# MULTIPLICATIVE MIXTURE OF EXPERTS FOR RANK-EFFICIENT LLM FINETUNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large language models (LLMs) have achieved impressive results in many general-purpose domains, but their performance on specific tasks can still be improved through finetuning. Parameter-efficient finetuning (PEFT) tailors an LLM to one or more tasks through a small amount of trainable parameters, requiring reduced computational resources. On one hand, techniques like low-rank adaptation (LoRA) provide the required parameter efficiency with *additive* adapters of low, and fixed, rank, which limits their flexibility. On the other hand, mixture of experts (MoEs) enhance the flexibility of a model at the cost of parameter count and memory budget. The combination of the two approaches, parameter-efficient MoEfication, has shown promise in addressing the issues of both. In this work, we show that **replacing additive with multiplicative interactions** improves the efficiency of PEFT adapters, increasing the flexibility and reducing the number of parameters involved in MoEfication. We find that our quantum-inspired multiplicative method, OperA, is optimal given the same parameter budget for 7/8 models considered, using fewer or equal parameters than the baseline. In a multi-task setting, OperA fully surpasses all baselines on all models. We also propose a class of DMRG-based algorithms to accelerate OperA in large-scale deployments by alleviating its computational complexity, resulting in up to a 2.5× throughput improvement and up to a 8.9× memory reduction over the naïve implementation. Finally, we provide evidence that OperA surpasses the rank of competing solutions by more than two orders of magnitude.

## 1 INTRODUCTION

Large language models (LLMs) have demonstrated impressive capabilities (OpenAI, 2025; 2024; Qwen AI, 2025; Meta AI, 2024) when generalizing to novel tasks, making them the models of choice for a variety of domains (Wang et al., 2024; Liu et al., 2024a; 2023). Still, LLMs can be further finetuned to improve their performance in challenging scenarios (Hu et al., 2022; Parthasarathy et al., 2024). However, full finetuning remains prohibitively expensive in many cases (Xia et al., 2024b), motivating parameter-efficient finetuning (PEFT). Among PEFT methods (Han et al., 2024; Hu et al., 2022; Liu et al., 2024b; Singhal et al., 2026), low-rank adaptation (LoRA; Hu et al. 2022) is particularly effective but relies on fixed low-rank additive adapters, which limits its flexibility. While LoRA is often on par with its full-rank counterparts, its performance degrades when tasks can no longer be represented in low-rank subspaces (Feng et al., 2024; Xia et al., 2024a)

At the same time, mixture of experts (MoEs; Shazeer et al. 2017; Fedus et al. 2022) has been widely adopted in recent LLM architectures as a technique to improve performance across tasks without increasing the active parameter count (Jiang et al., 2024; IBM, 2024; DeepSeek-AI, 2025). MoEs leverage multiple linear models (i.e., experts) that process tokens in parallel, offering a factorization of large spaces into manageable chunks. By instantiating a large number of experts, but only activating a few of them for each token, LLMs can be internally specialized while keeping throughput high. However, in practice MoEs still suffer from increased memory requirements, as all experts need to be kept in memory even when not in use. Finally, LoRAs and MoEs can be combined to obtain a PEFT approach that is parameter-efficient, memory-efficient, and highly flexible (Tian et al., 2024; Li et al., 2024; Zeng et al., 2025). In particular, each expert is represented as a low-rank linear model, while maintaining the same gating and routing structure of regular MoEs (see Figure 1a), which have already been shown to be sufficiently powerful.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

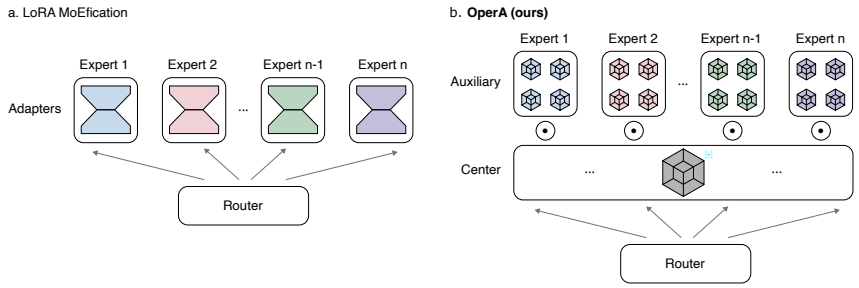



Figure 1: **Rank-efficient mixture of experts. a.** Regular LoRA MoEfication uses one low-rank adapter per expert. **b.** The experts of OperA are instead built through a multiplicative combination of 4D cores, obtained by matrix product operator (MPO) decomposition.

In this work, we show that replacing the additive interactions of LoRA with multiplicative interactions **increases the *stable rank* (Rudelson & Vershynin, 2007; Ipsen & Saibaba, 2025) attainable under a fixed parameter budget**, a property we refer to as rank efficiency. Therefore, our aim is to show that multiplicative approaches increase the rank efficiency of PEFT. Indeed, we find that our proposed multiplicative approach is most advantageous when parameter memory, rather than compute, is the dominant bottleneck: a regime increasingly common in large-scale MoE. To mitigate the additional computational cost of multiplicative adaptation, we further develop DMRG-inspired contraction algorithms that reorder tensor operations to improve both throughput and memory usage.

In particular, we introduce OperA (see Figure 1b), a PEFT technique that **replaces the fixed low-rank additive adapters of LoRA with multiplicative interactions** with the model weights. We borrow matrix product operators (MPOs; Pirvu et al. 2010; Schollwöck 2011; Orús 2014) from the quantum physics literature to provide an efficient multiplicative decomposition of arbitrary matrices into *core* and *auxiliary* 4D tensors. Through the MPO configuration we can concentrate the majority of the weights onto a frozen core, while learning the leaner auxiliary tensors during training. Therefore, the original weights participate directly in the construction of the OperA adapter.

Our main contributions are:

- We introduce OperA, a multiplicative alternative to additive LoRA MoEfication that removes the fixed-rank constraint of LoRA by constructing experts through trainable auxiliary factors of the pretrained weights.
- Across **8 LLMs and 8 downstream tasks**, OperA achieves **higher accuracy with fewer or equal parameters on 7/8 models** than the strongest additive baseline; in multi-task finetuning, it **surpasses all baselines on every model tested**.
- We develop **DMRG-inspired contraction algorithms** that exploit the structure of OperA to efficiently trade computation for memory, providing up to a $2.5\times$ throughput improvement and an $8.9\times$ peak memory reduction over the naïve implementation.
- We show empirically that OperA produces adapters with over two orders of magnitude greater stable rank than competing additive methods, supporting our hypothesis that multiplicative interactions enable substantially more rank-efficient adaptation.

## 2 Background

**Low-rank adaptation (LoRA).** LoRA is based on low-rank approximations of a matrix. In particular, given a pretrained weight $W \in \mathbb{R}^{M\times N}$, LoRA consists of two matrices $A \in \mathbb{R}^{M\times r}$ and $B \in \mathbb{R}^{r\times N}$, with $r \ll M, N$ being the LoRA *rank*. Crucially, while the LoRA rank places an upper bound on the dimensionality of the adapter, the actual subspace spanned by the adapter might be notably smaller. We explore this effect in detail through the use of the *stable rank* in Section 5.1.

Next, the product $AB \in \mathbb{R}^{M\times N}$ results in a shape-compatible matrix with $W$ that only requires a fraction of its parameters to store, producing considerable space savings. This comes to cost of the rank of $AB$, which is limited to $r$, and therefore to its flexibility. The final weight matrix becomes

$$W' = W + AB \tag{1}$$

**Mixture of experts (MoEs).** MoEs increase the flexibility and expressivity of LLMs, while maintaining a constrained increase in the number of parameters. Experts are matrices $E_1, ..., E_n \in \mathbb{R}^{M \times N}$ that independently operate on a given input vector $\boldsymbol{x} \in \mathbb{R}^M$, factorizing a large representation space into multiple additive components. The most popular implementation of MoEs uses a noisy top-k routing mechanism that selects $k$ out of $n$ experts for each $\boldsymbol{x}$, resulting in considerable computational savings at the cost of memory. Specifically, given a router $T$

$$y = \sum_i^n T(\boldsymbol{x})_i E_i(\boldsymbol{x}) \tag{2}$$

with

$$T(\boldsymbol{x})_i = \begin{cases} t_i & \text{if expert } i \text{ has been selected} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and $\sum_i^n t_i = 1$ are the routing weights, usually enforced via a *softmax* applied over the router outputs.

**Matrix product operators (MPO).** MPOs, in the setup used in this work, form a multiplicative representation of an arbitrary matrix $W \in \mathbb{R}^{M \times N}$ into 4D tensors (or cores) with a specific configuration. Specifically,

$$W = L_1 L_2 ... L_m C R_1 R_2 ... R_m \tag{4}$$

with $L_i,\ i \leq m$ being the left auxiliary cores, $C$ being the center core, and $R_i,\ i \leq m$ being the right auxiliary cores. The left and right auxiliary cores are analogous; we will refer to them simply as auxiliary cores from now on. Therefore, specifying a decomposition with $m$ left (or right) auxiliary cores results in a total of $2m + 1$ tensors. Each tensor has shape $\mathbb{R}^{d_{in}, r, c, d_{out}}$, with $r, c$ being the *physical row resp. column legs* and $d_{in}, d_{out}$ being the *bond dimensions*. The specification of $r$ and $c$ for each core represents the main hyperparameter choice of the MPO system, as we do not restrict the bond dimensions here (while it is more typical in quantum physics applications; Cui et al. 2015). Finally, a straightforward tensor contraction of all cores recovers the original matrix.

## 3 OperA: multiplicative mixture of experts

The aim of this work is to show that multiplicative interactions notably improve the parameter-efficiency of MoE-based PEFT while increasing the size of the subspace spanned by the experts, thereby also reducing the memory required by the adapters. Our OperA technique focuses on building the experts (or adapters) to obtain greater rank efficiency at lower parameter count.

As mentioned above, OperA leverages multiplicative interactions, which have been shown to be key to deep neural networks (Jayakumar et al., 2020). Here, these interactions are described by MPO, and help OperA achieve an even greater stable rank with lower parameter count compared to additive adapters. In contrast to LoRA, OperA constructs multiplicative adapters without an explicit low matrix-rank constraint by freezing the MPO core of the pretrained weight $W$ and training only expert-specific auxiliary tensors (unlike MPOE; Gao et al. 2022). The first step of OperA is to obtain the MPO representation of $W$ using tensor-train singular value decomposition (TT-SVD; Oseledets 2011). TT-SVD needs to be performed only once prior to training for each model, and does not represent a notable performance bottleneck. Without loss of generality, consider a decomposition into three cores

$$W = LCR \tag{5}$$

Given $n$ desired experts, we define the multiplicative adapters $L_i, R_i,\ i \leq n$ by initializing them to equal $L, R$. Next, each expert $E_i$ becomes

$$E_i = L_i C R_i \tag{6}$$

with $C$ fixed. Finally, given the same router $T$ as before the input ($\boldsymbol{x}$)-dependent OperA layer becomes

$$W' = \frac{1}{2} W + \sum_i^n T(\boldsymbol{x})_i \frac{1}{2} E_i \tag{7}$$

with the constant $1/2$ necessary to achieve equality at the beginning of training, analogous to setting the $B$ matrix to 0 in LoRA.

The key to the parameter-efficiency of OperA is to concentrate most of the parameters into the shared, frozen $C$. We remark that the main hyperparameters of TT-SVD consist in the sizes of the *physical legs* $r, c$ of the MPO system. In particular, we set row legs $r_1, r_2, r_3$ (with $r_1 \times r_2 \times r_3 = M$) and column legs $c_1, c_2, c_3$ (with $c_1 \times c_2 \times c_3 = N$). With these parameters set, we can now fully describe the shapes of the MPO:

$$L \in \mathbb{R}^{1 \times r_1 \times c_1 \times d_1}, C \in \mathbb{R}^{d_1 \times r_2 \times c_2 \times d_2}, R \in \mathbb{R}^{d_2 \times r_3 \times c_3 \times 1} \tag{8}$$

The bond dimensions $d_1, d_2$ are an effect of TT-SVD and cannot be directly lowered without resulting in a lossy reconstruction of $W$ as in regular SVD. Therefore, we can simply choose $r_2 \gg r_{i \neq 2}$ and $c_2 \gg c_{i \neq 2}$ to obtain the desired parameter-reduction effect.

The multiplicative adapters $L_i, R_i$ do not directly limit the rank of the expert, allowing the system to exhibit greater variety and flexibility. In particular, the physical operator is completely determined by the contraction of auxiliary tensors. Concretely, the subspace of representable weight matrices is limited by the bond dimensions resulting from TT-SVD (Pirvu et al., 2010), which are in practice orders of magnitude larger than the LoRA rank (see Figure 2).

Nonetheless, the greater parameter efficiency comes at a computational cost, as each expert is now full-rank and must be obtained from a tensor contraction at every step. Therefore, OperA trades additional computation for reduced adapter parameter storage compared with LoRA-based MoEs. This trade-off is particularly attractive when the memory footprint of the adapters is the dominant constraint.

### 3.1 Efficient implementations of OperA

The multiplicative nature of OperA necessarily implies higher computational costs with respect to additive solutions like LoRA, resulting in a compute-memory trade-off (we evaluate this effect concretely in Appendix C). At the same time, reducing parameter memory is a key component of PEFT MoEfication, especially as the number of experts grows, and is therefore the focus of this work.

A common optimization for LoRA is to share computation involving the frozen base weights across adapters. Reusing the notation of Section 2, this principle extends directly to LoRA MoEfication, as shown in Algorithm 1: the multiplication with $W$ is performed only once, while the cheaper adapter-specific operations are repeated for each expert.

**Algorithm 1** Efficient computation of LoRA experts for PEFT MoEfication.

**Input:** data $\boldsymbol{x}$, weight $W$, adapters $A_i, B_i \ \forall i \in [1, ..., K]$, number of experts $K$
Initialize $\boldsymbol{o} = W\boldsymbol{x}$
**for** $i = 1$ **to** $K$ **do**
  $E_i(\boldsymbol{x}) = \boldsymbol{o} + A_i B_i \boldsymbol{x}$
**end for**

**Algorithm 2** Efficient computation of OperA experts for PEFT MoEfication.

**Input:** data $\boldsymbol{x}$, central core $C$, auxiliary cores $L_i, R_i \ \forall i \in [1, ..., K]$, number of experts $K$
Initialize $O = C\boldsymbol{x}$
**for** $i = 1$ **to** $K$ **do**
  $E_i(\boldsymbol{x}) = OL_iR_i$
**end for**

Although this additive decomposition does not directly apply to OperA, its tensor-network structure provides an analogous opportunity for efficient computation. Importantly, the MPO does not need to be contracted in a fixed order. Rather than independently contracting each expert as $L_iCR_i$, we can reorder the tensor contractions such that the expensive operation involving the shared central core $C$ is performed only once, while the smaller auxiliary cores $L_i$ and $R_i$ remain expert-specific. This computation is summarized in Algorithm 2. This optimization is inspired by contraction strategies used in density matrix renormalization group (DMRG; Schollwöck 2011), where tensor-network contractions are organized to efficiently reuse intermediate results. Applied to OperA, this observation yields a class of DMRG-inspired implementations that vary the contraction order depending on the MPO configuration.

Table 1: **Accuracy of PEFT MoEfication on the downstream tasks.** For each model, we report the number of parameters and the task accuracy (%).

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | PMQA ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Llama3-8B | None | / | 66.04 | 81.19 | 62.24 | 67.80 | 59.27 | 48.43 | 67.80 | 54.40 | 63.40 |
| | LoRAMoE | 246M | 77.56 | 86.62 | 87.60 | 87.80 | 74.33 | 95.21 | 84.29 | 73.10 | 83.31 |
| | MixLoRA | 241M | 75.76 | 87.08 | 86.18 | 89.00 | 73.61 | 94.63 | 83.11 | 55.00 | 80.55 |
| | OperA | 173M | 77.39 | 86.28 | 88.30 | 87.60 | 76.36 | 96.15 | 81.45 | 74.10 | **83.45** |
| Llama3.1-8B | None | / | 68.34 | 81.19 | 67.08 | 69.20 | 57.37 | 55.35 | 52.72 | 58.30 | 63.69 |
| | LoRAMoE | 246M | 79.52 | 85.50 | 87.32 | 86.20 | 74.46 | 92.50 | 82.32 | 73.20 | 82.63 |
| | MixLoRA | 241M | 76.96 | 86.82 | 86.40 | 83.20 | 74.65 | 94.77 | 84.14 | 54.40 | 80.17 |
| | OperA | 173M | 76.88 | 86.32 | 87.92 | 85.20 | 75.23 | 96.10 | 80.58 | 74.70 | **82.87** |
| Llama3.2-3B | None | / | 53.41 | 66.79 | 63.82 | 55.00 | 57.61 | 45.98 | 51.93 | 55.30 | 56.23 |
| | LoRAMoE | 133M | 67.92 | 83.29 | 82.97 | 77.00 | 71.68 | 90.62 | 74.22 | 69.90 | 77.20 |
| | MixLoRA | 131M | 68.26 | 83.12 | 83.02 | 77.40 | 71.50 | 92.13 | 75.14 | 68.10 | **77.33** |
| | OperA | 92M | 66.98 | 81.73 | 83.19 | 77.20 | 70.98 | 92.93 | 74.19 | 71.10 | 77.29 |
| Phi4-14B | None | / | 79.95 | 84.76 | 90.04 | 79.40 | 68.72 | 87.19 | 77.74 | 72.90 | 80.09 |
| | LoRAMoE | 342M | 92.02 | 94.11 | 92.44 | 94.00 | 74.57 | 95.61 | 87.53 | 71.40 | 87.71 |
| | MixLoRA | 336M | 91.21 | 94.28 | 91.90 | 92.60 | 75.47 | 95.95 | 88.79 | 73.40 | **87.95** |
| | OperA | 182M | 88.14 | 92.84 | 91.84 | 91.00 | 74.49 | 95.25 | 86.27 | 70.60 | 86.30 |
| Qwen2.5-3B | None | / | 77.65 | 87.58 | 75.46 | 75.00 | 63.64 | 72.08 | 57.38 | 58.80 | 70.95 |
| | LoRAMoE | 192M | 80.11 | 88.34 | 85.58 | 85.30 | 70.70 | 91.57 | 78.93 | 70.50 | 81.38 |
| | MixLoRA | 188M | 80.12 | 88.64 | 86.62 | 87.60 | 70.92 | 91.67 | 80.03 | 70.40 | **82.00** |
| | OperA | 74M | 80.89 | 89.40 | 86.02 | 87.60 | 70.06 | 92.99 | 74.82 | 70.60 | 81.55 |
| Qwen2.5-7B | None | / | 85.07 | 91.33 | 81.99 | 84.60 | 63.09 | 86.00 | 58.64 | 63.10 | 76.73 |
| | LoRAMoE | 258M | 87.03 | 91.79 | 89.83 | 92.00 | 74.22 | 95.30 | 84.93 | 71.60 | 85.84 |
| | MixLoRA | 253M | 87.88 | 94.19 | 88.74 | 92.80 | 74.52 | 95.29 | 83.27 | 52.70 | 83.67 |
| | OperA | 192M | 87.71 | 93.39 | 90.42 | 91.40 | 74.74 | 95.91 | 83.66 | 72.50 | **86.22** |
| Qwen2.5-14B | None | / | 89.42 | 93.10 | 88.79 | 86.80 | 70.61 | 91.12 | 69.22 | 67.60 | 82.08 |
| | LoRAMoE | 384M | 90.66 | 93.42 | 91.46 | 96.20 | 76.69 | 97.01 | 91.32 | 74.10 | 88.86 |
| | MixLoRA | 376M | 90.78 | 93.73 | 92.27 | 95.60 | 77.58 | 96.82 | 90.21 | 53.10 | 86.26 |
| | OperA | 200M | 90.61 | 93.73 | 93.42 | 96.00 | 78.10 | 97.43 | 90.13 | 75.20 | **89.33** |
| Qwen2.5-32B | None | / | 92.24 | 94.23 | 89.77 | 89.80 | 70.03 | 94.21 | 77.35 | 62.10 | 83.72 |
| | LoRAMoE | 858M | 93.43 | 94.32 | 94.23 | 96.40 | 78.26 | 97.24 | 92.23 | 53.40 | 87.37 |
| | MixLoRA | 841M | 93.26 | 95.75 | 94.07 | 97.00 | 78.35 | 97.29 | 90.92 | 73.20 | 89.98 |
| | OperA | 343M | 93.94 | 94.49 | 94.29 | 97.40 | 79.51 | 97.38 | 90.53 | 74.20 | **90.22** |

The optimal contraction order depends on the specific OperA configuration, since different orders induce different intermediate tensor sizes and computational costs. We therefore select the contraction order to jointly reduce memory and computation. In practice, our optimized implementation (see Appendix C) yields up to a $2.5\times$ throughput increase and up to an $8.9\times$ memory reduction over the naïve implementation, substantially narrowing the efficiency gap with purely additive solutions.

## 4 Experimental setup

**Adapters.** We compare MoLA, LoRAMoE, MixLoRA, HydraLoRA, and OperA on a variety of models to ensure the consistency of our performance results. As MPOE (Gao et al., 2022) requires the same peak memory as full finetuning, we do not include it here (see Appendix D.8 for an ablation of the weight-derived initialization of OperA). We also compare our baselines with another additive adapter, SharedLoRA, that we develop based on a two-tier structure (further information on this schema can be found in Appendix A). We report the results using LoRAMoE and MixLoRA in the main text, as they achieved best average accuracy. The full results are reported in Appendix D.

**Models.** From Meta, we test: Llama3-8B, Llama3.1-8B, and Llama3.2-3B. From Qwen, we test: Qwen2.5-3B, Qwen2.5-7B, Qwen2.5-14B, and Qwen2.5-32B. Finally, from Microsoft we test Phi4-14B. This choice of the 8 models covers a representative range of sizes (from 3B to 32B parameters) and of model families.

**Tasks.** For each dataset, we finetune each adapter on the training split, and then evaluate the accuracy on the testing split. The task ensemble consists of question-and-answer tasks (ARC-e, ARC-c, PIQA, OpenBookQA), classification tasks (BoolQ), and completion tasks (HellaSwag, WinoGrande, PubmedQA). Moreover, we also test the adapters' ability in a multi-task context, where the adapters are trained and tested on *all* tasks simultaneously. All tasks are evaluated on accuracy.

Table 2: **Comparison of OperA given the same or lower parameter budget against competing solutions.** For each model, we report the number of parameters and the task accuracy (%).

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | PMQA ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Llama3-8B | LoRAMoE | 111M | 77.55 | 86.49 | 87.76 | 86.20 | 74.42 | 94.73 | 82.11 | 75.10 | 83.05 |
| | MixLoRA | 106M | 78.75 | 87.12 | 83.81 | 85.20 | 74.62 | 94.75 | 83.90 | 75.30 | 82.93 |
| | OperA | 95M | 77.64 | 86.41 | 87.60 | 84.20 | 74.74 | 95.71 | 83.03 | 76.10 | **83.18** |
| Llama3.1-8B | LoRAMoE | 111M | 77.33 | 86.32 | 88.52 | 85.60 | 74.68 | 95.61 | 82.35 | 75.50 | 83.24 |
| | MixLoRA | 106M | 78.33 | 86.45 | 83.60 | 87.40 | 75.69 | 95.16 | 83.42 | 76.10 | 83.27 |
| | OperA | 95M | 77.73 | 87.29 | 88.19 | 85.40 | 75.69 | 95.91 | 80.03 | 76.70 | **83.37** |
| Llama3.2-3B | LoRAMoE | 60M | 64.76 | 82.53 | 83.68 | 79.40 | 71.12 | 93.42 | 76.32 | 73.00 | 78.03 |
| | MixLoRA | 59M | 69.33 | 83.42 | 83.17 | 79.40 | 70.98 | 92.15 | 76.64 | 72.40 | 78.44 |
| | OperA | 59M | 69.08 | 84.64 | 82.64 | 79.40 | 71.53 | 93.60 | 75.16 | 73.00 | **78.63** |
| Phi4-14B | LoRAMoE | 154M | 91.81 | 96.70 | 92.71 | 93.20 | 76.02 | 95.26 | 88.08 | 75.00 | 88.60 |
| | MixLoRA | 148M | 91.47 | 96.68 | 92.27 | 93.60 | 76.02 | 95.81 | 87.85 | 76.2 | **88.74** |
| | OperA | 96M | 87.29 | 92.80 | 90.91 | 91.00 | 75.17 | 95.38 | 87.21 | 75.10 | 86.86 |
| Qwen2.5-3B | LoRAMoE | 87M | 80.20 | 89.60 | 84.71 | 87.20 | 71.35 | 92.62 | 78.85 | 69.90 | 81.80 |
| | MixLoRA | 83M | 82.08 | 88.39 | 85.70 | 84.93 | 70.80 | 92.76 | 79.79 | 69.90 | 81.79 |
| | OperA | 74M | 80.89 | 89.40 | 86.02 | 87.60 | 70.06 | 92.99 | 77.82 | 71.80 | **82.07** |
| Qwen2.5-7B | LoRAMoE | 116M | 87.46 | 92.21 | 90.91 | 91.80 | 73.98 | 95.17 | 83.90 | 72.30 | 85.97 |
| | MixLoRA | 111M | 86.77 | 91.92 | 88.93 | 92.40 | 75.17 | 95.36 | 85.79 | 72.50 | 86.11 |
| | OperA | 93M | 87.63 | 92.51 | 90.59 | 94.00 | 74.07 | 95.76 | 83.98 | 72.20 | **86.34** |
| Qwen2.5-14B | LoRAMoE | 173M | 91.33 | 93.50 | 93.09 | 97.00 | 77.80 | 97.09 | 90.21 | 74.60 | 89.33 |
| | MixLoRA | 166M | 91.47 | 93.64 | 92.82 | 96.00 | 77.74 | 97.09 | 91.63 | 74.80 | 89.40 |
| | OperA | 153M | 91.04 | 93.81 | 93.36 | 96.80 | 77.92 | 97.30 | 90.90 | 74.60 | **89.47** |
| Qwen2.5-32B | LoRAMoE | 439M | 93.69 | 94.65 | 94.40 | 96.40 | 78.44 | 97.53 | 91.87 | 73.70 | 90.09 |
| | MixLoRA | 422M | 93.52 | 94.49 | 93.85 | 96.20 | 79.11 | 97.29 | 91.27 | 73.20 | 89.87 |
| | OperA | 343M | 93.94 | 94.49 | 94.29 | 97.40 | 79.51 | 97.38 | 90.53 | 74.20 | **90.22** |

**MoEs.** For all experiments, we use 8 total experts with 2 active experts per token. Following best practices, we use noisy top-k routing with an auxiliary router balancing loss (see Appendix D.8 for an ablation).

**LoRA.** For all additive (LoRA-based) approaches, the rank of the adapters is the main hyperparameter. In particular, the rank is set to either 16 (original, see Section 5.1), 12 (optimal, see Appendix D.3), or 8 (same memory budget as OperA, see Section 5.1). In Appendix D.9 we provide an ablation of LoRA without experts.

**OperA.** We reuse the basic architecture of LoRA MoEs of MixLoRA (Li et al., 2024) (see Appendix D.8 for an ablation of MixLoRA's fusion). In particular, we replace the MLP layers with OperA MoEs and use standard LoRAs (with rank 6) for the attention layers. In contrast to LoRA, OperA does not have an explicit matrix-rank hyperparameter that directly bounds the rank of the resulting update. Its capacity is controlled by the MPO configuration, including the physical-leg sizes and the bond dimensions induced by TT-SVD. Specifically, we confirm that 5 legs is the optimal choice (Gao et al., 2022). We also find that a 40% to 60% concentration of the legs onto the central core provides a well-balanced solution, leading to empirical optimality. Indeed, if too little parameters are given to the auxiliary cores then no adaptation can take place, while too large auxiliary cores do not achieve the required parameter efficiency. The specific sizes can be found in Appendix B.

**Environment.** All experiments are performed on Nvidia A100 80GB or Nvidia H200 140GB using PyTorch 2.6 and the MoE-PEFT (Li et al., 2024) codebase, using 10 to 500 GPU hours per task depending on the model size. All models are trained using mixed fp32 and bf16.

# 5 Results

## 5.1 Downstream accuracy

**Baseline.** We replicate the setup of MixLoRA to obtain comparable results with established methods, and follow the original hyperparameter selection. In Table 1 we show that OperA achieves competitive accuracy across all models and tasks, also surpassing all baselines on 6/8 models, while obtaining on average a $36\%$ space saving across the tested models with respect to MixLoRA, the most efficient additive adapter. The comparison against all baselines is provided in Appendix D.1. In particular, OperA achieves $61\%$ parameter reduction against MixLoRA on Qwen2.5-3B and a

Table 3: **Accuracy of PEFT MoEfication on multiple downstream tasks simultaneously.** For each model, we report the number of parameters and the task accuracy (%).

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | PMQA ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Llama3-8B | LoRAMoE | 246M | 79.44 | 86.91 | 88.25 | 86.80 | 75.78 | 95.18 | 83.27 | 73.40 | 83.63 |
| | MixLoRA | 241M | 76.45 | 86.99 | 88.14 | 82.20 | 75.23 | 94.24 | 83.43 | 54.60 | 80.16 |
| | OperA | 173M | 78.13 | 87.42 | 88.68 | 86.60 | 74.56 | 96.00 | 84.16 | 73.90 | **83.68** |
| Llama3.1-8B | LoRAMoE | 246M | 79.01 | 86.99 | 88.74 | 84.80 | 75.99 | 94.53 | 84.45 | 74.30 | 83.60 |
| | MixLoRA | 241M | 76.71 | 86.45 | 88.19 | 85.80 | 73.24 | 94.76 | 80.43 | 54.50 | 80.01 |
| | OperA | 173M | 78.88 | 87.96 | 88.36 | 86.00 | 74.79 | 95.95 | 84.00 | 73.80 | **83.72** |
| Llama3.2-3B | LoRAMoE | 133M | 67.41 | 84.51 | 84.98 | 77.80 | 71.44 | 93.67 | 75.45 | 73.30 | 78.57 |
| | MixLoRA | 131M | 68.09 | 83.92 | 83.62 | 78.80 | 70.49 | 92.40 | 77.03 | 71.20 | 78.19 |
| | OperA | 92M | 69.62 | 84.72 | 83.46 | 79.20 | 70.76 | 93.71 | 76.32 | 72.30 | **78.76** |
| Phi4-14B | LoRAMoE | 342M | 91.47 | 94.11 | 92.66 | 92.60 | 76.97 | 96.15 | 87.69 | 75.70 | 88.42 |
| | MixLoRA | 336M | 91.89 | 94.91 | 91.89 | 93.40 | 75.50 | 95.93 | 88.40 | 75.80 | 88.47 |
| | OperA | 182M | 92.53 | 96.27 | 91.99 | 93.00 | 77.10 | 97.03 | 88.77 | 75.50 | **89.02** |
| Qwen2.5-3B | LoRAMoE | 192M | 82.51 | 89.86 | 86.11 | 87.40 | 70.46 | 92.34 | 79.16 | 69.30 | 82.14 |
| | MixLoRA | 188M | 80.89 | 88.76 | 86.60 | 88.00 | 71.44 | 93.29 | 78.37 | 70.20 | 82.19 |
| | OperA | 74M | 80.38 | 89.35 | 86.72 | 88.60 | 70.60 | 93.42 | 78.98 | 70.10 | **82.27** |
| Qwen2.5-7B | LoRAMoE | 258M | 87.37 | 91.96 | 89.77 | 92.60 | 74.95 | 95.24 | 84.21 | 72.70 | 86.10 |
| | MixLoRA | 253M | 87.88 | 93.94 | 90.64 | 91.40 | 75.44 | 94.63 | 85.08 | 52.30 | 83.91 |
| | OperA | 192M | 87.71 | 92.13 | 91.13 | 92.00 | 74.50 | 95.85 | 83.74 | 72.80 | **86.23** |
| Qwen2.5-14B | LoRAMoE | 384M | 91.30 | 93.73 | 92.93 | 94.80 | 78.01 | 97.12 | 89.74 | 74.10 | 88.97 |
| | MixLoRA | 376M | 90.96 | 93.73 | 92.11 | 95.20 | 78.38 | 96.75 | 89.58 | 74.00 | 88.84 |
| | OperA | 200M | 91.90 | 93.73 | 93.10 | 95.50 | 78.11 | 96.90 | 90.55 | 74.70 | **89.31** |

70% reduction against MoLA on the same model. We also provide a robust evaluation with 4 seeds of OperA and MixLoRA on a subset of models and tasks in Appendix D.2.

**Optimized additive adapters with a fixed parameter budget.** We have demonstrated that OperA is competitive with competing solutions while requiring notably less parameters. At the same time, in Appendix D.3 we also further optimize the additive LoRAs, and MixLoRA in particular, both for accuracy and for parameter count, and show that OperA outperforms the baselines in this setting as well. However, the question remains whether OperA can actually leverage its multiplicative interactions to be more parameter-efficient than the additive baselines. Table 2 indicates that OperA is optimal on 7 out of the 8 models, achieving higher accuracy at a lower parameter count than the baseline. As before, we again provide a multi-seed evaluation in Appendix D.4.

**Multi task.** We have shown that OperA maintains a high accuracy while notably reducing the parameter count across all tasks. At the same time, we wish to stress the representation capabilities of all PEFT methods by also considering a multi-task scenario, where the required dimensionality of the modified subspace increases. Qwen2.5-32B cannot be trained on a single GPU in this scenario, so it is excluded. In Table 3 we report the results of OperA on all the considered tasks at the same time (full results in Appendix D.5), consistent with the experimental setup of the MoE-PEFT codebase (Li et al., 2024). Here, OperA surpasses all baselines, consistent with the advantage expected from a less rank-constrained adapter in the more demanding multi-task setting.

**Stable rank.** While the LoRA rank puts an upper bound to the dimensionality of the PEFT, whether the adapters do actually make use of all available space can be directly investigated. Here, we choose the stable rank (Roy & Vetterli, 2007) as the measure of interest due to its inherent robustness. Here, we use the stable rank (Rudelson & Vershynin, 2007; Ipsen & Saibaba, 2025) as a robust measure of the dimensionality of an adapter:

$$\text{srank}(W) = \frac{\|W\|_F^2}{\|W\|_2^2} \tag{9}$$

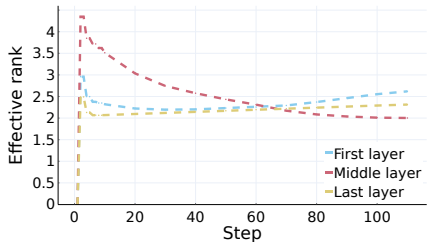
where $\|\cdot\|_F$ is the Frobenius norm and $\|\cdot\|_2$ is the L2 norm. Unlike matrix rank, stable rank is insensitive to arbitrarily small singular values and therefore provides a more robust measure of the effective dimensionality of an adapter.

Given the full MixLoRA expert $E_i = W + A_i B_i$, we compute the stable rank of the adapters as training progresses as $\text{srank}(W - E_i)$. We do the same in the case of OperA, where each $E_i = L_i C R_i$. This choice is suggested by the fact that, for OperA $E_i$ is full rank with high probability, and we wish to evaluate the additional dimensionality brought by $E_i$ to the system, rather than its

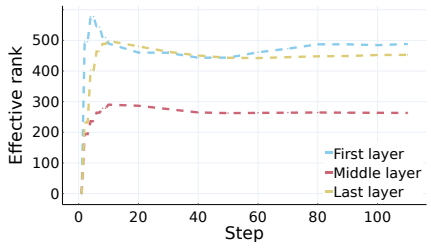
raw dimensionality. Figure 2 shows that the stable rank quickly rises at the beginning of training, as expected, but then progressively settles. In the case of MixLoRA, the stable rank degrades quickly, hovering around 2.5 for both shallow and deep layers, consistent with the findings in (Li et al., 2024) that performance is maintained down to a LoRA rank of 2. On the other hand OperA's stable rank is notably higher and more stable than the additive approaches, indicating that the updates induced by by OperA span a much greater subspace despite using fewer trainable parameters.

### 5.2 Ablations

**Optimal rank.** The main hyperparameter that determines the parameter count of LoRA-based approaches is the rank. As mentioned earlier, to obtain baseline results we first apply all adapters with rank 16, reported to be optimal for MixLoRA (Li et al., 2024). However, we also perform a fine-grained search using modern architectures to ensure as strong a baseline as possible. Figure 3 shows that performance peaks at rank 12 for Llama3-8B, while it remains stable after rank 12 for Qwen2.5-3B. Therefore, we choose rank 12 as the most parameter-efficient solution, and extend this evaluation to all architectures and other adapters to establish an optimized baseline in Appendix D.3.




(a) Stable rank of MixLoRA.




(b) Stable rank of OperA.

Figure 2: **Stable rank of the adapter updates.** Stable rank of $W - E_i$ as finetuning progresses.

**Optimal MPO size.** The parameter count of OperA is mainly determined by the configuration of the physical legs, which are specific to each model and are evaluated empirically for effectiveness: this is analogous to the choice of rank for LoRA. Nonetheless, the amount of legs also contributes to the downstream performance by varying the degree of factorization. To obtain the optimal number, we sweep from 3 to 7 legs on Qwen2.5-3B and Llama3-8B and check the average performance across 5 tasks. Figure 4 shows that 5 legs provide the best accuracy.

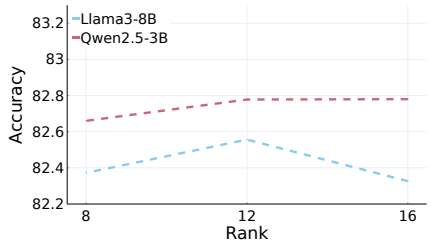



Figure 3: **MixLoRA rank ablation.** We report the average accuracy at varying LoRA ranks, using MixLoRA.

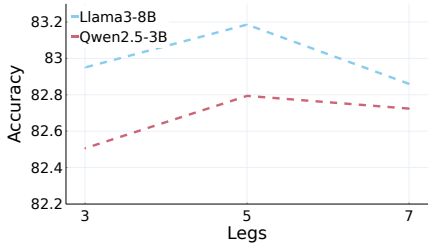



Figure 4: **OperA legs ablation.** We report the average accuracy at various leg configurations, using OperA.

## 6 Related Works

**Parameter-efficient finetuning (PEFT).** The most popular approach to PEFT is LoRA (Hu et al., 2022), which finetunes a low-dimensional subspace of the original weights to improve performance while reducing the required number of parameters. Further gains on parameter efficiency have been obtained by weight sharing (VeRA; Kopiczko et al. 2024), weight tying (Tied-LoRA; Renduchintala et al. 2024), quantization (QLoRA; Dettmers et al. 2023), and other techniques (Yang et al., 2025;

Hayou et al., 2025). The downstream performance of LoRA has also been improved upon, especially through normalization (DoRA; Liu et al. 2024b).

**Mixture of experts (MoEs).** MoEs (Shazeer et al., 2017; Fedus et al., 2022) allow LLMs to efficiently scale the number of parameters without proportionally scaling their compute budgets. There are multiple variants of MoEs, specifying various routing (Fedus et al., 2022; Ruiz et al., 2021; Yue et al., 2025) and gating mechanisms (Li et al., 2023; Puigcerver et al., 2024). Recent works (DeepSeek-AI, 2025; Qwen AI, 2025) have also shown that very large mixtures with few experts active at a time show remarkable performance across a wide variety of tasks.

**Parameter-efficient MoEfication.** The combination of PEFT with MoEs has recently started gaining traction (Dou et al., 2024; Tian et al., 2024; Li et al., 2024; Zeng et al., 2025; Gao et al., 2025), to address the limitations of both approaches. Most parameter-efficient MoEfication techniques (Luo et al., 2024; Gao et al., 2024), like LoRAMoE (Dou et al., 2024), replace all linear layers with MoEs. Even greater efficiency is achieved in HydraLoRA by splitting the LoRA adapters into heads (Tian et al., 2024), driving further factorization. Greater specialization is achieved by MoLA (Gao et al., 2025), that targets Transformer layers with different numbers of experts. MixLoRA (Li et al., 2024) selectively replaces the QKV matrices in the attention layers with simple LoRAs, while fusing the MLP layers with parameter-efficient MoEs. More recently, S'MoRE (Zeng et al., 2025) proposes hierarchical mixtures with structured graphs.

**Multiplicative adapters.** Multiplicative adapters for PEFT have seen some adoption, for example with Kronecker-Product (Yu et al., 2025), Khatri-Rao (Albert et al., 2025). A combination of additive and multiplicative adapters using Hadamard product has been shown (Singhal et al., 2026; Huang et al., 2025; Yeh et al., 2024) to notably increase the resulting rank, but still within the framework of additive LoRAs. We explore here matrix product operators (MPOs), a particularly effective multiplicative decomposition from quantum physics (Pirvu et al., 2010; Schollwöck, 2011; Orús, 2014). MPOs have been used for compressing linear layers (Gao et al., 2020) and as a direct alternative to LoRAs (Liu et al., 2021). Most closely related is MPOE (Gao et al., 2022), which also constructs experts through MPOs. Unlike OperA, however, MPOE trains the central core, whereas OperA freezes it and learns only expert-specific auxiliary tensors. Therefore, MPOE requires the same computational resources as full finetuning, while OperA and the other multiplicative adapters mentioned are parameter-efficient. See Appendix E for a detailed comparison. A more detailed comparison between OperA and other related adapters can be found in Appendix E.

## 7 Conclusion

In this work, we show that multiplicative interactions, which are fundamental to learning in deep neural networks (Jayakumar et al., 2020), notably benefit parameter-efficient MoEfication by improving parameter efficiency and increasing the stable rank of the adapters. OperA leverages matrix product operators (MPOs), a multiplicative decomposition stemming from quantum physics, to produce low-parameter adapters that yield high-rank experts. Moreover, we highlight the usefulness of the stable rank as a quantification of the dimensionality of an adapter, and show that OperA achieves substantially higher stable rank while using fewer trainable parameters.

Overall, we compare OperA against 4 established methods across 8 models and 8 downstream tasks. Specifically, OperA reaches a higher accuracy than all baselines at a lower parameter cost on 7/8 models tested. Therefore, OperA provides a parameter-efficient MoEfication technique that trades additional computation for reduced adapter parameter storage, making it appealing for large-scale deployment where memory is scarce.

**Limitations.** OperA strikes a different balance than LoRA, reducing adapter parameter storage at the expense of additional computation. While we have proposed a class of DMRG-style algorithms that notably narrow the throughput gap with additive adapters (see Section 3.1 and Appendix C), this trade-off remains an intrinsic characteristic of our method. Moreover, OperA does not outperform the additive baselines on Phi4-14B in the single-task setting, although it does so in the multi-task setting. Understanding which model and task characteristics determine the relative advantage of multiplicative adaptation remains an avenue for future work.

### AI USE STATEMENT

In this work, we used generative AI tools to help evaluate the efficient implementations. We did not use generative AI tools to generate the core method or to help write the paper. We have reviewed all AI-assisted work: the code was run and its outputs checked by the authors. We take responsibility for the final content of this work, including artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

All research conducted conforms with the Code of Ethics. There are many potential consequences of work regarding LLMs, however, there is no specific ethics impact deriving from this work.

### REPRODUCIBILITY STATEMENT

All datasets used are publicly available. The architecture, the training setup and the hyperparameters of every benchmark are in Section 4 and Appendix B. As indicated by ICLR's author guidelines, we are available to share the code anonymously during the rebuttal period.

## REFERENCES

Paul Albert, Frederic Z Zhang, Hemanth Saratchandran, Anton van den Hengel, and Ehsan Abbasnejad. Towards higher effective rank in parameter-efficient fine-tuning using khatri-rao product. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1292–1302. IEEE, 2025.

Jian Cui, J. Ignacio Cirac, and Mari Carmen Bañuls. Variational matrix product operators for the steady state of dissipative quantum systems. *Phys. Rev. Lett.*, 114, 2015.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2025.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *NeurIPS*, 2023.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research (JMLR)*, 23 (120):1–39, 2022.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. Mixture-of-loras: An efficient multitask tuning method for large language models. In *International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, 2024.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. Higher layers need more LoRA experts. *arXiv preprint arXiv:2402.08562*, 2024.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Ruibo Liu, Baochen Sun, Yawen Zhang, Daiyi Peng, Xiaoyuan Guo, and Vs Subrahmanian. MoLA: MoE LoRA with layer-wise expert allocation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025.

Ze-Feng Gao, Song Cheng, Rong-Qiang He, Z. Y. Xie, Hui-Hai Zhao, Zhong-Yi Lu, and Tao Xiang. Compressing deep neural networks by matrix product operators. *Physical Review Research*, 2(2), 2020.

Ze-Feng Gao, Peiyu Liu, Wayne Xin Zhao, Zhong-Yi Lu, and Ji-Rong Wen. Parameter-efficient mixture-of-experts architecture for pre-trained language models. In *Iternational Conference on Computational Linguistics (ICCL)*, 2022.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research (TMLR)*, 2024.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. PLoP: Precise LoRA placement for efficient finetuning of large models. *arXiv preprint arXiv:2506.20629*, 2025.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. Hira: Parameter-efficient hadamard high-rank adaptation for large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

IBM. Granite 3.0 language models. `https://github.com/ibm-granite/granite-3.0-language-models/`, 2024.

Ilse CF Ipsen and Arvind K Saibaba. Stable rank and intrinsic dimension of real and complex matrices. *SIAM Journal on Matrix Analysis and Applications*, 46(3):1988–2007, 2025.

Siddhant M. Jayakumar, Wojciech M. Czarnecki, Jacob Menick, Jonathan Schwarz, Jack Rae, Simon Osindero, Yee Whye Teh, Tim Harley, and Razvan Pascanu. Multiplicative interactions and where to find them. In *ICLR*, 2020.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *ICLR*, 2024.

Bo Li, Yifei Shen, Jingkang Yang, Yezhen Wang, Jiawei Ren, Tong Che, Jun Zhang, and Ziwei Liu. Sparse mixture-of-experts are domain generalizable learners. In *ICLR*, 2023.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. MixLoRA: Enhancing large language models fine-tuning with LoRA-based mixture of experts. *arXiv preprint arXiv:2404.15159*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, January 2024a. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

Peiyu Liu, Ze-Feng Gao, Wayne Xin Zhao, Zhi-Yuan Xie, Zhong-Yi Lu, and Ji-Rong Wen. Enabling lightweight fine-tuning for pre-trained language model compression based on matrix product operators. In *International Joint Conference on Natural Language Processing (IJCNLP)*, 2021.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *ICML*, 2024b.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. MoELoRA: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv:2402.12851*, 2024.

Meta AI. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

OpenAI. GPT-5 system card. https://openai.com/index/gpt-5-system-card/, 2025.

Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of physics*, 349:117–158, 2014.

I. V. Oseledets. Tensor-Train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. The ultimate guide to fine-tuning LLMs from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*, 2024.

Bogdan Pirvu, Valentin Murg, J Ignacio Cirac, and Frank Verstraete. Matrix product operator representations. *New Journal of Physics*, 12(2), 2010.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *ICLR*, 2024.

Qwen AI. Qwen3-Next. https://qwen.ai/blog?id=4074cca80393150c248e508aa62983f9cb7d27cd&from=research.latest-advancements-list, 2025.

Adithya Renduchintala, Tugrul Konuk, and Oleksii Kuchaiev. Tied-LoRA: Enhancing parameter efficiency of LoRA with weight tying. In *NAACL*, 2024.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *European signal processing conference*, 2007.

Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 54(4):21–es, 2007.

Carlos Riquelme Ruiz, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *NeurIPS*, 2021.

Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of physics*, 326(1):96–192, 2011.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

Raghav Singhal, Kaustubh Ponkshe, Rohit Vartak, and Praneeth Vepakomma. ABBA-adapters: Efficient and expressive fine-tuning of foundation models. In *International Conference on Learning Representations (ICLR)*, 2026.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng zhong Xu. HydraloRA: An asymmetric loRA architecture for efficient fine-tuning. In *NeurIPS*, 2024.

Jiaqi Wang, Hanqi Jiang, Yiheng Liu, Chong Ma, Xu Zhang, Yi Pan, Mengyuan Liu, Peiran Gu, Sichen Xia, Wenjun Li, Yutong Zhang, Zihao Wu, Zhengliang Liu, Tianyang Zhong, Bao Ge, Tuo Zhang, Ning Qiang, Xintao Hu, Xi Jiang, Xin Zhang, Wei Zhang, Dinggang Shen, Tianming Liu, and Shu Zhang. A comprehensive review of multimodal large language models: Performance and challenges across different tasks. *arXiv preprint arXiv:2408.01319*, 2024.

Yifei Xia, Fangcheng Fu, Wentao Zhang, Jiawei Jiang, and Bin CUI. Efficient multi-task LLM quantization and serving for multiple loRA adapters. In *NeurIPS*, 2024a.

Yuchen Xia, Jiho Kim, Yuhan Chen, Haojie Ye, Souvik Kundu, Cong Hao, and Nishil Talati. Understanding the performance and estimating the cost of LLM fine-tuning. *arXiv preprint arXiv:2408.04693*, 2024b.

Jiahui Yang, Yongjia Ma, Donglin Di, Hao Li, Wei Chen, Yan Xie, Jianxun Cui, Xun Yang, and Wangmeng Zuo. QR-LoRA: Efficient and disentangled fine-tuning via QR decomposition for customized generation. *arXiv preprint arXiv:2507.04599*, 2025.

Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. In *International Conference on Learning Representations (ICLR)*, 2024.

Beiming Yu, Zhenfei Yang, and Xiushuang Yi. MoKA: Parameter efficiency fine-tuning via mixture of Kronecker product adaption. In *International Conference on Computational Linguistics (ICCL)*, 2025.

Tongtian Yue, Longteng Guo, Jie Cheng, Xuange Gao, Hua Huang, and Jing Liu. Ada-K routing: Boosting the efficiency of MoE-based LLMs. In *ICLR*, 2025.

Hanqing Zeng, Yinglong Xia, Zhuokai Zhao, Gilbert Jiang, Qiang Zhang, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, and Benyu Zhang. S'more: Structural mixture of residual experts for llm fine-tuning. *arXiv preprint arXiv:2504.06426*, 2025.

## A IMPROVING THE STABLE RANK OF ADDITIVE ADAPTERS

While this work aims to show that multiplicative adapters are remarkably effective in terms of stable rank and performance, an interesting question that can be raised is whether it is possible to improve the stable rank of purely additive adapters. To answer this question, we propose a hierarchical PEFT technique, SharedLoRA(see Figure 5, which exhibits a higher stable rank than other solutions.

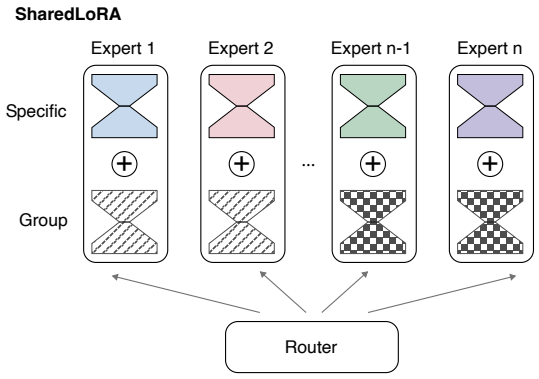
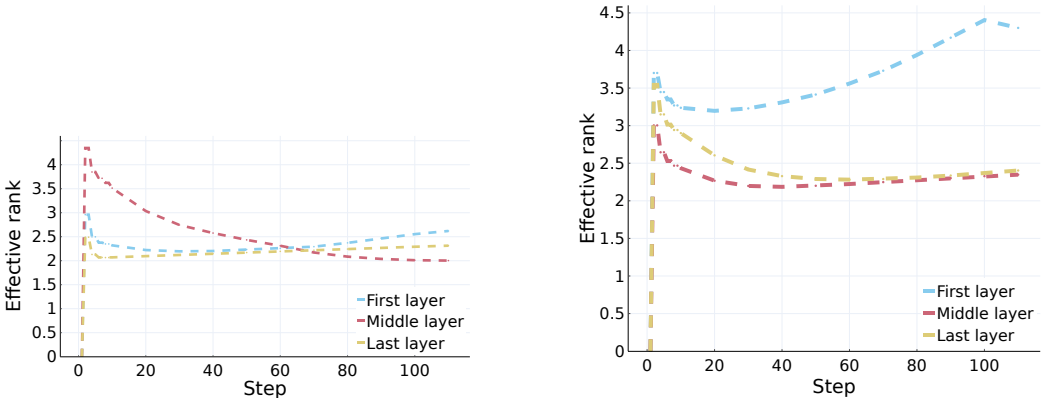



Figure 5: **SharedLoRA.** The experts of SharedLoRA are built by additively combining shared basis LoRAs with expert-specific LoRAs of the same rank. The SharedLoRA rank is therefore the sum of the two ranks.




(a) Stable rank of MixLoRA.

(b) Stable rank of SharedLoRA.

Figure 6: **The stable rank of additive adapters can be improved.** Development of the stable rank of SharedLoRA as finetuning progresses.

SharedLoRA uses a combination of two tiers of low-rank adapters to obtain experts with a higher stable rank without impacting the parameter count. Given a weight matrix $W$ and $n$ desired experts, we first initialize $n$ first-tier (or specific) adapters $A_i^s, B_i^s,\ i \leq n$ of rank $r$. Next, we initialize $g$ second-tier (or group) adapters $A_i^b, B_i^b,\ i \leq g$, with $g$ a divisor of $n$ such that $n = g \times m$. We then assign the first group adapters $A_1^b, B_1^b$ to the first $m$ experts, the second group adapters to the second $m$ experts, and so on. Next, each expert $E_i$ is defined as

$$E_i = A_i^s B_i^s + A_{i/m}^b B_{i/m}^b \tag{10}$$

Together with its noisy top-k router, the input ($\boldsymbol{x}$)-dependent SharedLoRA layer becomes

$$W' = W + \sum_i^n T(\boldsymbol{x})_i (A_i^s B_i^s + A_{i/m}^b B_{i/m}^b) \tag{11}$$

as is usual in MoEs.

Each $E_i$ has a maximum rank of $2r$, for a total number of parameters of $(n+g) \times (M+N) \times r$. In contrast, flat (non-tiered) LoRA MoEs require $2 \times n \times (M+N) \times r$ parameters to achieve the same maximum rank. The two-tier structure results in a $\frac{2n}{n+g} > 1$ (for $g < n$) reduction of parameters at the same rank.

We evaluate SharedLoRA on the same tasks as OperA, and the results can be found in Tables 10 and 16. These results indicate that SharedLoRA is an effective additive adapter that reduces the parameter count while maintaining accuracy. Moreover, Figure 6 shows that SharedLoRA does indeed improve the rank efficiency with respect to competing additive methods.

# B ADAPTER DETAILS

## B.1 OPERA

The hyperparameters for OperA depend on the chosen model, specifically the size of the physical legs. We report in Table 4 both the row legs (Row) and the column legs (Column) in the format $r_1 : r_2 : ... : r_l$ and $c_1 : c_2 : ... : c_l$, with $l$ the number of legs. The number of legs here is always 5, as found in Section 5.2.

Table 4: **Hyperparameters of OperA.** Legs and rank configuration of OperA change depending on the experimental setup.

| Model | Column | Row | Rank | Alpha | Dropout | Results |
|---|---|---|---|---|---|---|
| Llama3-8B | 2:8:32:4:2 | 4:7:32:4:4 | 6 | 12 | 0.2 | Section 5.1 |
| | 2:4:32:8:2 | 4:7:32:4:4 | 6 | 12 | 0.2 | Section 5.1, 5.2, Appendix D.3 |
| Llama3.1-8B | 2:8:32:4:2 | 4:7:32:4:4 | 6 | 12 | 0.2 | Section 5.1 |
| | 2:4:32:8:2 | 4:7:32:4:4 | 6 | 12 | 0.2 | Section 5.1, 5.2, Appendix D.3 |
| Llama3.2-3B | 2:8:12:8:2 | 2:8:32:8:2 | 6 | 12 | 0.2 | Section 5.1 |
| | 2:8:24:4:2 | 2:8:32:8:2 | 6 | 12 | 0.2 | Section 5.1, 5.2, Appendix D.3 |
| Phi4-14B | 2:8:40:4:2 | 4:8:35:4:4 | 6 | 12 | 0.2 | Section 5.1 |
| | 2:5:32:8:2 | 4:7:40:4:4 | 6 | 12 | 0.2 | Section 5.1, 5.2, Appendix D.3 |
| Qwen2.5-3B | 2:8:16:4:2 | 2:8:43:8:2 | 6 | 12 | 0.2 | Section 5.1 |
| | 2:8:16:4:2 | 2:8:43:8:2 | 6 | 12 | 0.2 | Section 5.1, 5.2, Appendix D.3 |
| Qwen2.5-7B | 2:8:28:4:2 | 4:8:37:8:2 | 6 | 12 | 0.2 | Section 5.1 |
| | 2:8:28:4:2 | 2:8:37:8:4 | 6 | 12 | 0.2 | Section 5.1, 5.2, Appendix D.3 |
| Qwen2.5-14B | 2:8:40:4:2 | 4:6:36:4:4 | 6 | 12 | 0.2 | Section 5.1 |
| | 2:5:32:8:2 | 4:6:48:4:3 | 6 | 12 | 0.2 | Section 5.1, 5.2, Appendix D.3 |
| Qwen2.5-32B | 2:8:40:4:2 | 4:6:36:8:4 | 6 | 12 | 0.2 | Section 5.1 |
| | 2:5:32:8:2 | 4:6:48:8:3 | 6 | 12 | 0.2 | Section 5.1, 5.2, Appendix D.3 |

## B.2 ADDITIVE ADAPTERS

We report the hyperparameters chosen for the baseline additive adapters in Table 5. The hyperparameters are chosen based on the findings in MixLoRA (Li et al., 2024), and then further finetuned as described in this work.

Table 5: **Hyperparameters of the additive adapters.** The configuration of the additive adapters change depending on the experimental setup.

| Model | Rank | Alpha | Dropout | Results |
|---|---|---|---|---|
| All | 16 | 32 | 0.05 | Section 5.1 |
| All | 12 | 24 | 0.05 | Appendix D.3 |
| All | 8 | 16 | 0.05 | Section 5.2 |
| All | 7 | 14 | 0.05 | Section 5.1 |

## B.3 SHAREDLORA

We report the hyperparameters chosen for SharedLoRA in Table 5. The hyperparameters are chosen based on an search conducted on the rank and dropout ratio.

Table 6: **Hyperparameters of SharedLoRA.** The configuration of SharedLoRA changes depending on the experimental setup.

| Model | Rank | Alpha | Dropout | Results |
|---|---|---|---|---|
| All | 12 | 24 | 0.2 | Appendix D |
| All | 8 | 16 | 0.2 | Appendix D.3 |

## C Optimized implementation

In the PEFT literature it is often the case that a tradeoff is made between computational complexity and performance, together with parameter efficiency. In Table 7, we evaluate the relative slowdown of our multiplicative approach with respect to the additive baselines. The results show that OperA is $1.8 - 4.1\times$ slower than the additive approaches, depending on the model configuration.

Nonetheless, as mentioned in Section 3.1, OperA can leverage contraction reordering to notably improve both its throughput and memory consumption. We evaluate an initial version of our optimized algorithm using custom contraction orders and *torch.compile* to produce an integrated Triton kernel. First, we benchmark a single layer of each considered model to isolate the effects of this optimization, reporting notable improvements across the board in Table 8, ranging from $1.7\times$ to $2.6\times$. Consistent with the practical evaluation setting, we evaluate both forwards and backwards pass at a batch size of 8 with 8 experts as well.

Using this same efficient implementation, we quantify the peak VRAM consumption of OperA, HiRA (another multiplicative approach), and MixLoRA. Table 9 shows that, despite requiring substantially fewer adapter parameters, OperA does not reduce end-to-end peak VRAM relative to additive adapters in these experiments. Indeed, the primary memory advantage of OperA is its reduced adapter parameter storage, whereas peak VRAM also includes activations, intermediate tensors, and other components of model execution which often dominate, such as with the quadratic attention computation. The contraction-order optimizations considered here instead target the additional intermediate memory introduced by the multiplicative computation.

Moreover, we also separately evaluate a memory efficient and a throughput efficient implementation on a single Llama3.1 8B layer across batch sizes. Figure 7 shows that throughput improves by up to $2.5\times$ and memory consumption by up to $8.9\times$, bringing OperA's performance on par with LoRAMoE and MoLA.

Table 7: **Throughput of PEFT MoEfication.** We report the throughput of all tested models and adapters in seconds per batch.

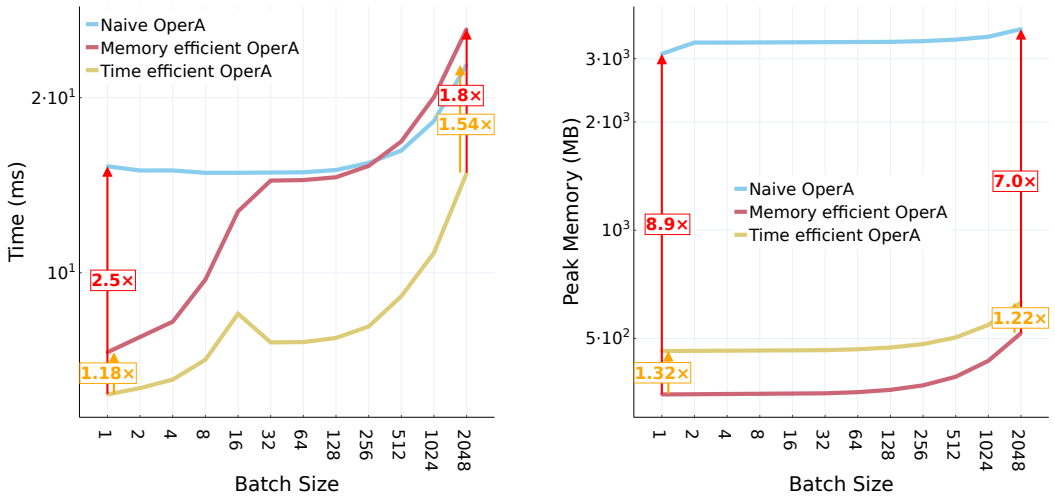
| Model | HydraLoRA | LoRAMoE | MoLA | MixLoRA | SharedLoRA | OperA |
|---|---|---|---|---|---|---|
| Llama3-8B | 0.9108 | 1.3354 | 1.4880 | 0.8456 | 1.0349 | 3.5019 |
| Llama3.1-8B | 0.8870 | 1.1037 | 1.5214 | 0.8423 | 1.1197 | 2.6230 |
| Llama3.2-3B | 0.6281 | 0.7581 | 0.8475 | 0.5028 | 0.7328 | 1.3363 |
| Phi4-14B | 1.3103 | 1.3313 | 1.0475 | 1.0760 | 1.3441 | 5.5511 |
| Qwen2.5-3B | 0.7249 | 0.9529 | 1.3986 | 0.6451 | 0.8959 | 1.5588 |
| Qwen2.5-7B | 0.8898 | 0.9745 | 1.3755 | 0.8194 | 0.9443 | 3.2040 |
| Qwen2.5-14B | 1.5238 | 1.3492 | 1.4427 | 1.2482 | 1.8647 | 4.9189 |
| Qwen2.5-32B | 1.5613 | 1.7872 | 2.0970 | 1.5837 | 1.9863 | 5.1422 |

Table 8: **Throughput of optimized OperA.** We report the throughput in ms of a single layer of all tested models using the naïve implementation of OperA and our optimized DMRG-based implementation.

| Model | Naive | Optimized |
|---|---|---|
| Llama3-3B | 9.78 | 4.08 |
| Llama3.1-8B | 21.08 | 8.57 |
| Llama3.2-8B | 21.08 | 8.57 |
| Phi4-14B | 37.35 | 19.81 |
| Qwen2.5-3B | 8.48 | 3.22 |
| Qwen2.5-7B | 27.37 | 16.01 |
| Qwen2.5-14B | 27.30 | 15.35 |
| Qwen2.5-32B | 58.87 | 32.41 |

Table 9: **Memory consumption of optimized OperA.** We report the peak VRAM consumption in GB for a subset of the tested models using our optimized DMRG-based implementation.

| Model | OperA | HiRA | MixLoRA |
|---|---|---|---|
| Llama3.1-8B | 34.86GB | 32.10GB | 30.60GB |
| Qwen2.5-7B | 35.46GB | 32.46GB | 31.66GB |




(a) Throughput of the naïve implementation of OperA against a speed-optimized and a memory-optimized version.

(b) Memory usage of the naïve implementation of OperA against a speed-optimized and a memory-optimized version.

Figure 7: **Optimized implementations of OperA.** Optimized implementations of OperA based on DMRG-like reordering provide notable benefits both in throughput and memory consumption.

# D ADDITIONAL RESULTS

## D.1 FULL COMPARISON WITH ADDITIVE ADAPTERS

We report here the complete set of results in the single-task scenario including all additive adapters. We also evaluate the accuracy of our two-tier additive schema, SharedLoRA, using the same setup as Section 5.1. In fact, we have seen in Appendix A that SharedLoRA increases the stable rank of the additive adapters while reducing the parameter count. Table 10 shows that our hypothesis holds in practice, with SharedLoRA having a lower parameter count than other adapters while maintaining high performance.

Table 10: **Accuracy of all tested adapters.** For each model, we report the number of parameters and the task accuracy.

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | PMQA ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Llama3-8B | None | / | 66.04 | 81.19 | 62.24 | 67.80 | 59.27 | 48.43 | 67.80 | 54.40 | 63.40 |
| | HydraLoRA | 241M | 76.62 | 81.65 | 83.79 | 81.20 | 70.80 | 92.78 | 79.79 | 54.70 | 77.67 |
| | LoRAMoE | 246M | 77.56 | 86.62 | 87.60 | 87.80 | 74.33 | 95.21 | 84.29 | 73.10 | 83.31 |
| | MoLA | 346M | 76.62 | 88.13 | 85.75 | 85.20 | 75.23 | 92.30 | 82.00 | 73.00 | 82.28 |
| | MixLoRA | 241M | 75.76 | 87.08 | 86.18 | 89.00 | 73.61 | 94.63 | 83.11 | 55.00 | 80.55 |
| | SharedLoRA | 224M | 75.51 | 86.28 | 86.02 | 86.60 | 74.13 | 94.03 | 82.79 | 54.70 | 80.01 |
| | OperA | 173M | 77.39 | 86.28 | 88.30 | 87.60 | 76.36 | 96.15 | 81.45 | 74.10 | **83.45** |
| Llama3.1-8B | None | / | 68.34 | 81.19 | 67.08 | 69.20 | 57.37 | 55.35 | 52.72 | 58.30 | 63.69 |
| | HydraLoRA | 241M | 77.65 | 84.09 | 82.37 | 81.00 | 70.92 | 91.09 | 81.93 | 55.60 | 78.08 |
| | LoRAMoE | 246M | 79.52 | 85.50 | 87.32 | 86.20 | 74.46 | 92.50 | 82.32 | 73.20 | 82.63 |
| | MoLA | 346M | 77.65 | 86.78 | 86.45 | 84.80 | 74.37 | 93.76 | 83.58 | 72.50 | 82.49 |
| | MixLoRA | 241M | 76.96 | 86.82 | 86.40 | 83.20 | 74.65 | 94.77 | 84.14 | 54.40 | 80.17 |
| | SharedLoRA | 224M | 76.11 | 87.21 | 86.18 | 83.00 | 74.43 | 94.52 | 83.74 | 74.40 | 82.45 |
| | OperA | 173M | 76.88 | 86.32 | 87.92 | 85.20 | 75.23 | 96.10 | 80.58 | 74.70 | **82.87** |
| Llama3.2-3B | None | / | 53.41 | 66.79 | 63.82 | 55.00 | 57.61 | 45.98 | 51.93 | 55.30 | 56.23 |
| | HydraLoRA | 131M | 67.52 | 83.71 | 81.56 | 78.20 | 71.87 | 90.07 | 74.61 | 70.70 | 77.28 |
| | LoRAMoE | 133M | 67.92 | 83.29 | 82.97 | 77.00 | 71.68 | 90.62 | 74.22 | 69.90 | 77.20 |
| | MoLA | 200M | 68.52 | 81.94 | 82.86 | 78.00 | 71.96 | 90.33 | 74.30 | 70.10 | 77.25 |
| | MixLoRA | 131M | 68.26 | 83.12 | 83.02 | 77.40 | 71.50 | 92.13 | 75.14 | 68.10 | 77.33 |
| | SharedLoRA | 121M | 69.28 | 83.29 | 84.00 | 79.00 | 72.20 | 91.70 | 76.72 | 69.50 | **78.21** |
| | OperA | 92M | 66.98 | 81.73 | 83.19 | 77.20 | 70.98 | 92.93 | 74.19 | 71.10 | 77.29 |
| Phi4-14B | None | / | 79.95 | 84.76 | 90.04 | 79.40 | 68.72 | 87.19 | 77.74 | 72.90 | 80.09 |
| | HydraLoRA | 336M | 91.38 | 95.20 | 90.86 | 90.80 | 75.50 | 95.53 | 89.50 | 73.00 | 87.72 |
| | LoRAMoE | 342M | 92.02 | 94.11 | 92.44 | 94.00 | 74.57 | 95.61 | 87.53 | 71.40 | 87.71 |
| | MoLA | 389M | 91.38 | 97.10 | 91.46 | 93.80 | 76.97 | 96.30 | 88.40 | 73.40 | **88.60** |
| | MixLoRA | 336M | 91.21 | 94.28 | 91.90 | 92.60 | 75.47 | 95.95 | 88.79 | 73.40 | 87.95 |
| | SharedLoRA | 314M | 90.36 | 95.20 | 92.44 | 92.00 | 76.18 | 96.12 | 88.32 | 75.50 | 88.27 |
| | OperA | 182M | 88.14 | 92.84 | 91.84 | 91.00 | 74.49 | 95.25 | 86.27 | 70.60 | 86.30 |
| Qwen2.5-3B | None | / | 77.65 | 87.58 | 75.46 | 75.00 | 63.64 | 72.08 | 57.38 | 58.80 | 70.95 |
| | HydraLoRA | 188M | 79.99 | 88.17 | 84.33 | 85.40 | 69.97 | 90.29 | 77.74 | 69.20 | 80.64 |
| | LoRAMoE | 192M | 80.11 | 88.34 | 85.58 | 85.30 | 70.70 | 91.57 | 78.93 | 70.50 | 81.38 |
| | MoLA | 246M | 80.03 | 88.26 | 85.64 | 86.00 | 70.92 | 92.44 | 77.24 | 69.90 | 81.30 |
| | MixLoRA | 188M | 80.12 | 88.64 | 86.62 | 87.60 | 70.92 | 91.67 | 80.03 | 70.40 | **82.00** |
| | SharedLoRA | 175M | 79.95 | 88.05 | 86.23 | 87.40 | 71.47 | 91.96 | 81.21 | 68.80 | 81.88 |
| | OperA | 74M | 80.89 | 89.40 | 86.02 | 87.60 | 70.06 | 92.99 | 74.82 | 70.60 | 81.55 |
| Qwen2.5-7B | None | / | 85.07 | 91.33 | 81.99 | 84.60 | 63.09 | 86.00 | 58.64 | 63.10 | 76.73 |
| | HydraLoRA | 253M | 86.18 | 91.13 | 86.51 | 90.00 | 71.83 | 95.20 | 84.37 | 53.10 | 82.29 |
| | LoRAMoE | 258M | 87.03 | 91.79 | 89.83 | 92.00 | 74.22 | 95.30 | 84.93 | 71.60 | 85.84 |
| | MoLA | 332M | 86.18 | 92.34 | 88.96 | 91.20 | 72.81 | 94.95 | 83.03 | 71.30 | 85.10 |
| | MixLoRA | 253M | 87.88 | 94.19 | 88.74 | 92.80 | 74.52 | 95.29 | 83.27 | 52.70 | 83.67 |
| | SharedLoRA | 235M | 87.03 | 92.04 | 88.36 | 92.20 | 74.43 | 94.85 | 85.16 | 52.90 | 83.37 |
| | OperA | 192M | 87.71 | 93.39 | 90.42 | 91.40 | 74.74 | 95.91 | 83.66 | 72.50 | **86.22** |
| Qwen2.5-14B | None | / | 89.42 | 93.10 | 88.79 | 86.80 | 70.61 | 91.12 | 69.22 | 67.60 | 82.08 |
| | HydraLoRA | 376M | 89.94 | 92.98 | 91.24 | 93.80 | 76.70 | 96.25 | 90.84 | 54.30 | 85.76 |
| | LoRAMoE | 384M | 90.66 | 93.42 | 91.46 | 96.20 | 76.69 | 97.01 | 91.32 | 74.10 | 88.86 |
| | MoLA | 568M | 91.47 | 93.44 | 91.12 | 95.20 | 75.75 | 95.50 | 91.08 | 74.20 | 88.47 |
| | MixLoRA | 376M | 90.78 | 93.73 | 92.27 | 95.60 | 77.58 | 96.82 | 90.21 | 53.10 | 86.26 |
| | SharedLoRA | 348M | 90.96 | 93.86 | 91.84 | 95.60 | 76.94 | 94.81 | 90.84 | 73.80 | 88.58 |
| | OperA | 200M | 90.61 | 93.73 | 93.42 | 96.00 | 78.10 | 97.43 | 90.13 | 75.20 | **89.33** |
| Qwen2.5-32B | None | / | 92.24 | 94.23 | 89.77 | 89.80 | 70.03 | 94.21 | 77.35 | 62.10 | 83.72 |
| | HydraLoRA | 841M | 93.00 | 95.20 | 93.63 | 96.20 | 76.61 | 96.06 | 91.08 | 74.70 | 89.64 |
| | LoRAMoE | 858M | 93.43 | 94.32 | 94.23 | 96.40 | 78.26 | 97.24 | 92.23 | 53.40 | 87.37 |
| | MoLA | 1104M | 93.60 | 94.74 | 93.74 | 97.20 | 77.43 | 97.42 | 91.24 | 51.00 | 87.05 |
| | MixLoRA | 841M | 93.26 | 95.75 | 94.07 | 97.00 | 78.35 | 97.29 | 90.92 | 73.20 | 89.98 |
| | SharedLoRA | 783M | 93.86 | 95.62 | 93.04 | 96.20 | 78.26 | 97.15 | 91.48 | 43.90 | 86.19 |
| | OperA | 343M | 93.94 | 94.49 | 94.29 | 97.40 | 79.51 | 97.38 | 90.53 | 74.20 | **90.22** |

### D.2 MULTI-SEED COMPARISON WITH MIXLORA

To provide more statistically robust results, we run 4 seeds of OperA and MixLoRA in the same setup as Appendix D.2. Table 11 shows that the advantage of OperA remains stable even with multi-seed evaluation.

Table 11: **Accuracy of PEFT MoEfication on 4 seeds.** For each model, we report the number of parameters, the task accuracy (%), and the standard deviation.

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| Llama3.1-8B | MixLoRA | 241M | 76.94 ± 0.42 | 87.10 ± 1.65 | 86.79 ± 0.56 | 84.45 ± 1.14 | 74.15 ± 0.71 | 94.57 ± 0.20 | 84.63 ± 2.00 | 84.09 ± 0.55 |
| | OperA | 173M | 77.41 ± 0.50 | 87.10 ± 0.63 | 88.13 ± 0.27 | 86.20 ± 1.05 | 75.71 ± 0.88 | 95.88 ± 0.21 | 82.60 ± 1.29 | 84.72 ± 0.51 |
| Qwen2.5-7B | MixLoRA | 253M | 87.23 ± 0.85 | 92.79 ± 1.32 | 89.08 ± 0.36 | 91.33 ± 2.04 | 74.36 ± 0.33 | 95.03 ± 0.44 | 84.61 ± 1.16 | 87.78 ± 0.37 |
| | OperA | 192M | 87.85 ± 0.25 | 92.48 ± 0.89 | 90.57 ± 0.30 | 92.40 ± 0.92 | 74.78 ± 0.37 | 95.52 ± 0.33 | 83.74 ± 0.75 | 88.19 ± 0.04 |

### D.3 RANK OPTIMIZATION

We perform an hyperparameter search on Llama3-8B and Qwen2.5-3B to obtain the optimal rank of MixLoRA on more recent architectures. Our rank sweep is more fine grained that in the original work, and our results indicate that substantial space savings can be achieved with rank 12 (see Section 5.1 for the ablation results). We extend this evaluation to the other additive baselines, and in parallel we reduce the parameters of OperA and SharedLoRA to produce comparable results. The results in Table 12 indicate that OperA is stable and amenable to different configurations, providing evidence for its flexibility. This is especially remarkable given the multiplicative nature of our method, which shows similar robustness to already established algorithms such as LoRA while employing a notably different technique. Specifically, the performance of OperA in Table 12 is, similarly to the main results with a higher rank, equal or superior to the additive baselines, with even more impressive space savings of $58\%$ against MixLoRA, the most efficient additive adapter once again.

Table 12: **Accuracy of rank-optimized MixLoRA, SharedLoRA, and OperA**. For each model, we report the number of parameters and the task accuracy.

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | PMQA ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Llama3-8B | HydraLoRA | 181M | 76.62 | 84.13 | 83.95 | 83.40 | 72.75 | 93.21 | 81.45 | 75.60 | 81.39 |
| | LoRAMoE | 186M | 78.67 | 85.86 | 86.94 | 86.20 | 74.98 | 95.78 | 82.95 | 74.90 | **83.29** |
| | MoLA | 262M | 77.05 | 86.07 | 86.18 | 87.00 | 74.74 | 94.94 | 84.29 | 74.70 | 83.12 |
| | MixLoRA | 181M | 77.13 | 87.84 | 85.90 | 87.20 | 74.71 | 94.70 | 83.35 | 74.50 | 83.17 |
| | SharedLoRA | 149M | 78.92 | 86.24 | 86.02 | 86.60 | 74.13 | 94.92 | 83.19 | 74.00 | 83.00 |
| | OperA | 95M | 77.64 | 86.41 | 87.60 | 84.20 | 74.74 | 95.71 | 83.03 | 76.10 | 83.18 |
| Llama3.1-8B | HydraLoRA | 181M | 75.34 | 85.73 | 85.15 | 82.60 | 74.37 | 92.87 | 83.19 | 75.60 | 81.86 |
| | LoRAMoE | 186M | 78.07 | 87.88 | 88.79 | 84.60 | 74.40 | 95.64 | 83.43 | 74.50 | 83.41 |
| | MoLA | 262M | 78.41 | 87.21 | 86.29 | 86.60 | 74.98 | 94.50 | 84.93 | 74.40 | 83.42 |
| | MixLoRA | 181M | 76.11 | 85.27 | 87.76 | 84.20 | 75.50 | 94.93 | 84.61 | 52.20 | 80.07 |
| | SharedLoRA | 149M | 76.62 | 87.46 | 86.62 | 86.60 | 74.65 | 94.92 | 84.29 | 74.40 | 83.20 |
| | OperA | 95M | 77.73 | 87.29 | 88.19 | 85.40 | 75.69 | 95.91 | 74.70 | 83.03 | **83.49** |
| Llama3.2-3B | HydraLoRA | 98M | 65.19 | 84.34 | 83.08 | 78.80 | 70.83 | 91.21 | 77.35 | 70.90 | 77.71 |
| | LoRAMoE | 101M | 67.49 | 84.68 | 83.90 | 80.00 | 72.32 | 92.42 | 75.69 | 71.80 | 78.54 |
| | MoLA | 152M | 68.77 | 82.87 | 82.92 | 78.20 | 70.37 | 92.35 | 78.53 | 73.20 | 78.40 |
| | MixLoRA | 98M | 69.28 | 84.85 | 83.57 | 79.80 | 72.20 | 92.53 | 73.09 | 72.10 | 78.43 |
| | SharedLoRA | 81M | 68.00 | 82.66 | 83.79 | 79.00 | 72.20 | 92.87 | 76.24 | 71.00 | 78.22 |
| | OperA | 59M | 69.08 | 84.64 | 82.64 | 79.40 | 71.53 | 93.60 | 75.16 | 73.00 | **78.63** |
| Phi4-14B | HydraLoRA | 252M | 92.83 | 93.86 | 91.24 | 91.20 | 74.80 | 95.28 | 89.03 | 72.30 | 87.57 |
| | LoRAMoE | 258M | 91.47 | 92.90 | 93.04 | 92.40 | 75.54 | 96.36 | 88.63 | 75.50 | 88.23 |
| | MoLA | 294M | 92.58 | 97.31 | 91.57 | 93.60 | 75.93 | 95.96 | 88.56 | 76.80 | **89.04** |
| | MixLoRA | 252M | 90.36 | 96.84 | 91.89 | 94.00 | 75.41 | 95.81 | 87.32 | 76.20 | 88.48 |
| | SharedLoRA | 209M | 90.87 | 96.70 | 91.78 | 92.80 | 75.72 | 96.01 | 88.08 | 76.50 | 88.56 |
| | OperA | 96M | 87.29 | 92.80 | 90.91 | 91.00 | 75.17 | 95.38 | 87.21 | 75.10 | 86.86 |
| Qwen2.5-3B | HydraLoRA | 141M | 80.20 | 88.89 | 85.20 | 86.80 | 70.24 | 91.78 | 78.22 | 68.70 | 81.25 |
| | LoRAMoE | 145M | 80.97 | 89.35 | 85.36 | 86.60 | 70.98 | 92.69 | 79.79 | 70.70 | 82.06 |
| | MoLA | 186M | 79.61 | 88.85 | 84.49 | 88.60 | 71.56 | 92.02 | 78.22 | 68.20 | 81.44 |
| | MixLoRA | 141M | 81.06 | 88.59 | 86.56 | 86.40 | 71.28 | 91.94 | 79.08 | 70.60 | 81.94 |
| | SharedLoRA | 117M | 81.57 | 89.31 | 85.96 | 87.60 | 71.56 | 92.16 | 79.32 | 70.30 | **82.22** |
| | OperA | 74M | 80.89 | 89.40 | 86.02 | 87.60 | 70.06 | 92.99 | 77.82 | 71.80 | 82.07 |
| Qwen2.5-7B | HydraLoRA | 190M | 86.01 | 90.95 | 88.25 | 91.80 | 75.00 | 93.67 | 84.06 | 52.90 | 82.83 |
| | LoRAMoE | 195M | 87.20 | 92.00 | 90.15 | 91.20 | 74.80 | 95.32 | 84.69 | 72.60 | 86.00 |
| | MoLA | 251M | 87.46 | 93.98 | 89.61 | 92.60 | 74.43 | 95.22 | 83.50 | 51.50 | 83.54 |
| | MixLoRA | 190M | 87.54 | 92.17 | 88.90 | 92.80 | 74.10 | 95.60 | 84.14 | 53.00 | 83.53 |
| | SharedLoRA | 157M | 87.03 | 91.54 | 89.99 | 92.20 | 72.90 | 95.10 | 84.14 | 73.10 | 85.75 |
| | OperA | 93M | 87.63 | 92.51 | 90.59 | 94.00 | 74.07 | 95.76 | 83.98 | 72.20 | **86.34** |
| Qwen2.5-14B | HydraLoRA | 283M | 89.59 | 93.56 | 91.68 | 94.60 | 75.75 | 96.34 | 90.69 | 53.10 | 85.66 |
| | LoRAMoE | 290M | 91.21 | 93.60 | 93.20 | 96.60 | 77.25 | 97.04 | 89.50 | 73.00 | 88.93 |
| | MoLA | 430M | 91.21 | 93.76 | 93.20 | 96.60 | 77.25 | 97.04 | 89.50 | 65.60 | 88.02 |
| | MixLoRA | 282M | 91.21 | 93.77 | 92.93 | 95.80 | 77.86 | 96.92 | 90.06 | 72.90 | 88.93 |
| | SharedLoRA | 232M | 91.56 | 93.60 | 92.76 | 96.20 | 77.30 | 97.14 | 89.82 | 72.90 | 88.91 |
| | OperA | 153M | 91.04 | 93.81 | 93.36 | 96.80 | 77.92 | 97.30 | 89.90 | 74.60 | **89.34** |
| Qwen2.5-32B | HydraLoRA | 632M | 93.26 | 95.75 | 93.36 | 97.20 | 77.74 | 97.11 | 91.00 | 75.20 | 90.08 |
| | LoRAMoE | 649M | 93.43 | 94.82 | 93.85 | 96.60 | 78.35 | 97.20 | 91.08 | 75.70 | 90.13 |
| | MoLA | 835M | 93.69 | 94.57 | 93.42 | 96.80 | 79.20 | 97.09 | 91.95 | 75.60 | 90.29 |
| | MixLoRA | 632M | 93.26 | 94.40 | 93.09 | 96.40 | 78.69 | 97.02 | 91.55 | 75.10 | 89.94 |
| | SharedLoRA | 523M | 93.77 | 94.61 | 93.96 | 96.00 | 78.90 | 97.15 | 92.03 | 74.20 | 90.08 |
| | OperA | 330M | 93.77 | 95.83 | 93.91 | 96.80 | 78.32 | 97.39 | 91.79 | 75.70 | **90.44** |

### D.4 MULTI-SEED COMPARISON WITH OPTIMIZED MIXLORA

To provide more statistically robust results, we run 4 seeds of OperA and MixLoRA in the same setup as Appendix D.3. Table 13 shows that the advantage of OperA remains stable even with multi-seed evaluation, and is not an artifact.

Table 13: **Accuracy of rank-optimized PEFT MoEfication on 4 seeds.** For each model, we report the number of parameters, the task accuracy (%), and the standard deviation.

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| Llama3.1-8B | MixLoRA | 106M | 77.84 ± 0.44 | 87.03 ± 0.86 | 86.20 ± 2.00 | 85.60 ± 1.48 | 75.25 ± 0.51 | 94.85 ± 0.40 | 84.13 ± 0.78 | 84.41 ± 0.14 |
| | OperA | 95M | 77.97 ± 0.19 | 86.81 ± 0.82 | 88.03 ± 0.62 | 85.65 ± 0.38 | 75.24 ± 0.42 | 95.66 ± 0.32 | 83.41 ± 2.34 | 84.68 ± 0.34 |
| Qwen2.5-7B | MixLoRA | 111M | 87.14 ± 0.77 | 92.09 ± 0.44 | 88.96 ± 0.43 | 91.75 ± 1.91 | 74.12 ± 1.05 | 94.93 ± 0.45 | 85.52 ± 0.28 | 87.79 ± 0.29 |
| | OperA | 93M | 87.56 ± 0.57 | 92.08 ± 0.45 | 90.62 ± 0.22 | 93.00 ± 0.78 | 74.09 ± 1.04 | 95.55 ± 0.25 | 84.10 ± 0.80 | 88.14 ± 0.24 |

### D.5 MULTI-TASK

We evaluate OperA also in a multi-task scenario to further test the rank-efficiency of our methods. This task includes all previous tasks at the same time, again following the setup of MoE-PEFT (Li et al., 2024). As shown in Table 14, OperA once more surpasses all additive alternatives in average accuracy with a notable reduction in parameters.

Table 14: **Accuracy of all tested adapters on multiple tasks simultaneously.** For each model, we report the number of parameters and the task accuracy.

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | PMQA ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Llama3-8B | HydraLoRA | 241M | 64.08 | 62.71 | 52.56 | 25.80 | 47.09 | 24.22 | 50.28 | 54.30 | 47.63 |
| | LoRAMoE | 246M | 79.44 | 86.91 | 88.25 | 86.80 | 75.78 | 95.18 | 83.27 | 73.40 | 83.63 |
| | MoLA | 346M | 75.77 | 86.91 | 88.63 | 85.20 | 74.86 | 94.14 | 83.11 | 74.30 | 82.87 |
| | MixLoRA | 241M | 76.45 | 86.99 | 88.14 | 82.20 | 75.23 | 94.24 | 83.43 | 54.60 | 80.16 |
| | SharedLoRA | 224M | 77.65 | 86.91 | 87.65 | 86.20 | 75.72 | 94.21 | 85.48 | 67.00 | 82.60 |
| | OperA | 173M | 78.13 | 87.42 | 88.68 | 86.60 | 74.56 | 96.00 | 84.16 | 73.90 | **83.68** |
| Llama3.1-8B | HydraLoRA | 241M | 72.01 | 85.65 | 82.10 | 79.40 | 70.12 | 86.69 | 67.32 | 74.10 | 77.17 |
| | LoRAMoE | 246M | 79.01 | 86.99 | 88.74 | 84.80 | 75.99 | 94.53 | 84.45 | 74.30 | 83.60 |
| | MoLA | 346M | 78.58 | 86.95 | 88.03 | 84.80 | 74.46 | 94.48 | 84.61 | 73.80 | 83.21 |
| | MixLoRA | 241M | 76.71 | 86.45 | 88.19 | 85.80 | 73.24 | 94.76 | 80.43 | 54.50 | 80.01 |
| | SharedLoRA | 224M | 78.07 | 86.62 | 87.92 | 82.60 | 73.39 | 95.01 | 80.98 | 72.90 | 82.19 |
| | OperA | 173M | 78.88 | 87.96 | 88.36 | 86.00 | 74.79 | 95.95 | 84.00 | 73.80 | **83.72** |
| Llama3.2-3B | HydraLoRA | 131M | 61.09 | 77.82 | 77.81 | 73.20 | 66.21 | 89.73 | 64.56 | 72.30 | 72.84 |
| | LoRAMoE | 133M | 67.41 | 84.51 | 84.98 | 77.80 | 71.44 | 93.67 | 75.45 | 73.30 | 78.57 |
| | MoLA | 200M | 67.49 | 84.39 | 84.17 | 80.20 | 71.13 | 92.89 | 76.48 | 72.50 | 78.66 |
| | MixLoRA | 131M | 68.09 | 83.92 | 83.62 | 78.80 | 70.49 | 92.40 | 77.03 | 71.20 | 78.19 |
| | SharedLoRA | 121M | 69.20 | 85.14 | 83.95 | 77.20 | 71.19 | 92.32 | 77.19 | 71.60 | 78.47 |
| | OperA | 92M | 69.62 | 84.72 | 83.46 | 79.20 | 70.76 | 93.71 | 76.32 | 72.30 | **78.76** |
| Phi4-14B | HydraLoRA | 336M | 90.53 | 93.22 | 49.35 | 95.20 | 77.25 | 96.07 | 87.53 | 53.00 | 80.27 |
| | LoRAMoE | 342M | 91.47 | 94.11 | 92.66 | 92.60 | 76.97 | 96.15 | 87.69 | 75.70 | 88.42 |
| | MoLA | 389M | 92.41 | 95.41 | 91.95 | 93.40 | 77.16 | 96.09 | 89.58 | 75.80 | 88.98 |
| | MixLoRA | 336M | 91.89 | 94.91 | 91.89 | 93.40 | 75.50 | 95.93 | 88.40 | 75.80 | 88.47 |
| | SharedLoRA | 314M | 90.78 | 97.56 | 91.89 | 92.00 | 76.15 | 96.03 | 89.19 | 74.50 | 88.51 |
| | OperA | 182M | 92.53 | 96.27 | 91.99 | 93.00 | 77.10 | 97.03 | 88.77 | 75.50 | **89.02** |
| Qwen2.5-3B | HydraLoRA | 188M | 81.31 | 89.65 | 49.51 | 86.40 | 69.17 | 25.30 | 74.59 | 70.30 | 68.28 |
| | LoRAMoE | 192M | 82.51 | 89.86 | 86.11 | 87.40 | 70.46 | 92.34 | 79.16 | 69.30 | 82.14 |
| | MoLA | 246M | 80.20 | 88.80 | 86.34 | 85.00 | 71.28 | 92.88 | 78.69 | 69.90 | 81.64 |
| | MixLoRA | 188M | 80.89 | 88.76 | 86.60 | 88.00 | 71.44 | 93.29 | 78.37 | 70.20 | 82.19 |
| | SharedLoRA | 175M | 80.55 | 88.89 | 87.21 | 87.40 | 71.04 | 92.70 | 77.58 | 71.20 | 82.07 |
| | OperA | 74M | 80.38 | 89.35 | 86.72 | 88.60 | 70.60 | 93.42 | 78.98 | 70.10 | **82.27** |
| Qwen2.5-7B | HydraLoRA | 253M | 87.97 | 91.79 | 87.76 | 90.60 | 69.45 | 91.86 | 75.69 | 50.90 | 80.75 |
| | LoRAMoE | 258M | 87.37 | 91.96 | 89.77 | 92.60 | 74.95 | 95.24 | 84.21 | 72.70 | 86.10 |
| | MoLA | 332M | 86.01 | 91.96 | 91.35 | 91.60 | 75.29 | 94.69 | 85.87 | 49.40 | 83.27 |
| | MixLoRA | 253M | 87.88 | 93.94 | 90.64 | 91.40 | 75.44 | 94.63 | 85.08 | 52.30 | 83.91 |
| | SharedLoRA | 235M | 87.54 | 91.88 | 90.32 | 91.40 | 74.80 | 94.65 | 84.69 | 71.80 | 85.89 |
| | OperA | 192M | 87.71 | 92.13 | 91.13 | 92.00 | 74.50 | 95.85 | 83.74 | 72.80 | **86.23** |
| Qwen2.5-14B | HydraLoRA | 376M | 92.15 | 94.07 | 93.31 | 95.20 | 77.92 | 97.39 | 91.00 | 73.20 | 89.28 |
| | LoRAMoE | 384M | 91.30 | 93.73 | 92.93 | 94.80 | 78.01 | 97.12 | 89.74 | 74.10 | 88.97 |
| | MoLA | 568M | 91.64 | 93.52 | 93.14 | 94.20 | 77.31 | 97.08 | 90.84 | 74.40 | 89.02 |
| | MixLoRA | 376M | 90.96 | 93.73 | 92.11 | 95.20 | 78.38 | 96.75 | 89.58 | 74.00 | 88.84 |
| | SharedLoRA | 348M | 90.61 | 93.35 | 92.60 | 96.60 | 77.92 | 97.04 | 90.29 | 75.00 | 89.18 |
| | OperA | 200M | 91.90 | 93.73 | 93.10 | 95.50 | 78.11 | 96.90 | 90.55 | 74.70 | **89.31** |

**Rank optimized setup.** As before, we use the rank-optimized version of the additive adapters to confirm the performance of our models. In Table 15, OperA surpasses all additive alternatives here as well.

Table 15: **Accuracy of rank-optimized MixLoRA, SharedLoRA, and OperA on multiple tasks at once.** For each model, we report the number of parameters and the task accuracy.

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | PMQA ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Llama3-8B | HydraLoRA | 181M | 73.46 | 85.02 | 78.67 | 79.60 | 62.45 | 90.61 | 71.90 | 72.20 | 76.74 |
| | LoRAMoE | 186M | 77.41 | 87.37 | 88.90 | 86.80 | 75.81 | 95.45 | 83.98 | 73.10 | 83.60 |
| | MoLA | 262M | 77.22 | 86.99 | 86.94 | 81.60 | 73.09 | 95.02 | 84.69 | 75.00 | 82.57 |
| | MixLoRA | 181M | 77.73 | 86.36 | 88.68 | 85.80 | 74.80 | 94.18 | 83.35 | 73.10 | 83.00 |
| | SharedLoRA | 149M | 76.96 | 87.88 | 87.92 | 86.60 | 73.82 | 94.97 | 82.64 | 73.80 | 83.07 |
| | OperA | 95M | 77.13 | 87.33 | 89.28 | 86.40 | 75.35 | 95.54 | 84.29 | 74.30 | **83.70** |
| Llama3.1-8B | HydraLoRA | 181M | 74.15 | 85.90 | 81.77 | 80.80 | 71.35 | 90.80 | 73.88 | 75.30 | 79.24 |
| | LoRAMoE | 186M | 78.07 | 88.47 | 88.96 | 85.80 | 73.79 | 95.50 | 83.98 | 73.70 | 83.53 |
| | MoLA | 262M | 77.22 | 86.99 | 86.94 | 81.60 | 73.09 | 95.02 | 84.69 | 75.00 | 82.57 |
| | MixLoRA | 181M | 78.07 | 86.49 | 87.16 | 83.00 | 73.06 | 95.16 | 82.08 | 54.70 | 79.97 |
| | SharedLoRA | 149M | 77.47 | 86.53 | 88.57 | 84.80 | 73.36 | 94.97 | 82.48 | 74.50 | 82.84 |
| | OperA | 95M | 76.54 | 86.20 | 88.74 | 86.80 | 75.14 | 96.23 | 83.50 | 75.80 | **83.62** |
| Llama3.2-3B | HydraLoRA | 98M | 61.69 | 80.22 | 79.33 | 76.40 | 68.29 | 90.72 | 67.48 | 72.10 | 74.53 |
| | LoRAMoE | 101M | 68.60 | 83.50 | 85.36 | 78.60 | 71.19 | 93.53 | 75.14 | 73.80 | 78.72 |
| | MoLA | 152M | 68.09 | 84.43 | 84.33 | 79.60 | 70.70 | 92.63 | 77.11 | 69.30 | 78.27 |
| | MixLoRA | 98M | 69.80 | 84.72 | 85.26 | 79.40 | 70.31 | 92.12 | 77.58 | 71.60 | 78.85 |
| | SharedLoRA | 81M | 68.77 | 83.29 | 85.15 | 77.60 | 70.00 | 92.84 | 78.69 | 72.80 | 78.64 |
| | OperA | 59M | 67.96 | 84.51 | 84.68 | 79.00 | 70.76 | 93.46 | 77.69 | 73.20 | **78.91** |
| Phi4-14B | HydraLoRA | 252M | 91.55 | 93.43 | 91.29 | 92.40 | 74.19 | 93.99 | 83.74 | 75.00 | 86.95 |
| | LoRAMoE | 258M | 91.64 | 94.15 | 91.82 | 92.80 | 76.61 | 96.00 | 86.58 | 76.60 | 88.28 |
| | MoLA | 294M | 91.98 | 96.76 | 91.73 | 94.40 | 75.50 | 96.16 | 87.92 | 75.50 | 88.74 |
| | MixLoRA | 252M | 91.13 | 95.29 | 92.71 | 94.20 | 76.85 | 95.94 | 88.24 | 75.10 | 88.68 |
| | SharedLoRA | 209M | 90.87 | 93.94 | 91.51 | 92.60 | 76.88 | 96.02 | 88.40 | 76.10 | 88.29 |
| | OperA | 96M | 91.44 | 95.55 | 91.73 | 94.80 | 76.60 | 96.79 | 88.30 | 75.50 | **88.84** |
| Qwen2.5-3B | HydraLoRA | 141M | 82.08 | 89.69 | 83.57 | 88.20 | 68.72 | 92.20 | 74.27 | 68.80 | 80.94 |
| | LoRAMoE | 145M | 81.66 | 90.03 | 86.29 | 87.00 | 70.21 | 93.38 | 79.16 | 69.90 | 82.20 |
| | MoLA | 186M | 80.46 | 89.65 | 86.29 | 86.00 | 70.43 | 92.61 | 79.24 | 71.90 | 82.07 |
| | MixLoRA | 141M | 81.06 | 89.27 | 86.89 | 87.80 | 70.34 | 92.05 | 78.93 | 71.10 | 82.18 |
| | SharedLoRA | 117M | 81.23 | 89.98 | 86.83 | 87.20 | 70.52 | 92.50 | 78.30 | 70.90 | 82.18 |
| | OperA | 74M | 80.38 | 89.35 | 86.72 | 87.60 | 69.60 | 93.42 | 79.98 | 71.10 | **82.27** |
| Qwen2.5-7B | HydraLoRA | 190M | 88.65 | 92.26 | 89.23 | 91.20 | 70.95 | 25.20 | 76.01 | 71.40 | 75.61 |
| | LoRAMoE | 195M | 86.52 | 92.30 | 90.81 | 92.60 | 74.59 | 95.63 | 85.08 | 71.00 | 86.07 |
| | MoLA | 251M | 86.86 | 92.34 | 90.70 | 93.20 | 75.44 | 94.87 | 83.90 | 53.30 | 83.83 |
| | MixLoRA | 190M | 87.80 | 91.54 | 89.45 | 91.00 | 74.77 | 94.92 | 85.64 | 53.20 | 83.54 |
| | SharedLoRA | 157M | 87.12 | 92.17 | 90.42 | 92.40 | 75.14 | 95.07 | 84.45 | 70.00 | 85.85 |
| | OperA | 93M | 87.46 | 91.92 | 90.75 | 93.20 | 75.41 | 95.91 | 83.90 | 72.00 | **86.32** |
| Qwen2.5-14B | HydraLoRA | 283M | 91.89 | 93.52 | 92.33 | 94.40 | 77.19 | 86.90 | 87.37 | 52.80 | 84.55 |
| | LoRAMoE | 290M | 91.47 | 93.43 | 92.76 | 95.20 | 77.22 | 97.44 | 89.98 | 73.00 | 88.81 |
| | MoLA | 430M | 91.64 | 93.27 | 93.09 | 94.60 | 77.37 | 97.02 | 90.69 | 53.30 | 86.37 |
| | MixLoRA | 282M | 91.47 | 93.43 | 92.49 | 95.40 | 77.77 | 97.00 | 90.77 | 74.90 | 89.15 |
| | SharedLoRA | 232M | 90.44 | 93.90 | 92.98 | 96.00 | 78.04 | 97.13 | 89.58 | 74.90 | 89.12 |
| | OperA | 153M | 91.64 | 93.33 | 92.78 | 95.90 | 77.37 | 97.22 | 90.97 | 74.50 | **89.21** |

### D.6 RANK ABLATION

Table 16 reports all the results of the rank ablation performed on Qwen2.5-3B and Llama3-7B. Too high and too low a rank harm performance, with 12 being the optimal rank.

Table 16: **Results of the rank ablation**. We sweep the rank of MixLoRA from 16 to 8 on Qwen2.5-3B and Llama3-8B, to determine the optimal configuration.

| Adapter | Model | Rank ↓ | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|
| MixLoRA | Qwen2.5-3B | 16 | 188M | 80.12 | 88.64 | 86.62 | 87.60 | 70.92 | **82.78** |
| | | 12 | 142M | 81.06 | 88.59 | 86.56 | 86.40 | 71.28 | **82.78** |
| | | 8 | 95M | 81.91 | 89.01 | 83.19 | 87.60 | 71.59 | 82.66 |
| MixLoRA | Llama3-8B | 16 | 241M | 75.76 | 87.08 | 86.18 | 89.00 | 73.61 | 82.33 |
| | | 12 | 181M | 77.13 | 87.84 | 85.90 | 87.20 | 74.71 | **82.56** |
| | | 8 | 121M | 78.84 | 87.33 | 83.59 | 86.80 | 75.30 | 82.37 |

### D.7 MPO SIZE ABLATION

Table 17 reports all the results of the ablation on the number of legs of the MPO performed on Qwen2.5-3B. A moderate number of legs, 5, provides the best performance.

Table 17: **Results of the MPO size ablation**. We sweep the number of legs in OperA from 3 to 7 on Llama3-8B and Qwen2.5-3B, to determine the optimal configuration.

| Adapter | Model | Column | Row | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| OperA | Llama3-8B | 2:2:4:32:2:2:2 | 2:2:7:32:2:2:4 | 172M | 77.22 | 86.24 | 87.65 | 86.80 | 76.39 | 82.84 |
| | | 2:8:32:4:2 | 4:7:32:4:4 | 172M | 77.39 | 86.28 | 88.30 | 87.60 | 76.36 | **83.19** |
| | | 16:32:8 | 28:32:16 | 172M | 78.07 | 86.36 | 87.38 | 87.40 | 75.54 | 82.95 |
| OperA | Qwen2.5-3B | 2:2:4:16:2:2:2 | 2:2:4:43:4:2:2 | 74M | 80.55 | 89.52 | 85.53 | 88.20 | 69.82 | 82.72 |
| | | 2:8:16:4:2 | 2:8:43:8:2 | 74M | 80.89 | 89.40 | 86.02 | 87.60 | 70.06 | **82.79** |
| | | 16:16:8 | 16:43:16 | 74M | 80.03 | 89.35 | 85.53 | 88.60 | 69.02 | 82.51 |

### D.8 OperA components

We analyze the impact of the various components of OperA in isolation. First, the adapter fusion approach of MixLoRA (Li et al., 2024) has been shown to notably improve the performance of additive adapters with LoRA. In Table 18 we report that such approach is helpful for multiplicative adapters such as OperA as well. Next, we verify whether our weight-derived initialization and the auxiliary loss added to the top-k MoE routing also have a positive effect. Table 19 confirms that this is the case.

Table 18: **Results of the fusion ablation**. We evaluate whether the adapter fusion approach is successful for OperA as well on Llama3-8B and Qwen2.5-3B.

| Model | Adapter | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|
| Llama3-8B | OperA (w/o fusion) | 73.46 | 84.30 | 87.11 | 86.6 | 74.07 | 81.11 |
| | OperA | 77.64 | 86.41 | 87.60 | 84.20 | 74.74 | **82.11** |
| Qwen2.5-3B | OperA (w/o fusion) | 76.54 | 87.84 | 82.92 | 85.80 | 69.17 | 80.45 |
| | OperA | 80.89 | 89.40 | 86.02 | 87.60 | 70.06 | **82.79** |

Table 19: **Results of the components ablation**. For each model, we report the number of parameters and the task accuracy (%).

| Model | Adapter | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|
| Llama3.1-8B | OperA w/ rand. aux. init. | 69.03 | 82.37 | 85.91 | 81.8 | 72.57 | 95.12 | 77.03 | 80.55 |
| | OperA w/o aux loss | 77.73 | 87.50 | 88.52 | 86.80 | 75.05 | 95.68 | 82.08 | 84.12 |
| | OperA | 77.73 | 86.66 | 87.43 | 85.20 | 75.05 | 96.20 | 81.06 | **84.19** |
| Qwen2.5-7B | OperA w/ rand. aux. init. | 86.35 | 91.50 | 88.74 | 90.60 | 73.18 | 95.48 | 81.97 | 86.83 |
| | OperA w/o aux loss | 87.12 | 92.09 | 89.77 | 92.00 | 75.08 | 93.32 | 83.98 | 87.62 |
| | OperA | 87.63 | 92.51 | 90.59 | 94.00 | 74.07 | 95.76 | 83.98 | **88.36** |

### D.9 Non-MoE LoRA performance

We also evaluate the performance of LoRA without experts, to verify whether the tasks are simple enough not to require them. Table 20 shows that when given similar parameter budget, LoRA without MoE notably underperforms OperA.

Table 20: **Accuracy of non-MoE LoRA on the downstream tasks.** For each model, we report the number of parameters and the task accuracy (%).

| Model | Adapter | No. params ↓ | ARC-c ↑ | ARC-e ↑ | PIQA ↑ | OBQA ↑ | BoolQ ↑ | HS ↑ | WG ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| Llama3.1-8B | LoRA | 106M | 77.56 | 87.37 | 70.89 | 66.40 | 68.53 | 90.92 | 54.38 | 73.72 |
| | OperA | 95M | 77.73 | 86.66 | 87.43 | 85.20 | 75.05 | 96.20 | 81.06 | 84.19 |
| Qwen2.5-7B | LoRA | 111M | 86.86 | 93.43 | 82.48 | 83.40 | 73.73 | 92.46 | 58.01 | 81.48 |
| | OperA | 93M | 87.63 | 92.51 | 90.59 | 94.00 | 74.07 | 95.76 | 83.98 | 88.39 |

# E RELATED ADAPTERS

Given the large literature surrounding PEFT adapter, we dedicate this Appendix to clarify the connection between some of the most recent works related to OperA. In particular, in Table 21 we compare OperA with LoRA, LoHA, HiRA, ABBA-Adapters, KRAdapter, and MPOE. In particular, while some adapters are constructed multiplicatively, most are only applied additively as in the LoRA formulation. In contrast, MPOE is also applied multiplicatively, but is not parameter-efficient as it requires the same computational complexity as full fine-tuning. Additionally, in contrast to MPOE, OperA keeps the residual path as the shared expert, and fuses the adapters with the weights.

Table 21: **Comparison of related adapters.** We compare OperA with relevant related adapters. OperA is the only PEFT method that both constructs and applies multiplicatively.

| | **MoE** | **PEFT** | **Construction** | **Trainable comp.** | **Application** |
|---|---|---|---|---|---|
| LoRA | No | Yes | Matrix multiplication (MM) | Adapters | Addition |
| LoHA | No | Yes | MM + Hadamard product | Adapters | Addition |
| HiRA | No | Yes | MM + Hadamard product | Adapters | Addition |
| ABBA-Adapters | No | Yes | MM + Hadamard product | Adapters | Addition |
| KRAdapter | No | Yes | Khatri-Rao product | Adapters | Addition |
| MPOE | Yes | No | TT-SVD (only once) | Center + auxiliary | Contraction |
| **OperA** | Yes | Yes | TT-SVD (only once) | Auxiliary | Contraction |