RESEARCH-ARTICLE

ResMoE: Space-efficient Compression of Mixture of Experts LLMs via Residual Restoration

MENGTING AI, University of Illinois Urbana-Champaign, Urbana, IL, United States

TIANXIN WEI, University of Illinois Urbana-Champaign, Urbana, IL, United States

YIFAN CHEN, Hong Kong Baptist University, Hong Kong, Hong Kong

ZHICHEN ZENG, University of Illinois Urbana-Champaign, Urbana, IL, United States

RITCHIE ZHAO, NVIDIA, Santa Clara, CA, United States

GIRISH VARATKAR

View all

Published: 20 July 2025

KDD '25: The 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining
August 3 - 7, 2025
Toronto ON, Canada

Conference Sponsors:
SIGMOD
SIGKDD

ResMoE: Space-efficient Compression of Mixture of Experts LLMs via Residual Restoration

Mengting Ai*
UIUC
Champaign, IL, USA
mai10@illinois.edu

Tianxin Wei*
UIUC
Champaign, IL, USA
twei10@illinois.edu

Yifan Chen*†
HKBU
Kowloon, HK
yifanc@hkbu.edu.hk

Zhichen Zeng
UIUC
Champaign, IL, USA
zhichenz@illinois.edu

Ritchie Zhao
NVIDIA
Seattle, CA, USA
rz252@cornell.edu

Girish Varatkar
Apple
Cupertino, CA, USA
girish_v_varatkar@apple.com

Bita Darvish Rouhani
NVIDIA
Seattle, CA, USA
brouhani@nvidia.com

Xianfeng Tang
Amazon
Palo Alto, CA, USA
xianft@amazon.com

Hanghang Tong
UIUC
Champaign, IL, USA
htong@illinois.edu

Jingrui He†
UIUC
Champaign, IL, USA
jingrui@illinois.edu

Abstract

Mixture-of-Experts (MoE) Transformer, the backbone architecture of multiple phenomenal language models, leverages sparsity by activating only a fraction of model parameters for each input token. The sparse structure, while allowing constant time costs, results in space inefficiency: we still need to load all the model parameters during inference. We introduce ResMoE, an innovative MoE approximation framework that utilizes Wasserstein barycenter to extract a common expert (barycenter expert) and approximate the residuals between this barycenter expert and the original ones. ResMoE enhances the space efficiency for inference of large-scale MoE Transformers in a one-shot and data-agnostic manner without retraining while maintaining minimal accuracy loss, thereby paving the way for broader accessibility to large language models. We demonstrate the effectiveness of ResMoE through extensive experiments on Switch Transformer, Mixtral, and DeepSeekMoE models. The results show that ResMoE can reduce the number of parameters in an expert by up to 75% while maintaining comparable performance.[1]

CCS Concepts

• **Computing methodologies** → **Machine learning**; **Natural language processing**.

*Mengting, Tianxin, and Yifan contributed equally to this work.
†Correspondence to: Yifan Chen and Jingrui He.
[1]The code is available at https://github.com/iDEA-iSAIL-Lab-UIUC/ResMoE, and the supplementary appendix is available at https://famous-blue-raincoat.github.io/mengtingai/files/ResMoE_Appendix.pdf.

KDD '25, August 3–7, 2025, Toronto, ON, Canada

ACM ISBN 979-8-4007-1245-6/25/08
https://doi.org/10.1145/3690624.3709196

Keywords

Mixture-of-Experts, Compression, Optimal Transport, Wasserstein Barycenter

ACM Reference Format:
Mengting Ai, Tianxin Wei, Yifan Chen, Zhichen Zeng, Ritchie Zhao, Girish Varatkar, Bita Darvish Rouhani, Xianfeng Tang, Hanghang Tong, and Jingrui He. 2025. ResMoE: Space-efficient Compression of Mixture of Experts LLMs via Residual Restoration. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3690624.3709196

1 Introduction

The profound impact of the Transformer architecture in the domain of machine learning [67] is undeniable, for the fields including natural language processing [3, 6, 9, 15, 19, 29, 49, 54–56, 67, 86, 88, 89], computer vision [18, 43, 70, 72, 75], information retrieval [8, 32, 74, 81, 83], and graph modeling [39, 41, 52, 77, 79]. To further improve the capabilities of pre-trained large language models (LLMs), one general strategy is to scale up their parameters. Mixture-of-Experts (MoE) [59] extends the traditional feedforward neural network (FFN) layer by replacing a single multilayer perceptron (MLP) with multiple MLPs, referred to as "experts". While enhancing the performance, sparse MoE keeps computing costs (FLOPs) comparable to the original dense model, as only a few selected experts will be activated each time. The framework of an MoE layer is demonstrated in Fig. 1. Specifically, the input token x is passed to the router gate network, returning the sparse and normalized top-k scores used to activate the following experts. Only experts with a score larger than 0 will be activated, and the continued results will then be calculated through those activated expert MLPs. The output y will then be obtained through a weighted sum of each activated expert's output y_k. Switch Transformer [19] exemplifies this approach by expanding the T5 model [56] to an MoE structure, scaling it up to at most 2,048 times the size of the original dense T5 model. Similarly, Mixtral [31] upscales Mistral 7B [30] to an 8×7B MoE structure, achieving performance that matches

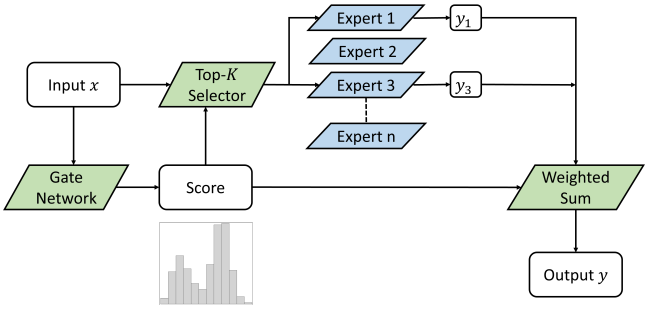

Figure 1: In this illustrative example of MoE layers, the Top-K Selector, along with the Gate Network–often referred to as the 'router'–selects Experts 1 and 3 based on their scores for the given input. Figure taken from [1].

or even surpasses that of Llama2 70B [66]. DeepSeekMoE [12] utilizes fine-grained experts compared to the other structures, with 64 experts per layer.

However, the enormous number of parameters has now become a bottleneck for MoE Transformers [34], since they require much more GPU memory to load the model even if only part of the parameters are activated each time. The expert size for Mixtral reaches 176.2M, and the presence of 8 or even more experts in each layer exacerbates the memory demands, bringing a strong need to compress the experts in the MoE structure. To give an example, the total model size of Mixtral is 87.0 GB, while the corresponding size of the dense model Mistral is only 13.5 GB.

To leverage the capabilities of MoE LLMs, we revisit several (seemingly unrelated while inherently connected) research avenues below. One approach is model fusion [2, 60], which involves combining multiple general MLPs. This technique can be adapted to merge experts in MoE models as well. More recently, various studies have introduced the concept of expert merging [28, 37, 42, 62, 78] and expert pruning [45], as a method to reduce the number of experts within each layer of the MoE model. Nevertheless, we note the direct reduction in the total number of experts potentially leads to a substantial loss of the specialized knowledge that individual experts possess (see an illustrative analysis in Section 4.1).

To address the aforementioned issues, we introduce ResMoE, an MoE approximation framework. Our approach capitalizes on approximating the MoE models with fewer parameters by utilizing Wasserstein barycenter techniques [51]. We formulate a distributional representation of experts and extract their common characteristics to obtain the barycenter expert. Subsequently, we propose to employ either unstructured pruning [35] or singular value decomposition (SVD) [13] (as a pilot example) to approximate the residual matrices between this barycenter expert and each specific expert. In summary, the contribution of our work is three-fold:

- We introduce Wasserstein barycenter and residual restoration into MoE approximation, aiming to maintain the common and distinctive attributes of each expert with fewer parameters.
- We propose ResMoE, a practical MoE Transformer approximation framework that aims to improve space efficiency in a one-shot and data-agnostic manner, with no extra training required.
- We validate ResMoE through extensive experiments on both the encoder-decoder Switch Transformer model, as well as the decoder-only models, Mixtral and DeepSeekMoE. Our results demonstrate that ResMoE can reduce the number of parameters in an expert by up to 75% while incurring only marginal performance loss, verifying its effectiveness and versatility.

2 Related Work

General model compression techniques. The focus of deep learning compression research [35, 53, 63] primarily involves system-level optimization. *Quantization* aims at hardware efficiency by reducing model weight bit-depth from 32-bit floating point (FP32) to 8-bit integers (INT8) [4, 14, 85] or even lower bits [7, 22, 40, 65]. While such optimization techniques have seen growing adoption across various machine learning applications [87], our focus is on reducing the parameter count of the MoE model, making quantization methods not directly related.

Additionally, *knowledge distillation* [24, 27, 33] aims to transfer knowledge from pre-trained LLMs to smaller models. However, this approach requires extensive retraining, involving both the original LLM and the compact model. *Truncated singular value decomposition* (SVD) [13] has been used to streamline CNNs by reducing redundancy through linear structure exploitation within networks, yet it faces limits in representational capacity, often leading to decreased performance due to overly aggressive dimension reduction. *Pruning techniques* [36, 44], evolving with the Lottery Tickets Hypothesis [20, LTH], seek efficient sub-networks within larger models but require extensive retraining to maintain accuracy. While some one-shot pruning methods [63, 69] do exist, they remain computationally expensive and are not specifically tailored for the structure of MoE, bringing concerns about whether such methods can adequately ensure that the compressed models retain their effectiveness for downstream tasks.

Mixture-of-Expert (MoE) transformer compression. Rather than applying existing compression techniques individually to the expert MLP, MC-SMoE [37], MEO [28], and OneS [78] merge the experts into smaller groups, reducing the count of the experts. Expert pruning [45, 48] follows a similar aspect, pruning the less important experts to reduce the size. This approach faces challenges in deciding the experts to retain, potentially leading to loss of information due to sub-optimal decisions. Gao et al. [23] instead proposed to keep each expert, divide them into several sections, and share the core section among them. This method does not align with our goal since they aimed to efficiently train a new MoE-like structure from scratch, instead of compressing an existing one. Alternatively, we note fusion-based methods [2, 60], originally proposed for consolidating distinct models into a single one, can be dynamically adapted for consolidating MoE's experts. These methods utilize the principles of permutation and optimal transport and are implemented layer-wise, which requires applying the permutations derived from preceding layers to the next one. The characteristic incurs overhead due to the extra time required for permutations.

3 Preliminaries and Notation

This section provides the background of MoE, optimal transport, and Wasserstein barycenter.

3.1 Mixture-of-Experts Modules

Throughout this paper, we consider the classical setting of MoE modules for the ease of analysis, where each expert takes the form of a multilayer perceptron (MLP) in a feed-forward network (FFN) sub-layer of a Transformer. It is worth noting that there exist different types of expert network architectures (c.f. Appendix B.3.)

Each Mixture-of-Experts (MoE) layer comprises N experts. The k-th expert E_k (a function to transform input vector $\mathbf{x}$ to a new feature) in an FFN sub-layer is denoted as:

$$E_k(\mathbf{x}) = \mathbf{W}_k^{(2)} \sigma\left(\mathbf{W}_k^{(1)}\mathbf{x} + \mathbf{b}_k^{(1)}\right) + \mathbf{b}_k^{(2)},$$

where $\sigma(\cdot)$ is the element-wise activation function. The input $\mathbf{x} \in \mathbb{R}^p$, and $(\mathbf{W}_k^{(1)}, \mathbf{b}_k^{(1)}) \in \mathbb{R}^{p_\mathrm{I} \times (p+1)}$, $(\mathbf{W}_k^{(2)}, \mathbf{b}_k^{(2)}) \in \mathbb{R}^{p \times (p_\mathrm{I}+1)}$ are respectively the weight matrices and bias vectors in the linear transforms of the MLP (with input/output dimension p and inner dimension p_I). The output of the MoE layer is given by: $\sum_{k=1}^N [G(\mathbf{x})]_k \cdot E_k(\mathbf{x})$. Here $G(\mathbf{x}) = \mathrm{Softmax}\left(\mathrm{TopK}\left(\mathbf{W}_g \mathbf{x}\right)\right)$ returns the normalized sparse router gating vector for all experts, where $\mathrm{TopK}(\mathbf{g}_i) = \mathbf{g}_i$ when $\mathbf{g}_i$ is within the top-k values of $\mathbf{g} \in \mathbb{R}^N$, otherwise $\mathrm{TopK}(\mathbf{g}_i) = -\infty$; $\mathbf{W}_g \in \mathbb{R}^{N \times p}$ represents the linear transform, turning the input $\mathbf{x}$ into the logit for each expert. The whole framework of the MoE layer is shown in Fig. 1.

The space bottleneck [34] comes from the large size of experts (ranging from 8 to 64 and even more [12, 19]) and the tremendous size of the weight matrices in each expert (e.g., 176.2M parameters for each expert in Mixtral [31]). The sparse design renders the total number of parameters redundant compared to the base dense model. Even though only a part of the parameters is activated each time, the whole model still needs to be loaded in the RAM. In this paper, we aim to address the redundancy problem while retaining the effectiveness of pre-trained MoE models.

3.2 Optimal Transport and Wasserstein Barycenter

Optimal transport (OT) theory has achieved great success in depicting the underlying geometry of distributions [11, 51, 82, 84]. We consider two distributions $\mu = \sum_{i=1}^n \boldsymbol{\alpha}_i \delta_{x_i}$ and $\nu = \sum_{j=1}^m \boldsymbol{\beta}_j \delta_{y_j}$ with $\boldsymbol{\alpha}_i, \boldsymbol{\beta}_j$ as masses respectively assigned to points x_i, y_j, and δ_x being the Dirac unit mass located on x. (In this paper, μ, ν will always be the discrete distributions.) OT reflects a process of transporting the mass from positions x_i's to y_j's (transforming the source distribution μ to the target distribution ν) with the minimal overall cost, in which the cost of transporting a unit mass from x_i to y_j is given by the cost function $D(x_i, y_j)$.

A *transport plan* can be specified by a matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$, where $\mathbf{M}_{i,j}$ indicates the mass to be transported from x_i to y_j. We note that the column and row sums of $\mathbf{M}$ respectively equal to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, implying all the masses in μ are transported to the desired points in ν, i.e., $\mathbf{M} \in \Pi(\boldsymbol{\alpha}, \boldsymbol{\beta}) := \{\mathbf{M} \in \mathbb{R}^{n \times m} | \sum_{j=1}^m \mathbf{M}_{i,j} = \boldsymbol{\alpha}_i, \sum_{i=1}^n \mathbf{M}_{i,j} = \boldsymbol{\beta}_j\}$. OT seeks the optimal plan to transport μ to ν w.r.t the overall transportation cost, formulated as [51]:

$$\mathrm{OT}(\mu, \nu) := \underset{\mathbf{M} \in \Pi(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\arg\min} \sum_{i,j} \mathbf{M}_{i,j} \mathbf{C}_{i,j} = \underset{\mathbf{M} \in \Pi(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\arg\min} \langle \mathbf{M}, \mathbf{C} \rangle,$$

where $\mathbf{C} := \left[D(x_i, y_j)\right]_{ij} \in \mathbb{R}^{n \times m}$ is the cost matrix.

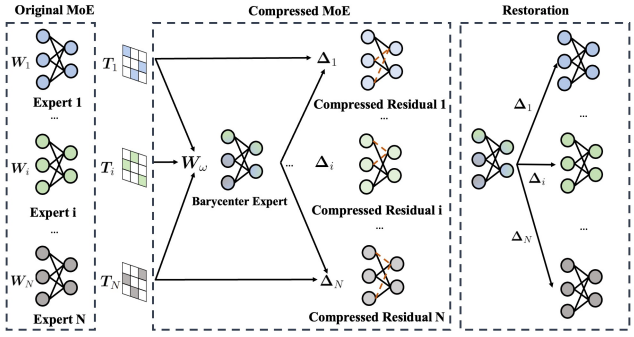

Figure 2: The overall framework of ResMoE. We introduce permutation matrices T to obtain the barycenter expert $\mathbf{W}_\omega$ from a distributional view. Instead of compressing the original experts directly, we opt to compress the residual matrices (Δ, illustrated with lighter colors) between each expert and the barycenter expert, with the capability to dynamically and efficiently restore the original matrices during inference. We illustrate the concept using unstructured pruning as an example, with dashed orange lines indicating the pruned connections within the network.

Setting the cost function as $D(x_i, y_j) = \|x_i - y_j\|^2$, we can obtain 2-Wasserstein distance [51] as:

$$W_2^2(\mu, \nu) := \min_{\mathbf{M} \in \Pi(\boldsymbol{\alpha}, \boldsymbol{\beta})} \langle \mathbf{M}, \mathbf{C} \rangle .$$

In this paper, we specifically focus on the free-support Wasserstein barycenter problem induced by 2-Wasserstein distance. Given a set of distributions $\mu_1, \ldots, \mu_N$, the Wasserstein barycenter $\bar{\mu}$ is the "average" distribution in terms of the Wasserstein distance. To regulate the form of $\bar{\mu}$ in numerical computation, we specify $\bar{\mu}$ as a uniform distribution on n points, and optimize it through:

$$\bar{\mu} = \underset{\{x_i\}_{i=1}^n}{\arg\min} \frac{1}{N} \sum_{k=1}^N W_2^2\left(\mu_k, \sum_{i=1}^n \frac{1}{n} \delta_{x_i}\right). \tag{1}$$

We comment Cuturi and Doucet [11] have provided efficient numerical algorithms for the free-support Wasserstein barycenter problem above, which will be heavily utilized in our implementations. In a nutshell, we conclude Wasserstein barycenter captures the underlying advection in the distribution space, offering a powerful tool for aggregating distributions in a geometrically meaningful way.

4 Proposed Methodology

In this section, we first analyze the limitations of existing fusion strategies, and then give a detailed introduction to ResMoE, along with its visualization provided in Figure 2 and pseudocode in Appendix A.9. For a comprehensive visual comparison between ResMoE and previous baseline methods, please refer to Figure 3.

4.1 Limitations of Existing Fusion Strategies

In advance of our proposal in Section 4.2, we first review the limitations of existing fusion/merge strategies, which mainly serves as the motivation for developing the ResMoE framework.

Alignment-based model fusion [2, 60] is proposed to fuse multiple models, that can be adapted in the MoE structure to fuse MLPs (which may have more than two layers) and relates to OT. In OT Fusion [60], for a two-layer MLP, their algorithm starts from aligning the first layer of each expert and then pre-aligns the second layer with the permutation matrix obtained from the first layer, repeating the procedure to align the second layer further. Similarly, in Git Re-Basin [2], they propose to align the weights through a greedy loop for each layer, which demonstrates zero-barrier linear mode connectivity [21] between independently trained models on the same dataset. We note the alignment-based model fusion technique implies a layer-by-layer strategy to formulate MoE layers as distributions and to merge experts. Moreover, expert merging [28, 37, 78] has been recently introduced to reduce the number of experts in MoE modules. This approach merges experts using task-specific information, such as router gating score distribution or router activation frequency.

Both the model fusion and expert merging reduce the number of experts to compress the model, while we remark that the direct reduction in the number of experts may lead to a huge deviation from the original module output, especially in the zero-shot setting.

To better understand this, we first revisit the MoE layer from an all-experts matrix perspective. We define the router matrix $\mathbf{R}$ as:

$$\mathbf{R} \coloneqq \operatorname{diag}(G(\mathbf{x})) \otimes \mathbf{I}_{p_{\mathrm{I}}} = \begin{bmatrix} [G(\mathbf{x})]_1\mathbf{I} & & \\ & \dots & \\ & & [G(\mathbf{x})]_N\mathbf{I} \end{bmatrix}_{N\cdot p_{\mathrm{I}} \times N\cdot p_{\mathrm{I}}},$$

where $\otimes$ is the Kronecker product, $\mathbf{I}$ is the identity matrix, and $G(\mathbf{x})$ is a sparse score vector. The weight matrices in the MoE layer are overall denoted as:

$$\mathbf{W}^{(1)} = \begin{pmatrix} \mathbf{W}_1^{(1)} \cdots \mathbf{W}_N^{(1)} \end{pmatrix}^{\mathrm{T}}_{N\cdot p_{\mathrm{I}} \times p},$$
$$\mathbf{W}^{(2)} = \begin{pmatrix} \mathbf{W}_1^{(2)} \cdots \mathbf{W}_N^{(2)} \end{pmatrix}_{p \times N\cdot p_{\mathrm{I}}}.$$

The output of the MoE layer can accordingly be expressed as (omitting the bias term for simplicity):

$$\mathbf{y} = \mathbf{W}^{(2)}\mathbf{R}\sigma\left(\mathbf{W}^{(1)}\mathbf{x}\right), \tag{2}$$

in which $\mathbf{R}$ encapsulates crucial expert knowledge and exhibits high sparsity, as typically only a selected number of experts are activated within each MoE layer.

To analyze the difficulty of compressing the space-occupying $N \cdot p_{\mathrm{I}} \times p$ matrices $\mathbf{W}^{(1)}, (\mathbf{W}^{(2)})^{\mathrm{T}}$, we turn to a theoretical framework *oblivious subspace embedding* [10, OSE] which can preserve any p-dimensional subspace of $\mathbb{R}^{N\cdot p_{\mathrm{I}}}$ (here, we focus on the subspaces spanned by $\mathbf{W}^{(1)}$ and $(\mathbf{W}^{(2)})^{\mathrm{T}}$) through a $d \times N \cdot p_{\mathrm{I}}$ random projection matrix $\mathbf{\Pi}$, with d being the projection dimension. The projection $\mathbf{\Pi}\mathbf{W}^{(1)}$ and $\mathbf{\Pi}(\mathbf{W}^{(2)})^{\mathrm{T}}$ can be considered as a sketch of the expert merging strategy.

We then recognize the limitation of expert merging via the lens of OSE. As per Cohen [10], d should be at least $O(p \log p/\varepsilon^2)$, where ε is the error tolerance level; in compressing MoE (N will NOT go to infinity), however, the scale of $p \log p/\varepsilon^2$ will be even larger than $N \cdot p_{\mathrm{I}}$ by simply setting $\varepsilon = 0.05$. The space gain from expert merging is thus usually marginal in this very practical case. In this regard, reducing the number of experts per MLP layer might not be that practical. Instead, we propose each expert should be kept during compression.

To alleviate the deficiency of OSE above, MC-SMoE [37] leverages the information from training data to reduce the scale of d, rendering the merging no longer data-"oblivious". However, due to the requirement to first do fine-tuning and the restriction that during inference the test data have to be i.i.d. as the training data, this therapy will be less valid in the zero-shot setting we mainly consider. We also empirically verify the above observations on the expert merging strategies in Section 5.4.

4.2 An Extraction Strategy Specific to the MoE Structure

Following our speculation that each expert in the MLP layer should be kept, we propose ResMoE, a framework to compress the representation of those experts through a Wasserstein barycenter expert and the residual matrices between each expert and the barycenter expert. Specifically, we extract an expert E_ω with the common pattern from all the experts, and then model the difference between E_ω and E_k by fewer parameters (we will introduce the difference modeling in Section 4.3). We first revisit a viewpoint that an MLP can be taken as the ensemble of multiple bottleneck-1 sub-MLPs [68, 73, 80]. We rewrite the MLP output as follows:

$$E_k(\mathbf{x}) = \sum_{i=1}^{p_{\mathrm{I}}} \left[\mathbf{W}_{k,\cdot,i}^{(2)} \cdot \sigma\left(\left\langle \mathbf{W}_{k,i,\cdot}^{(1)}, \mathbf{x}\right\rangle + \mathbf{b}_{k,i}^{(1)}\right)\right] + \mathbf{b}_k^{(2)}, \tag{3}$$

where by convention we represent the i-th row (resp. column) in the weight matrix $\mathbf{W}_k^{(1)}$ (resp. $\mathbf{W}_k^{(2)}$) as $\mathbf{W}_{k,i,\cdot}^{(1)}$ (resp. $\mathbf{W}_{k,\cdot,i}^{(2)}$), and $\sigma(\cdot)$ is the activation function. The summation implies that an MLP is the ensemble of a few bottleneck-1 sub-MLPs (the sum on the right-hand-side above), which allows a distributional perspective of MLP since the order of the sum does not matter.

Note that various expert network architectures can all be expressed using the structure of multiple bottleneck-1 sub-MLPs. The FFN in both Mixtral and DeepSeekMoE models uses a gated network following Llama [66], whose detailed form can be found in Appendix B.3.

Since $\mathbf{b}_i^{(2)}$ is not involved in the summation in Equation (3), we accordingly quantify the extraction as, after proper permutation, minimizing the squared Frobenius norm of differences between the original weight matrices in each expert and the weight matrices in the barycenter expert:

$$\min_{\substack{\mathbf{W}_\omega^{(1)}, \mathbf{b}_\omega^{(1)}, \mathbf{W}_\omega^{(2)} \\ \mathbf{T}_k \in \mathcal{P}, k \in [N]}} \frac{1}{N} \sum_{k=1}^{N} \left[\left\| \mathbf{T}_k \left[\mathbf{W}_k^{(1)}, \mathbf{b}_k^{(1)}\right] - \left[\mathbf{W}_\omega^{(1)}, \mathbf{b}_\omega^{(1)}\right] \right\|_F^2 + \left\| \mathbf{W}_k^{(2)} \mathbf{T}_k^{\mathrm{T}} - \mathbf{W}_\omega^{(2)} \right\|_F^2 \right], \tag{4}$$

where $\mathcal{P}$ is the class of p_{I}-by-p_{I} permutation matrices and $\mathbf{W}_\omega^{(1)} \in \mathbb{R}^{p_{\mathrm{I}} \times p}, \mathbf{b}_\omega^{(1)} \in \mathbb{R}^{p_{\mathrm{I}}}, \mathbf{W}_\omega^{(2)} \in \mathbb{R}^{p \times p_{\mathrm{I}}}$ are the weight matrices in the

barycenter expert E_ω. The introduction of the permutation matrices $\mathbf{T}_k$'s aligns with the distributional perspective of MLPs, that an MLP E_k is equivariant to the row permutation of its design matrix $\mathbf{W}_k = \left[\mathbf{W}_k^{(1)}, \mathbf{b}_k^{(1)}, (\mathbf{W}_k^{(2)})^{\mathrm{T}}\right] \in \mathbb{R}^{p_{\mathrm{I}} \times (2p+1)}$ as the sum's order in Equation (3) is inconsequential. It is worth noting that simultaneously permuting $\mathbf{W}_k^{(1)}$ and $\mathbf{W}_k^{(2)}$ does not affect the expert's output since the permutation matrix is orthogonal.

To solve problem (4), we propose to address the distribution of sub-MLPs within each expert, rather than layer-by-layer. The summation in Equation (3) clearly shows the correspondence between the i-th row $\mathbf{W}_{k,i,\cdot}^{(1)}$ of $\mathbf{W}_k^{(1)}$ and the i-th column $\mathbf{W}_{k,\cdot,i}^{(2)}$ of $\mathbf{W}_k^{(2)}$; to obtain the "embedding" of the sub-MLPs for the distributional formulation, we consider the original MLP E_k as a design matrix $\mathbf{W}_k$. We then run the algorithm for free-support Wasserstein barycenter [11] to obtain the weight matrix $\mathbf{W}_\omega = \left[\mathbf{W}_\omega^{(1)}, \mathbf{b}_\omega^{(1)}, (\mathbf{W}_\omega^{(2)})^{\mathrm{T}}\right]$ for the barycenter expert, with $\mathbf{W}_\omega$ being exactly the solution to the minimization problem (4).

To present the result, we respectively define μ_k's as the uniform distributions defined on the rows of the given $\mathbf{W}_k \in \mathbb{R}^{p_{\mathrm{I}} \times (2p+1)}$, for all $k = 1, 2, \cdots, N$, i.e., $\mu_k = \sum_{i=1}^{p_{\mathrm{I}}} 1/p_{\mathrm{I}} \cdot \delta_{\mathbf{W}_{k,i,\cdot}}$. Similarly μ_ω is uniformly distributed over the rows of $\mathbf{W}_\omega \in \mathbb{R}^{p_{\mathrm{I}} \times (2p+1)}$, and the notation μ_ω is interchangeable with $\mu_\omega(\mathbf{W}_\omega)$, which highlights the dependence on $\mathbf{W}_\omega$. We further denote the optimal transport matrix (w.r.t. W_2 distance) from μ_k to μ_ω as $\mathrm{OT}(\mu_k, \mu_\omega)$ as the solution of Equation (1). We can then give the following proposition (the proof is deferred to Appendix C.

Proposition 4.1. *Consider the solution $\mathbf{W}_\omega$ to the following free-support WB problem*

$$\min_{\mathbf{W}_\omega} \frac{1}{N} \sum_{k=1}^{N} W_2^2(\mu_k, \mu_\omega(\mathbf{W}_\omega)). \tag{5}$$

Then $\mathbf{W}_\omega$, along with $\mathbf{T}_k = p_{\mathrm{I}} \cdot \mathrm{OT}\left(\mu_k, \mu_\omega(\mathbf{W}_\omega)\right)$, is the solution to the optimization problem (4).

Remark. We note that all the experts E_k and the barycenter expert E_ω share the same size, i.e., $\mathbf{W}_k, \mathbf{W}_\omega \in \mathbb{R}^{p_{\mathrm{I}} \times (2p+1)}$. Therefore, the supports for distributions μ_k, μ_ω are of the same size p_{I}. In discrete optimal transport, there is a special property that for two discrete uniform distributions with support of the same size, the optimal transport matrix between the two distributions will be re-scaled as a permutation matrix [51]. The conclusion simplifies the computation since the permutation matrix is orthogonal ($\mathbf{T}_k\ (\mathbf{T}_k)^{\mathrm{T}} = \mathbf{I}$). The output of the extracted expert $E_\omega(\mathbf{x}) = \mathbf{W}_\omega^{(2)} \sigma\big(\mathbf{W}_\omega^{(1)} \mathbf{x} + \mathbf{b}_\omega^{(1)}\big)$ (adding $\mathbf{b}_\omega^{(2)}$) is automatically aligned with any expert $E_k(\mathbf{x}), \forall k \in [N]$ without additional transformations.

4.3 Residual Approximation and Expert Restoration

As the last step, we need to recover the selected expert from the barycenter expert. We choose two representative methods: unstructured pruning and SVD to remove the redundant parameters. (For unstructured pruning, we follow Han et al. [25] to zero out the parameters with small magnitude, in order to minimize the loss in problem (4).)

Table 1: Approximation error of Switch Transformer and Mixtral. The experts are frozen during the fine-tuning stage hence most of the deterministic methods give zero standard deviation. All the numbers are normalized by a factor p_{I} for better reference. UP stands for Unstructured Pruning, and SP stands for Structured Pruning.

	Switch Transformer	Mixtral
UP	34.27±0.00	10.26±0.00
Wanda	22.93±0.03	13.47±0.00
SP	87.00±0.00	27.09±0.00
SVD	56.44±0.00	21.70±0.00
M-SMoE	278.76±0.00	16.73±0.00
MEO	63.25±0.00	15.81±0.00
MLP Fusion	83.45±0.02	27.28±0.01
ResMoE (UP)	**22.05±0.02**	**6.60±0.01**
ResMoE (SVD)	48.91±0.01	14.63±0.04

We will store a compressed matrix $\mathbf{\Delta}_k$ to approximate $\mathbf{T}_k\mathbf{W}_k - \mathbf{W}_\omega$ and then use $\mathbf{\Delta}_k + \mathbf{W}_\omega$ to recover $\mathbf{T}_k\mathbf{W}_k$. We provide more implementation tricks for them in Appendix A.7 and the pseudocode of the algorithm in Appendix A.9. We remark that unstructured pruning produces comparable experimental results while SVD leads to more profound memory reduction.

In Figure 3, we visually compare ResMoE with previous baselines. Expert merging techniques consolidate multiple experts into a single entity, while pruning strategies involve the direct removal of connections within individual experts. ResMoE first obtains the common barycenter expert and subsequently compresses the residuals between each expert and the barycenter expert, which can be effectively and efficiently used for restoring in the inference stage.

5 Experimental Results

This section starts with our experimental setup, followed a preliminary evaluation of ResMoE's approximation error against various baselines in Table 1 and its performance in Tables 2 and 3, concluding with an ablation study. All models and methods are implemented in PyTorch. Switch Transformer is fine-tuned on a Tesla V100 32GB GPU, while Mixtral is tested on four such GPUs. Detailed experimental information is available in Appendix A.1.

5.1 Experiment Setup

Model backbones. Our evaluation encompasses two primary architectures: the GPT-style causal decoder-only model and the T5-style encoder-decoder model. Specifically, we utilize Mixtral [31] for the decoder-only model, featuring 8 experts per layer across 32 layers. For the encoder-decoder model, we employ the Switch Transformer [19] with a similar expert-layer configuration (switch-base-8) but with 12 encoder layers followed by 12 decoder layers. We fix the router and the experts during the supervised fine-tuning stage, based on the observation that preserving the original LLM's universal world information can enhance their performance [17, 26, 47]. This observation is empirically supported by our findings,

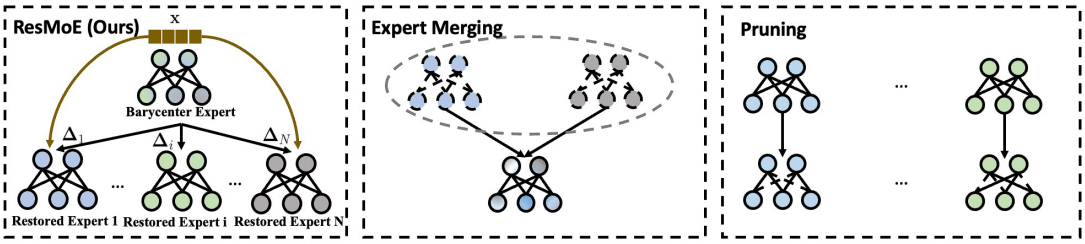

Figure 3: Comparisons between ResMoE and baselines. Dash lines denote the connections or neurons are deleted. Expert Merging reduces the number of experts by consolidating several into one, while pruning is applied directly to the experts. In contrast, ResMoE compresses the residual and barycenter experts, with the input x directed to the restored experts.

which demonstrate improved model performance. We also provide the efficiency analysis in Appendix A.8.

Compared methods. We compare our method with different types of baselines. For pruning, we employ both single-shot unstructured pruning [25, 35, 64] and structured pruning [38]. We also employ Wanda [63] for a more enhanced unstructured pruning method. We employ truncated SVD following Denton et al. [13]. For merging, we employ M-SMoE (the better-performing uncompressed version of MC-SMoE) [37] and MEO [28]. As the experimental results drop is profound in expert pruning [45] (50%), we employ it here only to Mixtral since our compression rate is more extreme (25%). We employ Git Re-Basin [2] for model fusion. Note that Git Re-Basin is not initially designed for MoE models, and we dynamically apply it as a fusion (merging) method according to its applicability to merge multiple models. We also compare our method with MLP Fusion [1], which aims to reduce the intermediate dimension of one expert MLP unit. As our compression rate is set to 25%, we perform different setups to all the methods to make sure they match this setting, detailed in Appendix A.3.

5.2 Preliminary Evaluation of Approximation Error

We calculate the approximation error of each method on the top 8 layers of Switch Transformer and the top 24 layers of Mixtral as a sanity check. The approximation error is defined as the Frobenius norm difference between the original and compressed weight matrices in the experts. Take Switch Transformer for example, since there is no bias in this model, the approximation error ϵ for one layer is defined as:

$$\epsilon = \frac{1}{N}\sum_{k=1}^{N}\left[\left\|\mathbf{T}_k\mathbf{W}_k^{(1)} - \hat{\mathbf{W}}_k^{(1)}\right\|_F^2 + \left\|\mathbf{W}_k^{(2)}\mathbf{T}_k^T - \hat{\mathbf{W}}_k^{(2)}\right\|_F^2\right],$$

where $\hat{\mathbf{W}}_k$ denotes the matrix post-application of each method. For ResMoE, $\hat{\mathbf{W}} = \mathbf{W}_\omega + \mathbf{\Delta}_k$, with $\mathbf{\Delta}_k$ being the compressed residual matrix derived from each layer, and $\mathbf{W}_\omega$ as the Wasserstein barycenter matrix. For merge methods, $\hat{\mathbf{W}} = \mathbf{W}_\omega$, where $\mathbf{W}_\omega$ is the merged center of each group. For methods not involving permutation operations, we set $\mathbf{T}_k = \mathbf{I}$. Specifically for MLP fusion which reduces the MLP's weight matrix size, it still allows approximation error computation, as detailed in A.5. Notably, as we freeze the experts during fine-tuning, most methods show zero standard deviation. It is worth mentioning that given Wanda is not data-agnostic, it has a standard deviation for different tasks Switch Transformer is fine-tuned on. As for Mixtral, we follow the zero-shot setting of Sun et al. [63] to use the C4 dataset [56] to perform the algorithm, hence leading to the zero standard deviation of it on Mixtral.

Table 1 shows that ResMoE achieves the lowest Approximation error among all the methods. We use the acronyms UP to represent Unstructured Pruning and SP for Structured Pruning. The results prove that ResMoE manages to retain not only the output of the original model but also the integrity of the internal matrices. Thus, this preliminary experiment successfully validates Proposition 4.1.

5.3 Natural Language Understanding

Experiment setup. Switch Transformer is fine-tuned then compressed during the inference stage on four natural language understanding (NLU) GLUE tasks, SST-2 [61], MRPC [16], CoLA [71] and MNLI [76]. All the results are reported with accuracy. Here, all the experiments are conducted using different seeds for three rounds. As not all the layers of Switch Transformer are sparse MoE layers, we perform all the methods at the top 4 encoder's MoE layers and the top 4 decoder's MoE layers.

Results. Table 2 provides the results of Switch Transformer. ResMoE (UP) consistently surpasses all baseline methods, while ResMoE (SVD) manages to surpass most of the baseline methods, underscoring its efficiency. Unstructured pruning effectively preserves the original performance, whereas structured pruning, applied neuron-wise, exhibits a more pronounced drop. This observation aligns well with our choice of unstructured pruning over structured pruning. We also observe that Wanda performs even worse than vanilla unstructured pruning. This may be due to the fact that Sun et al. [63] set the compression ratio to 50%, while our setting retains only 25% of the parameters, leading to a more significant performance drop. We note a significant difference in performance when applying pruning and SVD to experts, depending on whether the weights were concatenated or separate. A possible explanation is that pruning dynamically zeroes out less important weights, retaining crucial ones when expert weights are concatenated, indicating the benefit of preserving expert-level relationships for model performance. In addition, the suboptimal results from Git Re-Basin further support our proposition that previous layer-wise fusion methods are limited in their effectiveness. These methods may fail to adequately capture the complexities of layer interactions, leading to less optimal outcomes when compared to more holistic approaches. Although most methods manage to preserve

Table 2: Evaluation results of Switch Transformer on four GLUE NLU tasks (measured in accuracy). UP stands for Unstructured Pruning, and SP stands for Structured Pruning. We use "concat" and "sep" to denote the concatenated and separate processing of the expert weights. Bold indicates the best score for each metric, while underlined values represent the second-best.

	SST-2	MRPC	CoLA	MNLI
Switch Transformer	93.92±0.18	89.54±0.86	82.29±0.28	87.82±0.15
UP (concat)	93.12±0.23	87.75±1.12	81.40±0.48	85.32±0.66
UP (sep)	90.21±0.44	78.92±3.96	79.33±0.80	76.06±5.47
Wanda	92.39±0.18	86.74±0.93	80.59±1.01	84.20±0.19
SP (concat)	90.67±0.36	<u>88.72±0.85</u>	79.39±0.42	85.19±0.22
SP (sep)	83.60±1.15	80.88±2.12	76.89±0.93	81.87±1.19
SVD (concat)	92.47±0.04	87.58±1.02	75.90±0.03	85.86±0.07
SVD (sep)	92.59±0.25	87.25±0.93	<u>81.62±0.32</u>	86.04±0.05
M-SMoE	<u>93.31±0.53</u>	87.42±1.06	80.06±0.68	85.72±0.27
Git Re-Basin	84.94±0.86	85.70±0.46	61.90±1.24	83.76±0.84
MEO	92.73±0.39	86.77±0.93	79.99±0.83	85.29±0.37
MLP Fusion	91.86±0.30	88.40±0.59	79.64±0.03	85.72±0.19
ResMoE (UP)	**93.58±0.07**	**89.21±0.49**	**82.13±0.07**	**86.13±0.09**
ResMoE (SVD)	92.85±0.05	88.18±0.48	76.88±0.08	<u>86.08±0.03</u>

Table 3: Zero-shot results of Mixtral. Most of the methods are deterministic based on the model's weights, resulting in a 0 standard deviation. Bold indicates the best score for each metric, while underlined values represent the second-best. On the WikiText dataset, where perplexity serves as the evaluation metric. The down-arrow notation (↓) indicates that a lower metric represents better performance.

	WikiText (PPL) ↓	LAMBADA (ACC)	PIQA (ACC)	WinoGrande (ACC)
Mixtral	3.87±0.00	74.05±0.00	82.37±0.00	77.11±0.00
UP	13.03±0.00	36.10±0.00	72.09±0.00	68.59±0.00
Wanda	34.57±0.00	18.73±0.00	63.82±0.00	59.75±0.00
SP	13851.63±0.00	0.00±0.00	53.05±0.00	47.91±0.00
SVD	267.94±0.00	16.09±0.00	59.47±0.00	56.99±0.00
M-SMoE	10.45±0.00	58.57±0.00	73.56±0.00	69.61±0.00
Git Re-Basin	9.96±0.00	59.09±0.00	74.70±0.00	69.22±0.00
MEO	8.32±0.00	62.93±0.00	75.84±0.00	70.48±0.00
Expert Pruning	8.14±0.00	59.07±0.00	76.82±0.00	70.88±0.00
MLP Fusion	80.06±5.55	5.12±0.49	66.67±0.25	56.80±0.97
ResMoE (UP)	**5.38±0.04**	**69.44±0.16**	**80.81±0.19**	**74.45±0.23**
ResMoE (SVD)	<u>7.26±0.05</u>	<u>64.72±0.15</u>	<u>78.02±0.14</u>	<u>73.16±0.09</u>

the model's performance well in these NLU tasks, they result in a dramatic drop in subsequent zero-shot natural language generation (NLG) tasks.

5.4 Zero-shot Natural Language Generation

Experiment setup. Mixtral is tested on WikiText (Language Modelling) [46], LAMBADA (Language Modelling) [50], PIQA (Question Answering) [5] and WinoGrande (Common Sense Reasoning) [57]. The result of WikiText is given by perplexity, while accuracy metrics are used for the others. As Mixtral's results are tested with zero-shot and fixed weights, this ensures deterministic outcomes for most of the evaluated methods, leading to a standard deviation of 0 for them. However, the Fusion and OT methods, which seek approximate optimization solutions starting from different initial conditions, exhibit variability and therefore have a non-zero standard deviation. Specifically, for Wanda, we follow the zero-shot setting of Sun et al. [63] to perform the algorithm on the C4 dataset [56]. All the methods are performed on the top 24 layers, and reduce the parameter counts of the experts to 25%.

Results. Table 3 presents the results for Mixtral, where both ResMoE (UP) and ResMoE (SVD) consistently outperform all baseline methods, demonstrating their effectiveness in both NLU and NLG tasks. Notably, structured pruning results in a substantial performance loss for Mixtral, likely due to its larger hidden dimension (4,096), where neuron-wise weight pruning could lead to significant information loss, a situation reminiscent of the MLP Fusion

Table 4: The comparison of accuracy between vanilla pruning, average expert, Git Re-Basin expert, vanilla SVD, and our method. Here UP means Unstructured Pruning, WB stands for Wasserstein barycenter. Bold results are better scores under each metric.

	Switch Transformer			Mixtral		
	SST-2	MRPC	MNLI	LAMBADA	PIQA	WinoGrande
UP	93.12±0.18	87.75±1.12	85.32±0.66	36.10±0.00	72.09±0.00	68.59±0.00
Avg + UP	92.81±0.42	89.13±0.96	86.00±0.18	67.38±0.00	78.89±0.00	73.95±0.00
Git + UP	92.62±0.17	88.89±0.56	**86.23±0.13**	46.11±0.00	70.95±0.00	67.72±0.00
WB + UP	**93.58±0.07**	**89.21±0.49**	86.13±0.09	**69.44±0.16**	**80.81±0.19**	**74.45±0.23**
SVD	92.47±0.04	87.58±1.02	85.86±0.07	16.09±0.00	59.47±0.00	56.99±0.00
WB + SVD	**92.85±0.05**	**88.18±0.48**	**86.08±0.03**	**64.72±0.15**	**78.02±0.14**	**73.16±0.09**

case. It is important to note that Mixtral's experts are initialized through a copy-and-paste method, as opposed to the random Gaussian initialization in Switch Transformer, leading to more uniform weight distributions in Mixtral. This uniformity might contribute to the enhanced performance observed with merge methods in Mixtral. However, the superior performance of ResMoE over merge methods further supports our hypothesis about the latter's reduced effectiveness in more generalized scenarios.

5.5 Ablation Studies

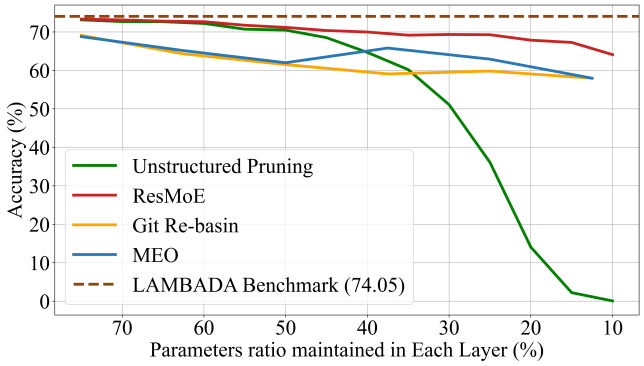

Figure 4: Performance of selected baseline methods on Mixtral w.r.t. various compression rates on the LAMBADA dataset. Note that MEO and Git Re-Basin can only merge experts into at least one so they cannot reach the 10% compression rate.

The effectiveness of Wasserstein barycenter. In ResMoE, we choose to compress the residual matrices between the original experts and the barycenter expert. Here, we study the effectiveness of this choice by conducting the vanilla unstructured pruning/SVD without this barycenter expert.

We also conduct the ablation study on the choice of using optimal transport to calculate the barycenter expert. We compare our barycenter with Git Re-Basin [2] center and with the average center. Considering the generally better performance of unstructured pruning, we conduct these two ablations under unstructured pruning. The difference between the ablation here and Tables 2 and 3 for Git Re-Basin is, during the ablation study, we merge the 8 experts into one expert to obtain the center expert, and follow the framework of ResMoE to prune the residuals. While in Tables 2 and 3, we follow the setting of Ainsworth et al. [2], Li et al. [37] to merge 8 experts into 2 experts, as our compression ratio is set to 25%. Note that we do not contain OT fusion [60] here, since our experiment on calculating the barycenter in their layer-by-layer form takes more than 4 days to complete, empirically supports our state that the layer-by-layer strategy introduces computational overhead to the process. When performing the algorithm, we observe that Git Re-Basin returns the average center for most of the layers for Mixtral, likely because the dimension of Mixtral reaches a high level (4,096 for the hidden dimension and 14,336 for the inner dimension), and this method is not scalable for such a large model. The outcomes in terms of output performance in Table 4 clearly demonstrate the beneficial impact of incorporating the barycenter expert.

The impact of compression rate. In the main experiment, we set the compression rate to retain 25% of the parameter counts. Additionally, we explored the impact of adjusting this rate to different levels. Figure 4 provides the results of Mixtral on the LAMBADA dataset. Remarkably, with the compression rate set to 10%, ResMoE (UP) manages to achieve results that are not only comparable but even surpass those of baseline methods set at a 30% compression rate.

The scalability of our method. Our main experiments are conducted on Switch Transformer (switch-base-8) and Mixtral, both with 8 experts per layer. To test the scalability of ResMoE, we conduct additional experiments on switch-base-16 and DeepseekMoE (64 experts per layer), to verify its ability to maintain performance with an increased number of experts.

Following the same fine-tuning settings as those used for switch-base-8, detailed in Appendix A.1, we limited our testing of switch-base-16 to the MRPC dataset due to the constraints of time-intensive supervised fine-tuning. Despite this limitation, Table 5 allows us to draw a similar conclusion as with switch-base-8, that ResMoE consistently demonstrates impressive results, affirming its efficacy in maintaining model accuracy with more experts per layer. Additionally, akin to the observations from switch-base-8, we notice that the choice between pruning or applying SVD to the weights, whether concatenated or separated, significantly influences the outcomes. This consistency across different model scales reinforces the impact of these compression techniques on the model's performance, highlighting the nuanced balance between efficiency and accuracy in model optimization. Due to the page limit, the details

Table 5: Evaluation Results of Switch Transformer (switch-base-16), with 16 experts per layer. UP stands for Unstructured Pruning, and SP stands for Structured Pruning. We use 'concat' to denote concatenate and 'sep' to denote separate.

	MRPC
Switch Transformer	90.03±0.45
UP (concat)	89.47±0.01
UP (sep)	88.48±0.75
SP (concat)	88.40±0.94
SP (sep)	87.34±0.46
SVD (concat)	88.48±0.62
SVD (sep)	88.48±0.62
M-SMoE	88.89±0.59
MEO	88.51±0.93
MLP Fusion	87.91±0.90
ResMoE (UP)	**89.62±0.01**

of DeepseekMoE can be found in Appendix A.2. Additionally, we provide the adaptability of ResMoE with expert parallelism and tensor parallelism in Appendix B.1.

6 Conclusions and Limitations

In this paper, we propose ResMoE, a data-agnostic MoE model approximation framework that reduces the memory usage of MoE LLMs without retraining. Instead of directly compressing the experts, we turn to approximating the residuals between the Wasserstein barycenter and the original experts. We prove the effectiveness of our method through comprehensive experiments on various backbone models, including Switch Transformer (with an encoder-decoder architecture) and the decoder-only Mixtral and DeepSeek-MoE. With ResMoE, we reduce the counts of parameters by up to 75% with both successful preservation of the original weight matrices and minimal performance loss in the downstream tasks. The future direction of this work can be the exploration of adopting different compression rates for each layer or even each expert (as experimented in LASER [58] and MC-SMoE [37]), or further combining our method with hardware quantization methods.

Limitations. While we have illustrated the success of ResMoE, it is also crucial to understand the limitations that arise in more complex settings: 1) Although producing impressive results, the space efficiency of storing the sparse matrices obtained from unstructured pruning is limited as detailed in Appendix A.7. 2) ResMoE is currently applied during the model inference stage. The resulting performance of applying it to fine-tuning is an open question that requires further investigation.

Acknowledgments

This work is supported by National Science Foundation under Award No. IIS-2117902, and Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021-32799/project accession no.1024178 from the USDA National Institute of Food and Agriculture. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

[1] Mengting Ai, Tianxin Wei, Yifan Chen, Zeming Guo, and Jingrui He. 2025. MLP Fusion: Towards Efficient Fine-tuning of Dense and Mixture-of-Experts Language Models. arXiv:2307.08941 [cs.LG] https://arxiv.org/abs/2307.08941

[2] Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. 2023. Git Re-Basin: Merging Models modulo Permutation Symmetries. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=CQsmMYmlP5T

[3] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403* (2023).

[4] Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-Bit Quantization of Transformer Neural Machine Language Translation Model. arXiv:1906.00532 [cs.LG]

[5] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7432–7439.

[6] Lingjie Chen, Ruizhong Qiu, Siyu Yuan, Zhining Liu, Tianxin Wei, Hyunsik Yoo, Zhichen Zeng, Deqing Yang, and Hanghang Tong. 2024. WAPITI: A Watermark for Finetuned Open-Source LLMs. *arXiv preprint arXiv:2410.06467* (2024).

[7] Yankai Chen, Huifeng Guo, Yingxue Zhang, Chen Ma, Ruiming Tang, Jingjie Li, and Irwin King. 2022. Learning Binarized Graph Representations with Multi-faceted Quantization Reinforcement for Top-K Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 168–178. https://doi.org/10.1145/3534678.3539452

[8] Yifan Chen, Rentian Yao, Yun Yang, and Jie Chen. 2023. A Gromov–Wasserstein Geometric View of Spectrum-Preserving Graph Coarsening. In *Proceedings of the 40th International Conference on Machine Learning*.

[9] Yifan Chen, Qi Zeng, Heng Ji, and Yun Yang. 2021. Skyformer: Remodel self-attention with gaussian kernel and nystr\" om method. In *Advances in Neural Information Processing Systems*.

[10] Michael B Cohen. 2016. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 278–287.

[11] Marco Cuturi and Arnaud Doucet. 2014. Fast computation of Wasserstein barycenters. In *International conference on machine learning*. PMLR, 685–693.

[12] Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models. arXiv:2401.06066 [cs.CL]

[13] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. 2014. Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/2afe4567e1bf64d32a5527244d104cea-Paper.pdf

[14] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. arXiv:2208.07339 [cs.LG]

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[16] William B. Dolan and Chris Brockett. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. https://aclanthology.org/I05-5002

[17] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, et al. 2023. LORAMOE: REVOLUTIONIZING MIXTURE OF EX-PERTS FOR MAINTAINING WORLD KNOWLEDGE IN LANGUAGE MODEL ALIGNMENT. *arXiv preprint arXiv:2312.09979* (2023).

[18] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. 2021. Multiscale Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 6824–6835.

[19] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research* 23, 1 (2022), 5232–5270.

[20] Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635* (2018).

[21] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. 2020. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 305, 11 pages.

[22] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. arXiv:2210.17323 [cs.LG]

[23] Ze-Feng Gao, Peiyu Liu, Wayne Xin Zhao, Zhong-Yi Lu, and Ji-Rong Wen. 2022. Parameter-Efficient Mixture-of-Experts Architecture for Pre-trained Language Models. In *Proceedings of the 29th International Conference on Computational Linguistics*, Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (Eds.). International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 3263–3273. https://aclanthology.org/2022.coling-1.288

[24] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129 (2021), 1789–1819.

[25] Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both Weights and Connections for Efficient Neural Network. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf

[26] Guande He, Jianfei Chen, and Jun Zhu. 2023. Preserving Pre-trained Features Helps Calibrate Fine-tuned Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=NI7StoWHJPT

[27] Huarui He, Jie Wang, Zhanqiu Zhang, and Feng Wu. 2022. Compressing Deep Graph Neural Networks via Adversarial Knowledge Distillation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 534–544. https://doi.org/10.1145/3534678.3539315

[28] Shwai He, Run-Ze Fan, Liang Ding, Li Shen, Tianyi Zhou, and Dacheng Tao. 2023. Merging Experts into One: Improving Computational Efficiency of Mixture of Experts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 14685–14691.

[29] Xinrui He, Yikun Ban, Jiaru Zou, Tianxin Wei, Curtiss B Cook, and Jingrui He. 2024. LLM-Forest for Health Tabular Data Imputation. *arXiv preprint arXiv:2410.21520* (2024).

[30] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL]

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

[32] Bowen Jin, Hansi Zeng, Guoyin Wang, Xiusi Chen, Tianxin Wei, Ruirui Li, Zhengyang Wang, Zheng Li, Yang Li, Hanqing Lu, et al. 2023. Language models as semantic indexers. *arXiv preprint arXiv:2310.07815* (2023).

[33] SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. 2020. DE-RRD: A Knowledge Distillation Framework for Recommender System. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 605–614. https://doi.org/10.1145/3340531.3412005

[34] Rui Kong, Yuanchun Li, Qingtian Feng, Weijun Wang, Linghe Kong, and Yunxin Liu. 2023. SwapMoE: Efficient Memory-Constrained Serving of Large Sparse MoE Models via Dynamic Expert Pruning and Swapping. arXiv:2308.15030 [cs.AI]

[35] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. 2019. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1VZqjAcYX

[36] Junyan Li, Li Lyna Zhang, Jiahang Xu, Yujing Wang, Shaoguang Yan, Yunqing Xia, Yuqing Yang, Ting Cao, Hao Sun, Weiwei Deng, Qi Zhang, and Mao Yang. 2023. Constraint-aware and Ranking-distilled Token Pruning for Efficient Transformer Inference. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 1280–1290. https://doi.org/10.1145/3580305.3599284

[37] Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. 2023. Merge, Then Compress: Demystify Efficient SMoE with Hints from Its Routing Policy. *arXiv preprint arXiv:2310.01334* (2023).

[38] Yixiao Li, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. LoSparse: Structured Compression of Large Language Models based on Low-Rank and Sparse Approximation. arXiv:2306.11222 [cs.LG]

[39] Zihao Li, Dongqi Fu, Mengting Ai, and Jingrui He. 2024. APEX^2: Adaptive and Extreme Summarization for Personalized Knowledge Graphs. arXiv:2412.17336 [cs.LG] https://arxiv.org/abs/2412.17336

[40] Zheng Li, Zijian Wang, Ming Tan, Ramesh Nallapati, Parminder Bhatia, Andrew Arnold, Bing Xiang, and Dan Roth. 2022. DQ-BART: Efficient Sequence-to-Sequence Model via Joint Distillation and Quantization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Dublin, Ireland, 203–211. https://doi.org/10.18653/v1/2022.acl-short.22

[41] Jiacheng Lin, Kun Qian, Haoyu Han, Nurendra Choudhary, Tianxin Wei, Zhongruo Wang, Sahika Genc, Edward W Huang, Sheng Wang, Karthik Subbian, et al. 2024. Unleashing the Power of LLMs as Multi-Modal Encoders for Text and Graph-Structured Data. *arXiv preprint arXiv:2410.11235* (2024).

[42] Chang Liu, Chenfei Lou, Runzhong Wang, Alan Yuhan Xi, Li Shen, and Junchi Yan. 2022. Deep Neural Network Fusion via Graph Matching with Applications to Model Ensemble and Federated Learning. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 13857–13869. https://proceedings.mlr.press/v162/liu22k.html

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10012–10022.

[44] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270* (2018).

[45] Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. 2024. Not All Experts are Equal: Efficient Expert Pruning and Skipping for Mixture-of-Experts Large Language Models. arXiv:2402.14800 [cs.CL] https://arxiv.org/abs/2402.14800

[46] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843* (2016).

[47] Jishnu Mukhoti, Yarin Gal, Philip H. S. Torr, and Puneet K. Dokania. 2023. Fine-tuning can cripple your foundation model; preserving features may be the solution. arXiv:2308.13320 [cs.LG]

[48] Alexandre Muzio, Alex Sun, and Churan He. 2024. SEER-MoE: Sparse Expert Efficiency through Regularization for Mixture-of-Experts. arXiv:2404.05089 [cs.CL] https://arxiv.org/abs/2404.05089

[49] OpenAI. 2022. Techniques for training large neural networks. https://openai.com/research/techniques-for-training-large-neural-networks

[50] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Katrin Erk and Noah A. Smith (Eds.). Association for Computational Linguistics, Berlin, Germany, 1525–1534. https://doi.org/10.18653/v1/P16-1144

[51] Gabriel Peyre and Marco Cuturi. 2020. Computational Optimal Transport. arXiv:1803.00567 [stat.ML]

[52] Yunzhe Qi, Yikun Ban, and Jingrui He. 2023. Graph neural bandits. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1920–1931.

[53] Ruizhong Qiu and Hanghang Tong. [n. d.]. Gradient Compressed Sensing: A Query-Efficient Gradient Estimator for High-Dimensional Zeroth-Order Optimization. In *Forty-first International Conference on Machine Learning*.

[54] Ruizhong Qiu, Zhe Xu, Wenxuan Bao, and Hanghang Tong. 2024. Ask, and it shall be given: Turing completeness of prompting. *arXiv preprint arXiv:2411.01992* (2024).

[55] Ruizhong Qiu, Weiliang Will Zeng, Hanghang Tong, James Ezick, and Christopher Lott. 2024. How Efficient is LLM-Generated Code? A Rigorous & High-Standard Benchmark. *arXiv preprint arXiv:2406.06647* (2024).

[56] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683 [cs.LG] https://arxiv.org/abs/1910.10683

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

[58] Pratyusha Sharma, Jordan T. Ash, and Dipendra Misra. 2023. The Truth is in There: Improving Reasoning in Language Models with Layer-Selective Rank Reduction. arXiv:2312.13558 [cs.LG]

[59] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. arXiv:1701.06538 [cs.LG]

[60] Sidak Pal Singh and Martin Jaggi. 2020. Model fusion via optimal transport. *Advances in Neural Information Processing Systems* 33 (2020), 22045–22055.

[61] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.

[62] George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. 2024. ZipIt! Merging Models from Different Tasks without Training. arXiv:2305.03053 [cs.CV] https://arxiv.org/abs/2305.03053

[63] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024. A Simple and Effective Pruning Approach for Large Language Models. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=PxoFut3dWW

[64] Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. 2020. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 6377–6389. https://proceedings.neurips.cc/paper_files/paper/2020/file/46a4378f835dc8040c8057beb6a2da52-Paper.pdf

[65] Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022. Compression of Generative Pre-trained Language Models via Quantization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 4821–4836. https://doi.org/10.18653/v1/2022.acl-long.331

[66] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL]

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[68] Benyou Wang, Yuxin Ren, Lifeng Shang, Xin Jiang, and Qun Liu. 2022. Exploring extreme parameter compression for pre-trained language models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=RftryyYyjiG

[69] Chaoqi Wang, Guodong Zhang, and Roger Grosse. 2020. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376* (2020).

[70] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2021. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 568–578.

[71] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics* 7 (2019), 625–641. https://doi.org/10.1162/tacl_a_00290

[72] Tianxin Wei, Yifan Chen, Xinrui He, and Jingrui He. 2024. Connecting Domains and Contrasting Samples: A Ladder for Domain Generalization. (2024).

[73] Tianxin Wei, Zeming Guo, Yifan Chen, and Jingrui He. 2023. NTK-approximating MLP Fusion for Efficient Language Model Fine-tuning. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 36821–36838. https://proceedings.mlr.press/v202/wei23b.html

[74] Tianxin Wei, Bowen Jin, Ruirui Li, Hansi Zeng, Zhengyang Wang, Jianhui Sun, Qingyu Yin, Hanqing Lu, Suhang Wang, Jingrui He, et al. 2024. Towards unified multi-modal personalization: Large vision-language models for generative recommendation and beyond. *arXiv preprint arXiv:2403.10667* (2024).

[75] Tianxin Wei, Ruizhong Qiu, Yifan Chen, Yunzhe Qi, Jiacheng Lin, Wenju Xu, Sreyashi Nag, Ruirui Li, Hanqing Lu, Zhengyang Wang, et al. 2024. Robust Watermarking for Diffusion Models: A Unified Multi-Dimensional Recipe. (2024).

[76] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Marilyn Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. https://doi.org/10.18653/v1/N18-1101

[77] Zhe Xu, Kaveh Hassani, Si Zhang, Hanqing Zeng, Michihiro Yasunaga, Limei Wang, Dongqi Fu, Ning Yao, Bo Long, and Hanghang Tong. 2024. Language

Models are Graph Learners. *arXiv preprint arXiv:2410.02296* (2024).
[78] Fuzhao Xue, Xiaoxin He, Xiaozhe Ren, Yuxuan Lou, and Yang You. 2022. One Student Knows All Experts Know: From Sparse to Dense. arXiv:2201.10890 [cs.LG]
[79] Yuchen Yan, Baoyu Jing, Lihui Liu, Ruijie Wang, Jinning Li, Tarek Abdelzaher, and Hanghang Tong. 2024. Reconciling competing sampling strategies of network embedding. *Advances in Neural Information Processing Systems* 36 (2024).
[80] Binhang Yuan, Cameron R Wolfe, Chen Dun, Yuxin Tang, Anastasios Kyrillidis, and Chris Jermaine. 2022. Distributed learning of fully connected neural networks using independent subnet training. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1581–1590.
[81] Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2024. Scalable and effective generative information retrieval. In *Proceedings of the ACM on Web Conference 2024*. 1441–1452.
[82] Zhichen Zeng, Boxin Du, Si Zhang, Yinglong Xia, Zhining Liu, and Hanghang Tong. 2024. Hierarchical multi-marginal optimal transport for network alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16660–16668.
[83] Zhichen Zeng, Xiaolong Liu, Mengyue Hang, Xiaoyi Liu, Qinghai Zhou, Chaofei Yang, Yiqun Liu, Yichen Ruan, Laming Chen, Yuxin Chen, et al. 2024. InterFormer: Towards Effective Heterogeneous Interaction Learning for Click-Through Rate Prediction. *arXiv preprint arXiv:2411.09852* (2024).
[84] Zhichen Zeng, Si Zhang, Yinglong Xia, and Hanghang Tong. 2023. Parrot: Position-aware regularized optimal transport for network alignment. In *Proceedings of the ACM Web Conference 2023*. 372–382.
[85] Beichuan Zhang, Chenggen Sun, Jianchao Tan, Xinjun Cai, Jun Zhao, Mengqi Miao, Kang Yin, Chengru Song, Na Mou, and Yang Song. 2023. SHARK: A Lightweight Model Compression Approach for Large-scale Recommender Systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 4930–4937. https://doi.org/10.1145/3583780.3615499
[86] Yuji Zhang, Sha Li, Jiateng Liu, Pengfei Yu, Yi R Fung, Jing Li, Manling Li, and Heng Ji. 2024. Knowledge overshadowing causes amalgamated hallucination in large language models. *arXiv preprint arXiv:2407.08039* (2024).
[87] Yu-Qi Zhang, Wen-Qi Wu, Jie Xu, Zai-Xiang Tang, Shi-Jia Li, Ling Li, He-Qing Wu, Xiao Ma, Ji-Sheng Liu, De-Pei Wu, et al. 2024. A clinical predictive model for pre-transplantation Klebsiella pneumoniae colonization and relevance for clinical outcomes in patients receiving allogeneic hematopoietic stem cell transplantation. *Microbiology Spectrum* 12, 2 (2024).
[88] Jiaru Zou, Qing Wang, Pratyush Thakur, and Nickvash Kani. 2024. STEM-POM: Evaluating Language Models Math-Symbol Reasoning in Document Parsing. *arXiv preprint arXiv:2411.00387* (2024).
[89] Jiaru Zou, Mengyu Zhou, Tao Li, Shi Han, and Dongmei Zhang. 2024. Promptintern: Saving inference costs by internalizing recurrent prompt during large language model fine-tuning. *arXiv preprint arXiv:2407.02211* (2024).