T-REX: MIXTURE-OF-RANK-ONE-EXPERTS WITH SEMANTIC-AWARE INTUITION FOR MULTI-TASK LARGE LANGUAGE MODEL FINETUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) encounter significant adaptation challenges in diverse multitask finetuning. Mixture-of-experts (MoE) provides a promising solution with a dynamic architecture, enabling effective task decoupling. However, scaling up the number of MoE experts incurs substantial parameter and computational overheads and suffers from limited performance gain due to naive routing mechanisms. In this paper, we design a novel framework, mix**T**ure-of-**R**ank-on**E**-e**X**perts (`T-REX`), 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 to quadratically expand the vector subspace of experts with linear parameter overheads, achieving approximate error reduction with optimal efficiency. In addition, T-REX offers implicit guidance to the router, leveraging the inherent semantic clustering of training embeddings as prior knowledge, enabling optimized feature allocation across experts for a smoother convergence. Extensive theoretical and empirical results demonstrate that T-REX achieves superior efficiency and generalizability across diverse tasks in both in-distribution and out-of-distribution scenarios. Compared with traditional LoRA, T-REX achieves up to 1.78% mean accuracy improvement with around 30%-40% less trainable parameters across 14 public datasets.

1 INTRODUCTION

Large language models (LLMs) (Touvron et al., 2023; Jiang et al., 2023; Team et al., 2024) exhibit profound capabilities across various domains, excelling in specific downstream tasks including content generation, enhancing interactive entertainment, and inspiring artistic endeavors. However, despite the transformative impact of LLMs on natural language processing (NLP), as evidenced by their mastery of unsupervised pretraining and subsequent supervised fine-tuning (Taori et al., 2023; Hu et al., 2021), the intricate settings of complex multitask learning (MTL) environments present a unique challenge. The diversity of tasks in these environments tests the adaptability of LLMs (Zhang et al., 2023a; Sung et al., 2022), pushing the boundaries of their application and capability.

Mixture-of-Experts (MoE) (Masoudnia & Ebrahimpour, 2014; Shazeer et al., 2017; Dai et al., 2022; Jacobs et al., 1991) has recently gained attention as a practical approach for multi-task finetuning, leveraging its dynamic architecture to adapt the generally pretrained LLMs to diverse downstream tasks efficiently. Like humans, who demonstrate superior multitasking abilities by selectively activating specific regions of the brain based on explicit cues, MoE dynamically engages and combines different experts for each particular 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 (Zadouri et al., 2023; 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 (Luo et al., 2023; Zhang et al., 2024b) have introduced uncertainty-aware mechanisms (Lakshminarayanan et al., 2017; Zhang et al., 2023b; Liu et al., 2022) and explicit feature guidance to enhance routing, but their effectiveness in MTL remains unproven.

To address these challenges, we propose a novel paradigm, mix**T**ure-of-**R**ank-on**E**-e**X**perts (`T-REX`), which leverages ❶ ultra-lightweight rank-1 experts with a mix-and-match mechanism. Specifically, T-REX 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 mix-and-match mechanism enables T-REX to construct more LoRA experts to cover a higher-dimensional subspace in the weight matrix space with a sub-linear growth of additional parameters. This mix-and-match strategy also facilitates more effective interaction and cross-fusion of shared feature representations across tasks, ultimately boosting the model's generalization capability. Mathematically, T-REX is proven to increase representational capacity over other LoRA-based methods, therefore achieving higher accuracy empirically across tasks.

In addition, we propose ❷ a semantic-aware router inspired by the brain's ability for intuitive cognition, enabling T-REX to better allocate experts for complex multitask scenarios. Naively, the router can utilize the human-annotated task categorization as a condition proxy as *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 OpenCompass[1], 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 using the clustering of training embeddings. We assign inputs to experts based on the similarity between input embeddings and the corresponding task centroids. Specifically, T-REX leverages the correlation between the input and these centroids as an *implicit intuition*, which conditions the router's decision-making process and guides all experts to converge smoothly.

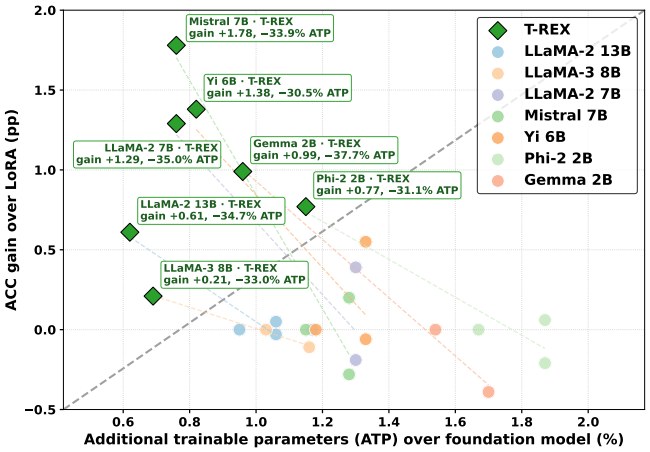

Figure 1: Accuracy $v.s.$ efficiency for different methods. Circles with identical colors denote different PEFT baselines on the same LLM.

Both theoretical derivations and extensive experiments demonstrate the superiority of our proposed T-REX against state-of-the-art baselines. As shown in Fig. 1, T-REX enables a sizable gain of both performance and efficiency over the Pareto frontier of tradeoff for existing LoRA-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 T-REX include:

- We devise a novel LoRA MoE paradigm that elaborately leverages ultra-lightweight rank-1 experts with a mix-and-match mechanism for efficient multi-task finetuning.
- 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.
- We provide a theoretical analysis showing that T-REX achieves expert linear combination subspace expansion, enabling approximate error reduction, while 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 (Kendall et al., 2018; Chen et al., 2018) propose adaptive loss weighting to balance multi-task losses. Models like Cross-stitch (Misra et al., 2016) and Sluice (Ruder et al., 2019) networks dynamically blend task-specific representations but use static task weights. Our method utilizes a MoE to achieve precise task allocation by assigning input to a combination of specialized experts.

Mixture of Experts. MoE was initially introduced by (Jacobs et al., 1991; Jordan & Jacobs, 1994) to process different samples with independent modules. For example, (Shazeer et al., 2017) employs MoE in large-scale LSTM-based (Hochreiter & Schmidhuber, 1997) models. GShard (Lepikhin et al., 2021) and Switch Transformer (Fedus et al., 2021) firstly introduce MoE in Transformer, and largely scale up model size with top-1/2 routing strategies. (Roller et al., 2021; Dai et al., 2022) stabilize the training process with fixed routing strategies. (Zhou et al., 2022) refines the routing strategy, which allows tokens to be assigned to varying numbers of experts. ST-MoE (Zoph, 2022) optimizes the training instability and fine-tuning complexity in MoE models. (Zadouri et al., 2023; Zhu et al., 2023) propose parameter-efficient adapters as experts to decrease MoE size. (Puigcerver et al., 2023) proposes a fully-differentiable sparse MoE to address the training instability and token dropping problems. Our method enhances routing with multi-task data insights and employs a rank-1 expert for efficiency.

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

[1] https://github.com/open-compass/opencompass

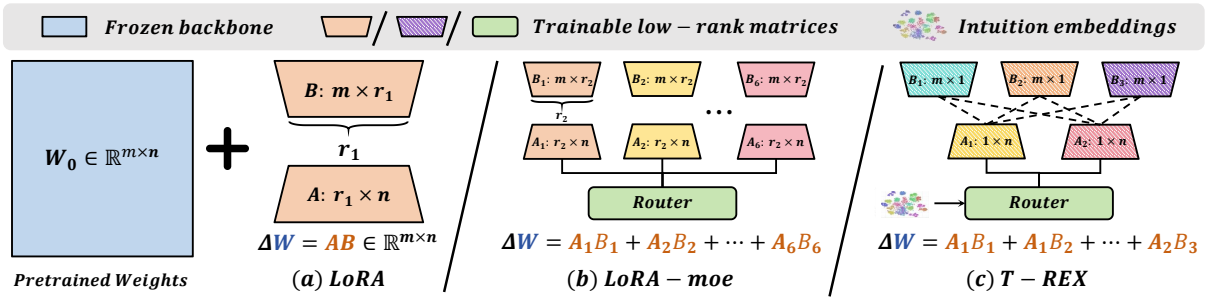

Figure 2: Illustration of the trainable parameters in (a) Vanilla LoRA, (b) LoRA-MoE, and (c) our proposed T-REX. $r_1, r_2 \ll min\{m, n\}$. 6 experts are demonstrated for both LoRA-MoE and T-REX.

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 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:

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

where $\mathbf{G}(\mathbf{x})$ represents the routing weight for the i-th expert, dynamically computed for each input $\mathbf{x}$.

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 and enhance the flexibility of MoE. Specifically, we start with a decoupled row-column subspace decomposition of the LoRA matrix, 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:

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

$\{\mathbf{a}_i\}$ and $\{\mathbf{b}_i\}$ represent trainable base vectors spanning the row and column subspaces, respectively. With N such experts, the layer output is computed as:

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

where $\mathbf{A} = [\mathbf{a}_1|\cdots|\mathbf{a}_N]$ and $\mathbf{B} = [\mathbf{b}_1|\cdots|\mathbf{b}_N]$ form the factorized LoRA weights.

The above rank-1 MoE serves as the most compact design of the LoRA-MoE. However, the number of experts in such design is still limited by the rank of LoRA weights $\mathbf{A}$ and $\mathbf{B}$, where at most N experts can be established given rank-N LoRA weights. To enhance flexibility and reduce parameter overheads in scaling up the number of experts, we decouple the one-to-one correspondence between the row and column subspaces represented by $\mathbf{a}_i$ and $\mathbf{b}_i$ in conventional LoRA. 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 a linear scaling of the rank to a quadratic scaling by enabling cross-combinations. The overall output is computed as:

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

where $I = \text{Rank}(\mathbf{A})$, $J = \text{Rank}(\mathbf{B})$, and the routing matrix $\mathbf{G}(\mathbf{x}) \in \mathbb{R}^{I \times J}$ no longer required to be square or diagonal, dynamically activates these cross-combined experts. Furthermore, the mix-and-match mechanism can be expressed as a Kronecker product expansion:

$$\sum_{i,j=1}^{N} G_{ij}(\mathbf{x})(\mathbf{a}_i \otimes \mathbf{b}_j) = (\mathbf{A} \otimes \mathbf{B})\text{vec}(\mathbf{G}), \tag{5}$$

where $\mathbf{A} \otimes \mathbf{B}$ represents the Kronecker product of the row and column subspaces, $\text{vec}(\mathbf{G})$ flattens the routing matrix, and $N = I \times J$ is the total number of experts. This design achieves quadratic scaling of expressive power with only linearly increased additional parameters, fundamentally differing from the capacity-parameter tradeoff of standard LoRA. The pseudo-code is shown in Algorithm 1.

3.3 Semantic-aware router with intuition clustering

Naive MoE lacks a meaningful correlation between the expert routing and the task, leading to either under-trained or redundant experts. In T-REX, we propose to implicitly assign a specific "task" for each expert to specialize in, therefore effectively covering the MTL space. However, we observe that the human-defined training task in the MTL process often leads to ambiguity in the data distribution and the model capability required. For example, we visualize token embeddings of 20 datasets across five human-defined categories[2]. We observe that embeddings of the same task category distribute throughout the space, whereas some embeddings from diverse tasks appear in close clusters. This demonstrates that predefined training tasks are coarse-grained and involve diverse skills. To this end, we propose to design the "task" of each expert based on the semantic clusters of the training set embeddings. Each expert will be encouraged to better process the tokens within a semantic cluster, while the token will be routed to the experts based on its similarity with the cluster centroids.

Formally, suppose we have N experts in the T-REX. We start by gathering a set of embeddings from the multitask training tokens $\hat{\mathbf{e}} = \boldsymbol{E}(\hat{\mathbf{x}}) \in \mathbb{R}^d$ using an Ada-embedding model. These embeddings are then grouped into N semantic clusters, corresponding to N experts, using k-means clustering:

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

where $\boldsymbol{\mu}_i \in \mathbb{R}^d$ is the centroid of the i^{th} cluster. During finetuning, we compute *intuition scores* $\mathbf{I}(\mathbf{x}) \in \mathbb{R}^N$ for an input $\mathbf{x}$ using cosine similarity between its embedding e and the cluster centroids. For each cluster centroid $\boldsymbol{\mu}_i$, the intuition score $\mathbf{I}_i$ measures the similarity:

$$\mathbf{I}_i(\boldsymbol{x}) = \cos(\angle(\boldsymbol{e}, \boldsymbol{\mu}_i)) = \frac{\boldsymbol{e}^\top \boldsymbol{\mu}_i}{||\boldsymbol{e}|| \cdot ||\boldsymbol{\mu}_i||}. \tag{7}$$

The intuition vector $\mathbf{I}(\mathbf{x}) \in \mathbb{R}^N$ gathering all intuition scores is then fused with the base routing weights $\mathbf{G}(\mathbf{x}) \in \mathbb{R}^N$ through element-wise addition:

$$\widetilde{\mathbf{G}}(\mathbf{x}) = \mathbf{G}(\mathbf{x}) \oplus \mathbf{I}(\mathbf{x}), \tag{8}$$

where $\oplus$ denotes element-wise addition. This enhanced routing signal is used to construct the final expert combination, enabling a more semantically informed selection of experts.

4 Theoretical analysis of T-REX

4.1 Rank-1 experts with subspace expansion

Lemma 4.1 (Subspace Expansion with Mix-and-Match)**.** *The adaptation matrix $\Delta W = \mathbf{AGB}^\top$ generated by the Mix-and-Match mechanism spans a subspace whose dimensionality grows with a speed of $\mathcal{O}(IJ)$ as LoRA weight ranks I and J increase. Specifically, the vectorized form of ΔW, $\text{vec}(\Delta W)$, spans a space with dimensionality up to:*

$$\dim(\text{span}(\text{vec}(\Delta W))) = I \times J, \tag{10}$$

while the rank of the matrix ΔW itself is bounded by $\text{rank}(\Delta W) \leq \min(I, J)$.

Proof. Under the Mix-and-Match mechanism, any $\mathbf{a}_i$ can pair with any $\mathbf{b}_j$, forming experts $f_{ij}(\mathbf{x}) = \mathbf{a}_i \mathbf{b}_j^\top \mathbf{x}$. Thus, the vectorized form of ΔW can be expressed as:

$$\text{vec}(\Delta W) = (\mathbf{A} \otimes \mathbf{B})\text{vec}(\mathbf{G}), \tag{11}$$

where $\mathbf{A} \in \mathbb{R}^{m \times I}$, $\mathbf{B} \in \mathbb{R}^{n \times J}$ are the Rank-1 expert basis matrices, $\mathbf{G} \in \mathbb{R}^{I \times J}$ is the dynamic routing matrix, and $\otimes$ denotes the Kronecker product. The Kronecker product $\mathbf{A} \otimes \mathbf{B}$ spans a subspace with dimensionality:

$$\dim(\text{span}(\mathbf{A} \otimes \mathbf{B})) = \dim(\mathcal{S}_a \otimes \mathcal{S}_b) = I \times J, \tag{12}$$

which significantly expands the potential representation capability of ΔW. ■

[2]Details are shown in the Appendix B.

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

Input: Pre-trained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{m\times n}$, inputs X, LoRA weights ranks I, J, task centroids $\{\boldsymbol{\mu}_k\}_{k=1}^N, N = I \times J$, embedding model $\boldsymbol{E}$, maximum iterations `MaxIter`.
Output: Adapted model weights $\mathbf{W}_{\text{adapted}}$.
Initialize: $\{\mathbf{a}_i, \mathbf{b}_j\}$ for $i = 1 \dots I, j = 1 \dots J$, routing weight matrix $\mathbf{G}(\cdot) \in \mathbb{R}^{I\times J}$.
for *Iter* $= 1$ *to* *MaxIter* **do**
▷ Step 1: Compute Routing Weights with Intuition
foreach $\mathbf{x} \in X$ **do**
Base routing: $\mathbf{g}_{\text{base}} = \texttt{softmax}(\mathbf{G}(\mathbf{x}))$.
Intuition scores: $\mathbf{I}(\mathbf{x}) = [\cos(\boldsymbol{E}(\mathbf{x}), \boldsymbol{\mu}_k)]_{k=1}^N$.
Fuse routing: $\widetilde{\mathbf{G}}(\mathbf{x}) = \texttt{softmax}(\mathbf{g}_{\text{base}} \oplus \mathbf{I}(\mathbf{x}))$. ▷ Enhanced routing
▷ Step 2: Compute Mix-and-Match with Rank-1 Experts

$$\Delta\mathbf{W}_{\text{combined}}(\mathbf{x}) = \sum_{i=1}^{I}\sum_{j=1}^{J} \widetilde{\mathbf{G}}_{ij}(\mathbf{x})\mathbf{a}_i\mathbf{b}_j^\top \tag{9}$$

Return: $\mathbf{W}_{\text{adapted}} = \mathbf{W}_0 + \Delta\mathbf{W}_{\text{combined}}$.

Theorem 4.2 (Approximation Error Bound with Rank-1 Experts)**.** *For a target adaptation matrix $\Delta W^\star$, T-REX decomposes the matrix into in-subspace and residual components:*

$$\Delta W^\star = \Delta\mathbf{W}_{\text{in}} + \Delta\mathbf{W}_{\text{out}}, \quad \Delta\mathbf{W}_{\text{in}} \in \mathcal{S}_a \otimes \mathcal{S}_b, \quad \Delta\mathbf{W}_{\text{out}} \perp \mathcal{S}_a \otimes \mathcal{S}_b. \tag{13}$$

The Frobenius norm approximation error satisfies:

$$\|\Delta W - \Delta W^\star\|_F^2 \le \|\Delta\mathbf{W}_{\text{out}}\|_F^2 + C \cdot \frac{1}{IJ}\|\Delta W^\star\|_F^2 + \mathcal{O}\left(\frac{1}{\tau}\right), \tag{14}$$

where: - $\|\Delta\mathbf{W}_{\text{out}}\|_F^2$ quantifies the residual error outside the subspace $\mathcal{S}_a \otimes \mathcal{S}_b$; - The second term reflects the alignment error of $\Delta W^\star$ with the subspace, which decreases as IJ increases; - The third term $\mathcal{O}(1/\tau)$ accounts for the effect of the softmax temperature τ in the dynamic routing matrix $\mathbf{G}(\mathbf{x})$, which sharpens the subspace activation.

Proof. By the orthogonal projection theorem, the target matrix $\Delta W^\star$ can be decomposed into in-subspace $\Delta\mathbf{W}_{\text{in}}$ and residual $\Delta\mathbf{W}_{\text{out}}$ components. The in-subspace component is represented by $(\mathbf{A} \otimes \mathbf{B})\text{vec}(\mathbf{G})$. Thus, the approximation error can be written as:

$$\|\Delta W - \Delta W^\star\|_F^2 = \|\Delta\mathbf{W}_{\text{out}}\|_F^2 + \|\Delta\mathbf{W}_{\text{in}} - \Delta W^\star\|_F^2. \tag{15}$$

For the subspace alignment error $\|\Delta\mathbf{W}_{\text{in}} - \Delta W^\star\|_F^2$, the error decreases with the subspace dimensionality IJ. Specifically, the error term satisfies:

$$\|\Delta\mathbf{W}_{\text{in}} - \Delta W^\star\|_F^2 \propto \frac{1}{IJ}\|\Delta W^\star\|_F^2. \tag{16}$$

Additionally, the routing matrix $\mathbf{G}(\mathbf{x})$'s softmax temperature τ controls the activation sharpness of virtual experts. As τ increases, $\mathbf{G}(\mathbf{x})$ becomes more selective, reducing the error fluctuation, with a contribution approximated as $\mathcal{O}(1/\tau)$. A detailed derivation is provided in Appendix A. ■

4.2 Intuition Leads to Low-Loss Region

The previous section proves the ability of mix-and-match rank-1 experts to optimally approximate the adaptation matrix of a specific task. In this section, we discuss how the intuition vector proposed in Eq. (7) guides the combination of task-specific experts to fulfill diverse tasks.

Theorem 4.3 (Expert Combination with Intuition Guidance)**.** *For a multitask learning task objective defined on an input space $\mathcal{X} \subset \mathbb{R}^{1\times m}$ as*

$$\mathcal{L}(\Delta W, x) = \sum_{i=1}^{N} ||x\Delta W - \mu_i \Delta W_i^\star||^2, \tag{17}$$

where $\mu_1, \cdots \mu_N$ forms the orthonormal basis of the input subspace $\mathcal{X}$ and $\Delta W_1^\star, \cdots, \Delta W_N^\star$ are the target adaptation matrices for the task-specific experts, the intuition vector $\mathbf{I}_i(\boldsymbol{x}) = \cos(\angle(\boldsymbol{x}, \boldsymbol{\mu}_i))$ is proportional to the coordinate vector of the optimal ΔW under the basis of $\Delta W_1^\star, \cdots, \Delta W_N^\star$.

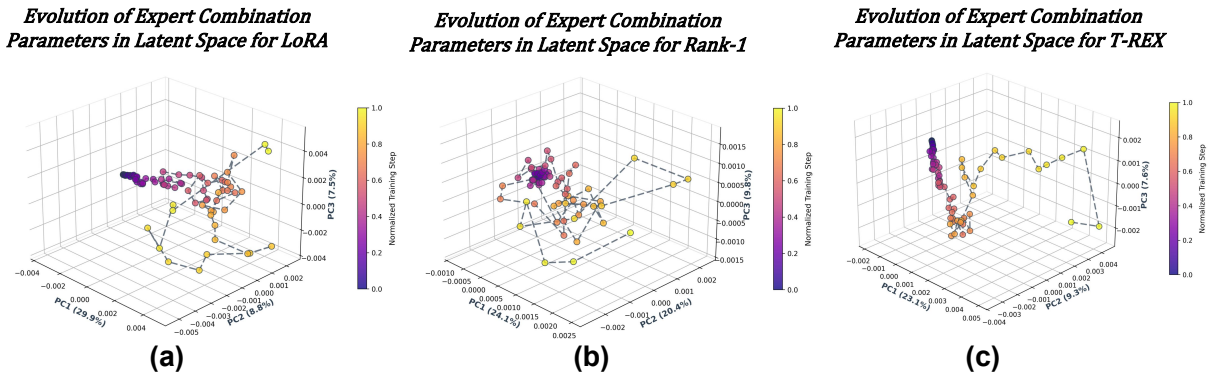

Figure 3: 3D trajectory of router weights for (a) Traditional LoRA, (b) Rank-1 experts with Mix-and-Match, and (c) T-REX with intuition in the MoE training process. Intuition helps to enable a smoother convergence.

Proof. Eq. (17) can be minimized by solving the gradient equation

$$\frac{\partial \mathcal{L}}{\partial \Delta W} = \sum_{i=1}^{N} x^T (x\Delta W - \mu_i \Delta W_i^\star) = 0; \text{solved when } \Delta W = \frac{\sum x^T \mu_i \Delta W_i^\star}{N x^T x}. \tag{18}$$

Therefore, the coordinate vector of the optimal ΔW under the basis of $\Delta W_1^\star, \cdots, \Delta W_N^\star$ is proportional to $x^T \mu_i$. Since $\mu_1, \cdots \mu_N$ forms the orthonormal basis of $\mathcal{X}$, the input x can be represented as $x = \sum_{i=1}^{N} \alpha_i \mu_i$, where $\alpha_i = \cos(\angle(\boldsymbol{x}, \boldsymbol{\mu}_i)) := \mathbf{I}_i(\boldsymbol{x})$ by definition, we have $x^T \mu_i = \mathbf{I}_i(\boldsymbol{x})$. ■

In the case of T-REX, though much more complicated than the task in the theorem, we show that the intuition vector serves as a good initialization and guidance for the learnable router to decompose the optimal adaptation matrix of each input to the basis of mix-and-match rank-1 experts. This is demonstrated by the smoother router convergence trajectory evidenced in Fig. 3.

5 Experiment

5.1 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), Phi (Javaheripi et al., 2023), Gemma (Team et al., 2024), and Tiny-LLaMA (Zhang et al., 2024a). The MoE module is implemented in the transformer blocks, explicitly targeting the query, key, value, and feed-forward network components, aligning with conventional LoRA standards. To generate instance embeddings, we utilize nomic-embed-text-v1 (Nussbaum et al., 2024), the most downloaded embedding model on Hugging Face.

Our PEFT procedures are performed with a 64 batch size, a 5e-5 learning rate adjusted with a cosine learning rate scheduler and Adam optimizer, spanning three epochs. We must note that we do not include additional examples in the prompts during the evaluation process, as various prompting techniques can influence the final results. All experiments are conducted on NVIDIA A100 GPUs, with the GPU hours varying between 30 and 300.

5.2 Multitask Datasets and Baselines

Datasets. We curate a comprehensive multitask dataset by aggregating twenty datasets from diverse sources, spanning five empirically determined dimensions (Contributors, 2023): reasoning, examination, language, understanding, and knowledge. The reasoning category includes datasets such as ANLI (Nie & Williams, 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 (ai2, 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. Additionally, Alpaca (Taori et al., 2023), which encompasses mixed dimensions, is also included in our training set.

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 two state-of-the-art LoRA-based MoE methods, including 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 introduces a novel expert dropout method to mitigate overfitting,

Table 1: Performance evaluation across 14 datasets on 7 LLM backbones compared with four baselines. MUL. indicates MULTIRC and BOO. indicates BOOLQ. #ATP stands for additional trainable parameters.

Method	MUL.	MMLU	BOO.	WIC	WG	WSC	ANLI	PIQA	SIQA	RTE	COPA	OBQA	CSQA	HS	#ATP ↓	AVG ↑
LLaMA-2 13B																
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
T-REX	**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**
LLaMA-3 8B																
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
T-REX	**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**
LLaMA-2 7B																
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
T-REX	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**
Mistral 7B																
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
T-REX	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**
Yi 6B																
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
T-REX	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**
Phi-2 2B																
LoRA	88.00	54.80	86.36	71.94	77.90	61.54	64.10	84.11	81.37	**87.36**	96.00	83.80	79.52	91.69	+1.67%	79.18
MoLoRA	88.02	54.54	86.27	72.41	78.45	59.62	63.70	83.68	81.06	86.28	96.00	84.20	80.18	91.17	+1.87%	78.97
SiRA	87.71	54.67	86.33	**72.88**	78.14	60.58	64.90	84.39	81.32	87.00	97.00	83.00	80.02	91.48	+1.87%	79.24
HMoRA	87.66	53.62	86.36	72.72	79.16	**65.38**	65.20	83.94	80.24	86.28	**98.00**	84.40	80.50	91.14	+2.24%	79.61
T-REX	**88.31**	**55.83**	**86.51**	72.66	**80.21**	60.22	**67.40**	**84.61**	**81.62**	86.37	97.00	**84.80**	**81.41**	**92.35**	**+1.15%**	**79.95**
Gemma 2B																
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
T-REX	**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**
Tiny-LLaMA 1B																
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
T-REX	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**

and HMoRA (Liao et al., 2025) that combines MoE and LoRA with hierarchical hybrid routing and auxiliary loss, enhancing efficiency, task generalization, and parameter optimization.. The experts in SiRA, MoLoRA, and HMoRA are set as 8 × LoRA-4, aligning the learnable parameters of LoRA-32, while our T-REX adopts a total of 32 Rank-1 experts via the mix-and-match of rank-4 and rank-8 LoRA in the results.

5.3 Quantitative Results

Performance enhancement. We evaluate T-REX across 14 datasets on seven LLM backbones against four baselines without using additional prompt examples to ensure fair comparison. As shown in Table 1, T-REX achieves the highest average score (AVG) across all models, showcasing superior multitask capabilities. Specifically,

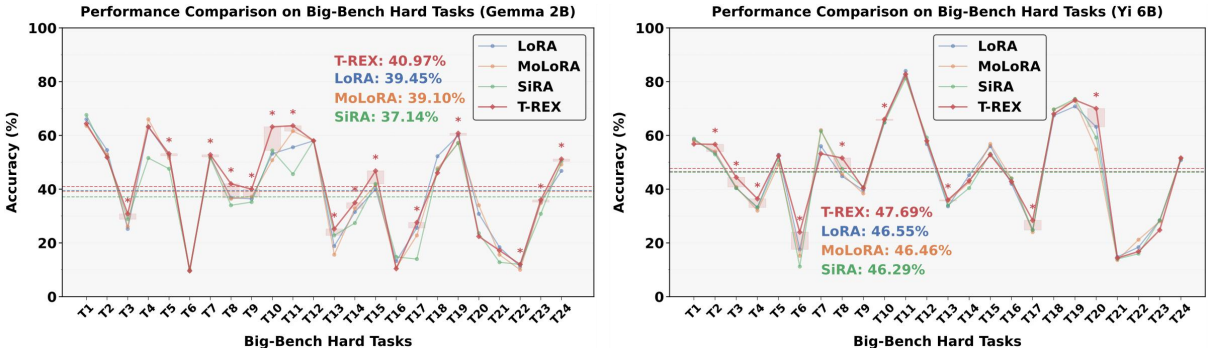

Figure 4: Out-of-distribution generalization capabilities of T-REX compared with baselines, including LoRA, MoLoRA, and SiRA, on the BBH dataset based on model backbone (a) Gemma 2B and (b) Yi 6B.

T-REX demonstrates significant performance improvements in challenging tasks such as WSC and ANLI. For instance, T-REX improves WSC performance by 7.54% on the Mistral model and achieves a substantial gain of 4.80% on LLaMA-2 7B, showcasing its ability to handle complex reasoning tasks. **On the WG, COPA, and HS tasks, T-REX outperforms almost all the baselines across multiple backbones**, demonstrating exceptional consistency and robustness, with minimal performance variance across different model. Notably, T-REX 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 relative to HMoRA, which requires more than 2% #ATP attributed 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, leading to poor average performance.** These results solidify T-REX as a robust and effective MTL solution, excelling in both performance and efficiency.

Efficiency analysis. We extend our analysis to evaluate the efficiency of T-REX. To this end, we benchmark T-REX 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 Table 2, demonstrate a significant reduction in computational overhead during training compared to LoRA, SiRA, MoLoRA, and especially for HMoRA. Specifically, T-REX only takes 39.9M additional parameters, an additional 7.86G FLOPs, and a training speed of 6.92it/s on a commercial-grade RTX 3090, while achieving a superior performance of 84.86 accuracy. In contrast, HMoRA fails to capitalize on its 130M additional trainable parameters, resulting in a slow throughput of 6.48it/s and yielding only 81.29 accuracy. In addition, although LoRA attains a similar throughput, 6.87 it/s for LoRA and 6.92 it/s for T-REX, T-REX shows a 1.33% performance gain, which clearly demonstrates the effectiveness of our approach. These findings highlight that **T-REX optimizes resource usage, delivering better performance with a smaller training footprint.**

Table 2: Additional computational overhead with different methods on GeForce RTX 3090.

Method	Rank	#ATP ↓	FLOPs ↓	Speed ↑	Acc. ↑
Mistral 7B					
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
T-REX	4+8	**39.9M**	**7.86G**	**6.92it/s**	**84.86**

Intuition generalizability. We investigate the generalization ability of T-REX, attributed to its intuition-guided design. To isolate the impact of intuition, **we exclude the Mix-and-Match from T-REX, 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 Table 3, T-REX consistently achieves improvements in both the OOD and ID scenarios. Notably, for the *Drop-A* setting, T-REX 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 3: OOD and ID data exploration of T-REX 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
T-REX	ID	**71.22**	**72.61**	**71.69**	**68.03**
LoRA	OOD	43.87	52.68	50.35	59.41
T-REX	OOD	**45.42**	**53.43**	**50.55**	**59.71**

In this study, we evaluated the OOD generalization capabilities of our proposed T-REX 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 results[3], presented in Fig. 4, demonstrate that T-REX 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, T-REX achieves accuracies of 40.97% and 47.59% on Gemma 2B and Yi 6B, consistently surpassing LoRA by over 1%. Detailed results for the figures and tables are shown in Appendix C and F.

[3]Zero-value points were excluded for clarity in visualization.

5.4 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 Table 4. We first compare a traditional rank-1 MoE-LoRA with 32 experts to T-REX 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 T-REX reaches 69.93 accuracy with more than 2× fewer #ATP. **These results provide initial evidence for the effectiveness of the MaM strategy.**

Table 4: Computational costs with different MaM strategies on GeForce RTX 3090.

Method	Experts	#ATP ↓	FLOPs ↓	Speed ↑	Acc. ↑
Tiny-LLaMA 1B					
Rank-1	32×32	37.85M	8.49G	9.13it/s	69.40
T-REX	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 attains a higher 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. Consequently, this leads to a smoother and more favorable convergence. **For fair computational comparisons, we use the 4×8 configuration as the default in all experiments.** Complete results are provided in Section D.

Different Embedders. By harnessing the multitask priors derived from the embedding model, the intuition of T-REX significantly enhances the MoE routing strategy, yielding a substantial improvement in multitask learning performance. In this context, we delve into the impact that various embedding models have on our approach. We contrast the effects of two distinct embedding models, Ada (Neelakantan et al., 2022) and Nomic-embed-text-v1 (Nussbaum et al., 2024) across three different LLM backbones, as shown in Table 5. We observe that introducing the implicit intuition with embedding model can greatly contribute to multitask learning. Nevertheless, the **choice of embedding model does not diminish the advantages offered by our proposed method** with the largest margin remains within 0.38% on Gemma 2B, thereby underscoring the robustness of our proposed T-REX.

Table 5: Consistent enhancement via the incorporation of diverse embedding models for intuition.

Method	Embedder	LLaMA-2	Gemma	T-LLaMA
T-REX	None	79.90	70.26	69.01
	Nomic	**81.11**	**75.40**	70.36
	Ada	81.19	75.02	**69.93**

Module functionality. To validate the superior performance of T-REX, we evaluated the effectiveness of implicit intuition and Rank-1 experts on various LLM backbones, including LLaMA-2 7B, Mistral 7B, Phi-2 2B, and Tiny-LLaMA 1B. As shown in Table 6, both components consistently enhance the performance of the LLMs. For instance, incorporating implicit intuition and Rank-1 experts into LLaMA-2 7B yields performance gains of up to 1.12% and 0.70%, respectively. These components also complement each other, working synergistically to further elevate T-REX's capabilities. Notably, their combination results in a significant improvement of 1.35% on the Tiny-LLaMA, underscoring the effectiveness of our approach. Moreover, **when integrating the MaM mechanism, T-REX achieves a performance boost of 1.02% while reducing parameter overhead by 40.6% for Mistral 7B.**

Table 6: Ablation study and efficiency improvement of T-REX with different modules.

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%

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%.** These results underscore the effectiveness and efficiency of T-REX. The complete results for each tasks are shown in Section E.

6 Conclusions

We propose T-REX, 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, T-REX enables a linear combination of low-rank matrices, expanding the vector subspace for improved performance and efficiency with the MaM mechanism. By extracting implicit intuition from semantic diversity via text embeddings, it guides all experts toward a generalizable convergence. Rigorous theoretical analysis shows that T-REX expands the model's subspace and steers optimization toward low-loss regions during fine-tuning. Comprehensive experiments further demonstrate that T-REX outperforms state-of-the-art baselines in both in-distribution and out-of-distribution settings, achieving superior performance and efficiency.

7 Ethics Statement

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets used were sourced in compliance with relevant usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

8 Reproducibility Statement

We have made every effort to ensure that the results presented in this paper are reproducible. All code and datasets have been made publicly available in an anonymous repository to facilitate replication and verification. The experimental setup, including training steps, model configurations, and hardware details, is described in detail in the paper. We have also provided a full description to assist others in reproducing our experiments.

Additionally, all used datasets are publicly available, ensuring consistent and reproducible evaluation results.

We believe these measures will enable other researchers to reproduce our work and further advance the field.

References

Winogrande: An adversarial winograd schema challenge at scale. In *Communications*, volume 64, pp. 99–106, 2021.

Yonatan Bisk, Rowan Zellers, et al. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

Rich Caruana. Multitask learning. *In ML*, 28:41–75, 1997.

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.

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, 2021.

Zihao Fu, Haoran Yang, et al. On the effectiveness of parameter-efficient fine-tuning. In *AAAI*, volume 37, pp. 12799–12807, 2023.

Gerd Gigerenzer. *Gut feelings: The intelligence of the unconscious*. 2007.

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.

Christian Harteis, Tina Koch, and Barbara Morgenthaler. How intuition contributes to high performance: An educational perspective. *Online Submission*, 5(1):68–80, 2008.

Dan Hendrycks, Collin Burns, et al. Measuring massive multitask language understanding. *In ICLR*, 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *In Neural Computing*, 9(8):1735–1780, 1997.

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.

Javaheripi, Mojan, Bubeck, et al. Phi-2: The surprising power of small language models. *In Microsoft Research Blog*, 2023.

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.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pp. 7482–7491, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *In NIPS*, 30, 2017.

Dmitry Lepikhin, HyoukJoong Lee, et al. Gshard: Scaling giant models with conditional computation and automatic sharding. In *ICLR*, 2021.

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.

Jiaming Liu, Rongyu Zhang, et al. Multi-latent space alignments for unsupervised domain adaptation in multi-view 3d object detection. *arXiv preprint arXiv:2211.17126*, 2022.

Yulin Luo, Rui Zhao, et al. Mowe: Mixture of weather experts for multiple adverse weather removal. *arXiv preprint arXiv:2303.13739*, 2023.

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.

Arvind Neelakantan, Tao Xu, et al. Text and code embeddings by contrastive pre-training. *ArXiv*, abs/2201.10005, 2022.

Yixin Nie and Adina others Williams. Adversarial nli: A new benchmark for natural language understanding. In *ACL*, 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.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*, 2023.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. abs/2106.04426, 2021.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *AAAI*, volume 33, pp. 4822–4829, 2019.

Maarten Sap, Hannah Rashkin, et al. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Noam Shazeer, Azalia Mirhoseini, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *CVPR*, pp. 5227–5237, 2022.

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.

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.

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*, 2023a.

Rongyu Zhang, Xiaowei Chi, et al. Unimodal training-multimodal prediction: Cross-modal federated learning with hierarchical aggregation. *arXiv preprint arXiv:2303.15486*, 2023b.

Rongyu Zhang, Lixuan Du, et al. Repcam: Re-parameterization content-aware modulation for neural video delivery. In *NOSSDAV*, pp. 1–7, 2023c.

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.

Yanqi Zhou, Tao Lei, et al. Mixture-of-experts with expert choice routing. In *NeurIPS*, 2022.

Yun Zhu, Nevan Wichers, et al. Sira: Sparse mixture of low rank adaptation. *arXiv preprint arXiv:2311.09179*, 2023.

Barret Zoph. Designing effective sparse expert models. In *IPDPS Workshops*, pp. 1044, 2022.

APPENDIX

This document provides supplementary materials for the main paper. In Section A, we present a detailed derivation of the approximation error bound stated in Theorem 4.2. Additionally, we elaborate on the motivation behind integrating the intuition into our T-REX framework in Section B. In Section C, we discuss how our approach consistently improves over LoRA baselines on both in-distribution and out-of-distribution tasks, even if some of the training tasks are dropped. Furthermore, Section D includes a comprehensive evaluation of different Mix-and-Match strategies. In Section E, we provide detailed results demonstrating the effectiveness of each component in our work, including the intuition and rank-1 experts. In Section F, we present the Big-Bench Hard (BBH) results, highlighting the generalization capabilities of the proposed T-REX framework.

A DETAILED DERIVATION OF APPROXIMATION ERROR BOUND

This section presents the detailed derivation for the approximation error bound stated in Theorem 1 of Section 4. Recall that we aim to bound the Frobenius norm of the difference between the T-REX adaptation matrix ΔW and the target adaptation matrix $\Delta W^\star$:

$$\|\Delta W - \Delta W^\star\|_F^2 \leq \|\Delta \mathbf{W}_{\text{out}}\|_F^2 + C \cdot \frac{1}{IJ}\|\Delta W^\star\|_F^2 + \mathcal{O}\left(\frac{1}{\tau}\right). \tag{19}$$

A.1 SUBSPACE DECOMPOSITION OF $\Delta W^\star$

The target adaptation matrix $\Delta W^\star \in \mathbb{R}^{m\times n}$ can be decomposed as:

$$\Delta W^\star = \Delta \mathbf{W}_{\text{in}} + \Delta \mathbf{W}_{\text{out}}, \tag{20}$$

where: - $\Delta \mathbf{W}_{\text{in}} \in \mathcal{S}_a \otimes \mathcal{S}_b$ is the component within the subspace spanned by the rank-1 components $\{\mathbf{a}_i \mathbf{b}_j^\top\}$, and - $\Delta \mathbf{W}_{\text{out}} \perp \mathcal{S}_a \otimes \mathcal{S}_b$ is the residual component orthogonal to the subspace.

This decomposition directly follows from the projection theorem in linear algebra, ensuring that $\Delta \mathbf{W}_{\text{in}}$ and $\Delta \mathbf{W}_{\text{out}}$ are orthogonal:

$$\langle \Delta \mathbf{W}_{\text{in}}, \Delta \mathbf{W}_{\text{out}} \rangle = 0. \tag{21}$$

The Frobenius norm of the difference $\|\Delta W - \Delta W^\star\|_F^2$ can now be split into two terms:

$$\|\Delta W - \Delta W^\star\|_F^2 = \|\Delta \mathbf{W}_{\text{in}} - \mathbf{A}\mathbf{G}\mathbf{B}^\top\|_F^2 + \|\Delta \mathbf{W}_{\text{out}}\|_F^2. \tag{22}$$

A.2 APPROXIMATION OF $\Delta \mathbf{W}_{\text{in}}$ BY T-REX

T-REX constructs the adaptation matrix ΔW as:

$$\Delta W = \mathbf{A}\mathbf{G}\mathbf{B}^\top = \sum_{i=1}^{N}\sum_{j=1}^{N} G_{ij}(\mathbf{x}) \mathbf{a}_i \mathbf{b}_j^\top, \tag{23}$$

where $\mathbf{G}(\mathbf{x}) \in \mathbb{R}^{I\times J}$ is the dynamic routing matrix, and $\mathbf{a}_i \in \mathbb{R}^m$, $\mathbf{b}_j \in \mathbb{R}^n$ are the learned basis vectors.

The term $\mathbf{A}\mathbf{G}\mathbf{B}^\top$ spans the subspace $\mathcal{S}_a \otimes \mathcal{S}_b$, which has dimensionality:

$$\dim(\mathcal{S}_a \otimes \mathcal{S}_b) = IJ. \tag{24}$$

Using the projection property, $\Delta \mathbf{W}_{\text{in}}$ can be approximated within $\mathcal{S}_a \otimes \mathcal{S}_b$ as:

$$\Delta \mathbf{W}_{\text{in}} = \mathbf{A}\mathbf{G}\mathbf{B}^\top + \Delta \mathbf{W}_{\text{res}}, \tag{25}$$

where $\Delta \mathbf{W}_{\text{res}} \in \mathcal{S}_a \otimes \mathcal{S}_b$ represents the residual error due to the limited expressiveness of the rank-1 combinations.

The Frobenius norm of the residual error $\Delta \mathbf{W}_{\text{res}}$ depends on the alignment between $\Delta W^\star$ and the subspace $\mathcal{S}_a \otimes \mathcal{S}_b$:

$$\|\Delta \mathbf{W}_{\text{res}}\|_F^2 \leq C \cdot \frac{1}{IJ}\|\Delta W^\star\|_F^2, \tag{26}$$

where C is a constant that depends on the expressiveness of the subspace.

This term decreases as IJ increases, reflecting the improved subspace coverage of the Mix-and-Match mechanism compared to traditional LoRA, which only spans an N-dimensional subspace with LoRA rank N.

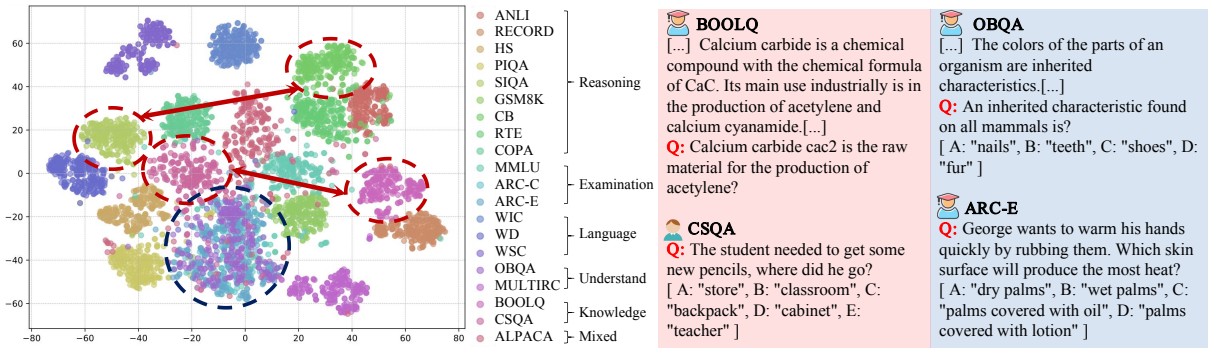

Figure 5: Embedding visualization of 20 datasets. Semantic clusters do not align with predefined task groupings.

A.3 Residual Error $\Delta\mathbf{W}_{\text{out}}$

The component $\Delta\mathbf{W}_{\text{out}} \perp \mathcal{S}_a \otimes \mathcal{S}_b$ lies orthogonal to the subspace and cannot be captured by the rank-1 combinations. Its contribution to the total error is:

$$\|\Delta\mathbf{W}_{\text{out}}\|_F^2 = \|\Delta W^\star - \Delta\mathbf{W}_{\text{in}}\|_F^2 = \|\Delta W^\star - \mathbf{A}\mathbf{G}\mathbf{B}^\top - \Delta\mathbf{W}_{\text{res}}\|_F^2. \tag{27}$$

Since $\Delta\mathbf{W}_{\text{out}}$ lies outside $\mathcal{S}_a \otimes \mathcal{S}_b$, its contribution to the total error cannot be reduced by increasing $I \times J$. However, in practice, $\Delta\mathbf{W}_{\text{out}}$ tends to be small if $\Delta W^\star$ aligns well with the learned subspace, making its contribution negligible.

A.4 Impact of Routing Matrix $\mathbf{G}(\mathbf{x})$

The routing matrix $\mathbf{G}(\mathbf{x})$ dynamically activates combinations of rank-1 components based on the input $\mathbf{x}$. The sharpness of these activations is controlled by the softmax temperature τ:

$$G_{ij}(\mathbf{x}) = \frac{\exp(z_{ij}/\tau)}{\sum_{i',j'} \exp(z_{i'j'}/\tau)}, \tag{28}$$

where z_{ij} are the logits for the rank-1 components. A higher temperature τ results in smoother activations across multiple rank-1 components, whereas a lower τ sharpens the focus on specific components. The influence of τ on the approximation error is captured by the term $\mathcal{O}(1/\tau)$, reflecting the trade-off between flexibility and alignment in subspace activation.

A.5 Final Error Bound

Combining the results from the previous sections, the total approximation error is:

$$\|\Delta W - \Delta W^\star\|_F^2 = \|\Delta\mathbf{W}_{\text{in}} - \mathbf{A}\mathbf{G}\mathbf{B}^\top\|_F^2 + \|\Delta\mathbf{W}_{\text{out}}\|_F^2 \tag{29}$$

$$\leq \|\Delta\mathbf{W}_{\text{res}}\|_F^2 + \|\Delta\mathbf{W}_{\text{out}}\|_F^2 \tag{30}$$

$$\leq \|\Delta\mathbf{W}_{\text{out}}\|_F^2 + C \cdot \frac{1}{IJ}\|\Delta W^\star\|_2^2 + \mathcal{O}\left(\frac{1}{\tau}\right). \tag{31}$$

Here: - The first term $\|\Delta\mathbf{W}_{\text{out}}\|_F^2$ quantifies the residual error outside the subspace $\mathcal{S}_a \otimes \mathcal{S}_b$. - The second term $\frac{1}{IJ}\|\Delta W^\star\|_2^2$ reflects the approximation error within the subspace, which decreases with increasing IJ. - The third term $\mathcal{O}(1/\tau)$ captures the effect of the routing matrix $\mathbf{G}(\mathbf{x})$ on dynamically activating rank-1 components.

This completes the derivation.

B Motivations

Research (Gigerenzer, 2007; Harteis et al., 2008) has shown that beyond *explicit awareness*, humans leverage *implicit intuition* to make swift decisions and instant reactions in critical situations. Yet, capturing this intuitive process within artificial intelligence is challenging due to its intangible essence. In this paper, we assert that an ideal AI intuition representation hinges on three core criteria: ① it must capture the nuanced semantic variability of each instance; ② it should be derivable via an autonomous process; and ③ it must seamlessly integrate into LLMs with low computational overhead.

Table 7: Tiny-LLaMA 1B performance under different training data settings. Gray colors refer to the five datasets excluded during training.

Models	Settings	MULTIRC	MMLU	BOOLQ	WIC	WG	WSC	ANLI	PIQA	SIQA	RTE	COPA	OBQA	CSQA	OOD	ID
LoRA	A	43.36	38.41	79.82	63.64	54.14	52.88	34.70	72.69	72.47	79.42	82.00	60.20	56.67	43.87	68.58
	B	81.15	38.34	80.12	53.45	53.28	63.46	51.00	67.08	72.72	81.59	86.00	47.60	56.92	52.68	72.43
	C	43.28	38.60	80.00	60.66	54.14	48.08	32.80	69.21	73.80	67.87	82.00	61.00	73.71	50.35	68.45
	D	82.51	41.80	80.80	64.42	58.80	53.85	52.70	66.38	58.50	84.48	85.00	67.40	58.72	59.41	67.15
T-REX	A	42.66	39.52	80.52	66.93	56.75	58.65	32.50	74.97	73.85	80.87	83.00	65.40	57.58	**45.42**	**71.22**
	B	82.63	38.08	81.35	51.57	54.22	57.69	52.20	68.34	73.39	83.03	85.00	48.40	60.77	**53.43**	**72.61**
	C	46.23	39.19	81.07	68.81	57.06	53.85	33.20	70.62	74.26	63.54	85.00	66.20	75.27	**50.55**	**71.69**
	D	83.56	43.31	82.14	69.28	59.59	48.08	54.50	65.18	59.26	84.84	87.00	70.40	58.89	**59.71**	**68.03**

Table 8: Performance evaluation across 14 datasets on Tiny-LLaMA 1B compared with three baselines. MUL. indicates MULTIRC and BOO. indicates BOOLQ.

Method	Experts	MUL.	MMLU	BOO.	WIC	WG	WSC	ANLI	PIQA	SIQA	RTE	COPA	OBQA	CSQA	HS	AVG ↑
Tiny-LLaMA 1B																
Rank-1	32	81.59	42.69	79.67	64.04	56.37	49.96	53.70	74.84	72.95	81.31	88.00	71.60	73.69	81.11	69.40
T-REX	8×8	**82.94**	**43.70**	**81.28**	**68.81**	**59.83**	**55.77**	**52.80**	**77.20**	**74.16**	82.67	83.00	**73.80**	72.15	**82.43**	**70.75**
	2×16	82.57	43.24	79.91	59.72	57.70	42.31	51.20	73.83	73.34	83.75	75.00	68.60	72.81	81.30	67.52
	4×8	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	69.93

In this paper, we integrate *implicit intuition* to improve LLM multitask performance. Initially, we explicitly informed the model with human-derived task categorizations, mapping datasets like BoolQ (Wang et al., 2019) to knowledge assessment and ANLI (Nie & Williams, 2020) to reasoning. However, this method underperformed compared to baseline fine-tuning. This led us to question the effectiveness of traditional task categorizations. By visualizing embeddings of 20 datasets across five categories, we found mismatches between expected and actual semantic clusters as in Fig. 5. Datasets like openbookqa (Mihaylov et al., 2018) and multirc (Wang et al., 2019) for comprehension, and piqa (Bisk et al., 2020) and siqa (Sap et al., 2019) for reasoning, did not cluster as anticipated. This discrepancy, particularly evident with datasets intended for comprehension and reasoning, highlights the shortcomings of relying on human-annotated task categorizations in guiding multitask learning. Thus, we aim to embed multitask knowledge into the MoE block implicitly via the proposed intuition score mechanism.

C Dropping out training data

We randomly drop out five datasets from the training set each time, denoting the dropped datasets as out-of-distribution (OOD) datasets and the remainder as in-distribution (ID). As shown in Table 7, we demonstrate our method consistently improves average accuracy for both ID and OOD tasks regardless of data configurations on Tiny-LLaMA 1B (Zhang et al., 2024a). We conducted four independent experiments with different configurations, labeled as A, B, C, and D:

- **Experiment A:** Datasets excluded HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), CommonSenseQA (Talmor et al., 2019), ANLI (Nie & Williams, 2020), and MultiRC (Wang et al., 2019), accounting for 31% of the total training samples.
- **Experiment B:** Datasets excluded WiC (Wang et al., 2019), OpenBookQA (Mihaylov et al., 2018), MMLU (Hendrycks et al., 2021), CommonSenseQA (Talmor et al., 2019), and PiQA (Bisk et al., 2020), accounting for 37% of the total training samples.
- **Experiment C:** Datasets excluded PiQA (Bisk et al., 2020), RTE (Wang et al., 2019), MMLU (Hendrycks et al., 2021), ANLI (Nie & Williams, 2020), and MultiRC (Wang et al., 2019), accounting for 44% of the total training samples.
- **Experiment D:** Datasets excluded OpenBookQA (Mihaylov et al., 2018), SiQA (Bisk et al., 2020), CommonSenseQA (Talmor et al., 2019), HellaSwag (Zellers et al., 2019), and PiQA (Bisk et al., 2020), accounting for 24% of the total training samples.

D Analysis of different Mix-and-Match strategies

We report comprehensive results for Tiny-LLaMA-1B using several Mix-and-Match (MaM) configurations on 14 benchmark datasets, as shown in Table 8. The 8×8 setting attains the highest overall accuracy, ranking first on 9 of

Table 9: Comparative analysis of the contributions of rank-1 and intuition across different models: LLaMA-2 7B, Phi-2 2.7B, and Tiny-LLaMA 1B. Rank-1, intuition, and the Mix-and-Match mechanism individually contribute to the final result.

Model	Intuition	Rank-1	MUL.	MMLU	BOO.	WIC	WG	WSC	ANLI	PIQA	SIQA	RTE	COPA	OBQA	CSQA	HS	AVG
Llama 2 7B	✗	✗	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	79.90
	✓	✗	88.57	54.21	88.62	74.45	83.66	60.58	72.10	84.49	81.37	88.81	96.00	84.40	82.88	94.18	81.02
	✗	✓	89.21	55.32	88.29	73.04	81.69	57.69	71.70	83.68	81.37	88.09	96.00	84.80	83.05	94.41	80.60
	✓	✓	89.13	54.54	88.44	73.51	82.64	58.65	72.80	85.69	81.99	89.17	97.00	84.20	83.29	94.45	**81.11**
	+ Mix-and-Match		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**	**81.19**
Phi-2 2.7B	✗	✗	88.00	54.80	86.36	71.94	77.90	61.54	64.10	84.11	81.37	87.36	96.00	83.80	79.52	91.69	79.18
	✓	✗	88.20	55.06	87.09	73.04	79.87	57.69	67.80	83.90	80.81	88.09	97.00	86.40	80.51	92.29	79.84
	✗	✓	88.37	56.17	87.22	72.10	80.11	54.81	66.40	84.49	81.68	86.64	95.00	85.20	80.92	92.57	79.41
	✓	✓	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**	**81.19**
	+ Mix-and-Match		**88.31**	**55.83**	**86.51**	72.66	**80.21**	60.22	**67.43**	**84.61**	**81.62**	86.37	**97.00**	**84.80**	**81.41**	**92.35**	79.95
Tiny-LLaMA 1B	✗	✗	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	69.01
	✓	✗	83.19	43.37	81.47	61.44	59.75	50.96	52.40	75.63	75.08	85.20	82.00	71.40	73.79	83.26	69.92
	✗	✓	83.27	43.63	80.43	63.79	59.35	52.88	51.30	75.63	74.10	83.39	81.00	71.00	73.05	82.97	69.70
	✓	✓	83.25	42.72	80.95	64.42	60.85	49.04	52.40	75.52	74.51	85.56	88.00	71.80	73.05	82.92	**70.36**
	+ Mix-and-Match		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	69.93

the 14 tasks. Nevertheless, the 4×8 variant clearly outperforms all other settings on RTE, COPA, and CSQA, where it leads by sizable margins. Although 8×8 is marginally stronger on average, it incurs about 50% more parameter overhead and correspondingly higher compute cost than 4×8. To achieve a better accuracy-versus-efficiency trade-off, we therefore adopt the 4×8 configuration in all subsequent experiments.

E Synergistic Effects of Intuition and Rank-1 Experts

As elaborated in the main paper, both implicit intuition and rank-1 experts elevate the LLM's performance on average accuracy across 14 datasets. We provide more accurate details on each subset in this section, shown in Table 9. For our study, we select three LLMs of varying model sizes, ranging from 1 billion to 7 billion parameters. Our results indicate that implicit intuition and rank-1 experts individually boost model performance. Specifically, Intuition is tailored for expert routing, optimizing the model's decision-making pathways, while Rank-1 Experts facilitate more efficient parameter utilization across the model's architecture. The synergistic combination of these approaches consistently delivers optimal results.

F Big-Bench Hard Results

As elaborated in the main paper, our proposed T-REX demonstrates remarkable out-of-distribution (OOD) generalization capabilities, surpassing other methods, including LoRA (Hu et al., 2021), MoLoRA (Zadouri et al., 2023), and SiRA (Zhu et al., 2023), on the Big-Bench Hard (BBH) dataset. Detailed comparative results are presented in Tables 10 to 12. Our evaluation, conducted on a diverse set of models including Yi 6B (Young et al., 2024), Gemma 2B (Team et al., 2024), Mistral 7B (Jiang et al., 2023), Llama 2 7B (Touvron et al., 2023), Tiny-LLaMA 1B (Zhang et al., 2024a), and Phi-2 2B (Javaheripi et al., 2023), is performed without finetuning on BBH samples. We observe superior OOD performance with our T-REX models, which achieves 1.52% enhancement over competing methods.

G LLM Usage

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

It is important to note that the LLM was not involved in the ideation, research methodology, or experimental design. All research concepts, ideas, and analyses were developed and conducted by the authors. The contributions of the LLM were solely focused on improving the linguistic quality of the paper, with no involvement in the scientific content or data analysis.

The authors take full responsibility for the content of the manuscript, including any text generated or polished by the LLM. We have ensured that the LLM-generated text adheres to ethical guidelines and does not contribute to plagiarism or scientific misconduct.

Table 10: Zero-shot generalization on the Big-Bench Hard (BBH) benchmark. Using diverse models, we evaluate the generalization capabilities of various methods, including LoRA, MoLoRA, SiRA, and our T-REX. This table demonstrates the results of Yi 6B and Gemma 2B.

Big-Bench Hard	Yi 6B				Gemma 2B			
	LoRA	MoLoRA	SiRA	T-REX	LoRA	MoLoRA	SiRA	T-REX
boolean_expressions	58.40	58.00	58.80	56.80	66.00	63.60	67.60	64.40
causal_judgement	53.48	54.01	52.94	56.68	54.55	52.94	51.87	51.87
date_understanding	40.40	40.80	40.40	44.40	25.20	26.00	28.80	30.80
disambiguation_qa	33.20	32.00	33.20	36.40	63.20	66.00	51.60	63.20
dyck_languages	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
formal_fallacies	52.80	49.20	50.80	52.40	52.40	51.60	47.60	53.20
geometric_shapes	17.60	15.20	11.20	24.00	9.60	10.00	10.00	9.60
hyperbaton	56.00	62.00	61.60	53.20	52.00	52.00	51.60	52.80
logical_deduction_five_objects	44.80	45.60	47.60	51.60	36.80	36.40	34.00	42.00
logical_deduction_seven_objects	39.60	38.40	40.80	40.40	36.40	37.20	35.20	40.00
logical_deduction_three_objects	64.80	65.60	64.80	66.00	53.20	50.80	54.40	63.20
movie_recommendation	84.00	82.00	81.20	82.80	55.60	61.60	45.60	63.60
multistep_arithmetic_two	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
navigate	56.80	57.60	59.20	58.00	58.00	58.00	58.00	58.00
object_counting	33.60	35.60	34.00	36.00	18.80	15.60	22.80	25.20
penguins_in_a_table	45.21	42.47	40.41	43.15	31.51	32.88	27.40	34.93
reasoning_about_colored_objects	56.00	56.80	53.20	52.80	40.00	41.60	42.00	46.80
ruin_names	42.00	44.00	44.00	42.80	13.20	10.80	14.80	10.40
salient_translation_error_detection	24.80	24.00	24.80	28.40	25.60	22.80	14.00	27.60
snarks	67.42	69.66	69.66	67.98	52.25	47.75	47.19	46.07
sports_understanding	70.80	72.80	73.60	73.20	60.00	57.20	57.20	60.80
temporal_sequences	63.20	54.80	59.20	70.00	30.80	34.00	23.60	22.40
tracking_shuffled_objects_five_objects	14.80	13.60	14.00	14.40	18.40	15.60	12.80	17.20
tracking_shuffled_objects_seven_objects	18.40	21.20	16.00	16.80	11.20	10.00	12.00	12.00
tracking_shuffled_objects_three_objects	28.40	28.00	28.40	24.80	35.20	34.80	30.80	36.00
web_of_lies	50.80	51.60	51.20	51.60	46.80	49.20	50.40	51.20
word_sorting	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.40
AVG	41.38	41.29	41.15	**42.39**	35.06	34.75	33.01	**36.58**

Table 11: Zero-shot generalization on the Big-Bench Hard (BBH) benchmark. Using diverse models, we evaluate the generalization capabilities of various methods, including LoRA, MoLoRA, SiRA, and our T-REX. This table demonstrates the results of Mistral 7B and Llama-2 7B.

Big-Bench Hard	Mistral 7B				Llama-2 7B			
	LoRA	MoLoRA	SiRA	T-REX	LoRA	MoLoRA	SiRA	T-REX
boolean_expressions	62.00	65.20	58.40	63.60	58.00	58.00	56.00	56.80
causal_judgement	56.15	56.68	58.29	56.15	57.75	56.68	56.68	57.22
date_understanding	54.00	55.60	51.60	50.80	42.80	41.60	44.80	40.80
disambiguation_qa	36.40	52.00	44.80	47.60	47.60	42.80	34.40	53.20
dyck_languages	3.20	1.20	2.00	0.80	0.00	0.00	0.00	0.00
formal_fallacies	54.80	53.60	55.20	55.20	57.20	58.80	58.40	58.40
geometric_shapes	11.20	22.00	13.20	14.00	34.00	37.20	38.80	38.00
hyperbaton	79.20	69.20	66.80	74.80	57.60	66.80	62.00	60.40
logical_deduction_five_objects	56.40	58.40	58.40	62.80	54.80	52.80	52.80	52.80
logical_deduction_seven_objects	50.40	49.20	46.80	52.40	47.60	48.00	47.20	46.80
logical_deduction_three_objects	86.40	84.40	82.00	86.40	70.80	72.80	75.20	73.20
movie_recommendation	60.80	60.40	59.60	70.40	56.80	60.40	56.40	60.00
multistep_arithmetic_two	0.00	0.40	0.00	0.00	0.00	0.00	0.40	0.00
navigate	57.20	60.00	47.20	58.00	46.80	42.80	57.60	42.80
object_counting	38.80	33.60	40.40	34.00	30.00	30.80	28.80	34.40
penguins_in_a_table	54.11	54.11	50.00	54.11	41.10	40.41	43.15	41.78
reasoning_about_colored_objects	59.60	59.20	64.00	61.60	55.60	46.00	52.00	54.40
ruin_names	40.00	40.40	41.60	38.80	52.00	42.80	54.80	44.40
salient_translation_error_detection	48.80	46.80	45.20	49.20	39.60	38.00	33.20	36.00
snarks	67.42	67.98	67.42	67.98	52.25	52.25	59.55	53.93
sports_understanding	66.40	66.00	62.80	66.40	60.00	58.80	58.00	60.00
temporal_sequences	43.20	50.00	50.80	55.60	24.00	41.20	34.80	38.00
tracking_shuffled_objects_five_objects	22.00	23.20	22.00	23.60	15.60	19.20	17.20	22.00
tracking_shuffled_objects_seven_objects	21.20	17.60	19.60	22.00	16.00	12.80	13.60	16.00
tracking_shuffled_objects_three_objects	21.60	22.40	22.40	23.60	29.20	24.40	24.00	24.00
web_of_lies	46.00	48.80	50.40	47.60	49.60	50.00	49.20	50.40
word_sorting	14.80	14.40	18.40	16.80	15.20	14.40	15.60	18.80
AVG	44.89	45.66	44.42	**46.45**	41.18	41.10	41.65	**42.02**

Table 12: Zero-shot generalization on the Big-Bench Hard (BBH) benchmark. Using diverse models, we evaluate the generalization capabilities of various methods, including LoRA, MoLoRA, SiRA, and our T-REX. This table demonstrates the results of Tiny-LLaMA and Phi-2.

Big-Bench Hard	TinyLlama 1B				Phi-2			
	LoRA	**MoLoRA**	**SiRA**	**T-REX**	**LoRA**	**MoLoRA**	**SiRA**	**T-REX**
boolean_expressions	56.80	58.00	58.40	55.20	77.20	78.40	79.20	79.60
causal_judgement	51.87	50.27	51.34	53.48	58.82	56.68	56.15	57.75
date_understanding	26.00	38.40	41.20	30.40	39.20	39.60	38.00	40.40
disambiguation_qa	30.00	30.00	31.20	36.80	62.80	64.80	64.80	63.60
dyck_languages	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
formal_fallacies	54.00	53.20	52.40	52.80	55.60	56.00	56.00	55.20
geometric_shapes	0.00	0.00	9.60	0.00	23.60	27.60	29.20	31.60
hyperbaton	51.60	51.60	51.60	51.60	69.60	72.00	66.40	68.00
logical_deduction_five_objects	26.80	27.20	21.20	28.00	48.00	50.40	54.80	54.00
logical_deduction_seven_objects	19.20	17.20	18.40	18.80	56.80	56.40	53.20	51.60
logical_deduction_three_objects	45.20	46.80	40.00	47.20	76.00	73.60	73.60	78.80
movie_recommendation	55.20	54.80	56.80	53.20	51.20	52.80	54.80	52.40
multistep_arithmetic_two	0.40	0.40	0.40	0.00	0.40	0.80	1.20	0.80
navigate	54.80	43.20	42.00	56.80	46.00	43.20	46.00	58.00
object_counting	5.60	7.20	5.60	6.40	38.80	38.00	38.40	40.80
penguins_in_a_table	32.19	30.14	28.77	32.88	48.63	50.00	45.89	48.63
reasoning_about_colored_objects	32.40	28.80	29.20	30.00	58.00	55.20	57.20	56.40
ruin_names	6.00	6.40	5.20	7.60	53.60	59.20	61.20	51.60
salient_translation_error_detection	10.40	11.60	11.20	19.20	43.60	40.80	43.60	44.40
snarks	49.44	43.26	48.88	51.12	66.85	68.54	73.60	74.16
sports_understanding	64.40	48.80	46.00	63.60	56.40	58.40	56.40	55.20
temporal_sequences	9.20	8.80	10.40	11.60	79.60	77.20	77.20	62.40
tracking_shuffled_objects_five_objects	18.00	17.20	18.40	17.60	18.40	18.80	18.00	20.80
tracking_shuffled_objects_seven_objects	14.00	12.00	13.60	12.80	9.60	12.00	11.60	13.20
tracking_shuffled_objects_three_objects	33.60	33.20	35.20	34.00	32.00	31.60	30.40	31.60
web_of_lies	48.80	51.20	51.60	50.00	52.40	53.20	51.60	54.00
word_sorting	2.00	1.20	1.60	3.20	20.00	18.80	17.60	18.00
AVG	29.55	28.55	28.90	**30.53**	46.04	46.45	46.52	**46.78**