

We observed that the default ratio of 6% em-

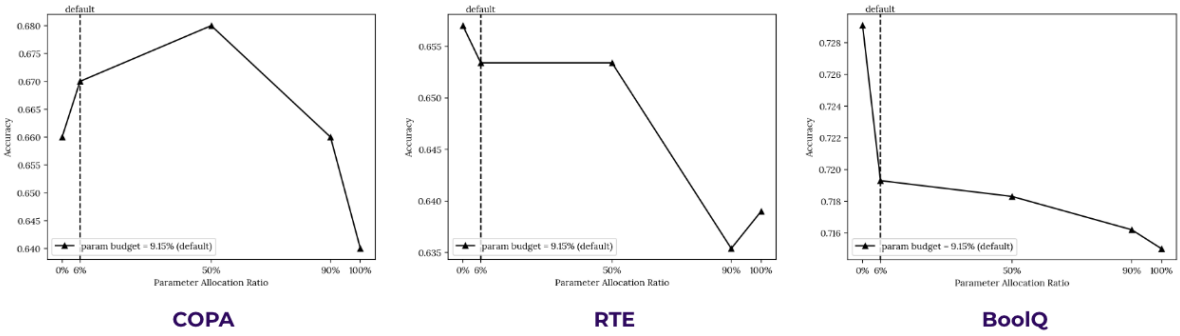

Figure 2: Accuracy results of Experiment 1 under three SUPERGLUE tasks with different Parameter Allocation Ratios and default 9.15% Tunable Parameter Budget of the MAM Adapter.

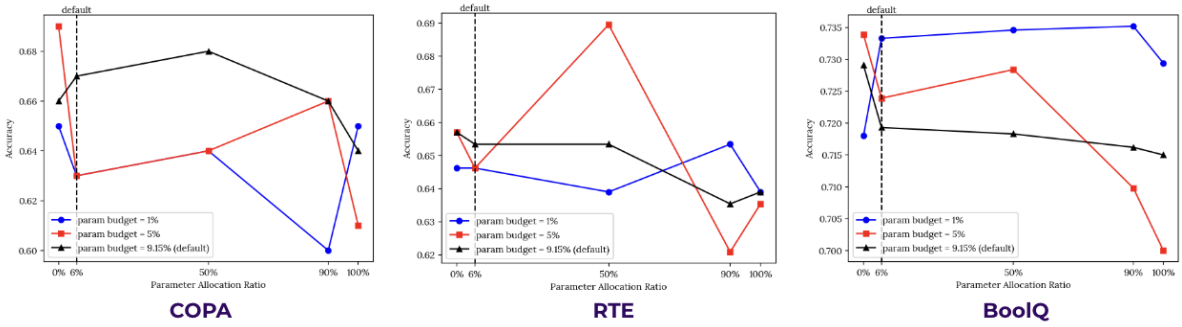

Figure 3: Accuracy results of Experiment 2 under three SUPERGLUE tasks with different Parameter Allocation Ratios and three Tunable Parameter Budgets of the MAM Adapter.

ployed by the MAM Adapter is suboptimal for all three tasks, and the true optimal allocation ratio varies across tasks. In addition, the optimal allocation ratio also varies across different parameter budgets, suggesting that applying a universal allocation ratio may not be appropriate for model tuning across different tasks. These findings supporting our initial hypothesis that using a standalone fixed allocation strategy could lack generalizability across scales. As a result, additional parameter tuning on the parameter allocation ratio for each parameter budget is necessary to maximize the performance of MoA, thus potentially weakening the parameter efficiency advantage of MoA.

7 Limitations

Our study encounters several limitations that may affect the generalizability and robustness of our findings. First, our analysis is confined by a relatively narrow hyperparameter search space. The current results, including the identified optimal parameter allocations, might be influenced by the limited range of batch sizes, learning rates, and other tuning parameters tested. Expanding the search space could potentially unveil more effective parameter configurations that were not discovered in this study.

Second, the inconsistency in the optimal parameter allocation across different budgets highlights the need for further experimentation. Our findings suggest that the effectiveness of PEFT components—Parallel Adapter and Prefix-Tuning—varies significantly depending on the tunable parameter budget. This variability indicates that additional experiments are required to deeply understand the dynamics between different PEFT components and their impact on model performance. Specifically, exploring a broader range of parameter allocations could provide more insights into how each component contributes to fine-tuning success under varying conditions.

Lastly, the results of Experiment 2 demonstrated that the optimal allocation ratio is not robust across different tunable parameter budgets, suggesting that the suitability of a particular allocation ratio may depend heavily on the specific task and parameter budget. Future studies should consider a wider array of tasks and model architectures to verify the generalizability of these findings across different contexts and datasets.

Contribution Statement

Peiyang Wen implemented the model and experiment pipelines for the COPA, RTE, and BoolQ benchmarks. Yancheng He ran the experiments for the COPA benchmark. Lake Wang ran the experiments for the RTE benchmark. Zhihao Su ran the experiments for the BoolQ benchmark. The report was collaboratively written by all four team members.

References

Jiaao Chen, Aston Zhang, Xingjian Shi, Mu Li, Alex Smola, and Diyi Yang. 2023. Parameter-efficient fine-tuning design spaces. *arXiv preprint arXiv:2301.01821*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

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

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.