PSR-LoRA: Power-Similarity Regularized Low-Rank Adaptation

Anonymous authors
Paper under double-blind review

Abstract

Low-Rank Adaptation (LoRA) has become a widely adopted approach for parameter-efficient fine-tuning of large language models. However, its rank components are typically optimized jointly without explicitly modeling their mutual dependencies, which may cause multiple components to learn highly correlated adaptation patterns and lead to inefficient utilization of the limited rank budget. In this work, we propose Power-Similarity Regularized LoRA (PSR-LoRA), a parameter-efficient adaptation framework that explicitly models and regulates the relationships among rank-wise update components. Specifically, PSR-LoRA first decomposes the LoRA update into rank-one components and maps them into a power-transformed space using a signed power function controlled by an exponent hyperparameter. Pairwise dependencies among the transformed components are then characterized through rank-wise similarity modeling. Based on the resulting similarity structure, we introduce a redundancy regularization objective to suppress excessive dependencies among rank components and further derive similarity-guided weights to adaptively adjust their contributions to the final low-rank update. To maintain efficiency, the proposed similarity computation is performed directly on the low-rank factors without explicitly constructing full rank-one update matrices. Extensive experiments across multiple downstream tasks demonstrate that PSR-LoRA consistently improves adaptation performance under constrained trainable parameter budgets while inducing less redundant rank-wise adaptation patterns. These results show that explicitly organizing the internal structure of a fixed low-rank space provides an effective alternative to simply increasing the LoRA rank.

1 Introduction

Large language models (LLMs) have demonstrated strong generalization capabilities across a wide range of downstream tasks (Naveed et al., 2025; Ling et al., 2025). However, adapting increasingly large pretrained models to task-specific data through full-parameter fine-tuning is computationally expensive and requires substantial memory and storage. Parameter-efficient fine-tuning (PEFT) has therefore emerged as an attractive alternative, where only a small subset of parameters is optimized while the pretrained backbone remains frozen (Wang et al., 2025a; Xu et al., 2026). Among existing parameter-efficient fine-tuning (PEFT) methods, Low-Rank Adaptation (LoRA) (Hu et al., 2021) has become one of the most widely adopted approaches because of its simple formulation, low optimization cost, and compatibility with weight merging at inference time (Shuttleworth et al., 2026). Rather than updating the full pretrained weight matrix, LoRA introduces a compact low-rank parameterization of task-specific updates, substantially reducing the number of trainable parameters while maintaining competitive adaptation performance Li et al. (2026a); Lun & Ming (2026).

Specifically, LoRA assumes that task-specific parameter updates can be approximated within a low-dimensional subspace. Given a frozen weight matrix W_0, LoRA parameterizes its update as $\Delta W = BA$, where $A \in \mathbb{R}^{r\times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}}\times r}$, and the rank r is substantially smaller than the input and output dimensions. Although this formulation effectively constrains the dimensionality of the adaptation space, it imposes little explicit structure on the relationships among individual rank components (Han et al., 2026; Lee et al., 2026). Since $BA = \sum_{k=1}^{r} b_k a_k^\top$, a rank-$r$ LoRA update can naturally be interpreted as the aggregation of r rank-one adaptation components. These components are jointly optimized by the downstream objective, but their mutual relationships are not explicitly

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.

modeled (Wei et al., 2026; Lin et al., 2025). Consequently, different components may learn partially overlapping or correlated adaptation patterns, resulting in inefficient utilization of the limited rank budget. Increasing r provides additional adaptation capacity, but simultaneously increases the number of trainable parameters and does not inherently guarantee that the newly introduced components encode sufficiently distinct information Xu et al. (2026).

Recent LoRA variants improve parameter-efficient adaptation from several complementary perspectives. AdaLoRA Zhang et al. (2023) dynamically allocates rank budgets according to parameter importance, while LoRA-drop Zhou et al. (2025) identifies influential LoRA modules based on their output responses and reduces redundant layer-wise parameters through selective retention and sharing. L2-LoRA Zhang et al. (2026) introduces layer-specific regularization to balance downstream adaptation with preservation of pretrained knowledge, and Stable-LoRA Wu et al. (2026) improves feature-learning stability during optimization. Collectively, these methods improve where adaptation capacity is allocated, which layers are emphasized, and how low-rank parameters are regularized or optimized.

Despite these advances, the rank dimension itself is still largely treated as an undifferentiated adaptation space, with individual rank-one components jointly optimized without explicitly modeling their mutual dependencies (Li et al., 2026a; Refael et al., 2025). Consequently, improvements in rank allocation, layer selection, or optimization do not necessarily prevent different components from learning overlapping adaptation patterns. This motivates a complementary perspective where, rather than treating rank solely as a capacity hyperparameter, we explicitly model the geometric relationships among rank-one components and encourage them to encode complementary adaptation directions, which leads to a fundamental question: **Can LoRA achieve more effective adaptation by explicitly regulating dependencies among its rank-one components rather than relying solely on increasing, reallocating, or optimizing the rank budget?**

Addressing this question is non-trivial. Similarity measured directly in the original parameter space mainly captures linear relationships and may miss richer structural differences among rank components. Meanwhile, explicitly constructing each rank-one matrix $b_k a_k^\top$ incurs substantial memory and computation overhead. An effective solution should therefore model rank-wise dependencies in a richer space without sacrificing LoRA's efficiency. To this end, we propose **Power-Similarity Regularized LoRA (PSR-LoRA)**, which explicitly models nonlinear relationships among rank-one adaptation components. For each component $M_k = b_k a_k^\top$, we apply a signed power transformation $\phi_p(x) = \mathrm{sign}(x)|x|^p$, where p controls the geometry of the transformed space by amplifying or compressing parameter variations of different magnitudes. PSR-LoRA then constructs a rank-wise similarity structure in this transformed space and uses it to reduce redundancy and promote complementary adaptation directions.

Importantly, PSR-LoRA avoids explicitly materializing the full rank-one matrices. The similarity between transformed components can be factorized into operations on their corresponding low-rank vectors a_k and b_k, preserving the parameter and computational efficiency of standard LoRA. The learned update can still be merged into the pretrained weights after training, introducing no additional inference-time modules. Conceptually, PSR-LoRA shifts the focus from how much rank capacity is available to how effectively that capacity is organized, aiming to obtain more informative adaptation from a fixed low-rank budget.

Our main contributions are summarized as follows:

1. We revisit LoRA from a rank-wise perspective by decomposing its low-rank update into a set of rank-one adaptation components. In existing LoRA variants, dependencies among individual rank components remain largely implicit. This motivates us to explicitly model inter-component relationships and improve the utilization of a fixed rank budget.
2. We propose PSR-LoRA, which introduces a signed power transformation to construct a nonlinear representation space for rank-wise dependency modeling. By adjusting the power exponent p, PSR-LoRA can flexibly reshape the geometry of the adaptation space and capture richer relationships among rank-one components than direct similarity measurement in the original parameter space.
3. We develop two complementary mechanisms based on the learned rank-wise similarity structure: similarity-based redundancy regularization and similarity-guided component weighting. The former discourages highly dependent rank components, while the latter

dynamically assigns larger contributions to more distinctive directions. Together, they encourage the limited rank dimensions to encode more complementary adaptation information.

4. We compute the similarity between power-transformed rank-one updates directly from their corresponding low-rank factors, without explicitly constructing full-dimensional matrices. This reduces the computational and memory overhead of rank-wise relational modeling while preserving LoRA's parameter efficiency and inference-time mergeability. Extensive experiments further show that PSR-LoRA improves downstream adaptation under constrained parameter budgets and produces less redundant rank representations.

2 METHOD

2.1 MOTIVATION

Low-Rank Adaptation (LoRA) represents task-specific updates using two trainable low-rank matrices while keeping pretrained weights frozen:

$$W = W_0 + \Delta W, \qquad \Delta W = \frac{\alpha}{r} BA, \tag{1}$$

where $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}} \times r}$, and $r \ll \min(d_{\text{in}}, d_{\text{out}})$. Although LoRA is parameter-efficient, its rank dimensions are jointly optimized without explicitly modeling their mutual dependencies. As a result, different components may learn overlapping adaptation patterns, leading to redundant rank capacity. Moreover, directly characterizing such dependencies in the full update space is computationally expensive.

To address these limitations, we propose **PSR-LoRA**, which explicitly models relationships among LoRA rank components in a nonlinear power space. The resulting similarity structure is used to regularize and adaptively modulate component contributions, encouraging more complementary adaptation directions while preserving the efficiency of standard LoRA. Figure 1 presents the overall architecture of PSR-LoRA. The left part shows that PSR-LoRA is integrated into the LoRA adapters of a frozen Transformer-based LLM, while the right part details its optimization pipeline. Specifically, the LoRA update is first decomposed into rank-one components, which are then transformed into a nonlinear power space for similarity modeling. Based on the resulting rank-wise similarity structure, PSR-LoRA performs redundancy-aware regularization and adaptive component weighting, leading to a more effective low-rank update. This design enhances rank utilization without changing the standard LoRA inference mechanism.

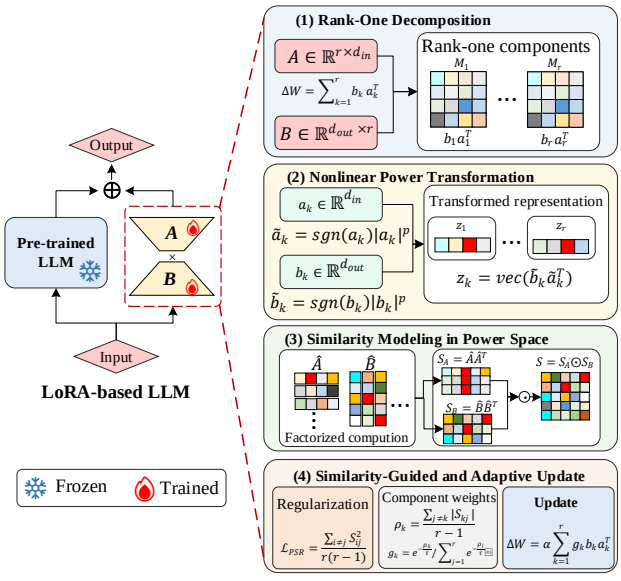


Figure 1: The architecture of PSR-LoRA. It includes four modules, (1) Rank-One Decomposition, (2) Nonlinear Power Transformation, (3) Similarity Modeling in Power Space, (4) Similarity-Guided and Adaptive Update.

2.2 RANK-ONE DECOMPOSITION

To expose the internal structure of the low-rank update, let $a_k^\top$ denote the k-th row of A and b_k the k-th column of B. Then

$$BA = \sum_{k=1}^{r} b_k a_k^\top. \tag{2}$$

We define $M_k = b_k a_k^\top$ as the k-th rank-one adaptation component, yielding $\Delta W = \frac{\alpha}{r} \sum_{k=1}^{r} M_k$. This decomposition provides the basis for component-level relational modeling in PSR-LoRA. Each M_k represents one adaptation component, and the following sections characterize their nonlinear geometric relationships.

2.3 POWER-SPACE RANK REPRESENTATION

Direct similarity measurement in the original parameter space evaluates rank components under the geometry induced by their raw parameter values, without explicitly controlling the relative influence of coefficients with different magnitudes. To obtain a more flexible geometry for dependency modeling, we introduce the signed power transformation

$$\phi_p(x) = \mathrm{sgn}(x)|x|^p, \quad p > 0, \tag{3}$$

which is applied element-wise. sgn denotes the sign function. The transformed representation of the k-th component is

$$z_k = \mathrm{vec}\left(\phi_p(M_k)\right). \tag{4}$$

The exponent p controls the geometry of the resulting *power adaptation space*. When $p = 1$, the representation reduces to the original parameter space. For $p > 1$, large-magnitude coefficients receive greater emphasis, whereas $0 < p < 1$ compresses magnitude differences and increases the relative influence of weaker coefficients. The signed formulation preserves the polarity of LoRA parameters and remains valid for non-integer p.

2.4 RANK-WISE SIMILARITY AND REGULARIZATION

Given transformed representations $\{z_k\}_{k=1}^{r}$, we construct a rank-wise similarity matrix $S \in \mathbb{R}^{r \times r}$. By default, we use cosine similarity:

$$S_{ij} = \frac{z_i^\top z_j}{\|z_i\|_2 \|z_j\|_2 + \epsilon}. \tag{5}$$

Large $|S_{ij}|$ indicates strong dependence between two adaptation components, whereas values close to zero indicate relatively distinct directions. More generally, the similarity operator can be written as $S_{ij} = \kappa(z_i, z_j)$, where $\kappa(\cdot, \cdot)$ denotes a differentiable dependence measure. In this work, we adopt cosine similarity as the default operator because it admits an exact and efficient factorized computation over the low-rank factors. Other measures, such as Pearson correlation, can also be considered but may require operator-specific formulations. To discourage redundant rank components, we introduce the power-similarity regularizer

$$\mathcal{L}_{\mathrm{PSR}} = \frac{1}{r(r-1)} \sum_{i \neq j} S_{ij}^2. \tag{6}$$

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{PSR}} \mathcal{L}_{\mathrm{PSR}}, \tag{7}$$

where λ_{PSR} controls the strength of rank-wise diversity regularization. This objective encourages different rank components to encode complementary adaptation patterns without changing the standard LoRA forward structure.

2.5 EFFICIENT FACTORIZED SIMILARITY

Explicitly constructing $M_k \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$ for every rank component would introduce unnecessary memory and computation overhead. The signed power transformation admits an exact factorization

because $\phi_p(xy) = \phi_p(x)\phi_p(y)$. Therefore, $\phi_p(M_k) = \phi_p(b_k)\phi_p(a_k)^\top$. Let $\tilde{b}_k = \phi_p(b_k), \tilde{a}_k = \phi_p(a_k)$. Using the identities

$$\langle uv^\top, xy^\top \rangle_F = (u^\top x)(v^\top y) \tag{8}$$

and $\|uv^\top\|_F = \|u\|_2\|v\|_2$, the cosine similarity between two transformed rank-one components becomes

$$S_{ij} = \cos(\tilde{b}_i, \tilde{b}_j)\cos(\tilde{a}_i, \tilde{a}_j). \tag{9}$$

The complete similarity matrix can be computed directly from the transformed low-rank factors:

$$S_A = \hat{A}\hat{A}^\top, \qquad S_B = \hat{B}^\top\hat{B}, \qquad S = S_A \odot S_B, \tag{10}$$

where $\hat{A}$ and $\hat{B}$ denote normalized transformed factors and $\odot$ is the Hadamard product. This formulation avoids materializing full rank-one matrices and reduces the similarity computation from the full parameter space to the low-rank factors.

2.6 Similarity-Guided Component Modulation

The learned similarity structure can further be used to adaptively control the contribution of each rank component. We define the redundancy score of component k as

$$\rho_k = \frac{1}{r-1}\sum_{j\neq k}|S_{kj}|. \tag{11}$$

Components with larger ρ_k share stronger dependencies with other ranks and are therefore assigned smaller weights:

$$g_k = \frac{\exp(-\rho_k/\tau)}{\sum_{j=1}^{r}\exp(-\rho_j/\tau)}, \tag{12}$$

where τ controls the sharpness of the weighting distribution. The LoRA update is then written as

$$\Delta W = \alpha\sum_{k=1}^{r} g_k b_k a_k^\top. \tag{13}$$

Since the softmax normalization guarantees $\sum_{k=1}^{r} g_k = 1$, the proposed formulation naturally recovers standard LoRA when all components receive equal importance, i.e., $g_k = 1/r$. Therefore, similarity-guided modulation can be interpreted as replacing the uniform rank weighting of standard LoRA with an adaptive, redundancy-aware weighting scheme. This mechanism assigns larger multiplicative weights to components with lower redundancy scores. Their contributions to the final update also depend on the magnitudes of the corresponding rank-one components. Since g_k is derived directly from the rank-wise similarity structure, it introduces no additional free parameters. PSR-LoRA jointly incorporates power-similarity regularization and similarity-guided component modulation. The former discourages redundant dependencies among rank components, while the latter adaptively adjusts their contributions according to the learned similarity structure.

3 Experiments

3.1 Datasets and Tasks

We evaluate PSR-LoRA on three domain-specific datasets, including FaultQA[1], Patent[2], and SMRT[3]. Rather than restricting the evaluation to a single downstream setting, we construct tasks with different semantic domains and output structures to examine whether the proposed rank-wise modeling strategy generalizes across adaptation scenarios.

For Patent and SMRT, we formulate the original classification problems as question-answering-style classification tasks. Given a domain-specific textual description, the model is prompted to identify the corresponding category and generate the target label as its answer. The patent dataset evaluates semantic categorization of patent descriptions, whereas the maintenance-log dataset focuses

[1]https://huggingface.co/datasets/zhengr/Automotive_Industry_Fault_Data_Set

[2]https://huggingface.co/datasets/charozvb/patent-ipc-classifier

[3]https://huggingface.co/datasets/ASNIJKDNI/SMRT-Maintenance-Logs

on identifying categories from industrial maintenance records. Although both are classification tasks, they differ substantially in domain terminology, semantic granularity, and input characteristics, providing complementary settings for evaluating low-rank adaptation. For FaultQA, we adopt a generative question-answering task. Given an automotive fault description, the model generates the corresponding target output rather than selecting a predefined class. This task therefore evaluates whether a PEFT method can preserve fine-grained domain information and produce accurate structured responses. Together, these tasks cover both discriminative QA-style classification and generative QA, allowing us to evaluate PSR-LoRA under heterogeneous downstream objectives rather than a single task formulation. This task diversity is particularly important for assessing whether improvements arise from more effective organization of low-rank adaptation capacity rather than task-specific optimization.

3.2 Backbone and Evaluation Metrics

We compare PSR-LoRA with standard LoRA and several recent LoRA variants, including AdaLoRA, LoRA-Pro, AuroRA, LoRA-drop, and L2-LoRA. These baselines cover complementary directions of LoRA improvement, including dynamic rank allocation, gradient optimization, nonlinear transformation, adapter selection, and layer-wise regularization. We use Qwen2.5-7B as the common pretrained backbone for all experiments and continuously adapt it to each downstream task using different PEFT methods. The pretrained backbone is kept identical across all comparisons, while only the corresponding adaptation parameters are optimized. To ensure a controlled comparison, PSR-LoRA and all baseline methods use the same backbone, task formulation, training split, evaluation protocol, and optimization configuration whenever applicable. In particular, the target modules and LoRA rank are kept consistent across methods unless a baseline requires a method-specific configuration. This setup isolates the effect of different low-rank adaptation strategies and prevents performance differences from being confounded by variations in backbone size or downstream training protocol.

For the QA-style classification tasks, we report Accuracy (Acc), Precision (Pre), Recall (Rec), and F1-score (F1). Accuracy measures overall prediction correctness, while Precision, Recall, and F1 provide complementary views of class-level prediction quality. For multi-class evaluation, Precision, Recall, and F1 are computed using macro averaging so that each category contributes equally to the final metric. For the automotive fault generation task (FaultQA), we evaluate generated responses using Normalized Edit Similarity (NES), ROUGE-L, METEOR, and BLEU-4. NES is obtained by subtracting the normalized edit distance from one, so a higher value indicates greater sequence-level similarity between the generated response and the reference. ROUGE-L evaluates longest-subsequence overlap. METEOR provides a more flexible token-level matching criterion, and BLEU-4 measures local n-gram consistency. Using these complementary metrics allows us to evaluate both structural fidelity and lexical agreement of the generated responses.

3.3 Main Results

Table 1 reports the main results of PSR-LoRA and the competing PEFT methods across the Patent, SMRT, and FaultQA tasks. PSR-LoRA achieves the best performance across all reported metrics on the three tasks, demonstrating the effectiveness of explicitly modeling relationships among rank components for both classification and text generation. For each method, the hyperparameter configuration is selected on the validation set using the same model-selection protocol.

On Patent, PSR-LoRA achieves the best results across all four metrics, reaching 0.7458 accuracy and 0.7446 F1, with absolute improvements of 0.0083 and 0.0084 over the strongest baseline, respectively. The consistent gains in precision and recall further indicate that the improvement is not confined to a single evaluation metric. The advantage becomes more pronounced on the more challenging SMRT task. PSR-LoRA reaches 0.7701 accuracy and 0.3845 F1, compared with 0.7396 and 0.3401 for the strongest baseline. This corresponds to absolute gains of 0.0305 in accuracy and 0.0444 in F1. Precision and recall also improve substantially, reaching 0.4002 and 0.4024, respectively. Notably, PSR-LoRA achieves these results with $r = 4$, while several baselines use substantially larger ranks, including $r = 8$ for LoRA and $r = 15$ for AuroRA, LoRA-drop, and AdaLoRA. This result suggests that strong adaptation performance does not necessarily require a large nominal rank when inter-component relationships are explicitly modeled.

Dataset	Metric	LoRA	L2-LoRA	LoRAPro	AuroRA	LoRA-drop	AdaLoRA	PSR-LoRA
Patent	Acc	0.7375	0.6792	0.6792	0.7167	0.7042	0.7208	**0.7458**
	Pre	0.7412	0.6854	0.6853	0.7197	0.7166	0.7254	**0.7497**
	Rec	0.7425	0.6776	0.6888	0.7246	0.7174	0.7206	**0.7481**
	F1	0.7362	0.6682	0.6793	0.7159	0.7041	0.7132	**0.7446**
	Rank	2	2	2	2	13	4	**4,** $p = 1.3$
SMRT	Acc	0.7396	0.6636	0.6432	0.6812	0.7183	0.6525	**0.7701**
	Pre	0.3523	0.2761	0.1918	0.2826	0.3084	0.2618	**0.4002**
	Rec	0.3597	0.2796	0.1999	0.2846	0.3142	0.2597	**0.4024**
	F1	0.3401	0.2632	0.1817	0.2726	0.2988	0.2496	**0.3845**
	Rank	8	4	2	15	15	15	**4,** $p = 0.8$
FaultQA	NES	0.5641	0.4510	0.5101	0.5028	0.5063	0.5124	**0.5735**
	ROUGE-L	0.6454	0.5483	0.5869	0.5805	0.5941	0.5925	**0.7085**
	METEOR	0.3001	0.2875	0.2165	0.2141	0.2117	0.2298	**0.6523**
	BLEU-4	0.5674	0.3396	0.4737	0.4528	0.4702	0.4613	**0.7757**
	Rank	2	5	8	12	13	14	**4,** $p = 0.7$

Table 1: Main results on Patent, SMRT, and FaultQA. The best results are highlighted in bold.

PSR-LoRA also shows clear advantages on the FaultQA task. It achieves the best results on NES, ROUGE-L, METEOR, and BLEU-4, with scores of 0.5735, 0.7085, 0.6523, and 0.7757, respectively. Compared with the strongest baseline, PSR-LoRA improves ROUGE-L by 0.0631 and BLEU-4 by 0.2083. The particularly large improvement in METEOR indicates that the generated responses exhibit substantially better lexical and semantic alignment with the reference outputs. These results show that the proposed rank-wise relational modeling is not limited to classification, but also transfers effectively to structured text generation. These results demonstrate that PSR-LoRA can achieve strong downstream performance with a relatively compact rank configuration, supporting the effectiveness of explicitly organizing rank-wise adaptation components.

3.4 Sensitivity Analysis of Power Exponent

Table 2 evaluates the sensitivity of PSR-LoRA to the power exponent (p). Since (p=1) corresponds to the original parameter geometry, (p<1) compresses magnitude differences, while (p>1) emphasizes large-magnitude coefficients. Overall, the preferred value of (p) varies across tasks, indicating that rank-wise similarity modeling benefits from task-dependent power-space geometry. On Patent, performance remains relatively stable across a broad range of (p). The best Recall and F1 are obtained at (p=0.9), while (p=1.3) gives the highest Precision; the best Accuracy is shared by (p=0.9), (1.2), and (1.3). This suggests that Patent is relatively insensitive to the exact exponent, although several non-unit settings outperform (p=1). On SMRT, the best results are concentrated in the sublinear region. Specifically, (p=0.8) achieves the highest Accuracy, while (p=0.7) obtains the best Precision, Recall, and F1. A similar pattern is observed on FaultQA, where (p=0.7) achieves the best ROUGE-L, METEOR, and BLEU-4, whereas (p=1.1) yields the highest NES. These results indicate that compressing magnitude differences is beneficial for SMRT and most FaultQA metrics. Overall, the consistently different optimal values across tasks show that the fixed (p=1) geometry is not universally optimal, supporting the role of power transformation in adapting rank-wise similarity modeling to different downstream tasks.

3.5 Performance vs. Rank

Figure 2 shows that PSR-LoRA is sensitive to the rank budget, but the performance gains gradually saturate as the rank increases. Across all three tasks, increasing the rank from 1 to 4 consistently improves the aggregated score. For Patent, the mean score rises from 0.7201 at $r = 1$ to 0.7471 at $r = 4$; similar trends are observed on SMRT and FaultQA, where the scores increase from 0.4472 to 0.4893 and from 0.6235 to 0.6775, respectively. However, further increasing the rank to 8 or 16 does not yield additional gains and instead results in slight performance degradation across all tasks. This suggests that once sufficient adaptation capacity is available, simply enlarging the nominal rank introduces limited benefit and may increase redundancy among rank components. Overall, the results support the motivation of PSR-LoRA: effective organization and diversification of rank components are more important than continuously increasing the rank budget.

Dataset	Metric	p = 0.6	p = 0.7	p = 0.8	p = 0.9	p = 1.0	p = 1.1	p = 1.2	p = 1.3
Patent	Acc	0.7417	0.7417	0.7333	**0.7458**	0.7375	0.7375	**0.7458**	**0.7458**
	Pre	0.7479	0.7440	0.7319	0.7495	0.7387	0.7442	0.7461	**0.7497**
	Recall	0.7480	0.7486	0.7408	**0.7525**	0.7447	0.7441	0.7503	0.7481
	F1	0.7422	0.7423	0.7310	**0.7472**	0.7353	0.7385	0.7440	0.7446
SMRT	Acc	0.7600	0.7702	**0.7720**	0.7646	0.7451	0.7451	0.7600	0.7600
	Pre	0.3738	**0.4002**	0.3902	0.3981	0.3610	0.3610	0.3738	0.3738
	Recall	0.3738	**0.4024**	0.3921	0.3948	0.3655	0.3655	0.3738	0.3738
	F1	0.3625	**0.3845**	0.3753	0.3807	0.3467	0.3467	0.3625	0.3625
FaultQA	NES	0.5642	0.5735	0.5310	0.5625	0.5840	**0.5865**	0.5669	0.5777
	ROUGE-L	0.6672	**0.7085**	0.6270	0.6739	0.6628	0.6560	0.6421	0.6880
	METEOR	0.6448	**0.6523**	0.6155	0.6458	0.3577	0.3288	0.3230	0.6495
	BLEU-4	0.6963	**0.7757**	0.6086	0.6915	0.5988	0.5810	0.5466	0.7152

Table 2: Sensitivity analysis of the power exponent p on different downstream tasks. The best result for each metric is highlighted in bold.

3.6 RANK SENSITIVITY

Figure 3 shows the rank sensitivity results. The PSR-LoRA not only achieves higher performance than standard LoRA, but is also considerably more robust to the choice of rank. On Patent, LoRA peaks at $r = 2$ and then degrades rapidly as the rank increases, whereas PSR-LoRA continues to improve until $r = 4$ and remains stable for larger ranks. On SMRT, the difference is even more pronounced: LoRA is highly unstable at small and large ranks, while PSR-LoRA maintains consistently strong performance across all settings. For FaultQA, PSR-LoRA also consistently outperforms LoRA by a clear margin at every rank. The advantage of PSR-LoRA can be explained by its explicit modeling of inter-component relationships. In standard LoRA, newly added rank dimensions are optimized jointly but without any mechanism to prevent them from learning overlapping directions. As the rank increases, this can introduce redundant components and make the effective adaptation capacity grow much more slowly than the nominal rank. In contrast, PSR-LoRA evaluates rank-wise dependencies in the power-transformed space and penalizes excessive similarity, encouraging different components to capture more complementary information. The similarity-guided weighting further suppresses highly redundant components and emphasizes more distinctive directions. As a result, PSR-LoRA can make more effective use of a moderate rank budget and is less sensitive to unnecessary increases in rank.

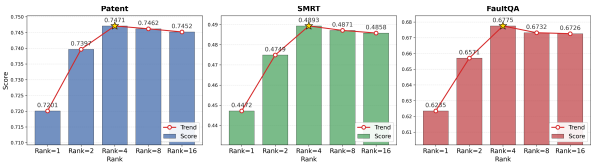


Figure 2: Performance vs. Rank on Three Tasks. ★ indicates the best result. The results report the arithmetic mean of the four evaluation metrics.

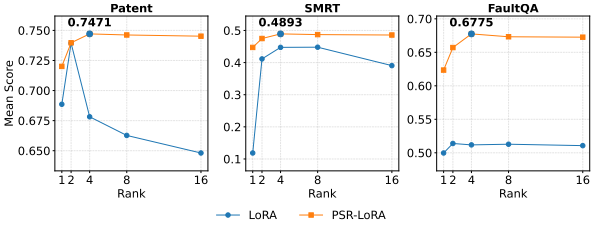


Figure 3: Rank Sensitivity Comparison, PSR-LoRA consistently maintains optimal performance.

3.7 SIMILARITY FUNCTION ANALYSIS

Table 3 compares four relation functions for modeling dependencies among power-transformed rank components. The cosine similarity and Pearson correlation consistently outperform Hamming similarity and the RBF kernel, indicating that continuous directional and correlation-based relationships are more suitable for characterizing the internal geometry of LoRA rank components. On the Patent task, cosine similarity achieves the best Accuracy and Recall, reaching 0.7458 and 0.7481, respectively, while Pearson correlation obtains the highest Precision and F1 scores of 0.7563 and 0.7508. A similar pattern is observed on SMRT: cosine provides the highest Accuracy and Recall, whereas

Pearson achieves the best Precision and F1. These results suggest that cosine and Pearson capture complementary properties of rank-wise dependencies. Cosine directly measures directional alignment, while Pearson additionally removes component-wise offsets and emphasizes correlated variation patterns.

On FaultQA, the two continuous correlation measures again provide the strongest generation performance. Cosine achieves the highest NES and METEOR scores, whereas Pearson obtains the best ROUGE-L and BLEU-4 scores. RBF remains competitive but consistently underperforms cosine and Pearson, suggesting that purely distance-based local similarity is less effective than directional dependency modeling for the proposed power adaptation space. Hamming similarity produces substantially lower performance across all three tasks. This is expected because its binary representation preserves only sign-pattern agreement and discards the magnitude and continuous geometric information contained in the power-transformed rank components. The performance degradation is particularly pronounced on Patent and FaultQA, indicating that fine-grained continuous relationships are important for effective rank redundancy modeling.

Dataset	Metric	Cosine	Pearson	Hamming	RBF
Patent	Acc	**0.7458**	0.7382	0.4500	0.7125
	Pre	0.7497	**0.7563**	0.4326	0.7286
	Recall	**0.7481**	0.7415	0.4441	0.7203
	F1	0.7446	**0.7508**	0.4331	0.7158
SMRT	Acc	**0.7701**	0.7623	0.6913	0.7303
	Pre	0.4002	**0.4126**	0.2876	0.3295
	Recall	**0.4024**	0.3947	0.2934	0.3531
	F1	0.3845	**0.3921**	0.2761	0.3265
FaultQA	NES	**0.5735**	0.5682	0.4836	0.5247
	ROUGE-L	0.7085	**0.7156**	0.4152	0.6738
	METEOR	**0.6523**	0.6458	0.4628	0.6154
	BLEU-4	0.7757	**0.7823**	0.3847	0.7426

Table 3: Performance comparison of different similarity functions in PSR-LoRA.

3.8 Sensitivity to λ_{PSR}

Table 4 examines the effect of the regularization coefficient λ_{PSR}. When $\lambda_{\mathrm{PSR}} = 0$, the model does not explicitly penalize inter-component redundancy, resulting in the highest mean similarity and relatively lower performance across all three tasks. Increasing λ_{PSR} initially improves task performance while progressively reducing inter-component similarity. The best overall results are obtained at $\lambda_{\mathrm{PSR}} = 10^{-2}$, where Patent, SMRT, and FaultQA reach mean scores of 0.7471, 0.4893, and 0.6775, respectively. When λ_{PSR} is further increased, the mean similarity continues to decrease, but downstream performance begins to deteriorate. This indicates that excessive decorrelation may over-constrain the adaptation space and suppress useful cooperation among rank components. Therefore, PSR-LoRA benefits from a moderate regularization strength that balances redundancy reduction with task-specific representation learning.

λ_{PSR}	Patent	SMRT	FaultQA	Mean Similarity S
0	0.7398	0.4764	0.6612	0.2841
1×10^{-3}	0.7446	0.4847	0.6718	0.2415
1×10^{-2}	**0.7471**	**0.4893**	**0.6775**	0.1987
5×10^{-2}	0.7439	0.4841	0.6710	0.1438
1×10^{-1}	0.7384	0.4769	0.6608	<u>0.1085</u>

Table 4: Sensitivity analysis of the regularization coefficient λ_{PSR}.

4 Conclusion

We propose PSR-LoRA, a parameter-efficient adaptation method that explicitly models dependencies among LoRA rank-one components. PSR-LoRA maps these components into a nonlinear power space, regularizes redundant directions, and adaptively reweights component contributions. An exact factorized formulation enables this relational modeling directly on the low-rank factors, preserving LoRA's efficiency and inference-time mergeability. Across classification and generation tasks, PSR-LoRA consistently outperforms standard LoRA and recent variants, while remaining robust to different rank settings. The ablations further show that nonlinear power-space modeling and similarity-aware regularization improve rank utilization, allowing strong performance with moderate rank budgets. These results suggest that the internal organization of low-rank components is an important, yet underexplored, dimension of parameter-efficient adaptation.

Future work could explore more efficient similarity estimation, adaptive hyperparameter selection, and cross-layer or cross-module relational modeling, as well as evaluate whether the same principles extend to larger models and broader adaptation settings.

AI USE STATEMENT

In this work, we did not use generative AI tools for any required disclosure tasks. No generative AI tools were used for any recommended disclosure tasks. All content of this manuscript was produced solely by the authors. We take responsibility for the final content of this work.

REFERENCES

Kefan Cao and Shuaicheng Wu. Orthogonal low-rank adaptation in lie groups for continual learning of large language models. *arXiv preprint arXiv:2509.06100*, 2025.

Yupeng Chang, Yi Chang, and Yuan Wu. Ba-lora: Bias-alleviating low-rank adaptation to mitigate catastrophic inheritance in large language models. In *International Conference on Learning Representations*, volume 2026, pp. 156585–156612, 2026a.

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)*, pp. 3993–4005, 2026b.

Haonan Dong, Wenhao Zhu, Guojie Song, and Liang Wang. Aurora: Breaking low-rank bottleneck of lora with nonlinear mapping. *Advances in Neural Information Processing Systems*, 38:36929–36961, 2026.

Xiao Han, Zimo Zhao, Wanyu Wang, Maolin Wang, Zitao Liu, Yi Chang, and Xiangyu Zhao. Data efficient adaptation in large language models via continuous low-rank fine-tuning. *Advances in Neural Information Processing Systems*, 38:165157–165182, 2026.

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.

Yu-Ang Lee, Ching-Yun Ko, Pin-Yu Chen, and Mi-Yen Yeh. Learning rate matters: Vanilla lora may suffice for llm fine-tuning. *arXiv preprint arXiv:2602.04998*, 2026.

Bingcong Li, Yilang Zhang, and Georgios B Giannakis. Low-rank adaptation redux for large models. *arXiv preprint arXiv:2604.21905*, 2026a.

Zhizhong Li, Sina Sajadmanesh, Jingtao Li, and Lingjuan Lyu. Stella: Subspace learning in low-rank adaptation using stiefel manifold. *Advances in Neural Information Processing Systems*, 38: 75066–75092, 2026b.

Tianwei Lin, Jiang Liu, Wenqiao Zhang, Yang Dai, Haoyuan Li, Zhelun Yu, Wanggui He, Juncheng Li, Jiannan Guo, Hao Jiang, et al. Teamlora: Boosting low-rank adaptation with expert collaboration and competition. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13622–13637, 2025.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, et al. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ACM Computing Surveys*, 58(3): 1–39, 2025.

Darren Chai Xin Lun and Lim Tong Ming. H-lora: Rethinking rank selection for controllable knowledge retention in edge ai. *Computers, Materials & Continua*, 88(1), 2026.

Minrui Luo, Fuhang Kuang, Yu Wang, Zirui Liu, and Tianxing He. Sc-lora: Balancing efficient fine-tuning and knowledge preservation via subspace-constrained lora. *arXiv preprint arXiv:2505.23724*, 2025.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

Yehonathan Refael, Jonathan Svirsky, Boris Shustin, Wasim Huleihel, and Ofir Lindenbaum. Adarankgrad: Adaptive gradient rank and moments for memory-efficient llms training and fine-tuning. In *International Conference on Learning Representations*, volume 2025, pp. 102591–102612, 2025.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *Advances in Neural Information Processing Systems*, 38: 174627–174662, 2026.

Luping Wang, Sheng Chen, Linnan Jiang, Shu Pan, Runze Cai, Sen Yang, and Fei Yang. Parameter-efficient fine-tuning in large language models: a survey of methodologies. *Artificial Intelligence Review*, 58(8):227, 2025a.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized? In *International Conference on Learning Representations*, volume 2025, pp. 93787–93808, 2025b.

Xiwen Wei, Mark Nutter, Madhusudhanan Srinivasan, and Radu Marculescu. Pareto lora: Mitigating modality imbalance in unified multimodal models via pareto-optimal gradient integration. *arXiv preprint arXiv:2606.17296*, 2026.

Yize Wu, Ke Gao, Ling Li, and Yanjun Wu. Stable-lora: Stabilizing feature learning of low-rank adaptation. *arXiv preprint arXiv:2603.05204*, 2026.

Lingling Xu, Haoran Xie, S Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Nuocheng Yang, Sihua Wang, Ouwen Huan, Mingzhe Chen, Tony QS Quek, and Changchuan Yin. Wireless federated multi-task llm fine-tuning via sparse-and-orthogonal lora. *arXiv preprint arXiv:2602.20492*, 2026.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

Xiang Zhang, Rui Xie, and Shikun Zhang. L2-lora: Improving low-rank adaptation with layer-specific regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34817–34826, 2026.

Ziyu Zhao, Yixiao Zhou, Xin Yu, Zhi Zhang, Didi Zhu, Tao Shen, Zexi Li, Jinluan Yang, Xuwu Wang, Jing Su, et al. Each rank could be an expert: Single-ranked mixture of experts lora for multi-task learning. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1998–2009, 2026.

Hongyun Zhou, Xiangyu Lu, Wang Xu, Conghui Zhu, Tiejun Zhao, and Muyun Yang. Lora-drop: Efficient lora parameter pruning based on output evaluation. In *Proceedings of the 31st international conference on computational linguistics*, pp. 5530–5543, 2025.

Yue Zhu, Haiwen Diao, Shang Gao, Jiazuo Yu, Jiawen Zhu, Yunzhi Zhuge, Shuai Hao, Xu Jia, Lu Zhang, Ying Zhang, et al. Regularizing subspace redundancy of low-rank adaptation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 1666–1675, 2025.

Heming Zou, Yunliang Zang, Wutong Xu, Yao Zhu, and Xiangyang Ji. Flylora: Boosting task decoupling and parameter efficiency via implicit rank-wise mixture-of-experts. *Advances in Neural Information Processing Systems*, 38:10386–10419, 2026.

Jianhui Zuo, Xuemeng Song, Haokun Wen, Meng Liu, Yupeng Hu, Jiuru Wang, and Liqiang Nie. D2mora: Diversity-regulated asymmetric moe-lora decomposition for efficient multi-task adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29286–29294, 2026.

A Related Work

A.1 Low-Rank Adaptation and Its Variants

LoRA (Hu et al., 2021) represents task-specific weight updates with two low-rank matrices, greatly reducing the number of trainable parameters while preserving inference-time mergeability. A series of subsequent studies improve LoRA from different perspectives. AdaLoRA (Zhang et al., 2023) dynamically reallocates rank budgets according to parameter importance, allowing limited adaptation capacity to focus on more influential weight matrices. LoRA-Pro (Wang et al., 2025b) revisits LoRA from an optimization perspective and adjusts the gradients of the two low-rank factors so that the induced low-rank update more closely approximates the gradient of full fine-tuning.

Recent work also explores improving the representational capacity and structural efficiency of LoRA. AuroRA (Dong et al., 2026) introduces an adaptive nonlinear layer between the two low-rank projections, aiming to alleviate the intrinsic low-rank bottleneck of purely linear LoRA mappings. LoRA-drop (Zhou et al., 2025) evaluates the importance of LoRA modules from their output responses, retains dedicated adapters only for influential layers, and shares parameters across less important ones, thereby reducing training and storage overhead. L2-LoRA (Zhang et al., 2026) introduces layer-specific L2 regularization, applying stronger constraints to lower layers to better preserve pretrained knowledge while allowing greater adaptation in higher layers. These methods improve LoRA through rank allocation, gradient optimization, nonlinear transformation, parameter pruning, and layer-wise regularization. However, they mainly focus on how adaptation capacity is allocated, optimized, or parameterized at the layer or matrix level, while the dependencies among individual rank-one components within a single LoRA update remain largely implicit. Our proposed method, PSR-LoRA, instead focuses on the internal organization of the rank dimension by explicitly modeling inter-component geometric relationships.

A.2 Redundancy and Structural Diversity in Low-Rank Adaptation

Beyond improving optimization and parameter allocation, recent studies have increasingly recognized redundancy and representation overlap as important limitations of low-rank adaptation (Zhu et al., 2025; Zou et al., 2026; Chang et al., 2026a). Orthogonality-based approaches (Yang et al., 2026; Cao & Wu, 2025) encourage different adaptation directions to occupy less-overlapping subspaces, while spectral and component-wise methods attempt to remove redundant directions or reparameterize LoRA in more compact subspaces. Related studies also introduce diversity constraints across multiple LoRA modules or experts to avoid homogeneous adaptation patterns (Zuo et al., 2026; Zhao et al., 2026).

Despite these efforts, most existing methods model diversity through linear orthogonality (Chang et al., 2026b; **?**), pruning Zhou et al. (2025), or subspace decomposition (Li et al., 2026b; Luo et al., 2025). Such strategies do not explicitly characterize richer nonlinear dependencies among the individual rank-one components inside a single LoRA update. PSR-LoRA addresses this limitation by mapping rank-one components into a signed power space and modeling their pairwise similarity directly. The resulting relational structure is then used for both redundancy regularization and adaptive component weighting, allowing rank diversity to be controlled in a continuous nonlinear space rather than through strict orthogonality alone.

B Algorithm

Algorithm 1 summarizes the training procedure of PSR-LoRA. At each iteration, the LoRA update is decomposed into rank-wise components and mapped into the signed power space. Their pairwise similarities are then computed efficiently from the low-rank factors to estimate component redundancy. The resulting similarity structure is used for redundancy regularization and, when enabled, similarity-guided component weighting. Finally, the task loss and PSR regularization loss are jointly optimized to update only the LoRA parameters while keeping the pretrained backbone frozen.

Algorithm 1 PSR-LoRA

Require: Frozen weight W_0, LoRA factors A, B, rank r, power exponent p, λ_{PSR}, temperature τ

for each mini-batch (X, Y) **do**
  Extract rank-wise factors $\{a_k, b_k\}_{k=1}^{r}$ from A and B
  $\tilde{a}_k \leftarrow \mathrm{sgn}(a_k)|a_k|^p$
  $\tilde{b}_k \leftarrow \mathrm{sgn}(b_k)|b_k|^p$
  Normalize $\tilde{a}_k$ and $\tilde{b}_k$ to obtain $\hat{A}$ and $\hat{B}$
  $S_A \leftarrow \hat{A}\hat{A}^\top$
  $S_B \leftarrow \hat{B}^\top\hat{B}$
  $S \leftarrow S_A \odot S_B$
  $\mathcal{L}_{\mathrm{PSR}} \leftarrow \frac{1}{r(r-1)} \sum_{i \neq j} S_{ij}^2$
  for $k = 1, \ldots, r$ **do**
    $\rho_k \leftarrow \frac{1}{r-1} \sum_{j \neq k} |S_{kj}|$
    $g_k \leftarrow \mathrm{Softmax}(-\rho_k/\tau)$
  end for
  $\Delta W \leftarrow \alpha \sum_{k=1}^{r} g_k b_k a_k^\top$
  $\hat{Y} \leftarrow f(X; W_0 + \Delta W)$
  $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{task}}(\hat{Y}, Y) + \lambda_{\mathrm{PSR}}\mathcal{L}_{\mathrm{PSR}}$
  Update A and B using $\nabla\mathcal{L}$
end for
return A, B

C Datasets Details

Table 5 summarizes the three datasets used in our experiments. FaultQA is used for structured fault-response generation from automotive and industrial descriptions. Patent is formulated as a QA-style classification task over IPC sections. SMRT maintenance descriptions are mapped to fine-grained component categories. Together, these datasets cover automotive fault understanding, patent classification, and railway maintenance classification, providing heterogeneous downstream settings for evaluating the generalization of PSR-LoRA across both generative and discriminative tasks.

Dataset	Domain	Task	Output
FaultQA	Automotive/Industrial	Structured generation	Fault relation triples
Patent	Patent	Classification	8 IPC sections
SMRT	Railway maintenance	Classification	766 Component categories

Table 5: Summary of datasets

D Theoretical Analysis of PSR-LoRA

We provide theoretical justification for the main components of PSR-LoRA. Specifically, we show that the proposed power-space similarity admits an exact factorized computation, that the PSR regularizer explicitly suppresses inter-component redundancy, and that the similarity-guided component weights can be derived as the solution to an entropy-regularized redundancy minimization problem.

Notation. Let

$$A \in \mathbb{R}^{r \times d_{\mathrm{in}}}, \qquad B \in \mathbb{R}^{d_{\mathrm{out}} \times r},$$

and denote the k-th row of A by $a_k^\top$ and the k-th column of B by b_k. The k-th rank-one LoRA component is

$$M_k = b_k a_k^\top.$$

We define the signed power mapping as

$$\phi_p(x) = \mathrm{sign}(x)|x|^p, \qquad p > 0,$$

which is applied element-wise to vectors and matrices.

Theorem D.1 (Exact Factorization of Power-Space Similarity)**.** *For any two rank-one components $M_i = b_i a_i^\top$ and $M_j = b_j a_j^\top$, the cosine similarity between their power-transformed representations satisfies*

$$\cos\left(\mathrm{vec}(\phi_p(M_i)), \mathrm{vec}(\phi_p(M_j))\right) = \cos\left(\phi_p(b_i), \phi_p(b_j)\right) \cos\left(\phi_p(a_i), \phi_p(a_j)\right).$$

Proof. The signed power transformation is multiplicatively separable:

$$\phi_p(xy) = \mathrm{sign}(xy)|xy|^p = \phi_p(x)\phi_p(y).$$

Therefore,

$$\phi_p(M_k) = \phi_p(b_k a_k^\top) = \phi_p(b_k)\phi_p(a_k)^\top.$$

Let

$$\tilde{b}_k = \phi_p(b_k), \qquad \tilde{a}_k = \phi_p(a_k).$$

Then

$$\phi_p(M_k) = \tilde{b}_k \tilde{a}_k^\top.$$

Using the rank-one Frobenius inner-product identity,

$$\left\langle uv^\top, xy^\top \right\rangle_F = (u^\top x)(v^\top y),$$

we obtain

$$\left\langle \phi_p(M_i), \phi_p(M_j) \right\rangle_F = (\tilde{b}_i^\top \tilde{b}_j)(\tilde{a}_i^\top \tilde{a}_j).$$

Moreover,

$$\|\phi_p(M_k)\|_F = \|\tilde{b}_k\|_2 \|\tilde{a}_k\|_2.$$

Substituting these expressions into the definition of cosine similarity yields

$$\cos\left(\mathrm{vec}(\phi_p(M_i)), \mathrm{vec}(\phi_p(M_j))\right) = \frac{\tilde{b}_i^\top \tilde{b}_j}{\|\tilde{b}_i\|_2 \|\tilde{b}_j\|_2} \cdot \frac{\tilde{a}_i^\top \tilde{a}_j}{\|\tilde{a}_i\|_2 \|\tilde{a}_j\|_2}.$$

Hence,

$$\cos\left(\mathrm{vec}(\phi_p(M_i)), \mathrm{vec}(\phi_p(M_j))\right) = \cos(\tilde{b}_i, \tilde{b}_j)\cos(\tilde{a}_i, \tilde{a}_j).$$

□

Corollary D.1 (Matrix Form of Rank-Wise Similarity)**.** *Let $\hat{A} \in \mathbb{R}^{r \times d_{\mathrm{in}}}$ contain the ℓ_2-normalized transformed row vectors $\phi_p(a_k)$, and let $\hat{B} \in \mathbb{R}^{d_{\mathrm{out}} \times r}$ contain the ℓ_2-normalized transformed column vectors $\phi_p(b_k)$. Then the complete rank-wise similarity matrix is*

$$S = S_A \odot S_B,$$

where

$$S_A = \hat{A}\hat{A}^\top, \qquad S_B = \hat{B}^\top \hat{B},$$

and $\odot$ denotes the Hadamard product.

Proof. By Theorem D.1, each pairwise similarity satisfies

$$S_{ij} = (S_A)_{ij}(S_B)_{ij}.$$

Applying this relation to all pairs (i, j) yields

$$S = S_A \odot S_B.$$

□

Corollary D.2 (Computational Advantage)**.** *The factorized formulation avoids explicitly constructing the rank-one matrices $M_k \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$. Computing all pairwise similarities requires*

$$O\left(r^2(d_{\mathrm{in}} + d_{\mathrm{out}})\right)$$

operations, rather than operating in the full $d_{\mathrm{out}} d_{\mathrm{in}}$-dimensional parameter space.

Theorem D.2 (Redundancy Reduction by PSR Regularization)**.** *Let z_k denote the normalized power-space representation of the k-th rank-one component, and define*

$$S_{ij} = z_i^\top z_j.$$

The PSR regularization term

$$\mathcal{L}_{\mathrm{PSR}} = \frac{1}{r(r-1)} \sum_{i \neq j} S_{ij}^2$$

is equivalent to minimizing the off-diagonal energy of the rank-wise Gram matrix. Its minimum is attained when the transformed rank components are pairwise orthogonal, whenever such a configuration is feasible.

Proof. Since each normalized representation satisfies

$$\|z_k\|_2 = 1,$$

the diagonal entries of the Gram matrix S satisfy

$$S_{kk} = 1.$$

Therefore,

$$\|S - I\|_F^2 = \sum_{i \neq j} S_{ij}^2,$$

and hence

$$\mathcal{L}_{\mathrm{PSR}} = \frac{1}{r(r-1)} \|S - I\|_F^2.$$

Since every term S_{ij}^2 is non-negative,

$$\mathcal{L}_{\mathrm{PSR}} \geq 0.$$

The minimum value is obtained when

$$S_{ij} = 0, \qquad i \neq j,$$

which implies

$$z_i^\top z_j = 0, \qquad i \neq j.$$

Thus, minimizing $\mathcal{L}_{\mathrm{PSR}}$ explicitly suppresses off-diagonal similarity and encourages complementary rank-wise representations in the power adaptation space. □

Corollary D.3 (Improved Utilization of the Nominal Rank Budget)**.** *If two transformed rank components are identical, i.e.,*

$$z_i = z_j,$$

then

$$S_{ij} = 1,$$

and the two nominal rank dimensions encode the same direction. More generally, driving the Gram matrix S toward the identity matrix reduces linear dependence among transformed rank components and encourages the available rank dimensions to represent more distinct adaptation directions.

Proof. If $z_i = z_j$ and $\|z_i\|_2 = \|z_j\|_2 = 1$, then

$$S_{ij} = z_i^\top z_j = 1.$$

Thus the two components are maximally aligned and contribute redundant directions. By contrast, when

$$S \approx I,$$

off-diagonal similarities are small and the transformed components are closer to pairwise orthogonality. Therefore, the nominal rank budget is used to encode more distinct adaptation directions. □

Theorem D.3 (Optimality of Similarity-Guided Component Weights)**.** *Let the redundancy score of the k-th component be*

$$\rho_k = \frac{1}{r-1} \sum_{j \neq k} |S_{kj}|.$$

Consider the optimization problem

$$\min_{\mathbf{g} \in \Delta_r} \left[\sum_{k=1}^{r} g_k \rho_k + \tau \sum_{k=1}^{r} g_k \log g_k \right],$$

where

$$\Delta_r = \left\{ \mathbf{g} \in \mathbb{R}^r : g_k \geq 0, \ \sum_{k=1}^{r} g_k = 1 \right\},$$

and $\tau > 0$ is a temperature parameter. Its unique solution is

$$g_k = \frac{\exp(-\rho_k/\tau)}{\sum_{j=1}^{r} \exp(-\rho_j/\tau)}.$$

Proof. Introduce a Lagrange multiplier μ for the simplex constraint:

$$\mathcal{J} = \sum_{k=1}^{r} g_k \rho_k + \tau \sum_{k=1}^{r} g_k \log g_k + \mu \left(\sum_{k=1}^{r} g_k - 1 \right).$$

Taking the derivative with respect to g_k gives

$$\frac{\partial \mathcal{J}}{\partial g_k} = \rho_k + \tau(\log g_k + 1) + \mu.$$

Setting the derivative to zero yields

$$\log g_k = -\frac{\rho_k}{\tau} - \frac{\mu + \tau}{\tau}.$$

Therefore,

$$g_k \propto \exp(-\rho_k/\tau).$$

Applying the normalization constraint

$$\sum_{k=1}^{r} g_k = 1$$

gives

$$g_k = \frac{\exp(-\rho_k/\tau)}{\sum_{j=1}^{r} \exp(-\rho_j/\tau)}.$$

Because the negative entropy term is strictly convex over the simplex for $\tau > 0$, the solution is unique. □

Corollary D.4 (Redundancy-Aware Adaptive Update)**.** *The similarity-guided update*

$$\Delta W = \alpha \sum_{k=1}^{r} g_k b_k a_k^\top$$

assigns larger contributions to components with smaller redundancy scores while maintaining a smooth distribution over rank components. The temperature τ controls the trade-off between selective weighting and uniform rank utilization.

Proof. From Theorem D.3,

$$g_k \propto \exp(-\rho_k/\tau).$$

Hence, if

$$\rho_i < \rho_j,$$

then

$$g_i > g_j.$$

Thus less redundant components receive larger weights. Moreover, as $\tau \to \infty$, the weights approach a uniform distribution, whereas smaller τ produces a sharper preference toward low-redundancy components. □

Proposition D.1 (Controllable Geometry of the Power Transformation)**.** *For any two nonzero coefficients x_1 and x_2, the signed power transformation satisfies*

$$\frac{|\phi_p(x_1)|}{|\phi_p(x_2)|} = \left(\frac{|x_1|}{|x_2|}\right)^p.$$

Therefore, the exponent p provides explicit control over the relative importance of strong and weak adaptation coefficients.

Proof. By definition,

$$|\phi_p(x)| = |x|^p.$$

Therefore,

$$\frac{|\phi_p(x_1)|}{|\phi_p(x_2)|} = \frac{|x_1|^p}{|x_2|^p} = \left(\frac{|x_1|}{|x_2|}\right)^p.$$

For $p > 1$, magnitude differences are amplified; for $0 < p < 1$, they are compressed; and for $p = 1$, the transformation reduces to the original linear parameter space. □

The above results clarify the role of each PSR-LoRA component. Theorem D.1 shows that the proposed factorized similarity is exactly equivalent to similarity computed over full power-transformed rank-one updates. Theorem D.2 establishes that PSR regularization directly minimizes inter-component dependence in the transformed space. Finally, Theorem D.3 provides an optimization interpretation for the adaptive component weights as the solution to an entropy-regularized redundancy minimization problem. These results justify the design of PSR-LoRA without implying that downstream performance improvements are theoretically guaranteed.

E Similarity Functions

To examine whether PSR-LoRA is robust to different definitions of inter-component relationships, we consider four representative similarity measures: cosine similarity, Pearson correlation, Hamming similarity, and the radial basis function (RBF) kernel. Given two power-transformed rank representations z_i and z_j, these measures characterize their relationships from complementary perspectives.

Cosine Similarity. Cosine similarity measures the directional alignment between two rank components while being insensitive to their overall scale:

$$S_{ij}^{\mathrm{Cos}} = \frac{z_i^\top z_j}{\|z_i\|_2 \|z_j\|_2 + \epsilon}.$$

A large absolute value indicates strongly aligned or anti-aligned adaptation patterns. We use cosine similarity as the default operator in PSR-LoRA because of its simplicity and efficient factorized computation.

Pearson Correlation. Pearson correlation measures linear dependence after removing the mean of each representation:

$$S_{ij}^{\mathrm{Pearson}} = \frac{(z_i - \bar{z}_i)^\top (z_j - \bar{z}_j)}{\|z_i - \bar{z}_i\|_2 \|z_j - \bar{z}_j\|_2 + \epsilon}.$$

Compared with cosine similarity, Pearson correlation is less affected by component-wise offsets and focuses on whether two rank components exhibit similar variation patterns.

Hamming Similarity. Since Hamming distance is defined for discrete representations, we first convert each transformed component into a binary sign pattern:

$$q(z)_t = \mathbb{I}[z_t \geq 0].$$

Measure	Relationship	Range	Characteristic
Cosine	Directional alignment	$[-1, 1]$	Scale-invariant
Pearson	Centered dependence	$[-1, 1]$	Mean-centered
Hamming	Sign-pattern agreement	$[0, 1]$	Discrete structure
RBF	Distance-based similarity	$(0, 1]$	Nonlinear locality

Table 6: Comparison of similarity functions used in PSR-LoRA.

The normalized Hamming distance is defined as

$$D_{ij}^{\mathrm{Ham}} = \frac{1}{d} \sum_{t=1}^{d} \mathbb{I}\left[q(z_i)_t \neq q(z_j)_t\right].$$

We then convert it into a similarity score:

$$S_{ij}^{\mathrm{Ham}} = 1 - D_{ij}^{\mathrm{Ham}}.$$

This measure ignores coefficient magnitudes and instead evaluates whether two components share similar sign patterns, providing a discrete view of rank-wise structural redundancy.

RBF Kernel. The radial basis function kernel measures similarity according to the Euclidean distance between two transformed components:

$$S_{ij}^{\mathrm{RBF}} = \exp\left(-\gamma \|z_i - z_j\|_2^2\right),$$

where $\gamma > 0$ controls the sensitivity to distance. Unlike cosine and Pearson similarity, the RBF kernel captures nonlinear local relationships, assigning high similarity to nearby components and rapidly decreasing similarity as their distance increases.

Table 6 summarizes the characteristics of the four similarity measures.

Unless otherwise specified, we use cosine similarity as the default operator in PSR-LoRA. Pearson correlation, Hamming similarity, and the RBF kernel are used as alternative relation functions to evaluate whether the effectiveness of PSR-LoRA depends on a particular similarity definition.

F Module-wise adaptation analysis

Figure 4 shows the Module-wise adaptation analysis of PSR-LoRA. It reveals clear differences in where PSR-LoRA is most effective. For the two classification tasks, q_proj achieves the highest aggregate score, indicating that adapting the query transformation is particularly important for learning discriminative task-specific representations. In contrast, up_proj performs best on FaultQA and also attains the highest overall composite score, suggesting that the feed-forward expansion layer contributes more strongly to structured response generation. The remaining modules show smaller but still meaningful contributions, while k_proj is consistently weaker across tasks. These results indicate that the effectiveness of PSR-LoRA is module-dependent rather than uniform across the Transformer, with attention projections playing a larger role in classification and feed-forward projections becoming more important for generation. This also supports the use of selective module adaptation instead of applying the same adaptation strategy indiscriminately to all projection layers.

Table 7 reveals clear task-dependent differences in the importance of LoRA adaptation modules. On Patent, adapting q_proj consistently yields the best performance across all four classification metrics, indicating that query projection is particularly important for capturing task-specific discriminative information. A similar pattern appears on SMRT, where q_proj achieves the highest Precision, Recall, and F1, while $gate_proj$ gives the best Accuracy. In contrast, FaultQA shows a different preference: up_proj obtains the best NES, ROUGE-L, and METEOR scores, whereas $down_proj$ achieves the highest BLEU-4. These results suggest that attention-related projections are more influential for the two classification tasks, while feed-forward projections contribute more strongly to the generation task. Overall, the results highlight substantial module-wise heterogeneity

Task	Metric	down_proj	gate_proj	k_proj	o_proj	q_proj	up_proj	v_proj
Patent	Acc	0.6667	0.6833	0.7125	0.6917	**0.7208**	0.6875	0.6792
	Pre	0.6749	0.6929	0.7126	0.7053	**0.7316**	0.6927	0.6933
	Rec	0.6681	0.6878	0.7101	0.6820	**0.7256**	0.6890	0.6836
	F1	0.6651	0.6804	0.7080	0.6854	**0.7184**	0.6886	0.6789
SMRT	Acc	0.7303	**0.7627**	0.7006	0.7433	0.7451	0.7488	0.7294
	Pre	0.3310	0.3439	0.2883	0.3370	**0.3561**	0.3386	0.3341
	Rec	0.3372	0.3426	0.3095	0.3481	**0.3542**	0.3433	0.3442
	F1	0.3177	0.3290	0.2863	0.3298	**0.3431**	0.3256	0.3226
FaultQA	NES	0.5738	0.5579	0.4977	0.5680	0.5209	**0.5826**	0.5490
	RougeL	0.6505	0.6299	0.5807	0.6493	0.5975	**0.6531**	0.6240
	mentor	0.7396	0.7415	0.6384	0.7135	0.6830	**0.7432**	0.7268
	Bleu4	**0.5666**	0.5559	0.4399	0.5332	0.4785	0.5590	0.5274

Table 7: Details of Module-wise adaptation analysis of PSR-LoRA.

and indicate that the most effective adaptation location depends on the downstream task rather than following a universally optimal projection module.

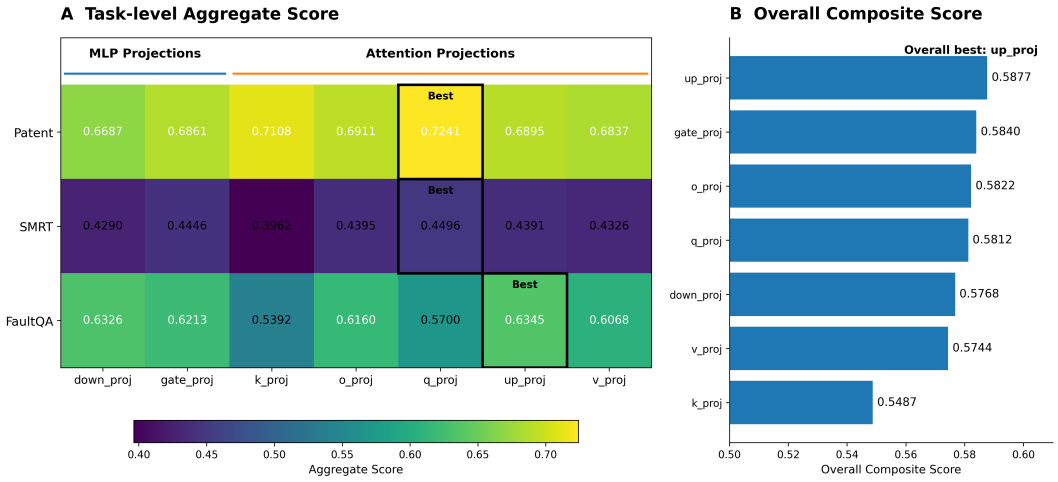


Figure 4: Module-wise adaptation analysis of PSR-LoRA. Task-level aggregate scores are obtained by averaging all evaluation metrics within each task, while the overall composite score averages the three task-level results. q_proj achieves the strongest performance on the two classification tasks, whereas up_proj performs best on FaultQA and attains the highest overall composite score.