PARAMETER-SPACE SYMMETRY IN FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient fine-tuning introduces structured trainable parameters into a frozen model, but distinct parameter settings can represent exactly the same function. We characterize the parameter-space symmetries of LoRA, prefix tuning, and prompt tuning in multihead attention, making explicit the redundancies in their parameterizations. These results provide a foundation for studying linear mode connectivity and model merging, where accounting for symmetries helps distinguish functional differences from differences in parameter representation.

1 INTRODUCTION

Modern deep learning models have grown substantially in depth and width, with dense models reaching hundreds of billions of parameters and sparsely activated architectures exceeding a trillion. Rather than training a new model for task, practitioners commonly adapt a general-purpose pretrained model through fine-tuning. At this scale, however, updating and storing a separate full set of weights for each task can be costly (Hu et al., 2021; Li & Liang, 2021). Even with pretrained initialization, full fine-tuning takes place in a high-dimensional, non-convex loss landscape, raising questions about the geometry of the solutions reached by optimization. One way to investigate this geometry is through linear mode connectivity (LMC): solutions from independent training runs can sometimes be joined by a straight path of low loss, particularly after accounting for parameter symmetries (Frankle et al., 2020). Understanding these relationships has informed studies of generalization and motivated methods for combining models in weight space (Izmailov et al., 2018).

Parameter-Efficient Fine-Tuning. Parameter-efficient fine-tuning (PEFT) freezing a pretrained backbone and optimizing a smaller set of task-specific parameters. Prominent methods include LoRA, which learns low-rank weight updates (Hu et al., 2021); prefix tuning, which learns continuous prefixes used by attention layers (Li & Liang, 2021); and prompt tuning, which optimizes continuous input prompts (Lester et al., 2021). These methods can achieve competitive performance while updating far fewer parameters than full fine-tuning and substantially reducing the memory and storage needed for task-specific adaptation.

(Linear) Mode Connectivity. To investigate the geometry of the loss landscape, (Goodfellow et al., 2014; Keskar et al., 2016; Sagun et al., 2018; Venturi et al., 2019) studied the connectivity of independently trained solutions. This line of works revealed *mode connectivity* (MC): the phenomenon that such solutions can be joined by continuous paths along which the loss remains low. A more restrictive form, *linear mode connectivity* (LMC), requires the low-loss path to be a straight-line interpolation (Frankle et al., 2020; Entezari et al., 2021). Because direct interpolation can cross regions of high loss, weight matching methods align independently trained models to reveal low-loss linear paths (Singh & Jaggi, 2020; Ainsworth et al., 2022; Guerrero-Peña et al., 2023). Studying these paths has informed analyses of loss landscape geometry and generalization behavior (Gotmare et al., 2018; Juneja et al., 2023; Lubana et al., 2023), as well as methods for weight averaging and model merging (Izmailov et al., 2018; Wortsman et al., 2022).

Symmetry of Parameter Space. Distinct parameter configurations can implement exactly the same input–output function (Hecht-Nielsen, 1990; Chen et al., 1993; Fefferman & Markel, 1993). For example, permuting neurons within a hidden layer and rearranging their connections accordingly leaves the network function unchanged (Brea et al., 2019; Novak et al., 2018; Bui Thi Mai & Lampert, 2020). This matters for LMC: direct interpolation between two trained models may cross a high-loss region because corresponding neurons are ordered differently. *LMC up to symmetry* considers whether rearranging one model without changing its function reveals a low-loss linear path

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

between the two models (Entezari et al., 2021; Singh & Jaggi, 2020; Ainsworth et al., 2022; Ferbach et al., 2024). Beyond neuron permutations, attention mechanisms and Mixture-of-Experts models possess additional function-preserving transformations (Tran et al., 2025; 2026a;c). Characterizing these transformations expands the ways models can be aligned when studying connectivity and model merging (Theus et al., 2026; Ainsworth et al., 2022).

Alignment Methods. Alignment methods seek correspondences between the internal components of independently trained models before their weights are interpolated. Some search for discrete permutations, using simulated annealing (Entezari et al., 2021) or matching units by their weights or activations (Ainsworth et al., 2022). Weight and activation matching reduce individual matching steps to linear assignment problems (Kuhn, 2010; Jonker & Volgenant, 1987; Crouse, 2016); an alternative uses a straight-through estimator to optimize loss along the interpolated path (Ainsworth et al., 2022). Other approaches use optimal transport or Wasserstein barycenters to construct soft correspondences for model fusion (Singh & Jaggi, 2020; Akash et al., 2022). Sinkhorn re-basin makes the alignment differentiable, allowing it to be optimized for a chosen objective (Guerrero-Peña et al., 2023). These methods can reduce interpolation barriers and reveal LMC that is hidden by a direct comparison of model weights.

Contribution. In this paper, we study the parameter-space symmetries of LoRA, prefix tuning, and prompt tuning in multihead attention. We characterize changes to the trainable parameters that preserve the attention output, with the pretrained weights held fixed. For each method, we examine whether and how distinct parameter settings can represent the same function, making the redundancies of its parameterization explicit. These results provide a foundation for future analyses of linear mode connectivity and model merging in PEFT. Section 2 recalls multihead attention and the symmetry characterization framework. Sections 3, 4, and 5 present the formulations and symmetry results for LoRA, prefix tuning, and prompt tuning, respectively. Section 7 reports the experiments.

2 PRELIMINARIES

Multihead attention. We consider a multihead attention (MHA) module (Vaswani et al., 2017). Let L be the sequence length, d the token dimension, and n the number of heads. Each head has dimension d_h, with $d = nd_h$. Let $\mathbf{x} \in \mathbb{R}^{L\times d}$ be the input token sequence, with x_u as its uth row. Head $i \in [n]$ has query, key, value, and output matrices $Q_i, K_i, V_i, O_i \in \mathbb{R}^{d\times d_h}$. Write $\theta = (Q_i, K_i, V_i, O_i)_{i=1}^n$ for the attention parameters. The projected queries, keys, and values are $\mathbf{x}Q_i$, $\mathbf{x}K_i$, and $\mathbf{x}V_i$. With softmax applied row-wise, the MHA output is

$$\mathrm{MHA}(\mathbf{x};\theta) = \sum_{i=1}^{n} \mathrm{softmax}\big((\mathbf{x}Q_i)(\mathbf{x}K_i)^\top\big)\,\mathbf{x}V_iO_i^\top \in \mathbb{R}^{L\times d}.$$

Here, we absorb the scaling factor $d_h^{-1/2}$ into the query matrices without affecting the analysis.

Rotary positional encoding (RoPE) (Su et al., 2024) uses a fixed orthogonal matrix $R \in \mathbb{R}^{d_h\times d_h}$. For each integer t, set $R_t = R^t$, so that $R_0 = I_{d_h}$ and $R_uR_v^\top = R_{u-v}$. At position u, the query and key rows are multiplied by R_u. For even d_h, the block-diagonal construction is

$$R = \mathrm{diag}\left(\begin{bmatrix}\cos(\varphi_j) & -\sin(\varphi_j)\\ \sin(\varphi_j) & \cos(\varphi_j)\end{bmatrix} \;:\; j \in [d_h/2]\right),$$

where $\varphi_j = 10000^{-2(j-1)/d}$ for $j \in [d_h/2]$. The MHA output with RoPE is

$$\mathrm{MHA}_{\mathrm{RoPE}}(\mathbf{x};\theta) = \sum_{i=1}^{n} \mathrm{softmax}\Big(\big[x_uQ_iR_{u-v}K_i^\top x_v^\top\big]_{u,v\in[L]}\Big)\,\mathbf{x}V_iO_i^\top.$$

Write $\ell_{uv,i} = x_uQ_iR_{u-v}K_i^\top x_v^\top$ for the score from token u to token v in head i. The choice $R = I_{d_h}$ recovers standard attention. We suppress the attention parameters when they are fixed.

Parameter-space symmetry. Following the symmetry characterization framework of Tran et al. (2025; 2026a;b), let $f(\mathbf{x};\theta)$ be a parameterized function with input sequence $\mathbf{x}$ and trainable parameter $\theta \in \Theta$. Two parameters $\theta, \bar\theta$ represent the same function if $f(\mathbf{x};\theta) = f(\mathbf{x};\bar\theta)$ for every input $\mathbf{x}$. A *group action* of a group G on Θ assigns to each $g \in G$ a transformation $\theta \mapsto g\theta$

satisfying $\mathrm{id}_G\theta = \theta$ and $g_1(g_2\theta) = (g_1g_2)\theta$, where id_G is the identity of G. The *orbit* of θ is $G\theta = \{g\theta : g \in G\}$. The group is a *symmetry group* if $f(\cdot; g\theta) = f(\cdot; \theta)$ for every g, θ.

Following the discussion of maximality in Tran et al. (2026b), a symmetry group G is *maximal* if there exists a sufficiently small subset $\mathcal{E} \subsetneq \Theta$ such that

$$f(\cdot;\theta) = f(\cdot;\bar{\theta}) \quad \text{iff} \quad \bar{\theta} = g\theta \text{ for some } g \in G, \qquad \theta, \bar{\theta} \notin \mathcal{E}. \tag{1}$$

Remark. The exceptional subset is usually chosen to be a proper real algebraic variety, namely a proper subset defined by finitely many polynomial equations. Throughout this paper, we use the term *sufficiently small subset*; in all our results, the exceptional subsets can be chosen as proper real algebraic varieties. A property holds *generically* if it holds outside such a subset.

In this paper, two MHA parameter settings represent the same function when their outputs agree for every sequence length $L \geq 1$ and every $\mathbf{x} \in \mathbb{R}^{L\times d}$, with the pretrained weights and R fixed. The following sections specify the trainable parameters and symmetry groups for each fine-tuning method. All proofs are given in Appendix A.

3 Low-rank adaptation

Formulation in MHA. Low-rank adaptation (LoRA) adds trainable low-rank updates to frozen pretrained weights. We consider the query and value adaptation used by Hu et al. (2021), keeping the key and output projections fixed. Let

$$Q^0 = [Q_1^0 \mid \cdots \mid Q_n^0], \qquad V^0 = [V_1^0 \mid \cdots \mid V_n^0] \in \mathbb{R}^{d\times d}$$

be the concatenated pretrained query and value matrices. For $T \in \{Q, V\}$, choose an adaptation rank $1 \leq r_T \leq d$ and a fixed nonzero scaling constant γ_T. The adapted matrices are

$$T(\theta) = T^0 + \gamma_T B_T A_T, \qquad A_T \in \mathbb{R}^{r_T\times d}, \quad B_T \in \mathbb{R}^{d\times r_T}. \tag{2}$$

The trainable parameter is $\theta = (A_Q, B_Q, A_V, B_V)$, with parameter space

$$\Theta_{\mathrm{LoRA}} = \prod_{T\in\{Q,V\}} \left(\mathbb{R}^{r_T\times d} \times \mathbb{R}^{d\times r_T}\right).$$

Partitioning $A_T = [A_{T,1} \mid \cdots \mid A_{T,n}]$ into blocks $A_{T,i} \in \mathbb{R}^{r_T\times d_h}$ gives the adapted projection in head i as $T_i(\theta) = T_i^0 + \gamma_T B_T A_{T,i}$. Thus one low-rank update is shared across the head blocks of each projection. The resulting MHA function is

$$\ell_{uv,i}^{\mathrm{LoRA}}(\theta) = x_u Q_i(\theta) R_{u-v} K_i^\top x_v^\top, \qquad f^{\mathrm{LoRA}}(\mathbf{x};\theta) = \sum_{i=1}^{n} \mathrm{softmax}\big((\ell_{uv,i}^{\mathrm{LoRA}}(\theta))_{u,v=1}^{L}\big)\, \mathbf{x} V_i(\theta) O_i^\top.$$

As in Section 2, $R = I_{d_h}$ gives standard attention.

Group action and symmetry characterization. Denote by $\mathrm{GL}(r)$ the general linear group of invertible real $r \times r$ matrices, with matrix multiplication as its group operation. Let $G_{\mathrm{LoRA}} = \mathrm{GL}(r_Q) \times \mathrm{GL}(r_V)$, with componentwise multiplication. For $g = (G_Q, G_V) \in G_{\mathrm{LoRA}}$, define

$$g\theta = (G_Q A_Q, B_Q G_Q^{-1}, G_V A_V, B_V G_V^{-1}). \tag{3}$$

This action changes the basis within each low-rank factorization and leaves every adapted query and value projection unchanged.

Write $O_{\mathrm{cat}} = [O_1 \mid \cdots \mid O_n] \in \mathbb{R}^{d\times d}$ for the concatenated output matrix. The next theorem characterizes the parameter-space symmetry of the LoRA-tuned MHA module.

Theorem 3.1 (Parameter-space symmetry of LoRA in MHA)**.** *Suppose that each frozen K_i has column rank d_h and O_{cat} is invertible. Fix an orthogonal R. The action in Equation (3) preserves f^{LoRA} for every parameter. Let $\theta, \bar{\theta} \in \Theta_{\mathrm{LoRA}}$ have factors $A_T, B_T, \bar{A}_T, \bar{B}_T$ of rank r_T for $T \in \{Q, V\}$, and suppose that $V_i(\theta) \neq 0$ in every head. Then*

$$f^{\mathrm{LoRA}}(\cdot;\theta) = f^{\mathrm{LoRA}}(\cdot;\bar{\theta}) \quad \textit{iff} \quad \bar{\theta} = g\theta \quad \textit{for a unique } g \in G_{\mathrm{LoRA}}.$$

Consequently, G_{LoRA} is a maximal symmetry group of query/value LoRA in MHA, both with and without RoPE.

The assumptions on the frozen projections hold generically. The excluded trainable parameters have a rank-deficient factor or a zero value projection in some head and form a proper algebraic variety. No rank assumption on the adapted query matrices is needed. In particular, the frozen key and output projections allow equality of attention functions to identify the adapted query and value matrices; equality of their low-rank factors is then determined up to the action in Equation (3).

If LoRA is instead parameterized independently in each head, the same result gives a product of one general linear group for each query and value factor pair. The placement of the updates matters. Jointly adapting query and key projections, or value and output projections, can produce the same attention function with different adapted projections. Thus the maximality statement above is specific to query/value adaptation with frozen keys and outputs.

At a zero-update initialization, for example $B_Q = B_V = 0$, the rank assumptions on the factors fail. Distinct factorizations can then give the same pretrained MHA function without belonging to one G_{LoRA} orbit. Appendix A.1 characterizes these orbits, including the zero updates.

4 PREFIX TUNING

Formulation in MHA. We study the direct key–value parameterization of prefix tuning (Li & Liang, 2021), with the pretrained attention weights fixed. Let $p \geq 1$ be the number of prefix entries. For each head $i \in [n]$, introduce trainable matrices $C_i, D_i \in \mathbb{R}^{p \times d_h}$. Their rth rows, denoted by $C_{r,i}$ and $D_{r,i}$, form one prefix key–value pair. These pairs are independent across heads. The full trainable parameter is $\theta = (C_i, D_i)_{i=1}^n$, with parameter space

$$\Theta_{\mathrm{pre}} = \prod_{i=1}^{n} \left(\mathbb{R}^{p \times d_h} \times \mathbb{R}^{p \times d_h}\right).$$

Each data query attends to the p prefix pairs together with the L projected data pairs. Thus the query matrix remains $\mathbf{x}Q_i$, while the key and value matrices are

$$K_i^{\mathrm{pre}}(\mathbf{x}) = \begin{bmatrix} C_i \\ \mathbf{x}K_i \end{bmatrix}, \qquad V_i^{\mathrm{pre}}(\mathbf{x}) = \begin{bmatrix} D_i \\ \mathbf{x}V_i \end{bmatrix} \in \mathbb{R}^{(p+L) \times d_h}.$$

For standard attention, the output on the data tokens is

$$f^{\mathrm{pre}}(\mathbf{x}; \theta) = \sum_{i=1}^{n} \mathrm{softmax}\big((\mathbf{x}Q_i)K_i^{\mathrm{pre}}(\mathbf{x})^\top\big)\, V_i^{\mathrm{pre}}(\mathbf{x})O_i^\top. \tag{4}$$

The attention matrix in each head has L rows and $p + L$ columns; its normalization includes both prefix and data entries.

For RoPE, assign position $t_r = r - p$ to prefix entry $r \in [p]$, and positions $1, \ldots, L$ to the data tokens. The score from data query u to prefix entry r is

$$a_{ur,i} = x_u Q_i R_{u-t_r} C_{r,i}^\top.$$

The data scores $\ell_{uv,i}$ are defined in Section 2. Let $H_{u,i}^{\mathrm{pre}} \in \mathbb{R}^{1 \times d_h}$ be the output of head i at data token u. Then

$$H_{u,i}^{\mathrm{pre}} = \frac{\sum_{r=1}^{p} e^{a_{ur,i}} D_{r,i} + \sum_{v=1}^{L} e^{\ell_{uv,i}} x_v V_i}{\sum_{r=1}^{p} e^{a_{ur,i}} + \sum_{v=1}^{L} e^{\ell_{uv,i}}}, \qquad f^{\mathrm{pre}}(\mathbf{x}; \theta)_u = \sum_{i=1}^{n} H_{u,i}^{\mathrm{pre}} O_i^\top. \tag{5}$$

Taking $R = I_{d_h}$ gives Equation (4); otherwise this formula defines f^{pre} with RoPE. The output has L data rows. Prefix entries supply keys and values only, without adding query rows. We analyze the matrices C_i, D_i themselves; a separate network generating them has its own parameterization.

Group action and symmetry characterization. Denote by S_p the symmetric group of all permutations of p elements, with composition as its group operation. For $\tau \in S_p$, let P_τ be the permutation matrix defined by $(P_\tau M)_r = M_{\tau^{-1}(r)}$ for a matrix M with p rows. Let $G_{\mathrm{pre}} = (S_p)^n$ be the direct product of n copies of S_p, one for each head, with componentwise composition. For $g = (\tau_1, \ldots, \tau_n) \in G_{\mathrm{pre}}$, the action for standard attention is

$$g\theta = (P_{\tau_i} C_i, P_{\tau_i} D_i)_{i=1}^n. \tag{6}$$

The same permutation is applied to the keys and values within each head, while different heads may use different permutations. With RoPE, define $g\theta = (C'_i, D'_i)_{i=1}^n$ by

$$C'_{r,i} = C_{\tau_i^{-1}(r),i} R_{t_{\tau_i^{-1}(r)} - t_r}, \qquad D'_{r,i} = D_{\tau_i^{-1}(r),i}. \tag{7}$$

This action permutes the rotated keys $C_{r,i}R_{t_r}$ and their corresponding values. It reduces to Equation (6) when $R = I_{d_h}$.

Write $O_{\mathrm{cat}} = [O_1 \mid \cdots \mid O_n] \in \mathbb{R}^{d\times d}$ for the horizontal concatenation of the output matrices.

Theorem 4.1 (Parameter-space symmetry of prefix tuning)**.** *The preceding actions preserve the corresponding prefix attention functions for every parameter. Assume that each Q_i has column rank d_h, each V_i is nonzero, and O_{cat} is invertible. Let $\theta = (C_i, D_i)_{i=1}^n$ and $\bar\theta = (\bar C_i, \bar D_i)_{i=1}^n$. Suppose that the keys $C_{r,i}R_{t_r}$ are pairwise distinct in every head, and likewise for $\bar C_{r,i}R_{t_r}$. Then*

$$f^{\mathrm{pre}}(\cdot;\theta) = f^{\mathrm{pre}}(\cdot;\bar\theta) \quad \textit{iff} \quad \bar\theta = g\theta \textit{ for a unique } g \in (S_p)^n.$$

Here the group acts by the corresponding rule above. Consequently, $(S_p)^n$ is a maximal symmetry group of prefix tuning both with and without RoPE.

These assumptions hold generically in the frozen weights, and distinctness holds generically in the prefix parameters. RoPE changes the action on the prefix keys but preserves the symmetry group. If several rotated keys are equal, only the sum of their corresponding values is determined. The proof and the characterization of these exceptional parameters are given in Appendix A.2.

5 Prompt tuning

Formulation in MHA. Prompt tuning (Lester et al., 2021) learns $p \geq 1$ input vectors $z_r \in \mathbb{R}^{1\times d}$, arranged as the rows of $Z \in \Theta_{\mathrm{prompt}} = \mathbb{R}^{p\times d}$. The pretrained attention weights remain fixed, and these vectors are shared by all heads. The augmented input is $\widetilde{\mathbf{x}} = [Z; \mathbf{x}] \in \mathbb{R}^{(p+L)\times d}$, with the prompt rows first. Applying the frozen projections gives

$$\widetilde{\mathbf{x}}Q_i = \begin{bmatrix} ZQ_i \\ \mathbf{x}Q_i \end{bmatrix}, \qquad \widetilde{\mathbf{x}}K_i = \begin{bmatrix} ZK_i \\ \mathbf{x}K_i \end{bmatrix}, \qquad \widetilde{\mathbf{x}}V_i = \begin{bmatrix} ZV_i \\ \mathbf{x}V_i \end{bmatrix}.$$

Attention produces $p + L$ output rows, but we compare only the outputs on the original data tokens. For a matrix Y with $p + L$ rows, write Y_{data} for its last L rows. The prompt-tuned map is

$$f^{\mathrm{prompt}}(\mathbf{x}; Z) = \mathrm{MHA}(\widetilde{\mathbf{x}})_{\mathrm{data}} = \sum_{i=1}^{n} \mathrm{softmax}\big((\mathbf{x}Q_i)(\widetilde{\mathbf{x}}K_i)^\top\big)\, \widetilde{\mathbf{x}}V_i O_i^\top$$

for standard attention. The queries for these output rows remain $\mathbf{x}Q_i$; the prompt queries ZQ_i enter only the prompt output rows.

With RoPE, assign prompt row r the position $t_r = r - p$, as for prefix tuning, and data row u the position $u \in [L]$. The score from data token u to prompt token r and its associated value are

$$a^{\mathrm{prompt}}_{ur,i} = x_u Q_i R_{u-t_r} K_i^\top z_r^\top, \qquad z_r V_i.$$

Its head output is the normalized sum in Equation (5), with $a_{ur,i}$ replaced by $a^{\mathrm{prompt}}_{ur,i}$ and $D_{r,i}$ by $z_r V_i$. Equivalently, for both standard attention and RoPE,

$$f^{\mathrm{prompt}}(\mathbf{x}; Z) = f^{\mathrm{pre}}\big(\mathbf{x}; (ZK_i, ZV_i)_{i=1}^n\big).$$

All heads derive their prefix pairs from the same prompt rows. We compare MHA outputs with $\mathbf{x}$ and Z as input representations.

Group action and symmetry characterization. Recall that S_p is the symmetric group on p elements and P_τ permutes rows according to τ. For standard attention, the group $G_{\mathrm{prompt}} = S_p$ acts on the prompt parameter space by

$$\tau Z = P_\tau Z, \qquad \tau \in S_p.$$

Since the projected keys and values are ZK_i and ZV_i, one prompt permutation induces the same permutation in every head. With RoPE, define

$$H_R = \{\tau \in S_p : R_{t_{\tau^{-1}(r)}} = R_{t_r} \text{ for all } r \in [p]\}. \tag{8}$$

This subgroup permutes positions with equal rotary matrices. In particular, $H_{I_{d_h}} = S_p$. Write id_{S_p} for the identity permutation.

Theorem 5.1 (Parameter-space symmetry of prompt tuning). *Fix R. The action $Z \mapsto P_\tau Z$, $\tau \in H_R$, preserves f^{prompt} for every prompt parameter. For generic frozen attention weights, there exists a sufficiently small subset $\mathcal{E}_R \subsetneq \Theta_{\mathrm{prompt}}$ such that*

$$f^{\mathrm{prompt}}(\cdot; Z) = f^{\mathrm{prompt}}(\cdot; \bar{Z}) \quad \textit{iff} \quad \bar{Z} = P_\tau Z \textit{ for some } \tau \in H_R, \qquad Z, \bar{Z} \notin \mathcal{E}_R.$$

Thus the maximal symmetry group is S_p for standard attention and H_R for RoPE. If $R_a \neq I_{d_h}$ for every integer a with $0 < |a| \leq p-1$, then $H_R = \{\mathrm{id}_{S_p}\}$ and two generic prompts that represent the same attention function must be equal.

The generic condition includes the assumptions of Theorem 4.1, nonzero K_i, invertibility of $V_{\mathrm{cat}} = [V_1 \mid \cdots \mid V_n] \in \mathbb{R}^{d \times d}$, and a compatibility condition on the frozen key and value matrices. Appendix A.3 gives an explicit rank test and proves that it holds outside a proper algebraic variety. The shared prompt forces permutations across heads to agree. With RoPE, it also prevents general rotations of projected keys while their values remain fixed, restricting the common permutation to H_R.

6 Matching algorithm

Let θ^A and θ^B be independently fine-tuned PEFT parameters sharing the same frozen backbone. We align θ^B to θ^A using transformations that preserve the function at the second endpoint. The resulting parameters $\widetilde{\theta}^B$ can then be used for linear interpolation or model merging. Our matching objectives compare parameters directly and require no calibration data. A small matching cost does not, by itself, guarantee a low interpolation loss barrier.

LoRA factor matching. Theorem 3.1 identifies a $\mathrm{GL}(r_T)$ change of basis for each adapted projection $T \in \{Q, V\}$. For efficient matching, we optimize over its orthogonal subgroup. At each layer, we solve

$$R_T^\star = \arg\min_{R \in \mathrm{O}(r_T)} \left\| A_T^A - R A_T^B \right\|_F^2 + \left\| B_T^A - B_T^B R^\top \right\|_F^2 .$$

Set $M_T = A_T^A (A_T^B)^\top + (B_T^A)^\top B_T^B$. If $M_T = U_T \Sigma_T V_T^\top$ is a singular value decomposition, the orthogonal Procrustes solution is $R_T^\star = U_T V_T^\top$. We replace the factors of model B by

$$\widetilde{A}_T^B = R_T^\star A_T^B, \qquad \widetilde{B}_T^B = B_T^B (R_T^\star)^\top.$$

Their product is unchanged: $\widetilde{B}_T^B \widetilde{A}_T^B = B_T^B A_T^B$. We match each LoRA factor pair separately; the algorithm searches an orthogonal subset of the full symmetry group.

Prefix key–value matching. For directly parameterized prefixes, Theorem 4.1 permits a separate permutation of the key–value pairs in each layer ℓ and independent key–value head i. We form the cost matrix

$$C_{rs}^{(\ell,i)} = \left\| C_{r,\ell,i}^A - C_{s,\ell,i}^B \right\|_2^2 + \left\| D_{r,\ell,i}^A - D_{s,\ell,i}^B \right\|_2^2, \qquad r, s \in [p], \tag{9}$$

and solve $\pi_{\ell,i}^\star = \arg\min_{\pi \in S_p} \sum_{r=1}^p C_{r,\pi(r)}^{(\ell,i)}$ using the Hungarian algorithm (Kuhn, 2010). We apply the resulting permutation jointly to the prefix keys and their associated values. If the prefix keys receive the position-dependent rotations in Equation (7), the key cost instead compares $C_{r,\ell,i}^A R_{t_r}$ with $C_{s,\ell,i}^B R_{t_s}$; moving key s to position r then requires the corresponding inverse position correction. In our Qwen2.5 prefix-tuning implementation, prefix keys enter the key–value cache without a rotary transformation, so we use the unrotated cost in Equation (9).

When a shared row-wise prefix encoder generates the key–value pairs, its input rows cannot be permuted independently for different heads or layers. We concatenate the generated keys and values across all heads and layers into one signature per input row, solve a single assignment problem on these signatures, and apply its permutation to the prefix encoder's input embeddings. The encoder weights remain fixed during matching.

Prompt matching. For input-space prompt tuning, the standard-attention symmetry in Theorem 5.1 applies one shared permutation to the rows of Z across all heads. We use costs $C_{rs}^{\mathrm{prompt}} = \|z_r^A - z_s^B\|_2^2$ and solve the corresponding assignment problem. With RoPE, matches are restricted to permutations in H_R from Equation (8); restrictions imposed by the full model must also be respected. In the causal Qwen2.5 setting, we retain the identity permutation. If an implementation

Table 1: Loss barrier along linear interpolation between independently fine-tuned LoRA models, averaged over 5 seeds and 10 model pairs. **Naive** denotes direct interpolation, while **Aligned** applies parameter alignment. Lower values indicate better linear mode connectivity across tasks.

Method	CIFAR10	CIFAR100	ImageNet	One Billion Word	Wikitext103	CCNEWS
Naive	0.2075 ± 0.0186	0.1058 ± 0.0082	0.2185 ± 0.0657	0.1965 ± 0.0171	0.1803 ± 0.0167	0.0758 ± 0.0016
Aligned	0.01247 ± 0.00242	0.0000 ± 0.0000	0.0021 ± 0.0006	0.0038 ± 0.0012	0.0142 ± 0.0142	0.0142 ± 0.0004

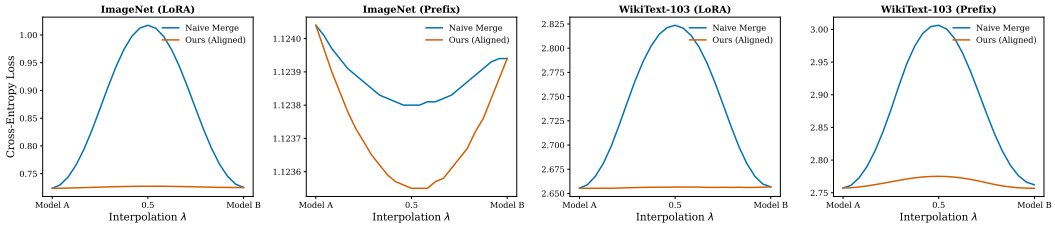


Figure 1: Linear mode connectivity under LoRA and prefix tuning. Interpolation plots for ViT on ImageNet-1K (panels 1 and 2) and Qwen 2.5 on WikiText-103 (panels 3 and 4).

instead uses prompt embeddings to generate prefix key–value pairs through a row-wise encoder, we apply the shared-row prefix procedure above.

Each assignment problem takes $O(p^3)$ time, while LoRA matching requires a small singular value decomposition for each adapted projection. We use the matched parameters for the aligned interpolations in Section 7 and align incoming LoRA adapters to the running merge in Section 7.3.

7 Experimental Results

We study linear mode connectivity, robustness, and model merging under parameter-efficient fine-tuning. For connectivity, we evaluate LoRA, prefix tuning, and prompt tuning with Vision Transformers (ViTs) (Dosovitskiy et al., 2020) on image-classification transfers and Qwen2.5 (Yang et al., 2024) on language-modeling tasks. For model merging, we evaluate LoRA adapters trained with BERT (Devlin et al., 2019) for natural language inference and Llama-3.2-1B (Grattafiori et al., 2024) for GLUE tasks. The robustness and merging studies use LoRA alone. It is the most widely deployed of the three methods. It is also the one for which large collections of independently trained task adapters already exist, which is the situation merging addresses. Experimental configurations and further results are provided in Appendix B, E and D.

7.1 Linear mode connectivity

We evaluate linear mode connectivity in three vision transfer settings. A ViT pretrained on CIFAR-100 (Krizhevsky et al., 2009) is fine-tuned on CIFAR-10 (Krizhevsky et al., 2009); a ViT pretrained on ImageNet-1K is fine-tuned on CIFAR-100; and a ViT pretrained on ImageNet-21K (Ridnik et al., 2021) is fine-tuned on ImageNet-1K (Deng et al., 2009). For language modeling, we start from a pretrained Qwen2.5 checkpoint and fine-tune it separately on WikiText-103 (Merity et al., 2017), One Billion Word (Chelba et al., 2014), and CC-News (Mackenzie et al., 2020).

Figure 1 shows that, after parameter alignment, LoRA and prefix tuning exhibit low-loss linear interpolations in both the ImageNet-21K-to-ImageNet-1K vision setting and Qwen 2.5 fine-tuning on WikiText-103. Table 1 quantifies this effect for LoRA across all six downstream tasks: alignment reduces the loss barrier in every case, with aligned barriers no greater than 0.0142. In contrast, prompt tuning retains a non-negligible interpolation barrier and does not exhibit comparable LMC in our experiments. Detailed results are provided in Appendix C.

7.2 Robustness and generalization

We evaluate robustness and generalization through two complementary experiments, both based on linear mode connectivity under distribution shifts.

Table 2: Per-task validation accuracy (%) after merging five BERT-base LoRA adapters into a single model. **Individual** reports each adapter evaluated on its own task and is the reference point rather than a merging result. Methods are ordered by **Avg**, the mean over the five tasks.

Method	MNLI	SNLI	QNLI	e-SNLI	ANLI	Avg
Individual	82.60	90.30	90.50	89.90	45.70	79.80
LMC-LoRA greedy	75.30	89.40	58.00	86.90	39.40	**69.80**
Task Arithmetic	66.30	75.30	66.50	75.40	36.70	64.00
Ties-Merging	70.70	75.70	60.40	75.80	35.30	63.60
LMC-LoRA dynamic	63.80	75.90	66.10	74.60	36.60	63.40
MagMax	66.60	74.40	60.10	74.90	35.10	62.20
RegMean (a=0.9)	64.20	71.10	63.10	71.40	39.50	61.90
LoRA-LEGO (k=r)	66.90	71.70	52.20	70.40	40.30	60.30
DARE-TA	61.90	68.40	63.60	67.60	35.20	59.30
LoRA-LEGO (k=$2r$)	63.00	66.60	51.40	65.40	40.00	57.30
DARE-Ties	47.80	51.10	59.40	52.00	33.40	48.70
Simple Average	37.10	38.90	55.40	38.60	33.80	40.80

Corruption-based distribution shifts. In the first experiment, we assess robustness to natural test-time distribution shifts induced by common corruptions. We train two models on the same original training dataset using different random seeds, yielding two distinct checkpoints. Connectivity is then evaluated on corrupted test sets, including CIFAR-10-C, CIFAR-100-C, and ImageNet-C. Results for this setting are reported in Tables 6, 7, and 8. Aligned interpolation has higher mean accuracy for most corruptions; Gaussian, shot, and speckle noise on CIFAR-100-C are exceptions.

Class-partitioned distribution shifts. In the second experiment, we construct a controlled distribution shift by partitioning each dataset into two subsets (A) and (B) using a class-conditional sampling ratio (α). Classes are split into two equal groups. For one group, an (α) fraction of samples is assigned to (A) and $(1-\alpha)$ to (B); for the remaining classes, the assignment is reversed. Models are trained independently on (A) and (B), and robustness is evaluated via linear mode connectivity. Results for this experiment are summarized in Table 9, showing that the proposed alignment method consistently improves generalization across datasets and partition ratios.

7.3 Model merging

Model merging combines models fine-tuned from a common backbone on different tasks into one model, without further training. We merge LoRA factors and keep their original inner dimension r.

Merging methods pair A with A and B with B, but the LoRA parameterization is not unique. The change of basis of Equation (3) sends a factor pair (A, B) to $(GA,\ BG^{-1})$ for $G \in \mathrm{GL}(r)$ and leaves the induced update $\Delta W = BA$ unchanged, with the fixed LoRA scaling constant suppressed. Theorem 3.1 shows that this is the whole of the freedom for query/value LoRA in attention. The loss depends on the factors only through BA, so independently trained adapters need not share a basis. Averaging factors then need not average their updates, since the product of the averaged factors carries cross terms B_iA_j for $i \neq j$ (Zhao et al., 2025). The LoRA factor matching of Section 6 removes this freedom without changing any update, and we evaluate it as a preprocessing step for merging. Appendices B and E give the merging algorithm, the configurations and the ablations.

Models and datasets. For natural language inference we freeze BERT-base-cased (Devlin et al., 2019) and train one LoRA (Hu et al., 2021) adapter of rank $r = 16$ per task, with a separate classification head; the five tasks are MNLI (Williams et al., 2018), SNLI (Bowman et al., 2015), QNLI (Wang et al., 2018), e-SNLI (Camburu et al., 2018), and ANLI (Nie et al., 2020). For the language-model domain we freeze Llama-3.2-1B (Grattafiori et al., 2024) and train one adapter of rank $r = 6$ per task, casting classification as generation with a per-task verbalizer over a shared language-model head; seven GLUE tasks (Wang et al., 2018) are used in-domain. In both cases only the LoRA factors on the query and value projections and the task head are trained, so $r_Q = r_V = r$ in the notation of Section 3. Each adapter is initialized with a different seed in both domains.

Table 3: Per-task accuracy (%) after merging seven Llama-3.2-1B LoRA adapters into a single model, evaluated on at most 1000 validation examples per task. SNLI and WNLI are held out and never trained on. **Individual** and **Base** are reference points rather than merging results. Methods are ordered by **Avg IID**.

Method	CoLA	SST-2	MRPC	QQP	MNLI	QNLI	RTE	Avg IID	SNLI	WNLI	Avg OOD
Individual	86.20	95.30	89.71	90.60	88.60	93.60	80.51	89.22	—	—	—
Base (zero-shot)	68.90	53.44	68.38	40.10	37.80	51.10	53.07	53.26	33.70	43.66	38.68
LMC-LoRA greedy	63.50	78.21	74.51	77.50	87.50	69.50	68.95	**74.24**	**84.80**	46.48	**65.64**
RegMean (a=0.9)	76.60	94.04	76.72	79.10	62.10	57.80	69.68	73.72	60.50	56.34	58.42
LoRA-LEGO (k=r)	53.30	93.35	75.00	81.00	61.60	74.00	64.62	71.84	60.70	45.07	52.89
Task Arithmetic	64.80	92.78	65.93	65.00	58.60	76.10	63.54	69.54	56.90	56.34	56.62
MagMax	66.00	83.37	72.06	67.60	60.00	71.70	60.29	68.72	57.90	52.11	55.00
LoRA-LEGO (k=$2r$)	33.70	91.63	74.75	84.00	63.20	64.50	67.87	68.52	63.30	52.11	57.71
LMC-LoRA dynamic	68.50	90.25	61.03	64.50	50.70	70.50	56.32	65.97	42.10	40.85	41.47
DARE-Ties	68.00	75.57	60.78	68.20	53.60	63.20	59.57	64.13	56.00	**59.15**	57.57
DARE-TA	68.90	79.24	66.42	64.50	42.60	69.40	53.79	63.55	35.60	45.07	40.33
Ties-Merging	68.90	53.90	68.14	67.80	42.60	58.50	55.23	59.30	34.70	47.89	41.29
Simple Average	68.90	57.34	68.38	37.70	37.30	51.60	53.07	53.47	33.70	43.66	38.68

Merging methods. Both LMC-LoRA variants apply the LoRA factor matching of Section 6 to each incoming adapter before interpolating. *LMC-LoRA dynamic* merges adapters sequentially, matching each to the running merge, with equal final weights on the matched factor vectors. *LMC-LoRA greedy* follows the same schedule but selects each interpolation coefficient by a grid search on the tasks merged so far. Appendix E.2 varies the spacing of that grid and Appendix B.1 gives the two schedules. We compare against Simple Average, Task Arithmetic (Ilharco et al., 2022), Ties-Merging (Yadav et al., 2023), DARE applied to both Task Arithmetic and Ties (Yu et al., 2024), MagMax (Marczak et al., 2024), RegMean (Jin et al., 2022), and LoRA-LEGO (Zhao et al., 2025) at target ranks $k = r$ and $k = 2r$. RegMean uses activation statistics, while LoRA-LEGO clusters paired rows and columns of the factors.

Evaluation. Several methods carry a scale parameter λ for the merged adapter. We report the setting with the highest average accuracy over the merged tasks on each method's grid. The values swept are listed in Appendix B.3. Methods are compared by that average accuracy, against the average accuracy of the individual adapters on their own tasks as a reference point.

Results. LMC-LoRA greedy has the highest reported merged average in both domains, and LMC-LoRA dynamic improves on Simple Average by a wide margin. The strongest baseline is Task Arithmetic on NLI and RegMean on the language-model tasks, and neither leads in both. LoRA-LEGO performs better at $k = r$ than at $k = 2r$ in both domains. The two variants also differ in cost. LMC-LoRA dynamic uses no data and adds one $r \times r$ singular value decomposition per adapted projection at each step, making it about as cheap as Simple Average. LMC-LoRA greedy needs a number of evaluations quadratic in N and a labelled subset of each task, no larger than the data the swept baselines use to select their scale. Tables 2 and 3 report one merge of one set of adapters and carry no dispersion. Only the LMC-LoRA variants merge sequentially, so only they depend on adapter order, which Appendix E.1 reports as mean and standard deviation over several orders.

8 Conclusion

We characterized the parameter-space symmetries of LoRA, prefix tuning, and prompt tuning in MHA with and without RoPE, and established their maximality under the stated assumptions. These results support the study of linear mode connectivity (LMC) and yield improved model-merging methods. Our experiments recover LMC for LoRA and prefix tuning after alignment, but not for some prompt-tuning configurations. Together with similar findings for Transformer models (Tran et al., 2026a), these observations raise the question of whether LMC always exists up to symmetry. Establishing when LMC exists, and distinguishing its absence from limitations of alignment methods, remains an important direction for rigorous future study.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on our experiments and implementing methods. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology, and interpreting results, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for modifying scientific figures, suggesting experimental parameters, editing software code, editing our research paper to improve readability, and identifying relevant literature. We have reviewed all AI-assisted work. In particular, we discussed and evaluated within our group the AI-generated suggestions for experimental design. We also carefully assessed AI suggestions for method implementation and wrote the final code ourselves. In addition, we reviewed all AI-generated or AI-edited figures and manuscript revisions, and carefully read and verified the literature suggested by AI. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Given the nature of the work, we do not foresee any negative societal and ethical impacts of our work.

REPRODUCIBILITY STATEMENT

Source codes for our experiments are provided in the supplementary materials of the paper. The details of our experimental settings are given in Section 7 and the Appendix. All datasets that we used in the paper are published, and they are easy to access in the Internet.

REFERENCES

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Aditya Kumar Akash, Sixu Li, and Nicolás García Trillos. Wasserstein barycenter-based model fusion and linear mode connectivity of neural networks. *CoRR*, abs/2210.06671, 2022. doi: 10.48550/ARXIV.2210.06671. URL https://doi.org/10.48550/arXiv.2210.06671.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 632–642, 2015.

Johanni Brea, Berfin Simsek, Bernd Illing, and Wulfram Gerstner. Weight-space symmetry in deep networks gives rise to permutation saddles, connected by equal-loss valleys across the loss landscape. *arXiv preprint arXiv:1907.02911*, 2019.

Phuong Bui Thi Mai and Christoph Lampert. Functional vs. parametric equivalence of relu networks. In *8th International Conference on Learning Representations*, 2020.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. *Advances in neural information processing systems*, 31, 2018.

Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie (eds.), *15th Annual Conference of the International Speech Communication Association, INTERSPEECH 2014, Singapore, September 14-18, 2014*, pp. 2635–2639. ISCA, 2014. doi: 10.21437/INTERSPEECH.2014-564. URL https://doi.org/10.21437/Interspeech.2014-564.

An Mei Chen, Haw-minn Lu, and Robert Hecht-Nielsen. On the geometry of feedforward neural network error surfaces. *Neural computation*, 5(6):910–927, 1993.

David Frederic Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Trans. Aerosp. Electron. Syst.*, 52(4):1679–1696, 2016. doi: 10.1109/TAES.2016.140952. URL https://doi.org/10.1109/TAES.2016.140952.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

Charles Fefferman and Scott Markel. Recovering a feed-forward net from its output. In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pp. 335–342. Morgan Kaufmann, 1993. URL http://papers.nips.cc/paper/748-recovering-a-feed-forward-net-from-its-output.

Damien Ferbach, Baptiste Goujaud, Gauthier Gidel, and Aymeric Dieuleveut. Proving linear mode connectivity of neural networks via optimal transport. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3853–3861. PMLR, 2024. URL https://proceedings.mlr.press/v238/ferbach24a.html.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International conference on machine learning*, pp. 3259–3269. PMLR, 2020.

Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. Using mode connectivity for loss landscape analysis. *CoRR*, abs/1806.06977, 2018. URL http://arxiv.org/abs/1806.06977.

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

Fidel A. Guerrero-Peña, Heitor Rapela Medeiros, Thomas Dubail, Masih Aminbeidokhti, Eric Granger, and Marco Pedersoli. Re-basin via implicit sinkhorn differentiation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 20237–20246. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01938. URL https://doi.org/10.1109/CVPR52729.2023.01938.

Robert Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In *Advanced Neural Computers*, pp. 129–135. Elsevier, 1990.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022.

Roy Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987. doi: 10.1007/BF02278710. URL https://doi.org/10.1007/BF02278710.

Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. Linear connectivity reveals generalization strategies. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=hY6M0JHl3uL.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

Harold W. Kuhn. The hungarian method for the assignment problem. In Michael Jünger, Thomas M. Liebling, Denis Naddef, George L. Nemhauser, William R. Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey (eds.), *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, pp. 29–47. Springer, 2010. doi: 10.1007/978-3-540-68279-0_2. URL https://doi.org/10.1007/978-3-540-68279-0_2.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pp. 4582–4597, 2021.

Ekdeep Singh Lubana, Eric J Bigelow, Robert P Dick, David Krueger, and Hidenori Tanaka. Mechanistic mode connectivity. In *International Conference on Machine Learning*, pp. 22965–23004. PMLR, 2023.

Joel Mackenzie, Rodger Benham, Matthias Petri, Johanne R Trippas, J Shane Culpepper, and Alistair Moffat. Cc-news-en: A large english news corpus. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 3077–3084, 2020.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. In *European Conference on Computer Vision*, pp. 379–395. Springer, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4885–4901, 2020.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

Levent Sagun, Utku Evci, V. Ugur Güney, Yann N. Dauphin, and Léon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJO1_M0Lf.

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

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Alexander Theus, Alessandro Cabodi, Sotiris Anagnostidis, Antonio Orvieto, Sidak Pal Singh, and Valentina Boeva. Generalized linear mode connectivity for transformers. *Advances in Neural Information Processing Systems*, 38:136197–136225, 2026.

Hoang V. Tran, Thieu Vo, An Nguyen The, Tho Tran Huu, Minh-Khoi Nguyen-Nhat, Thanh Tran, Duy-Tung Pham, and Tan Minh Nguyen. Equivariant neural functional networks for transformers. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=uBai0ukstY.

Viet-Hoang Tran, Vinh Khanh Bui, Van-Hoan Trinh, Tan Lai Ngoc, and Tan M Nguyen. Functional equivalence in attention: A comprehensive study with applications to linear mode connectivity. *arXiv preprint arXiv:2606.17830*, 2026a.

Viet-Hoang Tran, An Nguyen, Benoît Guérand, Thieu N. Vo, and Tan M. Nguyen. Quasi-equivariant metanetworks. In *International Conference on Learning Representations*, 2026b.

Viet-Hoang Tran, Van Hoan Trinh, Khanh-Vinh Bui, and Tan Nguyen. On linear mode connectivity of mixture-of-experts architectures. *Advances in Neural Information Processing Systems*, 38: 171929–172014, 2026c.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Luca Venturi, Afonso S Bandeira, and Joan Bruna. Spurious valleys in one-hidden-layer neural network optimization landscapes. *Journal of Machine Learning Research*, 20:133, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 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)*, pp. 1112–1122, 2018.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. Pmlr, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Ziyu Zhao, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, Fei Wu, et al. Merging loras like playing lego: Pushing the modularity of lora to extremes through rank-wise clustering. In *International Conference on Learning Representations*, volume 2025, pp. 72896–72913, 2025.

APPENDIX CONTENTS

A Appendix **15**

A.1 LoRA 15

A.2 Prefix tuning 17

A.3 Prompt tuning 18

B Experimental details **19**

B.1 Merging algorithms 19

B.2 Linear mode connectivity setup 20

B.3 Model merging setup 21

C Linear mode connectivity **23**

D Robustness and generalization **26**

E Model merging **28**

E.1 Sensitivity to merge order 28

E.2 Resolution of the greedy grid 28

A APPENDIX

A.1 LORA

We first show that the frozen key and output matrices make the adapted query and value matrices identifiable from the attention function.

Lemma A.1 (Identifiability with frozen keys and outputs)**.** *Fix an orthogonal rotary matrix R, key matrices K_i of column rank d_h, and an invertible matrix $O_{\mathrm{cat}} = [O_1 \mid \cdots \mid O_n]$. Suppose $V_i \neq 0$ for every head. Two MHA modules with respective query and value matrices $(Q_i, V_i)_{i=1}^n$ and $(\bar{Q}_i, \bar{V}_i)_{i=1}^n$ give the same outputs for every input if and only if $Q_i = \bar{Q}_i$ and $V_i = \bar{V}_i$ for every i.*

Proof. Let $H_i(\mathbf{x})$ denote the output of head i before its output projection. The MHA output is $[H_1(\mathbf{x}) \mid \cdots \mid H_n(\mathbf{x})]O_{\mathrm{cat}}^\top$. Invertibility of O_{cat} therefore makes equality of the MHA outputs equivalent to $H_i(\mathbf{x}) = \bar{H}_i(\mathbf{x})$ for every head and input. For a single-token input $\mathbf{x} = [x]$, the attention weight is one, giving $xV_i = x\bar{V}_i$ for every $x \in \mathbb{R}^{1\times d}$. Hence $V_i = \bar{V}_i$.

Now take the two-token input $\mathbf{x} = [x; y]$ at positions $1, 2$. The first output row of head i is

$$\begin{aligned} H_i(\mathbf{x})_1 &= yV_i + \sigma(\delta_i(x,y))(x-y)V_i, \\ \delta_i(x,y) &= xQ_iK_i^\top x^\top - xQ_iR_{-1}K_i^\top y^\top, \end{aligned} \tag{10}$$

where $\sigma(t) = e^t/(1+e^t)$ is the strictly increasing logistic function. Because $V_i \neq 0$, the set of pairs (x, y) with $(x-y)V_i \neq 0$ is dense. Equality of the head outputs and injectivity of σ give $\delta_i(x,y) = \bar{\delta}_i(x,y)$ on this set, and continuity extends the equality to every (x, y). Writing $\Delta Q_i = Q_i - \bar{Q}_i$, we obtain

$$x\Delta Q_iK_i^\top x^\top - x\Delta Q_iR_{-1}K_i^\top y^\top = 0 \qquad \text{for all } x, y.$$

Setting $y = 0$ and subtracting that identity shows $x\Delta Q_iR_{-1}K_i^\top y^\top = 0$ for all x, y, so $\Delta Q_iR_{-1}K_i^\top = 0$. Since R_{-1} is invertible and $K_i^\top$ has row rank d_h, this implies $\Delta Q_i = 0$. The converse follows directly from the attention formula. The argument includes standard attention by taking $R = I_{d_h}$. □

Proof of Theorem 3.1. At each adapted matrix $T \in \{Q, V\}$, the action in Equation (3) leaves the product $B_T A_T$ unchanged: $(B_T G_T^{-1})(G_T A_T) = B_T A_T$. It thus preserves the complete attention function for every choice of factors. Conversely, Lemma A.1 shows that equality of the MHA outputs determines the adapted query and value matrices. The pretrained matrices and nonzero scaling constants are the same for both parameters, so $B_T A_T = \bar{B}_T \bar{A}_T$ for $T \in \{Q, V\}$.

Fix either T, and abbreviate its factors by A, B and its rank by r. Full row rank of A and full column rank of B give $\operatorname{im}(BA) = \operatorname{im} B$ and $\operatorname{rank}(BA) = r$. Thus B and $\bar{B}$ are bases of the same column space, so $\bar{B} = BH$ for some invertible $r \times r$ matrix H. The equality $BA = BH\bar{A}$ and injectivity of B imply $\bar{A} = H^{-1}A$. Taking $G = H^{-1}$ gives the stated action. If $GA = \widetilde{G}A$, full row rank of A gives $G = \widetilde{G}$. Applying this argument to both adapted matrices proves the equivalence and uniqueness.

The exceptional subset $\mathcal{E}$ consists of rank-deficient factors and parameters for which some adapted value block V_i vanishes. Each condition is algebraic: rank deficiency is defined by the maximal minors, and $V_i = 0$ by its entries. Each is proper. In particular, for every head the polynomial matrix $V_i^0 + \gamma_V B_V A_{V,i}$ is not identically zero. A finite union of proper real algebraic sets remains proper. Thus $\mathcal{E}$ is a sufficiently small subset as specified in Section 2, and the product group is maximal for the LoRA-tuned MHA function in the sense of Equation (1). □

The next result describes the orbits when one or both factors are rank deficient. We state the result for arbitrary matrix dimensions, with $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}} \times r}$, and $1 \le r \le \min(d_{\text{in}}, d_{\text{out}})$; the two adapted matrices above have $d_{\text{in}} = d_{\text{out}} = d$. It distinguishes equality of updates from equivalence under a change of basis. Applied separately to (A_Q, B_Q) and (A_V, B_V), it classifies the G_{LoRA} orbits. Under the frozen-projection assumptions of Theorem 3.1, if every adapted V_i is nonzero, Lemma A.1 identifies equality of MHA functions with equality of these two updates, including for rank-deficient factors.

Proposition A.2 (Orbits of arbitrary LoRA factorizations)**.** *Two pairs (A, B) and $(\bar{A}, \bar{B})$ belong to the same $\mathrm{GL}(r)$ orbit if and only if*

$$BA = \bar{B}\bar{A}, \qquad \ker A = \ker \bar{A}, \qquad \operatorname{im} B = \operatorname{im} \bar{B}. \tag{11}$$

More precisely, fix $M \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ of rank $k \le r$. The orbits in $\{(A, B) : BA = M\}$ are in one-to-one correspondence with pairs of subspaces (K, L) satisfying

$$\begin{gathered} K \subseteq \ker M, \qquad \operatorname{im} M \subseteq L, \\ a = d_{\text{in}} - \dim K \le r, \qquad b = \dim L \le r, \qquad a + b - k \le r, \end{gathered} \tag{12}$$

where $K = \ker A$, $L = \operatorname{im} B$, $a = \operatorname{rank} A$, and $b = \operatorname{rank} B$.

Proof. All three quantities in Equation (11) are unchanged by $(A, B) \mapsto (GA, BG^{-1})$. To prove sufficiency, put $M = BA$ and write $E = \mathbb{R}^r$, $U = \operatorname{im} A$, and $\bar{U} = \operatorname{im} \bar{A}$. Equality of the kernels makes

$$g_U : U \longrightarrow \bar{U}, \qquad Av \longmapsto \bar{A}v$$

a well-defined isomorphism. Equality of the products gives $\bar{B} g_U = B|_U$, so g_U maps $U \cap \ker B$ onto $\bar{U} \cap \ker \bar{B}$.

Choose a complement L_0 of $\operatorname{im} M$ in the common image $L = \operatorname{im} B = \operatorname{im} \bar{B}$. Choose subspaces $T, \bar{T} \subseteq E$ such that $B|_T : T \to L_0$ and $\bar{B}|_{\bar{T}} : \bar{T} \to L_0$ are isomorphisms. Let N complement $U \cap \ker B$ in $\ker B$, and let $\bar{N}$ complement $\bar{U} \cap \ker \bar{B}$ in $\ker \bar{B}$. Then

$$E = U \oplus N \oplus T = \bar{U} \oplus \bar{N} \oplus \bar{T}.$$

Indeed, applying B to a putative relation separates the images in $\operatorname{im} M$ and L_0, after which the choice of N gives directness; the dimensions sum to r. The barred decomposition follows identically. Extend g_U to $G : E \to E$ by any isomorphism $N \to \bar{N}$ and by $(\bar{B}|_{\bar{T}})^{-1} B|_T$ on T. This extension is invertible and satisfies $GA = \bar{A}$ and $\bar{B}G = B$, proving Equation (11).

For feasibility, $BA = M$ implies $\ker A \subseteq \ker M$ and $\operatorname{im} M \subseteq \operatorname{im} B$. Moreover, $\dim(U \cap \ker B) = a - k \le r - b$, proving Equation (12). Conversely, given such K, L, choose $L = \operatorname{im} M \oplus L_0$ and identify

$$\mathbb{R}^r \cong (\mathbb{R}^{d_{\text{in}}}/K) \oplus L_0 \oplus \mathbb{R}^{r-a-b+k}.$$

The maps

$$A(v) = (v + K, 0, 0), \qquad B(v + K, t, z) = Mv + t$$

are well defined because $K \subseteq \ker M$. They have product M, kernel $\ker A = K$, and image $\operatorname{im} B = L$. The first part shows that these invariants specify exactly one orbit. □

Proposition A.3 (Zero LoRA updates)**.** *The set of factors producing the zero update is*

$$\{(A, B) : BA = 0\} = \{(A, B) : \operatorname{im} A \subseteq \ker B\}. \tag{13}$$

Ranks $a = \operatorname{rank} A$ and $b = \operatorname{rank} B$ occur in this set exactly when $a + b \leq r$, subject to the matrix dimensions. If A_0 has row rank r, the orbit of $(A_0, 0)$ is $\{(GA_0, 0) : G \in \mathrm{GL}(r)\}$, a proper subset of this set.

Proof. The equality $BA = 0$ is equivalent to $\operatorname{im} A \subseteq \ker B$. With $a = \operatorname{rank} A$ and $b = \operatorname{rank} B$, this implies $a \leq r - b$. Conversely, each feasible pair of ranks with $a + b \leq r$ is realized by the construction in Proposition A.2 with $M = 0$. That proposition also proves that the subspaces $\ker A$ and $\operatorname{im} B$, rather than their dimensions alone, determine the orbit within the zero fiber. In particular, when A has full row rank and $B = 0$, its orbit is

$$\{(\bar{A}, 0) : \ker \bar{A} = \ker A\}.$$

If $d_{\text{in}} > r$, different full-row-rank choices of A can therefore have the same zero update without belonging to the same orbit. For every $r \geq 1$, the orbit is a proper subset of the zero-update set because $(0, 0)$ does not belong to it. □

A.2 Prefix tuning

We use the fixed rotary matrix R from Section 2, taking $R = I_{d_h}$ for attention without positional encoding. Write $t_r = r - p$ and $O_{\text{cat}} = [O_1 \mid \cdots \mid O_n]$. We assume that every Q_i has column rank d_h, every V_i is nonzero, and O_{cat} is invertible.

Recovery of the keys and values. If H_i is the output of head i before its output projection, then

$$\sum_{i=1}^{n} H_i O_i^\top = [H_1 \mid \cdots \mid H_n] O_{\text{cat}}^\top.$$

Thus equality of attention outputs implies equality in every head by right multiplication with $O_{\text{cat}}^{-\top}$. Fix one head and write its prefix keys after rotation and corresponding values as

$$(a_r, b_r) = (C_{r,i} R_{t_r}, D_{r,i}), \qquad r \in [p].$$

For a row query $q \in \mathbb{R}^{1 \times d_h}$, define

$$S(q) = \sum_{r=1}^{p} e^{q a_r^\top}, \qquad T(q) = \sum_{r=1}^{p} e^{q a_r^\top} b_r, \tag{14}$$

and define $\bar{S}, \bar{T}$ in the same way for another parameter setting.

At the first data position, the query after rotation is $x_1 Q_i R_1$. Since $\Gamma_i = Q_i R_1$ has full column rank, choose $J_i \in \mathbb{R}^{d_h \times d}$ with $J_i \Gamma_i = I_{d_h}$ and set $x_1(q) = q J_i$. Take the data sequence to be $(x_1(q), y, 0, \ldots, 0)$ with ℓ zero rows. The first two data tokens contribute fixed functions $\delta(q) > 0$ and $\nu(q)$ to the denominator and numerator. Every additional zero token contributes one to the denominator and zero to the numerator, including under RoPE. Consequently, equality of the outputs for every input and sequence length gives

$$\frac{\nu(q) + T(q)}{\delta(q) + S(q) + \ell} = \frac{\nu(q) + \bar{T}(q)}{\delta(q) + \bar{S}(q) + \ell} \qquad (q \in \mathbb{R}^{1 \times d_h},\ \ell = 0, 1, \ldots). \tag{15}$$

Cross multiplication and comparison of the coefficients of ℓ yield $T = \bar{T}$ and $(\nu + T)(\bar{S} - S) = 0$. Choose y such that $y V_i + \sum_r b_r \neq 0$, which is possible because $V_i \neq 0$. At $q = 0$ this is exactly $(\nu + T)(0)$. Continuity therefore gives $S = \bar{S}$ on a nonempty open neighborhood.

For completeness, distinct vectors $a_1, \dots, a_m$ give linearly independent functions $q \mapsto e^{qa_j^\top}$ on every nonempty open set. Choose a row v for which all $va_j^\top$ are distinct, restrict a proposed dependence to $q = q_0 + sv$, and differentiate at $s = 0$ through order $m - 1$. The resulting Vandermonde matrix is invertible. Apply this fact first to $S - \bar{S}$ and then to $T - \bar{T}$. The first equality identifies every key and its multiplicity; the second identifies the sum of the values attached to each key. In particular, if the keys of the first parameter setting are pairwise distinct, so are those of the second, and there is a unique $\tau \in S_p$ such that

$$(\bar{a}_r, \bar{b}_r) = (a_{\tau^{-1}(r)}, b_{\tau^{-1}(r)}) \qquad (r \in [p]). \tag{16}$$

This also describes the exceptional case of repeated keys: only the sum of their associated values is determined.

Proof of Theorem 4.1. Apply Equation (16) independently in every head. With $s = \tau_i^{-1}(r)$, it gives

$$\bar{C}_{r,i} R_{t_r} = C_{s,i} R_{t_s}, \qquad \bar{D}_{r,i} = D_{s,i},$$

or equivalently

$$\bar{C}_{r,i} = C_{s,i} R_{t_s - t_r}, \qquad \bar{D}_{r,i} = D_{s,i}. \tag{17}$$

Conversely, this transformation permutes the rotated keys and corresponding values and hence leaves every attention numerator and denominator unchanged. The key action is the conjugate of the ordinary permutation action by the invertible rowwise map $(\mathcal{D}_R C_i)_r = C_{r,i} R_{t_r}$; the value action is ordinary row permutation. It therefore defines an action of $(S_p)^n$. Taking $R = I_{d_h}$ gives the ordinary row permutation action.

Let $\mathcal{E}_{\mathrm{pre}} \subset \Theta_{\mathrm{pre}}$ consist of the prefix parameters for which two rotated keys in a head coincide. It is the zero set of the nonzero polynomial

$$\prod_{i=1}^{n} \prod_{r<s} \|C_{r,i} R_{t_r} - C_{s,i} R_{t_s}\|_2^2.$$

Thus $\mathcal{E}_{\mathrm{pre}}$ is a proper real algebraic variety and hence a sufficiently small subset in Equation (1). It is fixed independently of the two parameters being compared. Outside it, two parameters give the same attention function exactly when they lie in the same group orbit, proving maximality. □

A.3 Prompt tuning

The key and value recovery argument in Appendix A.2 applies with $(a_r, b_r) = (z_r K_i R_{t_r}, z_r V_i)$. In addition to the assumptions there, we require every K_i to be nonzero and $V_{\mathrm{cat}} = [V_1 \mid \cdots \mid V_n]$ to be invertible. We establish the further generic condition below.

Compatibility of the head permutations. Fix $\boldsymbol{\tau} = (\tau_1, \dots, \tau_n) \in (S_p)^n$ and put $s_i(r) = \tau_i^{-1}(r)$. Let $\mathcal{E}_{\boldsymbol{\tau}}^R$ be the set of prompts Z for which there is a $\bar{Z}$ satisfying

$$\bar{z}_r V_i = z_{s_i(r)} V_i, \qquad \bar{z}_r K_i R_{t_r} = z_{s_i(r)} K_i R_{t_{s_i(r)}} \qquad (i \in [n],\ r \in [p]). \tag{18}$$

For $Z \in \mathbb{R}^{p \times d}$, let $\mathrm{vec}(Z) \in \mathbb{R}^{pd}$ be the column vector obtained by stacking the columns of Z in order. Stacking the scalar equations in Equation (18) gives $M_R \,\mathrm{vec}(\bar{Z}) = N_{\boldsymbol{\tau}}^R \,\mathrm{vec}(Z)$, where $M_R, N_{\boldsymbol{\tau}}^R \in \mathbb{R}^{2npd_h \times pd}$ have entries polynomial in the frozen projections. The value equations and invertibility of V_{cat} imply that M_R is injective, so $\mathrm{rank}(M_R) = pd$. Thus $\mathcal{E}_{\boldsymbol{\tau}}^R$ is a linear subspace, and

$$\mathcal{E}_{\boldsymbol{\tau}}^R \subsetneq \mathbb{R}^{p \times d} \quad \text{iff} \quad \mathrm{rank}[M_R \mid N_{\boldsymbol{\tau}}^R] > pd. \tag{19}$$

Indeed, the subspace is the whole domain precisely when $\mathrm{im}(N_{\boldsymbol{\tau}}^R) \subseteq \mathrm{im}(M_R)$.

Let

$$H_R = \{\tau \in S_p : R_{t_{\tau^{-1}(r)}} = R_{t_r} \text{ for all } r\}, \qquad \Delta H_R = \{(\tau, \dots, \tau) : \tau \in H_R\}.$$

We show that, for generic frozen projections, Equation (19) holds simultaneously for all $\boldsymbol{\tau} \notin \Delta H_R$. Split each row z into n blocks $z^{(j)} \in \mathbb{R}^{1 \times d_h}$, and let $E_j \in \mathbb{R}^{d \times d_h}$ select block j. Consider the explicit choice

$$V_j = Q_j = O_j = E_j, \qquad K_i = \begin{bmatrix} \lambda_{i1} I_{d_h} \\ \vdots \\ \lambda_{in} I_{d_h} \end{bmatrix}, \qquad \lambda_{ij} \neq 0. \tag{20}$$

All required ranks hold and $V_{\mathrm{cat}} = O_{\mathrm{cat}} = I_d$. The value equations in Equation (18) force $\bar{z}_r^{(j)} = z_{s_j(r)}^{(j)}$. Its key equations then become

$$\sum_{j=1}^{n} \lambda_{ij} z_{s_j(r)}^{(j)} R_{t_r} = \sum_{j=1}^{n} \lambda_{ij} z_{s_i(r)}^{(j)} R_{t_{s_i(r)}}. \tag{21}$$

If the head permutations are not all equal, choose i, j, r with $s_j(r) \neq s_i(r)$. The independent variable $z_{s_j(r)}^{(j)}$ has nonzero coefficient on the left of Equation (21) and is absent from the right. Hence these equations impose a nonzero linear constraint on Z. If the head permutations are equal but not in H_R, choose r with $R_{t_r} \neq R_{t_{s_i(r)}}$. Equation (21) then contains the nonzero constraint

$$\left(\sum_{j=1}^{n} \lambda_{ij} z_{s_i(r)}^{(j)} \right) (R_{t_r} - R_{t_{s_i(r)}}) = 0.$$

The matrix difference need only be nonzero; it need not be invertible. Thus every excluded tuple gives a proper subspace at the same choice Equation (20).

For each of the finitely many tuples outside ΔH_R, select a $(pd+1)$-minor in Equation (19) that is nonzero at Equation (20). Multiply these minors by $\det O_{\mathrm{cat}}$, $\det V_{\mathrm{cat}}$, and full-rank minors of Q_i, K_i, V_i that are nonzero there. The product is a nonzero polynomial in the frozen weights. All required conditions hold whenever this polynomial is nonzero, hence outside one fixed proper real algebraic variety in the frozen weights. This proves the generic assertion for every fixed R, including $R = I_{d_h}$.

Proof of Theorem 5.1. For any $\tau \in H_R$, replacing Z by $P_\tau Z$ simultaneously permutes the rotated keys and corresponding values in every head. This is an exact symmetry. Conversely, choose frozen weights for which the polynomial just constructed is nonzero. Define the exceptional subset of prompt space by

$$\mathcal{E}_R = \bigcup_{\substack{i \in [n] \\ 1 \leq r < s \leq p}} \{Z : z_r K_i R_{t_r} = z_s K_i R_{t_s}\} \cup \bigcup_{\boldsymbol{\tau} \notin \Delta H_R} \mathcal{E}_{\boldsymbol{\tau}}^R.$$

Each subspace in the first family is proper because K_i has positive rank, and the other subspaces are proper by Equation (19). Their finite union is a proper real algebraic variety: each subspace is the zero set of a nonzero sum of squares of linear equations, and the product of these polynomials defines their union. Therefore $\mathcal{E}_R$ is a sufficiently small subset as required by Theorem 5.1 and Equation (1). It depends only on the fixed weights and R.

If $Z \notin \mathcal{E}_R$ and $\bar{Z}$ gives the same function, recovery of the keys and values yields Equation (18) for some tuple $\boldsymbol{\tau}$. By the construction of the exceptional set this tuple belongs to ΔH_R. Hence there is a single $\tau \in H_R$ such that, with $s = \tau^{-1}(r)$, $(\bar{z}_r - z_s)V_i = 0$ for every i. Concatenation and invertibility of V_{cat} give $\bar{z}_r = z_s$, so $\bar{Z} = P_\tau Z$. This proves the orbit characterization. When $R = I_{d_h}$, one has $H_R = S_p$. For general R, H_R consists of permutations within classes of equal rotary matrices and is trivial when $R_a \neq I_{d_h}$ for every integer a with $0 < |a| \leq p-1$. □

B Experimental details

B.1 Merging algorithms

This subsection states the two merging methods introduced in Section 7.3. Throughout, the pre-trained weights stay frozen and adapter $i \in \{1, \ldots, N\}$ contributes, for each adapted projection, a pair of LoRA factors

$$A_i \in \mathbb{R}^{r \times d_{\mathrm{in}}}, \qquad B_i \in \mathbb{R}^{d_{\mathrm{out}} \times r}, \qquad \Delta W_i = B_i A_i. \tag{22}$$

These are the factors of Equation (2) with the fixed scaling constant suppressed, stored in the linear-layer convention of the implementation, so that ΔW_i acts on a row input as $x \Delta W_i^\top$ rather than as $x B_i A_i$. The change of basis of Equation (3) takes the same form in either convention and is

applied independently to each adapted projection. We write θ_i for the concatenation of (A_i, B_i) over all adapted layers. Both methods return a single parameter setting of the same shape and neither changes the frozen backbone.

Merging on the factors. A sum of updates, each of rank at most r, may have rank greater than r. Averaging factors of inner dimension r retains this dimension, but generally differs from averaging the updates (Zhao et al., 2025). Multiplying each averaged factor by λ gives

$$\Delta W_{\text{avg}} \;=\; \Big(\tfrac{\lambda}{N}\textstyle\sum_i B_i\Big)\Big(\tfrac{\lambda}{N}\textstyle\sum_j A_j\Big) \;=\; \tfrac{\lambda^2}{N^2}\Big[\textstyle\sum_i B_i A_i \;+\; \sum_{i\neq j} B_i A_j\Big], \tag{23}$$

The cross terms need not vanish, so this identity alone does not imply a fixed reduction in update norm.

The LMC-LoRA variants. Write $\Pi(\theta, \theta')$ for the adapter θ' matched to θ by the LoRA factor matching of Section 6, applied to each LoRA factor pair; tensors outside the LoRA factors, such as classification heads, are carried over untouched. Both variants share one sequential schedule. Starting from $\theta^{(1)} = \theta_1$, each adapter is aligned to the *running merge* rather than to a fixed anchor, and then interpolated into it:

$$\theta^{(i)} \;=\; (1-\lambda_i)\,\theta^{(i-1)} \;+\; \lambda_i\,\Pi\big(\theta^{(i-1)}, \theta_i\big), \qquad i = 2, \ldots, N. \tag{24}$$

The variants differ only in how λ_i is set.

- *LMC-LoRA dynamic*: $\lambda_i = 1/i$, the running mean. Each matched factor vector has final weight $1/N$. This is an average of matched factors, not of the updates $B_i A_i$, and the matchings can depend on the merge order.
- *LMC-LoRA greedy*: λ_i is chosen by search. At step i, each candidate in the grid $\{0.0, 0.1, \ldots, 1.0\}$ is substituted into Equation (24) and the resulting model is scored on each of the tasks merged so far, $\{1, \ldots, i\}$, under that task's own head or verbalizer; the λ maximizing the mean of those accuracies is kept and the step is committed. The scoring set is a 2% random subset of each task's validation split, capped at 1000 examples, for NLI and the full evaluation subset for the language model.

Neither variant takes a scale parameter from outside. Dynamic fixes its weights by the schedule, and greedy fixes them by the search above.

B.2 Linear mode connectivity setup

Pretrained backbones and tasks. We evaluate three vision transfer settings: CIFAR-100 to CIFAR-10, ImageNet-1K to CIFAR-100, and ImageNet-21K to ImageNet-1K. All use pretrained Vision Transformer (ViT) backbones. The CIFAR-100 checkpoint has a hidden dimension of 256 and was pretrained for 50 epochs with AdamW, a batch size of 256, an initial learning rate of 2.5×10^{-4}, five warm-up epochs, and cosine learning-rate decay. We use the checkpoint selected at epoch 41. The remaining vision settings start from ViT checkpoints pretrained on ImageNet-1K or ImageNet-21K, respectively. For language modeling, we fine-tune a pretrained Qwen 2.5 model on WikiText-103, One Billion Word, and CC-News. In every setting, the pretrained backbone is frozen while the method-specific adaptation parameters and any required task-specific output head are trained.

Vision fine-tuning. We evaluate LoRA with rank $r = 16$, scaling parameter $\alpha = 32$, and dropout 0.1; prefix tuning with 50 prefix tokens and dropout 0.1; and visual prompt tuning with 50 prompts, prompt depth -1, and dropout 0.1. These method-specific configurations are shared across the CIFAR-10 and CIFAR-100 transfer tasks. For CIFAR-100 to CIFAR-10 and ImageNet-1K to CIFAR-100, we fine-tune for 25 epochs using SGD, an initial learning rate of 5×10^{-2}, cosine decay to 10^{-6}, and no warm-up. The batch size is 256, except for the CIFAR-10 LoRA data-split experiment, which uses 128. Input images have resolution 32×32 for CIFAR-100 to CIFAR-10 and 224×224 for ImageNet-1K to CIFAR-100. For ImageNet-21K to ImageNet-1K, we fine-tune for 50 epochs with SGD, a batch size of 512, and an initial learning rate of 10^{-2}. The learning rate warms up for five epochs from 10^{-6} before cosine decay.

Language fine-tuning. We use a sequence length of 256 for WikiText-103, One Billion Word, and CC-News, with the same method-specific configurations across these datasets. LoRA uses $r = 16$ and $\alpha = 32$; prefix tuning uses 50 prefix tokens; and prompt tuning uses 50 learned prompts.

Table 4: Backbone, adapter, and head configuration for the two merging domains.

	NLI	LLM
Backbone	BERT-base-cased	Llama-3.2-1B
Backbone state	frozen	frozen
Classification head	per-task classification head	shared LM head, per-task verbalizer
LoRA rank r	16	6
LoRA α	32	12
LoRA dropout	0.1	0.05
Target modules	`query`, `value`	`q_proj`, `v_proj`
Number of LoRA tensors	48 ($12 \times 2 \times 2$)	64 ($16 \times 2 \times 2$)
Trainable parameters	—	638,976/1,236,453,376 (0.052%)
Number of merged tasks	5	7 in-domain, 2 held out

The LoRA and prefix runs shown in the launch configurations use dropout rates of 0.05 and 0.1, respectively. We train with AdamW for up to 50,000 steps using a batch size of 96, an initial learning rate of 2×10^{-4}, cosine decay to 10^{-6}, no warm-up, gradient clipping at 1.0, and bfloat16 precision. Weight decay is set according to the individual run.

Connectivity evaluation. For each method and task, we compare independently fine-tuned solutions initialized from the same pretrained checkpoint. We evaluate the loss along linear interpolations of their trainable task-specific parameters, keeping the backbone fixed. Results are averaged over ten model pairs formed from five runs with different random seeds. We report the loss barrier for direct interpolation and, where applicable, after function-preserving parameter alignment.

Runtime Environment. All experiments were conducted on NVIDIA H100 NVL GPUs, each with 96 GB of memory. We used bfloat16 precision and up to 10 CPU workers for data loading and preprocessing. Peak GPU memory usage reached approximately 94 GB for language modeling tasks. Most fine-tuning runs completed within five hours, whereas the ImageNet-21K-to-ImageNet-1K transfer required approximately 40 hours.

B.3 Model merging setup

This subsection records the full configuration behind Section 7.3.

Backbones and adapters. Table 4 summarizes both domains. In each case the backbone is frozen and only the LoRA factors and the task head are trained. The LoRA initialization seed differs across tasks in both domains.

Fine-tuning configuration. All five NLI adapters share the same optimization settings. We use AdamW with learning rate 2×10^{-4} and weight decay 10^{-2}, 10 epochs and batch size 32. Seeds 1 through 5 are assigned to MNLI, SNLI, QNLI, e-SNLI, and ANLI respectively. The checkpoint used for merging is the final epoch for all five tasks, chosen so that the checkpoint selection rule is uniform across tasks. The language-model adapters use AdamW with a learning rate of 3×10^{-4} and no weight decay, batch size 32, and maximum sequence length 256, and differ in the number of epochs and in the verbalizer, as listed in Table 5. Merging uses the best validation checkpoint for each task.

Method settings. The settings below are held fixed across both domains. Methods carrying a scale parameter λ are swept and reported at the point of their grid with the highest average accuracy over the merged tasks. The default grid is $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 2.0\}$ for NLI and $\lambda \in \{0.1, 0.2, \ldots, 1.0\}$ for the language model.

- *Simple Average.* The main tables use the unscaled mean of the factors, corresponding to $\lambda = 1$ in Equation (23).
- *Task Arithmetic.* Default grid in both domains.

Table 5: Per-task fine-tuning configuration for the seven in-domain language-model adapters and the two held-out tasks. SNLI and WNLI are never trained on and are listed only for their verbalizers.

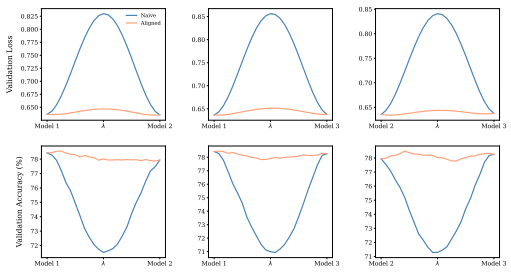
Dataset	Epochs	Batch	Seed	Verbalizer
CoLA	5	32	1	`[no, yes]`
SST-2	3	32	2	`[negative, positive]`
MRPC	10	32	3	`[no, yes]`
QQP	2	32	4	`[no, yes]`
MNLI	2	32	5	`[yes, maybe, no]`
QNLI	3	32	6	`[yes, no]`
RTE	10	32	7	`[yes, no]`
SNLI (held out)	—	—	—	`[yes, maybe, no]`
WNLI (held out)	—	—	—	`[no, yes]`

- *Ties-Merging.* Reset threshold 20.0, sign resolution by mass, disjoint-mean merge function. Swept on a widened grid, $0.2 \rightarrow 2.5$ for NLI and up to 3.0 for the language model.
- *DARE-TA and DARE-Ties.* Mask rate 0.5 with sampling seed 0; DARE-Ties keeps the Ties settings above. Default grid in both domains.
- *MagMax.* No parameters of its own beyond the scale. Default grid in both domains.
- *RegMean.* Off-diagonal reduction $a = 0.9$, Gram matrices estimated from 100 validation batches per task, LoRA factors merged and other tensors left alone. Swept over $0.8 \rightarrow 2.2$ for NLI and $0.2 \rightarrow 2.0$ for the language model.
- *LoRA-LEGO.* Target ranks $k = r$ and $k = 2r$, K-means with seed 0, 100 iterations and tolerance 10^{-4}, parameter-norm matching and output rescaling by $\sqrt{r}/\sqrt{k}$. No scale parameter.
- *LMC-LoRA dynamic.* Step weights $\lambda_i = 1/i$. No scale parameter from outside.
- *LMC-LoRA greedy.* Step grid $0.0 \rightarrow 1.0$ in steps of 0.1, varied in Appendix E.2, selection subset 2% of validation capped at 1000 examples for NLI and the full evaluation subset for the language model. No scale parameter from outside.

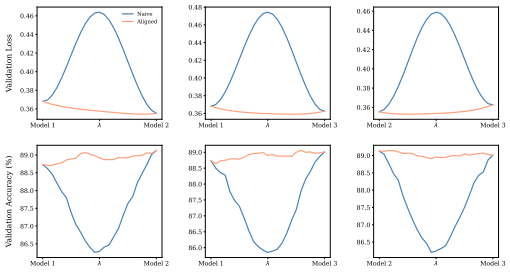
Using raw Gram matrices instead of off-diagonal reduction lowers the reported RegMean average from 61.9% to 42.6% on NLI. The best reported NLI settings for Ties-Merging and RegMean have $\lambda > 1$, so the wider grids affect their reported scores.

Evaluation protocol. NLI is evaluated on the validation split of each task with batch size 32. The language model is evaluated on a fixed subset of at most 1000 examples per task, drawn with sampling seed 0 and batch size 64, so that every method is scored on the same subset. The two held-out tasks are scored the same way. All tuning happens on these splits, including the scale parameter of every swept method and the coefficients that LMC-LoRA greedy selects internally. This can affect both absolute scores and method rankings.

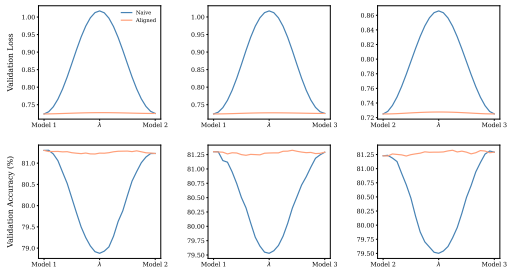
C LINEAR MODE CONNECTIVITY



(a) CIFAR-10

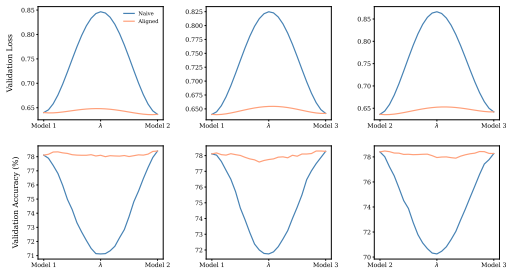


(b) CIFAR-100

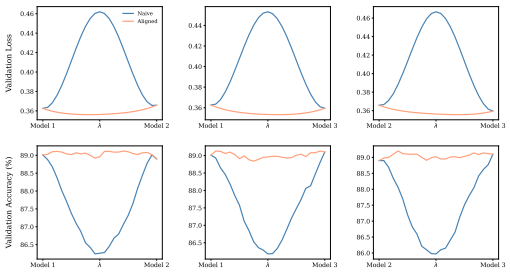


(c) ImageNet-1K

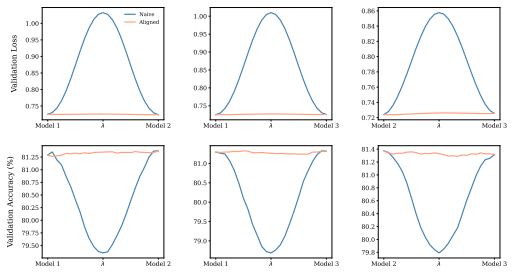
Figure 2: Loss along linear interpolations between LoRA-fine-tuned ViT models on three vision datasets. Each panel shows three model pairs formed from three random seeds.



(a) CIFAR-10

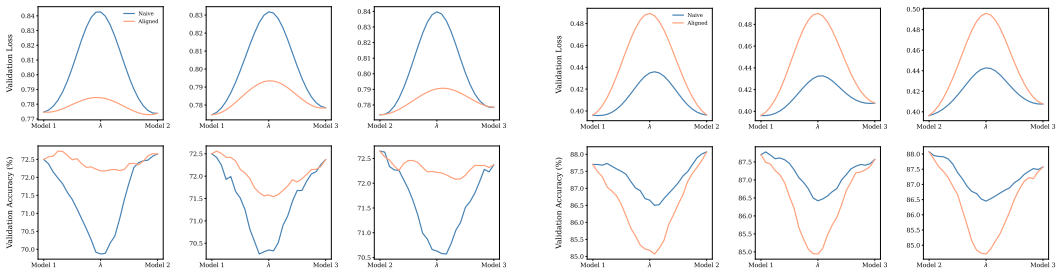


(b) CIFAR-100

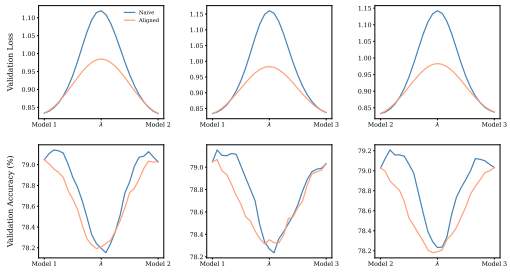


(c) ImageNet-1K

Figure 3: Loss along linear interpolations between prefix-tuned ViT models on three vision datasets. Each panel shows three model pairs formed from three random seeds.

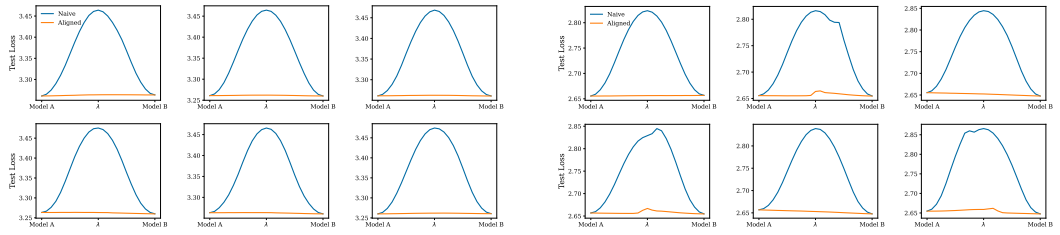


(a) CIFAR-10 (b) CIFAR-100

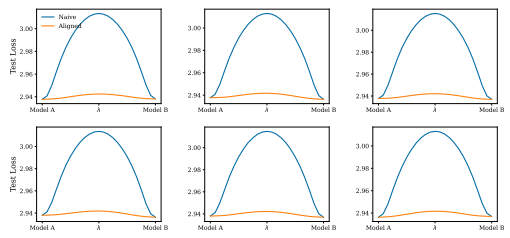


(c) ImageNet-1K

Figure 4: Loss along linear interpolations between prompt-tuning ViT models on three vision datasets. Each panel shows three model pairs formed from three random seeds.

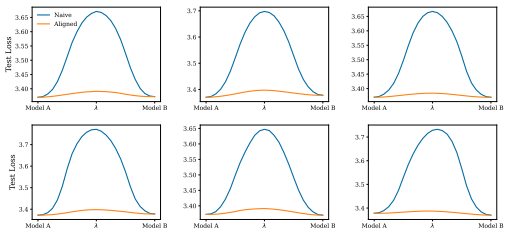


(a) One Billion Word (b) WikiText-103

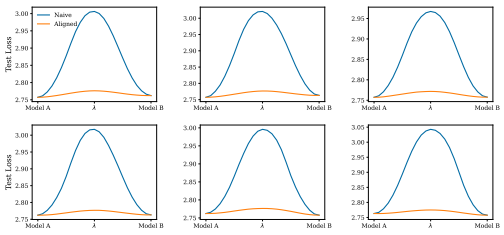


(c) CC-News

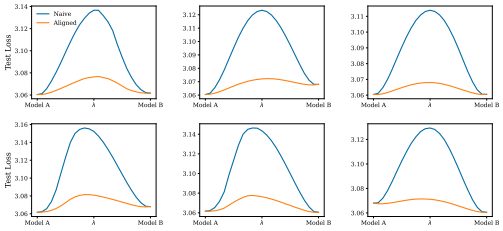
Figure 5: Loss barriers along parameter interpolation paths between LoRA-fine-tuned Qwen2.5 models on three language modeling datasets. Each panel shows six model pairs formed from four random seeds.



(a) One Billion Word

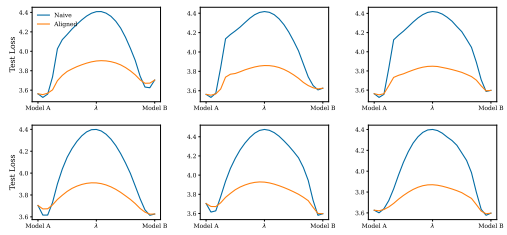


(b) WikiText-103

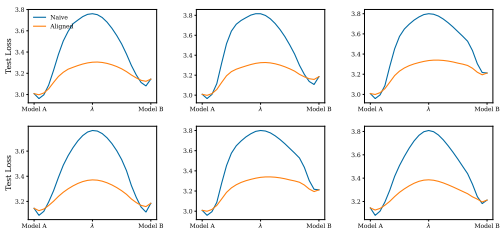


(c) CC-News

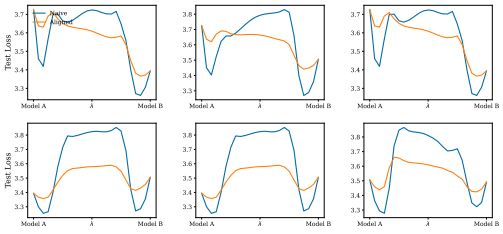
Figure 6: Loss barriers along parameter interpolation paths between prefix-tuned Qwen2.5 models on three language modeling datasets. Each panel shows six model pairs formed from four random seeds.



(a) One Billion Word



(b) WikiText-103



(c) CC-News

Figure 7: Loss barriers along parameter interpolation paths between prompt-tuning Qwen2.5 models on three language modeling datasets. Each panel shows six model pairs formed from four random seeds.

D ROBUSTNESS AND GENERALIZATION

Table 6: Robustness evaluation on CIFAR10-C across all corruptions. We report mean and standard deviation of accuracy (%) over severity levels for three methods: **Model**, **Naive**, and **Aligned**. Best results per corruption are in **bold**.

Corruption	Model	Naive	Aligned
Brightness	$\mathbf{74.24 \pm 3.82}$	67.03 ± 4.14	$\mathbf{74.24 \pm 3.61}$
Contrast	48.25 ± 18.68	45.66 ± 16.72	$\mathbf{48.79 \pm 18.39}$
Defocus Blur	68.58 ± 8.58	63.04 ± 7.51	$\mathbf{68.59 \pm 8.36}$
Elastic Transform	$\mathbf{70.89 \pm 1.89}$	64.86 ± 1.84	70.67 ± 1.76
Fog	60.16 ± 13.89	56.35 ± 12.35	$\mathbf{60.33 \pm 13.74}$
Frost	65.38 ± 6.30	60.09 ± 5.39	$\mathbf{65.45 \pm 6.29}$
Gaussian Blur	63.92 ± 10.30	58.95 ± 9.05	$\mathbf{64.03 \pm 10.13}$
Gaussian Noise	55.17 ± 10.72	51.96 ± 8.83	$\mathbf{55.20 \pm 10.60}$
Glass Blur	64.04 ± 4.06	59.17 ± 3.40	$\mathbf{64.40 \pm 3.97}$
Impulse Noise	55.77 ± 13.31	52.35 ± 10.49	$\mathbf{55.87 \pm 13.18}$
JPEG Compression	$\mathbf{72.11 \pm 2.34}$	65.64 ± 2.38	71.96 ± 2.31
Motion Blur	62.96 ± 6.00	57.97 ± 5.33	$\mathbf{62.97 \pm 5.90}$
Pixelate	$\mathbf{69.90 \pm 8.50}$	63.51 ± 7.82	69.58 ± 8.43
Saturate	67.24 ± 6.25	59.90 ± 6.46	$\mathbf{67.30 \pm 5.93}$
Shot Noise	59.93 ± 11.18	56.36 ± 8.89	$\mathbf{59.97 \pm 10.93}$
Snow	67.09 ± 4.36	61.29 ± 4.11	$\mathbf{67.10 \pm 4.27}$
Spatter	$\mathbf{72.19 \pm 3.12}$	65.61 ± 2.73	72.10 ± 2.93
Speckle Noise	59.75 ± 10.55	56.28 ± 8.27	$\mathbf{59.79 \pm 10.35}$
Zoom Blur	$\mathbf{65.82 \pm 4.57}$	60.11 ± 4.21	65.70 ± 4.54

Table 7: Robustness evaluation on CIFAR100-C across all corruptions. We report mean and standard deviation of accuracy (%) over severity levels for three methods: **Model**, **Naive**, and **Aligned**. Best results per corruption are in **bold**.

Corruption	Model	Naive	Aligned
Brightness	86.79 ± 2.29	83.35 ± 2.68	$\mathbf{86.97 \pm 2.20}$
Contrast	83.61 ± 5.16	81.57 ± 3.85	$\mathbf{84.18 \pm 4.75}$
Defocus Blur	83.56 ± 5.17	77.26 ± 7.82	$\mathbf{83.67 \pm 5.17}$
Elastic Transform	77.19 ± 9.21	70.81 ± 8.88	$\mathbf{77.45 \pm 9.04}$
Fog	79.12 ± 9.50	74.21 ± 10.73	$\mathbf{79.63 \pm 9.19}$
Frost	77.39 ± 5.95	74.62 ± 5.51	$\mathbf{78.02 \pm 5.67}$
Gaussian Blur	81.64 ± 5.76	74.50 ± 8.45	$\mathbf{81.77 \pm 5.76}$
Gaussian Noise	50.86 ± 14.14	$\mathbf{55.80 \pm 11.37}$	52.53 ± 13.82
Glass Blur	43.15 ± 8.44	44.27 ± 8.26	$\mathbf{44.40 \pm 8.64}$
Impulse Noise	56.65 ± 16.67	56.78 ± 14.31	$\mathbf{58.27 \pm 16.21}$
JPEG Compression	66.41 ± 5.91	63.22 ± 6.00	$\mathbf{66.97 \pm 5.92}$
Motion Blur	78.00 ± 5.17	71.94 ± 6.14	$\mathbf{78.40 \pm 5.04}$
Pixelate	70.91 ± 16.71	68.51 ± 15.10	$\mathbf{71.01 \pm 16.69}$
Saturate	83.02 ± 3.51	80.16 ± 3.55	$\mathbf{83.53 \pm 3.30}$
Shot Noise	56.32 ± 16.48	$\mathbf{60.62 \pm 13.18}$	57.81 ± 16.15
Snow	79.07 ± 3.84	75.11 ± 4.48	$\mathbf{79.44 \pm 3.80}$
Spatter	81.61 ± 3.51	77.45 ± 3.86	$\mathbf{81.98 \pm 3.48}$
Speckle Noise	56.66 ± 15.42	$\mathbf{60.53 \pm 13.01}$	58.10 ± 15.14
Zoom Blur	82.53 ± 2.30	76.12 ± 3.14	$\mathbf{82.65 \pm 2.40}$

Table 8: Robustness evaluation on ImageNet-C across all corruptions. We report mean and standard deviation of accuracy (%) over severity levels for three methods: **Model** , **Naive**, and **Aligned**. Best results per corruption are in **bold**.

Corruption	Model	Naive	Aligned
Brightness	76.18 ± 3.02	73.27 ± 4.46	$\mathbf{76.18 \pm 3.03}$
Contrast	$\mathbf{66.87 \pm 12.93}$	64.11 ± 15.98	66.84 ± 12.98
Defocus Blur	56.12 ± 13.47	54.21 ± 13.17	$\mathbf{56.40 \pm 13.29}$
Elastic Transform	61.79 ± 13.78	60.35 ± 13.47	$\mathbf{62.04 \pm 13.65}$
Fog	62.00 ± 11.16	60.27 ± 11.66	$\mathbf{62.22 \pm 10.99}$
Frost	51.63 ± 11.61	48.34 ± 11.96	$\mathbf{51.82 \pm 11.57}$
Gaussian Blur	58.27 ± 15.96	56.61 ± 15.48	$\mathbf{58.55 \pm 15.75}$
Gaussian Noise	48.56 ± 23.72	43.67 ± 24.00	$\mathbf{48.80 \pm 23.77}$
Glass Blur	45.06 ± 20.02	45.27 ± 18.62	$\mathbf{45.84 \pm 19.70}$
Impulse Noise	45.33 ± 22.56	40.77 ± 22.84	$\mathbf{45.66 \pm 22.69}$
JPEG Compression	67.74 ± 7.65	65.77 ± 7.88	$\mathbf{67.96 \pm 7.53}$
Motion Blur	60.40 ± 13.55	57.85 ± 13.75	$\mathbf{60.77 \pm 13.29}$
Pixelate	67.48 ± 11.21	66.43 ± 10.25	$\mathbf{67.94 \pm 10.79}$
Saturate	$\mathbf{72.58 \pm 5.78}$	67.85 ± 8.42	72.50 ± 5.78
Shot Noise	46.47 ± 23.60	40.79 ± 24.44	$\mathbf{46.69 \pm 23.69}$
Snow	52.93 ± 10.19	45.83 ± 12.92	$\mathbf{53.12 \pm 10.18}$
Spatter	64.90 ± 11.44	62.56 ± 11.71	$\mathbf{65.02 \pm 11.32}$
Speckle Noise	57.54 ± 15.07	52.53 ± 16.94	$\mathbf{57.73 \pm 15.07}$
Zoom Blur	51.45 ± 10.78	48.64 ± 10.91	$\mathbf{52.06 \pm 10.60}$

Table 9: Robustness and generalization under class-partitioned distribution shifts. Validation accuracy (↑) on complementary class-conditioned splits of CIFAR-10, CIFAR-100, and ImageNet with partition ratio α. *Naive* denotes standard training, while *Aligned* applies the proposed alignment method.

Ratio (α)	CIFAR10		CIFAR100		IMAGENET	
	Naive	Aligned	Naive	Aligned	Naive	Aligned
50%	70.23	74.98	86.0	88.37	79.142	80.642
60%	70.17	74.51	85.62	88.21	79.286	80.618
70%	69.78	74.21	85.88	87.92	78.906	80.484
80%	69.31	74.06	84.99	87.54	79.428	80.43
90%	69.09	72.56	83.79	87.49	78.42	80.166
100%	64.11	66.01	82.71	85.26	78.04	80.196

E MODEL MERGING

E.1 SENSITIVITY TO MERGE ORDER

Both LMC-LoRA variants merge adapters one at a time, matching each incoming adapter to the running merge, so their output can depend on the order in which the adapters are visited. The main tables merge the tasks in the order in which they are listed there. Tables 10 and 11 repeat both variants over several orders drawn at random and report the mean and standard deviation of each column. Every other setting is the one used in the main tables.

Table 10: NLI, per-task validation accuracy (%) over six merge orders, reported as mean $\pm$ standard deviation. Each column is the same quantity as in Table 2.

Method	MNLI	SNLI	QNLI	e-SNLI	ANLI	Avg
LMC-LoRA dynamic	63.78 ± 0.37	75.72 ± 0.15	65.97 ± 0.23	74.52 ± 0.15	36.58 ± 0.19	63.32 ± 0.10
LMC-LoRA greedy	72.33 ± 2.22	88.25 ± 1.36	59.23 ± 1.88	87.15 ± 0.87	38.58 ± 0.83	69.12 ± 0.56

Table 11: Language model, per-task accuracy (%) over five merge orders, reported as mean $\pm$ standard deviation. Each column is the same quantity as in Table 3.

Method	CoLA	SST-2	MRPC	QQP	MNLI	QNLI	RTE	Avg IID	SNLI	WNLI	Avg OOD
LMC-LoRA dynamic	68.46 ± 0.28	90.35 ± 0.67	56.72 ± 4.59	64.44 ± 0.05	52.42 ± 1.84	69.56 ± 1.13	58.19 ± 2.18	65.73 ± 0.26	45.36 ± 3.11	43.66 ± 2.23	44.51 ± 2.65
LMC-LoRA greedy	63.88 ± 2.16	79.82 ± 7.75	72.30 ± 3.03	74.66 ± 6.28	87.04 ± 1.82	69.76 ± 2.83	70.98 ± 2.89	74.06 ± 0.60	84.84 ± 2.27	47.61 ± 3.05	66.22 ± 2.34

Both variants are stable under reordering. LMC-LoRA dynamic is nearly insensitive, with a standard deviation of 0.10 on the NLI average and 0.26 on the in-domain language-model average. LMC-LoRA greedy varies more, since each of its coefficients is selected against the tasks merged so far and that set depends on the order. Its per-task spread is largest on the tasks visited late. Its averages still move by at most 1.5 points on NLI and 1.3 points on the language model. The ranking is also stable. LMC-LoRA greedy remains the highest-scoring method of Table 2 under every NLI order, and its weakest order stands 4.3 points above the best baseline there. It remains the highest-scoring method of Table 3 under four of the five language-model orders, falling 0.4 points below RegMean in the remaining one. The averages reported in Tables 2 and 3 lie within one standard deviation of the mean over orders.

E.2 RESOLUTION OF THE GREEDY GRID

LMC-LoRA greedy selects each coefficient from a grid on $[0, 1]$. Tables 12 and 13 vary the spacing of that grid, holding every other setting fixed. Against the step of 0.1 used in the main tables, the coarsest grid loses 1.2 points on NLI and 1.9 points on the language model. Halving the step and then halving it again moves the merged average by at most 0.4 points in either domain, downwards on NLI and upwards on the language model. The step of 0.1 is therefore not a limiting choice.

Table 12: NLI, per-task validation accuracy (%) as a function of the spacing of the greedy grid. Every other setting is the one used in Table 2, and the step 0.1 row is that run.

Grid step	Grid points	MNLI	SNLI	QNLI	e-SNLI	ANLI	Avg
0.2	6	69.70	87.10	58.70	89.80	37.70	68.60
0.1	11	75.30	89.40	58.00	86.90	39.40	**69.80**
0.05	21	75.60	89.00	57.80	86.40	39.60	69.70
0.02	51	75.30	88.90	58.00	85.50	39.30	69.40

Table 13: Language model, per-task accuracy (%) as a function of the spacing of the greedy grid. Every other setting is the one used in Table 3, and the step 0.1 row is that run.

Grid step	Grid points	CoLA	SST-2	MRPC	QQP	MNLI	QNLI	RTE	Avg IID	SNLI	WNLI	Avg OOD
0.2	6	66.50	83.37	71.32	74.20	80.90	64.90	65.34	72.36	77.80	43.66	60.73
0.1	11	63.50	78.21	74.51	77.50	87.50	69.50	68.95	74.24	**84.80**	46.48	**65.64**
0.05	21	63.80	78.56	74.75	77.50	87.60	69.90	68.59	74.39	84.50	46.48	65.49
0.02	51	64.80	79.36	75.00	77.50	87.30	69.60	68.95	74.64	84.50	46.48	65.49