DisLoRA: Task-specific Low-Rank Adaptation via Orthogonal Basis from Singular Value Decomposition

She Yifei[1]†, Xinhao Wei[2]†, and Yulong Wang[3]*
[1]School of Future, Beijing University of Posts and Telecommunications, Beijing, China
[2]International School, Beijing University of Posts and Telecommunications, Beijing, China
[3]State Key Laboratory of Networking and Switching Technology,
School of Computer Science (National Pilot Software Engineering School),
Beijing University of Posts and Telecommunications, Beijing, China
bupt3.1415926, weixinhao2022, wyl@bupt.edu.cn

Abstract

Parameter-efficient fine-tuning (PEFT) of large language models (LLMs) is critical for adapting to diverse downstream tasks with minimal computational cost. We propose **Di**rectional-**S**VD **Lo**w-**R**ank **A**daptation (DisLoRA), a novel PEFT framework that leverages Singular Value Decomposition (SVD) to decompose pretrained weight matrices into orthogonal backbone and task-specific subspaces, enabling precise capture of task-specific directions (TSDs). By dynamically identifying TSDs and employing adaptive soft orthogonal regularization with mean-normalization mechanism, DisLoRA balances task-specific and orthogonal losses without manual tuning, ensuring robust training stability. Extensive experiments on GLUE and Commonsense Reasoning benchmarks demonstrate that DisLoRA surpasses established PEFT methods, including LoRA, PiSSA, DoRA, LoRA-Dash, SORSA and MiLoRA. DisLoRA achieves superior performance on multiple individual GLUE datasets, surpassing baselines by up to 10.28% on SST-2 and 3.28% on CoLA, and consistently attains higher average accuracy than baselines across Commonsense Reasoning Tasks, with a maximum gain of 3.1%. Additionally, We also evaluated DisLoRA on the AQUA-RAT mathematical dataset, where it achieved the highest accuracy across all tested ranks. These results demonstrate DisLoRA's performance in efficient and high-performing LLM adaptation for domain-specific tasks while preserving generalization.

1 Introduction

Large language models (LLMs) have transformed natural language processing (NLP) by achieving extraordinary performance across diverse tasks, including text generation, machine translation, and complex question answering (Zhao et al., 2025). State-of-the-art models such as DeepSeek-R1 (DeepSeek-AI et al., 2025), LLaMA-3 (Grattafiori et al., 2024), and Qwen-2.5 (Qwen et al., 2025) demonstrate LLMs are competent for these tasks. LLMs have enabled practical advancements, such as conversational agents with near-human fluency (Jin et al., 2024) and automated systems for content summarization (Cajueiro et al., 2023) and code generation (Lu et al., 2025). Despite these successes, the escalating scale of modern LLMs, often encompassing hundreds of billions of parameters, introduces significant training challenges (Matarazzo and Torlone, 2025). Furthermore, guided by the scaling law (Kaplan et al., 2020), the trend of increasing model size has intensified. The required computational resources, including high-performance GPUs and extensive memory, incur substantial financial and energy costs.

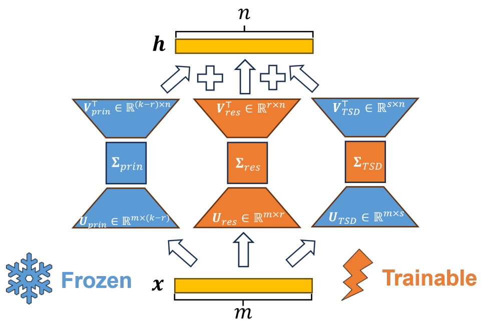

n
$\boldsymbol{h}$
$\boldsymbol{V}_{prin}^{\top} \in \mathbb{R}^{(k-r)\times n}$
$\boldsymbol{V}_{res}^{\top} \in \mathbb{R}^{r\times n}$
$\boldsymbol{V}_{TSD}^{\top} \in \mathbb{R}^{s\times n}$
$\boldsymbol{\Sigma}_{prin}$
$\boldsymbol{\Sigma}_{res}$
$\boldsymbol{\Sigma}_{TSD}$
$\boldsymbol{U}_{prin} \in \mathbb{R}^{m\times(k-r)}$
$\boldsymbol{U}_{res} \in \mathbb{R}^{m\times r}$
$\boldsymbol{U}_{TSD} \in \mathbb{R}^{m\times s}$
Frozen
$\boldsymbol{x}$
m
Trainable

Figure 1: DisLoRA decomposes $\boldsymbol{W}$ into backbone ($\boldsymbol{W}_{\text{prin}}$) and task-specific ($\boldsymbol{W}_{\text{res}}$) subspaces via SVD, further identifying task-specific directions ($\boldsymbol{W}_{\text{TSD}}$) for fine-tuning.

To mitigate the computational burden of full fine-tuning, parameter-efficient fine-tuning (PEFT) methods have emerged as a promising solution (Houlsby et al., 2019), adapting LLMs to downstream tasks by updating only a small subset of parameters while rivaling full fine-tuning performance (Han et al., 2024). Among these, Low-Rank

*Corresponding author: wyl@bupt.edu.cn. †These authors contributed equally to this work.

Adaptation (LoRA) stands out for its efficiency in updating model weights through low-rank matrices, preserving pretrained weights (Hu et al., 2022). Despite its efficiency, LoRA often suffers from training instability and suboptimal generalization due to the absence of regularization constraints (Bansal et al., 2018).

Efficient task adaptation relies on task-specific directions (TSDs), which are structured weight adjustment directions in the weight space critical for downstream tasks (Si et al., 2024). Conceptually akin to gradient vectors that guide optimization, TSDs differ fundamentally: while gradients are instantaneous and derived from loss function values, TSDs are identified through dynamic analysis of weight update patterns, enabling precise fitting of task-specific probability distributions, such as decision boundaries in classification or linguistic styles in generation. By focusing on TSDs, models achieve superior performance with minimal parameter updates (e.g., 0.48%–1.88% of parameters).

LoRA's limitations arise from two key shortcomings. First, its unconstrained low-rank updates do not explicitly identify TSDs, leading to inefficient adaptation across diverse tasks. Second, the absence of regularization, such as subspace orthogonality constraints, causes training instability, resulting in loss oscillations and reduced generalization (Zhan et al., 2024).

To overcome these challenges, we propose **Di**rectional-**S**VD **Lo**w-**R**ank **A**daptation (DisLoRA), a novel PEFT framework that leverages Singular Value Decomposition (SVD) to address LoRA's limitations. Inspired by the low intrinsic dimensionality of pretrained models (Aghajanyan et al., 2021), we decompose pretrained weight matrices $\boldsymbol{W}_0 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ into a frozen Backbone Subspace $\boldsymbol{W}_{\text{prin}} = \boldsymbol{U}_{\text{prin}}\boldsymbol{\Sigma}_{\text{prin}}\boldsymbol{V}_{\text{prin}}^\top$, preserving core capabilities, and a trainable Task-specific Subspace $\boldsymbol{W}_{\text{res}} = \boldsymbol{U}_{\text{res}}\boldsymbol{\Sigma}_{\text{res}}\boldsymbol{V}_{\text{res}}^\top$, capturing TSDs. Unlike prior SVD-based methods like PiSSA (Meng et al., 2024), we explicitly utilize orthogonal SVD forms to identify TSDs by adopting the change rate formula from recent work (Si et al., 2024). Building on this framework, DisLoRA introduces Adaptive Soft Orthogonal Regularization with Mean-Normalization to ensure subspace orthogonality, preventing coupling, and a softmax-like dynamic adaptive weighting mechanism to balance task-specific loss $\mathcal{L}_{\text{task}}$ and orthogonal loss $\mathcal{L}_{\text{ortho}}$, mitigating optimization instability. These mechanisms enable DisLoRA to outperform baselines, achieving a 10.28% accuracy gain over LoRA (Hu et al., 2022) on SST-2 (95.65% vs. 85.37%) and 5.61% on WNLI (78.87% vs. 73.24%) with DeBERTaV3-base, up to 2.5% higher average accuracy than LoRA-Dash (Si et al., 2024) (86.1% vs. 84.1%) across Commonsense Reasoning Tasks with Qwen2.5-7B-Instruct, and the highest accuracy on AQUA-RAT under every tested rank (details in Table 1, Table 2 and Table 3). The full code is avaiable at GitHub - DisLoRA.

The key contributions of this paper are summarized as follows:

1. We propose a new LLM fine-tuning method named DisLoRA. By decomposing pretrained weights into backbone and task-specific subspaces via SVD, our approach disentangles general capabilities from task-specific adaptations, providing a fine-grained strategy for efficient fine-tuning.

2. We pioneer Adaptive Soft Orthogonal Regularization with Mean-Normalization mechanism that leverages normalized averages and adaptive ratios to balance orthogonal and task-specific losses, eliminating manual tuning across heterogeneous model architectures for robust stability and generalization.

3. We demonstrate that DisLoRA outperforms LoRA (Hu et al., 2022), PiSSA (Meng et al., 2024), DoRA (Liu et al., 2024), LoRA-Dash (Si et al., 2024), SORSA (Cao, 2024) and MiLoRA(Wang et al., 2025) on GLUE benchmark and Commonsense Reasoning Tasks using DeBERTaV3-base and Qwen2.5-7B-Instruct respectively. On the GLUE benchmark, DisLoRA achieves the highest performance in 6/9 datasets. On Commonsense Reasoning Tasks, DisLoRA consistently attains the highest average accuracy across all evaluated ranks. Our experiments on the AQUA-RAT mathematical dataset demonstrate the versatility of DisLoRA.

2 Related Work

Parameter-efficient fine-tuning enables efficient adaptation of LLMs to downstream tasks with minimal parameters. To address limitations in Low-Rank Adaptation (LoRA), recent advancements leverage structured weight updates. This section

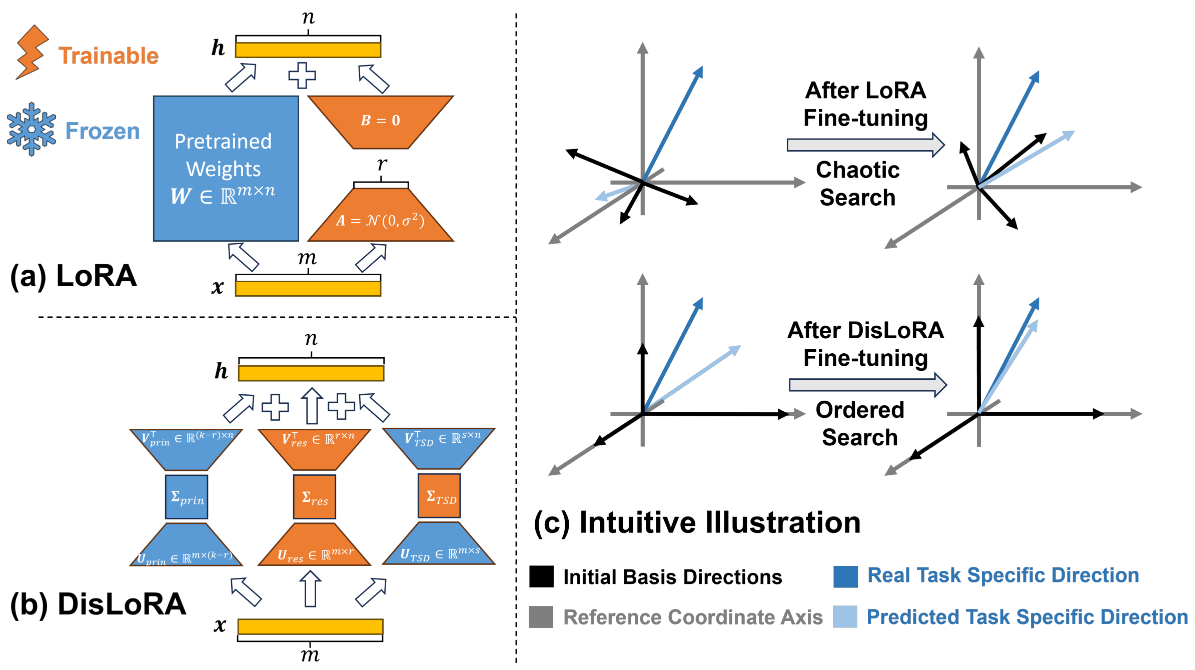

Trainable
Frozen
Pretrained Weights $W \in \mathbb{R}^{m\times n}$
h
n
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
m
x
(a) LoRA
$V_{prin}^{\top} \in \mathbb{R}^{(k-r)\times n}$
$V_{res}^{\top} \in \mathbb{R}^{r\times n}$
$V_{TSD}^{\top} \in \mathbb{R}^{s\times n}$
Σ_{prin}
Σ_{res}
Σ_{TSD}
$U_{prin} \in \mathbb{R}^{m\times(k-r)}$
$U_{res} \in \mathbb{R}^{m\times r}$
$U_{TSD} \in \mathbb{R}^{m\times s}$
(b) DisLoRA
After LoRA Fine-tuning
Chaotic Search
After DisLoRA Fine-tuning
Ordered Search
(c) Intuitive Illustration
Initial Basis Directions
Real Task Specific Direction
Reference Coordinate Axis
Predicted Task Specific Direction

Figure 2: (a) LoRA updates pretrained weights $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ by adding a low-rank adaptation $\boldsymbol{A} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{B} \in \mathbb{R}^{r\times n}$, where $\boldsymbol{A}$ is initialized with Gaussian distribution and $\boldsymbol{B}$ is initialized to zero. (b) DisLoRA decomposes $\boldsymbol{W}$ into backbone $\boldsymbol{W}_{\text{prin}}$ and task-specific $\boldsymbol{W}_{\text{res}}$ subspaces via SVD, further identifying TSDs $\boldsymbol{W}_{\text{TSD}}$ for fine-tuning. (c) Fine-tuning dynamics: LoRA performs chaotic search across coupled directions, while DisLoRA conducts ordered search by utilizing orthogonal basis to approach TSDs, shown in blue, from initial basis directions (black) within the reference coordinate axis (gray).

reviews two key developments: Singular Value Decomposition methods, which decompose weight matrices into orthogonal matrices and singular values, and Task-specific Directions, which identify critical weight adjustment directions for specific tasks.

2.1 Singular Value Decomposition

The integration of SVD into PEFT has undergone significant evolution, marked by three key developments: PiSSA, TriLoRA, SORSA and MiLoRA. PiSSA (Meng et al., 2024) introduced a novel approach by decomposing pre-trained weight matrices into principal and residual components using SVD. By initializing trainable adapters with the principal singular values and freezing the residual components, PiSSA achieved faster convergence and improved performance compared to traditional LoRA. However, its reliance on SVD for initialization introduced computational overhead, and its effectiveness depended on the assumption that residual components contained less critical information. Building on this, TriLoRA (Feng et al., 2024) integrated Compact SVD into the LoRA framework, enabling finer control over weight updates and enhancing adaptability in text-to-image generation tasks. While TriLoRA improved stability and reduced overfitting, its added complexity extended training time, and its performance remained tied to the quality of the pre-trained model. Finally, SORSA (Cao, 2024) advanced the field further by introducing orthonormal regularization to maintain the stability of singular vectors during training. This method not only preserved the benefits of SVD-based decomposition but also demonstrated faster convergence and superior performance in natural language tasks. Despite its strengths, SORSA's reliance on orthonormality constraints may limit its flexibility in varied scenarios. MiLoRA (Wang et al., 2025) also leverages SVD to decompose the pretrained weight, but freezes the principal components and updates only the minor singular values/vectors. While MiLoRA demonstrates consistent gains over vanilla LoRA, it lacks explicit identification of TSDs. Together, these methods illustrate the progressive refinement of SVD-based PEFT, with each method addressing the limitations of its predecessors while pushing the boundaries of efficiency and performance in fine-tuning LLMs.

2.2 Task-specific Direction

The concept of task-specific directions has emerged as a pivotal element in the development of PEFT methods, particularly in adapting large pre-trained models to downstream tasks. TSD refers to the specific directions in the pre-trained weight matrix that undergo significant changes when fine-tuning for a particular task, capturing the essential task-specific adaptations while minimizing the need for extensive parameter updates. Early work, such as DoRA (Liu et al., 2024), introduced the idea of decomposing weights into magnitude and directional components, implicitly addressing TSD by focusing on directional updates. However, it was LoRA-Dash (Si et al., 2024) that explicitly defined TSD as the core directions associated with smaller singular values of the pre-trained weights, which are crucial for task-specific performance. LoRA-Dash further proposed a two-phase framework to identify and optimize these directions, significantly enhancing fine-tuning efficiency. Despite these advancements, challenges remain in generalizing TSD across diverse tasks and models, as their identification and utilization often require careful hyperparameter tuning and may vary depending on the task complexity. The exploration of TSD continues to be a promising direction for improving the efficiency and effectiveness of PEFT methods.

3 Methodology

This section presents a detailed description of our proposed PEFT method, DisLoRA. Our approach leverages SVD to decompose pretrained weight matrices into orthogonal backbone and task-specific subspaces, dynamically identifies TSDs, and employs adaptive soft orthogonal regularization with mean-normalization mechanism to balance multiple loss functions, thereby significantly enhancing the performance of LLMs on downstream tasks.

3.1 Subspace Decomposition via SVD

We begin by performing SVD on the pretrained weight matrix $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$:

$$\boldsymbol{W}_0 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \tag{1}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ are the left and right singular vector matrices, respectively, and $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ is a diagonal matrix containing singular values $\sigma_1, \sigma_2, \dots, \sigma_{\min(m,n)}$ in descending order.

The weight matrix $\boldsymbol{W}_0$ is decomposed into two orthogonal subspaces:

- **Task-specific Subspace ($\boldsymbol{W}_{\textbf{res}}$)**: Comprises the smallest r singular values and their corresponding singular vectors, capturing directional information critical for specific tasks:

$$\boldsymbol{W}_{\text{res}} = \boldsymbol{U}_{\text{res}}\boldsymbol{\Sigma}_{\text{res}}\boldsymbol{V}_{\text{res}}^\top. \tag{2}$$

- **Backbone Subspace ($\boldsymbol{W}_{\textbf{prin}}$)**: Consists of the remaining larger singular values and vectors, preserving the pretrained LLM's general capabilities:

$$\boldsymbol{W}_{\text{prin}} = \boldsymbol{U}_{\text{prin}}\boldsymbol{\Sigma}_{\text{prin}}\boldsymbol{V}_{\text{prin}}^\top. \tag{3}$$

This decomposition is motivated by the hypothesis that singular vectors associated with smaller singular values are more likely to correspond to TSDs, as they represent underutilized directions in the pretrained model (Si et al., 2024). During fine-tuning, we freeze the backbone subspace $\boldsymbol{W}_{\text{prin}}$ and only train the task-specific subspace $\boldsymbol{W}_{\text{res}}$, updating its singular vectors and values as trainable parameters to capture TSDs while preserving core model capabilities.

3.2 Dynamic Identification of Task-specific Directions

Task-specific directions correspond to weight matrix directions critical for given downstream task (Lichtenstein et al., 2020), typically associated with smaller singular values and high change rates. Since the optimal weight matrix $\boldsymbol{W}^*$ is unknown during fine-tuning, we adopt a dynamic identification approach inspired by prior work (Si et al., 2024) to approximate TSDs based on weight updates.

We introduce a warmup phase at the start of training to update the task-specific subspace parameters and obtain the refined weights W_{res}. By default, this phase spans the first third of the total steps, allowing TSDs to be captured before overfitting injects noise—a design inspired by early stopping principles (Prechelt, 2012). Empirically, the task loss converges within this period, leaving the remaining two-thirds of training to fine-tune the weighting coefficients. The warmup duration is treated as a task-dependent hyperparameter, akin to the rank r and scaling factor α in LoRA: complex tasks (e.g., multi-step reasoning) benefit from longer warmups, whereas simpler tasks require shorter ones.

For each singular vector pair $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ from $\boldsymbol{W}_0$, we compute the change rate along the direction $\boldsymbol{u}_i\boldsymbol{v}_i^\top$:

$$\delta_i = \left| \frac{\boldsymbol{u}_i^\top \boldsymbol{W}_{\text{res}} \boldsymbol{v}_i}{\sigma_i} \right|, \tag{4}$$

where:

- $\boldsymbol{u}_i^\top \boldsymbol{W}_{\text{res}} \boldsymbol{v}_i$: The projection of the weight update onto the direction $\boldsymbol{u}_i\boldsymbol{v}_i^\top$, reflecting the magnitude of adjustment.
- σ_i: The i-th singular value of $\boldsymbol{W}_0$, indicating the direction's pretrained significance.

The change rate δ_i quantifies the direction's contribution to task adaptation, with higher values indicating TSDs that require significant adjustments. After the warmup phase, we select the top s_{tsd} directions with the highest δ_i as TSDs, consistent with the hypothesis that high-change directions reflect critical task adaptation needs. The hyperparameter s_{tsd} was set to 8 based on rigorous validation in (Si et al., 2024), ensuring optimal performance and consistent comparison.

3.3 Optimizing Weight Updates Using TSDs

Upon identifying TSDs, we optimize the weight matrix by adjusting their linear combination to enhance task-specific performance. We freeze the TSD directions and introduce learnable weighting coefficients α_i, updating the weight matrix as:

$$\boldsymbol{W} = \boldsymbol{W}_{\text{prin}} + (\boldsymbol{U}_{\text{res}} \boldsymbol{\Sigma}_{\text{res}} \boldsymbol{V}_{\text{res}}^\top) + \sum_{i=1}^{s_{\text{tsd}}} \alpha_i \boldsymbol{u}_i \boldsymbol{v}_i^\top. \tag{5}$$

During training, only the TSD directions $\boldsymbol{u}_i\boldsymbol{v}_i^\top$ are frozen, while the coefficients α_i and residual weights $\boldsymbol{U}_{\text{res}} \boldsymbol{\Sigma}_{\text{res}} \boldsymbol{V}_{\text{res}}^\top$ are optimized. Inputs are scaled by a factor α_{LoRA}/r, where r is the rank of the task-specific subspace and α_{LoRA} is set to $1.5r$ by default.

3.4 Adaptive Soft Orthogonal Regularization with Mean-Normalization

To ensure orthogonality of the task-specific subspace and enhance training stability, we employ soft orthogonal (SO) regularization (Bansal et al., 2018):

$$\mathcal{L}_{\text{ortho}} = \frac{1}{N_{\text{p}}} \sum_{i=1}^{N_{\text{p}}} \left(\|\boldsymbol{U}_i^\top \boldsymbol{U}_i - \boldsymbol{I}\|_F^2 + \|\boldsymbol{V}_i \boldsymbol{V}_i^\top - \boldsymbol{I}\|_F^2 \right), \tag{6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\boldsymbol{I}$ is the identity matrix, and N_{p} is the number of singular vector pairs. Normalizing by N_{p} mitigates the impact of varying layer counts across models, ensuring architecture-agnostic generalization.

However, the orthogonal loss $\mathcal{L}_{\text{ortho}}$, derived from matrix norms, inherently exhibits orders-of-magnitude disparities with the probabilistic task-specific loss $\mathcal{L}_{\text{task}}$ (e.g., cross-entropy) due to varying model dimensions, rendering manual weighting adjustments ($\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{L}_{\text{ortho}}$) challenging and poorly generalizable. To address this, we pioneer a softmax-like dynamic weighting mechanism that adaptively balances losses without manual tuning. We compute the relative loss ratio:

$$k = \frac{\mathcal{L}_{\text{ortho}}}{\mathcal{L}_{\text{task}}}, \tag{7}$$

and derive weights via a softmax-like mechanism:

$$\alpha_{\text{task}} = \frac{\exp(-k)}{\exp(-k) + \exp(-1/k)}, \tag{8}$$

$$\alpha_{\text{ortho}} = 1 - \alpha_{\text{task}}. \tag{9}$$

The total loss is:

$$\mathcal{L}_{\text{total}} = \alpha_{\text{task}} \cdot \mathcal{L}_{\text{task}} + \alpha_{\text{ortho}} \cdot \mathcal{L}_{\text{ortho}}. \tag{10}$$

This approach seamlessly adapts to heterogeneous model architectures and dynamically adjusts the contributions of the two types of loss function based on relative loss magnitudes to mitigate optimization biases, ensuring robust optimization and generalization. Detailed mathematical derivations and gradient analyses are provided in Appedix A.

4 Experiments

We evaluate the performance of DisLoRA, across natural language understanding (NLU) and Commonsense Reasoning Tasks, comparing it against established PEFT baselines. Our experiments aim to demonstrate the superior performance of DisLoRA in capturing TSDs and the effectiveness of its dynamic weighting algorithm. All experiments are conducted on NVIDIA V100 (32GB) GPUs and RTX4090 (24GB) GPU using the PyTorch framework , DeepSpeed inference (Aminabadi et al., 2022) and the AdamW optimizer with a linear learning rate schedule.

4.1 Baselines

For NLU tasks, we compare DiSLoRA with three PEFT methods: LoRA (Hu et al., 2022), PiSSA (Meng et al., 2024), and DoRA (Liu et al., 2024). To validate DisLoRA's performance for Commonsense Reasoning Tasks, ablation studies assess orthogonal TSD identification capacity and dynamic weighting mechanisms. Experiments compare DisLoRA against LoRA (Hu et al., 2022), LoRA-Dash (Si et al., 2024), SORSA (Cao, 2024), and DisLoRA* (a variant with fixed weighting). LoRA employs unconstrained low-rank updates without TSD focus, LoRA-Dash utilizes non-orthogonal basis to capture TSDs, and SORSA applies a fixed weighting coefficient for total loss $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda\mathcal{L}_{\text{ortho}}$. DisLoRA* uses the same total loss function with $\lambda = 5 \times 10^{-5}$ as SORSA, isolating the contribution of DisLoRA's dynamic weighting, which adjusts α_{task} and α_{ortho} based on the loss ratio $\frac{\mathcal{L}_{\text{ortho}}}{\mathcal{L}_{\text{task}}}$.

4.2 Natural Language Understanding

We fine-tune the $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v$ in DeBERTaV3-base (184M parameters) (He et al., 2023) on the GLUE benchmark (Wang et al., 2019), which comprises nine datasets (Detailed descriptions are in appedix B.1). All datasets are split into training, validation, and test sets aligned with the default in huggingface. These datasets evaluate a broad range of natural language understanding tasks, including sentiment analysis, semantic inference, and grammatical judgment.

4.2.1 Experimental Results and Discussion

DisLoRA achieves leading performance on the GLUE benchmark with DeBERTaV3-base, outperforming in SST-2 (95.65% Acc.), CoLA (68.33% Mcc.), and RTE (86.64% Acc.), with stable results across multiple tasks as shown in Table 1. Compared to full fine-tuning (FFT), DisLoRA delivers comparable or superior performance, surpassing FFT by 4.33% on RTE (86.64% vs. 82.31%) while using only 0.48%–1.88% of parameters. This efficiency stems from SVD separating pretrained weights into task-specific subspaces, with orthogonal basis capturing critical task features, enhanced by dynamic loss balancing.

DisLoRA outperforms LoRA (Hu et al., 2022) and its variants, PiSSA (Meng et al., 2024) and DoRA (Liu et al., 2024), due to its well-designed structure. LoRA's unconstrained updates yield unstable results, trailing by 3.28% on CoLA (65.05% vs. 68.33%) and 5.63% on WNLI (73.24% vs. 78.87%). PiSSA's static SVD initialization excels in QQP (91.58%) but lags by 18.61% on WNLI (60.26%). DoRA's direction-magnitude decoupling performs strongly in MRPC (93.69%) but falls short by 1.73% on CoLA (66.60%). DisLoRA's orthogonal basis capture task-specific directions, driving consistent task performance, with dynamic weighting ensuring generally excellent performance as shown in Table 1.

DisLoRA's advantages are not universal, with PiSSA (91.87% QQP) and DoRA (91.57% STS-B) occasionally outperforming it (Table 1). Increased parameters introduce redundancy, diluting task-specific direction capture in QQP's paraphrase similarity and STS-B's semantic correlation tasks. Despite this, DisLoRA sustains robust performance, achieving 68.33% Mcc. on CoLA and 86.64% Acc. on RTE, driven by dynamic loss balancing that prioritizes task optimization.

4.3 Commonsense Reasoning and Ablation Studies

We fine-tune the $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v, \boldsymbol{W}_o$ in Qwen2.5-7B-Instruct (Qwen et al., 2025) on the Commonsense Reasoning Tasks (Hu et al., 2023), which aggregates training sets from eight commonsense reasoning datasets (Detailed descriptions are in appedix B.1). These datasets evaluate diverse commonsense reasoning abilities, including factual, physical, social, and scientific inference. We fine-tune the model on Commonsense170k and evaluation is conducted on the individual test sets of these datasets.

4.3.1 Experimental Results and Ablation Studies

DisLoRA outperforms in Commonsense Reasoning Tasks, achieving top scores in BoolQ (73.4%), HellaSwag (93.1%) and ARC-C (87.5%) and consistently higher average accuracy than baselines according to Table 2. Baselines without regularization constraints, LoRA (Hu et al., 2022) and LoRA-Dash (Si et al., 2024), exhibit unstable performance, failing to achieve high accuracy across all tasks, thus trailing SORSA's average accuracy. SORSA (Cao, 2024), with regularization but no use of TSDs, sustains high average accuracy but struggles in individual tasks. DisLoRA surpasses MiLoRA (Wang et al., 2025) by +1.4 % average accuracy ($r = 16$), because MiLoRA only uses the minor singular components to initialize

Table 1: Performance of DeBERTaV3-base on the GLUE benchmark using FFT, LoRA (Hu et al., 2022), PiSSA (Meng et al., 2024), DoRA (Liu et al., 2024), and DisLoRA. We report accuracy (Acc.) for SST-2, QNLI, RTE, MNLI, QQP, and WNLI; F1 score (F1) for MRPC; Matthew's correlation coefficient (Mcc.) for CoLA; and Pearson correlation coefficient (Pcc.) for STS-B. Superscripts represent the standard deviation.

Method	Rank	Param. (%)	SST-2 (Acc.)	MRPC (F1.)	CoLA (Mcc.)	QNLI (Acc.)	RTE (Acc.)	STS-B (Pcc.)	MNLI (Acc.)	QQP (Acc.)	WNLI (Acc.)
FFT	-	100	94.8	90.2	69.19	92.8	82.31	91.2	89.90	91.87	63.38
LoRA	16	0.48	$85.37^{\pm0.34}$	$92.90^{\pm0.53}$	$65.05^{\pm2.54}$	$94.15^{\pm4.32}$	$84.47^{\pm1.12}$	$91.61^{\pm0.35}$	$89.90^{\pm0.48}$	$91.55^{\pm0.18}$	$73.24^{\pm5.01}$
PiSSA	16	0.48	$95.18^{\pm0.23}$	$92.67^{\pm0.6}$	$67.06^{\pm2.71}$	$93.36^{\pm1.02}$	$84.12^{\pm1.89}$	$90.27^{\pm1.85}$	$89.35^{\pm0.82}$	$\mathbf{91.58}^{\pm0.09}$	$60.26^{\pm3.32}$
DoRA	16	0.48	$95.41^{\pm0.38}$	$\mathbf{93.69}^{\pm0.44}$	$66.60^{\pm1.83}$	$94.02^{\pm0.45}$	$84.51^{\pm2.54}$	$\mathbf{91.57}^{\pm0.23}$	$90.06^{\pm0.35}$	$91.54^{\pm0.31}$	$70.42^{\pm4.15}$
DisLoRA (ours)	16	0.48	$\mathbf{95.65}^{\pm0.42}$	$93.12^{\pm0.57}$	$\mathbf{68.33}^{\pm0.52}$	$\mathbf{94.60}^{\pm0.25}$	$\mathbf{85.92}^{\pm1.53}$	$89.73^{\pm1.01}$	$\mathbf{90.12}^{\pm0.15}$	$91.17^{\pm0.23}$	$\mathbf{78.87}^{\pm5.81}$
LoRA	32	0.95	$95.13^{\pm0.17}$	$88.97^{\pm0.63}$	$65.78^{\pm1.28}$	$93.64^{\pm0.39}$	$84.47^{\pm1.27}$	$91.06^{\pm0.41}$	$89.38^{\pm0.47}$	$\mathbf{91.80}^{\pm0.13}$	$64.82^{\pm5.42}$
PiSSA	32	0.95	$94.61^{\pm0.23}$	$88.92^{\pm0.68}$	$64.45^{\pm1.57}$	$\mathbf{94.12}^{\pm0.14}$	$82.13^{\pm1.95}$	$90.75^{\pm0.53}$	$89.88^{\pm0.43}$	$91.63^{\pm0.17}$	$56.34^{\pm4.17}$
DoRA	32	0.95	$94.84^{\pm0.14}$	$89.95^{\pm0.47}$	$65.63^{\pm1.08}$	$94.11^{\pm0.19}$	$\mathbf{85.20}^{\pm0.97}$	$\mathbf{91.40}^{\pm0.23}$	$89.29^{\pm0.52}$	$91.74^{\pm0.14}$	$64.79^{\pm7.76}$
DisLoRA (ours)	32	0.95	$\mathbf{95.87}^{\pm0.16}$	$\mathbf{90.20}^{\pm0.49}$	$\mathbf{66.37}^{\pm1.18}$	$94.02^{\pm0.16}$	$83.75^{\pm1.23}$	$90.39^{\pm0.37}$	$\mathbf{90.11}^{\pm0.31}$	$91.56^{\pm0.22}$	$\mathbf{76.06}^{\pm6.24}$
LoRA	64	1.88	$95.36^{\pm0.27}$	$88.48^{\pm0.73}$	$64.57^{\pm1.52}$	$94.01^{\pm0.37}$	$84.84^{\pm1.43}$	$91.15^{\pm0.42}$	$90.08^{\pm0.47}$	$91.56^{\pm0.23}$	$73.24^{\pm6.38}$
PiSSA	64	1.88	$94.15^{\pm0.31}$	$89.95^{\pm0.57}$	$64.36^{\pm1.82}$	$\mathbf{94.27}^{\pm0.17}$	$81.23^{\pm2.53}$	$90.59^{\pm0.63}$	$\mathbf{90.34}^{\pm0.36}$	$\mathbf{91.87}^{\pm0.14}$	$56.34^{\pm3.15}$
DoRA	64	1.88	$95.18^{\pm0.22}$	$89.46^{\pm0.58}$	$64.90^{\pm1.32}$	$94.14^{\pm0.26}$	$85.92^{\pm1.12}$	$\mathbf{91.34}^{\pm0.31}$	$90.26^{\pm0.43}$	$91.77^{\pm0.19}$	$69.01^{\pm6.07}$
DisLoRA (ours)	64	1.88	$\mathbf{95.52}^{\pm0.29}$	$\mathbf{93.24}^{\pm0.51}$	$\mathbf{67.79}^{\pm1.43}$	$94.03^{\pm0.21}$	$\mathbf{86.64}^{\pm1.47}$	$90.52^{\pm0.46}$	$90.15^{\pm0.39}$	$91.56^{\pm0.24}$	$\mathbf{76.06}^{\pm5.84}$

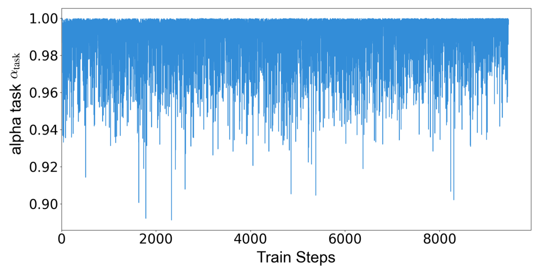
Table 2: The accuracy of Qwen2.5-7B-Instruct on Commonsense Reasoning Tasks using LoRA (Hu et al., 2022), LoRA-Dash (Si et al., 2024), SORSA (Cao, 2024), MiLoRA (Wang et al., 2025),DisLoRA, and DisLoRA* with fixed weighting. We report accuracy (Acc.) for all datasets. Superscripts represent the standard deviation.

Method	Rank	Param. (%)	BoolQ	PIQA	SIQA	HellaSwag	Winogrande	ARC-C	ARC-E	OBQA	Avg.
LoRA	8	0.07	$70.5^{\pm2.7}$	$84.9^{\pm1.3}$	$79.4^{\pm0.8}$	$73.7^{\pm3.1}$	$85.0^{\pm1.2}$	$86.0^{\pm0.9}$	$93.6^{\pm0.7}$	$88.4^{\pm1.6}$	$82.7^{\pm1.5}$
LoRA-Dash	8	0.07	$69.9^{\pm3.0}$	$83.6^{\pm1.7}$	$78.3^{\pm1.1}$	$83.1^{\pm1.9}$	$\mathbf{87.4}^{\pm1.5}$	$\mathbf{88.0}^{\pm1.2}$	$\mathbf{95.6}^{\pm0.6}$	$89.1^{\pm0.8}$	$84.4^{\pm1.9}$
SORSA	8	0.07	$68.6^{\pm2.2}$	$85.0^{\pm1.4}$	$78.5^{\pm0.9}$	$90.9^{\pm0.9}$	$84.6^{\pm1.3}$	$86.3^{\pm1.0}$	$94.4^{\pm0.6}$	$86.4^{\pm1.3}$	$84.3^{\pm1.6}$
MiLoRA	8	0.07	$69.3^{\pm2.1}$	$81.0^{\pm1.9}$	$\mathbf{79.6}^{\pm1.3}$	$87.1^{\pm1.8}$	$84.7^{\pm1.4}$	$87.8^{\pm1.2}$	$94.6^{\pm0.8}$	$\mathbf{90.2}^{\pm1.5}$	$84.3^{\pm2.7}$
DisLoRA (ours)	8	0.07	$72.3^{\pm0.4}$	$\mathbf{85.5}^{\pm0.6}$	$79.3^{\pm0.8}$	$\mathbf{92.3}^{\pm0.7}$	$83.7^{\pm0.6}$	$87.3^{\pm0.7}$	$95.1^{\pm0.7}$	$88.6^{\pm0.4}$	$\mathbf{85.5}^{\pm1.3}$
DisLoRA* (ours)	8	0.07	$\mathbf{72.4}^{\pm0.9}$	$84.9^{\pm0.8}$	$79.3^{\pm0.9}$	$91.7^{\pm0.5}$	$82.9^{\pm0.9}$	$87.0^{\pm0.9}$	$94.5^{\pm0.7}$	$87.4^{\pm0.4}$	$85.0^{\pm1.1}$
LoRA	16	0.13	$71.4^{\pm2.4}$	$77.1^{\pm1.9}$	$78.9^{\pm0.9}$	$88.3^{\pm2.7}$	$83.3^{\pm1.1}$	$85.2^{\pm0.8}$	$93.3^{\pm0.6}$	$86.6^{\pm1.7}$	$83.0^{\pm1.3}$
LoRA-Dash	16	0.13	$68.0^{\pm2.8}$	$80.5^{\pm2.4}$	$79.2^{\pm1.0}$	$91.0^{\pm1.0}$	$82.1^{\pm2.1}$	$85.4^{\pm1.4}$	$\mathbf{95.8}^{\pm0.5}$	$\mathbf{91.0}^{\pm0.7}$	$84.1^{\pm2.0}$
SORSA	16	0.13	$71.4^{\pm1.7}$	$85.1^{\pm1.8}$	$\mathbf{79.6}^{\pm0.7}$	$91.2^{\pm0.7}$	$85.4^{\pm0.9}$	$84.8^{\pm1.1}$	$94.7^{\pm0.5}$	$90.4^{\pm0.4}$	$85.3^{\pm1.5}$
MiLoRA	16	0.13	$69.2^{\pm2.0}$	$85.5^{\pm1.6}$	$74.9^{\pm1.8}$	$92.3^{\pm1.1}$	$\mathbf{85.5}^{\pm1.3}$	$86.7^{\pm1.3}$	$94.1^{\pm0.7}$	$89.2^{\pm1.4}$	$84.7^{\pm3.0}$
DisLoRA (ours)	16	0.13	$72.7^{\pm1.3}$	$\mathbf{85.9}^{\pm0.4}$	$79.2^{\pm0.7}$	$\mathbf{92.8}^{\pm0.8}$	$85.4^{\pm0.5}$	$\mathbf{87.0}^{\pm0.9}$	$95.0^{\pm0.3}$	$90.4^{\pm0.4}$	$\mathbf{86.1}^{\pm1.0}$
DisLoRA* (ours)	16	0.13	$\mathbf{72.8}^{\pm1.1}$	$85.0^{\pm1.0}$	$79.0^{\pm0.7}$	$\mathbf{92.8}^{\pm0.6}$	$83.4^{\pm0.4}$	$85.8^{\pm0.7}$	$94.4^{\pm0.4}$	$90.0^{\pm0.6}$	$85.4^{\pm1.3}$
LoRA	32	0.26	$71.2^{\pm2.3}$	$80.3^{\pm1.9}$	$78.6^{\pm0.8}$	$85.1^{\pm2.6}$	$84.5^{\pm0.9}$	$84.7^{\pm0.8}$	$93.8^{\pm0.5}$	$89.0^{\pm1.5}$	$83.5^{\pm1.2}$
LoRA-Dash	32	0.26	$64.3^{\pm2.7}$	$83.8^{\pm2.3}$	$79.7^{\pm0.9}$	$92.3^{\pm0.9}$	$85.5^{\pm2.1}$	$85.3^{\pm1.3}$	$\mathbf{95.5}^{\pm0.5}$	$90.8^{\pm0.6}$	$84.7^{\pm1.9}$
SORSA	32	0.26	$70.3^{\pm1.6}$	$85.1^{\pm1.7}$	$78.2^{\pm0.5}$	$92.7^{\pm0.6}$	$84.9^{\pm0.7}$	$85.9^{\pm0.9}$	$94.6^{\pm0.4}$	$\mathbf{90.9}^{\pm0.5}$	$85.3^{\pm1.4}$
MiLoRA	32	0.26	$72.0^{\pm1.9}$	$83.8^{\pm1.7}$	$78.9^{\pm1.2}$	$92.7^{\pm1.0}$	$\mathbf{86.1}^{\pm1.5}$	$85.1^{\pm1.4}$	$94.1^{\pm0.6}$	$90.8^{\pm1.1}$	$85.4^{\pm2.6}$
DisLoRA (ours)	32	0.26	$73.3^{\pm1.1}$	$\mathbf{85.7}^{\pm1.4}$	$\mathbf{80.0}^{\pm0.7}$	$\mathbf{93.1}^{\pm0.5}$	$85.0^{\pm0.4}$	$\mathbf{87.5}^{\pm1.3}$	$95.1^{\pm0.6}$	$89.8^{\pm0.5}$	$\mathbf{86.2}^{\pm1.4}$
DisLoRA* (ours)	32	0.26	$\mathbf{73.4}^{\pm1.3}$	$83.9^{\pm1.6}$	$79.2^{\pm0.6}$	$92.4^{\pm1.3}$	$83.9^{\pm0.9}$	$85.5^{\pm1.6}$	$93.9^{\pm1.0}$	$89.4^{\pm0.6}$	$85.2^{\pm1.4}$

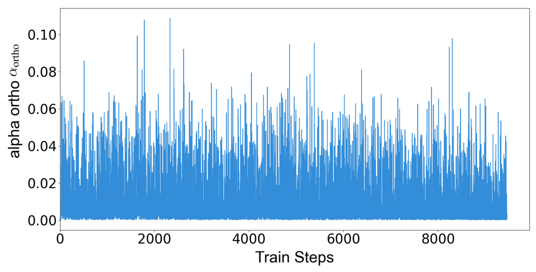
adapters—a static SVD initialization—whereas DisLoRA retains the explicit SVD matrix form and jointly optimizes TSD enhancement, yielding superior adaptation. DisLoRA*, employing TSDs and fixed-weight regularization, surpasses baselines in both individual tasks and average accuracy. DisLoRA, with dynamic weighting, further exceeds DisLoRA*'s performance, demonstrating the superiority of its adaptive approach.

The dynamic weighting mechanism in DisLoRA, as depicted in the fluctuations of α_{task} and α_{ortho} (Figure 3), ensures robust convergence by adaptively balancing task-specific optimization and orthogonal regularization. Throughout training, α_{task} remains above 0.9, while α_{ortho} stays below 0.1, prioritizing task-specific loss to effectively learn from training datasets, with orthogonality as a secondary constraint. This supports our adaptive weighting design, which outperforms fixed-weighting approaches (e.g., when rank is 32, the maximum average accuracy difference reaches to 1.0% on Commonsense Reasoning Tasks, Table 2). Frequent fluctuations in both weights highlight the necessity of adaptive weighting, enabling the model to dynamically adjust to varying loss magnitudes, unlike static methods, thus enhancing training stability and performance.

DisLoRA's training dynamics demonstrate robust optimization, achieving superior performance over baselines. Task-specific loss $\mathcal{L}_{\text{task}}$ decreases rapidly, then stabilizing with minimal fluctuations (Figure 4a), driven by task-specific direction (TSD) identification. Orthogonalization loss $\mathcal{L}_{\text{ortho}}$ declines steadily (Figure 4b), reflecting soft orthogonal regulation enforcing subspace independence, contributing to highest average accu-



(a) Dynamic weight of the task-specific component (α_{task}) over training steps, showing fluctuations between 0.9 and 1.



(b) Dynamic weight of the orthogonal component (α_{ortho}) over training steps, illustrating the contribution of orthogonalization during training.

Figure 3: Dynamic weight allocation in DisLoRA: (a) the task-specific weight (α_{task}), consistently above 0.9, exhibits frequent fluctuations, prioritizing task-specific optimization to achieve high performance (e.g., 93.1% on HellaSwag when rank is 32, Table 2); (b) the orthogonal weight (α_{ortho}), below 0.1, adjusts dynamically to enforce subspace orthogonality, enhancing generalization (e.g., 86.2% average accuracy on Commonsense Reasoning Tasks when rank is 32). These fluctuations underscore the adaptive weighting mechanism's ability to balance task-specific learning with structural regularization across training steps.

racy across Commonsense Reasoning Tasks. The loss ratio $\frac{\mathcal{L}_{\text{ortho}}}{\mathcal{L}_{\text{task}}}$ fluctuates (Figure 4c), showcasing DisLoRA's dynamic weighting mechanism is neccesary, which timely adjusts α_{task} and α_{ortho} to balance task optimization and orthogonality. Consequently, total loss $\mathcal{L}_{\text{total}} = \alpha_{\text{task}} \cdot \mathcal{L}_{\text{task}} + \alpha_{\text{ortho}} \cdot \mathcal{L}_{\text{ortho}}$ converges with minimal fluctuations (Figure 4d), ensuring training stability. Unlike LoRA's (Hu et al., 2022) non-orthogonal updates, LoRA-Dash's (Si et al., 2024) coupled TSD identification, SORSA's (Cao, 2024) fixed regularization, or MiLoRA's (Wang et al., 2025) static SVD initialization, DisLoRA's orthogonal TSD identification and adaptive loss balancing yield the highest accuracy, validating its efficacy for commonsense reasoning.

4.4 Mathematical Reasoning

To demonstrate the versatility of DisLoRA, we conduct supplementary experiments on mathematical reasoning. We fine-tune the Qwen2.5-Instruct on a mixture of mathematical datasets, including GSM8K (Grade School Math 8K) (Cobbe et al., 2021), MAWPS (MAth Word ProblemS) (Koncel-Kedziorski et al., 2016), and AQUA-RAT (Algebra Question Answering with Rationales) (Ling et al., 2017), following the setup from (Hu et al., 2023). The model's performance is then evaluated on the test dataset of the AQUA-RAT benchmark.

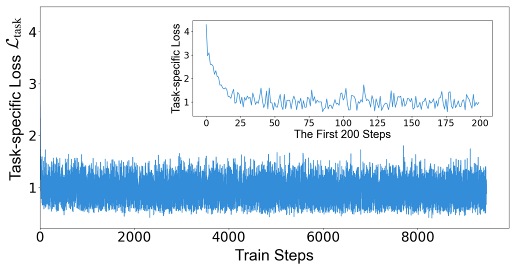
As shown in Table 3, DisLoRA consistently outperforms both LoRA and MiLoRA across all tested ranks. At rank 16, DisLoRA achieves 37.0% accuracy, surpassing LoRA by 1.9 percentage points, and it reaches a peak accuracy of 38.2% at rank 32. These results underscore DisLoRA's robust adaptability to different domains beyond natural language understanding. Its ability to effectively capture task-specific directions proves versatile for quantitative reasoning tasks.

Table 3: The performance of Qwen2.5-7B-Instruct on the AQUA-RAT mathematical reasoning benchmark (Accuracy, %) using LoRA (Hu et al., 2022), MiLoRA (Wang et al., 2025) and DisLoRA.

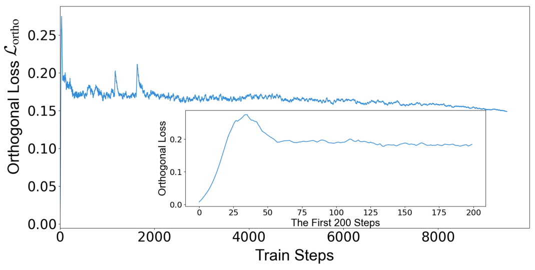
Method	Rank	AQUA-RAT (Acc.)
LoRA	8	$35.1^{\pm 1.4}$
MiLoRA	8	$35.4^{\pm 1.6}$
DisLoRA (ours)	8	$\mathbf{35.8}^{\pm 1.2}$
LoRA	16	$35.1^{\pm 1.5}$
MiLoRA	16	$36.2^{\pm 1.3}$
DisLoRA (ours)	16	$\mathbf{37.0}^{\pm 1.0}$
LoRA	32	$37.8^{\pm 1.7}$
MiLoRA	32	$37.4^{\pm 1.8}$
DisLoRA (ours)	32	$\mathbf{38.2}^{\pm 1.1}$

5 Conclusion

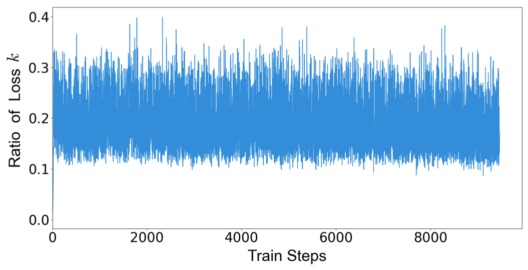
We propose DisLoRA, a parameter-efficient fine-tuning method that uses Singular Value Decomposition to separate pretrained weights into orthogonal backbone and task-specific subspaces, dynamically use orthogonal basis to identify task-specific directions, and adaptively balances losses. On the GLUE benchmark, it outperforms LoRA, PiSSA, and DoRA, achieving top scores in 6/9 tasks with 0.48% parameters. On Commonsense Reasoning



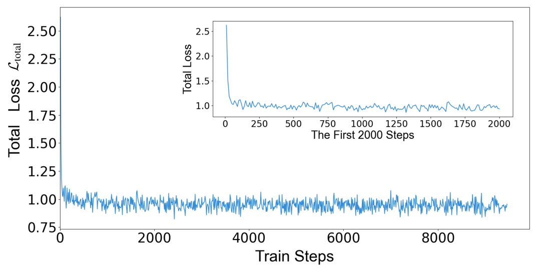
(a) Task-specific loss $\mathcal{L}_{\text{task}}$: the loss decreases rapidly and fluctuates near convergency value over training steps, reflecting effective optimization for the target task in DisLoRA, enabling high performance



(b) Adaptive soft orthogonal regularization with mean-normalization: the orthogonalization loss decreases steadily over training steps, reflecting the weights' increasing orthogonality, which enhances subspace independence and training stability in DisLoRA.



(c) Ratio of task-specific to orthogonalization loss $\frac{\mathcal{L}_{\text{ortho}}}{\mathcal{L}_{\text{task}}}$: the ratio fluctuates over training steps, highlighting the adaptive weighting mechanism's dynamic balance between task optimization and orthogonality in DisLoRA.



(d) Total loss $\mathcal{L}_{\text{total}} = \alpha_{\text{task}} \cdot \mathcal{L}_{\text{task}} + \alpha_{\text{ortho}} \cdot \mathcal{L}_{\text{ortho}}$: the loss decreases rapidly in the beginning and then fluctuates very little near the convergence value, reflecting robust convergence driven by adaptive weighting in DisLoRA.

Figure 4: Loss components in DisLoRA's fine-tuning: (a) task-specific loss converges rapidly, optimizing task performance; (b) orthogonalization loss decreases, ensuring subspace independence; (c) the task-to-orthogonal loss ratio highlights adaptive weighting; (d) total loss reflects overall convergence. These trends demonstrate effective balancing of task optimization and regularization, achieving overall high performance (e.g., The average accuracy of DisLoRA has always been the highest).

Tasks, it surpasses LoRA, LoRA-Dash, SORSA and MiLoRA, reaching 86.2% average accuracy. On AQUA-RAT mathematical dataset, our evaluation confirms DisLoRA's versatility. These results demonstrate its efficacy for efficient and high-performing LLM adaptation.

Limitations

While DisLoRA achieves superior performance in parameter-efficient fine-tuning, it is not without limitations. The SVD required to decompose pretrained weight matrices into backbone and task-specific subspaces is computationally intensive, particularly for large-scale models with high-dimensional weight matrices. This process can significantly increase preprocessing time. Nevertheless, for large-scale matrices the SVD can be accelerated by randomized Krylov iterations, making the overhead acceptable in practice. Additionally, the soft orthogonal regularization loss, which relies on computing the Frobenius norm of singular vector matrices (Equation 6), introduces further computational overhead during training. For models with large dimensions, such as those with billions of parameters, calculating these norms can slow down training speed, potentially offsetting the efficiency gains from parameter reduction. These trade-offs between performance and computational cost highlight the need for optimized SVD algorithms and loss computation strategies to make DisLoRA more practical for very large-scale models. Despite these challenges, the enhanced stability and generalization offered by our approach justify its use in scenarios where model performance is prioritized over training speed.

Acknowledgments

This work was supported by the Beijing Municipal Science and Technology Project (No. Z231100010323002).

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

Reza Yazdani Aminabadi, Samyam Rajbhandari, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Olatunji Ruwase, Shaden Smith, Minjia Zhang, Jeff Rasley, et al. 2022. Deepspeed-inference: enabling efficient inference of transformer models at unprecedented scale. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. 2018. Can we gain more from orthogonality regularizations in training deep cnns? In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4266–4276, Red Hook, NY, USA. Curran Associates Inc.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *CoRR*, abs/1911.11641.

Daniel O. Cajueiro, Arthur G. Nery, Igor Tavares, Maísa K. De Melo, Silvia A. dos Reis, Li Weigang, and Victor R. R. Celestino. 2023. A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding. *Preprint*, arXiv:2301.03403.

Yang Cao. 2024. Sorsa: Singular values and orthonormal regularized singular vectors adaptation of large language models. *Preprint*, arXiv:2409.00055.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, and etc. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Chengcheng Feng, Mu He, Qiuyu Tian, Haojie Yin, Xiaofang Zhao, Hongwei Tang, and Xingqiang Wei. 2024. Trilora: Integrating svd for advanced style personalization in text-to-image generation. *Preprint*, arXiv:2405.11236.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and etc. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Preprint*, arXiv:2403.14608.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. 2024. Agentreview: Exploring peer review dynamics with llm agents. *Preprint*, arXiv:2406.12708.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California. Association for Computational Linguistics.

Moshe Lichtenstein, Prasanna Sattigeri, Rogerio Feris, Raja Giryes, and Leonid Karlinsky. 2020. Tafssl: Task-adaptive feature sub-space learning for few-shot classification. In *European Conference on Computer Vision*, pages 522–539. Springer.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. DoRA: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Junyi Lu, Xiaojia Li, Zihan Hua, Lei Yu, Shiqi Cheng, Li Yang, Fengjun Zhang, and Chun Zuo. 2025. Deepcrceval: Revisiting the evaluation of code review comment generation. *Preprint*, arXiv:2412.18291.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *Preprint*, arXiv:2501.04040.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. Pissa: Principal singular values and singular vectors adaptation of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 121038–121072. Curran Associates, Inc.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, pages 2381–2391.

Leora Morgenstern and Charles L. Ortiz. 2015. The winograd schema challenge: evaluating progress in commonsense reasoning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 4024–4025. AAAI Press.

Lutz Prechelt. 2012. *Early Stopping — But When?*, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, and etc. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728.

Chongjie Si, Zhiyi Shi, Shifan Zhang, Xiaokang Yang, Hanspeter Pfister, and Wei Shen. 2024. Unleashing the power of task-specific directions in parameter efficient fine-tuning. *Preprint*, arXiv:2409.01035.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 2025. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4823–4836, Albuquerque, New Mexico. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *ACL (1)*, pages 4791–4800.

Zhuang Zhan, Xiequn Wang, Yulong Zhang, Wei Li, Yu Zhang, and Ying Wei. 2024. Copra: A progressive lora training strategy. *ArXiv*, abs/2410.22911.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Mathematical Analysis of the Dynamic Adaptive Weighting Mechanism

In the DisLoRA framework, we propose a dynamic adaptive weighting mechanism to balance the task-specific loss $\mathcal{L}_{\text{task}}$ (e.g., cross-entropy loss) and the soft orthogonal regularization loss $\mathcal{L}_{\text{ortho}}$. This appendix provides a rigorous mathematical analysis of the mechanism's definition, feasibility, numerical stability, and gradient properties to validate its correctness and robustness.

A.1 Method Definition

The dynamic adaptive weighting mechanism balances the task-specific loss $\mathcal{L}_{\text{task}}$ (e.g., cross-entropy) and the orthogonal regularization loss $\mathcal{L}_{\text{ortho}}$ to optimize the DisLoRA framework. This subsection defines these components and their integration.

The orthogonal regularization loss enforces orthonormality of the task-specific subspace singular vectors:

$$\mathcal{L}_{\text{ortho}} = \frac{1}{N_{\text{p}}} \sum_{i=1}^{N_{\text{p}}} \left(\left\| \boldsymbol{U}_i^\top \boldsymbol{U}_i - \boldsymbol{I} \right\|_F^2 + \left\| \boldsymbol{V}_i \boldsymbol{V}_i^\top - \boldsymbol{I} \right\|_F^2 \right), \tag{11}$$

where $U_i \in \mathbb{R}^{m\times r}$ and $V_i \in \mathbb{R}^{n\times r}$ are the singular vectors, I is the identity matrix, $\|\cdot\|_F$ denotes the Frobenius norm, and N_{p} is the number of singular vector pairs across model layers. The normalization by N_{p} ensures scalability across diverse architectures.

To balance $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{ortho}}$, we define a relative loss ratio:

$$k = \frac{\mathcal{L}_{\text{ortho}}}{\mathcal{L}_{\text{task}} + \varepsilon}, \tag{12}$$

where $\varepsilon > 0$ (e.g., 10^{-6}) prevents division by zero. Adaptive weights are computed via a softmax-like function:

$$\alpha_{\text{task}} = \frac{\exp(-k)}{\exp(-k) + \exp(-1/k + \varepsilon)}, \tag{13}$$

$$\alpha_{\text{ortho}} = 1 - \alpha_{\text{task}}. \tag{14}$$

The total loss is:

$$\mathcal{L}_{\text{total}} = \alpha_{\text{task}} \cdot \mathcal{L}_{\text{task}} + \alpha_{\text{ortho}} \cdot \mathcal{L}_{\text{ortho}}. \tag{15}$$

The constant ε in (12) ensures numerical stability, while the additional ε in (13) fine-tunes the weight balance by slightly adjusting the denominator. This mechanism dynamically prioritizes $\mathcal{L}_{\text{task}}$ or $\mathcal{L}_{\text{ortho}}$ based on their relative magnitudes, ensuring robust optimization without manual tuning.

A.2 Feasibility: Non-Negativity and Normalization

We verify that the weights α_{task} and α_{ortho}, defined in (13) and (14), satisfy non-negativity and normalization conditions, ensuring their suitability for weighted loss combination.

- **Non-Negativity**: Since $\mathcal{L}_{\text{ortho}} \geq 0$, $\mathcal{L}_{\text{task}} \geq 0$, and $\mathcal{L}_{\text{task}} + \varepsilon > 0$ with $\varepsilon > 0$ (e.g., 10^{-6}), the loss ratio $k = \mathcal{L}_{\text{ortho}}/(\mathcal{L}_{\text{task}} + \varepsilon)$ in (12) satisfies $k \geq 0$. For α_{task} in (13), the numerator $\exp(-k) \geq 0$, and the denominator $\exp(-k) + \exp(-1/k + \varepsilon) > 0$, as $\exp(-1/k + \varepsilon) \geq 0$ for $\varepsilon > 0$ (e.g., 10^{-6}) and all $k \geq 0$ (noting that $\exp(-1/k + \varepsilon) \to \exp(\varepsilon) > 0$ as $k \to 0$). Thus, $\alpha_{\text{task}} \geq 0$. Since $\exp(-k) \leq \exp(-k) + \exp(-1/k + \varepsilon)$, we have $\alpha_{\text{task}} \leq 1$. For $\alpha_{\text{ortho}} = 1 - \alpha_{\text{task}}$, it follows that $0 \leq \alpha_{\text{ortho}} \leq 1$.
- **Normalization**: By definition, $\alpha_{\text{ortho}} = 1 - \alpha_{\text{task}}$, so:

$$\alpha_{\text{task}} + \alpha_{\text{ortho}} = \alpha_{\text{task}} + (1 - \alpha_{\text{task}}) = 1.$$

These properties confirm that α_{task} and α_{ortho} are well-defined, non-negative, and normalized, making them suitable for balancing $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{ortho}}$ in (15).

A.3 Dynamic Adjustment Behavior

To validate the adaptive behavior of the weights α_{task} and α_{ortho} defined in (13) and (14), we analyze their variation with the loss ratio k from (12). Define:

$$a = \exp(-k), \quad b = \exp(-1/k + \varepsilon), \tag{16}$$

so that:

$$\alpha_{\text{task}} = \frac{a}{a+b}, \quad \alpha_{\text{ortho}} = \frac{b}{a+b}. \tag{17}$$

We examine three scenarios to illustrate how the mechanism prioritizes $\mathcal{L}_{\text{task}}$ or $\mathcal{L}_{\text{ortho}}$ based on their relative magnitudes.

Case 1: Small k ($\mathcal{L}_{\text{ortho}} \ll \mathcal{L}_{\text{task}}$). As $k \to 0$, which occurs when $\mathcal{L}_{\text{ortho}}$ is much smaller than $\mathcal{L}_{\text{task}} + \varepsilon$, we have:

$$a = \exp(-k) \to 1, \quad b = \exp(-1/k + \varepsilon) \to 0, \tag{18}$$

since $-1/k \to -\infty$ dominates ε. Thus, $\alpha_{\text{task}} = a/(a + b) \to 1$ and $\alpha_{\text{ortho}} = b/(a + b) \to 0$, so

$\mathcal{L}_{\text{total}} \to \mathcal{L}_{\text{task}}$ in (10), prioritizing task-specific optimization.

Case 2: Large k ($\mathcal{L}_{\text{ortho}} \gg \mathcal{L}_{\text{task}}$). As $k \to \infty$, which occurs when $\mathcal{L}_{\text{ortho}}$ is much larger than $\mathcal{L}_{\text{task}} + \varepsilon$, we have:

$$a = \exp(-k) \to 0, \quad b = \exp(-1/k+\varepsilon) \to \exp(\varepsilon), \tag{19}$$

where $\exp(\varepsilon) \approx 1$ for small ε (e.g., 10^{-6}). Thus, $\alpha_{\text{task}} \to 0$ and $\alpha_{\text{ortho}} \to 1$, so $\mathcal{L}_{\text{total}} \to \mathcal{L}_{\text{ortho}}$, emphasizing orthogonal regularization.

Case 3: Balanced k ($\mathcal{L}_{\text{ortho}} \approx \mathcal{L}_{\text{task}}$). When $k = 1$, implying $\mathcal{L}_{\text{ortho}} \approx \mathcal{L}_{\text{task}} + \varepsilon$, we have:

$$a = \exp(-1), \quad b = \exp(-1+\varepsilon) \approx \exp(-1), \tag{20}$$

since ε is small. Thus, $\alpha_{\text{task}} \approx a/(a+a) = 0.5$ and $\alpha_{\text{ortho}} \approx 0.5$, so $\mathcal{L}_{\text{total}}$ balances both losses equally.

This adaptive mechanism dynamically prioritizes the dominant loss, ensuring robust optimization and stability in (15) across varying loss magnitudes.

A.4 Gradient Analysis

We analyze the gradient of the total loss $\mathcal{L}_{\text{total}}$ with respect to model parameters θ, defined in (15) as:

$$\mathcal{L}_{\text{total}} = \alpha_{\text{task}} \cdot \mathcal{L}_{\text{task}} + \alpha_{\text{ortho}} \cdot \mathcal{L}_{\text{ortho}}. \tag{21}$$

In practice, α_{task} and α_{ortho} are computed using detached loss values (e.g., via `.detach()` in PyTorch) to treat them as constants during gradient computation. The gradient is:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial \boldsymbol{\theta}} = \alpha_{\text{task}} \cdot \frac{\partial \mathcal{L}_{\text{task}}}{\partial \boldsymbol{\theta}} + \alpha_{\text{ortho}} \cdot \frac{\partial \mathcal{L}_{\text{ortho}}}{\partial \boldsymbol{\theta}}. \tag{22}$$

Task Loss Gradient. In supervised fine-tuning (SFT), the task-specific loss is typically a cross-entropy loss tailored to the task. For classification tasks, it is defined as:

$$\mathcal{L}_{\text{task}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} \boldsymbol{y}_{i,c} \log(\hat{\boldsymbol{y}}_{i,c}), \tag{23}$$

where $\boldsymbol{y}_{i,c} \in \{0,1\}$ is the true label for class c, and $\hat{\boldsymbol{y}}_{i,c} = f(\boldsymbol{x}_i;\boldsymbol{\theta}) \in (0,1)$ is the predicted probability for class c from the model f parameterized by θ. For generative tasks, the loss is the negative log-likelihood:

$$\mathcal{L}_{\text{task}} = -\frac{1}{T}\sum_{t=1}^{T} \log P(\boldsymbol{x}_t|\boldsymbol{x}_{<t};\boldsymbol{\theta}). \tag{24}$$

The gradient $\frac{\partial \mathcal{L}_{\text{task}}}{\partial \boldsymbol{\theta}}$ is computed via backpropagation, with its form depending on the task and model architecture.

Orthogonal Loss Gradient. The orthogonal regularization loss, defined in (11), is:

$$\mathcal{L}_{\text{ortho}} = \frac{1}{N_{\text{p}}}\sum_{i=1}^{N_{\text{p}}}\left(\left\|\boldsymbol{U}_i^{\top}\boldsymbol{U}_i - \boldsymbol{I}\right\|_F^2 + \left\|\boldsymbol{V}_i\boldsymbol{V}_i^{\top} - \boldsymbol{I}\right\|_F^2\right), \tag{25}$$

where $\boldsymbol{U}_i \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V}_i \in \mathbb{R}^{n\times r}$ are singular vectors derived from weight matrices parameterized by $\boldsymbol{\theta}$, and N_{p} is the number of singular vector pairs. For the term $\left\|\boldsymbol{U}_i^{\top}\boldsymbol{U}_i - \boldsymbol{I}\right\|_F^2$, we express the Frobenius norm as:

$$\left\|\boldsymbol{U}_i^{\top}\boldsymbol{U}_i - \boldsymbol{I}\right\|_F^2 = \operatorname{tr}\left((\boldsymbol{U}_i^{\top}\boldsymbol{U}_i - \boldsymbol{I})^2\right), \tag{26}$$

with gradient:

$$\frac{\partial\left\|\boldsymbol{U}_i^{\top}\boldsymbol{U}_i - \boldsymbol{I}\right\|_F^2}{\partial \boldsymbol{U}_i} = 4\boldsymbol{U}_i(\boldsymbol{U}_i^{\top}\boldsymbol{U}_i - \boldsymbol{I}). \tag{27}$$

Similarly, for $\left\|\boldsymbol{V}_i\boldsymbol{V}_i^{\top} - \boldsymbol{I}\right\|_F^2$:

$$\frac{\partial\left\|\boldsymbol{V}_i\boldsymbol{V}_i^{\top} - \boldsymbol{I}\right\|_F^2}{\partial \boldsymbol{V}_i} = 4(\boldsymbol{V}_i\boldsymbol{V}_i^{\top} - \boldsymbol{I})\boldsymbol{V}_i. \tag{28}$$

The gradient of $\mathcal{L}_{\text{ortho}}$ with respect to θ is:

$$\frac{\partial \mathcal{L}_{\text{ortho}}}{\partial \boldsymbol{\theta}} = \frac{1}{N_{\text{p}}}\sum_{i=1}^{N_{\text{p}}}\left(\frac{\partial\left\|\boldsymbol{U}_i^{\top}\boldsymbol{U}_i - \boldsymbol{I}\right\|_F^2}{\partial \boldsymbol{\theta}} + \frac{\partial\left\|\boldsymbol{V}_i\boldsymbol{V}_i^{\top} - \boldsymbol{I}\right\|_F^2}{\partial \boldsymbol{\theta}}\right), \tag{29}$$

where the terms depend on the relationship between $\boldsymbol{U}_i$, $\boldsymbol{V}_i$, and $\boldsymbol{\theta}$ via the chain rule.

The total gradient in (22) is a convex combination of $\frac{\partial \mathcal{L}_{\text{task}}}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \mathcal{L}_{\text{ortho}}}{\partial \boldsymbol{\theta}}$, with weights α_{task} and α_{ortho} adapting to the relative magnitudes of the losses. Rapid fluctuations in $\mathcal{L}_{\text{task}}$ or $\mathcal{L}_{\text{ortho}}$ may cause variations in α_{task} and α_{ortho}, potentially leading to unstable gradient directions. Smoothing techniques, such as exponential moving averages of loss values, can stabilize the weights and improve optimization robustness.

B Datasets, Metrics and Hyperparameters

B.1 Description of Benchmarks

GLUE Benchmark: We evaluate our model on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019). GLUE comprises nine datasets designed to assess a broad range of natural language understanding tasks: CoLA (Warstadt et al., 2019) (Corpus of Linguistic Acceptability, grammatical acceptability), SST-2 (Socher et al., 2013) (Stanford

Table 4: Hyperparameter settings for GLUE benchmark datasets.

Hyperparameter	Debertav3-base								
	SST-2	**MRPC**	**CoLA**	**QNLI**	**RTE**	**STS-B**	**MNLI**	**QQP**	**WNLI**
	AdamW								
LR Schedule	Linear								
Rank r	16, 32, 64								
LoRA Scaling	1.5, 2.0								
Dropout	0.1								
Max Seq. Len.	128								
Batch Size	32	8	8	32	8	8	64	40	8
Learning Rate	1e-3	3e-3	8e-4	3e-3	3e-3	3e-3	3e-3	1e-3	3e-3
Training Duration	55min	3min	7 min	30min	2min	4min	110min	110min	1min
Epochs	4								

Sentiment Treebank 2, sentiment classification), MRPC (Dolan and Brockett, 2005) (Microsoft Research Paraphrase Corpus, paraphrase detection), QQP (Quora Question Pairs, question paraphrase detection), STS-B (Cer et al., 2017) (Semantic Textual Similarity Benchmark, semantic similarity), MNLI (Williams et al., 2018) (Multi-Genre Natural Language Inference, natural language inference, matched and mismatched), QNLI (Rajpurkar et al., 2016) (Question Natural Language Inference, question-answering entailment), RTE (Recognizing Textual Entailment, textual entailment), and WNLI (Morgenstern and Ortiz, 2015) (Winograd Natural Language Inference, coreference resolution). These tasks primarily involve single-sentence classification, sentence-pair similarity, and paraphrase evaluation

Commonsense Reasoning: For commonsense reasoning, we fine-tune our model on the Commonsense170K dataset (Hu et al., 2023). This dataset aggregates the training sets from eight distinct commonsense reasoning benchmarks: BoolQ (Clark et al., 2019) (Boolean Questions, binary question answering), PIQA (Bisk et al., 2019) (Physical Interaction Question Answering, physical commonsense), SIQA (Sap et al., 2019) (Social Interaction Question Answering, social commonsense), HellaSwag (Zellers et al., 2019) (sentence completion, narrative reasoning), WinoGrande (Sakaguchi et al., 2019) (Winograd Schema Challenge, pronoun disambiguation), ARC-e and ARC-c (Clark et al., 2018) (AI2 Reasoning Challenge, easy and challenge sets, scientific reasoning), and OBQA (Mihaylov et al., 2018) (Open Book Question Answering, scientific knowledge integration). These tasks, typically structured as multiple-choice questions, evaluate diverse commonsense reasoning abilities.

B.2 Hyperparameter Settings

This subsection details the hyperparameter configurations employed during the fine-tuning process for both the GLUE benchmark and the commonsense reasoning tasks. The specific settings for each are presented in the tables below.

B.2.1 GLUE Benchmark

The hyperparameter settings used for fine-tuning on the GLUE benchmark datasets are summarized in Table 4.

B.2.2 Commonsense Reasoning

Table 5 outlines the hyperparameter settings for the commonsense reasoning tasks.

Table 5: Hyperparameter settings for Commonsense Reasoning.

Hyperparameter	Qwen2.5-7B-Instruct		
	$r = 8$	$r = 16$	$r = 32$
Rank r	8	16	32
α	16	32	48
Learning Rate	5e-4	3e-4	3e-4
LR Scheduler	Linear		
Dropout	0.05		
Optimizer	AdamW		
Batch Size	18		
Micro Batch Size	6		
Epochs	3		
Training Duration	6.5 hours		

B.3 Hyperparameter Sensitivity Analysis

DisLoRA introduces a key hyperparameter s_{tsd} to determine the number of task-specific directions selected for optimization after the warmup phase. These directions correspond to the singular vectors with the highest change rates, signifying their importance for the downstream task. While the primary experiments in this paper use a fixed s_{tsd} of 8 based on prior validation, its optimal value may vary across different tasks and datasets.

To investigate the sensitivity of DisLoRA's performance to this hyperparameter, we conducted an analysis using the DeBERTaV3-base model on three distinct GLUE tasks: MRPC, CoLA, and STSB. We fine-tuned the model with s_{tsd} values of 6, 8, and 10. As shown in the Table 6, varying values of s_{tsd} do influence the results; nevertheless, the overall trend aligns with the recommendations in the (Si et al., 2024), with $s_{\text{tsd}} = 8$ yielding an optimal balance.

Table 6: Sensitivity analysis of the s_{tsd} hyperparameter. Performance of DeBERTaV3-base on MRPC, CoLA, and STSB datasets for different values of s_{tsd}.

s_{tsd}	MRPC (F1.)	CoLA (Mcc.)	STSB (Pcc.)
6	$92.55^{\pm 0.29}$	$67.82^{\pm 0.33}$	$90.72^{\pm 0.92}$
8	$93.12^{\pm 0.57}$	$68.33^{\pm 0.52}$	$89.73^{\pm 1.01}$
10	$92.57^{\pm 0.41}$	$66.92^{\pm 0.41}$	$91.03^{\pm 0.81}$