LoRA-DA: Data-Aware Initialization for Low-Rank Adaptation via Asymptotic Analysis

Anonymous authors
Paper under double-blind review

Abstract

With the widespread adoption of LLMs, LoRA has become a dominant method for PEFT, and its initialization methods have attracted increasing attention. However, existing methods have notable limitations: many methods do not incorporate target-domain data, while gradient-based methods exploit data only at a shallow level by relying on one-step gradient decomposition, which remains unsatisfactory due to the weak empirical performance of the one-step fine-tuning model that serves as their basis, as well as the fact that these methods either lack a rigorous theoretical foundation or depend heavily on restrictive isotropic assumptions. In this paper, we establish a theoretical framework for data-aware LoRA initialization based on asymptotic analysis. Starting from a general optimization objective that minimizes the expectation of the parameter discrepancy between the fine-tuned and target models, we derive an optimization problem with two components: a bias term, which is related to the parameter distance between the fine-tuned and target models, and is approximated using a Fisher–gradient formulation to preserve anisotropy; and a variance term, which accounts for the uncertainty introduced by sampling stochasticity through the Fisher information. By solving this problem, we obtain an optimal initialization strategy for LoRA. Building on this theoretical framework, we develop an efficient algorithm, LoRA-DA, which estimates the terms in the optimization problem from a small set of target domain samples and obtains the optimal LoRA initialization. Empirical results across multiple benchmarks demonstrate that LoRA-DA consistently improves final accuracy over existing initialization methods. Additional studies show faster, more stable convergence, robustness across ranks, and only a small initialization overhead for LoRA-DA. The source code will be released upon publication.

1 Introduction

In recent years, large language models (LLMs) have advanced at an unprecedented pace, reshaping research in natural language processing and neighboring areas. By scaling parameters, data, and compute, LLMs exhibit strong generalization across diverse domains. Representative studies report compelling progress in instruction-following dialogue (Ouyang et al., 2022), code reasoning and synthesis (Li et al., 2022; Wang et al., 2021), commonsense reasoning (Toroghi et al., 2025), mathematical problem solving (Wei et al., 2022; Wang et al., 2023a; Hendrycks et al., 2021), and multimodal understanding with vision–language modeling (Li et al., 2023; Dai et al., 2023). However, these models often require full-parameter fine-tuning, the cost of which is prohibitively large.

To alleviate the prohibitive cost of full-parameter fine-tuning, a growing body of research has focused on parameter-efficient fine-tuning (PEFT) techniques, which adapt large models by introducing only a small set of trainable parameters. Among them, Low-Rank Adaptation (LoRA) has emerged as a highly influential approach, which injects low-rank matrices into pretrained weight space to enable efficient adaptation without modifying the original parameters (Hu et al., 2022). Moreover, LoRA-FA simplifies LoRA by freezing A and training only B, cutting trainable parameters by half while maintaining competitive performance. Despite these advantages, LoRA and LoRA-FA share a common initialization scheme: the A matrix is randomly initialized (or frozen as random in LoRA-FA), while the B matrix is initialized to zero. Such a scheme ensures that no adaptation is applied at the very beginning of training, but it also introduces two drawbacks (Meng et al.,

2024; Zhang et al., 2025b): (i) the training process starts slowly due to the absence of informative initialization, and (ii) the models may fail to converge to an optimal solution.

Compared with conventional random initialization in LoRA, recent work has highlighted the critical role of initialization for both convergence speed and final performance, motivating alternative initialization strategies. Prior approaches were generally data-agnostic, without incorporating information from the target task, such as PiSSA (Meng et al., 2024) and MiLoRA (Wang et al., 2025), which adapt the singular vectors or singular values of pretrained weight matrices to exploit the inherent structural properties of the original parameters. More recently, several data-aware approaches have been proposed that leverage a small set of target-domain samples for initialization. For example, LoRA-GA (Wang et al., 2024a) and LoRA-One (Zhang et al., 2025b) exploit target-domain gradients to construct the low-rank subspace. However, their own experiments show that the performance of one-step fine-tuning is not only suboptimal but also markedly inferior to that of vanilla LoRA, raising concerns about whether gradient-only initialization is sufficient. Moreover, their conclusions either lack rigorous theoretical support or rely on strong mathematical assumptions about the input vector of the layer, such as isotropic centered sub-Gaussian assumption. However, recent empirical studies have shown that representations in transformer-based models are far from isotropic (Godey et al., 2024). In our view, relying solely on gradients to approximate the parameter discrepancy overlooks the anisotropy of the parameter space. Moreover, beyond parameter discrepancy, the variance induced by sampling stochasticity also contributes to the training error, yet such methods fail to account for it in their initialization strategies.

In this work, we propose a data-aware LoRA initialization method grounded in asymptotic analysis. Specifically, we formulate the optimization objective as minimizing the expectation of the discrepancy between the parameters of fine-tuned and target models. By applying asymptotic analysis, we reformulate the optimization of the bound of this objective into a quadratic optimization problem with the LoRA initialization parameters as target variables. The **Initialization Guidance Matrix**, serving as the coefficient matrix in the quadratic optimization, consists of two terms: a **variance term**, capturing the uncertainty caused by sampling stochasticity through Fisher information, and a **bias term**, which relates to the discrepancy between the fine-tuned and target models, approximated via a Fisher–gradient approach that preserves the anisotropic structure of the parameter space. Solving this problem yields an optimal initialization strategy for LoRA. Building upon this theoretical result, we propose LoRA-DA, a general data-aware LoRA initialization algorithm. Extensive experiments across multiple tasks demonstrate the effectiveness and robustness of our algorithm.

In summary, our contributions are as follows:

1. **Theoretical foundation.** We establish a theoretical framework for data-aware LoRA initialization based on asymptotic analysis. Our formulation decomposes the fine-tuning error into a variance term and a bias term, yielding a method for computing the optimal LoRA initialization.
2. **Algorithm design.** Building on these theoretical insights, we propose LoRA-DA, a practical algorithm that leverages a small set of target-domain samples to estimate the statistics required by our theoretical framework and derive the optimal initialization of LoRA. The proposed initialization algorithm is architecture-agnostic.
3. **Empirical validation.** We conduct extensive experiments on both natural language understanding benchmarks and natural language generation benchmarks, where LoRA-DA consistently outperforms state-of-the-art initialization methods, achieving average improvements of **0.3%** on natural language understanding and **1.0%** on natural language generation over prior SOTA. Additional studies confirm faster and more stable convergence, robustness across ranks, and a small initialization overhead of LoRA-DA.

2 PRELIMINARIES

In this section, we introduce the mathematical preliminaries required for our theoretical framework. Section 2.1 provides the formal definition and background of LoRA and LoRA-FA. Section 2.2 presents the asymptotic normality of the maximum likelihood estimator (MLE) used to model sampling stochasticity, and the Fisher information matrix which is not only related to asymptotic normality but also used to model the anisotropy of the parameter space. Finally, Section 2.3 reviews

the correspondence between the decomposition of eigenvalues and the solution of the quadratic optimization problem, which we will exploit in our method.

2.1 Low-Rank Adaptation (LoRA) and LoRA-FA

LoRA enables PEFT of pre-trained networks by inserting low-dimensional trainable components. Instead of fine-tuning all existing weights, LoRA appends two compact matrices $\boldsymbol{A} \in \mathbb{R}^{d_1 \times r}$ and $\boldsymbol{B} \in \mathbb{R}^{r \times d_2}$, with $r \ll \min(d_1, d_2)$. The weight adaptation of LoRA can be expressed as

$$Y = Z\hat{\boldsymbol{W}} = Z\big(\boldsymbol{W}_0 + \Delta\boldsymbol{W}\big) = Z\big(\boldsymbol{W}_0 + \boldsymbol{A}\boldsymbol{B}\big), \tag{1}$$

where Z is the input, Y is the output, and $\Delta\boldsymbol{W}$ denotes the low-rank update. Typically, $\boldsymbol{A}$ is initialized from a Kaiming uniform distribution (He et al., 2015), and $\boldsymbol{B}$ is initialized to zeros .

LoRA-FA (LoRA with Frozen-A) is a memory-efficient variant of standard LoRA. In conventional LoRA, both low-rank matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are trained, requiring storage for gradients and optimizer states of both. LoRA-FA freezes $\boldsymbol{A}$ after initialization, and only updates $\boldsymbol{B}$. This eliminates the need to store $\boldsymbol{A}$'s activations and optimizer states, substantially reducing fine-tuning memory usage (Zhang et al., 2023a). The design of LoRA-FA is further supported by empirical and theoretical evidence. The original LoRA work (Hu et al., 2022) found that assigning a larger learning rate to $\boldsymbol{B}$ than to $\boldsymbol{A}$ leads to better performance, suggesting that $\boldsymbol{B}$ contributes more to adaptation. More recent work (Zhu et al., 2024) systematically studied this asymmetry, showing that training $\boldsymbol{B}$ is inherently more effective, while frozen $\boldsymbol{A}$ has little impact on final accuracy.

2.2 Asymptotic Normality of the Maximum Likelihood Estimator (MLE)

When estimating the underlying parameter $\boldsymbol{\theta}^* \in \mathbb{R}^d$ from i.i.d. samples drawn from $P_{X;\boldsymbol{\theta}^*}$, we denote $\mathcal{D}$ as a set of N i.i.d. samples from the distribution. Then the MLE can be expressed as

$$\hat{\boldsymbol{\theta}}_{MLE} = \arg\max_{\boldsymbol{\theta}} \frac{1}{N} \sum_{x \in \mathcal{D}} \log P_{X;\boldsymbol{\theta}}(x). \tag{2}$$

Under standard regularity conditions, the MLE satisfies the following asymptotic normality:

$$\sqrt{N}\left(\hat{\boldsymbol{\theta}}_{\text{MLE}} - \boldsymbol{\theta}^*\right) \xrightarrow{d} \mathcal{N}\left(0, \boldsymbol{J}(\boldsymbol{\theta}^*)^{-1}\right), \tag{3}$$

where "-1" denotes the matrix inverse and $\boldsymbol{J}(\boldsymbol{\theta})$ is the Fisher information matrix defined as:

$$\boldsymbol{J}(\boldsymbol{\theta})^{d\times d} = \mathbb{E}\left[\left(\frac{\partial}{\partial \boldsymbol{\theta}} \log P_{X;\boldsymbol{\theta}}\right)\left(\frac{\partial}{\partial \boldsymbol{\theta}} \log P_{X;\boldsymbol{\theta}}\right)^\top\right]. \tag{4}$$

Intuitively, the Fisher information matrix measures the sensitivity of the model to perturbations along different parameter directions, thus serving as a descriptor of the anisotropy in the parameter space.

2.3 Eigenvalue Decomposition for Quadratic Form Minimization

We consider the constrained quadratic minimization problem

$$\min_{\boldsymbol{Q} \in \mathbb{R}^{d\times r}} \operatorname{tr}(\boldsymbol{Q}^\top \boldsymbol{M} \boldsymbol{Q}) \quad \text{s.t.} \quad \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}_r, \tag{5}$$

where $\boldsymbol{M} \in \mathbb{R}^{d\times d}$ is symmetric. By the eigenvalue decomposition

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top, \quad \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d), \quad \lambda_1 \geq \cdots \geq \lambda_d, \tag{6}$$

the Courant–Fischer min–max theorem (Horn & Johnson, 2012) ensures that the optimal solution is

$$\boldsymbol{Q}^* = [\boldsymbol{u}_d \quad \boldsymbol{u}_{d-1} \quad \cdots \quad \boldsymbol{u}_{d-r+1}], \tag{7}$$

where $\boldsymbol{u}_i$ is the eigenvector corresponding to the eigenvalue λ_i. The minimum objective value is

$$\min_{\boldsymbol{Q}} \operatorname{tr}(\boldsymbol{Q}^\top \boldsymbol{M} \boldsymbol{Q}) = \sum_{i=d-r+1}^{d} \lambda_i. \tag{8}$$

3 PROBLEM FORMULATION

Let $P_{X,\boldsymbol{W}_0}$ denote a pre-trained model, and $P_{X,\boldsymbol{W}_{\text{tgt}}}$ denotes the target model in the fine-tuning process, where $\boldsymbol{W} \in \mathbb{R}^{d_1 \times d_2}$ is the parameter matrix, and X represents the joint distribution of inputs and outputs. For example, when the task is a supervised classification task, $P_{X,\boldsymbol{W}_0}$ corresponds to the joint distribution model of input features Z and output labels Y, *i.e.*, $X = (Z, Y)$. Suppose that we are given a set of N samples $\{x_1, \ldots, x_N\}$ i.i.d. drawn from $P_{X,\boldsymbol{W}_{\text{tgt}}}$ for the fine-tuning of LoRA, whose form is

$$\hat{\boldsymbol{W}} = \boldsymbol{W}_0 + \boldsymbol{A}\boldsymbol{B}, \quad \boldsymbol{A} \in \mathbb{R}^{d_1 \times r},\ \boldsymbol{B} \in \mathbb{R}^{r \times d_2}, \quad r \ll \min(d_1, d_2). \tag{9}$$

Fine-tuning with cross-entropy is equivalent to MLE, while LoRA restricts updates to a low-rank subspace, yielding a constrained MLE:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W} \in \left\{ \boldsymbol{W}_0 + \boldsymbol{A}\boldsymbol{B} \;\middle|\; \boldsymbol{A} \in \mathbb{R}^{d_1 \times r},\ \boldsymbol{B} \in \mathbb{R}^{r \times d_2} \right\}}{\arg\max} \frac{1}{N} \sum_{i=1}^{N} \log P_{X,\boldsymbol{W}}(x_i). \tag{10}$$

We first investigate the initialization of matrix $\boldsymbol{A}$ under the LoRA-FA framework, i.e., the setting where $\boldsymbol{A}$ remains frozen during fine-tuning. Considering the practical similarity between LoRA-FA and full LoRA in terms of both training behavior and empirical performance, as illustrated in Subsection 2.1, our analysis in the LoRA-FA setting does not compromise generality. The initialization strategy derived from this theoretical analysis is not restricted to LoRA-FA but can also be directly applied to standard LoRA. In addition, we constrain the $\boldsymbol{A}$ to be column-orthogonal, i.e. $\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}_r$, which prevents redundancy among low-rank directions, and facilitates better-conditioned optimization. Our objective is to design a more effective initialization $\boldsymbol{A}_0$ such that the resulting estimator $\hat{\boldsymbol{W}}$ minimizes the expected squared Frobenius norm to the true parameter $\boldsymbol{W}_{\text{tgt}}$, i.e.,

$$\boldsymbol{A}_0^* = \arg\min_{\boldsymbol{A}} \ \mathbb{E}\left[\left\| \hat{\boldsymbol{W}} - \boldsymbol{W}_{\text{tgt}} \right\|_F^2 \right], \tag{11}$$

where $\|\cdot\|_F$ denotes the Frobenius norm (Horn & Johnson, 2012), and $\hat{\boldsymbol{W}}$ is defined as equation 10, thereby being associated with $\boldsymbol{A}$. To facilitate the subsequent mathematical derivations, we assume that the distance between the parameters of the target sample model and the source pre-trained model is sufficiently small, *i.e.*, $\|\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0\|_F = O\left(\frac{1}{\sqrt{N}}\right)$. This assumption is made without loss of generality, since fine-tuning typically targets tasks near the pre-training model. Beyond the well-known similarity in early layers (Raghu et al., 2019), cross-layer evidence shows that keeping weights close to the pre-trained parameters benefits fine-tuning stability and generalization (Lee et al., 2020; Chen et al., 2020). Similar assumptions have also been adopted in the transfer learning literature (Zhang et al., 2025a).

4 MAIN RESULT

In this section, we present the procedure by which the optimal initialization of LoRA is derived through minimizing the objective function equation 11. To make the mathematical derivation and results more accessible to the reader, we first provide the theoretical analysis in the simplified case where the output dimension is $d_2 = 1$ in Section 4.1. We then extend the result to the general high-dimensional setting of standard LoRA in Section 4.2, and finally describe the practical algorithm LoRA-DA in Section 4.3.

4.1 ONE-DIMENSIONAL CASE

We first consider the simple case where the output dimension is $d_2 = 1$, i.e., $\boldsymbol{W} \in \mathbb{R}^{d_1 \times 1}$. In this setting, the estimator can be expressed as $\hat{\boldsymbol{W}} = \boldsymbol{W}_0 + \boldsymbol{A}\,\boldsymbol{B}$, where $\boldsymbol{A} \in \mathbb{R}^{d_1 \times r}$ is a fixed column-orthogonal matrix and $\boldsymbol{B} \in \mathbb{R}^{r \times 1}$. Then we have the following theorem.

Theorem 1. *(proved in Appendix C.1) In the case where the output dimension $d_2 = 1$, the optimal initialization of the matrix A is given by*

$$\boldsymbol{A}_{\boldsymbol{0}}^* = \arg\min_{\boldsymbol{A}} \operatorname{tr}\left(\boldsymbol{A}^\top \boldsymbol{\Omega} \boldsymbol{A} \right), \tag{12}$$

and from Section 2.3 we know that the r column vectors of $\boldsymbol{A}_0^*$ correspond to the eigenvectors of $\boldsymbol{\Omega}$ associated with its r smallest eigenvalues, where $\boldsymbol{\Omega}$ is the ***Initialization Guidance Matrix*** *given by*

$$\boldsymbol{\Omega}^{d_1 \times d_1} = \left(\underbrace{\frac{\boldsymbol{J}(\boldsymbol{W}_0)^{-1}}{N}}_{\textit{variance term}} \underbrace{-\big(\boldsymbol{W}_{tgt} - \boldsymbol{W}_0\big)\big(\boldsymbol{W}_{tgt} - \boldsymbol{W}_0\big)^\top}_{\textit{bias term}} \right) \tag{13}$$

In Figure 1, we provide an explanation of the result in Theorem 1. Specifically, the bound of the objective function in equation 11 can be decomposed into two components: a variance term models the sampling stochasticity associated with the Fisher information and sample size, and a bias term arising from the discrepancy between the target parameter $\boldsymbol{W}_{\text{tgt}}$ and the LoRA subspace determined by $\boldsymbol{W}_0$ and fixed A. It should be noted that the bias term in equation 13 excludes the invariant component of the optimization, and thus appears with a minus sign. Its complete form can be found in equation 42.

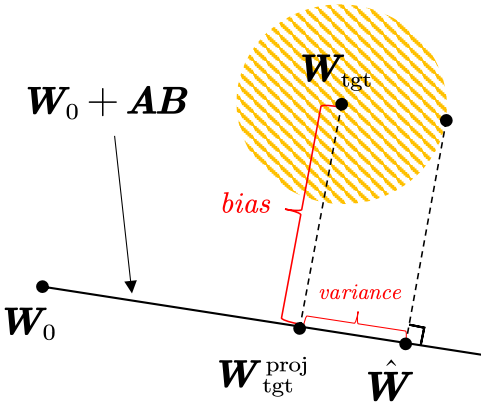


Figure 1: The yellow circle illustrates the estimation variance induced by the stochasticity of training samples in the unconstrained setting. The red variance term represents its projection onto the LoRA subspace under the fixed-$\boldsymbol{A}$ constraint, while the red bias term corresponds to the approximation error due to the distance between $\boldsymbol{W}_{\text{tgt}}$ and the LoRA subspace.

In equation 13, a natural question arises: how to approximate the term $\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0$, i.e., the discrepancy between the target parameters $\boldsymbol{W}_{\text{tgt}}$ and the pre-trained parameters $\boldsymbol{W}_0$. Prior works often approximate this discrepancy directly from the negative raw gradient. We instead adopt a more effective approach **Fisher-gradient**, approximating it as the negative inverse Fisher times the gradient. Unlike the raw gradient, this Fisher-weighted form adaptively scales directions by their uncertainty or information content, thereby capturing the model's anisotropy. The Fisher-gradient formulation builds on the classical notion of the natural gradient (Amari, 1998). In our framework, the Fisher matrix is already obtained in the process of evaluating the variance term and can therefore be reused, which provides a natural motivation for incorporating the Fisher-gradient formulation into our initialization strategy. We next provide the theoretical justification and formulation of this approach.

We employ a local second-order expansion around $\boldsymbol{W}_0$. Specifically, we define the empirical loss on the target distribution as $\mathcal{L}_{\text{tgt}}(\boldsymbol{W}) = \frac{1}{N}\sum_{j=1}^{N} \ell(\boldsymbol{W}; z_{\text{tgt}}^j, y_{\text{tgt}}^j)$, where the subscript "tgt" indicates that the loss is computed on the fine-tuning samples drawn from $P_{X,\boldsymbol{W}_{\text{tgt}}}$. Expanding this loss around $\boldsymbol{W}_0$ yields

$$\mathcal{L}_{\text{tgt}}(\boldsymbol{W}_{\text{tgt}}) \approx \mathcal{L}_{\text{tgt}}(\boldsymbol{W}_0) + \boldsymbol{G}^\top(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0) + \tfrac{1}{2}(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)^\top \boldsymbol{H}_0 (\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0), \tag{14}$$

where $\boldsymbol{G}^{d_1 \times 1} = \nabla \mathcal{L}_{\text{tgt}}(\boldsymbol{W}_0)$ is the gradient evaluated at the pre-trained parameters and H_0 denotes the Hessian computed at $\boldsymbol{W}_0$. The first-order optimality condition for the target optimum $\boldsymbol{W}_{\text{tgt}}$ is written as

$$\nabla_{\boldsymbol{W}} \mathcal{L}_{\text{tgt}}(\boldsymbol{W}_{\text{tgt}}) = 0 \approx \boldsymbol{G} + \boldsymbol{H}_0(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0). \tag{15}$$

Solving for the displacement gives $\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0 \approx -\boldsymbol{H}_0^{-1}\boldsymbol{G}$. In practice, the Hessian H_0 is typically approximated by the Fisher information matrix. This choice ensures numerical stability, theoretical consistency under maximum-likelihood estimation, and computational tractability since it can be estimated directly from gradients. Therefore,

$$\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0 \approx -\boldsymbol{J}(\boldsymbol{W}_0)^{-1}\boldsymbol{G}. \tag{16}$$

After obtaining an estimator of $\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0$ and computing $\boldsymbol{A}_0$ via Theorem 1, combining with equation 41, the initialization for $\boldsymbol{B}$ is

$$\boldsymbol{B}_0 = \boldsymbol{A}_0^\top(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0), \tag{17}$$

so that $\boldsymbol{W}_0 + \boldsymbol{A}_0\boldsymbol{B}_0$ coincides with the projection $\boldsymbol{W}_{\text{tgt}}^{\text{proj}}$.

4.2 STANDARD LORA CASE

We next consider the standard LoRA case where $\boldsymbol{W} \in \mathbb{R}^{d_1 \times d_2}$. In this setting, the estimator is given by $\hat{\boldsymbol{W}} = \boldsymbol{W}_0 + \boldsymbol{A}\,\boldsymbol{B}$, where $\boldsymbol{A} \in \mathbb{R}^{d_1 \times r}$ denotes a fixed column-orthogonal matrix, and $\boldsymbol{B} \in \mathbb{R}^{r \times d_2}$ is a trainable matrix.

Theorem 2. *(proved in Appendix C.2) In the standard LoRA case, the optimal initialization of the matrix A is given by*

$$\boldsymbol{A}_{\boldsymbol{0}}^{*} = \arg\min_{\boldsymbol{A}} \operatorname{tr}\left(\boldsymbol{A}^{\top} \boldsymbol{\Omega} \boldsymbol{A}\right), \tag{18}$$

*and from Section 2.3 we know that the r column vectors of $\boldsymbol{A}$ correspond to the eigenvectors of $\boldsymbol{\Omega}$ associated with its r smallest eigenvalues. $\boldsymbol{\Omega}$ is the **Initialization Guidance Matrix** and given by*

$$\boldsymbol{\Omega}^{d_1 \times d_1} = \left(\sum_{i=1}^{d_2} \frac{\boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1}}{N} - \sum_{i=1}^{d_2} \left(\boldsymbol{W}_{tgt} - \boldsymbol{W}_0\right)_{(:,i)} \left(\boldsymbol{W}_{tgt} - \boldsymbol{W}_0\right)_{(:,i)}^{\top} \right), \tag{19}$$

where the subscript $(:, i)$ denotes the i-th column of a matrix. It is important to note that the columns of $\boldsymbol{W}$ are not independent, and therefore the Fisher information matrix should be defined with respect to the entire parameter matrix rather than computed separately for each column. Specifically, $\boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1}$ denotes the i-th $d_1 \times d_1$ diagonal block of the inverse Fisher matrix $\boldsymbol{J}(\operatorname{vec}(\boldsymbol{W}_0))^{-1}$, i.e.,

$$\boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1} \;\triangleq\; \boldsymbol{J}(\operatorname{vec}(\boldsymbol{W}_0))^{-1}_{id_1+1:(i+1)d_1,\; id_1+1:(i+1)d_1)}, \tag{20}$$

where $\operatorname{vec}(\cdot)$ denotes the column-wise vectorization operator that flattens $\boldsymbol{W}$ into a vector.

Similar to equation 16, the approximation of $\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0$ is

$$\left(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0\right)_{(:,i)} \approx -\,\boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1} \boldsymbol{G}_{(:,i)}, \tag{21}$$

where $\boldsymbol{G}^{d_1 \times d_2} = \nabla \mathcal{L}_{\text{tgt}}(\boldsymbol{W}_0)$. Moreover, the initialization method is the same as equation 17.

Remark 3. *To relate our method to prior gradient-based fine-tuning and to justify the optimality of our initialization, we present the following observation. Both LoRA-GA and LoRA-One essentially perform a direct singular value decomposition (SVD) on the gradient. If we simplify our approach by (i) discarding the first term in the Initialization Guidance Matrix and (ii) directly using the negative gradient to replace the right side in Eq. equation 21, our theoretical result in Theorem 2 reduces to*

$$\boldsymbol{A}_{0}^{*} = \arg\min_{\boldsymbol{A}} -\operatorname{tr}\left(\boldsymbol{A}^{\top} \boldsymbol{G}\boldsymbol{G}^{\top} \boldsymbol{A}\right). \tag{22}$$

In other words, the initialization of $\boldsymbol{A}$ corresponds to the leading r eigenvectors of $GG^{\top}$, which are equivalent to the top r left singular vectors of $\boldsymbol{G}$. Thus, our method in this degenerate form coincides with the strategies adopted in LoRA-GA and LoRA-One. From this analysis, two advantages of our theoretical results become evident: (1) the first term of the Initialization Guidance Matrix explicitly models the estimation variance induced by the stochasticity of training samples, a factor overlooked in prior studies; and (2) the estimation in equation 21, compared with using raw gradients alone, incorporates the Fisher information matrix to account for the anisotropy of the model.

4.3 ALGORITHM

Grounded in the asymptotic results of Sections 4.1 and 4.2, we introduce LoRA-DA, which translates our theoretical insights into a practical LoRA initialization scheme. Specifically, among the N available target domain samples for subsequent PEFT, LoRA-DA requires only a small set of target samples $\mathcal{S}$ to estimate the necessary statistics for initialization, making it lightweight and data-aware. From these samples, we compute both the gradient and the Fisher matrix, where the Fisher information is computed using the **K-FAC** (Martens & Grosse, 2015), a scalable method that approximates the Fisher as a Kronecker product of smaller matrices formed by the layer input vectors and the backpropagated gradients. Moreover, for computing the eigenvalues and eigenvectors required in the initialization of $\boldsymbol{A}_0$, we employ the **LOBPCG** algorithm (Knyazev, 2001). The detailed procedure of LoRA-DA is summarized in Algorithm 1. Finally, Appendix D shows that our algorithm introduces no significant memory overhead compared with gradient-based methods.

Algorithm 1 LoRA-DA for one specific layer

Input: Pre-trained weight $\boldsymbol{W}_0$, total data size N, sampled data $\mathcal{S} = \{(z^j, y^j)\}_{j=1}^{|\mathcal{S}|}$, LoRA rank r, loss function ℓ // sampled data size $|\mathcal{S}| \ll N$

Initialize:

$\boldsymbol{G}^{d_1 \times d_2} = \frac{1}{|\mathcal{S}|} \sum_{(z^j, y^j) \in \mathcal{S}} \nabla_{\boldsymbol{W}} \ell(\boldsymbol{W}_0; z^j, y^j)$

$\boldsymbol{Z}_{\text{fisher}}^{d_1 \times d_1} = \frac{1}{|\mathcal{S}|} \sum_{(z^j, y^j) \in \mathcal{S}} z^j z^{j\top}$

$\boldsymbol{Y}_{\text{fisher}}^{d_2 \times d_2} = \frac{1}{|\mathcal{S}|} \sum_{(z^j, y^j) \in \mathcal{S}} \nabla_y \ell(\boldsymbol{W}_0; z^j, y^j) \nabla_y \ell(\boldsymbol{W}_0; z^j, y^j)^\top$

for $i = 1, \ldots, d_2$ **do**

  $\boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1} = \boldsymbol{Z}_{\text{fisher}}^{-1} \times [\boldsymbol{Y}_{\text{fisher}}^{-1}]_{(i,i)}$ // $[\boldsymbol{Y}_{\text{fisher}}^{-1}]_{(i,i)}$ is the (i,i) entry of $\boldsymbol{Y}_{\text{fisher}}^{-1}$.

  $(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)_{(:,i)} = \boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1} G_{(:,i)}$

end for

$\boldsymbol{A}_0 \leftarrow \arg\min_{\boldsymbol{A}} \operatorname{tr}\left(\boldsymbol{A}^\top \left(\sum_{i=1}^{d_2} \frac{\boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1}}{N} - \sum_{i=1}^{d_2} (\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)_{(:,i)} (\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)_{(:,i)}^\top\right) \boldsymbol{A}\right)$

$\boldsymbol{B}_0 \leftarrow \boldsymbol{A}_0^\top (\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)$

Return: $\boldsymbol{A}_0, \boldsymbol{B}_0$

Table 1: Commonsense evaluation results.

Model	PEFT	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
ChatGPT	–	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA 2-7B	LoRA	73.2	85.8	81.8	94.9	86.0	74.7	88.7	86.0	83.9
	PiSSA	72.5	85.2	81.9	94.2	86.7	73.7	87.1	**87.0**	83.5
	MiLoRA	73.1	85.3	81.8	95.1	86.3	75.3	88.6	86.8	84.0
	LoRA-One	72.8	85.3	81.9	**95.2**	85.6	74.9	**88.8**	86.4	83.9
	LoRA-DA	**73.2**	**86.0**	**82.4**	**95.2**	**87.1**	**75.7**	88.7	86.4	**84.3**

5 Experiments

In this section, we conduct experiments to evaluate LoRA-DA against representative initialization strategies for LoRA across several NLP benchmarks. All experiments are performed on eight NVIDIA A800 GPUs, unless stated otherwise. The small set of target-domain samples used to estimate the necessary statistics is by default set to 256 samples. Our baselines consist of vanilla LoRA (Hu et al., 2022) and its initialization variants with original structure: the data-agnostic methods PiSSA (Meng et al., 2024) and MiLoRA (Wang et al., 2025), and the gradient-based method LoRA-One (Zhang et al., 2025b). We do not include other PEFT methods that modify the original LoRA structure, despite their initialization modules, in order to maintain fairness in comparison.

5.1 Natural Language Understanding (NLU) Tasks

We adopt *commonsense reasoning* as a representative natural language understanding task. Specifically, we fine-tune **LLaMA 2–7B** (Touvron et al., 2023) on all samples from **Commonsense170K** (Hu et al., 2023), and evaluate on eight widely used benchmarks: **BoolQ** (Clark et al., 2019), **PIQA** (Bisk et al., 2020), **SIQA** (Sap et al., 2019), **HellaSwag** (Zellers et al., 2019), **WinoGrande** (Sakaguchi et al., 2021), **ARC-e** and **ARC-c** (Clark et al., 2018), and **OBQA** (Mihaylov et al., 2018). The task is formulated as a *multiple-choice problem*, and we report **accuracy (%)** on all test sets using the last checkpoint.

As shown in Table 1, LoRA-DA outperforms existing LoRA initialization methods in terms of average performance and across most benchmarks. On average, LoRA-DA attains 84.3%, exceeding the prior state-of-the-art MiLoRA (84.0%) with a margin of 0.3 percentage points. In particular, LoRA-DA achieves top performance on six out of eight benchmarks, demonstrating its robustness and consistency across diverse reasoning tasks.

Table 2: Math evaluation results.

Method	GSM8K	MATH	Avg.
LoRA	53.1	8.3	30.7
PiSSA	53.2	8.2	30.7
MiLoRA	52.9	8.3	30.6
LoRA-One	53.8	8.5	31.1
LoRA-DA	**55.0**	**9.2**	**32.1**

Table 3: Math evaluation with frozen-$\boldsymbol{A}$ setting.

Method	GSM8K
LoRA-FA	41.5
LoRA-DA-FA	**49.4**

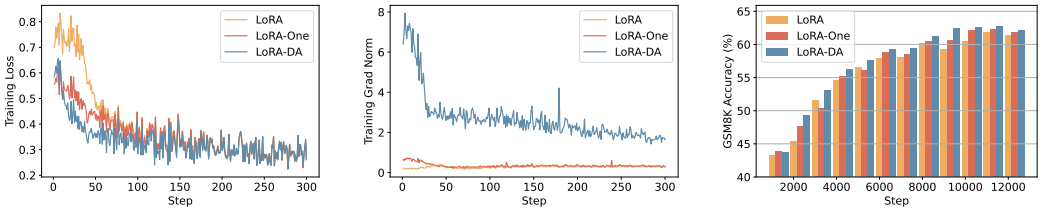
5.2 Natural Language Generation

We use *mathematical reasoning* as a representative natural language generation task, which requires models to generate a complete reasoning process and a final answer. Concretely, we fine-tune **LLaMA 2–7B** (Touvron et al., 2023) on 100K samples from **MetaMathQA** (Yu et al., 2023), and evaluate on the official test sets of **GSM8K** (Cobbe et al., 2021) and **MATH** (Hendrycks et al., 2021). Unless otherwise specified, results are reported from the last checkpoint, and performance is measured by the **Exact Match (EM)** ratio between predictions and ground-truth answers.

As shown in Table 2, LoRA-DA achieves clear improvements over existing initialization strategies on mathematical reasoning tasks. It obtains the highest accuracy on both GSM8K and MATH, reaching 55.0% and 9.2%, respectively. Compared to the strongest baseline LoRA-One, LoRA-DA improves the average accuracy by 1.0 percentage points. These results demonstrate that our theoretically grounded initialization is not only effective for natural language understanding but also consistently advantageous for natural language generation tasks. Across Table 1 and Table 2, methods that incorporate a small set of target-domain data, namely LoRA-One and LoRA-DA, consistently exhibit more competitive performance than data-agnostic approaches, underscoring the importance of leveraging target data in LoRA initialization. We also conduct a comparative experiment under the frozen-$\boldsymbol{A}$ setting of LoRA-FA, as shown in Table 3. The results demonstrate that our initialization theory remains effective even when A is frozen.

5.3 Performance Over Training Steps

We further analyze the optimization dynamics by tracking loss, gradient norm, and evaluation accuracy throughout training on GSM8K in Figure 2. Compared with vanilla LoRA, LoRA-One achieves faster reduction of loss and higher accuracy in the earliest steps, since its gradient-only initialization closely aligns with the steepest descent direction. Interestingly, LoRA-DA starts slightly behind LoRA-One at the beginning. This behavior arises because it better accounts for sample stochasticity and parameter-space anisotropy. Its early steps appear more conservative—not because the method fails to find good directions, but because it prioritizes stability over immediate descent, leading to more reliable convergence in later stages. As training proceeds, LoRA-DA exhibits more stable gradient norms, faster overall convergence, and higher final accuracy. These results confirm that although gradient-only methods may appear favorable in the short term, variance-aware initialization ultimately provides more robust optimization and superior long-run performance. We provide another visualization of the descent trajectory of LoRA-DA in the Appendix E.



(a) Loss over steps. (b) Gradient norm over steps. (c) Accuracy over steps.

Figure 2: The loss, grad norm, and evaluation accuracy on GSM8K over the training steps of LoRA (indicated in yellow), LoRA-One (in red), and LoRA-DA (in blue)

5.4 Experiments on Various Ranks

We further evaluate LoRA-DA under different ranks $r \in \{1, 2, 4, 8, 16\}$ on GSM8K, and compare it with LoRA and LoRA-One in Figure 4. Across all settings, LoRA-DA consistently outperforms both baselines, confirming the robustness of our initialization strategy. Notably, the advantage of LoRA-DA is more pronounced at smaller ranks. This can be explained by the fact that when the rank is severely limited, the choice of descent directions becomes particularly critical: a suboptimal initialization may constrain the model to an ineffective subspace that cannot be recovered during training. By contrast, LoRA-DA leverages Fisher-gradient and variance term to identify more informative low-rank directions, which substantially improves optimization in the low-rank regime. As the rank increases, the subspace coverage grows and the relative gap between methods narrows, yet LoRA-DA still achieves the best performance across all ranks.

5.5 Initialization Overhead

Table 5 summarizes the initialization time, training time, and total time of LoRA-DA under different ranks, evaluated on the GSM8K task. The results show that the initialization phase remains stable across all ranks, while the training phase increases slightly with larger ranks, leading to a modest growth in total time. Notably, initialization accounts for only about 6% of the overall time, indicating that the main computational cost lies in the training phase. This demonstrates the good scalability of LoRA-DA under different rank settings.

5.6 Ablation Study

Table 6 reports the ablation study on GSM8K, a standard benchmark for mathematical reasoning. Our method consists of two components: bias and variance. Removing the bias term (LoRA-DA w/o bias) or the variance term (LoRA-DA w/o var) both lead to slight drops in performance. Furthermore, LoRA-DA w/o var&fisher, which estimates the bias term using plain gradients instead of Fisher-gradient, performs slightly worse than LoRA-DA w/o var. The full method (LoRA-DA) achieves the best results, showing that both components are essential for optimal performance.

6 Related Work

PEFT methods fall into three categories: *adapter-based* (Houlsby et al., 2019; He et al., 2021); *prompt-based* (Lester et al., 2021; Razdaibiedina et al., 2023); and *LoRA-based* (Hu et al., 2022)s. LoRA variants such as AdaLoRA (Zhang et al., 2023b), DoRA (Liu et al., 2024), and VeRA (Kopiczko et al., 2024) improve efficiency or stability, but most rely on random initialization, leading to slow warm-up and possible suboptimal convergence. Principled initialization has thus been explored. PiSSA (Meng et al., 2024) and MiLoRA (Wang et al., 2025) are data-agnostic, based on singular components of pre-trained weights, whereas data-aware methods leverage target samples, typically by using early gradients as in LoRA-GA (Wang et al., 2024a) and LoRA-One (Zhang et al., 2025b). However, their single-step reliance limits effectiveness. Beyond initialization, LoRA-Pro (Wang et al., 2024b) re-weights low-rank gradients during fine-tuning to better approximate full-model updates. We provide an extended version of the related work in the Appendix B.

7 Conclusion

We theoretically derive a data-aware initialization method for LoRA via asymptotic analysis. The method estimates a variance term from Fisher information and an anisotropy-aware bias term from the Fisher-gradient, and combines them to construct the initialization subspace. Building on this theory, we propose LoRA-DA, which uses a small set of target domain samples to compute LoRA initialization. On NLU and NLG benchmarks, LoRA-DA improves final accuracy over prior LoRA initializations and remains effective under the frozen-$\boldsymbol{A}$ setting. Supplementary studies show faster and more stable convergence, robustness across ranks, and only a small initialization overhead.

ETHICS STATEMENT

This work is a methodological contribution that develops a theoretically grounded initialization strategy for LoRA. All experiments are conducted on publicly available benchmark datasets, which are widely used in the research community. No private or personally identifiable information is involved. Our method does not raise new ethical or societal risks beyond those already associated with large language models.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. A detailed description of the proposed algorithm is provided in Algorithm 1, and all theoretical results are accompanied by complete proofs in the appendix. All datasets used are publicly available. We provide the hyperparameter settings in the Appendix I. The source code will be released upon publication to facilitate reproducibility and allow further verification of our results.

REFERENCES

John Aitchison and SD Silvey. Maximum-likelihood estimation of parameters subject to restraints. *The annals of mathematical Statistics*, pp. 813–828, 1958.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7870–7881, 2020.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 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)*, pp. 2924–2936, 2019.

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

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

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Charles J Geyer. On the asymptotics of constrained m-estimation. *The Annals of statistics*, pp. 1993–2010, 1994.

Nathan Godey, Éric de la Clergerie, and Benoît Sagot. Anisotropy is inherent to self-attention in transformers. *arXiv preprint arXiv:2401.12143*, 2024.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. The impact of initialization on lora finetuning dynamics. *Advances in Neural Information Processing Systems*, 37:117015–117040, 2024.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in neural information processing systems*, 34:1022–1035, 2021.

Andrew V Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 23(2):517–541, 2001.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*, 2024.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. In *International Conference on Learning Representations (ICLR)*. International Conference on Learning Representations, 2020.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

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

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

Anastasiia Razdaibiedina, Yuning Mao, Madian Khabsa, Mike Lewis, Rui Hou, Jimmy Ba, and Amjad Almahairi. Residual prompt tuning: improving prompt tuning with residual reparameterization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6740–6757, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

Armin Toroghi, Ali Pesaranghader, Tanmana Sadhu, and Scott Sanner. LLM-based typed hyperresolution for commonsense reasoning with knowledge bases. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wNobG8bV5Q.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 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)*, pp. 4823–4836, 2025.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931, 2024a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, and Ed Chi. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023a. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yaqing Wang, Jialin Wu, Tanmaya Dabral, Jiageng Zhang, Geoff Brown, Chun-Ta Lu, Frederick Liu, Yi Liang, Bo Pang, Michael Bendersky, et al. Non-intrusive adaptation: Input-centric parameter-efficient fine-tuning for versatile multimodal modeling. *arXiv preprint arXiv:2310.12100*, 2023b.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *EMNLP 2021-2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 8696–8708. Association for Computational Linguistics (ACL), 2021.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized? In *The Thirteenth International Conference on Learning Representations*, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023a.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *11th International Conference on Learning Representations, ICLR 2023*, 2023b.

Qingyue Zhang, Haohao Fu, Guanbo Huang, Yaoyuan Liang, Chang Chu, Tianren Peng, Yanru Wu, Qi Li, Yang Li, and Shao-Lun Huang. A high-dimensional statistical method for optimizing transfer quantities in multi-source transfer learning. *NeurIPS 2025*, 2025a. URL https://arxiv.org/abs/2502.04242.

Yuanhe Zhang, Fanghui Liu, and Yudong Chen. Lora-one: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *Forty-second International Conference on Machine Learning*, 2025b.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez De Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 62369–62385, 2024.

APPENDIX

CONTENTS

A Notations **15**

B Extended Related Work **16**

B.1 Parameter-Efficient Fine-Tuning (PEFT) 16

B.2 Low-Rank Adaptation (LoRA) and its Variants and Initialization 16

C Proofs **17**

C.1 Proof of Theorem 1 17

C.2 Proof of Theorem 2 19

D Discussion on Space Complexity of the Algorithm **20**

E The Visualization of the Training **20**

F Accuracy Across Different Ranks **21**

G Time Consumption Across Different Ranks **21**

H Ablation Experiments **21**

I Hyperparameter Settings **21**

J LLM Usage **22**

A NOTATIONS

Table 4: Notation used in this paper.

Symbol	Meaning	Shape / Type
$P_{X,\boldsymbol{W}_0}$	Pre-trained model (data distribution)	
$P_{X,\boldsymbol{W}_{\text{tgt}}}$	Target model (fine-tuning distribution)	
X	Joint variable/distribution of inputs and outputs	
Z	Model inputs/features	
Y	Model outputs/labels	
$\boldsymbol{W}_0$	Pre-trained parameter matrix	$\mathbb{R}^{d_1\times d_2}$
$\boldsymbol{W}_{\text{tgt}}$	Target task true parameter matrix	$\mathbb{R}^{d_1\times d_2}$
$\hat{\boldsymbol{W}}$	Estimated parameter after fine-tuning	$\mathbb{R}^{d_1\times d_2}$
$\Delta\boldsymbol{W}$	Parameter update (LoRA low-rank)	$\mathbb{R}^{d_1\times d_2}$
$\boldsymbol{A}$	LoRA left matrix	$\mathbb{R}^{d_1\times r}$
$\boldsymbol{B}$	LoRA right matrix	$\mathbb{R}^{r\times d_2}$
r	LoRA rank	$r \ll \min(d_1, d_2)$
d_1, d_2	Row/column dimensions of parameter matrix	
$\mathcal{L}_{\text{tgt}}(\boldsymbol{W})$	Empirical loss on target domain	
$\nabla\mathcal{L}_{\text{tgt}}(\boldsymbol{W}_0)$	Gradient at $\boldsymbol{W}_0$	$\mathbb{R}^{d_1\times d_2}$
$\boldsymbol{G}$	Gradient at $\boldsymbol{W}_0$	$\mathbb{R}^{d_1\times d_2}$
$\boldsymbol{H}_0$	Hessian at $\boldsymbol{W}_0$	$\mathbb{R}^{(d_1d_2)\times(d_1d_2)}$
$\boldsymbol{J}(\cdot)$	Fisher information matrix	
$\boldsymbol{J}(\cdot)^{-1}$	Inverse Fisher information matrix	
$\boldsymbol{J}(\boldsymbol{W}_0)^{-1}_{[i]}$	the i-th $d_1 \times d_1$ diagonal block of $\boldsymbol{J}(\text{vec}(\boldsymbol{W}_0))^{-1}$	$\mathbb{R}^{d_1\times d_1}$
$\text{vec}(\cdot)$	the column-wise vectorization operator that flattens matrix into a vector	
N	Number of target-domain samples	$\mathbb{N}$
$\boldsymbol{W}_{\text{tgt}}^{\text{proj}}$	Projection of $\boldsymbol{W}_{\text{tgt}}$ onto $W_0 + \{AB\}$	$\mathbb{R}^{d_1\times d_2}$
$\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^\top$	Orthogonal projector onto subspace spanned by A	$\mathbb{R}^{d_1\times d_1}$
$\boldsymbol{Z}_{\text{fisher}}$	K-FAC left factor (input second moments)	$\mathbb{R}^{d_1\times d_1}$
$\boldsymbol{Y}_{\text{fisher}}$	K-FAC right factor (output-gradient second moments)	$\mathbb{R}^{d_2\times d_2}$
$\boldsymbol{A}_0$	Initialized A	$\mathbb{R}^{d_1\times r}$
$\boldsymbol{B}_0$	Initialized B	$\mathbb{R}^{r\times d_2}$
$\mathcal{S}$	Small target-domain sample set	
$\boldsymbol{\Omega}$	Initialization Guidance Matrix	$\mathbb{R}^{d_1\times d_1}$
$\boldsymbol{U}, \boldsymbol{\Lambda}$	Eigenvectors/eigenvalues (EVD)	
$\boldsymbol{u}_i$	i-th eigenvector	
λ_i	i-th eigenvalue (ascending)	
$(:, i)$	The i-th column of a matrix (colon indexing)	
(i, i)	The (i, i)-th (diagonal) entry	
$\|\cdot\|_F$	Frobenius norm	
$\text{tr}(\cdot)$	Trace operator	

B Extended Related Work

B.1 Parameter-Efficient Fine-Tuning (PEFT)

PEFT aims to reduce the computational and memory overhead of adapting large pre-trained models, while still achieving performance comparable to full model fine-tuning. Existing PEFT techniques can be broadly divided into three categories, namely *adapter-based*, *prompt-based*, and *LoRA-based* methods. **Adapter-based methods** (Houlsby et al., 2019; He et al., 2021; Karimi Mahabadi et al., 2021)insert small trainable modules into frozen layers of the backbone network, enabling efficient task adaptation without updating the full set of parameters. **Prompt-based methods** (Lester et al., 2021; Razdaibiedina et al., 2023; Wang et al., 2023b) optimize additional continuous tokens (soft prompts) or prefix vectors that steer the behavior of the pre-trained model. These approaches enable highly lightweight adaptation with minimal trainable parameters. However, their performance is often sensitive to initialization and may vary considerably across different tasks. The above two categories, by altering either the architecture or the input representation, tend to introduce extra inference overhead compared to the backbone model.

B.2 Low-Rank Adaptation (LoRA) and its Variants and Initialization

The third category, **LoRA-based methods**, performs parameter-efficient fine-tuning by constraining weight updates to a low-rank decomposition. Instead of directly updating the full parameter matrix, LoRA reparameterizes the update as the product of two low-rank matrices, which drastically reduces the number of trainable parameters while enabling the merged weights to be seamlessly integrated into the backbone at inference time without additional latency (Hu et al., 2022). For instance, LoRA-FA (Zhang et al., 2023a) introduces parameter freezing to reduce memory usage during adaptation; AdaLoRA (Zhang et al., 2023b) dynamically allocates rank across layers according to their importance, improving overall parameter efficiency; DoRA (Liu et al., 2024) decomposes pre-trained weights into magnitude and direction, applying low-rank updates only to the directional component for greater stability and expressiveness; VeRA (Kopiczko et al., 2024) re-parameterizes LoRA by introducing shared low-rank vectors across layers, thereby reducing the number of trainable parameters while maintaining competitive performance. However, LoRA and many of its variants typically adopt random initialization for the low-rank matrix A and zero initialization for B. Such uninformed initialization introduces two main drawbacks. First, the training process tends to progress slowly in the early stages, as the update directions are not aligned with informative subspaces. Second, the optimization may converge to suboptimal solutions, since the imposed low-rank structure restricts the search space to poorly initialized directions.

To overcome these limitations, several works propose more principled initialization strategies. Some work investigates LoRA's two zero-start initialization schemes—initializing $\boldsymbol{A}$ with $\boldsymbol{B} = \mathbf{0}$ versus initializing $\boldsymbol{B}$ with $\boldsymbol{A} = \mathbf{0}$—and demonstrates that $\boldsymbol{A}$-initialization is more effective (Hayou et al., 2024). PiSSA (Meng et al., 2024) initializes the low-rank matrices A and B using the principal singular components of the pre-trained weights, while the remaining components are used to initialize the residual weights W. In contrast, MiLoRA (Wang et al., 2025) leverages minor singular components for initializing A and B, thereby preserving dominant directions of the pre-trained model and exploiting underutilized subspaces for adaptation. Early approaches of this kind are generally *data-agnostic*, as they rely solely on the weight structure. More recently, *data-aware* methods have been explored, which explicitly incorporate information from the target task. For example, LoRA-GA (Wang et al., 2024a) employs gradient alignment to initialize the low-rank subspace, and LoRA-One (Zhang et al., 2025b) derives its initialization by decomposing the fine-tuning gradient at the first optimization step. However, most existing data-aware approaches rely on such a direct decomposition of the first-step gradient. Since model performance after a single update step is typically unsatisfactory, initialization strategies grounded solely in this early gradient can be limited in effectiveness. In addition, several variants leverage gradients to assist training but are not directly related to initialization. For instance, LoRA-Pro (Wang et al., 2024b) improves LoRA by strategically re-weighting the gradients of the low-rank matrices during fine-tuning, such that their product produces a low-rank update that more faithfully approximates the full-model gradient, thereby narrowing the performance gap to full fine-tuning.

C PROOFS

C.1 PROOF OF THEOREM 1

We assume that the orthogonal projection of $\boldsymbol{W}_{\text{tgt}}$ onto the LoRA subspace is denoted by $\boldsymbol{W}_{\text{tgt}}^{\text{proj}}$, as shown in Figure 1. Under this assumption, the training objective can be decomposed as

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{W}} - \boldsymbol{W}_{\text{tgt}}\right\|_F^2\right] = \mathbb{E}\left[\left\|\hat{\boldsymbol{W}} - \boldsymbol{W}_{\text{tgt}}^{\text{proj}}\right\|_F^2\right] + \left\|\boldsymbol{W}_{\text{tgt}}^{\text{proj}} - \boldsymbol{W}_{\text{tgt}}\right\|_F^2, \tag{23}$$

where the first term represents the expected estimation variance within the subspace, and the second term corresponds to the bias arising from the distance between $\boldsymbol{W}_{\text{tgt}}$ and the LoRA subspace.

For the first term of equation 23, by Lemma 3.3, we know that if $\hat{\boldsymbol{W}}$ is not restricted to the LoRA form, the training variance is given by $\frac{\text{tr}\left(\boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})^{-1}\right)}{N}$. In the presence of the LoRA structure with a fixed orthonormal basis $\boldsymbol{A}$, this variance is no longer full, but have a upper bound which is the projection of the unconstrained variance onto the subspace spanned by $\boldsymbol{A}$.

Lemma 4. *Let $A \in \mathbb{R}^{d_1 \times r}$ be fixed with orthonormal columns and let N i.i.d. target samples be given, with Fisher information $J(W_{\text{tgt}})$. Then the constrained estimator $\hat{\boldsymbol{W}}$ within $\boldsymbol{W}_0 + \boldsymbol{AB}$ satisfies*

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{W}} - \boldsymbol{W}_{tgt}^{proj}\right\|_F^2\right] \leq \frac{\text{tr}\left(\boldsymbol{A}^T J(\boldsymbol{W}_{tgt})^{-1}\boldsymbol{A}\right)}{N} + o(\tfrac{1}{N}). \tag{24}$$

Proof. According to prior analyses of constrained maximum likelihood estimation (Aitchison & Silvey, 1958; Geyer, 1994), we have

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{W}} - \boldsymbol{W}_{\text{tgt}}^{\text{proj}}\right\|_F^2\right] = \frac{\text{tr}\left((\boldsymbol{A}^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})\boldsymbol{A})^{-1}\right)}{N} + o(\tfrac{1}{N}). \tag{25}$$

However, this form is less convenient for the subsequent analysis, and we therefore apply a relaxation. In the following, we will show that

$$\text{tr}\left((\boldsymbol{A}^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})\boldsymbol{A})^{-1}\right) \leq \text{tr}\left(\boldsymbol{A}^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})^{-1}\boldsymbol{A}\right). \tag{26}$$

We recall a variational identity: for any symmetric positive definite matrix $M \in \mathbb{R}^{d\times d}$ and vector $v \in \mathbb{R}^d$,

$$v^\top M^{-1} v = \max_{x\in\mathbb{R}^d} \left(2v^\top x - x^\top M x\right). \tag{27}$$

This follows from completing the square, with the maximum attained at $x^\star = M^{-1}v$.

Now set $M = \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})$ and $v = Ay$ for arbitrary $y \in \mathbb{R}^r$. By equation 27, we have

$$y^\top A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})^{-1} A y = \max_{x\in\mathbb{R}^d} \left(2y^\top A^\top x - x^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})x\right). \tag{28}$$

Restricting the maximization in equation 28 to the subspace spanned by the columns of $\boldsymbol{A}$ cannot increase the maximum. By writing $x = Au$ with $u \in \mathbb{R}^r$, we obtain

$$y^\top A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})^{-1} A y \geq \max_{u\in\mathbb{R}^r} \left(2y^\top u - u^\top (A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})A)u\right). \tag{29}$$

Applying equation 27 again with $M = A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})A$ and $v = y$, it follows that

$$y^\top (A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})A)^{-1} y = \max_{x\in\mathbb{R}^r} \left(2y^\top x - x^\top (A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})A)x\right). \tag{30}$$

The right-hand side of equation 29 is equivalent to the right-hand side of equation 30. Thus, for all $y \in \mathbb{R}^r$,

$$y^\top A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})^{-1} A y \geq y^\top (A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})A)^{-1} y, \tag{31}$$

which implies the Loewner order

$$(A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})A)^{-1} \preceq A^\top \boldsymbol{J}(\boldsymbol{W}_{\text{tgt}})^{-1} A. \tag{32}$$

Finally, since the trace is monotone with respect to the Loewner order on positive semidefinite matrices, we conclude

$$\mathrm{tr}\big((A^\top \boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})A)^{-1}\big) \le \mathrm{tr}\big(A^\top \boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})^{-1}A\big)\,. \tag{33}$$

Equality holds if and only if the unconstrained maximizer $x^\star = \boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})^{-1}Ay$ always lies in the subspace spanned by the columns of $\boldsymbol{A}$. This is equivalent to requiring that the subspace spanned by the columns of $\boldsymbol{A}$ be invariant under $\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})$, for instance, when the columns of A are eigenvectors of $\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})$, or when $\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})$ is a scalar multiple of the identity matrix.

Combining equation 33 and equation 25, we can prove equation 24.

□

Given that $\boldsymbol{W}_{\mathrm{tgt}}$, $\boldsymbol{W}_0$ are already assumed to be sufficiently close, We can use this, along with a Taylor expansion, to approximate the distance between their Fisher information matrices. We conclude the following lemma.

Lemma 5. *The discrepancy between $\boldsymbol{J}(\boldsymbol{W}_{tgt})^{-1}$ and $\boldsymbol{J}(\boldsymbol{W}_0)^{-1}$ is of the order $O\left(\frac{1}{\sqrt{N}}\right)$,i.e.,*

$$\boldsymbol{J}(\boldsymbol{W}_{tgt})^{-1} = \boldsymbol{J}(\boldsymbol{W}_0)^{-1} + O(\tfrac{1}{\sqrt{N}}). \tag{34}$$

Proof.

$$\begin{aligned}
\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}}) &= \mathbb{E}_{\boldsymbol{W}_{\mathrm{tgt}}}\left[\left(\frac{\partial}{\partial \boldsymbol{W}}\log P_{X;\boldsymbol{W}_{\mathrm{tgt}}}\right)^2\right] \\
&= \mathbb{E}_{\boldsymbol{W}_{\mathrm{tgt}}}\left[\left(\frac{\partial}{\partial \boldsymbol{W}}\log P_{X;\boldsymbol{W}_0} + \frac{\partial^2 \log P_{X;\boldsymbol{W}_0}}{\partial \boldsymbol{W}^2}(\boldsymbol{W}_{\mathrm{tgt}} - \boldsymbol{W}_0) + O(\tfrac{1}{N})\right)^2\right] \\
&= \mathbb{E}_{\boldsymbol{W}_{\mathrm{tgt}}}\left[\left(\frac{\partial}{\partial \boldsymbol{W}}\log P_{X;\boldsymbol{W}_0}\right)^2\right] + O(\tfrac{1}{\sqrt{N}}) \\
&= \sum_{x\in\mathcal{X}} P_{X;\boldsymbol{W}_{\mathrm{tgt}}}(x)\left(\frac{\partial}{\partial \boldsymbol{W}}\log P_{X;\boldsymbol{W}_0}(x)\right)^2 + O(\tfrac{1}{\sqrt{N}}) \\
&= \sum_{x\in\mathcal{X}} \left(P_{X;\boldsymbol{W}_0}(x) + \frac{\partial P_{X;\boldsymbol{W}_0}}{\partial \boldsymbol{W}}(\boldsymbol{W}_{\mathrm{tgt}} - \boldsymbol{W}_0) + O(\tfrac{1}{N})\right)\left(\frac{\partial}{\partial \boldsymbol{W}}\log P_{X;\boldsymbol{W}_0}(x)\right)^2 + O(\tfrac{1}{\sqrt{N}}) \\
&= \sum_{x\in\mathcal{X}} P_{X;\boldsymbol{W}_0}(x)\left(\frac{\partial}{\partial \boldsymbol{W}}\log P_{X;\boldsymbol{W}_0}(x)\right)^2 + O(\tfrac{1}{\sqrt{N}}) \\
&= \boldsymbol{J}(\boldsymbol{W}_0) + O(\tfrac{1}{\sqrt{N}})
\end{aligned} \tag{35}$$

Assume that $\boldsymbol{J}(\boldsymbol{W}_0)$ is invertible with bounded condition number, and we have know that

$$\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}}) = \boldsymbol{J}(\boldsymbol{W}_0) + O\big(\tfrac{1}{\sqrt{N}}\big). \tag{36}$$

By the resolvent identity (Horn & Johnson, 2012), we have

$$\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})^{-1} - \boldsymbol{J}(\boldsymbol{W}_0)^{-1} = \boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})^{-1}\big(\boldsymbol{J}(\boldsymbol{W}_0) - \boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})\big)\boldsymbol{J}(\boldsymbol{W}_0)^{-1}, \tag{37}$$

which can be easily proved by multiplying both sides of the equation on the left by $\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})$.

Taking norms on both sides yields

$$\big\|\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})^{-1} - \boldsymbol{J}(\boldsymbol{W}_0)^{-1}\big\| \le \|\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})^{-1}\|\|\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}}) - \boldsymbol{J}(\boldsymbol{W}_0)\|\|\boldsymbol{J}(\boldsymbol{W}_0)^{-1}\|. \tag{38}$$

We analyze the right-hand side of the inequality. Since the Fisher matrix and its inverse are both of constant order and $\|\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}}) - \boldsymbol{J}(\boldsymbol{W}_0)\| = O(\frac{1}{\sqrt{N}})$, the right-hand side of the inequality is of order $O\Big(\frac{1}{\sqrt{N}}\Big)$. Therefore, from equation 38 we can know that

$$\boldsymbol{J}(\boldsymbol{W}_{\mathrm{tgt}})^{-1} = \boldsymbol{J}(\boldsymbol{W}_0)^{-1} + O\big(\tfrac{1}{\sqrt{N}}\big). \tag{39}$$

□

Combining Lemma 5 and equation 24, we have

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{W}}-\boldsymbol{W}_{\text{tgt}}^{\text{proj}}\right\|_F^2\right] \leq \frac{\operatorname{tr}\left(\boldsymbol{A}^T J(\boldsymbol{W}_0)^{-1}\boldsymbol{A}\right)}{N} + o(\tfrac{1}{N}) \tag{40}$$

For the second term of equation 23, from Figure 1, the projection of the difference $\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0$ onto the subspace spanned by the columns of $\boldsymbol{A}$ is

$$(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)^{\text{proj}} = \boldsymbol{A}\boldsymbol{A}^\top(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0) = \boldsymbol{W}_{\text{tgt}}^{\text{proj}} - \boldsymbol{W}_0. \tag{41}$$

Therefore, by the Pythagorean theorem in Euclidean space, we have

$$\left\|\boldsymbol{W}_{\text{tgt}}^{\text{proj}} - \boldsymbol{W}_{\text{tgt}}\right\|_F^2 = \|\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0\|_F^2 - \left\|\boldsymbol{A}^\top(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)\right\|_F^2. \tag{42}$$

Combining equation 23 and equation 40 and equation 42, we have,

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{W}}-\boldsymbol{W}_{\text{tgt}}\right\|_F^2\right] \leq & \|\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0\|_F^2 \\ & + \operatorname{tr}\left(\boldsymbol{A}^\top\left(\frac{\boldsymbol{J}(\boldsymbol{W}_0)^{-1}}{N} - (\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)^\top\right)\boldsymbol{A}\right) + o(\tfrac{1}{N})\end{aligned} \tag{43}$$

C.2 Proof of Theorem 2

Similar to equation 23, the training objective can be decomposed as

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{W}}-\boldsymbol{W}_{\text{tgt}}\right\|_F^2\right] = \mathbb{E}\left[\left\|\hat{\boldsymbol{W}}-\boldsymbol{W}_{\text{tgt}}^{\text{proj}}\right\|_F^2\right] + \left\|\boldsymbol{W}_{\text{tgt}}^{\text{proj}} - \boldsymbol{W}_{\text{tgt}}\right\|_F^2, \tag{44}$$

where the first term represents the expected estimation variance within the subspace, and the second term corresponds to the bias arising from the distance between $\boldsymbol{W}_{\text{tgt}}$ and the LoRA subspace.

For the first term of equation 44, similar to equation 40, we know that

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{W}}-\boldsymbol{W}_{\text{tgt}}^{\text{proj}}\right\|_F^2\right] &= \sum_{i=1}^{d_2}\mathbb{E}\left[\left\|(\hat{\boldsymbol{W}}-\boldsymbol{W}_{\text{tgt}}^{\text{proj}})_{(:,i)}\right\|_F^2\right] \\ &\leq \sum_{i=1}^{d_2}\operatorname{tr}\left(\frac{\boldsymbol{A}^T\boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1}\boldsymbol{A}}{N}\right) + o(\tfrac{1}{N}),\end{aligned} \tag{45}$$

where $\boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1}$ is denoted in Theorem 2.

For the second term of equation 44, similar to equation 42, we know that

$$\begin{aligned}\left\|\boldsymbol{W}_{\text{tgt}}^{\text{proj}} - \boldsymbol{W}_{\text{tgt}}\right\|_F^2 &= \sum_{i=1}^{d_2}\left[\left\|(\boldsymbol{W}_{\text{tgt}}^{\text{proj}} - \boldsymbol{W}_{\text{tgt}})_{(:,i)}\right\|_F^2\right] \\ &= \sum_{i=1}^{d_2}\left\|(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)_{(:,i)}\right\|_F^2 - \left\|\boldsymbol{A}^\top(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)_{(:,i)}\right\|_F^2\end{aligned} \tag{46}$$

Combining equation 44, equation 45 and equation 46, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{W}}-\boldsymbol{W}_{\text{tgt}}\right\|_F^2\right] \leq & \sum_{i=1}^{d_2}\left\|(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)_{(:,i)}\right\|_F^2 \\ & + \operatorname{tr}\left(\boldsymbol{A}^\top\sum_{i=1}^{d_2}\left(\frac{\boldsymbol{J}(\boldsymbol{W}_0)_{[i]}^{-1}}{N} - (\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)_{(:,i)}(\boldsymbol{W}_{\text{tgt}} - \boldsymbol{W}_0)_{(:,i)}^\top\right)\boldsymbol{A}\right) + o(\tfrac{1}{N})\end{aligned} \tag{47}$$

D Discussion on Space Complexity of the Algorithm

In our Algorithm 1, during the computation of statistics in Lines 1–3, we need to maintain $\boldsymbol{G} \in \mathbb{R}^{d_1 \times d_2}$, $\boldsymbol{Y} \in \mathbb{R}^{d_2 \times d_2}$, and $\boldsymbol{Z} \in \mathbb{R}^{d_1 \times d_1}$. In fact, only the diagonal entries of $\boldsymbol{Y}$ are required, which reduces its memory footprint to $\mathcal{O}(d_2)$. Moreover, $\boldsymbol{G}$ and the pair of $\boldsymbol{Z}, \boldsymbol{Y}$ can be computed in two separate passes rather than stored simultaneously. Excluding the storage of the base weight $\boldsymbol{W}_0$, which already requires $\mathcal{O}(d_1 d_2)$, the additional peak memory of our method is bounded by $\mathcal{O}(\max\{d_1^2, d_1 d_2\})$. In practical architectures such as LLaMA, LoRA is typically applied to attention and feed-forward projection layers where d_1 and d_2 are of comparable scale, so that d_1^2 and $d_1 d_2$ are of the same order. Hence, the peak memory usage of our algorithm is comparable to the gradient computation in LoRA-One which need a $\mathcal{O}(d_1 d_2)$ additional space to storage gradient. After these computations, all intermediate quantities can be offloaded from memory. At Line 7, we only need to maintain a $d_1 \times d_1$ coefficient matrix for the quadratic optimization, and the required statistics can be loaded in blocks from external storage. Furthermore, since we do not process all layers simultaneously, but instead compute the optimal LoRA initialization layer by layer in sequence, the overall peak memory usage remains bounded by the same order. **Therefore, our method achieves data-aware initialization without incurring higher space complexity than existing gradient-based approaches.**

E The Visualization of the Training

Here we present a visualization of the training process. Unlike fine-tuning and vanilla LoRA, which simply follow the raw gradient direction, our Fisher–gradient formulation incorporates parameter-space anisotropy, and the asymptotic normality analysis further models the variance arising from sampling stochasticity. As a result, the descent trajectory of LoRA-DA does not coincide with that of full fine-tuning at the beginning. Its steps appear more conservative—not due to failing to locate promising directions, but due to prioritizing stability over immediate descent—which ultimately guides the optimizer toward a more efficient convergence path.

We follow the experimental setup in Meng et al. (2024). Pre-training is conducted on 10,000 odd-numbered images from the MNIST dataset, followed by fine-tuning on 1,000 even-numbered images. The LoRA rank is set to 4, and the learning rate is set to 5×10^{-4}.

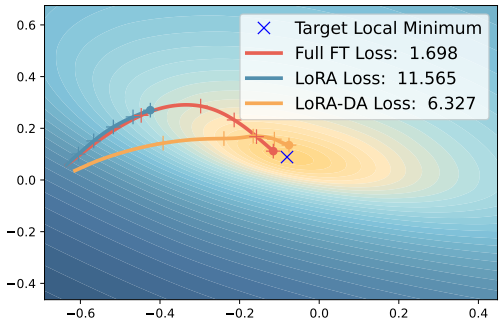


Figure 3: Loss landscape.

F Accuracy Across Different Ranks

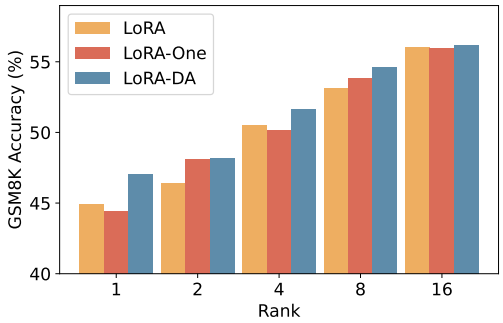


Figure 4: Accuracy of LoRA-DA across different ranks on the GSM8K task.

G Time Consumption Across Different Ranks

Table 5: Time consumption of LoRA-DA under different ranks on the GSM8K task, including initialization, training, and total time.

Rank	1	2	4	8	16
Initialization Time	2:03	2:04	2:04	2:05	2:06
Training Time	30:49	31:42	31:56	32:08	32:16
Total Time	32:52	33:46	34:00	34:13	34:22

H Ablation Experiments

Table 6: Ablation study of LoRA-DA on the GSM8K benchmark, evaluating the contributions of the bias and variance components.

Method	GSM8K
LoRA-DA w/o bias	53.6
LoRA-DA w/o var	53.3
LoRA-DA w/o var&fisher	53.0
LoRA-DA	**55.0**

I Hyperparameter Settings

To ensure a fair comparison among different low-rank adaptation methods, we used a unified hyperparameter configuration for LoRA-DA, LoRA, PiSSA, MiLoRA, and LoRA-One. This configuration was applied consistently to both NLG and NLU tasks, and the detailed hyperparameter settings are summarized in Table 7. We evaluated the models using the LLM-adapters framework (Hu et al., 2023). Notably, for LoRA-DA and LoRA-One, we pre-sampled 256 examples to estimate both the gradient and the Fisher information matrix during initialization.

Table 7: Hyperparameter settings for LoRA-DA and baseline methods.

Hyperparameter	**Value**
LoRA Rank (r)	8
LoRA Alpha (α)	16
LoRA Dropout	0
Target Modules	Q, K, V, O, Gate, Up, Down
Sequence Length	1024
Batch Size	32
Gradient Samples	256
Learning Rate	2×10^{-4}
Weight Decay	0
Warmup Ratio	0.03
LR Scheduler	Cosine
Epochs	1
Optimizer	AdamW

J LLM USAGE

The use of large language models (LLMs) was limited to general-purpose assistance, such as improving the clarity of the exposition and improving grammar. No parts of the research ideation or theoretical development relied on LLMs, and thus the models are not regarded as contributors to the intellectual content of this work. The authors take full responsibility for all statements and results presented in the paper, and acknowledge that LLMs are not eligible for authorship.