BEYOND CONTINUITY OR DISCONTINUITY: ZIGZAG ALGEBRA FOR COMPOSITIONAL RELATIONS IN LOW-RANK ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern neural networks continuously update parameters in a shared space, while existing methods provide only limited forms of relational organization. We introduce a framework that partitions the trainable parameter space into groups, infers their relations from the model's own training signals, and organizes them through zigzag algebra. Zigzag algebra is a path-based algebra in which vertices are connected by directional arrows and relations constrain how these arrows can compose. We map parameter groups to vertices and their interactions to arrows, allowing strongly related groups to remain closely connected, weakly related groups to interact less, and unrelated groups to become effectively separated. Unlike orthogonality, expert routing, or heuristically assigned group labels, our approach learns this graded relational structure directly from training dynamics while using algebraic paths to constrain higher-order interactions. We instantiate the framework within LoRA by partitioning the low-rank space and imposing these algebraic relations during optimization, without additional inference-time parameters, routing modules, or architectural branches.

1 INTRODUCTION

Modern neural networks are fundamentally built on continuous optimization. Through backpropagation, model parameters are repeatedly adjusted in response to data, allowing large language models (LLMs) to acquire rich representations (Jawahar et al., 2019; Zhang et al., 2021; Tenney et al., 2019) and adapt to diverse tasks. This framework is highly expressive, yet the organization of learned knowledge and parameter space remains largely implicit (Roberts et al., 2020; Hewitt and Manning, 2019; Petroni et al., 2019). Parameters interact through network computation, attention, and the loss landscape, while the optimization process does not explicitly specify which regions should remain strongly related, which should interact only weakly, which should become effectively separated, or how such relations should compose.

This motivates two broader questions:

1. How can a continuously optimized parameter space acquire an explicit relational organization among its learned regions?

2. Can standard optimization organize learned knowledge by graded relational distance, rather than uniform separation?

The first question concerns the absence of an explicit relational structure in standard parameter optimization. Although different parameter regions may specialize during training, gradient-based optimization does not explicitly distinguish which regions should respond to different learning signals or specify how those regions should interact. Consequently, distinct signals can become entangled across the parameter space, making their roles and interactions difficult to separate (Huben et al., 2024; Mueller et al., 2026; Zhang et al., 2026; Lim and Lauw, 2023). This motivates a clearer organization of parameter groups and their respective roles.

The second question concerns an important property of human reasoning: knowledge can be organized through different degrees and compositions of relation. Learning that a banana is yellow is directly related to the concept of a banana, more broadly connected to fruit, and only indirectly

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.

connected to concepts such as food or nutrition through intermediate categories. At the same time, learning that a banana is yellow should not alter our belief that $1 + 1 = 2$. These relations are compositional: relations can propagate through intermediate concepts, while longer relational paths need not have the same character or strength as local ones. Standard neural optimization, however, does not explicitly specify such graded and compositional organization among learned parameter regions.

Existing approaches introduce useful forms of structure or modularity, but provide only partial mechanisms for expressing such relational diversity. Orthogonal and structured LoRA variants divide the adaptation space into distinct directions or subspaces and reduce redundancy through orthogonality, decorrelation, or related geometric constraints (Park et al., 2025). These methods can reduce overlap and interference, but their relational structure is generally uniform: different subspaces are encouraged to remain separated according to the same criterion. They therefore distinguish groups effectively, while offering limited mechanisms for expressing that some group pairs should remain closely related, others weakly related, and others nearly independent.

Mixture-of-experts (Artetxe et al., 2022; Zhu et al., 2024) and LoRA-expert architectures (Zhu et al., 2023; Li et al., 2024; Zhang et al., 2024b; Chang et al., 2026) provide another form of modularity. These methods partition computation into multiple experts and use learned routing, gating, or mixing mechanisms to specialize different modules. In sparse MoE, however, top-k routing restricts each input to a small subset of experts, leaving the remaining experts inactive for that computation. This discrete selection can limit smooth information sharing and richer interactions across multiple experts. Moreover, inter-expert organization is largely mediated by routing scores and selection decisions, rather than by explicit relations among the experts themselves. As a result, the expressive structure among experts depends heavily on the router, while higher-order relations involving several experts are not directly modeled through compositional rules.

These approaches demonstrate the value of differentiating parameter groups, but the relations they impose are typically pairwise and uniform, such as orthogonality, rather than allowing different group pairs to exhibit distinct relational roles. We therefore equip a continuous parameter space with a discrete algebraic structure in which parameter groups are *vertices*, directional inter-group interactions are *arrows*, and their compositions are *paths*. Each group remains continuously trainable. The algebra does not determine what each group must learn; it determines how differently learned groups are allowed to relate, allowing interactions to range from closely coupled to weakly related or effectively separated according to their relational structure.

We develop this formulation using **zigzag algebra** (Huerfano and Khovanov, 2001). Zigzag algebra provides explicit relations over paths, allowing local and higher-order interactions to follow different constraints. Neighboring groups remain connected through learned directional relations, while longer paths are constrained through their algebraic compositions. Because these relations are constructed from the learned representations of the groups, the resulting objective induces distinct relational roles and organizes different degrees of interaction across the parameter space.

We instantiate this structure within LoRA by partitioning the rank space into parameter groups and applying the algebraic objective to their learned interactions. All groups remain trainable under the original task objective, while the additional regularization shapes their relational organization. The construction requires no additional experts, routing networks, architectural branches, or inference-time parameters. This provides a principled route toward richer relational structures and a practical mechanism for incorporating pure abstract algebra into differentiable neural training.

Our contributions are as follows:

- We introduce compositional paths among LoRA parameter groups as explicit optimization targets, extending existing subspace- and expert-based approaches beyond group differentiation or top-(k) selection toward structured inter-group relations.
- The resulting path constraints induce heterogeneous relational roles across parameter groups, allowing some interactions to remain strong, others weak, and selected compositions to be suppressed without disabling any group from training.
- We formulate zigzag algebra in a GPU-friendly manner, enabling efficient scaling and straightforward integration into large-scale neural training.

- The method requires no additional parameters or experts and can be implemented simply by adding algebraic constraints to the training loss objective.
- To the best of our knowledge, we provide the first direct application of zigzag algebra to LLM training.

2 Related Work

2.1 Algebraic Structures in Neural Learning

Prior works such as Ganev and Walters (2022); Armenta and Jodoin (2021) use quiver representation theory to describe neural architectures mainly. These studies establish important connections between representation theory and neural networks, but primarily use algebra for neural network's structural characterization rather than translating algebraic relations into optimization objectives for LLM adaptation. Our framework realizes the algebra directly inside the LoRA parameter space. Vertices are trainable low-rank parameter groups, arrows are cross-moment matrices between neighboring groups, and paths are ordinary matrix compositions of these arrows. Zigzag relations are then reduced to differentiable scalar losses, avoiding symbolic algebra while remaining compatible with GPU-parallel matrix operations and standard backpropagation. Compared to general advanced level math applying to Ai especially for LLM, Ours does not require prohibitively expensive computation.

2.2 LoRA Experts and Mixture-of-Experts

LoRA-expert and mixture-of-experts methods introduce multiple parameter modules to increase specialization, diversity, or parameter efficiency. OMoE (Feng et al., 2025) promotes expert diversity through orthogonality, while SiRA (Zhu et al., 2023), MixLoRA (Li et al., 2024), LoRAMoE (Dou et al., 2024), MoELoRA (Luo et al., 2024), MoLE (wu et al., 2024), HMoRA (Liao et al., 2025), and stronger low-rank expert mixtures (Sun et al., 2025) develop increasingly sophisticated routing, selection, and expert-combination mechanisms. SOS-LoRA (Chang et al., 2026) further combines multiple scales with orthogonal input-side directions. In these approaches, inter-expert structure is induced primarily through routing, orthogonality, or mixture design. Our framework instead makes the paths among parameter groups explicit optimization targets: all groups remain trainable, while algebraic constraints directly regulate how strongly groups relate and how these relations propagate through compositions. This provides a graded relational structure beyond expert selection or uniform separation.

2.3 Orthogonal Low-Rank Adaptation

Structured LoRA methods organize the low-rank parameter space into multiple components. CPS-LoRA (Xiong et al., 2026) partitions LoRA space into multiple subspace blocks with an orthogonal regularizer, while MELoRA (Ren et al., 2024) trains multiple small LoRA modules to increase effective rank and capture diverse information. LoRA^2 (Zhang et al., 2024a) learns low-rank components in mutually orthogonal planes. PiSSA (Meng et al., 2024) initializes adaptation from the principal singular components of pretrained weights, while HD-PiSSA (Wang et al., 2025) extends this idea by distributing adapters across distinct principal-component subspaces to obtain orthogonal update directions. These approaches improve diversity or reduce interference by structuring the adaptation space, but orthogonality and subspace separation primarily encode independence or low overlap. Such constraints do not express graded, directional, or compositional relations among groups and may suppress useful shared structure. Our framework instead makes these inter-group relations explicit optimization targets.

3 Method

3.1 Zigzag Algebra over the LoRA Rank Space

We impose a discrete zigzag-algebraic structure on the continuous LoRA rank space. Consider the three-vertex double quiver

$$1 \underset{A_{21}}{\overset{A_{12}}{\rightleftarrows}} 2 \underset{A_{32}}{\overset{A_{23}}{\rightleftarrows}} 3.$$

Its non-returning length-two paths vanish,

$$A_{23}A_{12} = 0, \qquad A_{21}A_{32} = 0,$$

the two return cycles at the central vertex are identified,

$$A_{12}A_{21} = A_{32}A_{23},$$

and composable paths of length three are annihilated. The latter follows from the exact quadratic zigzag relations; its role in our soft objective is discussed in Appendix 6.3.1.

For a standard LoRA update

$$\Delta W = \frac{\alpha}{r} BA,$$

we partition the rank dimension as $r = r_1 + r_2 + r_3$ and associate the three rank groups with the quiver vertices. Our primary formulation uses the LoRA-A bottleneck output at the supervised answer position,

$$X = A(x)_{s_{\text{ans}}} = [X_1 \mid X_2 \mid X_3] \in \mathbb{R}^{B \times r},$$

while the weight-side formulation partitions

$$B = [B_1 \mid B_2 \mid B_3]$$

and sets $X_i = B_i$. Each source is evaluated independently; when both are used, their corresponding algebraic losses are summed.

3.2 Arrows and Path Constraints

Given

$$X = [X_1 \mid X_2 \mid X_3], \qquad X_i \in \mathbb{R}^{n \times r_i},$$

we construct arrows directly from neighboring groups,

$$A_{ij} = \frac{1}{n} X_j^\top X_i,$$

so that

$$A_{12} = \frac{1}{n} X_2^\top X_1, \quad A_{21} = \frac{1}{n} X_1^\top X_2,$$

$$A_{23} = \frac{1}{n} X_3^\top X_2, \quad A_{32} = \frac{1}{n} X_2^\top X_3.$$

Thus, arrows introduce no additional trainable parameters and evolve directly with the underlying LoRA representation. Their dimensional consistency and transpose symmetries are detailed in Appendix 6.3.2.

Defining

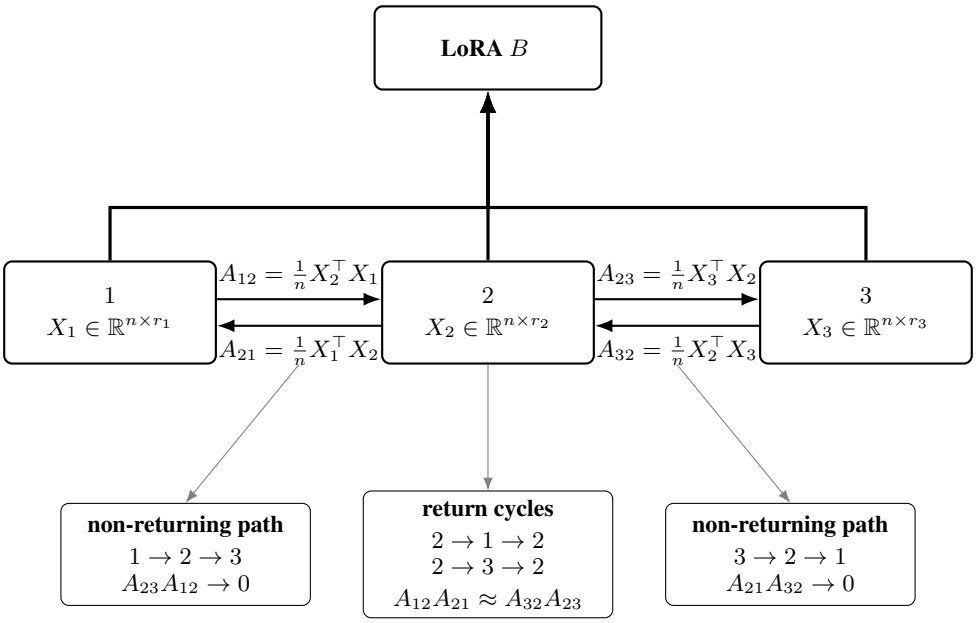

Figure 1: Zigzag-algebraic organization of the LoRA rank space. The LoRA A projection is partitioned into three learned rank-space groups X_1, X_2, X_3, each associated with a quiver vertex. These groups feed forward into the standard LoRA B projection. Directional interactions between neighboring groups are represented by cross-moment matrices A_{ij}, and compositions of these arrows realize algebraic paths constrained by the zigzag relations.

$$\phi(M) = \frac{1}{|M|}\|M\|_F^2,$$

we penalize the two non-returning paths by

$$\mathcal{L}_{123} = \phi(A_{23}A_{12}), \qquad \mathcal{L}_{321} = \phi(A_{21}A_{32}),$$

and impose the central cycle relation through

$$\mathcal{L}_{\text{cycle}} = \phi(A_{12}A_{21} - A_{32}A_{23})\,.$$

For the eight composable length-three paths,

$$\begin{aligned}\mathcal{P}_3 = \{&A_{12}A_{21}A_{12}, A_{32}A_{23}A_{12}, A_{21}A_{12}A_{21}, A_{23}A_{12}A_{21},\\ &A_{12}A_{21}A_{32}, A_{32}A_{23}A_{32}, A_{21}A_{32}A_{23}, A_{23}A_{32}A_{23}\},\end{aligned}$$

we define

$$\mathcal{L}_{\text{long}} = \frac{1}{8}\sum_{P \in \mathcal{P}_3} \phi(P).$$

Although length-three vanishing is implied by the exact zigzag relations, we penalize these paths explicitly to control higher-order residuals away from the exact relation manifold; see Appendix 6.3.3.

Each algebraic term is computed independently for every targeted LoRA module and averaged across modules. The resulting objective preserves direct neighboring interactions while suppressing their non-local compositions.

Table 1: Comparison across LoRA variants, ranks, and group configurations. We report accuracy (%) on MMLU-Pro ($n = 12{,}032$), MMLU ($n = 14{,}042$), and SciQ ($n = 1{,}000$). Avg. is the mean across the three benchmarks; best results are shown in bold. For Expert routing, methods activate few parts of parameter space.

Method	Rank	Groups	MMLU-Pro	MMLU	SciQ	Avg.
Zigzag algebra						
	24	3	43.12	71.79	94.70	69.87
	24	6	42.34	71.97	94.90	69.74
	36	3	42.95	71.92	94.90	69.92
	36	6	42.94	71.89	94.80	69.88
	72	3	43.73	72.18	95.00	70.30
	72	6	44.00	72.62	95.30	70.64
	126	3	**45.79**	72.73	95.00	71.17
	126	6	45.66	72.90	95.30	**71.29**
Plain low-rank adaptation						
Plain LoRA	24	–	40.56	71.53	94.40	68.83
	36	–	41.39	71.75	94.70	69.28
	72	–	42.67	72.35	95.10	70.04
	126	–	43.65	**73.19**	95.20	70.68
Weight decomposition						
DoRA	24	–	40.86	71.59	94.30	68.92
	36	–	41.56	71.85	94.80	69.40
	72	–	42.69	72.35	95.10	70.05
	126	–	43.40	73.11	95.20	70.57
Orthonormal initialization						
OLoRA	24	–	43.33	72.92	**95.60**	70.62
	36	–	43.47	72.92	95.10	70.50
	72	–	42.32	72.49	95.00	69.94
	126	–	41.16	72.56	94.70	69.47
Parallel decomposition + fixed scale + orthogonal subspace						
SOS-LoRA	24	3	44.04	72.99	95.30	70.78
	36	3	42.40	72.33	94.90	69.88
	36	6	42.56	72.56	95.30	70.14
	72	3	43.78	73.04	95.40	70.74
	72	6	43.87	72.97	95.40	70.75
	126	3	44.16	73.04	95.20	70.80
	126	6	43.99	72.98	95.30	70.76
Expert routing						
MoLA	24	3	36.43	70.45	93.90	66.93
	36	3	37.72	70.96	94.50	67.73
	36	6	36.93	70.70	94.20	67.28
	72	3	40.72	71.75	94.40	68.96
	126	3	41.76	72.24	95.00	69.67
MixLoRA	24	3	36.00	69.57	93.40	66.32
	36	3	36.39	70.15	94.30	66.95
	36	6	36.28	69.81	94.00	66.70
	72	3	37.03	70.45	95.00	67.49
	72	6	36.77	70.14	94.70	67.20
	126	3	38.00	71.08	95.40	68.16

3.3 Training Objective

We define

$$\mathcal{L}_{\text{zigzag}} = \frac{1}{4}\left(\mathcal{L}_{123} + \mathcal{L}_{321} + \mathcal{L}_{\text{cycle}} + \mathcal{L}_{\text{long}}\right).$$

Because minimizing path compositions alone may collapse the primitive arrows, we preserve local interactions using

$$\mathcal{L}_{12} = -\phi(A_{12}), \qquad \mathcal{L}_{32} = -\phi(A_{32}).$$

Since

$$A_{21} = A_{12}^{\top}, \qquad A_{23} = A_{32}^{\top},$$

Table 2: Effect of the Zigzag loss coefficient λ_{zig} in $\mathcal{L}_{\text{CE}} + \lambda_{\text{zig}}\mathcal{L}_{\text{zig}}$, without the locality terms.

λ_{zig}	0.0	0.4	0.8	1.0	1.2	1.6	2.0
Accuracy	40.72	43.25	43.22	43.42	43.42	**43.48**	43.43

the reverse directions have identical Frobenius energy and need not be counted separately. Under the transpose-tied cross-moment realization, the locality terms counteract collapse toward the zero-arrow configuration under the soft objective.

The final objective is

$$\boxed{\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{zigzag}}\mathcal{L}_{\text{zigzag}} + \lambda_{12}\mathcal{L}_{12} + \lambda_{32}\mathcal{L}_{32}}.$$

$\mathcal{L}_{\text{task}}$ is the cross-entropy loss. All algebraic terms are computed from existing LoRA activations or weights, and their gradients are backpropagated through the standard training graph, introducing no additional trainable parameters.

Design principle. The objective does not seek exact satisfaction of the zigzag relations; under the transpose-tied realization, the only exact solution is trivial Appendix 6.3.1. Instead, the relations define a suppression hierarchy over path types: primitive arrows are not penalized directly and are supported by the locality terms, return cycles are constrained only through their mismatch, while non-returning length-two paths and all length-three paths are penalized directly. All inter-group relations may therefore shrink, but by different amounts depending on path type. This contrasts with pairwise orthogonality, which applies a uniform penalty to every group pair, including the direct relation between non-adjacent groups that our objective leaves unconstrained.

3.4 Extension to More Than Three Rank Groups

The three-group construction is the A_3 instance of the zigzag algebra defined as a quotient of the path algebra,

$$Z(\Gamma) = \Bbbk\overline{\Gamma}/I,$$

where $\overline{\Gamma}$ is the doubled quiver and I is generated by non-returning length-two paths and differences of length-two return cycles based at the same vertex. For graphs with more than two vertices, paths of degree three or higher consequently vanish.

For a chain $\Gamma = A_K$, we associate the vertices with

$$X = [X_1 \mid \cdots \mid X_K], \qquad \sum_{i=1}^{K} r_i = r,$$

and realize adjacent arrows by

$$A_{i,i+1} = \frac{1}{n} X_{i+1}^{\top} X_i.$$

The quotient therefore restricts path composition rather than the number of vertices: arbitrary K is allowed while non-local compositions are annihilated. Our implementation uses $K = 3, 6$ and the same construction extends directly to A_K or more general graphs Γ, see Appendix 6.3.4.

4 Experiment

4.1 Main Results

We evaluate low-rank adaptation methods on Qwen2.5-7B-Instruct across three multiple-choice benchmarks. Table 1 compares different rank and group configurations. Zigzag LoRA shows par-

Table 3: Performance across different random seeds. Seed 42 is default setting. Base model accuracy is 28.80.

Method	Seed 40	Seed 41	Seed 43	Seed 44
Plain LoRA	38.33	39.30	39.41	39.28
Ours	**41.46**	**42.56**	**42.00**	**41.53**

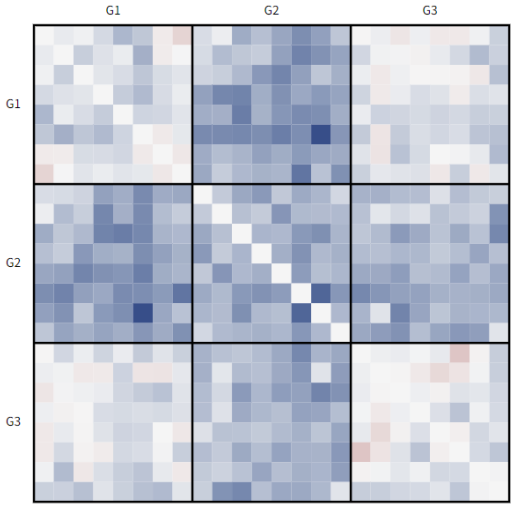

Figure 2: Analysis of our LoRA-A parameter space.

ticularly strong gains on MMLU-Pro, reaching the best accuracy of 45.79% at rank 126 with three groups, and achieves the best overall average of 71.29% with six groups. Plain LoRA remains strongest on MMLU (73.19%), while OLoRA obtains the best SciQ result (95.60%), indicating that the advantage of Zigzag LoRA is not uniform across individual tasks. At smaller ranks, OLoRA and SOS-LoRA are competitive or stronger in average accuracy, whereas Zigzag LoRA improves more consistently as the rank budget increases.

Unlike expert-routing methods, Zigzag LoRA introduces no additional experts or input-dependent router and keeps all rank groups jointly trainable. Also, whereas expert-routing methods activates few expert subspaces, ours are fully activated. We further compare against activation- and parameter-space orthogonality baselines; Zigzag LoRA consistently performs better in these controlled comparisons, suggesting that orthogonal separation alone does not reproduce the observed gains. Detailed results are provided in Appendix 6.2 and Table 6.

4.2 Ablation Studies

Table 2 shows that all nonzero zigzag coefficients ($\lambda_{\text{zig}} \geq 0.4$) consistently improve over $\lambda_{\text{zig}} = 0$ with only small variation across settings, indicating both effectiveness and hyperparameter robustness. Across all tested seeds, our method also outperforms Plain LoRA, suggesting that the gain is not tied to a particular initialization.

Table 4 shows that locality weights are robust over a broad range: at rank 126, accuracy remains near 45.6–45.9% for coefficients from 1 to 10, but drops to 35.59% at 100. The combined ablations in Table 5 further show that equal group allocation performs best among the tested ratios.

The A-only and $A{+}B$ objectives achieve similar improvements over Plain LoRA (42.96% and 43.13% vs. 40.56%, corresponding to $+2.40$ and $+2.57$ percentage points), whereas the B-only objective remains near the Plain LoRA baseline (40.50%). In particular, the similar gain obtained when both LoRA factors are constrained makes a simple one-sided A/B rescaling effect insufficient to explain the improvement.

Table 4: Ablation on the locality coefficients A_{12} and A_{32} at different LoRA ranks on MMLU-Pro.

Rank 24

A_{12}	0.1	0.1	0.2	0.2	0.2	0.4	0.8	0.8	1.0
A_{32}	0.1	0.4	0.2	0.3	0.4	0.2	0.8	1.0	0.8
Acc.	43.33	43.52	43.45	43.43	43.47	43.36	**43.53**	43.39	43.29

Rank 126

A_{12}	1.0	2.0	3.0	4.0	5.0	6.0	10.0	100.0
A_{32}	1.0	2.0	3.0	4.0	5.0	6.0	10.0	100.0
Acc.	45.74	**45.94**	45.82	45.71	45.84	45.93	45.55	35.59

Table 5: Ablation studies on group allocation, zigzag targets, and loss components.

Group ratio	Acc.
0.4–0.2–0.4	42.35
1.0–1.0–1.0	**42.96**
0.2–0.6–0.2	42.48

Target	Acc.
A	42.96
B	40.50
A+B	**43.13**

Ablation	Acc.	Ablation	Acc.
Full	42.96	Local12 $= 0$	**43.06**
Plain LoRA	40.56	Local32 $= 0$	43.04
Zigzag $= 0$	11.39	Local12,32 $= 0$	42.99

Removing either or both locality terms changes accuracy only slightly. The "Zigzag $= 0$" setting is qualitatively different: the negative locality terms remain active without the positive path constraints that control their growth, producing an unbalanced objective. Its 11.39% accuracy therefore reflects the stabilizing role of the zigzag terms within the coupled objective.

4.3 Analysis of Rank-Space Algebraic Structure

We analyze rank-24 LoRA-A with three groups G_1, G_2, G_3. Figure 2 reports post-hoc cosine similarities between group activations. Our method reduces the overall mean similarity from 0.2996 to 0.2656 ($\Delta = -0.0340$), with the largest decreases around G_2, indicating non-uniform reorganization of the rank space.

Algebraic diagnostics also show substantially smaller relation magnitudes for our model: $A_{12} = 0.0312$, $A_{32} = 0.0247$, and cycle mismatch 0.0175, compared with 2.22, 2.30, and 2.50×10^2 for Plain LoRA. Plain LoRA additionally yields composed-path and length-three statistics of 1.41×10^3 and 1.96×10^6, while MMLU-Pro improves from 40.60% to 43.15%.

Because reverse arrows in our cross-moment realization are transpose-tied, the exact simultaneous satisfaction of the zigzag relations admits no nonzero exact solution. This does not imply collapse of the learned LoRA update: our method uses these relations as soft regularizers jointly with the task objective and locality terms rather than solving for exact algebraic satisfaction. The comparable performance of the A-only and $A{+}B$ variants further indicates that the observed gain cannot be attributed solely to trivial one-sided factor rescaling.

Removing the zigzag term causes the structural quantities to diverge, with path statistics reaching 10^{22} and accuracy falling to 11.39%. Together with the cosine and orthogonality analyses, these results indicate that the soft zigzag objective changes the organization and composition of LoRA rank components rather than merely imposing uniform separation.

5 Conclusion

We introduce a relational organization of the LoRA rank space using zigzag algebra. By deriving vertices from trainable parameter groups or their representations, and paths from the learned relations among them, both the objects and their interactions evolve with the training signal. This allows parameter groups to acquire distinct relational roles rather than being separated only geometrically or architecturally. Our results suggest that algebraic structure can serve as a lightweight inductive bias for organizing how learned parameter groups interact and compose, without introducing additional trainable modules.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and translation, code implementation, debugging and refactoring, experimental design, and the organization and presentation of technical material. They were also used during the development and examination of mathematical formulations and claims, exploratory proof discussions, interpretation of experimental results, and identification of potentially useful conceptual connections. All AI-assisted content was reviewed by the authors, who take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Our construction follows the definition of zigzag algebra given by Huerfano and Khovanov Huerfano and Khovanov (2001) (Sec. 3, p. 4), where the path algebra of the doubled graph is quotiented such that non-returning length-two paths vanish and length-two return paths at the same vertex are identified. Our A_3 realization, cross-moment construction, exact relations, and their extension are described in Sec. 3 and Appendix A. We report the datasets, evaluation protocol, training hyperparameters, rank/group configurations, and ablation settings used in all experiments. The supplementary code contains the training and evaluation scripts with fixed random seeds and the exact configurations required to reproduce the reported results.

REFERENCES

Marco Armenta and Pierre-Marc Jodoin. The representation theory of neural networks. *Mathematics*, 9(24):3216, December 2021. ISSN 2227-7390. doi: 10.3390/math9243216. URL http://dx.doi.org/10.3390/math9243216.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giridharan Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeffrey Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Veselin Stoyanov. Efficient large scale language modeling with mixtures of experts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.804/.

Yupeng Chang, Yuan Wu, and Yi Chang. SOS-LoRA: Static orthogonal-subspace low-rank adaptation with fixed multi-scale scaling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3993–4005, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.184. URL https://aclanthology.org/2026.acl-long.184/.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.106. URL https://aclanthology.org/2024.acl-long.106/.

Jinyuan Feng, Zhiqiang Pu, Tianyi Hu, Dongmin Li, Xiaolin Ai, and Huimu Wang. Omoe: Diversifying mixture of low-rank adaptation by orthogonal finetuning, 2025. URL https://arxiv.org/abs/2501.10062.

Iordan Ganev and Robin Walters. Quiver neural networks, 2022. URL https://arxiv.org/abs/2207.12773.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419/.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 7827–7845, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

Ruth Stella Huerfano and Mikhail Khovanov. A category for the adjoint representation, 2001. URL https://arxiv.org/abs/math/0002060.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://aclanthology.org/P19-1356/.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts, 2024. URL https://arxiv.org/abs/2404.15159.

Mengqi Liao, Wei Chen, Junfeng Shen, Shengnan Guo, and Huaiyu Wan. Hmora: Making llms more effective with hierarchical mixture of lora experts. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 91309–91330, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e43a33994a28f746dcfd53eb51ed3c2d-Paper-Conference.pdf.

Jia Peng Lim and Hady Lauw. Disentangling transformer language models as superposed topic models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8646–8666, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.534. URL https://aclanthology.org/2023.emnlp-main.534/.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models, 2024. URL https://arxiv.org/abs/2402.12851.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 121038–121072. Curran Associates, Inc., 2024. doi: 10.52202/079017-3846. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/db36f4d603cc9e3a2a5e10b93e6428f2-Paper-Conference.pdf.

Aaron Mueller, Andrew Lee, Shruti Joshi, Ekdeep Singh Lubana, Dhanya Sridhar, and Patrik Reizinger. From isolation to entanglement: When do interpretability methods identify and disentangle known concepts? In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17188–17210, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.782. URL https://aclanthology.org/2026.acl-long.782/.

JuneYoung Park, Minjae Kang, Seongbae Lee, Haegang Lee, Seongwan Kim, and Jaeho Lee. Riemannian optimization for LoRA on the stiefel manifold. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20971–20985, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1143. URL https://aclanthology.org/2025.findings-emnlp.1143/.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250/.

Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Jiahuan Pei. MELoRA: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3052–3064, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.168. URL https://aclanthology.org/2024.acl-long.168/.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.437. URL https://aclanthology.org/2020.emnlp-main.437/.

Mengyang Sun, Yihao Wang, Tao Feng, Dan Zhang, Yifan Zhu, and Jie Tang. A stronger mixture of low-rank experts for fine-tuning foundation models. *arXiv preprint arXiv:2502.15828*, 2025.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/.

Yiding Wang, Fanxu Meng, Xuefeng Zhang, Fan Jiang, Pingzhi Tang, and Muhan Zhang. HD-PiSSA: High-rank distributed orthogonal adaptation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6515–6528, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.330. URL https://aclanthology.org/2025.emnlp-main.330/.

xun wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 47302–47318, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/ce806d8b4bf38cd92d483e5a0490d983-Paper-Conference.pdf.

Jiahao Xiong, Yihe Liu, Xianming Hu, Hongbo Zhao, Nuoyi Chen, Jie Zhang, and Kai Zhang. Localized low-rank adaptation within clustered parameter subspaces. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26556–26572, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1223. URL https://aclanthology.org/2026.acl-long.1223/.

Jia-Chen Zhang, Yu-Jie Xiong, He-Xi Qiu, Dong-Hai Zhu, and Chun-Ming Xia. Lora^2 : Multi-scale low-rank approximations for fine-tuning large language models, 2024a. URL https://arxiv.org/abs/2408.06854.

Jingfan Zhang, Yi Zhao, Dan Chen, Xing Tian, Huanran Zheng, and Wei Zhu. MiLoRA: Efficient mixture of low-rank adaptation for large language models fine-tuning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17071–17084, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.994. URL https://aclanthology.org/2024.findings-emnlp.994/.

Mengqi Zhang, Zisheng Zhou, Xiaotian Ye, Qiang Liu, Zhaochun Ren, Zhumin Chen, and Pengjie Ren. Disentangling knowledge representations for large language model editing. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 81086–81111, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/835ced593e668e39f19c9ad2f4ee5683-Paper-Conference.pdf.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. When do you need billions of words of pretraining data? In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 1112–1125, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.90. URL https://aclanthology.org/2021.acl-long.90/.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. LLaMA-MoE: Building mixture-of-experts from LLaMA with continual pre-training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15913–15923, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.890. URL https://aclanthology.org/2024.emnlp-main.890/.

Yun Zhu, Nevan Wichers, Chu-Cheng Lin, Xinyi Wang, Tianlong Chen, Lei Shu, Han Lu, Canoee Liu, Liangchen Luo, Jindong Chen, and Lei Meng. Sira: Sparse mixture of low rank adaptation, 2023. URL https://arxiv.org/abs/2311.09179.

6 APPENDIX

6.1 DEFINITION OF ZIGZAG ALGEBRA

Followed by Huerfano and Khovanov (2001),

Definition 6.1 (Zigzag algebra). *Let $\Gamma = (V, E)$ be a connected graph without loops or multiple edges. Its double $D\Gamma$ is obtained by replacing every edge $\{a, b\} \in E$ with two oppositely oriented arrows $a \to b$ and $b \to a$.*

For $|V| > 2$, let $\mathbb{C}D\Gamma$ denote the path algebra of $D\Gamma$. The zigzag algebra of Γ is

$$A(\Gamma) = \mathbb{C}D\Gamma/I,$$

where I is the two-sided ideal generated by the following quadratic relations:

$$(a|b|c) = 0 \qquad \text{if } a \sim b,\ b \sim c,\ a \neq c, \tag{Z1}$$

and

$$(a|b|a) = (a|c|a) \qquad \text{if } b \sim a,\ c \sim a. \tag{Z2}$$

If Γ consists of a single vertex, define

$$A(\Gamma) = \mathbb{C}[X]/(X^2), \qquad \deg X = 2.$$

If Γ consists of two vertices joined by one edge, define $A(\Gamma)$ as the path algebra of $D\Gamma$ quotiented by all paths of length greater than two.

Proposition 6.2 (Grading and path length). *For $|V| > 1$, $A(\Gamma)$ is naturally graded by path length. Its only nonzero graded components occur in degrees 0, 1, and 2. Hence every path of length at least three vanishes in $A(\Gamma)$.*

Proposition 6.3 (Graded symmetry). *The zigzag algebra $A(\Gamma)$ is a graded symmetric algebra. In particular, it admits a nondegenerate symmetric trace*

$$\mathrm{tr} : A(\Gamma) \to \mathbb{C}$$

that maps each length-two return path to 1 and all other paths to 0.

Corollary 6.4 (A_3 zigzag relations). *For the three-vertex chain*

$$1 \rightleftarrows 2 \rightleftarrows 3,$$

the defining relations become

$$A_{23}A_{12} = 0, \qquad A_{21}A_{32} = 0, \qquad A_{12}A_{21} = A_{32}A_{23}.$$

Consequently, every composable path of length at least three vanishes.

6.2 ORTHOGONALITY VS. ZIGZAG STRUCTURE

A natural question is whether the improvement of Zigzag LoRA can be explained simply by separating rank groups through orthogonality. We therefore perform a dedicated comparison against two orthogonality-based alternatives in Table 6.

The first baseline, denoted *Orthogonal*, operates in the same activation space as our method. It uses the same LoRA-A answer-position activations and the same rank-group partitions, but replaces the zigzag-algebraic objective with a standard pairwise orthogonality penalty between groups. This comparison isolates whether merely encouraging the grouped representations to become mutually orthogonal is sufficient to obtain the observed gains.

The second baseline, denoted *Parameter-space Orthogonal*, instead directly enforces orthogonality on the trainable LoRA parameter subspaces themselves. This baseline separates the trainable directions geometrically rather than imposing orthogonality on the activation representations, and is evaluated on MMLU-Pro.

As shown in Table 6, Zigzag LoRA consistently outperforms the activation-space orthogonal baseline across all tested rank sizes and group configurations. At rank 24 with three groups, for example,

Zigzag LoRA achieves 43.12% on MMLU-Pro compared with 37.92% for the orthogonal baseline. The same pattern remains at larger ranks: with rank 72 and six groups, the corresponding accuracies are 44.00% and 40.00%.

Direct parameter-space orthogonalization is stronger than the activation-space orthogonal baseline, reaching 40.53%, 41.35%, and 42.95% at ranks 24, 36, and 72, respectively. Nevertheless, it also remains below the corresponding Zigzag LoRA results.

These comparisons indicate that the benefit of our method cannot be reproduced by orthogonal separation alone. Orthogonality primarily suppresses overlap between groups, whereas the zigzag objective additionally constrains how inter-group relations compose: non-returning paths are suppressed, return cycles are related, and higher-order compositions are controlled. The improvement therefore appears to arise from the richer relational structure imposed on the rank space rather than from group separation itself.

6.3 Theoretical Properties of Zigzag-Algebraic LoRA

6.3.1 Exact A_3 Zigzag Relations

For the doubled quiver

$$1 \rightleftarrows 2 \rightleftarrows 3,$$

the A_3 zigzag relations are

$$A_{23}A_{12} = 0, \tag{1}$$
$$A_{21}A_{32} = 0, \tag{2}$$
$$A_{12}A_{21} = A_{32}A_{23}. \tag{3}$$

Theorem 6.5 (Length-three annihilation)**.** *If Eqs. equation 1–equation 3 hold, every composable path of length three vanishes.*

Proof. Paths containing $1 \to 2 \to 3$ or $3 \to 2 \to 1$ vanish directly.

$$A_{12}A_{21}A_{12} = A_{32}A_{23}A_{12} = 0,$$

and

$$A_{32}A_{23}A_{32} = A_{12}A_{21}A_{32} = 0.$$

The remaining cases follow analogously. □

Corollary 6.6 (Bounded relational depth)**.** *Under the exact relations, nonzero paths have length at most two.*

However, since ours apply zigzag algebra within training loss objectives, we use $\mathcal{L}_{\text{long}}$ directly to regularize a property implied by the exact quadratic relations.

6.3.2 Cross-Moment Realization

Recall

$$A_{ij} = \frac{1}{n} X_j^\top X_i, \qquad X_i \in \mathbb{R}^{n \times r_i}.$$

Proposition 6.7 (Cross-moment realization)**.** *Each arrow satisfies*

$$A_{ij} \in \mathbb{R}^{r_j \times r_i},$$

and

$$A_{21} = A_{12}^\top, \qquad A_{23} = A_{32}^\top.$$

Table 6: Comparison between Zigzag LoRA, an orthogonal baseline, and explicit parameter-space orthogonalization across different rank sizes and group configurations. Parameter-space orthogonalization is evaluated only on MMLU-Pro.

Method	Rank / Groups	MMLU-Pro	MMLU	OpenBookQA	SciQ
Zigzag (Ours)	24 / 3	43.12	71.79	89.4	94.7
	24 / 6	42.34	71.97	90.4	94.9
	36 / 3	42.95	71.92	89.8	94.9
	36 / 6	42.94	71.89	89.2	94.8
	72 / 3	43.73	72.18	90.2	95.0
	72 / 6	44.00	72.62	89.8	95.3
	126 / 3	45.79	72.73	90.0	95.0
	126 / 6	45.66	72.90	90.0	95.3
Orthogonal	24 / 3	37.92	70.37	86.8	94.0
	24 / 6	38.07	70.52	87.0	93.7
	36 / 3	38.71	70.64	88.2	93.9
	36 / 6	38.64	70.72	87.8	93.9
	72 / 3	40.23	71.31	88.8	94.2
	72 / 6	40.00	71.26	88.4	94.3
	126 / 3	42.16	71.99	89.8	94.7
	126 / 6	41.66	71.85	90.0	94.7
Parameter-space Orthogonal	24 / –	40.53	–	–	–
	36 / –	41.35	–	–	–
	72 / –	42.95	–	–	–

Hence,

$$A_{21}A_{32} = (A_{23}A_{12})^\top,$$

and

$$\phi(A_{12}) = \phi(A_{21}), \qquad \phi(A_{32}) = \phi(A_{23}), \qquad \mathcal{L}_{123} = \mathcal{L}_{321}.$$

Proof. The claims follow from

$$X_j^\top X_i \in \mathbb{R}^{r_j \times r_i}$$

and invariance of the Frobenius norm under transpose. □

In particular,

$$A_{23}A_{12} \in \mathbb{R}^{r_3 \times r_1}, \qquad A_{21}A_{32} \in \mathbb{R}^{r_1 \times r_3},$$

while both return cycles lie in $\mathbb{R}^{r_2 \times r_2}$. Thus unequal group sizes are allowed.

6.3.3 Soft Relations and Higher-Order Control

Define

$$C = A_{12}, \qquad D = A_{32}, \qquad R = D^\top C, \qquad Q = CC^\top - DD^\top.$$

Lemma 6.8 (Cubic residual control)**.**

$$CC^\top C = QC + DR,$$

and

$$\|CC^\top C\|_F \le \|Q\|_F\|C\|_2 + \|D\|_2\|R\|_F.$$

Likewise,

$$DD^\top C = DR, \qquad CC^\top D = CR^\top, \qquad DD^\top D = CR^\top - QD.$$

Proof.

$$CC^\top C = (CC^\top - DD^\top)C + DD^\top C = QC + DR.$$

The bound follows from the triangle inequality and submultiplicativity. □

Corollary 6.9 (Approximate higher-order control)**.** *If the primitive arrow norms remain bounded and*

$$\|R\|_F, \|Q\|_F \to 0,$$

then all length-three compositions converge to zero.

Thus $\mathcal{L}_{\text{long}}$ supplies direct finite-step control before the quadratic relations converge.

6.3.4 Extension to A_K and General Graphs

Let

$$X = [X_1 \mid \cdots \mid X_K], \qquad \sum_{i=1}^{K} r_i = r.$$

Proposition 6.10 (Graph extension)**.** *For a finite graph $\Gamma = (V, E)$, assign*

$$X_i \in \mathbb{R}^{n\times r_i}$$

to each vertex and define

$$A_{ij} = \frac{1}{n} X_j^\top X_i, \qquad \{i, j\} \in E.$$

Then

$$A_{jk}A_{ij} \approx 0, \qquad i \neq k,$$

and

$$A_{ji}A_{ij} \approx A_{ki}A_{ik}, \qquad j, k \sim i,$$

are dimensionally well-defined.

Proof. Since

$$A_{ij} \in \mathbb{R}^{r_j\times r_i},$$

adjacent arrows compose naturally and each return cycle is an endomorphism of $\mathbb{R}^{r_i}$. □

For A_K,

$$A_{i,i+1} = \frac{1}{n} X_{i+1}^\top X_i, \qquad A_{i+1,i} = A_{i,i+1}^\top.$$

For sparse Γ, cross-moment computation scales as

$$O\left(n \sum_{\{i,j\} \in E} r_i r_j \right).$$

Corollary 6.11 (No inference-time overhead)**.** *When used only in the training objective, deterministic arrows formed from existing LoRA activations or weights introduce neither additional trainable parameters nor inference-time modules.*

Hence,

$$A_3 \longrightarrow A_K \longrightarrow Z(\Gamma).$$

6.4 Inducing Algebraic Paths from the Trainable LoRA Space

Vertices and paths are induced directly from trainable LoRA representations:

$$X_i = X_i(\theta; x)$$

for activations, and

$$X_i = B_i(\theta)$$

for weights. Therefore,

$$A_{ij} = \frac{1}{n} X_j^\top X_i$$

and all paths are deterministic functions of existing trainable representations.

Conceptually,

$$\text{task learning} \longrightarrow \text{represented information},$$

whereas

$$\text{algebraic structure} \longrightarrow \text{relations among groups}.$$

Proposition 6.12 (End-to-end induced structure)**.** *If $X_i(\theta; x)$ is differentiable in θ, then every finite path*

$$P(\theta; x) = A_{i_k i_{k+1}} \cdots A_{i_1 i_2}$$

and differentiable loss $\ell(P)$ are differentiable in θ without additional trainable algebraic parameters.

Proof. Each arrow is a differentiable matrix product, so gradients propagate through

$$P \to A_{ij} \to X_i \to \theta.$$

□

Thus task and algebraic gradients optimize the same LoRA parameters.

Proposition 6.13 (Low-rank computational realization)**.** *For*

$$X_i \in \mathbb{R}^{n\times r_i}, \qquad X_j \in \mathbb{R}^{n\times r_j},$$

forming

$$A_{ij} = X_j^\top X_i/n$$

costs $O(nr_ir_j)$, while path products operate only in the LoRA rank dimensions.

The construction therefore uses dense low-rank matrix operations without geodesic computation, iterative manifold projection, or a separate algebraic solver.

For activations,

$$A_{ij}(\theta;x) = \frac{1}{n} X_j(\theta;x)^\top X_i(\theta;x),$$

so the relations depend on both the current model and data rather than prescribing vertex semantics.

data learns the representation + algebra organizes its relational flow.

Corollary 6.14 (No inference-time structural module)**.** *If arrows and paths are used only in the training loss, no additional inference-time module is required.*

6.5 Relation to Orthogonality and Expert-Based Separation

Orthogonality, expert decomposition, and routing all introduce forms of separation, but they do not by themselves define a structured relational organization among parameter groups. Zigzag algebra instead specifies which interactions are primitive and how those interactions may compose.

6.6 Orthogonality Versus Relational Structure

A typical orthogonality constraint takes the form

$$X_i^\top X_j \approx 0, \qquad i \neq j.$$

This encourages pairwise geometric separation, but assigns essentially the same relation—separation—to every constrained pair.

Proposition 6.15 (Pairwise orthogonality is relation-agnostic)**.** *If*

$$X_i^\top X_j = 0, \qquad i \neq j,$$

then any permutation of the group indices preserves the complete set of orthogonality constraints. Hence orthogonality alone does not define an intrinsic ordering, neighborhood, or hierarchy among the groups.

Proof. For any permutation π,

$$X_{\pi(i)}^\top X_{\pi(j)} = 0$$

whenever $i \neq j$. Therefore the constraint cannot distinguish one relational arrangement of the groups from another. □

As the number of groups grows, fully pairwise orthogonalization involves up to

$$\binom{K}{2} = O(K^2)$$

pairwise constraints, while still encoding only mutual separation. It does not distinguish

$$1 \to 2, \qquad 2 \to 3, \qquad 1 \to 2 \to 3.$$

Zigzag algebra instead acts on

$$A_{12}, \qquad A_{23}, \qquad A_{23}A_{12},$$

so local interactions and their compositions are represented as distinct algebraic objects.

6.7 Experts and Routing

A generic expert model is

$$F(x) = \sum_{k=1}^{K} g_k(x) F_k(x).$$

Expert decomposition provides distinct learned parameter modules, while the coefficients $g_k(x)$ determine how strongly each expert contributes to the output. However, this decomposition does not by itself specify relations among the experts.

Proposition 6.16 (Expert separation does not specify relational composition)**.** *For the standard expert mixture*

$$F(x) = \sum_{k=1}^{K} g_k(x) F_k(x),$$

the model specifies individual expert functions and their contributions, but does not explicitly define pairwise relations between experts or compositions of such relations.

Proof. The mixture depends on the expert outputs

$$F_k(x)$$

and their coefficients

$$g_k(x).$$

Its standard formulation contains no explicit inter-expert map

$$R_{ij} : F_i \to F_j$$

and therefore no composed relation such as

$$R_{jk} R_{ij}.$$

Such relational structure requires an additional mechanism beyond expert separation and aggregation.

□

Thus experts may learn distinct functions, but their separation alone does not distinguish structures such as

$$1 \to 2, \qquad 2 \to 3, \qquad 1 \to 2 \to 3.$$

Expert scaling determines the contribution of each module rather than how different modules are related or how those relations compose.

Input-dependent routing further introduces

$$g(x) = \mathrm{Router}_{\psi}(x),$$

which allows conditional expert selection.

Table 7: Common training configuration.

Setting	Value
Backbone	Qwen2.5-7B-Instruct
Training data	MMLU auxiliary_train
Training samples	99,842
Target modules	gate_proj, up_proj, down_proj
LoRA rank	24
LoRA α	24
LoRA dropout	0.0
Batch size	6
Gradient accumulation	5
Effective batch size	30
Learning rate	1×10^{-5}
Optimizer	AdamW
LR scheduler	Linear
Warmup steps	0
Weight decay	0
Max. gradient norm	1.0
Scheduler horizon	600 steps
Training stop step	500
Maximum sequence length	16,384
Evaluation batch size	32
Random seed	42
Frozen backbone dtype	BF16
Trainable parameter dtype	FP32

6.8 Separation Versus Organization

$$\begin{aligned} \text{orthogonality} &: \text{separates directions,} \\ \text{experts/scaling} &: \text{separates modules and weights contributions,} \\ \text{routing} &: \text{selects modules,} \\ \text{zigzag relations} &: \text{organize inter-group relations and compositions.} \end{aligned}$$

Corollary 6.17 (Separation and relational organization are distinct)**.** *Geometric or architectural separation alone does not specify how parameter groups are related or how their relations compose. Relational organization is therefore distinct from parameter separation, contribution weighting, or module selection.*

Our objective is not intended merely to produce distinct parameter groups. Instead, it assigns relational roles to those groups: groups act as vertices, pairwise interactions as arrows, and multi-step interactions as paths subject to distinct algebraic constraints.

6.9 Experimental Settings

6.9.1 Training Setup

All experiments use Qwen2.5-7B-Instruct as the backbone. We train on the MMLU auxiliary training split and use the same optimization setup across all adaptation methods whenever applicable. The frozen backbone is maintained in BF16, while trainable adapter and routing parameters are maintained in FP32.

LoRA is applied to the `gate_proj`, `up_proj`, and `down_proj` projections. The default rank is $r = 24$ with $\alpha = 24$ and zero LoRA dropout.

Training uses completion-only supervised fine-tuning. Each example is formatted as a multiple-choice question using the backbone chat template, and the supervised target consists of the correct answer label.

Table 8: Evaluation datasets and splits used in the main experiments.

Dataset	Split	Samples
MMLU-Pro	test	12,032
MMLU	test	14,042
SciQ	test	1,000

Table 9: Implementation of the compared adaptation methods.

Method	Implementation
Ours	Standard PEFT LoRA with the proposed zigzag loss applied to three rank-space groups. The main configuration partitions rank 24 equally into three groups of size 8.
Plain LoRA	Standard Hugging Face PEFT LoRA under the shared training configuration.
DoRA	Hugging Face PEFT implementation with `use_dora=True`.
OLoRA	Hugging Face PEFT implementation using `init_lora_weights=olora`.
SOS-LoRA	SOS-LoRA implementation with four experts and total-rank mode. Its method-specific orthogonal-subspace and scaling settings are retained, while the shared training protocol is matched to ours.
MixLoRA	MoE-PEFT implementation of MixLoRA with eight experts and top-2 routing. The expert ranks are chosen so that their total LoRA rank matches the rank-24 reference budget.
MoLA	MoLA routing is implemented following the method, using $2/4/6/8$ experts across successive depth regions with top-2 routing. Per-layer expert ranks are selected to match the rank-24 LoRA core parameter budget.

6.9.2 Evaluation Data

Table 8 summarizes the evaluation sets used in the main experiments. We use the complete evaluation split for each benchmark without subsampling.

For evaluation, the model is given the multiple-choice prompt and we compare the next-token logits corresponding to the valid answer labels. The label with the largest logit is selected as the prediction, so no free-form generation or answer parser is used.

6.9.3 Baseline Implementations

We match the layer-wise LoRA core rank budget to the rank-24 reference configuration whenever the method permits it.

Method-specific components such as DoRA magnitude parameters, orthogonal initialization, expert scaling, routing, and auxiliary losses are retained rather than removed for parameter-count equality. The common LoRA core budget and training protocol are matched across methods, while method-specific overhead is treated as part of the corresponding method.