ZipMoE: A Theoretically-Grounded Mixture of Experts Approach for Parameter-Efficient Deep Learning

Lin Chen
Google Research

Kyriakos Axiotis
Google Research

Gang Fu
Google Research

Kaiyuan Wang
Google

MohammadHossein Bateni
Google Research

Vahab Mirrokni
Google Research

Abstract

The relentless growth of large language models (LLMs) presents formidable challenges for their training and deployment. To address this critical bottleneck, we introduce ZipMoE, a novel family of parameter-efficient building blocks inspired by the Mixture of Experts (MoE) paradigm. ZipMoE offers a modular and highly efficient substitute for conventional fully connected layers. We provide a rigorous theoretical analysis of ZipMoE's expressiveness, formally demonstrating its superior representational capacity over low-rank factorization. Furthermore, in a least squares regression setting, we prove that ZipMoE achieves a lower test error bound. Our empirical results—featuring comprehensive comparisons against low-rank, Monarch, and Kronecker methods—corroborate these theoretical findings. We demonstrate that ZipMoE consistently attains superior model quality under equivalent parameter or FLOP budgets, establishing it as a potent component for building efficient and powerful deep learning architectures.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable success in natural language processing, establishing state-of-the-art performance across a diverse range of tasks (Brown et al., 2020; Devlin et al., 2018). Despite their capabilities, fundamental limitations persist in areas such as factual accuracy (Ji et al., 2023), logical reasoning (Teng et al., 2023), and mathematical proficiency (Collins et al., 2024). A prevailing strategy to mitigate these shortcomings has been to scale up model size, a trend exemplified by the progression from GPT-3 (175B parameters) (Brown et al., 2020) to PaLM (540B parameters) (Chowdhery et al., 2023) and GPT-4 (estimated at 1.8 trillion parameters) (Achiam et al., 2023). However, this approach exacerbates the already substantial computational and memory demands of these models. This trajectory raises a crucial question: is it possible to achieve commensurate performance and learning capacity with substantially fewer parameters and floating-point operations (FLOPs)? This question frames the core problem of **parameter-efficient deep learning**, which this paper aims to address.

The drive for parameter and FLOP efficiency is not merely an academic exercise; it is a practical necessity. Reducing model parameters directly translates to lower memory footprints, enabling the deployment of powerful LLMs on resource-constrained hardware and mitigating the high costs of model storage (Xu et al., 2024). Concurrently, minimizing FLOPs accelerates inference, reduces energy consumption, and lowers the associated carbon footprint (Strubell et al., 2020)—critical factors for real-time applications and environmental sustainability. Ultimately, more efficient models democratize access to cutting-edge AI, empowering a broader community of researchers and developers to innovate. These challenges are particularly acute in the context of multimodal models, where architectural complexity often grows to accommodate diverse data streams (Baltrušaitis et al., 2018).

While parameter-efficient fine-tuning (PEFT) techniques (Ding et al., 2023; Han et al., 2024), such as the popular low-rank adaptation (LoRA) method (Hu et al., 2022), have proven effective for adapting pre-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

trained models, they primarily address efficiency during the fine-tuning stage. We argue that greater efficiency gains can be realized by integrating parameter-efficient principles into the entire model lifecycle, including pre-training. Instead of appending lightweight adapters to existing layers, we propose a fundamental redesign of the layers themselves.

To this end, we introduce ZipMoE, a novel neural network architecture inspired by the Mixture of Experts (MoE) framework (Jacobs et al., 1991; Jordan and Jacobs, 1994). In contrast to recent MoE models that function as ensembles of large sub-models (e.g., Mistral 8x7B (Jiang et al., 2024)), ZipMoE is conceived as a modular and efficient *building block* designed to replace standard fully connected layers. This granular design allows for seamless integration into existing network architectures with minimal modification, offering the potential for significant gains in both performance and flexibility without a corresponding explosion in parameter count.

Our main contributions are as follows:

- We introduce ZipMoE, a new family of parameter-efficient building blocks for neural networks. We present three architectural variants, ZipMoE-I, ZipMoE-II, and ZipMoE-III, which provide a flexible trade-off between expressive power and computational overhead.
- We conduct a rigorous theoretical analysis of ZipMoE, proving that it possesses strictly greater representational capacity than low-rank factorization with only a negligible increase in parameters. Furthermore, we prove that ZipMoE achieves a lower test error bound than optimal low-rank factorization in a least squares regression setting.
- We empirically demonstrate the superiority of ZipMoE through extensive experiments on language modeling tasks. Our results show that ZipMoE consistently achieves higher model quality compared to competitive baselines, including low-rank factorization, Monarch (Dao et al., 2022), and Kronecker (Edalati et al., 2025) methods, under identical parameter or FLOP constraints.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents our proposed ZipMoE method and its theoretical guarantees. Section 4 presents our experimental results. Finally, Section 5 and Section 6 concludes the paper with a discussion of limitations and potential future directions.

2 RELATED WORK

Our work, ZipMoE, introduces a novel parameter-efficient layer by drawing inspiration from two converging lines of research: Mixture of Experts (MoE) architectures and parameter-efficient structured matrices.

Mixture of Experts (MoE) Architectures. The MoE concept, originating from early work on adaptive function approximation (Jacobs et al., 1991; Jordan and Jacobs, 1994), has been revitalized as a primary strategy for scaling large language models. Modern architectures such as GShard (Lepikhin et al., 2021), the Switch Transformer (Fedus et al., 2022), GLaM (Du et al., 2022), and Mixtral (Jiang et al., 2024) employ sparse, conditional computation. In this paradigm, a routing mechanism activates only a small subset of large "expert" sub-networks (typically feed-forward blocks) for each input token. This allows for a dramatic increase in total model parameters and capacity while keeping the computational cost for inference relatively constant. Recent efforts have also focused on improving expert specialization at a finer granularity (Dai et al., 2024). ZipMoE fundamentally diverges from this paradigm. Instead of using MoE for sparse routing between large network blocks, it re-imagines the concept as a deterministic, compositional method for constructing a single, dense, and highly expressive weight matrix from smaller, low-rank "expert" components.

Parameter-Efficient Structured Matrices. The second area of relevant work is the development of parameter-efficient layers to reduce the computational and memory footprint of large models. A foundational approach in this domain is low-rank factorization, which forms the basis of popular Parameter-Efficient Fine-Tuning (PEFT) methods like Low-Rank Adaptation (LoRA) (Hu et al., 2022). Recognizing the expressive limitations of a simple low-rank structure, subsequent research has explored more sophisticated structured matrices. These include methods based on the Kronecker product, such as Compacter (Mahabadi et al., 2021), and those leveraging block-wise matrix multiplication, like Monarch matrices (Dao et al., 2022; Fu et al., 2024).

More recently, a new frontier has emerged that directly challenges the underlying low-rank assumption of many PEFT methods. This line of work on Structured Unrestricted-Rank Matrices (SURM) (Sehanobish et al., 2024) posits that the weight updates required for fine-tuning are generally not low-rank. Instead, SURM proposes using matrices with rich algebraic structures (e.g., Circulant or Toeplitz matrices) that are parameter-efficient but can be full-rank, offering a significant leap in expressive power. This body

of work presents two paths to overcoming the low-rank bottleneck: an *algebraic* path (SURM) and a *compositional* path. ZipMoE pioneers this compositional path. Its expressive power arises not from a predefined algebraic structure, but from the learned interactions between many simple, low-rank components, establishing it as a fundamentally new design principle for parameter-efficient layers.

3 MAIN RESULT

3.1 Model Description

Low-Rank Factorization as a Baseline To motivate the proposed ZipMoE method, we first revisit the well-established low-rank factorization technique for enhancing parameter efficiency in neural networks. Consider a fully connected layer with weight matrix $\boldsymbol{W} \in \mathbb{R}^{d_2 \times d_1}$, bias vector $\boldsymbol{b} \in \mathbb{R}^{d_2}$, and activation function σ, where d_1 and d_2 denote the input and output dimensions, respectively. Let $\boldsymbol{x}_{\text{in}} \in \mathbb{R}^{d_1}$ and $\boldsymbol{x}_{\text{out}} \in \mathbb{R}^{d_2}$ denote the input and output of this layer. The standard forward pass is given by

$$\boldsymbol{x}_{\text{out}} = \sigma(\boldsymbol{W}\boldsymbol{x}_{\text{in}} + \boldsymbol{b}).$$

The dense weight matrix $\boldsymbol{W}$ contains $d_1 d_2$ parameters.

Low-rank factorization replaces $\boldsymbol{W}$ with the product of two smaller matrices, $\boldsymbol{U} \in \mathbb{R}^{d_2 \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times d_1}$, where $r < \min(d_1, d_2)$ is the chosen rank. This yields the modified forward pass:

$$\boldsymbol{x}_{\text{out}} = \sigma(\boldsymbol{U}\boldsymbol{V}\boldsymbol{x}_{\text{in}} + \boldsymbol{b}).$$

This factorization reduces the number of parameters to $(d_1 + d_2)r$, which is significantly less than $d_1 d_2$ when r is sufficiently small.

ZipMoE Architecture ZipMoE builds upon the principle of low-rank factorization but introduces a novel mixture-of-experts approach to enhance expressiveness. ZipMoE utilizes two matrices $\boldsymbol{U} \in \mathbb{R}^{d_2 \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times d_1}$, each partitioned into K row-wise and column-wise blocks, respectively:

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{U}_1^\top & \boldsymbol{U}_2^\top & \cdots & \boldsymbol{U}_K^\top \end{pmatrix}^\top \in \mathbb{R}^{d_2 \times r},$$
$$\boldsymbol{V} = \begin{pmatrix} \boldsymbol{V}_1 & \boldsymbol{V}_2 & \cdots & \boldsymbol{V}_K \end{pmatrix} \in \mathbb{R}^{r \times d_1},$$

where $\boldsymbol{U}_i \in \mathbb{R}^{d_2/K \times r}$ and $\boldsymbol{V}_i \in \mathbb{R}^{r \times d_1/K}$. Similarly, the input vector $\boldsymbol{x}_{\text{in}} \in \mathbb{R}^{d_1}$ and output vector $\boldsymbol{x}_{\text{out}} \in \mathbb{R}^{d_2}$ are partitioned into K blocks:

$$\boldsymbol{x}_{\text{in}} = \begin{pmatrix} \boldsymbol{x}_1^\top & \boldsymbol{x}_2^\top & \cdots & \boldsymbol{x}_K^\top \end{pmatrix}^\top,$$
$$\boldsymbol{x}_{\text{out}} = \begin{pmatrix} \boldsymbol{x}_1'^\top & \boldsymbol{x}_2'^\top & \cdots & \boldsymbol{x}_K'^\top \end{pmatrix}^\top,$$

where $\boldsymbol{x}_i \in \mathbb{R}^{d_1/K}$ and $\boldsymbol{x}_i' \in \mathbb{R}^{d_2/K}$.

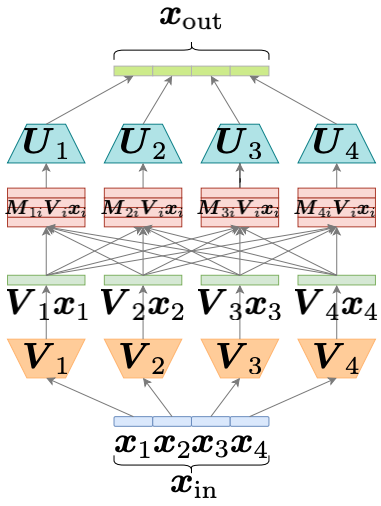


Figure 1: Overview of the ZipMoE Framework, highlighting its key components: input/output partitions, expert matrices ($\boldsymbol{U}_i, \boldsymbol{V}_j$), and the mixing matrix ($\boldsymbol{M}$).

Each product matrix $\boldsymbol{U}_i\boldsymbol{V}_j \in \mathbb{R}^{d_2/K \times d_1/K}$ acts as an "expert," mapping an input block $\boldsymbol{x}_j$ to a corresponding output block $\boldsymbol{x}_i'$. With K^2 such experts, a mixing matrix $\boldsymbol{M}$ is introduced to combine their outputs. This matrix is also expressed in block form:

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} & \cdots & \boldsymbol{M}_{1K} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} & \cdots & \boldsymbol{M}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{M}_{K1} & \boldsymbol{M}_{K2} & \cdots & \boldsymbol{M}_{KK} \end{pmatrix} \in \mathbb{R}^{Kr \times Kr},$$

where $\boldsymbol{M}_{ij} \in \mathbb{R}^{r \times r}$.

Let $\text{blockdiag}(\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_K)$ denote the block diagonal matrix with $\boldsymbol{U}_i$ on the diagonal. The ZipMoE parameterization is then defined as:

$$\tilde{\boldsymbol{U}}\boldsymbol{M}\tilde{\boldsymbol{V}}\boldsymbol{x}_{\text{in}} = \begin{pmatrix} \sum_{i\in[K]} \boldsymbol{U}_1\boldsymbol{M}_{1i}\boldsymbol{V}_i\boldsymbol{x}_i \\ \sum_{i\in[K]} \boldsymbol{U}_2\boldsymbol{M}_{2i}\boldsymbol{V}_i\boldsymbol{x}_i \\ \vdots \\ \sum_{i\in[K]} \boldsymbol{U}_K\boldsymbol{M}_{Ki}\boldsymbol{V}_i\boldsymbol{x}_i \end{pmatrix} \in \mathbb{R}^{d_2}, \quad (1)$$

where

$$\begin{aligned} \tilde{\boldsymbol{U}} &= \text{blockdiag}\left(\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_K\right) \in \mathbb{R}^{d_2 \times Kr}, \\ \tilde{\boldsymbol{V}} &= \text{blockdiag}\left(\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_K\right) \in \mathbb{R}^{Kr \times d_1}. \end{aligned} \quad (2)$$

The ZipMoE framework is illustrated in Fig. 1.

Equation 1 reveals that each output block row is a mixture of the outputs of these experts, weighted by the entries of $\boldsymbol{M}$. Specifically, the i-th block row is a mixture of the experts $\{\boldsymbol{U}_i\boldsymbol{V}_j \mid j \in [K]\}$. By inserting

Parameterization	Number of Parameters
Fully Connected	$d_1 d_2$
Low-Rank	$(d_1 + d_2)r$
ZipMoE-I	$(d_1 + d_2)r + K^2$
ZipMoE-II	$(d_1 + d_2)r + K^2 r$
ZipMoE-III	$(d_1 + d_2)r + 3K^2 r$

Table 1: The number of parameters of different parameterizations

$\boldsymbol{M}_{ij}$ between $\boldsymbol{U}_i$ and $\boldsymbol{V}_j$, we enable a more flexible and expressive mixture, enhancing the representation capacity of ZipMoE. While $\boldsymbol{M}$ has shape $Kr \times Kr$, it is parameterized with far fewer parameters to maintain efficiency, as detailed in the following proposed ZipMoE variants.

ZipMoE Variants We propose three variants of ZipMoE, each with increasing complexity in how they parameterize the mixing matrix $\boldsymbol{M}$:

- **ZipMoE-I**: Parameterizes $\boldsymbol{M}$ using $K \times K$ parameters (encoded in a matrix $\boldsymbol{A} \in \mathbb{R}^{K \times K}$ with entries a_{ij}), where $\boldsymbol{M}_{ij} = a_{ij} \boldsymbol{I}_r$.
- **ZipMoE-II**: Employs $K^2 r$ parameters $\{b_{ijk} \mid i, j \in [K], k \in [r]\}$ to parameterize $\boldsymbol{M}$, with $\boldsymbol{M}_{ij} = \operatorname{diag}(b_{ij})$, where $b_{ij} \triangleq (b_{ij1}, b_{ij2}, \ldots, b_{ijr}) \in \mathbb{R}^r$.
- **ZipMoE-III**: Utilizes $3K^2 r$ parameters $\{c_{ijk} \in \mathbb{R}, \boldsymbol{\alpha}_{ij} \in \mathbb{R}^r, \boldsymbol{\beta}_{ij} \in \mathbb{R}^r \mid i, j \in [K], k \in [r]\}$ to parameterize $\boldsymbol{M}$, with $\boldsymbol{M}_{ij} = \operatorname{diag}(c_{ij}) + \boldsymbol{\alpha}_{ij} \boldsymbol{\beta}_{ij}^\top$.

Remark 1. Note that ZipMoE-III generalizes both ZipMoE-II and ZipMoE-I. Specifically, ZipMoE-II can be recovered from ZipMoE-III by setting all $\boldsymbol{\alpha}_{ij}$ and $\boldsymbol{\beta}_{ij}$ to zero. Similarly, ZipMoE-I is a special case of ZipMoE-II where $b_{ijk} = a_{ij}$ for all $i, j \in [K]$ and $k \in [r]$.

Table 1 summarizes the parameter counts for the proposed ZipMoE variants, along with traditional low-rank factorization and fully connected layers. Compared to low-rank factorization, ZipMoE-I, II, and III introduce K^2, $K^2 r$, and $3K^2 r$ additional parameters, respectively. In practice, we typically set $K = 2, 4, 8,$ or 16, which is much smaller than d_1, d_2, and r. Therefore, the number of additional parameters is small compared to $(d_1 + d_2)r$, the parameter count for low-rank factorization.

3.2 Expressivity Analysis

This section analyzes the expressivity of the ZipMoE parameterization by examining the space of matrices it can represent. We theoretically compare ZipMoE to low-rank factorization, demonstrating that ZipMoE can represent a strictly larger space of matrices and can achieve a higher maximum rank.

Theorem 1 (Expressivity of ZipMoE, **proof in Supplementary Material**). *Consider the multilinear maps representing the low-rank factorization (LR) and ZipMoE-I parameterizations:*

$$\begin{aligned} T_{\text{LR}} : \mathbb{R}^{d_2 \times r} \times \mathbb{R}^{r \times d_1} &\to \mathbb{R}^{d_2 \times d_1}, \\ (\boldsymbol{U}, \boldsymbol{V}) &\mapsto \boldsymbol{U}\boldsymbol{V}, \\ T_{\text{NM-I}} : \mathbb{R}^{d_2 \times r} \times \mathbb{R}^{K \times K} \times \mathbb{R}^{r \times d_1} &\to \mathbb{R}^{d_2 \times d_1}, \\ (\boldsymbol{U}, \boldsymbol{A}, \boldsymbol{V}) &\mapsto \tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}}, \end{aligned}$$

where $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{V}}$ are as defined in Equation 2 and $\otimes$ denotes the Kronecker product. Let $\operatorname{im} T_{\text{LR}}$ and $\operatorname{im} T_{\text{NM-I}}$ denote the images of T_{LR} and $T_{\text{NM-I}}$, respectively.

Then, the following statements hold:

(i) Inclusion*: $\operatorname{im} T_{\text{LR}} \subseteq \operatorname{im} T_{\text{NM-I}}$.*

(ii) Strict Inclusion*: The inclusion is strict, i.e., $\operatorname{im} T_{\text{LR}} \subsetneq \operatorname{im} T_{\text{NM-I}}$, if and only if $r < \min\{d_1, d_2\}$ and $K > 1$.*

(iii) Rank Characterization*: In the case of strict inclusion, the maximum ranks attainable by matrices in the two images differ:*

$$\begin{aligned} \max_{\boldsymbol{W} \in \operatorname{im} T_{\text{LR}}} \operatorname{rank}(\boldsymbol{W}) &= r, \\ \max_{\boldsymbol{W} \in \operatorname{im} T_{\text{NM-I}}} \operatorname{rank}(\boldsymbol{W}) &= \min\{d_1, d_2, Kr\} > r. \end{aligned}$$

Remark 2. Item iii of Theorem 1 establishes a clear separation between the maximum attainable rank with low-rank factorization (which is r) and with ZipMoE-I (which is $\min\{d_1, d_2, Kr\}$). When r is sufficiently small to ensure $Kr < \min\{d_1, d_2\}$, this implies a potential K-fold increase in the maximum attainable rank using the ZipMoE-I parameterization.

Remark 3. Recall that ZipMoE-I is a special case of ZipMoE-II and ZipMoE-III (Remark 1). Denoting the images of the ZipMoE-II and ZipMoE-III parameterizations by $\operatorname{im} T_{\text{NM-II}}$ and $\operatorname{im} T_{\text{NM-III}}$ respectively, we have the following chain of inclusions:

$$\operatorname{im} T_{\text{LR}} \subseteq \operatorname{im} T_{\text{NM-I}} \subseteq \operatorname{im} T_{\text{NM-II}} \subseteq \operatorname{im} T_{\text{NM-III}}.$$

Furthermore, if $r < \min\{d_1, d_2\}$ and $K > 1$, the inclusions $\operatorname{im} T_{\text{LR}} \subseteq \operatorname{im} T_{\text{NM-II}}$ and $\operatorname{im} T_{\text{LR}} \subseteq \operatorname{im} T_{\text{NM-III}}$ are strict. Moreover, the maximum rank attainable by matrices in $\operatorname{im} T_{\text{NM-II}}$ and $\operatorname{im} T_{\text{NM-III}}$ is also $\min\{d_1, d_2, Kr\}$.

As shown in Table 1, compared to low-rank factorization, ZipMoE-I introduces an additional K^2 parameters, but as demonstrated in Theorem 1, it achieves a maximum rank K times that of low-rank factorization. This highlights the potential of ZipMoE to represent a richer set of matrices with only a modest increase in parameter count.

3.3 Theoretical Comparison: Least Squares Regression

To theoretically compare the performance of ZipMoE-I against low-rank factorization, we analyze their respective capabilities in fitting a least squares regression model. We consider a simplified setting where the output is a linear function of the input. This allows us to derive closed-form expressions for the optimal test error achieved by each method and to highlight the potential advantages of the ZipMoE-I parameterization.

Theorem 2 (Proof in Supplementary Material). *Consider a matrix $\boldsymbol{W} \in \mathbb{R}^{d_2 \times d_1}$ partitioned into $K \times K$ blocks:*

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}_{11} & \boldsymbol{W}_{12} & \cdots & \boldsymbol{W}_{1K} \\ \boldsymbol{W}_{21} & \boldsymbol{W}_{22} & \cdots & \boldsymbol{W}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{W}_{K1} & \boldsymbol{W}_{K2} & \cdots & \boldsymbol{W}_{KK} \end{pmatrix},$$

where each $\boldsymbol{W}_{ij} \in \mathbb{R}^{d_2/K \times d_1/K}$ is a submatrix of dimension $d_2/K \times d_1/K$. Let $\boldsymbol{x}_{\text{in}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{d_1})$ be an input vector, and define $\boldsymbol{x}_{\text{out}} = \boldsymbol{W}\boldsymbol{x}_{\text{in}}$. Recall that the ZipMoE-I parameterization is defined as the multilinear map $T_{\text{NM-I}}$ described in Theorem 1. We define the test errors of low-rank factorization and ZipMoE-I parameterization as

$$R_{\text{LR}} = \inf_{\boldsymbol{U} \in \mathbb{R}^{d_1 \times r}, \boldsymbol{V} \in \mathbb{R}^{r \times d_2}} \mathbb{E}\left[\left\|\boldsymbol{U}\boldsymbol{V}\boldsymbol{x}_{\text{in}} - \boldsymbol{x}_{\text{out}}\right\|_2^2\right],$$

$$R_{\text{NM-I}} = \inf_{\substack{\boldsymbol{U} \in \mathbb{R}^{d_1 \times r}, \\ \boldsymbol{A} \in \mathbb{R}^{K \times K}, \\ \boldsymbol{V} \in \mathbb{R}^{r \times d_2}}} \mathbb{E}\left[\left\|T_{\text{NM-I}}(\boldsymbol{U}, \boldsymbol{A}, \boldsymbol{V})\, \boldsymbol{x}_{\text{in}} - \boldsymbol{x}_{\text{out}}\right\|_2^2\right].$$

For any matrix $\boldsymbol{X}$, let $\sigma_i(\boldsymbol{X})$ denote its i-th largest singular value.

Then the following statements hold:

(i) $R_{\text{LR}} \geq R_{\text{NM-I}}$.

(ii) $R_{\text{LR}} = \sum_{l=r+1}^{\min\{d_1,d_2\}} \sigma_l^2(\boldsymbol{W})$ *and*

$$R_{\text{NM-I}} \geq \sum_{i,j \in [K]} \sum_{l=r+1}^{\min\{d_1,d_2\}/K} \sigma_l^2(\boldsymbol{W}_{ij}). \qquad (3)$$

(iii) Define a graph $G = (V, E)$ based on the nonzero blocks of $\boldsymbol{W}$. The vertex set V is defined as $V \triangleq \{(i,j) \in [K] \times [K] \mid \boldsymbol{W}_{ij} \neq \boldsymbol{0}\}$. The edge set E consists of pairs of distinct vertices that share a common index: $E \triangleq \{((i,j),(i',j')) \in V \times V \mid (i,j) \neq (i',j') \text{ and } (i = i' \text{ or } j = j')\}$. The equality in Equation 3 holds under either of the following sufficient conditions:

(a) The edge set is empty: $E = \varnothing$.

(b) The dimensions satisfy $d_1 = d_2 = Kr$, and the graph G is a forest. In this case, $R_{\text{NM-I}} = 0$.

Corollary 1. *If $d_1 = d_2 = d$ and $\boldsymbol{W}$ is the identity matrix, then $R_{\text{NM-I}} = (\frac{d}{K} - r)K = d - rK$ and $R_{\text{LR}} = d - r$.*

Proof of Corollary 1. Applying Item iiia of Theorem 2 yields $R_{\text{NM-I}} = (\frac{d}{K} - r)K = d - rK$ and $R_{\text{LR}} = d - r$. □

We now discuss the implications of Theorem 2. Item i demonstrates that ZipMoE-I consistently achieves a smaller or equal test error compared to low-rank factorization. Item ii establishes that the minimum test error for low-rank factorization is given by the sum of the squares of the singular values of $\boldsymbol{W}$ starting from the $(r+1)$-th largest. In essence, low-rank factorization discards the top r singular values. In contrast, for ZipMoE-I, the best-case minimum test error is attained by summing the squares of the singular values of each block $\boldsymbol{W}_{ij}$, starting from the $(r+1)$-th largest singular value within each block. This implies that ZipMoE-I effectively removes the top r singular values from *every* block $\boldsymbol{W}_{ij}$.

To illustrate the potential improvement offered by ZipMoE-I over low-rank factorization, consider the scenario where $d_1 = d_2 = Kr$. In this setting, each block $\boldsymbol{W}_{ij}$ is an $r \times r$ square matrix. For low-rank factorization, $R_{\text{LR}} = \sum_{l=r+1}^{Kr} \sigma_l^2(\boldsymbol{W})$, involving $(K-1)r$ singular values. However, for ZipMoE-I, each $r \times r$ block $\boldsymbol{W}_{ij}$ has only r singular values, all of which are effectively removed by the parameterization. Consequently, the right-hand side of Equation 3 becomes 0.

While the right-hand side of Equation 3 provides a lower bound for $R_{\text{NM-I}}$, Item iii presents two sufficient conditions under which this lower bound is achieved, implying that $R_{\text{NM-I}}$ can indeed be 0 in those cases.

Corollary 1 illustrates the implications of Theorem 2 for identity matrices. It demonstrates a clear gap between low-rank factorization and ZipMoE-I in approximating even a simple identity matrix. The squared Frobenius norm of a $d \times d$ identity matrix is d. Low-rank factorization reduces this to $d - r$, while ZipMoE-I reduces it

to $d - Kr$. Thus, the reduction achieved by ZipMoE-I is K times greater than that of low-rank factorization.

4 EXPERIMENTS

To validate our theoretical findings and assess the practical efficacy of ZipMoE, we conduct a comprehensive set of experiments designed to answer three key questions: (1) Does ZipMoE empirically demonstrate the superior expressive power suggested by our theoretical analysis (Theorem 1)? (2) Does this expressivity translate to improved performance on standard downstream tasks? (3) Can ZipMoE efficiently approximate complex, real-world components from state-of-the-art large language models?

To this end, we evaluate our three ZipMoE variants against a suite of strong and representative baselines for parameter-efficient structured matrices: standard Low-Rank Factorization, Monarch Matrices (Dao et al., 2022), and Kronecker Products (Edalati et al., 2025). Our evaluation spans three distinct tasks: learning a random permutation to probe intrinsic representational capacity, text classification on the AG News dataset to measure downstream performance, and approximating an FFN block from OPT-13B to test real-world applicability. Across all experiments, we focus on the Pareto efficiency of each method, comparing the achieved training loss against computational FLOPs, parameter count, and wall-clock time. Unless otherwise stated, all models were trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 512. All reported results are averaged over three independent runs to ensure stability. Experiments were conducted on a CPU-based cluster with 128GB of memory per node; while wall-clock times may be influenced by scheduler load, they are included for transparency.

4.1 Evaluating Intrinsic Expressive Power: The Random Permutation Task

Objective Our first experiment is designed as a direct empirical test of the expressivity analysis presented in Section 3.2. We challenge the models to learn a fixed random permutation, a task that requires capturing a high-rank, one-to-one mapping across a high-dimensional space. Methods with limited rank, such as low-rank factorization, are theoretically ill-suited for this problem, making it an ideal testbed for validating the enhanced representational capacity of ZipMoE.

Setup The network architecture mirrors the FFN block from OPT-13B (see Section 4.3): a two-layer MLP where the structured parameterization is applied. The first layer maps from $d_1 = 5120$ to an intermediate dimension of 20480 (with bias and ReLU activation), and the second layer maps back to $d_2 = 5120$ (with bias). The model is trained to approximate the mapping $y = P(x)$, where P is a fixed, randomly generated permutation matrix of size 5120×5120. The training data consists of 100,000 synthetic vectors $x \in \mathbb{R}^{5120}$ with features drawn from $\mathcal{N}(0, 5^2)$, split 90/10 for training and testing. For ZipMoE variants, we sweep the number of experts $K \in \{2, 4, \ldots, 128\}$ and the rank $r \in \{40, 80, \ldots, 1280\}$. A corresponding range of ranks is used for the low-rank baseline.

Results The results, presented in Fig. 2, unequivocally demonstrate the superior expressive power of the ZipMoE family. The baseline methods, including Low-rank, Monarch, and Kronecker, fail to make meaningful progress, with training losses stagnating between 10^0 and 10^{-2} (Figs. 2a to 2c). This outcome is expected, as their inherent structural constraints prevent them from capturing the complexity of a full-rank permutation.

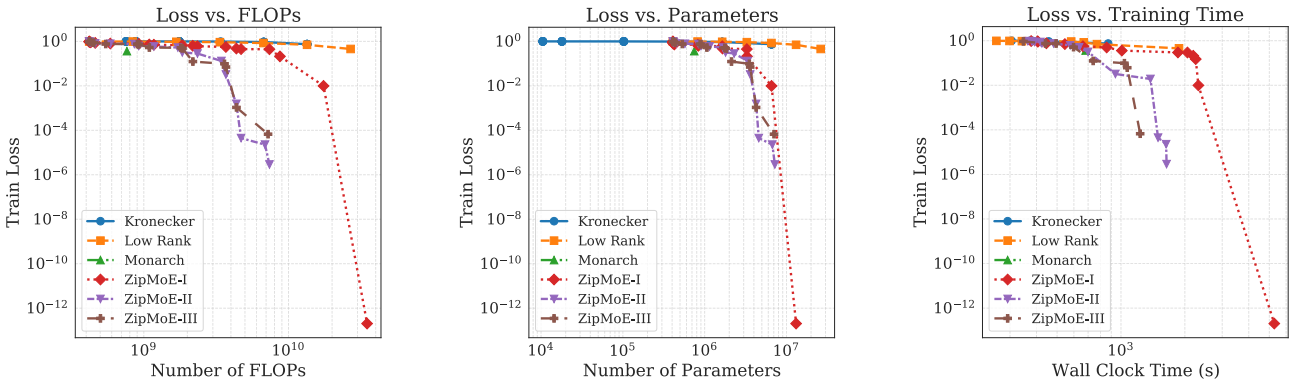
In stark contrast, all three ZipMoE variants successfully learn the permutation, achieving training losses that are orders of magnitude lower. ZipMoE-I, in particular, exhibits remarkable efficiency, reducing the loss to as low as 10^{-12}. ZipMoE-II and ZipMoE-III also attain exceptionally low loss values (e.g., 10^{-4} to 10^{-6}), decisively outperforming the baselines. This result provides strong empirical evidence for Theorem 1, confirming that ZipMoE's architecture unlocks a significantly richer and more expressive space of representable functions with only a modest parameter overhead.

4.2 Validating Performance on a Downstream NLP Task (AG News)

Objective Having established ZipMoE's superior intrinsic expressivity, we next investigate whether this advantage translates into improved performance on a practical, downstream application. We chose text classification on the widely-used AG News dataset for this purpose.

Setup The model consists of a text vectorization layer (max sequence length of 250), a 300-dimensional embedding layer, a global average pooling layer, the specific parameter-efficient layer being evaluated (ZipMoE or a baseline), and a final 4-way softmax classification head. We use the original dataset split from Zhang et al. (2015), with 120,000 training and 7,600 test samples. Models are trained for 10 epochs. For this task, we performed a hyperparameter sweep over the number of experts $K \in [2, 150]$ and rank $r \in [2, 300]$ for ZipMoE and the low-rank baseline.

Results As shown in Fig. 3, the ZipMoE variants again demonstrate a clear advantage. They consistently



(a) Loss vs. FLOPs (b) Loss vs. Parameters (c) Loss vs. Time

Figure 2: Training loss on the random permutation task. The ZipMoE family successfully learns the high-rank mapping, achieving dramatically lower loss than the baselines. Note the logarithmic scale on the y-axis.

outperform Monarch and Kronecker baselines across all three efficiency metrics (Figs. 3a to 3c). When compared against the strong low-rank factorization baseline, ZipMoE-II and ZipMoE-III are highly competitive, especially in terms of FLOPs-to-performance ratio (Fig. 3a). Notably, ZipMoE-I consistently establishes the most favorable Pareto frontier, achieving a lower training loss for any given budget of FLOPs, parameters, or training time. For example, ZipMoE-I reaches a training loss of approximately 0.175 with only 10^7 FLOPs. This experiment confirms that the theoretical benefits of ZipMoE are not merely an artifact of a synthetic task but yield tangible performance gains in a standard NLP application.

4.3 Approximating Real-World LLM Components (OPT-13B FFN Block)

Objective Our final and most challenging experiment assesses the ability of ZipMoE to approximate a real-world, large-scale neural network component: a feed-forward network (FFN) block from the OPT-13B language model (Zhang et al., 2022). This task serves as a practical analogue to our least squares regression analysis in Section 3.3 and directly tests the applicability of ZipMoE for building next-generation efficient LLMs.

Setup We isolate the first two fully connected layers of the OPT-13B FFN block. The first layer maps from $d_1 = 5120$ to 20480 (with bias and ReLU), and the second maps from 20480 back to $d_2 = 5120$ (with bias). We train our parameter-efficient models to minimize the mean squared error against the output of this frozen, pre-trained block. The input data is a synthetic dataset of 100,000 samples with 5120 features drawn from $\mathcal{N}(0, 5^2)$ (90/10 train/test split). Hyperparameter ranges for K and r are the same as in the permutation task.

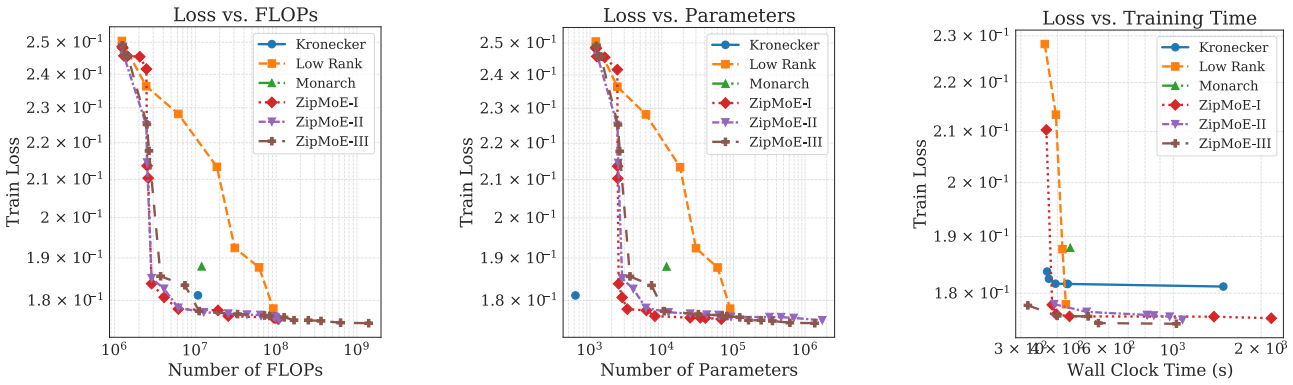
Results The results are summarized in Fig. 4. The top row (Figs. 4a to 4c) confirms that all ZipMoE variants significantly outperform the Monarch and Kronecker baselines. To provide a clearer view of the most competitive methods, the bottom row (Figs. 4d to 4f) focuses on the comparison between ZipMoE and standard low-rank factorization.

Across all metrics, the ZipMoE family demonstrates a superior trade-off between approximation quality (lower loss) and resource consumption. ZipMoE-I again proves to be the most efficient variant, consistently achieving the lowest loss for a given parameter or FLOP count. For instance, ZipMoE-I reaches a training loss of approximately 1.30×10^1 using around 10^{10} FLOPs, a level of performance that low-rank factorization cannot match with a similar budget. ZipMoE-II and ZipMoE-III also show strong, competitive performance, particularly in the FLOPs- and parameter-constrained regimes. These findings strongly support the potential of ZipMoE as a drop-in replacement for standard dense layers in large-scale models, offering a path towards more efficient yet powerful architectures.

4.4 Experiments on Gemma-2

To further evaluate ZipMoE, we conducted experiments on the Gemma-2-2B model (Gemma Team et al., 2024), specifically approximating the FFN block at Layer 13. As shown in Figure 5, ZipMoE consistently outperforms all baselines (Low-Rank, Monarch, and Kronecker) in terms of approximation quality.

The results in Figure 5 demonstrate that ZipMoE maintains a superior Pareto frontier across various parameter and FLOP budgets. This confirms that ZipMoE effectively captures the characteristics of `GeGLU` activa-



(a) Loss vs. FLOPs (b) Loss vs. Parameters (c) Loss vs. Time

Figure 3: Training loss on the AG News text classification task. ZipMoE variants, particularly ZipMoE-I, demonstrate superior efficiency, achieving lower loss for a given computational or parameter budget compared to baselines.

tions used in Gemma-2, providing a more expressive and efficient alternative to standard low-rank factorization.

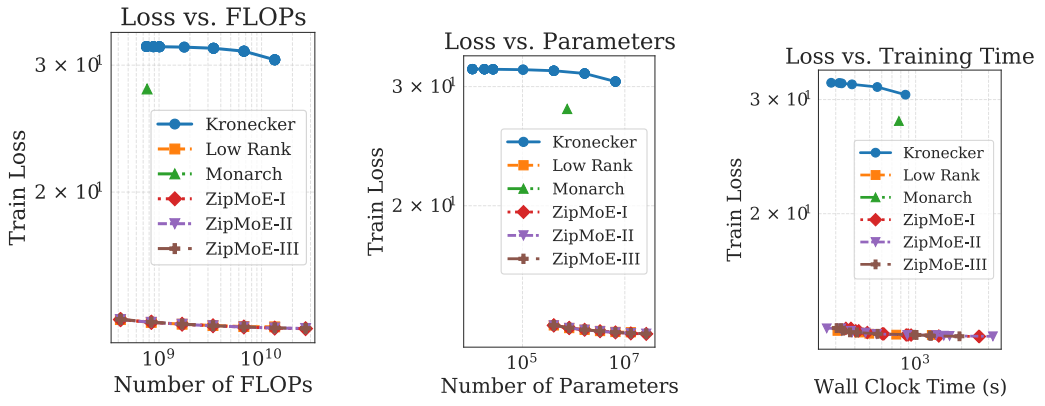
5 CONCLUSION

This paper introduces ZipMoE, a novel family of parameter-efficient building blocks designed to replace traditional fully connected layers in neural networks. We have theoretically demonstrated that ZipMoE, particularly the ZipMoE-I variant, possesses significantly greater expressive power than low-rank factorization, while incurring only a minimal increase in the number of parameters. Furthermore, we have established theoretical bounds on the test error of ZipMoE in a least squares regression setting, highlighting its potential for improved generalization. Our empirical results on language modeling tasks provide strong evidence for the practical effectiveness of ZipMoE. Across a range of experiments, ZipMoE variants consistently outperformed low-rank factorization, as well as Monarch and Kronecker methods, when compared under equivalent parameter or FLOP budgets. These findings underscore the potential of ZipMoE to enable the development of more efficient and powerful neural network architectures, particularly in the context of large language models where parameter and computational efficiency are paramount. By offering a modular and readily-integrable alternative to standard fully connected layers, ZipMoE paves the way for building the next generation of more compact and efficient deep learning models.

6 LIMITATIONS AND FUTURE WORK

While this work demonstrates the significant potential of ZipMoE, certain limitations warrant further investigation. Our current theoretical analysis focuses primarily on ZipMoE-I and its comparison to low-rank factorization. Extending this analysis to ZipMoE-II and ZipMoE-III, as well as to other matrix factorization methods like Monarch and Kronecker, would provide a more comprehensive theoretical understanding. Additionally, while we demonstrate the empirical effectiveness of ZipMoE, this work does not explore the impact of different optimizers and their hyperparameters on ZipMoE's performance.

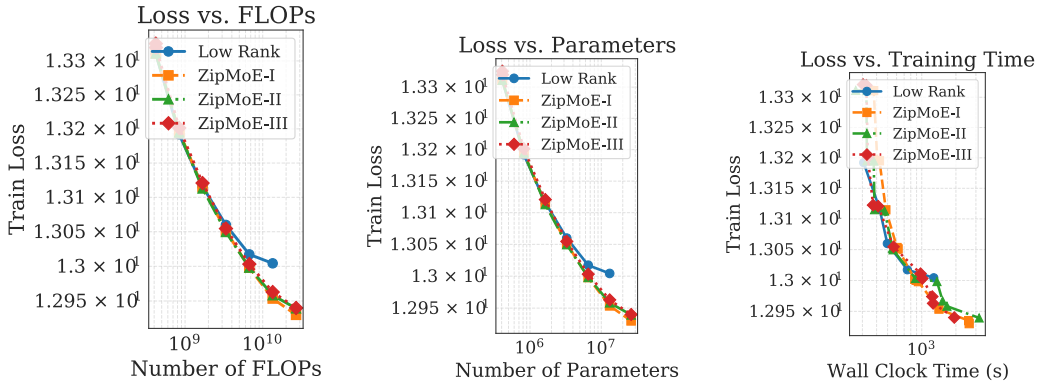
This study presents several promising avenues for future work. First, developing principled methods for selecting the optimal hyperparameters K (number of experts) and r (rank) is crucial to maximizing ZipMoE's efficiency. This could involve exploring adaptive mechanisms that dynamically adjust these parameters during training. Second, investigating the performance of ZipMoE within the context of parameter-efficient fine-tuning methods like LoRA (Hu et al., 2022) is a natural extension. Third, exploring the effectiveness of ZipMoE in the pre-training stage of large language models is of particular interest, given the potential for significant computational savings. Finally, evaluating the performance of stacked ZipMoE architectures, where multiple fully connected layers are replaced by ZipMoE blocks, could lead to further improvements in model quality and efficiency.



(a) Loss vs. FLOPs (all methods)

(b) Loss vs. Parameters (all methods)

(c) Loss vs. Time (all methods)

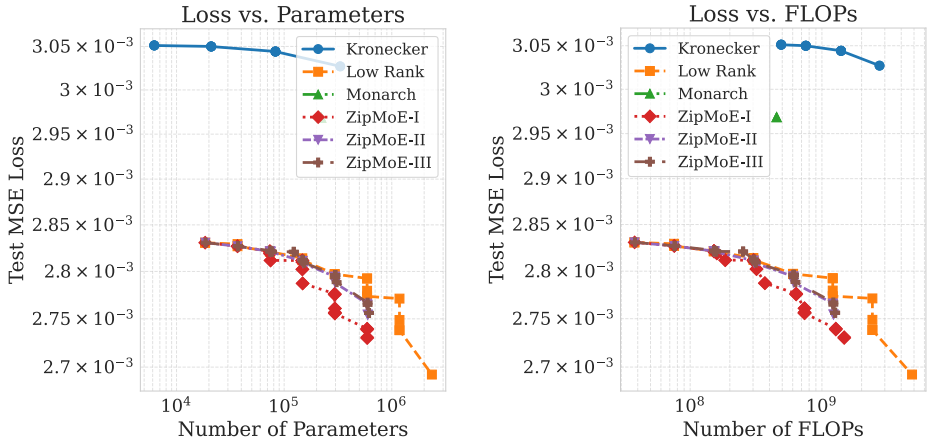


(d) Loss vs. FLOPs (zoom on best)

(e) Loss vs. Parameters (zoom on best)

(f) Loss vs. Time (zoom on best)

Figure 4: Training loss for approximating the OPT-13B FFN block. **Top row (a-c)**: Comparison of all methods. **Bottom row (d-f)**: A detailed comparison showing the consistent advantage of ZipMoE variants over the strong low-rank factorization baseline.



(a) Loss vs. Parameters

(b) Loss vs. FLOPS

Figure 5: Comparison of ZipMoE and baselines on Gemma-2-2B. Our method achieves lower MSE under the same parameter and computation constraints.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2023). Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Collins, K. M., Jiang, A. Q., Frieder, S., Wong, L., Zilka, M., Bhatt, U., Lukasiewicz, T., Wu, Y., Tenenbaum, J. B., Hart, W., et al. (2024). Evaluating language models for mathematics through interactions. *Proceedings of the National Academy of Sciences*, 121(24):e2318124121.

Dai, D., Deng, C., Zhao, C., Xu, R., Gao, H., Chen, D., Li, J., Zeng, W., Yu, X., Wu, Y., et al. (2024). Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297.

Dao, T., Chen, B., Sohoni, N. S., Desai, A., Poli, M., Grogan, J., Liu, A., Rao, A., Rudra, A., and Ré, C. (2022). Monarch: Expressive structured matrices for efficient and accurate training. In *International Conference on Machine Learning*, pages 4690–4721. PMLR.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., et al. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A. W., Firat, O., et al. (2022). Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

Edalati, A., Tahaei, M., Kobyzev, I., Nia, V. P., Clark, J. J., and Rezagholizadeh, M. (2025). Krona: Parameter-efficient tuning with kronecker adapter. *Enhancing LLM Performance: Efficacy, Fine-Tuning, and Inference Techniques*, 7:49.

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

Fu, D. Y., Arora, S., Grogan, J., Johnson, I., Eyuboglu, E. S., Thomas, A. W., Spector, B., Poli, M., Rudra, A., and Ré, C. (2024). Monarch mixer: A simple sub-quadratic gemm-based architecture. In *Advances in Neural Information Processing Systems*, volume 36.

Gemma Team, Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., et al. (2024). Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Han, Z., Gao, C., Liu, J., Zhang, S. Q., et al. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. (2022). Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. (2024). Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. (2021). GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Mahabadi, R. K., Henderson, J., and Ruder, S. (2021). Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, volume 34, pages 1022–1035.

Sehanobish, A., Dubey, A., Choromanski, K., Chowdhury, S. B. R., Jain, D., Sindhwani, V., and Chaturvedi, S. (2024). Structured unrestricted-rank matrices for parameter efficient finetuning. In *Advances in Neural Information Processing Systems*, volume 37.

Strubell, E., Ganesh, A., and McCallum, A. (2020). Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13693–13696.

Teng, Z., Ning, R., Liu, J., Zhou, Q., Zhang, Y., et al. (2023). Glore: Evaluating logical reasoning of large language models. *arXiv preprint arXiv:2310.09107*.

Xu, J., Li, Z., Chen, W., Wang, Q., Gao, X., Cai, Q., and Ling, Z. (2024). On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088*.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. (2022). Opt: Open pre-trained transformer language models.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] [Justification: Parameter complexity (space) is analyzed in Section 3.1 and Table 1. Time complexity (FLOPs) is empirically measured and compared in Section 4.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] [Justification: Theorems 1 and 2 clearly state their mathematical setting and assumptions.]
   (b) Complete proofs of all theoretical results. [Yes] [Justification: Proofs are provided in the supplementary material, as referenced in the main text.]
   (c) Clear explanations of any assumptions. [Yes] [Justification: The assumptions for the least squares regression (Section 3.3) are explained as a simplified setting for theoretical comparison.]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [No] [Justification: Synthetic data generation is described, but full experimental code is not provided.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] [Justification: Section 4 (Experiments) intro provides global details (optimizer, LR, batch size). Each subsection provides task-specific details (e.g., K and r ranges, epoch counts, data splits).]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [No] [Justification: The paper states results are "averaged over three independent runs," but the plots do not include

error bars (e.g., standard deviation) to show variance, which is a significant weakness.]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] [Justification: Section 4 intro states: "all experiments were conducted on a CPU with 128GB of memory."]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes] [Justification: We cite the creators of the AG News dataset (Zhang et al., 2015) and the OPT-13B model (Zhang et al., 2022).]

   (b) The license information of the assets, if applicable. [No] [Justification: License information for AG News and OPT-13B is not currently included.]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

   (d) Information about consent from data providers/curators. [Not Applicable] [Justification: We use standard, public datasets.]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A PROOF OF THEOREM 1

Proof of Theorem 1. **Proof of Item i.** The inclusion $\operatorname{im} T_{\text{LR}} \subseteq \operatorname{im} T_{\text{NM-I}}$ is straightforward. Setting $\boldsymbol{A} = \mathbf{1}_{K\times K}$ (the all-ones matrix), we have

$$T_{\text{NM-I}}(\boldsymbol{U}, \boldsymbol{A}, \boldsymbol{V}) = \tilde{\boldsymbol{U}}(\mathbf{1}_{K\times K} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}} = \boldsymbol{U}\boldsymbol{V} = T_{\text{LR}}(\boldsymbol{U}, \boldsymbol{V}).$$

Hence, $\operatorname{im} T_{\text{LR}} \subseteq \operatorname{im} T_{\text{NM-I}}$.

Proof of Item ii ("only if" part). Next, we establish that if $r \geq \min\{d_1, d_2\}$ or $K = 1$, then $\operatorname{im} T_{\text{LR}} = \operatorname{im} T_{\text{NM-I}}$.

Case 1: $K = 1$. In this case, $\boldsymbol{A}$ reduces to a scalar a, and we can absorb it into $\tilde{\boldsymbol{U}}$ or $\tilde{\boldsymbol{V}}$. Without loss of generality, let $\tilde{\boldsymbol{U}}' = a\boldsymbol{U}$. Then $\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}} = a\boldsymbol{U}\boldsymbol{V} = \tilde{\boldsymbol{U}}'\boldsymbol{V}$. Since any element in $\operatorname{im} T_{\text{NM-I}}$ can be represented as $a\boldsymbol{U}\boldsymbol{V}$ for some $a, \boldsymbol{U}, \boldsymbol{V}$, it is also in $\operatorname{im} T_{\text{LR}}$. Thus, $\operatorname{im} T_{\text{NM-I}} \subseteq \operatorname{im} T_{\text{LR}}$. The reverse inclusion has already been established, so $\operatorname{im} T_{\text{LR}} = \operatorname{im} T_{\text{NM-I}}$.

Case 2: $r \geq \min\{d_1, d_2\}$. The rank of any matrix in $\operatorname{im} T_{\text{NM-I}}$ is bounded above by $\min\{d_1, d_2\}$. Since $r \geq \min\{d_1, d_2\}$, for any $\boldsymbol{W} \in \operatorname{im} T_{\text{NM-I}}$ there exist matrices $\boldsymbol{U} \in \mathbb{R}^{d_2\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times d_1}$ such that $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{V}$. This implies $\boldsymbol{W} \in \operatorname{im} T_{\text{LR}}$, and hence $\operatorname{im} T_{\text{NM-I}} \subseteq \operatorname{im} T_{\text{LR}}$. The reverse inclusion $\operatorname{im} T_{\text{LR}} \subseteq \operatorname{im} T_{\text{NM-I}}$ has already been established, so we conclude $\operatorname{im} T_{\text{LR}} = \operatorname{im} T_{\text{NM-I}}$.

Proof of Item ii ("if" part) and Item iii. To show that the inclusion is strict under the assumptions $r < \min\{d_1, d_2\}$ and $K > 1$, we will prove that $\operatorname{im} T_{\text{LR}} \neq \operatorname{im} T_{\text{NM-I}}$. We will show that in both cases (case 1: $Kr \geq \max\{d_1, d_2\}$; case 2: $Kr < \max\{d_1, d_2\}$) there exist $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K \in \mathbb{R}^{d_2/K\times r}$ and $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_K \in \mathbb{R}^{r\times d_1/K}$, and $\boldsymbol{A} \in \mathbb{R}^{K\times K}$ such that $\operatorname{rank}(\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}}) = \min\{d_1, d_2, Kr\}$. Since $r < \min\{d_1, d_2\}$ and $K > 1$, this rank is strictly greater than r, and thus the corresponding matrix cannot be in $\operatorname{im} T_{\text{LR}}$.

Case 1: $Kr \geq \max\{d_1, d_2\}$. We choose $\boldsymbol{A} = \boldsymbol{I}_K$. We then pick K full-rank matrices $\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_K \in \mathbb{R}^{\frac{d_2}{K}\times\frac{d_1}{K}}$. Since $r \geq \max\{d_1/K, d_2/K\}$, for each $\boldsymbol{S}_i$, we can find $\boldsymbol{U}_i \in \mathbb{R}^{d_2/K\times r}, \boldsymbol{V}_i \in \mathbb{R}^{r\times d_1/K}$ such that $\boldsymbol{S}_i = \boldsymbol{U}_i\boldsymbol{V}_i$. (This can be achieved using the singular value decomposition of $\boldsymbol{S}_i = \boldsymbol{P}_i\boldsymbol{\Sigma}_i\boldsymbol{Q}_i^\top$, where $\boldsymbol{P}_i \in \mathbb{R}^{d_2/K\times d_2/K}$ and $\boldsymbol{Q}_i \in \mathbb{R}^{d_1/K\times d_1/K}$ are orthogonal matrices, and $\boldsymbol{\Sigma}_i \in \mathbb{R}^{d_2/K\times d_1/K}$ is a rectangular diagonal matrix. Then we set $\boldsymbol{U}_i = \begin{pmatrix}\boldsymbol{P}_i\boldsymbol{\Sigma}_i & \mathbf{0}_{d_2/K\times(r-d_1/K)}\end{pmatrix}$ and $\boldsymbol{V}_i = \begin{pmatrix}\boldsymbol{Q}_i^\top \\ \mathbf{0}_{(r-d_1/K)\times d_1/K}\end{pmatrix}$.) Then we have

$$\begin{aligned}
&\operatorname{rank}(\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}}) \\
=\ &\operatorname{rank}(\tilde{\boldsymbol{U}}\tilde{\boldsymbol{V}}) \\
=\ &\operatorname{rank}(\operatorname{blockdiag}(\boldsymbol{U}_1\boldsymbol{V}_1, \ldots, \boldsymbol{U}_K\boldsymbol{V}_K)) \\
=\ &\sum_{i=1}^{K} \operatorname{rank}(\boldsymbol{U}_i\boldsymbol{V}_i) \\
=\ &\sum_{i=1}^{K} \operatorname{rank}(\boldsymbol{S}_i) \\
=\ &K\min\{d_1/K, d_2/K\} = \min\{d_1, d_2, Kr\}\,.
\end{aligned}$$

Case 2: $Kr < \max\{d_1, d_2\}$. We choose full-rank matrices $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K \in \mathbb{R}^{d_2/K\times r}$ and $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_K \in \mathbb{R}^{r\times d_1/K}$, and a full-rank matrix $\boldsymbol{A} \in \mathbb{R}^{K\times K}$. We then have:

$$\begin{aligned}
\operatorname{rank}(\tilde{\boldsymbol{U}}) &= \sum_{i\in[K]} \operatorname{rank}(\boldsymbol{U}_i) = \min\{d_2, Kr\}, \\
\operatorname{rank}(\tilde{\boldsymbol{V}}) &= \sum_{i\in[K]} \operatorname{rank}(\boldsymbol{V}_i) = \min\{d_1, Kr\}, \\
\operatorname{rank}(\boldsymbol{A} \otimes \boldsymbol{I}_r) &= \operatorname{rank}(\boldsymbol{A})\operatorname{rank}(\boldsymbol{I}_r) = Kr.
\end{aligned}$$

By Sylvester's rank inequality:

$$\begin{aligned} & \operatorname{rank}(\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)) \\ \geq\ & \operatorname{rank}(\tilde{\boldsymbol{U}}) + \operatorname{rank}(\boldsymbol{A} \otimes \boldsymbol{I}_r) - Kr \\ =\ & \min\{d_2, Kr\}. \end{aligned}$$

We now want to demonstrate that

$$\operatorname{rank}(\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}}) = \min\{d_1, d_2, Kr\}.$$

Applying Sylvester's rank inequality again:

$$\begin{aligned} & \operatorname{rank}(\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}}) \\ \geq\ & \operatorname{rank}(\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)) + \operatorname{rank}(\tilde{\boldsymbol{V}}) - Kr \\ \geq\ & \min\{d_2, Kr\} + \min\{d_1, Kr\} - Kr. \end{aligned}$$

We can simplify this expression by considering two cases:

If $Kr \geq \min\{d_1, d_2\}$, then $\min\{d_1, Kr\} + \min\{d_2, Kr\} - Kr = \min\{d_1, d_2\} \geq \min\{d_1, d_2, Kr\}$.

If $Kr < \min\{d_1, d_2\}$, then $\min\{d_1, Kr\} + \min\{d_2, Kr\} - Kr = Kr + Kr - Kr = Kr = \min\{d_1, d_2, Kr\}$.

Therefore, in both cases we have $\operatorname{rank}(\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}}) \geq \min\{d_1, d_2, Kr\}$.

Since $\tilde{\boldsymbol{U}} \in \mathbb{R}^{d_2 \times Kr}$ and $\tilde{\boldsymbol{V}} \in \mathbb{R}^{Kr \times d_1}$, we also have $\operatorname{rank}(\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}}) \leq \min\{d_1, d_2, Kr\}$.

Combining the lower and upper bounds, we conclude that

$$\operatorname{rank}(\tilde{\boldsymbol{U}}(\boldsymbol{A} \otimes \boldsymbol{I}_r)\tilde{\boldsymbol{V}}) = \min\{d_1, d_2, Kr\}.$$

Since $r < \min\{d_1, d_2\}$ and $K > 1$, we have $\min\{d_1, d_2, Kr\} > r$. Therefore, there exists a matrix in $\operatorname{im} T_{\text{NM-I}}$ with rank strictly greater than r, which cannot be in $\operatorname{im} T_{\text{LR}}$ because all matrices in $\operatorname{im} T_{\text{LR}}$ have rank at most r. Thus, $\operatorname{im} T_{\text{LR}} \neq \operatorname{im} T_{\text{NM-I}}$.

□

B PROOF OF THEOREM 2

Proof. **Proof of Item i.** Item i follows directly from Theorem 1, Item i.

Proof of Item ii. We begin by proving an auxiliary result. For any matrix $\hat{\boldsymbol{W}}$ with the same dimensions as $\boldsymbol{W}$, we have

$$\begin{aligned} \mathbb{E}\left[\left\|\hat{\boldsymbol{W}}\boldsymbol{x}_{\text{in}} - \boldsymbol{x}_{\text{out}}\right\|_2^2\right] &= \mathbb{E}\left[\left\|\hat{\boldsymbol{W}}\boldsymbol{x}_{\text{in}} - \boldsymbol{W}\boldsymbol{x}_{\text{in}}\right\|_2^2\right] \\ &= \mathbb{E}\left[\operatorname{Tr}\left(\boldsymbol{x}_{\text{in}}^\top \left(\hat{\boldsymbol{W}} - \boldsymbol{W}\right)^\top \left(\hat{\boldsymbol{W}} - \boldsymbol{W}\right)\boldsymbol{x}_{\text{in}}\right)\right] \\ &= \mathbb{E}\left[\operatorname{Tr}\left(\left(\hat{\boldsymbol{W}} - \boldsymbol{W}\right)^\top \left(\hat{\boldsymbol{W}} - \boldsymbol{W}\right)\boldsymbol{x}_{\text{in}}\boldsymbol{x}_{\text{in}}^\top\right)\right] \\ &= \operatorname{Tr}\left(\left(\hat{\boldsymbol{W}} - \boldsymbol{W}\right)^\top \left(\hat{\boldsymbol{W}} - \boldsymbol{W}\right)\mathbb{E}\left[\boldsymbol{x}_{\text{in}}\boldsymbol{x}_{\text{in}}^\top\right]\right) \\ &= \operatorname{Tr}\left(\left(\hat{\boldsymbol{W}} - \boldsymbol{W}\right)^\top \left(\hat{\boldsymbol{W}} - \boldsymbol{W}\right)\right) \\ &= \left\|\hat{\boldsymbol{W}} - \boldsymbol{W}\right\|_F^2, \end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Consequently,

$$R_{\text{LR}} = \inf_{\hat{\boldsymbol{W}} \in \mathbb{R}^{d_1 \times d_2} : \operatorname{rank}(\hat{\boldsymbol{W}}) \leq r} \left\| \hat{\boldsymbol{W}} - \boldsymbol{W} \right\|_F^2,$$
$$R_{\text{NM-I}} = \inf_{\substack{\boldsymbol{U} \in \mathbb{R}^{d_1 \times r}, \\ \boldsymbol{A} \in \mathbb{R}^{K \times K}, \\ \boldsymbol{V} \in \mathbb{R}^{r \times d_2}}} \left\| T_{\text{NM-I}}(\boldsymbol{U}, \boldsymbol{A}, \boldsymbol{V}) - \boldsymbol{W} \right\|_F^2.$$

The equality $R_{\text{LR}} = \sum_{l=r+1}^{\min\{d_1,d_2\}} \sigma_l^2(\boldsymbol{W})$ is a direct application of the Eckart-Young-Mirsky theorem for the Frobenius norm. Now we compute $R_{\text{NM-I}}$. Let a_{ij} denote the (i,j)-entry of the matrix $\boldsymbol{A}$. We have

$$\begin{aligned}
R_{\text{NM-I}} &= \inf_{\substack{\boldsymbol{U} \in \mathbb{R}^{d_1 \times r}, \\ \boldsymbol{A} \in \mathbb{R}^{K \times K}, \\ \boldsymbol{V} \in \mathbb{R}^{r \times d_2}}} \left\| T_{\text{NM-I}}(\boldsymbol{U}, \boldsymbol{A}, \boldsymbol{V}) - \boldsymbol{W} \right\|_F^2 \\
&= \inf_{\substack{\boldsymbol{U} \in \mathbb{R}^{d_1 \times r}, \\ \boldsymbol{A} \in \mathbb{R}^{K \times K}, \\ \boldsymbol{V} \in \mathbb{R}^{r \times d_2}}} \sum_{i,j \in [K]} \left\| a_{ij} \boldsymbol{U}_i \boldsymbol{V}_j - \boldsymbol{W}_{ij} \right\|_F^2 \\
&= \inf_{\substack{\boldsymbol{U} \in \mathbb{R}^{d_1 \times r}, \\ \boldsymbol{V} \in \mathbb{R}^{r \times d_2}}} \sum_{i,j \in [K]} \left\| \frac{\operatorname{Tr}\left(\boldsymbol{W}_{ij}^\top \boldsymbol{U}_i \boldsymbol{V}_j\right)}{\left\| \boldsymbol{U}_i \boldsymbol{V}_j \right\|_F^2} \boldsymbol{U}_i \boldsymbol{V}_j - \boldsymbol{W}_{ij} \right\|_F^2 \\
&\quad \left(\text{choosing } a_{ij} = \frac{\operatorname{Tr}\left(\boldsymbol{W}_{ij}^\top \boldsymbol{U}_i \boldsymbol{V}_j\right)}{\left\| \boldsymbol{U}_i \boldsymbol{V}_j \right\|_F^2} \text{ minimizes } \left\| a_{ij} \boldsymbol{U}_i \boldsymbol{V}_j - \boldsymbol{W}_{ij} \right\|_F^2 \right) \\
&= \inf_{\substack{\boldsymbol{U} \in \mathbb{R}^{d_1 \times r}, \\ \boldsymbol{V} \in \mathbb{R}^{r \times d_2}}} \sum_{i,j \in [K]} \left[\left\| \boldsymbol{W}_{ij} \right\|_F^2 - \frac{\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top \boldsymbol{U}_i \boldsymbol{V}_j\right)}{\left\| \boldsymbol{U}_i \boldsymbol{V}_j \right\|_F^2} \right].
\end{aligned}$$

Next, we upper bound $\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top \boldsymbol{U}_i \boldsymbol{V}_j\right)$:

$$\begin{aligned}
\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top \boldsymbol{U}_i \boldsymbol{V}_j\right) &\leq \left(\sum_{l \in [r]} \sigma_l\left(\boldsymbol{W}_{ij}\right) \sigma_l\left(\boldsymbol{U}_i \boldsymbol{V}_j\right) \right)^2 \\
&\quad \left(\text{by Von Neumann's trace inequality, noting that } \operatorname{rank}\left(\boldsymbol{U}_i \boldsymbol{V}_j\right) \leq r, \text{ so } \sigma_l\left(\boldsymbol{U}_i \boldsymbol{V}_j\right) = 0 \text{ for } l > r \right) \\
&\leq \left(\sum_{l \in [r]} \sigma_l^2\left(\boldsymbol{W}_{ij}\right) \right) \left(\sum_{l \in [r]} \sigma_l^2\left(\boldsymbol{U}_i \boldsymbol{V}_j\right) \right) \\
&\quad \left(\text{by the Cauchy-Schwarz inequality}\right) \\
&= \left(\sum_{l \in [r]} \sigma_l^2\left(\boldsymbol{W}_{ij}\right) \right) \left\| \boldsymbol{U}_i \boldsymbol{V}_j \right\|_F^2.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
R_{\text{NM-I}} &\geq \inf_{\substack{\boldsymbol{U} \in \mathbb{R}^{d_1 \times r}, \\ \boldsymbol{V} \in \mathbb{R}^{r \times d_2}}} \sum_{i,j \in [K]} \left[\left\| \boldsymbol{W}_{ij} \right\|_F^2 - \sum_{l \in [r]} \sigma_l^2\left(\boldsymbol{W}_{ij}\right) \right] \\
&= \sum_{i,j \in [K]} \sum_{l=r+1}^{\min\{d_1,d_2\}/K} \sigma_l^2\left(\boldsymbol{W}_{ij}\right).
\end{aligned}$$

Proof of Item iii. If $E = \varnothing$, then

$$R_{\text{NM-I}} = \inf_{\substack{\boldsymbol{U} \in \mathbb{R}^{d_1 \times r}, \\ \boldsymbol{V} \in \mathbb{R}^{r \times d_2}}} \sum_{(i,j) \in V} \left[\left\| \boldsymbol{W}_{ij} \right\|_F^2 - \frac{\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top \boldsymbol{U}_i \boldsymbol{V}_j\right)}{\left\| \boldsymbol{U}_i \boldsymbol{V}_j \right\|_F^2} \right].$$

Since $E = \varnothing$, for each summand, $\boldsymbol{U}_i$ and $\boldsymbol{V}_j$ can be optimized independently. Let $\boldsymbol{W}_{ij} = \tilde{\boldsymbol{U}}_{ij}\tilde{\boldsymbol{\Sigma}}_{ij}\tilde{\boldsymbol{V}}_{ij}^\top$ be the truncated singular value decomposition of $\boldsymbol{W}_{ij}$, including the r largest singular values, where $\tilde{\boldsymbol{U}}_{ij} \in \mathbb{R}^{d_2/K\times r}$, $\tilde{\boldsymbol{\Sigma}}_{ij} \in \mathbb{R}^{r\times r}$, and $\tilde{\boldsymbol{V}}_{ij} \in \mathbb{R}^{d_1/K\times r}$. Setting $\boldsymbol{U}_i = \tilde{\boldsymbol{U}}_{ij}$ and $\boldsymbol{V}_j = \tilde{\boldsymbol{V}}_{ij}\tilde{\boldsymbol{\Sigma}}_{ij}$, we have

$$R_{\text{NM-I}} \le \sum_{(i,j)\in V}\left[\|\boldsymbol{W}_{ij}\|_F^2 - \frac{\left(\sum_{l\in[r]}\sigma_l^2\left(\boldsymbol{W}_{ij}\right)\right)^2}{\sum_{l\in[r]}\sigma_l^2\left(\boldsymbol{W}_{ij}\right)}\right] = \sum_{(i,j)\in V}\sum_{l=r+1}^{\min\{d_1,d_2\}/K}\sigma_l^2\left(\boldsymbol{W}_{ij}\right).$$

Now consider the case where $d_1 = d_2 = Kr$ and G is a forest. In this case, $\boldsymbol{W}_{ij}$, $\boldsymbol{U}_i$, and $\boldsymbol{V}_j$ are all $r \times r$ square matrices. Since G is a forest, it is a disjoint union of trees. Let $V_1, V_2, \ldots, V_L$ be the vertex sets of these trees. Due to the disjointness of the trees, for each tree V_k, we can optimize the variables $\{\boldsymbol{U}_i, \boldsymbol{V}_j \mid (i,j) \in V_k\}$ independently. We have

$$\begin{aligned}
R_{\text{NM-I}} &= \inf_{\substack{\boldsymbol{U}\in\mathbb{R}^{d_1\times r},\\ \boldsymbol{V}\in\mathbb{R}^{r\times d_2}}}\sum_{(i,j)\in V}\left[\|\boldsymbol{W}_{ij}\|_F^2 - \frac{\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top\boldsymbol{U}_i\boldsymbol{V}_j\right)}{\|\boldsymbol{U}_i\boldsymbol{V}_j\|_F^2}\right]\\
&= \inf_{\substack{\boldsymbol{U}\in\mathbb{R}^{d_1\times r},\\ \boldsymbol{V}\in\mathbb{R}^{r\times d_2}}}\sum_{k\in[L]}\sum_{(i,j)\in V_k}\left[\|\boldsymbol{W}_{ij}\|_F^2 - \frac{\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top\boldsymbol{U}_i\boldsymbol{V}_j\right)}{\|\boldsymbol{U}_i\boldsymbol{V}_j\|_F^2}\right]\\
&= \sum_{k\in[L]}\inf_{\{\boldsymbol{U}_i,\boldsymbol{V}_j\mid(i,j)\in V_k\}}\sum_{(i,j)\in V_k}\left[\|\boldsymbol{W}_{ij}\|_F^2 - \frac{\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top\boldsymbol{U}_i\boldsymbol{V}_j\right)}{\|\boldsymbol{U}_i\boldsymbol{V}_j\|_F^2}\right].
\end{aligned}$$

Consider the minimization problem $\inf_{\{\boldsymbol{U}_i,\boldsymbol{V}_j\mid(i,j)\in V_k\}}\sum_{(i,j)\in V_k}\left[\|\boldsymbol{W}_{ij}\|_F^2 - \frac{\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top\boldsymbol{U}_i\boldsymbol{V}_j\right)}{\|\boldsymbol{U}_i\boldsymbol{V}_j\|_F^2}\right]$. First, assume that $\boldsymbol{W}_{ij}$ is non-singular for all $(i,j) \in V_k$. Pick a root vertex in the tree V_k, say (i_1, j_1). Since $\boldsymbol{W}_{i_1j_1}$ is non-singular, we can choose non-singular matrices $\boldsymbol{U}_{i_1}$ and $\boldsymbol{V}_{j_1}$ such that $\boldsymbol{U}_{i_1}\boldsymbol{V}_{j_1} = \boldsymbol{W}_{i_1j_1}$. For any child vertex (i', j'), if $i' = i_1$, then since $\boldsymbol{W}_{i'j'}$ is also non-singular, we can choose a non-singular matrix $\boldsymbol{V}_{j'} = \boldsymbol{U}_{i_1}^{-1}\boldsymbol{W}_{i'j'}$, so $\boldsymbol{W}_{i'j'} = \boldsymbol{U}_{i_1}\boldsymbol{V}_{j'}$. Similarly, if $j' = j_1$, we can choose a non-singular $\boldsymbol{U}_{i'} = \boldsymbol{W}_{i'j'}\boldsymbol{V}_{j_1}^{-1}$. Inductively, we can do the same for all children vertices of (i_1, j_1). Then we have

$$\inf_{\{\boldsymbol{U}_i,\boldsymbol{V}_j\mid(i,j)\in V_k\}}\sum_{(i,j)\in V_k}\left[\|\boldsymbol{W}_{ij}\|_F^2 - \frac{\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top\boldsymbol{U}_i\boldsymbol{V}_j\right)}{\|\boldsymbol{U}_i\boldsymbol{V}_j\|_F^2}\right] \le \sum_{(i,j)\in V_k}\left[\|\boldsymbol{W}_{ij}\|_F^2 - \frac{\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top\boldsymbol{W}_{ij}\right)}{\|\boldsymbol{W}_{ij}\|_F^2}\right] = 0.$$

If any $\boldsymbol{W}_{ij}$ is singular, since any singular matrix can be approximated by a non-singular matrix in the Frobenius norm, we can choose matrices $\{\boldsymbol{U}_i, \boldsymbol{V}_j \mid (i,j) \in V_k\}$ for their non-singular approximations. Therefore, the infimum $\inf_{\{\boldsymbol{U}_i,\boldsymbol{V}_j\mid(i,j)\in V_k\}}\sum_{(i,j)\in V_k}\left[\|\boldsymbol{W}_{ij}\|_F^2 - \frac{\operatorname{Tr}^2\left(\boldsymbol{W}_{ij}^\top\boldsymbol{U}_i\boldsymbol{V}_j\right)}{\|\boldsymbol{U}_i\boldsymbol{V}_j\|_F^2}\right]$ remains zero even if any $\boldsymbol{W}_{ij}$ is singular. Thus, in this case, $R_{\text{NM-I}} = 0$. □