

MoANT: Mixture-of-Rank-One-Experts with Semantic-aware Intuition for Multi-task Large Language Model Finetuning

Rongyu Zhang*

HKPolyU

Nanjing University

rongyu.zhang@connect.polyu.hk

Yijiang Liu*

Nanjing University of

Information Science and Technology

liuyijiang@nuist.edu.cn

Huanrui Yang*

University of Arizona

huanruiyang@arizona.edu

Shenli Zheng

Nanjing University

shenlizheng@smail.nju.edu.cn

Li Du

Nanjing University

duli@nju.edu.cn

Dan Wang

HKUST

wangdan@hkust.hk

Yuan Du[✉]

Nanjing University

yuandu@nju.edu.cn

Shanghang Zhang^{✉†}

Peking University

shanghang@pku.edu.cn

Abstract

Large language models (LLMs) encounter significant challenges when adapting to diverse multitask fine-tuning settings. Mixture-of-experts (MoE) offers a promising solution with a dynamic architecture that enables effective task decoupling. However, scaling up the number of MoE experts incurs substantial parameter and computational overhead and yields limited performance gains due to naive routing mechanisms. In this paper, we design a novel framework, Mixture-of-rAnk-oNe-expErTs (MoANT), which leverages the combination of ultra-low-rank experts to construct LoRA weights on pretrained LLMs. The rank-1 experts enable a mix-and-match mechanism that quadratically expands the vector subspace of experts with linear parameter overhead, achieving approximate error reduction with optimal efficiency. In addition, MoANT offers implicit guidance to the router by leveraging the inherent semantic clustering of training embeddings as prior knowledge, enabling optimized feature allocation across experts for smoother convergence. Extensive theoretical and empirical results demonstrate that MoANT achieves superior efficiency and generalizability across a diverse range of tasks. Compared with traditional LoRA, MoANT achieves up to 1.78% mean accuracy improvement with around 30%-40% fewer trainable parameters across 14 public datasets.

1 Introduction

Large language models (LLMs) (Touvron et al., 2023; Jiang et al., 2023; Wen et al., 2026) exhibit profound capabilities across various domains, excelling in downstream tasks such as content generation, enhancing interactive entertainment, and inspiring artistic endeavors. Despite the transformative impact of LLMs on natural language processing (NLP), as evidenced by their mastery of unsupervised pretraining and subsequent supervised fine-tuning (Hu et al., 2021), the intricate settings of complex multitask learning (MTL) environments pose a unique challenge. The diversity of tasks in these environments tests the adaptability of LLMs (Zhang et al., 2023; 2026c), pushing the boundaries of their application and capability.

*Equal contributions; [✉] Corresponding authors; [†] State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University.

Mixture-of-Experts (MoE) (Masoudnia & Ebrahimpour, 2014; Jacobs et al., 1991; Zhang et al., 2026b) has recently gained attention as a practical approach to multi-task fine-tuning, leveraging its dynamic architecture to efficiently adapt pretrained LLMs to diverse downstream tasks. Like humans, who demonstrate superior multitasking abilities by selectively activating specific brain regions in response to explicit cues, MoE dynamically engages and combines different experts for each task. However, applying MoE to LLM multi-task learning raises several unresolved challenges. ① **Significant parameter and computational overheads:** Unlike single-task MoE, where only a few experts suffice, multitask MoE requires scaling the number of experts with the number of tasks, leading to significant costs in training and storage. While prior works (Liao et al., 2025; Zhu et al., 2023) have explored parameter-efficient finetuning (PEFT) experts using Low-rank adapter (LoRA) (Hu et al., 2021; Huang et al., 2023), the parameter overhead still scales linearly with the number of experts. ② **Unfair router task allocation:** Traditional MoE routers, often based on a linear layer with softmax, struggle with fine-grained task differentiation, resulting in suboptimal expert allocation. Recent approaches (Chen et al., 2026; Zhang et al., 2024b) have introduced uncertainty-aware mechanisms (Lakshminarayanan et al., 2017; He et al., 2026) and explicit feature guidance to enhance routing, but their effectiveness in MTL remains unproven.

To address these challenges, we propose Mixture-of-rAnk-oNe-experts (MoANT), which leverages ① ultra-lightweight rank-1 experts with a Mix-and-Match (MaM) mechanism. Specifically, MoANT allows multiple rank-1 vectors $a_i \in \mathbb{R}^{m \times 1}$ and $b_j \in \mathbb{R}^{n \times 1}$ to pair freely into rank-1 LoRA experts $\Delta W_{ij} = a_i b_j^T$. This MaM mechanism enables MoANT to construct more LoRA experts to cover a higher-dimensional subspace of the weight-matrix space with sublinear growth in the number of additional parameters. MaM also facilitates more effective interaction and cross-fusion of shared feature representations across tasks, ultimately boosting the model’s generalization capability. Mathematically, MoANT has been shown to increase representational capacity relative to other LoRA-based methods, thereby achieving higher accuracy across tasks.

In addition, we propose ② a semantically aware router inspired by the brain’s ability to perform intuitive cognition, enabling MoANT to allocate experts more effectively to complex multitask scenarios. Naively, the router can use the human-annotated task categorization as a proxy for the condition *explicit intuition*. However, this empirically leads to 2.50% and 3.77% accuracy drops for LLaMA-2 (Touvron et al., 2023) and Gemma (Team et al., 2024) on OpenCompass¹, respectively. This highlights that the human-labeled training tasks are coarse-grained and each requires a mixture of diverse skills, whose ambiguity contradicts the expert’s need to identify and learn task-specific capabilities. To address this, we redefine expert tasks by clustering training embeddings. We assign inputs to experts based on the similarity between input embeddings and the corresponding task centroids. Specifically, MoANT leverages the correlation between the input and these centroids as an *implicit intuition*, which conditions the router’s decision-making and leads to smooth convergence.

Both theoretical derivations and extensive experiments demonstrate the superiority of our proposed MoANT against state-of-the-art baselines. As shown in Fig. 1, MoANT enables a sizable gain of both performance and efficiency over the Pareto frontier of trade-off for existing LoRA-MoE-based PEFT and MoE methods, delivering an increase in overall accuracy by up to 1.78% with around 30%-40% less parameter overhead. The contributions of MoANT include:

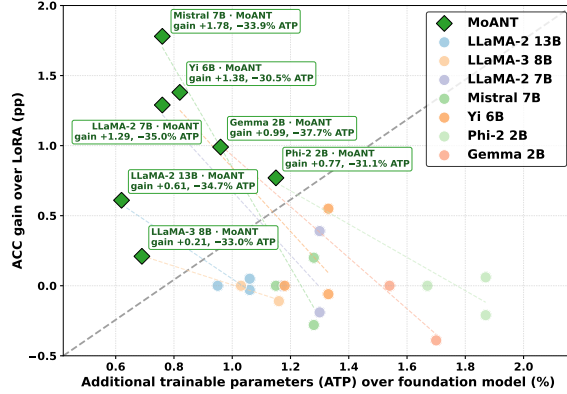


Figure 1: Accuracy *v.s.* efficiency for different methods. Circles of different colors denote different PEFT baselines.

¹<https://github.com/open-compass/opencompass>

- We devise a novel LoRA MoE paradigm that leverages ultra-lightweight rank-1 experts via a mix-and-match mechanism for efficient multi-task fine-tuning.
- Drawing on the concept of intuition akin to that of a human brain, we provide conditional guidance for the router by leveraging the resemblance between the input task instance and predefined embedding clusters to guide all the experts to converge smoothly.
- We provide a theoretical analysis showing that MoANT achieves expert-linear-combination subspace expansion, enabling approximate error reduction and guiding experts to converge smoothly, ultimately enhancing overall model performance.

2 Related Work

Multi-task learning. MTL aims to enhance a model’s generalization across multiple tasks by sharing insights. The shared-bottom model (Caruana, 1997) uses hard-shared parameters but can suffer from negative transfer. To address this, studies (Kirchdorfer et al., 2026; Chen et al., 2018) propose adaptive loss weighting to balance multi-task losses. Models like Cross-stitch (Misra et al., 2016; Beardsley et al., 2026) dynamically blend task-specific representations while using static task weights. Our method utilizes an MoE to achieve precise task allocation by assigning input to a combination of specialized experts.

Mixture of Experts. MoE was initially introduced by (Jordan & Jacobs, 1994) to process different samples with independent modules. For example, (Shazeer et al., 2017) employs MoE in large-scale LSTM-based models. Switch Transformer (Fedus et al., 2021) was the first to introduce MoE in Transformers and largely scale up model size using top-1/2 routing strategies. (Dai et al., 2022) stabilize the training process with fixed routing strategies. (Zhu et al., 2023) proposes parameter-efficient adapters as experts to decrease MoE size, while (Zhang et al., 2026b) leverages MoE to achieve layer-skipping in the robot manipulation model. Our method enhances routing using multi-task data insights and employs a rank-1 expert to improve efficiency.

Parameter-efficient finetuning. PEFT (Ding et al., 2023) adjusts minimal parameters to reduce storage demands. PEFT methods include specification-based (Zaken et al., 2021), reparameterization-based (Ding et al., 2021), and addition-based approaches. The most prevalent addition-based PEFTs, such as Adapter-Tuning (Zhang et al., 2023; 2026a), Prefix Tuning (Li & Liang, 2021; Ding et al., 2026), and Prompt Tuning (Lester et al., 2021; Yang et al., 2026), add new modules to the base model. Recent studies have combined low-rank adapters (LoRA) (Hu et al., 2021; Che et al., 2026) with MoE to enhance flexibility in PEFT. However, existing LoRA-MoE methods (Zadouri et al., 2023; Zhu et al., 2023; Liao et al., 2025) face challenges in effectively balancing the parameter efficiency and the scalability to more experts. To address this, MoANT incorporates rank-1 mix-and-match experts to scale up LoRA-MoE with minimal overhead.

3 Methodology

3.1 Preliminary

Problem formulation. In multitask learning with K distinct tasks, each task k is defined over an input space $\mathcal{X}_k \subset \mathbb{R}^m$, an output space $\mathcal{Y}_k$, and a distinct objective such as classification, generation, or retrieval. Given a pretrained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{m \times n}$ representing the parameters of an LLM, the goal of the multitask fine-tuning is to adapt $\mathbf{W}_0$ for each task efficiently while avoiding catastrophic forgetting. For an input $\mathbf{x} \in \mathcal{X}_k$ from task k , the model’s output $\mathbf{z} \in \mathcal{Y}_k$ is computed as $\mathbf{z} = f(\mathbf{W}_0, \mathbf{x})$, where $f(\cdot)$ denotes the forward pass to generalize across heterogeneous domains.

LoRA-MoE. LoRA-MoE (Zadouri et al., 2023) introduces N experts added onto $\mathbf{W}_0$, and a router dynamically selects and combines these experts based on the input. As illustrated

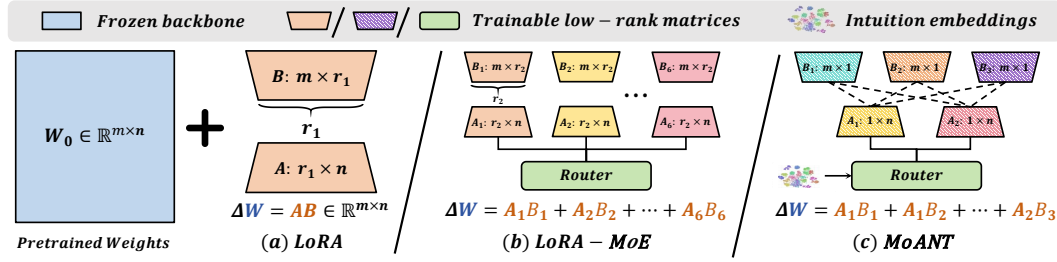


Figure 2: Illustration of (a) Vanilla LoRA, (b) LoRA-MoE, and (c) MoANT. $r_1, r_2 \ll \min\{m, n\}$. Six experts are demonstrated for both **LoRA-MoE** $\Rightarrow 6W_a$ & $6W_b = 6W_{\text{experts}}^{\text{Lora}}$ and **MoANT** $\Rightarrow 3W_a \times 2W_b = 6W_{\text{experts}}^{\text{MoANT}}$.

in Fig. 2 (b), the i -th expert is parameterized as: $f_i(\mathbf{x}) = \Delta \mathbf{W}_i \mathbf{x}$, where $\Delta \mathbf{W}_i$ is a low-rank matrix achieved via parameter-efficient finetuning methods like LoRA. The overall layer output is computed as:

$$\mathbf{z} = \mathbf{W}_0 \mathbf{x} + \underbrace{\sum_{i=1}^N \mathbf{G}_i(\mathbf{x}) \Delta \mathbf{W}_i \mathbf{x}}_{\text{Expert Contribution}}, \quad (1)$$

where $\mathbf{G}_i(\mathbf{x})$ represents the routing weight for the i -th expert, dynamically computed for each input $\mathbf{x}$. We integrate subspace projections and semantic-aware intuition scores to improve the routing mechanism, as detailed in Sec. 3.3.

3.2 Mix-and-Match for rank-1 experts

To tackle the scalability challenges of MoE, we introduce Rank-1 Experts, designed to maximize parameter efficiency while preserving the flexibility of expert composition. Specifically, we start with a decoupled left-right basis decomposition of the LoRA update, as shown in Fig. 2(c). Formally, for a pretrained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{m \times n}$, the i -th rank-1 LoRA expert is defined as

$$\Delta \mathbf{W}_i = \mathbf{a}_i \mathbf{b}_i^\top, \quad \text{where } \mathbf{a}_i \in \mathbb{R}^m, \mathbf{b}_i \in \mathbb{R}^n. \quad (2)$$

Here, $\{\mathbf{a}_i\}$ and $\{\mathbf{b}_i\}$ are trainable left and right basis vectors, spanning the column and row subspaces of the adaptation matrix, respectively. With N such one-to-one paired experts, the layer output is computed as:

$$\mathbf{z} = \mathbf{W}_0 \mathbf{x} + \mathbf{A} \text{diag}(\mathbf{g}(\mathbf{x})) \mathbf{B}^\top \mathbf{x}, \quad (3)$$

where $\mathbf{A} = [\mathbf{a}_1 | \dots | \mathbf{a}_N] \in \mathbb{R}^{m \times N}$, $\mathbf{B} = [\mathbf{b}_1 | \dots | \mathbf{b}_N] \in \mathbb{R}^{n \times N}$, and $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^N$ denotes the input-dependent routing weights.

The above rank-1 MoE is a compact design for LoRA-MoE. However, the number of available experts is still tied to the number of paired basis vectors: with N left and N right vectors, it constructs only N rank-1 experts through one-to-one pairing. To enhance flexibility and reduce the parameter overhead of scaling up the expert pool, we decouple this one-to-one correspondence between the left and right bases. Specifically, any $\mathbf{a}_i$ can be paired with any $\mathbf{b}_j$, forming a mix-and-match expert $f_{ij}(\mathbf{x}) = \mathbf{a}_i \mathbf{b}_j^\top \mathbf{x}$. This mechanism expands the expert pool from linear to quadratic scaling through cross-combinations. The overall output is:

$$\mathbf{z} = \mathbf{W}_0 \mathbf{x} + \sum_{i=1}^I \sum_{j=1}^J \tilde{\mathbf{G}}_{ij}(\mathbf{x}) (\mathbf{a}_i \mathbf{b}_j^\top) \mathbf{x} = \mathbf{W}_0 \mathbf{x} + \mathbf{A} \tilde{\mathbf{G}}(\mathbf{x}) \mathbf{B}^\top \mathbf{x}, \quad (4)$$

where $\mathbf{A} = [\mathbf{a}_1 | \dots | \mathbf{a}_I] \in \mathbb{R}^{m \times I}$ and $\mathbf{B} = [\mathbf{b}_1 | \dots | \mathbf{b}_J] \in \mathbb{R}^{n \times J}$ contain I left and J right basis vectors, respectively. The routing matrix $\tilde{\mathbf{G}}(\mathbf{x}) \in \mathbb{R}^{I \times J}$ is no longer required to be square or

diagonal, and it dynamically activates the cross-combined rank-1 experts. Furthermore, the mix-and-match mechanism can be expressed through standard column-wise vectorization:

$$\text{vec}(\Delta \mathbf{W}(\mathbf{x})) = \sum_{i=1}^I \sum_{j=1}^J \tilde{G}_{ij}(\mathbf{x}) (\mathbf{b}_j \otimes \mathbf{a}_i) = (\mathbf{B} \otimes \mathbf{A}) \text{vec}(\tilde{\mathbf{G}}(\mathbf{x})). \quad (5)$$

Here, $\otimes$ denotes the Kronecker product, $\text{vec}(\cdot)$ denotes column-wise vectorization, and $N = I \times J$ is the total number of virtual rank-1 experts. This design achieves quadratic scaling of the virtual expert pool with only linear growth in additional basis parameters, fundamentally differing from the capacity-parameter trade-off of standard LoRA. The pseudo-code is shown in Algorithm 1.

3.3 Semantic-aware router with intuition clustering

Naive MoE often exhibits weak alignment between routing decisions and the underlying task structure, yielding undertrained or redundant experts. Therefore, MoANT introduces a semantic prior to guide expert allocation over the MTL space. However, human-defined task labels are often coarse and ambiguous: token embeddings from the 20 training datasets show that samples within the same labeled category can scatter widely, while samples from different categories may cluster closely. We therefore define expert specialization using semantic clusters of training embeddings: each virtual expert is associated with one semantic cluster, and inputs are routed according to their similarity to the corresponding cluster centroids.

Formally, suppose MoANT has $N = I \times J$ virtual experts. We first collect embeddings from the multitask training tokens, $\hat{\mathbf{e}} = E(\hat{\mathbf{x}}) \in \mathbb{R}^d$, using a text embedding model. These embeddings are grouped into N semantic clusters, corresponding to the N virtual experts, using k -means clustering:

$$\mathcal{C}_i = \left\{ \hat{\mathbf{e}}_j : \|\hat{\mathbf{e}}_j - \boldsymbol{\mu}_i\|_2^2 \leq \|\hat{\mathbf{e}}_j - \boldsymbol{\mu}_r\|_2^2, \forall r \in [1, N] \right\}, \quad (6)$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^d$ is the centroid of the i -th cluster. During fine-tuning, for an input $\mathbf{x}$ with embedding $\mathbf{e} = E(\mathbf{x})$, we compute the semantic intuition score for each centroid as

$$s_i(\mathbf{x}) = \cos \angle(\mathbf{e}, \boldsymbol{\mu}_i) = \frac{\mathbf{e}^\top \boldsymbol{\mu}_i}{\|\mathbf{e}\|_2 \|\boldsymbol{\mu}_i\|_2}. \quad (7)$$

The scores are collected into a vector $\mathbf{s}(\mathbf{x}) = [s_i(\mathbf{x})]_{i=1}^N \in \mathbb{R}^N$ and reshaped into a semantic score matrix $\mathbf{S}(\mathbf{x}) \in \mathbb{R}^{I \times J}$. We then fuse this semantic prior with the learnable routing logits $\mathbf{R}(\mathbf{x}) \in \mathbb{R}^{I \times J}$ before normalization:

$$\tilde{\mathbf{G}}(\mathbf{x}) = \text{reshape}_{I \times J} (\text{softmax}(\text{vec}(\mathbf{R}(\mathbf{x})) + \lambda \text{vec}(\mathbf{S}(\mathbf{x})))), \quad (8)$$

where $\lambda \geq 0$ controls the strength of the semantic prior. This enhanced routing signal is used to construct the final expert combination, enabling a more semantically informed allocation of rank-1 experts.

4 Theoretical Analysis of MoANT

4.1 Rank-1 Experts with Subspace Expansion

Let $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_I] \in \mathbb{R}^{m \times I}$ and $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_J] \in \mathbb{R}^{n \times J}$ denote the left and right bases. Given an input-dependent routing matrix $\tilde{\mathbf{G}}(\mathbf{x}) \in \mathbb{R}^{I \times J}$, MoANT produces

$$\Delta \mathbf{W}(\mathbf{x}) = \mathbf{A} \tilde{\mathbf{G}}(\mathbf{x}) \mathbf{B}^\top = \sum_{i=1}^I \sum_{j=1}^J \tilde{G}_{ij}(\mathbf{x}) \mathbf{a}_i \mathbf{b}_j^\top. \quad (9)$$

Thus, MoANT routes over IJ virtual rank-1 experts while storing only $I + J$ basis vectors.

Algorithm 1: MoANT: Mixture of Rank-1 Experts with Semantic-Aware Routing

Input : Input $\mathbf{x}$, pretrained weight $\mathbf{W}_0 \in \mathbb{R}^{m \times n}$, basis sizes I, J , semantic centroids $\{\boldsymbol{\mu}_k\}_{k=1}^{IJ}$, embedding model E , prior strength λ .

Output: Layer output $\mathbf{z}$.

Initialize: Left bases $\{\mathbf{a}_i\}_{i=1}^I$, right bases $\{\mathbf{b}_j\}_{j=1}^J$, and router logits $\mathbf{R}(\cdot) \in \mathbb{R}^{I \times J}$.

Compute semantic scores $\mathbf{s}(\mathbf{x}) = [\cos \angle(E(\mathbf{x}), \boldsymbol{\mu}_k)]_{k=1}^{IJ}$. // Step 1: Semantic prior

Reshape $\mathbf{S}(\mathbf{x}) = \text{reshape}_{I \times J}(\mathbf{s}(\mathbf{x}))$. // Prior over expert slots

Compute routing weights

$$\tilde{\mathbf{G}}(\mathbf{x}) = \text{reshape}_{I \times J}(\text{softmax}(\text{vec}(\mathbf{R}(\mathbf{x})) + \lambda \text{vec}(\mathbf{S}(\mathbf{x}))))$$

// Semantic-aware routing

Compute update $\Delta \mathbf{W}(\mathbf{x}) = \sum_{i=1}^I \sum_{j=1}^J \tilde{G}_{ij}(\mathbf{x}) \mathbf{a}_i \mathbf{b}_j^\top$. // Step 2: Mix-and-Match mixture

Return: $\mathbf{z} = \mathbf{W}_0 \mathbf{x} + \Delta \mathbf{W}(\mathbf{x}) \mathbf{x}$.

// Adapted output

Lemma 4.1 (Tensor-product subspace). *The reachable update set*

$$\mathcal{S}_{\mathbf{A}, \mathbf{B}} = \{\mathbf{A} \mathbf{G} \mathbf{B}^\top : \mathbf{G} \in \mathbb{R}^{I \times J}\} \quad (10)$$

has dimension $\text{rank}(\mathbf{A})\text{rank}(\mathbf{B}) \leq IJ$, with equality when $\mathbf{A}$ and $\mathbf{B}$ are full column rank. Moreover,

$$\text{rank}(\Delta \mathbf{W}(\mathbf{x})) \leq \min\{\text{rank}(\mathbf{A}), \text{rank}(\tilde{\mathbf{G}}(\mathbf{x})), \text{rank}(\mathbf{B})\} \leq \min\{I, J\}. \quad (11)$$

Proof. By column-wise vectorization,

$$\text{vec}(\mathbf{A} \mathbf{G} \mathbf{B}^\top) = (\mathbf{B} \otimes \mathbf{A}) \text{vec}(\mathbf{G}). \quad (12)$$

Hence,

$$\dim(\mathcal{S}_{\mathbf{A}, \mathbf{B}}) = \text{rank}(\mathbf{B} \otimes \mathbf{A}) = \text{rank}(\mathbf{B})\text{rank}(\mathbf{A}). \quad (13)$$

The rank bound follows from the submultiplicativity of matrix rank. ■

Proposition 4.2 (Best approximation in the learned subspace). *Let $\Delta \mathbf{W}^* \in \mathbb{R}^{m \times n}$ be a target update. Let $\mathbf{P}_{\mathbf{A}}$ and $\mathbf{P}_{\mathbf{B}}$ be the orthogonal projectors onto $\text{col}(\mathbf{A})$ and $\text{col}(\mathbf{B})$, respectively. Then,*

$$\min_{\mathbf{G} \in \mathbb{R}^{I \times J}} \|\mathbf{A} \mathbf{G} \mathbf{B}^\top - \Delta \mathbf{W}^*\|_F^2 = \|\Delta \mathbf{W}^* - \mathbf{P}_{\mathbf{A}} \Delta \mathbf{W}^* \mathbf{P}_{\mathbf{B}}\|_F^2. \quad (14)$$

Proof. The space $\mathcal{S}_{\mathbf{A}, \mathbf{B}}$ consists of matrices with column space in $\text{col}(\mathbf{A})$ and row space in $\text{col}(\mathbf{B})$. Therefore, the Frobenius-orthogonal projection of $\Delta \mathbf{W}^*$ onto this space is $\mathbf{P}_{\mathbf{A}} \Delta \mathbf{W}^* \mathbf{P}_{\mathbf{B}}$. This projected matrix lies in $\mathcal{S}_{\mathbf{A}, \mathbf{B}}$, and the residual is orthogonal to every element in $\mathcal{S}_{\mathbf{A}, \mathbf{B}}$. The result follows from the projection theorem. ■

This proposition shows that the approximation error depends on the alignment between the target update and the learned left-right subspaces. Increasing I or J helps only when it improves this subspace coverage.

4.2 Semantic Intuition as a Routing Prior

MoANT further uses semantic intuition to guide routing within the expanded virtual expert space. The intuition signal serves as a soft prior rather than a hard assignment.

Proposition 4.3 (Centroid similarity as coordinate prior). *Let $\{\boldsymbol{\mu}_i\}_{i=1}^N$ be unit-norm orthonormal semantic centroids, and let $\hat{\mathbf{e}} = \mathbf{e} / \|\mathbf{e}\|_2$ lie in their span. The least-squares coordinates*

$$\mathbf{c}^* = \arg \min_{\mathbf{c} \in \mathbb{R}^N} \left\| \hat{\mathbf{e}} - \sum_{i=1}^N c_i \boldsymbol{\mu}_i \right\|_2^2 \quad (15)$$

are

$$c_i^* = \boldsymbol{\mu}_i^\top \hat{\mathbf{e}} = \cos \angle(\mathbf{e}, \boldsymbol{\mu}_i). \quad (16)$$

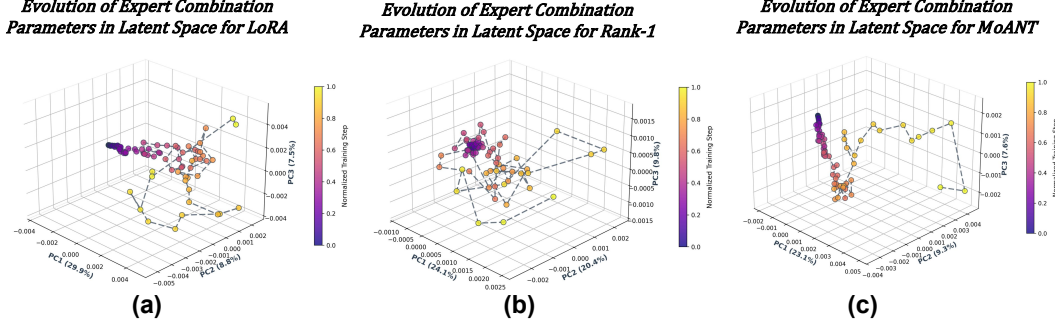


Figure 3: 3D trajectory of router weights for (a) traditional LoRA, (b) rank-1 experts with Mix-and-Match, and (c) MoANT with semantic intuition during MoE training.

Proof. For orthonormal centroids, the least-squares solution is the orthogonal projection of $\hat{\mathbf{e}}$ onto their span. The coordinates are the inner products $\mu_i^\top \hat{\mathbf{e}}$, which equal cosine similarities since both vectors are normalized. ■

In MoANT, $N = IJ$. The cosine scores are reshaped into $\mathbf{S}(\mathbf{x}) \in \mathbb{R}^{I \times J}$ and fused with learnable routing logits $\mathbf{R}(\mathbf{x})$:

$$\tilde{\mathbf{G}}(\mathbf{x}) = \text{reshape}_{I \times J}(\text{softmax}(\text{vec}(\mathbf{R}(\mathbf{x})) + \lambda \text{vec}(\mathbf{S}(\mathbf{x}))). \quad (17)$$

The final update is $\Delta \mathbf{W}(\mathbf{x}) = \mathbf{A} \tilde{\mathbf{G}}(\mathbf{x}) \mathbf{B}^\top$. Although real centroids are not necessarily orthogonal, the proposition motivates centroid similarity as a coordinate-like routing prior, while the learnable router can adapt this prior during fine-tuning.

5 Experiment

5.1 Implementation Details, multitask Datasets and baselines

Implementation details. We leverage an extensive collection of open-source LLM backbones, including LLaMA-3 (Grattafiori et al., 2024), LLaMA-2 (Touvron et al., 2023), Mistral (Jiang et al., 2023), Yi (Young et al., 2024), Gemma (Team et al., 2024), and TinyLLaMA (Zhang et al., 2024a). The MoE module is implemented in alignment with conventional LoRA standards. To generate instance embeddings, we utilize nomic-embed-text-v1 (Nussbaum et al., 2024). Our PEFT procedures are performed over three epochs with a batch size of 64, using Adam with a learning rate of $5e-5$ and cosine decay.

Datasets. We create a comprehensive multitask dataset by aggregating 20 datasets from diverse sources, spanning 5 empirically determined dimensions (Contributors, 2023). The reasoning category includes ANLI (Nie et al., 2020), ReCoRD (Wang et al., 2019), and HellaSwag (Zellers et al., 2019); for examination, we feature MMLU (Hendrycks et al., 2021) and ARC (Clark et al., 2018); WiC (Wang et al., 2019) and WinoGrande (Sakaguchi et al., 2021) represent language; OpenBookQA (Mihaylov et al., 2018) and MultiRC (Wang et al., 2019) embody understanding; CommonSenseQA (Talmor et al., 2019) and BoolQ (Wang et al., 2019) epitomize knowledge; Alpaca (Taori et al., 2023) encompasses mixed dimensions.

Baselines. The comparative methodologies include conventional LoRA (Hu et al., 2021) with the rank values set to 32, marked as LoRA-32. Additionally, we involve MoLoRA (Zadouri et al., 2023), which introduces lightweight experts and is capable of generalizing to new tasks independently of prior knowledge, SiRA (Zhu et al., 2023) that enhances LoRA by integrating sparse MoE with capacity-limited top-k expert routing, and HMoRA (Liao et al., 2025) that combines MoE and LoRA with hierarchical hybrid routing and auxiliary loss. The experts in SiRA, MoLoRA, and HMoRA are set as $8 \times \text{LoRA-4}$, aligning the learnable parameters of LoRA-32, while our MoANT adopts a total of 32 Rank-1 experts via the Mix-and-Match (MaM).

Table 1: Performance evaluation across 14 datasets on 4 LLaMA-series backbones.

Method	MUL.	MMLU	BOO.	WIC	WG	WSC	ANLI	PIQA	SIQA	RTE	COPA	OBQA	CSQA	HS	#ATP ↓	AVG ↑
<i>LLaMA-2 13B</i>																
LoRA	90.31	60.48	90.13	75.47	87.66	64.42	76.10	88.70	82.40	90.71	100.00	84.90	84.80	96.49	+0.95%	83.76
MoLoRA	90.35	62.30	89.73	74.84	86.50	66.35	76.90	87.83	82.60	90.00	98.00	85.60	84.64	96.36	+1.06%	83.73
SiRA	90.20	60.94	90.34	74.61	87.19	67.31	76.30	88.79	82.96	90.61	100.00	84.20	84.44	95.44	+1.06%	83.81
HMoRA	90.65	59.76	90.42	73.66	87.13	59.61	75.30	88.03	82.90	90.97	98.00	86.80	86.24	95.80	+2.13%	83.23
MoANT	90.98	59.33	91.24	73.26	86.71	69.27	76.40	88.98	83.65	93.78	98.00	86.40	86.42	96.67	+0.62%	84.37
<i>LLaMA-3 8B</i>																
LoRA	90.00	61.87	90.03	74.39	84.54	65.25	75.70	89.36	83.06	89.95	99.00	88.40	84.38	95.10	+1.03%	83.80
MoLoRA	89.91	61.01	90.00	74.14	83.98	58.65	72.60	89.88	82.96	88.09	97.00	89.40	83.70	95.49	+1.16%	82.63
SiRA	90.18	61.07	90.52	74.76	85.56	63.46	75.90	90.26	83.88	88.81	99.00	87.80	84.93	95.52	+1.16%	83.69
HMoRA	90.18	61.85	90.48	73.04	86.10	71.15	74.70	88.08	82.13	91.33	97.00	86.20	83.94	94.79	+2.24%	83.64
MoANT	90.35	61.33	90.43	74.92	86.19	65.38	78.10	89.45	83.01	90.25	99.00	87.80	84.93	94.95	+0.69%	84.01
<i>LLaMA-2 7B</i>																
LoRA	88.00	52.71	88.56	70.69	80.51	57.69	73.80	84.93	81.47	86.28	95.00	82.20	83.29	93.50	+1.17%	79.90
MoLoRA	88.90	54.02	87.80	70.38	78.69	53.85	71.60	84.82	82.60	87.36	97.00	82.80	82.31	93.86	+1.30%	79.71
SiRA	88.59	53.23	87.49	71.63	78.06	63.46	70.80	84.82	81.63	87.36	97.00	83.80	82.31	93.85	+1.30%	80.29
HMoRA	88.59	54.86	88.44	71.94	81.61	49.03	70.90	84.65	81.98	89.89	96.00	85.80	83.45	94.12	+2.65%	80.09
MoANT	88.51	53.76	88.32	71.16	80.66	68.27	71.50	85.31	82.65	89.89	96.00	83.40	82.80	94.38	+0.76%	81.19
<i>Tiny-LLaMA 1B</i>																
LoRA	83.04	41.80	79.94	60.34	58.72	47.12	51.60	75.41	74.16	84.84	83.00	70.60	73.63	81.99	+2.24%	69.01
MoLoRA	81.70	44.61	78.69	58.62	56.75	52.88	49.60	75.63	72.42	83.03	83.00	67.80	72.40	80.82	+2.51%	68.42
SiRA	81.54	42.85	79.05	57.37	53.91	43.27	47.40	74.10	72.62	81.95	80.00	65.00	71.99	79.51	+2.51%	66.47
HMoRA	80.87	42.58	76.91	67.86	52.48	58.65	52.40	72.68	71.44	80.14	82.00	66.60	70.35	81.84	+4.22%	68.34
MoANT	82.78	43.50	80.61	66.61	59.43	48.08	51.20	75.68	74.05	85.92	87.00	68.80	73.55	81.80	+1.55%	69.93

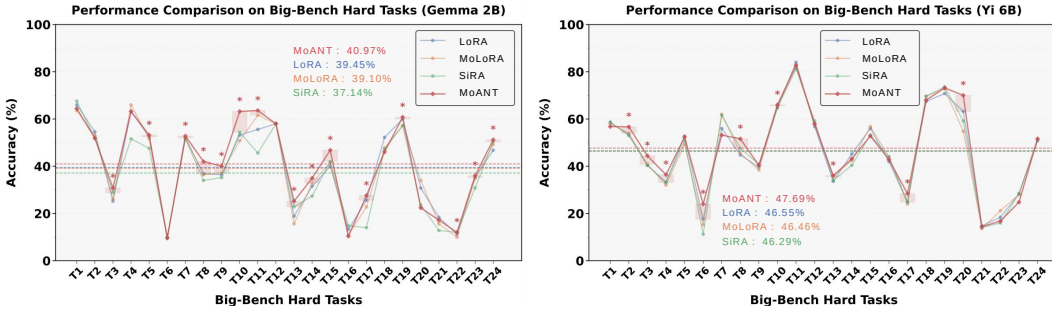


Figure 4: Out-of-distribution generalization capabilities of MoANT compared with baselines, including LoRA, MoLoRA, and SiRA, on the BBH dataset based on model backbone (a) Gemma 2B and (b) Yi 6B.

5.2 Quantitative Results

Performance enhancement. We evaluate MoANT across 14 datasets on seven LLM backbones using four baselines, without additional prompt examples to ensure a fair comparison. As shown in Tab. 1 where #ATP stands for additional trainable parameters, MoANT achieves the highest average score (AVG) across all models, showcasing superior multitask capabilities. Specifically, MoANT demonstrates significant performance improvements in challenging tasks such as WSC and ANLI. For instance, MoANT improves WSC performance by 7.54% on the Mistral model and achieves a substantial 4.80% gain on LLaMA-2 7B, showcasing its ability to handle complex reasoning tasks. **On the WG, COPA, and HS tasks, MoANT outperforms almost all the baselines across multiple backbones**, demonstrating exceptional consistency and robustness, with minimal performance variance across different models. Notably, MoANT attains these gains with substantially fewer additional trainable parameters (#ATP), consistently across settings, e.g., only +0.62% on LLaMA-2 13B and +0.76% on Mistral, highlighting both efficiency and scalability. This contrast is particularly pronounced compared with HMoRA, which requires more than 2% #ATP due to its hybrid router structure. However, **HMoRA remains unstable. It can outperform others by up to 5.77% with LLaMA-3, yet it underperforms by nearly 20% with LLaMA-2 7B on WSC, resulting in a poor average performance.** These results solidify MoANT as an effective MTL solution, excelling in both performance and efficiency.

Table 2: Performance evaluation across 14 datasets on 4 various LLM backbones.

Method	MUL.	MMLU	BOO.	WIC	WG	WSC	ANLI	PIQA	SIQA	RTE	COPA	OBQA	CSQA	HS	#ATP ↓	AVG ↑
<i>Mistral 7B</i>																
LoRA	90.37	59.63	90.61	72.41	86.03	65.38	76.30	87.83	81.93	90.97	96.00	86.20	84.44	95.06	+1.15%	83.08
MoLoRA	90.14	59.05	90.49	73.82	86.50	66.35	75.50	88.52	81.63	90.61	97.00	87.80	83.37	95.12	+1.28%	83.28
SiRA	90.00	58.46	89.91	74.14	86.98	60.58	75.10	89.23	82.86	89.89	96.00	87.40	83.62	95.07	+1.28%	82.80
HMoRA	89.89	57.54	90.03	71.47	83.50	53.84	73.90	88.13	81.11	88.44	98.00	85.20	83.21	93.77	+2.47%	81.29
MoANT	90.24	62.97	90.76	74.76	87.06	70.19	78.30	90.70	83.57	90.61	99.00	87.60	86.08	96.21	+0.76%	84.86
<i>Yi 6B</i>																
LoRA	89.93	62.77	89.08	73.82	83.35	63.46	71.20	87.70	81.93	88.81	96.00	87.60	84.36	95.18	+1.18%	82.51
MoLoRA	89.83	61.66	88.96	73.98	83.82	63.46	71.70	87.92	82.45	88.81	96.00	87.20	83.37	95.10	+1.33%	82.45
SiRA	89.44	61.79	88.96	73.51	83.50	68.27	72.10	87.92	83.01	88.09	97.00	88.60	85.42	95.21	+1.33%	83.06
HMoRA	89.33	59.56	88.16	73.82	82.24	69.23	68.60	84.76	79.52	86.64	95.00	85.60	83.78	93.34	+2.82%	81.40
MoANT	89.47	62.57	88.86	73.75	83.87	64.50	72.10	88.05	82.43	88.92	98.00	87.20	85.06	95.68	+0.82%	83.89
<i>Gemma 2B</i>																
LoRA	84.53	43.50	85.29	71.16	67.56	60.58	62.50	79.54	76.87	85.56	83.00	75.20	73.63	87.49	+1.54%	74.03
MoLoRA	85.00	45.20	85.11	70.06	65.59	49.04	60.50	79.92	76.87	86.28	88.00	77.00	75.43	86.92	+1.70%	73.64
SiRA	84.74	44.55	84.83	69.75	66.85	52.88	60.30	78.94	75.64	84.12	88.00	75.80	75.51	86.29	+1.70%	73.44
HMoRA	83.45	42.32	83.09	68.96	65.19	53.84	56.10	76.87	73.33	84.47	85.00	71.80	70.51	81.76	+2.29%	71.20
MoANT	85.52	45.85	84.89	68.65	68.67	57.69	63.00	80.36	76.82	88.09	89.00	77.20	76.74	87.85	+0.96%	75.02

Efficiency analysis. We extend our analysis to evaluate MoANT’s efficiency. To this end, we benchmark MoANT against four established baselines by comparing the number of #ATP and the **additional** computational cost during training, measured in FLOPs and throughput speed, on the Mistral 7B model. The results, summarized in Tab. 3, demonstrate a significant reduction in computational overhead during training compared to LoRA, SiRA, MoLoRA, and especially for HMoRA. Specifically, MoANT adds only 39.9M parameters, an additional 7.86G FLOPs, and a training speed of 6.92 it/s on a commercial-grade RTX 3090, while achieving superior performance with 84.86 accuracy. In contrast, HMoRA fails to capitalize on its 130M additional trainable parameters, resulting in a slow throughput of 6.48 it/s and an accuracy of only 81.29. In addition, although LoRA attains a similar throughput of 6.87 it/s, MoANT shows a 1.33% performance gain, clearly demonstrating the effectiveness of our approach. These findings highlight that **MoANT optimizes resource usage, delivering better performance with a smaller training footprint.**

Table 3: Additional computational overhead with different methods on GeForce RTX 3090.

Method	Rank	#ATP ↓	FLOPs ↓	Speed ↑	Acc. ↑
<i>Mistral 7B</i>					
LoRA	32+32	83.9M	9.04G	6.87it/s	83.13
SiRA	32+32	93.8M	9.45G	6.32it/s	82.80
MoLoRA	32+32	93.8M	9.45G	6.43it/s	83.28
HMoRA	32+32	129.7M	13.05G	6.48it/s	81.29
MoANT	4+8	39.9M	7.86G	6.92it/s	84.86

Intuition generalizability. We investigate the generalization ability of MoANT, attributed to its intuition-guided design. To isolate the impact of intuition, **we exclude the Mix-and-Match from MoANT, focusing solely on the effectiveness of prior knowledge guidance.** Specifically, we randomly exclude five sub-datasets from the training set in each experiment, treating the excluded *Drop-X* datasets as out-of-distribution (OOD) and the remaining ones as in-distribution (ID). As shown in Tab. 4, MoANT consistently achieves improvements in both the OOD and ID scenarios. Notably, for the *Drop-A* setting, MoANT improves ID performance by 2.48% and OOD performance by 1.55%, respectively. These results highlight the strong generalization capability of our intuition-guided routing.

Table 4: OOD and ID data exploration of MoANT based on Tiny-LLaMA 1B.

Method	Data	Drop-A	Drop-B	Drop-C	Drop-D
LoRA	ID	68.58	72.43	68.45	67.15
MoANT	ID	71.22	72.61	71.69	68.03
LoRA	OOD	43.87	52.68	50.35	59.41
MoANT	OOD	45.42	53.43	50.55	59.71

In this study, we evaluated the OOD generalization capabilities of our proposed MoANT on the Big-Bench Hard (BBH) dataset (Suzgun et al., 2022) in a zero-shot setting, **with similar computational overhead baselines without any fine-tuning.** The results², presented in Fig 4, demonstrate that MoANT consistently outperforms competing methods across various LLM backbones. This superior performance is attributed to its ability to effectively integrate multitask priors into the MoE, thereby enhancing OOD knowledge acquisition. For instance, MoANT achieves accuracies of 40.97% and 47.69% on Gemma 2B and Yi 6B, consistently surpassing LoRA by over 1%.

²Zero-value points were excluded for clarity in visualization.

5.3 Ablation Studies

Mix-and-Match strategy. We conduct additional experiments to evaluate the effectiveness of the Mix-and-Match (MaM) strategy, which leverages the flexible combination of Rank-1 experts, as detailed in Tab. 5. We first compare a traditional rank-1 MoE-LoRA with 32 experts to MoANT with 4×8 experts. The rank-1 LoRA-MoE attains 69.40 accuracy at a substantially higher computational cost of only 9.13it/s, whereas MoANT reaches 69.93 accuracy with more than $2 \times$ fewer #ATP. **These results provide initial evidence for the effectiveness of the MaM strategy.**

Table 5: Computational costs with different Mix-and-Match strategies on Nvidia GeForce RTX 3090.

Method	Experts	#ATP ↓	FLOPs ↓	Speed ↑	Acc. ↑
<i>Tiny-LLaMA 1B</i>					
Rank-1	32×32	37.85M	8.49G	9.13it/s	69.40
MoANT	4×16	33.12M	8.13G	9.41it/s	70.11
	8×8	31.54M	8.00G	9.54 it/s	70.75
	1×32	25.63M	7.99G	10.01it/s	67.52
	2×16	19.71M	7.00G	10.14 it/s	67.65
	4×8	17.35M	6.81G	10.32 it/s	69.93

In addition, we further analyze five configurations with different numbers of matrices A and B. Among these, the 8×8 configuration achieves the highest accuracy of 70.75, albeit at the cost of increased computational overhead, with an 8G increase in FLOPs. Interestingly, we observe that **the closer the ranks of the low-rank matrices are, the better the model’s performance.** For example, compared to the 4×16 configuration, the 8×8 configuration achieves an accuracy of 70.75. Among the 32-expert settings, the 4×8 configuration achieves the highest accuracy of 69.93. From a theoretical perspective, this can be attributed to the fact that balancing the number of A and B experts ensures a higher rank in the final AGB^T matrix and equalizes the gradient scale between the A and B matrices. **For fair computational comparisons, we use the 4×8 configuration as the default in all experiments.**

Module functionality. To validate the superior performance of MoANT, we evaluated the effectiveness of implicit intuition and Rank-1 experts on various LLM backbones. As shown in Tab. 6, both components consistently enhance LLM performance. For instance, incorporating implicit intuition and Rank-1 experts into LLaMA-2 7B yields performance MoANT of up to 1.12% and 0.70%, respectively. These components also complement each other, working synergistically to further elevate MoANT’s capabilities. Notably, their combination yields a significant 1.35% improvement over Tiny-LLaMA, underscoring the effectiveness of our approach. Moreover, **when integrating the MaM mechanism, MoANT achieves a performance boost of 1.02% while reducing parameter overhead by 40.6% for Mistral 7B.** In addition, since k-means clustering is computationally lightweight and requires only a single offline step, its centroids can be frozen and the process offloaded to the CPU, preventing interference with GPU training. It should be noted that the **intuition module reveals a minimal performance impact, with inference speeds of 10.83 iter/s compared to the full model’s, indicating an overhead of just 4.94%.**

Table 6: Ablation study and efficiency improvement of MoANT with different modules on different backbones.

Intu.	R-1	LLaMA-2	Mistral	Phi-2	T-LLaMA
✗	✗	79.90	83.08	79.18	69.01
✓	✗	81.02	83.66	79.84	69.92
✗	✓	80.60	83.32	79.41	69.70
✓	✓	81.11	83.84	80.00	70.36
+ MaM		81.19	84.86	79.95	69.93
#ATP		-41.5%	-40.6%	-38.5%	-38.3%

6 Conclusions

We propose MoANT, a method that enhances LLM adaptability in multi-task learning by leveraging semantic-aware implicit intuitions and Mix-and-Match Rank-1 experts. Through extra-low-rank experts, MoANT enables a linear combination of low-rank matrices, expanding the vector subspace to improve performance and efficiency via the MaM mechanism. Extracting implicit intuition from semantic diversity via text embeddings guides all experts toward generalizable convergence. Rigorous theoretical analysis shows that MoANT expands the model’s subspace and steers optimization toward low-loss regions during fine-tuning. Comprehensive experiments further demonstrate that MoANT outperforms state-of-the-art baselines in both in-distribution and out-of-distribution settings, achieving superior performance and efficiency.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (62476011) and (625B2090), the National Key Research and Development Program of China under Grant 2021YFA0717700, the Beijing Natural Science Foundation (L252060), and the Beijing Major Science and Technology Project under Contract no. Z191100010618003, and also in part by RGC GRF 15201322, 15230624, 15239925, 16213526, CRF C5020-25GF, ITC ITS/052/23MX, and a start up fund of HKUST.

References

- Colton T Beardsley, Mathieu Amiraux, and Chad Stucki. Application of multi-task learning architectures to aerodynamic surrogate modeling. In *AIAA SCITECH 2026 Forum*, pp. 1116, 2026.
- Rich Caruana. Multitask learning. In *ML*, 28:41–75, 1997.
- Chang Che, Ziqi Wang, Pengwan Yang, Cheems Wang, Hui Ma, and Zenglin Shi. Lora in lora: Towards parameter-efficient architecture expansion for continual visual instruction tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19978–19986, 2026.
- Yuanteng Chen, Peisong Wang, Nanxin Zeng, Yuantian Shao, Gang Li, Jing Liu, and Jian Cheng. Certain head, uncertain tail: Expert-sample for test-time scaling in fine-grained moe. *arXiv preprint arXiv:2602.02443*, 2026.
- Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pp. 794–803, 2018.
- 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.
- OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. <https://github.com/open-compass/opencompass>, 2023.
- Damai Dai, Li Dong, et al. Stablemoe: Stable routing strategy for mixture of experts. In *ACL*, pp. 7085–7095, 2022.
- Ning Ding, Yujia Qin, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. In *NMI*, 5(3):220–235, 2023.
- Ruogu Ding, Xin Ning, Ulf Schlichtmann, and Weikang Qian. Prefixgpt: Prefix adder optimization by a generative pre-trained transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20808–20815, 2026.
- Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *CVPR*, pp. 13733–13742, 2021.
- William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. [abs/2101.03961](https://arxiv.org/abs/2101.03961), 2021.
- 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.
- Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. A survey on uncertainty quantification methods for deep learning. *ACM Computing Surveys*, 58(7):1–35, 2026.
- Dan Hendrycks, Collin Burns, et al. Measuring massive multitask language understanding. In *ICLR*, 2021.

- J. Edward Hu, Yelong Shen, et al. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021.
- Chengsong Huang, Qian Liu, et al. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*, 2023.
- Robert A. Jacobs, Michael I. Jordan, et al. Adaptive mixtures of local experts. In *Neural Computing*, 3(1):79–87, 1991.
- Albert Qiaochu Jiang, Alexandre, et al. Mistral 7b. *ArXiv*, abs/2310.06825, 2023.
- Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. In *Neural Computing*, 6(2):181–214, 1994.
- Lukas Kirhdorfer, Tobias Sesterhenn, Christian Bartelt, Heiner Stuckenschmidt, Lukas Schott, and Jan M Köhler. Investigating uncertainty weighting for multi-task learning: Insights and analytical alternative. *International Journal of Computer Vision*, 134(1):8, 2026.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 30, 2017.
- Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.
- Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.
- 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.
- Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. In *AIR*, 42:275–293, 2014.
- 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.
- Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pp. 3994–4003, 2016.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, 2020.
- Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*, 2024.
- 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.
- Noam Shazeer, Azalia Mirhoseini, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.
- Mirac Suzgun, Nathan Scales, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *ACL: Human Language Technologies*, volume 1, 2019.
- Rohan Taori, Ishaan Gulrajani, et al. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

- Gemma Team, Thomas Mesnard, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.
- Hugo Touvron, Louis Martin, et al. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023.
- Alex Wang, Yada Pruksachatkun, et al. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*, 2019.
- Pengcheng Wen, LU Guoying, Sirui Han, and GUO Yike. Legal aligner: Transforming generic llms into domain experts for enhanced accuracy. 2026.
- Yuanbo Yang, Jiahui Qu, Wenqian Dong, Ling Huang, and Yunsong Li. Prototype-based meta-prompt tuning: Toward rehearsal-free few-shot class-incremental learning for multi-modal remote sensing image. *IEEE Transactions on Image Processing*, 2026.
- Alex Young, Bei Chen, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.
- Ted Zadouri, Ahmet Üstün, et al. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arXiv preprint arXiv:2309.05444*, 2023.
- Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *ACL*, 2019.
- Honglei Zhang, Zhiwei Li, Haoxuan Li, Xin Zhou, Jie Zhang, and Yidong Li. Transfr: Transferable federated recommendation with adapter tuning on pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28212–28220, 2026a.
- Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *ArXiv*, abs/2401.02385, 2024a.
- Renrui Zhang et al. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.
- Rongyu Zhang, Yulin Luo, et al. Efficient deweahter mixture-of-experts with uncertainty-aware feature-wise linear modulation. In *AAAI*, volume 38, pp. 16812–16820, 2024b.
- Rongyu Zhang, Menghang Dong, Yuan Zhang, Liang Heng, Xiaowei Chi, Gaole Dai, Li Du, Dan Wang, Yuan Du, and Shanghang Zhang. Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18764–18772, 2026b.
- Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, et al. Instruction tuning for large language models: A survey. *ACM Computing Surveys*, 58(7):1–36, 2026c.
- Yun Zhu, Nevan Wichers, et al. Sira: Sparse mixture of low rank adaptation. *arXiv preprint arXiv:2311.09179*, 2023.