MoORE: SVD-based Model MoE-ization for Conflict- and Oblivion-Resistant Multi-Task Adaptation

Shen Yuan[1,2] **Yin Zheng**[2] **Taifeng Wang**[2] **Binbin Liu**[2] **Hongteng Xu**[1,3,4*]
[1]Gaoling School of Artificial Intelligence, Renmin University of China [2]ByteDance
[3]Beijing Key Laboratory of Research on Large Models and Intelligent Governance
[4]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE

Abstract

Adapting large-scale foundation models in multi-task scenarios often suffers from task conflict and oblivion. To mitigate such issues, we propose a novel "model MoE-ization" strategy that leads to a conflict- and oblivion-resistant multi-task adaptation method. Given a weight matrix of a pre-trained model, our method applies SVD to it and introduces a learnable router to adjust its singular values based on tasks and samples. Accordingly, the weight matrix becomes a Mixture of Orthogonal Rank-one Experts (MoORE), in which each expert corresponds to the outer product of a left singular vector and the corresponding right one. We can improve the model capacity by imposing a learnable orthogonal transform on the right singular vectors. Unlike low-rank adaptation (LoRA) and its MoE-driven variants, MoORE guarantees the experts' orthogonality and maintains the column space of the original weight matrix. These two properties make the adapted model resistant to the conflicts among the new tasks and the oblivion of its original tasks, respectively. Experiments on various datasets demonstrate that MoORE outperforms existing multi-task adaptation methods consistently, showing its superiority in terms of conflict- and oblivion-resistance. The code is available at `https://github.com/DaShenZi721/MoORE`.

1 Introduction

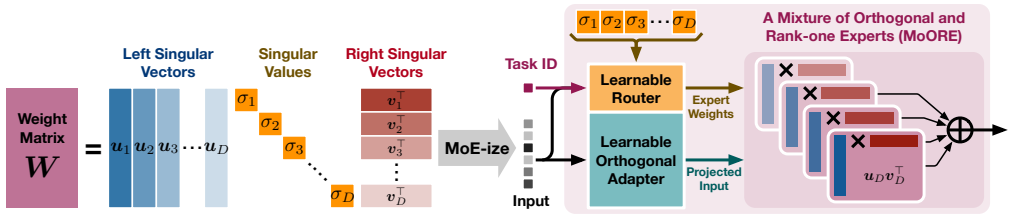
Parameter-efficient adaptation/fine-tuning [71] plays a central role in the practical deployment of large-scale pre-training models, especially in multi-task learning (MTL) scenarios [80, 61, 11, 62, 42]. Take large language models (LLMs) [82] in natural language processing (NLP) tasks as an example. To increase intrinsic knowledge and maintain generalization power, a pre-trained LLM often needs to learn multiple downstream tasks in different domains simultaneously or sequentially. To achieve multi-task adaptation, parameter-efficient fine-tuning (PEFT) methods like low-rank adaptation (LoRA) [25] and its variants [68, 2, 60, 65, 31] have been proposed. However, the practical applications of these methods are often limited because they often suffer from **task conflict and oblivion**: i) The model adapted for one task often leads to the performance degradation in other tasks (also known as negative transfer [70, 79] or destructive interference [11]). ii) Learning new tasks may result in catastrophic forgetting [62] — the model performance deteriorates severely in previously learned tasks.

Essentially, task conflict and oblivion arise because the diversity of different tasks requires the models to adapt their parameters in different directions [72, 74, 13, 26, 15]. Some recent methods [31, 30, 60] combine the Mixture-of-Experts (MoE) architecture [27, 55, 29] with PEFT, mitigating the interferences across different tasks by activating task-specific parameters. Given a layer of a pre-trained model, these methods learn multiple adapters associated with a router in multi-task scenarios.

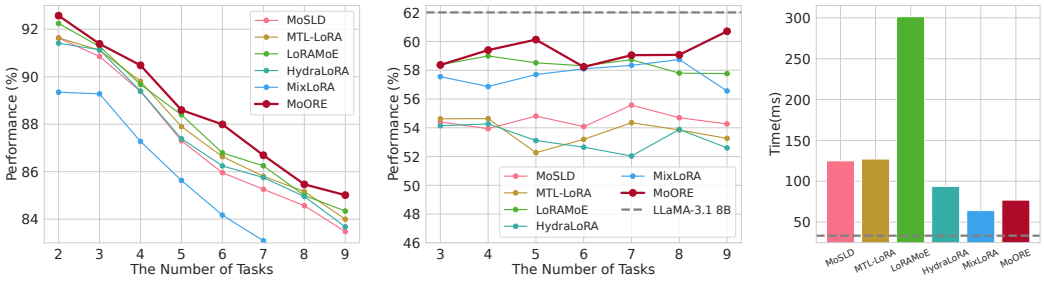
*Corresponding author. Email: hongtengxu313@gmail.com

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-1795



(a) An illustration of our SVD-based model MoE-ization strategy and MoORE architecture



(b) Resistance to task conflict (c) Resistance to task oblivion (d) Inference efficiency

Figure 1: (a) An illustration of our model MoE-ization strategy and the corresponding MoORE architecture. (b) The comparison for various multi-task adaptation methods on fine-tuning LLaMA-3.1 8B [20] on the CSR-MTL constructed by nine tasks [9, 8, 45, 5, 54, 77, 53, 59]. MoORE consistently works better than the baselines when the number of tasks is larger than one. (c) Before adaptation, LLaMA-3.1 8B achieves encouraging overall performance (i.e., the gray dashed line) in seven tasks [24, 23, 83, 58, 51, 7, 4, 10]. MoORE mitigates the performance degradation and outperforms the most competitive baseline LoRAMoE [14]. (d) MoORE's runtime is comparable to that of its competitors. Compared to the original LLaMA-3.1 8B (i.e., the gray dashed line), MoORE increases the inference time moderately.

Each adapter may inherit task-specific knowledge, and the router selects/fuses these adapters based on the input data and tasks.

In principle, we call the above strategy "**Model MoE-ization**" because it converts the original linear layer to an MoE architecture. In this study, we propose a novel model MoE-ization strategy, with the help of the singular value decomposition (SVD), leading to a conflict- and oblivion-resistant multi-task adaptation method. As illustrated in Figure 1(a), given a weight matrix of a pre-trained model, our method applies SVD [1] to it and introduces a learnable router to adjust its singular values based on tasks and samples. Accordingly, the weight matrix becomes a Mixture of Orthogonal Rank-one Experts (**MoORE**), where each expert is constructed by the outer product of a left singular vector and the corresponding right one. To further improve the capacity of MoORE, we impose a learnable orthogonal adapter, which is implemented by a Householder reflection adaptation module [75].

Unlike existing methods, which learn some additional low-rank experts without any constraints on their relations, MoORE extracts many rank-one experts intrinsically from the SVD of the original weight matrix. Such a design *imposes the orthogonality of the experts*, avoiding their information redundancy and undesired interferences. Moreover, similar to existing orthogonal fine-tuning strategies [49, 75], MoORE *maintains the column space of the original weight matrix* and thus inherit the pre-training ability better. Thanks to the above two properties, MoORE consistently outperforms SOTA methods in various multi-task adaptation scenarios. The results in Figure 1 show the superiority of MoORE in mitigating task conflict and oblivion and its competitive inference efficiency.

2 Related Work and Preliminaries

2.1 Adapter-based Methods for Multi-Task Adaptation

Multi-task adaptation aims to fine-tune a pre-trained foundation model simultaneously or sequentially in multiple downstream tasks [34, 32]. Focusing on this problem, many adapter-based methods have been proposed [37, 50, 57, 65, 40]. In particular, denote the data of K tasks as $\{\mathcal{D}_k\}_{k=1}^{K}$. Given a

pre-trained foundation model, whose parameters are denoted as $\boldsymbol{\theta}$, multi-task adaptation can often be formulated in the framework of maximum likelihood estimation:

$$\max_{\Delta\boldsymbol{\theta}} \sum_{k=1}^{K} P_{\boldsymbol{\theta}\cup\boldsymbol{\Delta\theta}}(\mathcal{D}_k), \tag{1}$$

where $P_{\boldsymbol{\theta}\cup\boldsymbol{\Delta\theta}}(\mathcal{D}_k)$ denotes the likelihood of the k-th task's data, which is parameterized by the original model parameters $\boldsymbol{\theta}$ and the adapters' parameters $\Delta\boldsymbol{\theta}$.

Unlike learning and ensembling different models, adapter-based methods improve the overall model performance by sharing parameters and domain knowledge across various tasks, leading to moderate increases in parameters and complexity. For instance, Hyperformer [44] utilizes a shared hyper-network to generate task-specific adapters, reducing the number of learnable parameters. Recently, some methods extend LoRA [25] to multi-task adaptation, e.g., MultiLoRA [68], MTL-LoRA [73], HydraLoRA [60], and so on, which learns multiple low-rank adapters to handle diverse tasks.

2.2 Connections to MoE Architectures

The Mixture-of-Experts (MoE) architecture was initially introduced by the work in [27], which is constructed by multiple specialized networks (called experts) and a router. Given an input data $\boldsymbol{x} \in \mathcal{X}$, the MoE derives the output $\boldsymbol{y} \in \mathcal{Y}$ as $\boldsymbol{y} = \sum_{m=1}^{M} g_m(\boldsymbol{x}) f_m(\boldsymbol{x})$, where $f_m : \mathcal{X} \mapsto \mathcal{Y}$ denotes the m-th expert, which achieves a mapping from the sample space $\mathcal{X}$ to the output space $\mathcal{Y}$. $g : \mathcal{X} \mapsto \mathbb{R}^M$ denotes the router, and g_m denotes its m-th output. The router adjusts the experts' significance based on input data. When applying a sparse routing strategy, i.e., activating only a few experts for each input [55, 29, 17, 3, 16, 46], the MoE architecture supports building large-scale models while maintaining computational efficiency. Due to its advantages, many large language models, e.g., DeepSeek [12], Grok3, and Qwen3[2], are built based on MoE, and many efforts have been made to convert well-trained dense models into MoE architectures [28, 22, 52, 47, 78].

As mentioned before, most existing adapter-based multi-task adaptation methods actually "MoE-ize" pre-trained models. Given a weight matrix $\boldsymbol{W}$ and its input $\boldsymbol{x}$, these methods apply multiple low-rank adapters as experts [76, 14, 35, 41] and add them to the pre-trained models, i.e.,

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \sum_{m=1}^{M} g_m(\boldsymbol{x}) \boldsymbol{B}_m \boldsymbol{A}_m \boldsymbol{x}, \tag{2}$$

where $\boldsymbol{A}_m$ and $\boldsymbol{B}_m$ are low-rank matrices constructing the m-th expert f_m. For the router $g(\boldsymbol{x})$, some attempts have been made to develop advanced routing strategies, e.g., the dynamic routing in AdaMoLE [38] and the token-task hybrid routing in HMoRA [31]. Recently, some new MoE architectures have been proposed, including the asymmetric "Hydra" structure in HydraLoRA [60], the LEMoE [66] for lifelong model editing, and the OMoE [18] for orthogonal output. However, the above methods rely purely on data-driven strategies to determine the experts' functionality and domain knowledge. Without necessary regularization on the relations across different experts, the experts often suffer from the load imbalance issue, i.e., a limited number of experts are over-trained and applied for multiple tasks, while many experts are seldom used. This issue harms the capacity and generalizability of the models, increasing the risk of task conflict and oblivion.

3 Proposed Method

In this study, we propose a new model MoE-ization method for multi-task adaptation. In principle, our method imposes orthogonality on the experts and further regularizes their output spaces, which helps mitigate task conflict and oblivion.

3.1 SVD-based Model MoE-ization

Consider a pre-trained weight matrix $\boldsymbol{W} \in \mathbb{R}^{D_{\text{out}} \times D}$. Without loss of generality, we assume $D_{\text{out}} \geq D$ and $\text{Rank}(\boldsymbol{W}) = D$. The SVD of the matrix is denoted as

$$\boldsymbol{W} = \boldsymbol{U}\text{diag}(\boldsymbol{\sigma})\boldsymbol{V}^\top = \sum_{d=1}^{D} \underbrace{\sigma_d}_{\text{weight}} \cdot \underbrace{(\boldsymbol{u}_d \boldsymbol{v}_d^\top)}_{\text{expert}}, \tag{3}$$

[2] https://github.com/QwenLM/Qwen3

https://doi.org/10.52202/085713-1795

where $\boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_D] \in \mathbb{R}^{D_{\text{out}} \times D}$ contains left singular vectors, $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_D] \in \mathbb{R}^{D \times D}$ contains right singular vectors, and $\boldsymbol{\sigma} = [\sigma_1, \cdots, \sigma_D]^\top$ is a vector of singular values.

As shown in (3), there exists an intrinsic but static MoE architecture hidden in the SVD of the weight matrix — each $\boldsymbol{W}$ corresponds to the mixture of D orthogonal and rank-one experts, in which the d-th expert is the outer product of $\boldsymbol{u}_d$ and $\boldsymbol{v}_d$ and its weight is fixed as σ_d. Motivated by this intrinsic MoE, our method derives the proposed MoORE model for multi-task adaptation, which reuses the experts while introducing the following two modifications:

- **A hybrid routing strategy:** Inspired by HMoRA [31], we adjust the experts' weights according to input data and tasks, leading to a hybrid routing strategy. Given an input of the k-th task, denoted as $\boldsymbol{x}^{(k)} \in \mathbb{R}^d$, we determines the weight of the d-th expert as

$$g_d(\boldsymbol{x}^{(k)}) = \underbrace{\boldsymbol{p}_d^\top \boldsymbol{t}_k}_{\text{task-level}} + \underbrace{\boldsymbol{q}_d^\top \boldsymbol{\Gamma} \boldsymbol{x}^{(k)}}_{\text{sample-level}}, \tag{4}$$

  where $\boldsymbol{t}_k \in \mathbb{R}^{D_t}$ denotes the embedding of the k-th task, and the vector $\boldsymbol{p}_d \in \mathbb{R}^{D_t}$ projects the task embedding to a task-level weight. Similarly, applying the matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{D_s \times D}$ and the vector $\boldsymbol{q}_d \in \mathbb{R}^{D_s}$, we project the input $\boldsymbol{x}^{(k)}$ to a sample-level weight. In practice, we set $D_s, D_t \ll D$ to reduce the router's parameters and computational cost. The final weight $g_d(\boldsymbol{x}^{(k)})$ is the summation of the task- and sample-level weights.

- **An orthogonal adapter of input:** To further increase the model capacity, we can apply a learnable orthogonal transform to the input, i.e., $\boldsymbol{H}\boldsymbol{x}$, where the learnable orthogonal transform $\boldsymbol{H}$ can be implemented efficiently by the butterfly orthogonal fine-tuning (BOFT) module in [36], the Givens rotation adapter [43], or the Householder reflection adapter (HRA) [75]. In this study, we implement $\boldsymbol{R}$ by HRA, which corresponds to the product of L Householder reflections, i.e.,

$$\boldsymbol{H} = \prod_{\ell=1}^{L} \Big(\boldsymbol{I} - \frac{1}{\|\boldsymbol{r}_\ell\|_2^2} \boldsymbol{r}_\ell \boldsymbol{r}_\ell^\top \Big), \tag{5}$$

  whose learnable parameters are $\boldsymbol{R} = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_L] \in \mathbb{R}^{D \times L}$. Note that applying orthogonal adapters maintains the angles between neurons (i.e., the rows of the weight matrix $\boldsymbol{W}$), which helps preserve the knowledge of the pre-trained model when enhancing model capacity [49, 36, 75].

Applying the above SVD-based model MoE-ization strategy, we obtain the proposed MoORE model, which introduces D orthogonal and rank-one experts into the model and encodes the input data as

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{H}\boldsymbol{x}^{(k)} + \sum\nolimits_{d=1}^{D} \underbrace{g_d(\boldsymbol{x}^{(k)})}_{\text{router}} \underbrace{(\boldsymbol{u}_d \boldsymbol{v}_d^\top \boldsymbol{H})}_{\text{expert}} \boldsymbol{x}^{(k)} = \boldsymbol{U}\text{diag}(g(\boldsymbol{x}^{(k)}) + \boldsymbol{\sigma})\boldsymbol{V}^\top \boldsymbol{H}\boldsymbol{x}^{(k)}, \tag{6}$$

where $g(\boldsymbol{x}^{(k)}) = \boldsymbol{P}^\top \boldsymbol{t}_k + \boldsymbol{Q}^\top \boldsymbol{\Gamma} \boldsymbol{x}^{(k)} \in \mathbb{R}^D$.

The learnable parameters of MoORE include $\boldsymbol{T} = [\boldsymbol{t}_k] \in \mathbb{R}^{D_t \times K}$, $\boldsymbol{P} = [\boldsymbol{p}_d] \in \mathbb{R}^{D_t \times D}$, $\boldsymbol{Q} = [\boldsymbol{q}_d] \in \mathbb{R}^{D_s \times D}$, $\boldsymbol{\Gamma} \in \mathbb{R}^{D_s \times D}$, and $\boldsymbol{R} \in \mathbb{R}^{D \times L}$. As shown in (6), given the SVD of $\boldsymbol{W}$ (which can be computed in advance), we can implement MoORE with low complexity via the following steps:

$$1)\ \boldsymbol{z} = \underbrace{\boldsymbol{V}\boldsymbol{H}\boldsymbol{x}^{(k)}}_{\mathcal{O}(D(L+D))}, \quad 2)\ g(\boldsymbol{x}^{(k)}) = \underbrace{\boldsymbol{P}^\top \boldsymbol{t}_k}_{\mathcal{O}(DD_t)} + \underbrace{\boldsymbol{Q}^\top \boldsymbol{\Gamma} \boldsymbol{x}^{(k)}}_{\mathcal{O}(DD_s)}, \quad 3)\ \boldsymbol{y} = \underbrace{\boldsymbol{U}(g(\boldsymbol{x}^{(k)}) + \boldsymbol{\sigma})\boldsymbol{z}}_{\mathcal{O}(DD_{\text{out}})}. \tag{7}$$

As shown in (7), the overall complexity of MoORE is $\mathcal{O}(D(D_{\text{out}} + D + D_t + D_s + L))$. In practice, we set $L, D_t, D_s \ll \min\{D, D_{\text{out}}\}$ to reduce the complexity. Moreover, we can merge the orthogonal adapter into each expert in the inference phase, i.e., obtaining $\boldsymbol{V}' = \boldsymbol{V}^\top \boldsymbol{H}$, and the complexity further reduces to $\mathcal{O}(D(D_{\text{out}} + D + D_t + D_s))$. Figure 1(d) shows that the inference efficiency of MoORE is comparable to its competitors.

3.2 Comparisons with Existing MoE-based Multi-Task Adaptation Methods

Our SVD-based model MoE-ization method provides a new technical route for multi-task adaptation: **Instead of learning an MoE with few strong extrinsic experts, our method constructs an MoE with many simple but structured experts intrinsically based on the pre-trained weight matrix.** Tables 1 and 2 compare different methods on their MoE architectures, theoretical properties, and computational efficiency, respectively.

Table 1: Comparisons for various MoE-based multi-task adaptation methods on their implementations and properties, where M (D in our method) is the number of experts, k is the task index, GS$(\cdot)$ denotes the Gram-Schmidt orthogonalization [6].

Method	Router	Experts	#Experts	Rank of Experts	Orthogonality of Experts	Maintaining Range$(\boldsymbol{W})$
LoRAMoE [14]	Softmax$(\boldsymbol{S}\boldsymbol{x})$	$\{\boldsymbol{B}_m\boldsymbol{A}_m\}_{m=1}^{M}$	M	r	No	No
MixLoRA [30]	Top$_2$(Softmax$(\boldsymbol{S}\boldsymbol{x})$)	$\{\boldsymbol{B}_m\boldsymbol{A}_m\}_{m=1}^{M}$	M	r	No	No
MoSLD [81]	Softmax$(\boldsymbol{S}\boldsymbol{x})$	$\{\boldsymbol{B}\boldsymbol{A}_m\}_{m=1}^{M}$	M	r	No	No
HydraLoRA [60]	Softmax$(\boldsymbol{S}\boldsymbol{x})$	$\{\boldsymbol{B}_m\boldsymbol{A}\}_{m=1}^{M}$	M	r	No	No
MTL-LoRA [73]	Softmax$(\boldsymbol{\phi}_k)$	$\{\boldsymbol{B}_m\boldsymbol{\Lambda}_k\boldsymbol{A}\}_{m=1,k=1}^{M,K}$	MK	r	No	No
OMoE [18]	Softmax$(\boldsymbol{S}\boldsymbol{x})$	GS$(\{\boldsymbol{B}_m\boldsymbol{A}_m\}_{m=1}^{M})$	M	r	Yes	No
MoORE (Ours)	$\boldsymbol{P}^\top\boldsymbol{t}_k + \boldsymbol{Q}^\top\boldsymbol{\Gamma}\boldsymbol{x}$	$\{\boldsymbol{u}_d\boldsymbol{v}_d^\top\boldsymbol{H}\}_{d=1}^{D}$	D	1	Yes	Yes

- **The design of router.** Given an input $\boldsymbol{x} \in \mathbb{R}^D$, most existing methods leverage a sample-level routing strategy. Typically, they apply a linear projection $\boldsymbol{S} \in \mathbb{R}^{M\times D}$ to it and pass the projection result through a softmax operator, leading to a nonnegative and normalized weight vector for M experts. Among them, MixLoRA [30] further applies a sparse routing mechanism — for each input, it only activates the two experts that correspond to the top-2 weights. Instead, MTL-LoRA [73] leverages a task-level routing strategy, which determines the experts' weights by passing a task-specific embedding (i.e., $\boldsymbol{\phi}_k \in \mathbb{R}^M$, $k \in \{1, \cdots, K\}$ indicates the task index) through a softmax operator. Unlike existing methods, our MoORE considers the task- and sample-level information jointly. The advantage of this hybrid routing strategy is that when the same sample serves for different tasks, MoORE can assign task-specific weights to the experts and thus leverage different domain knowledge accordingly.

- **The design of experts.** Most existing methods apply M low-rank adapters as experts, i.e., $\{\boldsymbol{B}_m\boldsymbol{A}_m\}_{m=1}^{M}$, where $\boldsymbol{B} \in \mathbb{R}^{D_{\text{out}}\times r}$ and $\boldsymbol{A} \in \mathbb{R}^{r\times D}$ are two rank-r matrices. To reduce the number of learnable parameters, some methods, e.g., MoSLD [81], HydraLoRA [60], and MTL-LoRA [73], reuse the same $\boldsymbol{A}$ or $\boldsymbol{B}$ for all experts. MTL-LoRA [73] further introduces a task-specific matrix $\boldsymbol{\Lambda}_k$ for each expert, where $k = 1, ..., K$ indicates the task index. As a result, it creates MK low-rank experts and activates M experts per task. Our MoORE contains D orthogonal rank-one experts $\{\boldsymbol{u}_d\boldsymbol{v}_d^\top\boldsymbol{H}\}_{d=1}^{D}$, in which the orthogonal adapter $\boldsymbol{H}$ is shared by all experts. Unlike existing methods, MoORE applies many simple but structured experts. Such a design has several advantages:

  1. **Imposing orthogonality for mitigating task conflict:** The experts of MoORE are orthogonal to each other because $\boldsymbol{u}_d^\top\boldsymbol{u}_{d'} = 0$ for all $d \neq d'$. The orthogonality ensures that the experts have different functionalities and domain knowledge, without redundant information. In particular, by activating different experts for different tasks, MoORE suppresses the interferences across the tasks in the training phase, thus mitigating the task conflict issue. It is worth noting that O-LoRA [67] also introduces orthogonality constraints within its LoRA adapters. However, there are two notable differences between O-LoRA and MoORE: i) O-LoRA uses a regularization term to pursue orthogonality, which can not strictly make its LoRA adapters orthogonal to each other. ii) O-LoRA does not adopt a MoE architecture. To deal with different tasks, O-LoRA adds new LoRA adapters for each sequentially incoming task, rather than designing and learning a flexible routing mechanism for given and fixed experts. As a result, the model complexity of O-LoRA is linear to the number of tasks. Among existing MoE-based methods, OMoE [18] is the only one imposing orthogonality on experts. However, it applies Gram-Schmidt orthogonalization algorithm [6] to the concatenation of M experts' output, i.e., GS$([\boldsymbol{B}_1\boldsymbol{A}_m\boldsymbol{x}, \cdots, \boldsymbol{B}_M\boldsymbol{A}_m\boldsymbol{x}])$. As a result, imposing orthogonality requires additional $\mathcal{O}(D_{\text{out}}M^2)$ operations per sample, which is less efficient than ours.
  2. **Maintaining Range$(\boldsymbol{W})$ for mitigating task oblivion:** The column space of each expert, i.e., Range$(\boldsymbol{u}_d\boldsymbol{v}_d^\top\boldsymbol{H})$, is the same with Range$(\boldsymbol{u}_d)$. Accordingly, the output space of MoORE is the direct sum of $\{\text{Range}(\boldsymbol{u}_d)\}_{d=1}^{D}$, which is the same as $\boldsymbol{W}$'s column space, i.e.,

$$\bigoplus\nolimits_{d=1}^{D} \text{Range}(\boldsymbol{u}_d\boldsymbol{v}_d^\top\boldsymbol{H}) = \bigoplus\nolimits_{d=1}^{D} \text{Range}(\boldsymbol{u}_d) = \text{Range}(\boldsymbol{U}) = \text{Range}(\boldsymbol{W}). \tag{8}$$

https://doi.org/10.52202/085713-1795

Table 2: Comparisons for various MoE-based multi-task adaptation methods on their computational efficiency.

Method	#Learnable Parameters		Computational Complexity
	Router	Experts	
LoRAMoE [14]	MD	$M(D_{\text{out}}+D)r$	$\mathcal{O}(D_{\text{out}}D + M(D + D_{\text{out}}r + Dr))$
MixLoRA [30]	MD	$M(D_{\text{out}}+D)r$	$\mathcal{O}(D_{\text{out}}D + MD + (D_{\text{out}}r + Dr))$
MoSLD [81]	MD	$(D_{\text{out}}+MD)r$	$\mathcal{O}(D_{\text{out}}D + D_{\text{out}}r + M(Dr + D))$
HydraLoRA [60]	MD	$(MD_{\text{out}}+D)r$	$\mathcal{O}(D_{\text{out}}D + Dr + M(D_{\text{out}}r + D))$
MTL-LoRA [73]	MK	$(MD_{\text{out}}+Kr+D)r$	$\mathcal{O}(D_{\text{out}}D + Dr + MD_{\text{out}}r + r^2)$
OMoE [18]	MD	$M(D_{\text{out}}+D)r$	$\mathcal{O}(D_{\text{out}}(D+M^2) + M(D + D_{\text{out}}r + Dr))$
MoORE (Ours)	$D_tK + (D_t + 2D_s)D$	LD	$\mathcal{O}((D_{\text{out}} + D + D_s + D_t + L)D)$

Table 3: Comparisons for various SVD-based fine-tuning methods on their implementations.

Method	Model in Training	Learnable Parameters	Model in Inference
SVF [56]	$\boldsymbol{U}\text{diag}(\sigma)\boldsymbol{V}^\top$	Nonparametric σ	Original architecture: $\boldsymbol{W}_{\text{new}}$
SVDiff [21]	$\boldsymbol{U}\text{diag}(\text{ReLU}(\sigma+\delta))\boldsymbol{V}^\top$	Nonparametric δ	Original architecture: $\boldsymbol{W}_{\text{new}}$
SVFT [33]	$\boldsymbol{U}(\text{diag}(\sigma)+\boldsymbol{M})\boldsymbol{V}^\top$	Nonparametric $\boldsymbol{M}$	Original architecture: $\boldsymbol{W}_{\text{new}}$
KaSA [64]	$\boldsymbol{W}_{lr} + \Delta\boldsymbol{U}\text{diag}(\Delta\sigma)\Delta\boldsymbol{V}^\top$	Nonparametric $\Delta\boldsymbol{U}, \Delta\sigma, \Delta V^\top$	Original architecture: $\boldsymbol{W}_{\text{new}}$
SVFCL [69]	$\boldsymbol{U}\text{diag}(\sigma + \sum_{i=0}^{t}\sigma_i)\boldsymbol{V}^\top$	Nonparametric σ_i's	Original architecture: $\boldsymbol{W}_{\text{new}}$
MoORE (Ours)	$\boldsymbol{W}\boldsymbol{H} + \sum_{d=1}^{D} g_d(\boldsymbol{x}^{(k)})(\boldsymbol{u}_d\boldsymbol{v}_d^\top\boldsymbol{H})$	Parametric $g(\cdot)$ and Householder reflections $\boldsymbol{H}$	An MoE, where $\boldsymbol{W}\boldsymbol{H}$ and $\boldsymbol{v}_d^\top\boldsymbol{H}$ are pre-computed

In single task adaptation scenarios, the work in [75] has shown that maintaining the column space of the weight matrix makes the adapted model inherit the ability of the pre-trained model better, mitigating the oblivion of the previously pre-trained task. In our work, we find that such maintenance is helpful in multi-task adaptation scenarios as well.

- **Computational efficiency.** Table 2 shows each method's learnable parameters and computational complexity. For the methods using the sample-based routing strategy, their routers contain MD learnable parameters. Given a sample, the complexity of the router is $\mathcal{O}(MD)$. For the method [73] using the task-based routing strategy, its router is lightweight, containing MK learnable parameters and determining the weights of its experts with complexity $\mathcal{O}(M)$. For the experts, using M rank-r experts leads to the complexity $\mathcal{O}(M(D_{\text{out}}+D)r)$. Reusing $\boldsymbol{A}$ or $\boldsymbol{B}$ (e.g., MoSLD [81] and HydraLoRA [60]) and applying sparse routing (e.g., MixLoRA [30]) can reduce the computational complexity significantly. In contrast, introducing additional parameters (e.g., the $\boldsymbol{\Lambda}_k$ in MTL-LoRA [73]) or operations (e.g., the Gram-Schmidt orthogonalization in OMoE [18]) leads to higher complexity. Existing methods construct an MoE with M rank-r experts, while our MoORE is an MoE with D rank-one experts, whose number of experts is determined by the input dimension and thus much larger than M. To improve the computational efficiency of MoORE, we set D_s and D_t comparable to the rank r and set L comparable to M, respectively. As a result, the number of learnable parameters and the complexity of MoORE become comparable to most existing methods.

3.3 Comparisons with Existing SVD-based Fine-Tuning Methods

Our method employs SVD to perform model MoE-ization, constructing a complete MoE architecture. This fundamentally differs from existing SVD-based fine-tuning methods [48, 56, 21, 33, 64, 69]. To more clearly illustrate the differences between MoORE and these methods, we select five recent ones for comparison, including SVF [56], SVDiff [21], SVFT [33], KaSA [64], and SVFCL [69]. Tables 3 and 4 compare different methods in terms of implementation and application, respectively.

- **In the aspect of architecture**, MoORE is the only method that learns a parametric router to adjust singular values, leading to an MoE architecture for inference. In MoORE, each expert is constructed by the outer product of singular vectors and the learnable Householder reflections.
- **In the aspect of application**, only MoORE and SVFCL consider multi-task adaptation. MoORE is applicable for both simultaneous and sequential multi-task adaptation, while SVFCL only

Table 4: Comparisons for various SVD-based fine-tuning methods on their applications.

Method	Multi-Task Learning	Consider Task Conflict	Consider Task Oblivion
SVF [56]	No	No	No
SVDiff [21]	No	No	No
SVFT [33]	No	No	No
KaSA [64]	No	No	No
SVFCL [69]	Yes (incremental learning)	No	Partially discussed
MoORE (Ours)	Yes	Yes, by orthogonal experts	Yes, by maintaining Range($\boldsymbol{W}$)

considers incremental learning. Moreover, MoORE makes the first attempt to connect task conflict and oblivion to the orthogonality of experts and the column space of the parameter matrix.

4 Experiments

To demonstrate the effectiveness of MoORE in multi-task adaptation, we apply three MTL datasets and conduct comprehensive experiments on them. Representative results are shown in Figure 1 and the following content. More experimental details, e.g., the basic information of datasets, hyperparameter settings, ablation studies, routing weight analysis, and numerical results associated with figures, are shown in the Appendix.

4.1 Implementation Details

Base model and baselines. In the following experiments, we utilize LLaMA-3.1 8B[3] [20] as the base model and adapt it by various multi-task adaptation methods. In particular, we compare MoORE with LoRA [25] and the methods incorporating low-rank adapters as MoEs, including LoRAMoE [14], MoSLD [81], MTL-LoRA [73], HydraLoRA [60], and MixLoRA [30]. We implement the MoE architectures of the baselines mainly based on their default settings. For a fair comparison, we modify some baselines' hyperparameters to make the number of learnable parameters comparable for each method. For MoORE, we MoE-ize all linear layers of LLaMA-3.1 8B, including the "$QKVO$" modules of attention layers and the weight matrices of FFNs.

Two datasets for evaluating conflict-resistance. We consider two MTL datasets for commonsense reasoning (CSR) and natural language understanding (NLU), respectively. The **CSR-MTL** dataset is constructed by nine tasks, including ARC-Challenge (ARC-C), ARC-Easy (ARC-E) [9], OpenBookQA (OBQA) [45], PIQA [5], SocialIQA (SIQA) [54], BoolQ [8], Hellaswag (HellaS) [77], Winogrande (WinoG) [53], and CommonsenseQA (CSQA) [59]. These tasks are widely used to evaluate LLMs on various commonsense reasoning challenges, ranging from genuine grade-school level science questions to physical commonsense reasoning. The **NLU-MTL** dataset consists of seven tasks from GLUE [63], including CoLA, SST-2, MRPC, QQP, MNLI, QNLI, and RTE. These tasks are applied to evaluate the natural language understanding capabilities of LLMs, including natural language inference, textual entailment, sentiment analysis, semantic similarity, and so on.

One dataset for evaluating oblivion-resistance. In addition, we construct one more dataset, called **OR-MTL**, for evaluating the oblivion-resistance of different methods. The dataset includes seven tasks, including MMLU [24, 23], IFEval [83], BIG-Bench Hard (BBH) [58], GPQA [51], HumanEval [7], MBPP [4], and GSM-8K [10]. The base model, LLaMA-3.1 8B, achieves encouraging performance in these tasks. After adapting it on CSR-MTL, we record the performance of the adapted models in OR-MTL and assess the ability of different methods to mitigate task oblivion.

Experimental settings. When adapting the pre-trained model on CSR-MTL and NLU-MTL, we set the training epoch to 2 and 5, respectively. The learning rate is set to 3×10^{-4}, with AdamW [39] as the optimizer. For CSR-MTL, we set the batch size to 8, whereas for NLU-MTL, we set the batch size to 64. Both training and testing are conducted on one NVIDIA A100 GPU.

[3] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

https://doi.org/10.52202/085713-1795

Table 5: Results (%) of various methods on CSR-MTL. The best results on each dataset are shown in **bold**, and the second best results are shown in underline.

Method	#Params	ARC-C	ARC-E	BoolQ	OBQA	PIQA	SIQA	HellaS	WinoG	CSQA	Overall
Full Fine-tuning	100.00%	80.12	87.92	74.56	87.60	88.96	80.55	**95.91**	81.69	81.16	84.27
LoRA [25]	2.09%	77.56	85.77	70.43	81.60	82.97	76.00	93.00	71.11	77.40	79.54
MixLoRA [30]	3.00%	79.18	87.50	72.02	86.60	87.38	78.86	93.65	77.35	80.43	82.55
MoSLD [81]	1.49%	80.29	86.87	74.16	86.40	88.58	79.73	95.03	80.58	80.34	83.55
HydraLoRA [60]	2.72%	79.27	88.09	74.34	85.60	88.96	79.84	95.36	**83.03**	80.10	83.84
MTL-LoRA [73]	2.69%	80.55	88.38	**75.26**	85.80	88.41	80.45	95.57	81.61	80.75	84.09
LoRAMoE [14]	2.19%	80.97	88.51	74.37	86.20	89.45	**80.91**	95.26	81.29	82.06	84.34
MoORE $_{L=0}$	2.72%	**82.51**	**89.35**	74.59	87.80	89.39	79.89	95.52	81.53	84.22	84.98
MoORE $_{L=2}$	2.75%	81.91	**89.35**	74.56	87.60	**89.83**	80.45	95.46	81.53	84.52	85.02
MoORE $_{L=4}$	2.78%	82.25	89.23	74.74	**88.60**	**89.83**	80.04	95.41	80.98	84.11	85.02
MoORE $_{L=8}$	2.84%	82.25	89.31	74.74	87.80	89.55	79.89	95.48	82.40	**84.60**	**85.11**

Table 6: Results (%) of various methods on NLU-MTL. The best results on each dataset are shown in **bold**, and the second best results are shown in underline. We report the matched accuracy for MNLI, Matthew's correlation for CoLA, and average correlation for STS-B.

Method	#Params	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	Overall
LoRA [25]	2.09%	39.69	86.20	80.88	88.16	87.80	80.14	90.60	79.07
MixLoRA [30]	3.00%	61.71	89.98	82.84	94.47	90.11	85.56	95.53	85.74
MoSLD [81]	1.49%	58.55	90.34	85.29	94.95	90.08	89.53	96.33	86.44
MTL-LoRA [73]	2.69%	60.65	90.26	87.01	95.26	**91.95**	90.98	96.10	87.46
HydraLoRA [60]	2.72%	64.92	89.97	86.76	95.31	91.62	87.73	96.44	87.54
LoRAMoE [14]	2.19%	62.84	**91.01**	88.97	**95.55**	91.01	90.61	**96.56**	88.08
MoORE $_{L=0}$	2.72%	69.09	89.86	**91.91**	94.44	90.73	**91.34**	96.22	**89.08**
MoORE $_{L=2}$	2.75%	**69.23**	90.35	89.70	95.37	90.75	**91.34**	**96.56**	89.04

4.2 Performance in Conflict-Resistance

Tables 5 and 6 compare MoORE with its competitors on CSR-MTL and NLU-MTL, respectively. Using a comparable number of learnable parameters for each dataset, MoORE achieves the best or comparable performance across all tasks and thus obtains the best average performance. These results demonstrate the superiority of MoORE in mitigating task conflict.

The impact of orthogonal adapter. In the experiment on CSR-MTL, we increase the number of Householder reflections in $\boldsymbol{H}$ (i.e., L) from 0 to 8 and find that MoORE exhibits consistent improvements in performance. In the experiment on NLU-MTL, however, applying the orthogonal adapter may not improve performance. In our opinion, this phenomenon indicates that the commonsense reasoning tasks in CSR-MTL require more domain knowledge not covered by the pre-trained model. As a result, introducing the orthogonal adapter increases the number of learnable parameters and enhances the model capacity accordingly. In contrast, the text classification tasks in NLU-MTL rely more on the non-specific natural language knowledge captured by the pre-trained model. Therefore, without introducing more learnable parameters, adjusting the singular values of the pre-trained weight matrix is sufficient to achieve encouraging performance.

Conflict-resistance regarding task number and difficulty. To further compare and analyze the conflict-resistance capabilities of different methods, we conduct comparative experiments on CSR-MTL by varying the number and difficulty of the tasks. In particular, for each task in CSR-MTL, we first calculate the average of all the methods' performance based on the results in Table 5. The average performance of the methods in a task measures the difficulty of the task for the base model — the lower the average performance is, the more difficult the task is. Then, we sort the tasks in ascending order based on their difficulty. Finally, we adapt the base model for the top-K tasks, $K = 2, ..., 9$, and show the performance of different methods in Figure 1(b). With the increase of task number and difficulty, all the methods suffer performance degradation because i) task conflict becomes severe as the number of tasks increases, and ii) difficult tasks are more likely to have conflicts with other tasks. Notably, MoORE outperforms all other baselines across all settings.

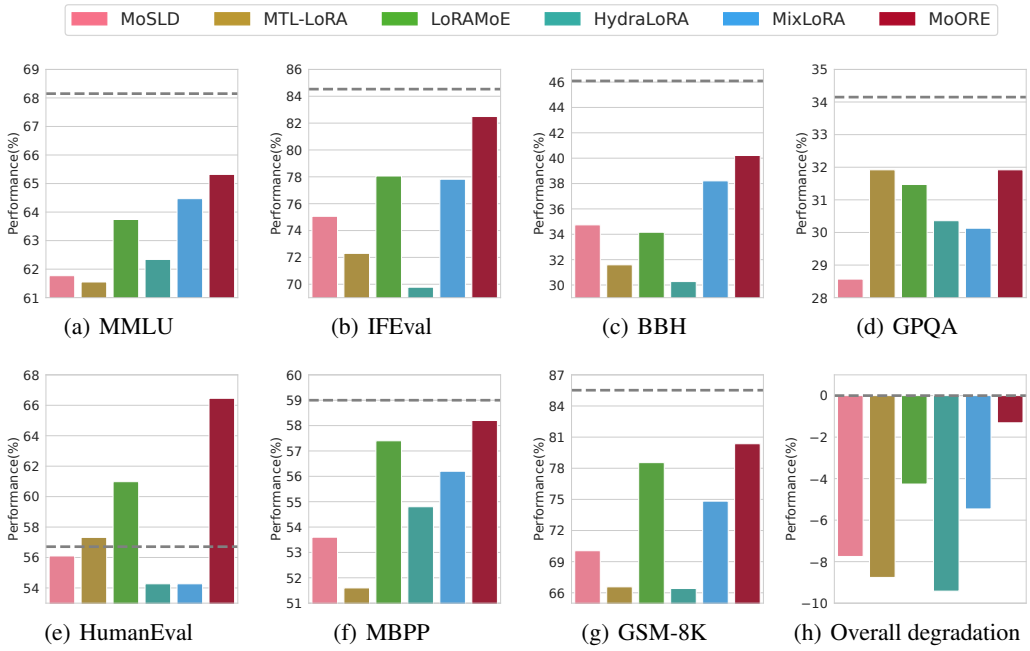


Figure 2: The loss of performance in OR-MTL. (a-g) The gray dashed line represents the performance of LLaMA-3.1 8B before adaptation. (h) The overall performance degradation across all tasks.

4.3 Performance in Oblivion-Resistance

To demonstrate the superiority of MoORE in mitigating task oblivion, we compare various methods in the Language Model Evaluation Harness framework [19]. In particular, we first adapt the base model on CSR-MTL using different methods. Then, we evaluate the adapted models on OR-MTL, comparing them with the base model. Figure 2 shows that MoORE consistently mitigates task oblivion across all seven tasks of OR-MTL, with an average performance drop of only 1.31% compared to the base model. It significantly outperforms the other adaptation methods. On HumanEval, MoORE achieves performance exceeding the original model, with an improvement of 9.75%. Similarly, LoRAMoE and MTL-LoRA also obtain slight improvements of 4.27% and 0.16%, respectively. This intriguing phenomenon may imply that the datasets in CSR-MTL have some relevance to HumanEval, providing information and capabilities beneficial for solving this task.

Oblivion-resistance regarding task number and difficulty. Furthermore, we conduct experiments to investigate how the ability of oblivion-resistance changes with increased task number and difficulty. The results in Figure 1(c) show that after adapting to the top-3 tasks of CSR-MTL, MoORE and its strongest competitor, LoRAMoE [14], achieve comparable performance on OR-MTL. However, with the increase of task number and difficulty, the performance of LoRAMoE changes slightly while the performance of MoORE increases and becomes superior to that of LoRAMoE. This result demonstrates that MoORE has stronger oblivion-resistance than its competitors.

The impact of task correlation. We investigate the reason for MoORE's oblivion-resistance empirically. In particular, we consider the samples of several sub-tasks in MMLU and those of six tasks in CSR-MTL. For each task/sub-task, we compute the average weights of MoORE's experts over its samples, i.e., $\boldsymbol{g}_k = \frac{1}{|\mathcal{D}_k|} \sum_{\boldsymbol{x}^{(k)} \in \mathcal{D}_k} g(\boldsymbol{x}^{(k)})$. Given a sub-task of MMLU and a task of CSR-MTL, denoted as $\mathcal{D}_k$ and $\mathcal{D}_{k'}$, respectively, we measure their correlation by $\|\boldsymbol{g}_k - \boldsymbol{g}_{k'}\|_2$. For each sub-task of MMLU, we record the performance degradation of MoORE compared to the base model. Figure 3 shows normalized performance degradation and task correlation. This visualization result indicates that the oblivion-resistance arises from the correlation of tasks — for correlated tasks, the model can learn some common domain knowledge during adaptation and thus avoid catastrophic forgetting.

The generalization to unseen task IDs. The computation of task-level weights requires task IDs as input, but the ID of a new task is unavailable during inference. Therefore, we use the average of

https://doi.org/10.52202/085713-1795

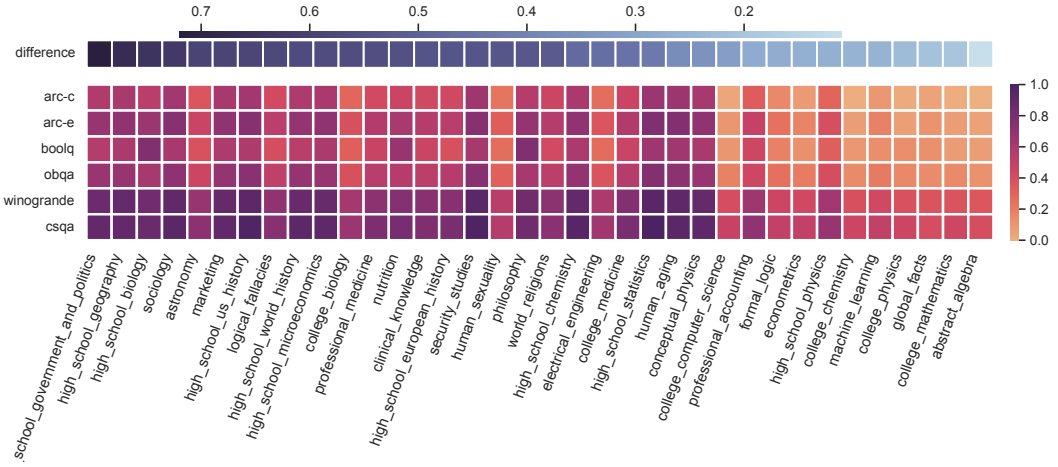


Figure 3: The visualization of normalized performance degradation and task correlation. The "difference" shown in the first row is the normalized performance degradation, i.e., $(\text{Acc}_{\text{Base}} - \text{Acc}_{\text{MoORE}})/100\%$. The following matrix records the normalized task correlation. The element in the j-th row and the i-th column is $\|\boldsymbol{g}_i - \boldsymbol{g}_j\|_2 / \max_{k,k'} \|\boldsymbol{g}_k - \boldsymbol{g}'_k\|_2$.

the existing task ID embeddings as the embedding of the new task. Specifically, after adapting the base model on CSR-MTL, MoORE only learns the task IDs of the nine tasks in CSR-MTL. During evaluation on OR-MTL, we use the average of the nine task ID embeddings as the task ID embedding of each new task. The results in Figure 1(c) and 2 validate the effectiveness of this approach.

In addition, this experiment also explains the result in Figure 1(c). In particular, the more tasks considered in the adaptation phase, the more likely some tasks correlate with those covered in the pre-training phase. As a result, increasing the number of tasks in the adaptation phase helps enhance the oblivion-resistance of MoORE.

4.4 Inference Efficiency

To compare inference efficiency, we adapt the base model using MoORE and other baselines, and evaluate their inference time on GPQA. For a fair comparison, all experiments are conducted on a single NVIDIA A100 GPU with the same batch size. The results in Figure 1(d) show the average inference time per batch for each method, which indicates that MoORE achieves superior inference efficiency compared to most baselines and is comparable to MixLoRA. Compared to the base model (i.e., the gray dashed line), MoORE moderately increases inference time.

5 Conclusion and Future Work

In this paper, we propose a simple but effective model MoE-ization strategy for multi-task adaptation, leading to a novel MoE architecture, called MoORE. Experiments on various datasets show the effectiveness and superiority of MoORE, demonstrating that imposing orthogonality on experts and maintaining the column space of the pre-trained weight matrix help improve the resistance of the adapted model to task conflict and oblivion.

Limitations and future work. As shown in Figure 1(d), the inference efficiency of MoORE is comparable to that of baselines. However, further reducing its complexity for large-scale applications is challenging. In particular, MoORE applies many more experts than classic MoE models, and the analytic experiments in Appendix C.4 show that it learns dense weights for the experts. In addition, besides imposing orthogonal adapters, we need to find a more effective solution to enhance model capacity in broader scenarios. In theory, the results in Figures 1(b) and 1(c) show that increasing the number of tasks leads to a higher risk of task conflict while helping mitigate task oblivion. How to understand this "trade-off"? How can the limitation in multi-task adaptation be quantified regarding the number of tasks? These are still open problems. Based on the above analysis, we plan to explore efficient implementations of MoORE and study its theoretical properties in the future.

https://doi.org/10.52202/085713-1795

Acknowledgments and Disclosure of Funding

This work was supported by National Natural Science Foundation (92270110), Beijing Natural Science Foundation (L233008), the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China. We also acknowledge the support provided by the fund for building world-class universities (disciplines) of Renmin University of China and by the funds from Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, and from Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China.

References

[1] Hervé Abdi. Singular value decomposition (svd) and generalized singular value decomposition. *Encyclopedia of measurement and statistics*, 907(912):44, 2007.

[2] Ahmed Agiza, Marina Neseem, and Sherief Reda. Mtlora: Low-rank adaptation approach for efficient multi-task learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16196–16205, 2024.

[3] Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. Efficient large scale language modeling with mixtures of experts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11699–11732, 2022.

[4] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[5] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[6] Åke Björck. Numerics of gram-schmidt orthogonalization. *Linear Algebra and Its Applications*, 197:297–316, 1994.

[7] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[8] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michaelan Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

[9] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

[10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[11] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[12] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

[13] Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. Della-merging: Reducing interference in model merging through magnitude-based sampling. *arXiv preprint arXiv:2406.11617*, 2024.

 https://doi.org/10.52202/085713-1795

[14] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, 2024.

[15] Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, et al. Parameter competition balancing for model merging. *arXiv preprint arXiv:2410.02396*, 2024.

[16] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning*, pages 5547–5569. PMLR, 2022.

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

[18] Jinyuan Feng, Zhiqiang Pu, Tianyi Hu, Dongmin Li, Xiaolin Ai, and Huimu Wang. Omoe: Diversifying mixture of low-rank adaptation by orthogonal finetuning. *arXiv preprint arXiv:2501.10062*, 2025.

[19] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

[20] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[21] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7323–7334, 2023.

[22] Ethan He, Abhinav Khattar, Ryan Prenger, Vijay Korthikanti, Zijie Yan, Tong Liu, Shiqing Fan, Ashwath Aithal, Mohammad Shoeybi, and Bryan Catanzaro. Upcycling large language models into mixture of experts. *arXiv preprint arXiv:2410.07524*, 2024.

[23] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[24] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[25] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

[26] Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. Emr-merging: Tuning-free high-performance model merging. *Advances in Neural Information Processing Systems*, 37:122741–122769, 2024.

[27] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[28] Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

[30] Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, et al. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts. *arXiv preprint arXiv:2404.15159*, 2024.

[31] Mengqi Liao, Wei Chen, Junfeng Shen, Shengnan Guo, and Huaiyu Wan. Hmora: Making llms more effective with hierarchical mixture of lora experts. In *The Thirteenth International Conference on Learning Representations*, 2025.

[32] Zhisheng Lin, Han Fu, Chenghao Liu, Zhuo Li, and Jianling Sun. Pemt: Multi-task correlation guided mixture-of-experts enables parameter-efficient transfer learning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6869–6883, 2024.

[33] Vijay Chandra Lingam, Atula Neerkaje, Aditya Vavre, Aneesh Shetty, Gautham Krishna Gudur, Joydeep Ghosh, Eunsol Choi, Alex Dimakis, Aleksandar Bojchevski, and Sujay Sanghavi. Svft: Parameter-efficient fine-tuning with singular vectors. *Advances in Neural Information Processing Systems*, 37:41425–41446, 2024.

[34] Bingchang Liu, Chaoyu Chen, Zi Gong, Cong Liao, Huan Wang, Zhichao Lei, Ming Liang, Dajun Chen, Min Shen, Hailian Zhou, et al. Mftcoder: Boosting code llms with multitask fine-tuning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5430–5441, 2024.

[35] Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1104–1114, 2024.

[36] Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, et al. Parameter-efficient orthogonal finetuning via butterfly factorization. In *International Conference on Learning Representations*, 2024.

[37] Yen-Cheng Liu, Chih-Yao Ma, Junjiao Tian, Zijian He, and Zsolt Kira. Polyhistor: parameter-efficient multi-task adaptation for dense vision tasks. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 36889–36901, 2022.

[38] Zefang Liu and Jiahua Luo. Adamole: Fine-tuning large language models with adaptive mixture of low-rank adaptation experts. In *First Conference on Language Modeling*, 2024.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[40] Yuxiang Lu, Shalayiding Sirejiding, Yue Ding, Chunlin Wang, and Hongtao Lu. Prompt guided transformer for multi-task dense prediction. *IEEE Transactions on Multimedia*, 26:6375–6385, 2024.

[41] 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.

[42] Fan Lyu, Shuai Wang, Wei Feng, Zihan Ye, Fuyuan Hu, and Song Wang. Multi-domain multi-task rehearsal for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8819–8827, 2021.

[43] Xinyu Ma, Xu Chu, Zhibang Yang, Yang Lin, Xin Gao, and Junfeng Zhao. Parameter efficient quasi-orthogonal fine-tuning via givens rotation. In *International Conference on Machine Learning*, pages 33686–33729. PMLR, 2024.

 https://doi.org/10.52202/085713-1795

[44] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576, 2021.

[45] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

[46] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576, 2022.

[47] Taishi Nakamura, Takuya Akiba, Kazuki Fujii, Yusuke Oda, Rio Yokota, and Jun Suzuki. Drop-upcycling: Training sparse mixture of experts with partial re-initialization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[48] Zelin Peng, Zhengqin Xu, Zhilin Zeng, Xiaokang Yang, and Wei Shen. Sam-parser: Fine-tuning sam efficiently by parameter space reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4515–4523, 2024.

[49] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362, 2023.

[50] Dripta S Raychaudhuri, Yumin Suh, Samuel Schulter, Xiang Yu, Masoud Faraki, Amit K Roy-Chowdhury, and Manmohan Chandraker. Controllable dynamic multi-task architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2022.

[51] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[52] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

[53] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[54] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

[55] 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 *International Conference on Learning Representations*, 2017.

[56] Yanpeng Sun, Qiang Chen, Xiangyu He, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Jian Cheng, Zechao Li, and Jingdong Wang. Singular value fine-tuning: Few-shot segmentation requires few-parameters fine-tuning. *Advances in neural information processing systems*, 35:37484–37496, 2022.

[57] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5227–5237, 2022.

[58] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[59] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[60] Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng-Zhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37:9565–9584, 2024.

[61] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021.

[62] Nelson Vithayathil Varghese and Qusay H Mahmoud. A survey of multi-task deep reinforcement learning. *Electronics*, 9(9):1363, 2020.

[63] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations*, 2019.

[64] Fan Wang, Juyong Jiang, Chansung Park, Sunghun Kim, and Jing Tang. Kasa: Knowledge-aware singular-value adaptation of large language models. *arXiv preprint arXiv:2412.06071*, 2024.

[65] Mengmeng Wang, Jiazheng Xing, Boyuan Jiang, Jun Chen, Jianbiao Mei, Xingxing Zuo, Guang Dai, Jingdong Wang, and Yong Liu. A multimodal, multi-task adapting framework for video action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5517–5525, 2024.

[66] Renzhi Wang and Piji Li. Lemoe: Advanced mixture of experts adaptor for lifelong model editing of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2551–2575, 2024.

[67] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*, 2023.

[68] Yiming Wang, Yu Lin, Xiaodong Zeng, and Guannan Zhang. Multilora: Democratizing lora for better multi-task learning. *arXiv preprint arXiv:2311.11501*, 2023.

[69] Zhiwu Wang, Yichen Wu, Renzhen Wang, Haokun Lin, Quanziang Wang, Qian Zhao, and Deyu Meng. Singular value fine-tuning for few-shot class-incremental learning. *arXiv preprint arXiv:2503.10214*, 2025.

[70] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11293–11302, 2019.

[71] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*, 2023.

[72] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

[73] Yaming Yang, Dilxat Muhtar, Yelong Shen, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Weiwei Deng, Feng Sun, Qi Zhang, et al. Mtl-lora: Low-rank adaptation for multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22010–22018, 2025.

 https://doi.org/10.52202/085713-1795

[74] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

[75] Shen Yuan, Haotian Liu, and Hongteng Xu. Bridging the gap between low-rank and orthogonal adaptation via householder reflection adaptation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[76] Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

[77] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[78] Qizhen Irene Zhang, Nikolas Gritsch, Dwaraknath Gnaneshwar Talupuru, Simon Guo, David Cairuz, Bharat Venkitesh, Jakob Foerster, Phil Blunsom, Sebastian Ruder, Ahmet Üstün, et al. Bam! just like that: Simple and efficient parameter upcycling for mixture of experts. *Advances in Neural Information Processing Systems*, 37:56304–56321, 2024.

[79] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2022.

[80] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021.

[81] Lulu Zhao, Weihao Zeng, Shi Xiaofeng, and Hua Zhou. Mosld: An extremely parameter-efficient mixture-of-shared loras for multi-task learning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1647–1659, 2025.

[82] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

[83] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our claims made in the abstract and introduction accurately reflect the contributions and scope of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the proposed method in Limitations, such as its inability to fully leverage the potential of orthogonal fine-tuning.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/085713-1795

Answer: [Yes]

Justification: We provide the complete proof process in both the main text and Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The detailed hyperparameter configurations as well as experimental setups are thoroughly documented in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have released the code on an anonymous GitHub repository. The datasets used in the experiments are all publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The detailed hyperparameter configurations as well as experimental setups are thoroughly documented in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We did not have sufficient computational resources to run multiple rounds of experiments to obtain error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/085713-1795

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided detailed information about the computational resources used in the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Every aspect of the paper complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed both potential positive societal impacts and negative societal impacts of the work performed in Section D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the creators or original owners of assets including code, data, and models used in the paper are properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

https://doi.org/10.52202/085713-1795

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the comments in the released code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The proposed method is specifically designed for multi-task adaptation of LLMs, making LLMs an essential component of this paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-1795

A Hyperparameters and Implementation Details

The detailed hyperparameter setups are presented in Table 7. Both training and testing are conducted on one NVIDIA A100 GPU.

Table 7: Hyperparameter configurations of MoORA on the CSR-MTL and the NLU-MTL.

Hyperparameters	MoORA
Cutoff Length	512
Batch Size	8 / 64
Epochs	2 / 5
Learning Rate	3E-04
LR scheduler	Warmup-Stable-Decay
Warmup Ratio	5%
Decay Ratio	5%
Optimizer	AdamW
Dropout Rate	0.0
Target Modules	Q, K, V, O, Up, Down, Gate
D_t	128
D_s	64
L	8

B Tasks and Datasets

Detailed information about the CSR-MTL, the NLU-MTL and the OR-MTL is presented in Tables 8, Table 9 and Table 10, respectively. These tables include the sizes of the training and test sets, as well as the task types.

Table 8: The basic information of CSR-MTL.

Task Name	#Train	#Test	Task Type
ARC-Challenge	1,119	1,172	Question Answering
ARC-Easy	2,250	2,380	Question Answering
BoolQ	9,427	3,270	Text Classification
OpenBookQA	4,957	500	Question Answering
PIQA	16,100	1,840	Question Answering
SocialIQA	33,410	1,954	Question Answering
HellaSwag	39,905	10,042	Sentence Completion
WinoGrande	9,248	1,267	Fill in the Blank
CommonsenseQA	9,741	1,140	Question Answering

Table 9: The basic information of NLU-MTL.

Task Name	#Train	#Validation	Task Type
CoLA	8,551	1,043	Text Classification
MNLI	392,702	9,815	-
MRPC	3,668	408	-
QNLI	104,743	5,463	-
QQP	363,846	40,430	-
RTE	67,350	873	-
SST-2	2,491	277	-

Table 10: The basic information of OR-MTL.

Task Name	#Test	Task Type
MMLU	14,042	Question Answering
IFEval	541	Text Generation
BBH	6,511	Question Answering
GPQA	448	Question Answering
HumanEval	164	Text Generation
MBPP	500	Text Generation
GSM-8K	1,319	Text Generation

C More Experimental Results

C.1 Performance on other Model Architectures in Conflict-Resistance

To evaluate the cross-model performance of MoORE, we take Qwen-2.5 7B as the backbone model and apply various MoE-based multi-task adaptation methods to it. Experimental results on CSR-MTL are shown in Table 11. We can find that MoORE outperforms baselines consistently under different hyperparameter settings, and the performance of MoORE is robust to the change of L. These results verify the robustness of our method and its compatibility with various model architectures.

Table 11: Results (%) of Qwen-2.5 7B adapted by various methods on CSR-MTL. The best results on each dataset are shown in **bold**, and the second best results are shown in underline.

Method	#Params	ARC-C	ARC-E	BoolQ	OBQA	PIQA	SIQA	HellaS	WinoG	CSQA	Overall
LoRA [25]	2.12%	87.29	91.88	70.37	90.60	87.92	79.02	92.73	74.98	84.36	84.35
MixLoRA [30]	3.32%	86.86	91.75	71.38	91.40	89.28	80.45	95.06	78.45	85.09	85.53
MoSLD [81]	1.43%	86.69	92.13	71.96	90.40	89.66	80.40	95.17	78.30	**85.50**	85.58
HydraLoRA [60]	2.83%	88.23	91.54	72.57	91.20	85.80	80.09	95.95	**82.24**	85.26	85.87
MTL-LoRA [73]	2.80%	87.80	92.09	72.72	92.20	86.07	79.79	95.92	80.66	84.36	85.73
LoRAMoE [14]	2.21%	87.20	91.84	**73.61**	90.40	**90.37**	**81.27**	**95.99**	80.27	84.60	86.17
MoORE $_{L=0}$	2.29%	88.06	92.17	73.49	92.20	**90.37**	79.58	95.40	81.53	84.03	86.31
MoORE $_{L=2}$	2.32%	88.23	92.34	73.09	91.60	90.32	80.60	95.50	81.45	84.19	86.37
MoORE $_{L=4}$	2.35%	**88.40**	92.21	73.18	91.40	90.21	**81.27**	95.25	80.58	84.93	86.38
MoORE $_{L=16}$	2.53%	88.23	**92.59**	73.30	**92.40**	90.26	81.12	95.50	81.37	84.93	**86.63**

C.2 Numerical Performance in Oblivion-Resistance

Following LLaMA-3.1 8B, we select MMLU [24, 23], IFEval [83], BIG-Bench Hard (BBH) [58], GPQA [51], HumanEval [7], MBPP [4], and GSM-8K [10] as evaluation tasks and adopt the same few-shot settings. More detailed results of the oblivion-resistance experiments are presented in Table 12, corresponding to the results shown in Figures 1(c) and 2.

C.3 Ablation Study

To analyze the impact of the task-level routing, sample-level routing, and orthogonal adapter quantitatively, we conduct ablation study on the CSR-MTL dataset. In Table 13, we record the overall performance achieved when only one of these three modules is applied. In particular, without the orthogonal adapter, our method degrades to the MoE with fixed experts. When merely applying the orthogonal adapter without any routing, our method is simplified as the single-task adaptation method HRA [75]. The results show that when only one of these three modules is applied, even with a wider network and more trainable parameters, the model's performance declines significantly across the entire dataset. These results demonstrate that all these three modules contribute to the overall effectiveness of MoORE.

https://doi.org/10.52202/085713-1795

Table 12: The loss of performance in the original tasks. Each method is represented by two rows of data: the first row indicates the performance on the current task, while the second row shows the difference between the post-fine-tuning score and the pre-fine-tuning score.

Category	General			Reasoning	Code		Math	
Task	MMLU	IFEval	BBH	GPQA	HumanEval	MBPP	GSM-8K	Overall
#Shots	5		0	0	0	3	8	
Llama-3.1-8B-Instruct	68.15	84.53	46.09	34.15	56.71	59.00	85.52	62.02
LoRA	32.40 -35.75	29.14 -55.39	13.65 -32.44	27.23 -6.92	0.00 -56.71	0.00 -59.00	2.12 -83.40	14.93 -47.09
LoRAMoE	63.74 -4.41	78.06 -6.47	34.16 -11.93	31.47 -2.68	60.98 4.27	57.40 -1.60	78.54 -6.98	57.76 -4.26
MoSLD	61.77 -6.38	75.06 -9.47	34.74 -11.35	28.57 -5.58	56.10 -0.61	53.60 -5.40	70.05 -15.47	54.27 -7.75
MTL-LoRA	61.55 -6.60	72.30 -12.23	31.59 -14.50	31.92 -2.23	57.32 0.61	51.60 -7.40	66.57 -18.95	53.26 -8.76
HydraLoRA	62.34 -5.81	69.78 -14.75	30.27 -15.82	30.36 -3.79	54.27 -2.44	54.80 -4.20	66.41 -19.11	52.60 -9.42
MixLoRA	64.47 -3.68	77.82 -6.71	38.21 -7.88	30.13 -4.02	54.27 -2.44	56.20 -2.80	74.83 -10.69	56.56 -5.46
MoORE	65.32 -2.83	82.49 -2.04	40.21 -5.88	31.92 -2.23	66.46 9.75	58.20 -0.80	80.36 -5.16	60.71 -1.31

Table 13: Ablation studies of MoORE on CSR-MTL. ✖ indicates the absence of that component.

Method	D_t	D_s	L	#Params	Overall
Merely Apply Task-level Routing	16	✖	✖	0.14%	79.55
	32	✖	✖	0.29%	81.80
	64	✖	✖	0.58%	83.06
	128	✖	✖	1.16%	83.53
Merely Apply Sample-level Routing	✖	16	✖	0.39%	82.30
	✖	32	✖	0.78%	82.91
	✖	64	✖	1.57%	83.57
	✖	128	✖	3.13%	83.80
Merely Apply Orthogonal Adapter	✖	✖	2	0.03%	80.76
	✖	✖	4	0.06%	82.15
	✖	✖	8	0.12%	82.98
	✖	✖	16	0.25%	83.90
MoORE	128	64	8	2.84%	85.11

C.4 Analysis of Routing Weights

To investigate the router's preferences for the experts in MoORE, we analyze the distribution of routing weights across different tasks in the CSR-MTL dataset. Take the Q module from the first attention layer of the adapted model as an example. For each task, we compute and show the mean (red point) and variance (blue stem) of each expert's weight over all samples of the task in Figures 4 and 6. In each figure, the mean weight of an expert is shown as a red point, and the blue stem associated with the point shows the standard deviation. These visualization results provide us with some insights in the mechanism behind MoORE:

- **Preference on the originally important experts.** Compared to other components, the routing weights corresponding to the principal components of $\boldsymbol{W}$ (i.e., the experts corresponding to large singular values) are adjusted more significantly. The routing weights corresponding to the significant experts are distributed relatively symmetrically around the zero scale, whereas the routing weights for the minor components are more concentrated above the zero scale. This

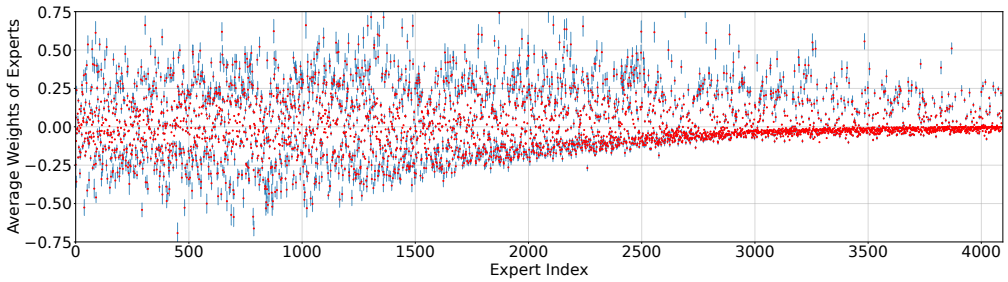


Figure 4: The mean and variance of routing weights obtained in HellaS.

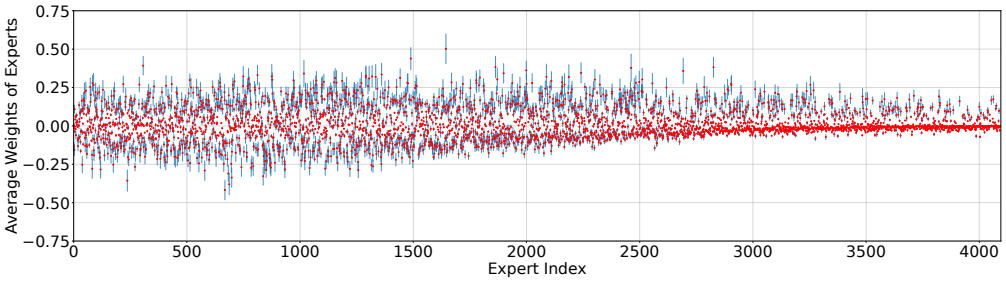
indicates that the model tends to focus on adjusting the routing weights of the experts that are important in the pre-training phase. In other words, these originally important experts often maintained their significance when adapting for new task.

- Based on the mean and variance of the routing weights, the nine tasks in CSR-MTL can be grouped into three categories:
  - **Large Mean, Large Variance:** Tasks such as Hellaswag fall into this group. MoORE performs well on the Hellaswag. A large mean indicates that the model significantly enhances or discards well-learned pretraining knowledge, while a large variance means that the model can capture substantial differences between samples and assign different experts to them with high flexibility.
  - **Small Mean, Large Variance:** Tasks such as OBQA, PIQA, SIQA, and CSQA belong to this group. This result indicates that for each of these tasks, its samples have significant diversity, and MoORE needs to activate different experts to handle different samples.
  - **Small Mean and Variance:** Tasks such as ARC-C, ARC-E, BoolQ, and Winogrande (WinoG) are in this group. Except for ARC-E, MoORE performs poorly on these tasks. A small mean indicates minimal adjustment to pretraining knowledge, and a small variance suggests minor differences between samples. For the relatively simple ARC-E task, the model may have already mastered the ability to solve it, requiring little adjustment. However, for the more challenging tasks in this group, the model may struggle to learn the ability to solve them within the column space spanned by the singular vectors.

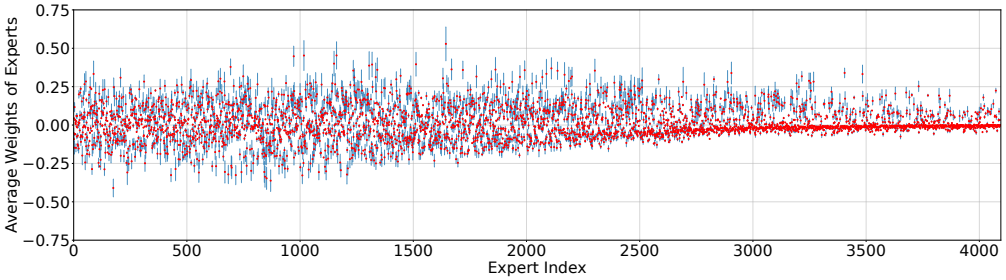
D Broader Impacts

MoORE can effectively enhance the performance of LLMs in multi-task adaptation and facilitate the application of LLMs in real-world scenarios. Similar to other multi-task adaptation methods, such as MixLoRA and HydraLoRA, MoORE may result in some potential negative social impacts, such as the risk of abuse for adapting LLMs in illegal activities. However, these issues are not unique to MoORE — other adaptation methods are also susceptible to similar risks. Addressing these issues thoroughly will require advancements in technology, the establishment of relevant legal frameworks, and shifts in societal perspectives.

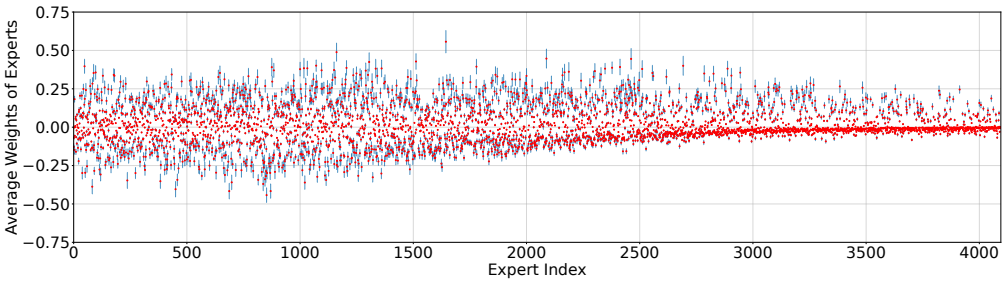
https://doi.org/10.52202/085713-1795



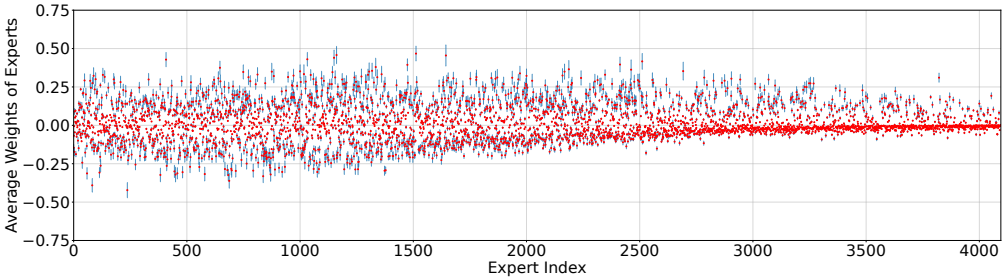
(a) OBQA



(b) PIQA



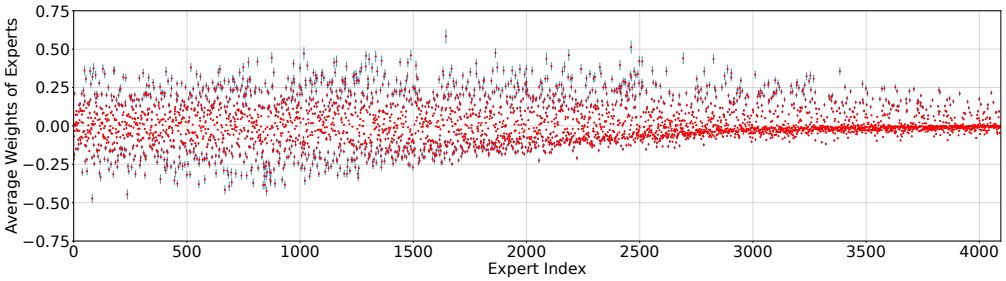
(c) SIQA



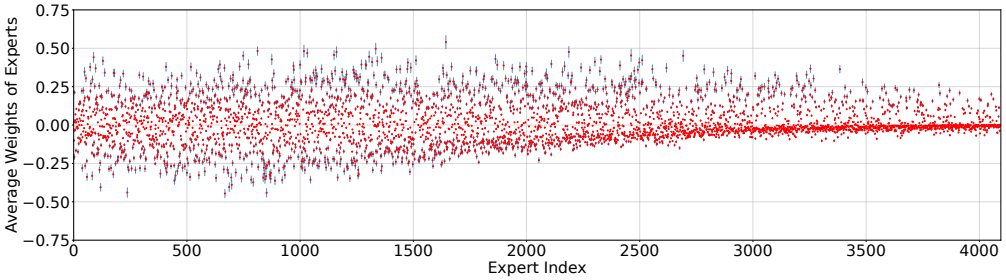
(d) CSQA

Figure 5: The mean and variance of routing weights obtained in four QA tasks.

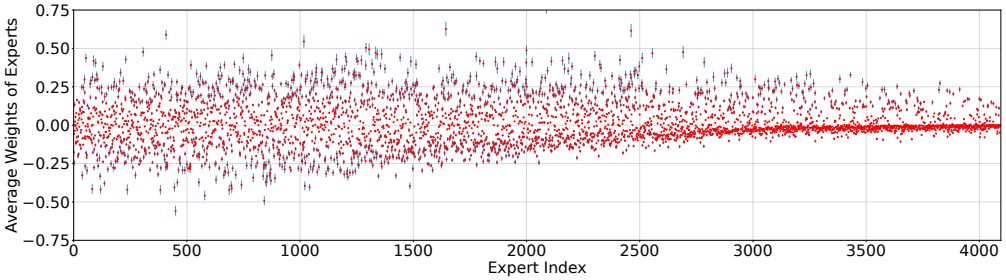
https://doi.org/10.52202/085713-1795



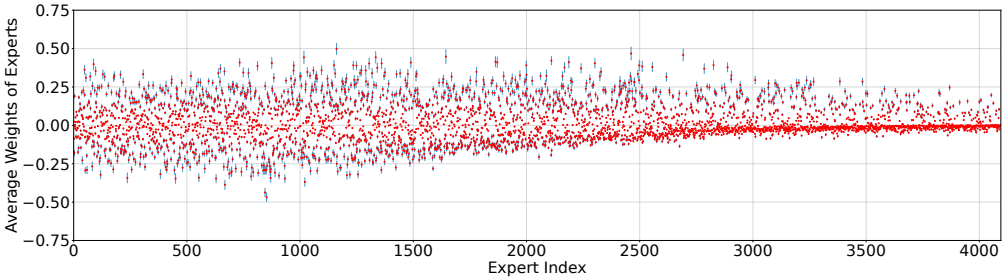
(a) ARC-C



(b) ARC-E



(c) BoolQ



(d) WinoG

Figure 6: The mean and variance of routing weights obtained in the remaining four tasks of CSR-MTL.

https://doi.org/10.52202/085713-1795