PRODA: PROFILE-DECOMPOSED ADAPTATION FOR CAPTURING THE STRUCTURED RESIDUAL IN LOW-RANK UPDATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-Rank Adaptation (LoRA), a cornerstone of parameter-efficient fine-tuning (PEFT), relies on the core assumption that weight updates are inherently low-rank. We challenge this premise, revealing that this low-rank approximation systematically neglects a significant and highly structured residual, which we term the **missing profile**. To harness this insight, we introduce Profile-Decomposed Adaptation (ProDA), a novel method that captures this residual using highly efficient, axis-aligned vectors. Critically, instead of treating this component as a static error to be corrected, ProDA integrates it multiplicatively, allowing it to dynamically re-scale and modulate the primary low-rank update. Our extensive experiments validate the effectiveness of this approach. ProDA establishes a new state-of-the-art on commonsense reasoning benchmarks and, remarkably, surpasses even full fine-tuning on the GLUE benchmark, suggesting it can act as a powerful regularizer that fosters generalizability. Moreover, on complex generative tasks where standard LoRA falters, ProDA dramatically narrows the performance gap to full fine-tuning. These findings validate our central thesis: the structured residual in PEFT is not mere noise, but a rich signal for synergistic exploitation.

1 INTRODUCTION

Parameter-Efficient Fine-Tuning (PEFT) has become indispensable for adapting Large Language Models (LLMs) (Brown et al., 2020; Touvron et al., 2023; Chang et al., 2024), which are predominantly based on the Transformer architecture (Vaswani et al., 2017) and whose immense scale renders full fine-tuning computationally infeasible. In response, a spectrum of PEFT techniques has emerged. These range from methods that add tunable soft prompts (Lester et al., 2021) or prefixes (Li & Liang, 2021) to the input, to those that insert small adapter modules (Houlsby et al., 2019) or tune only bias terms (Zaken et al., 2021). Among these, Low-Rank Adaptation (LoRA) (Hu et al., 2021) has become a dominant paradigm due to its effectiveness and efficiency.

The success of LoRA has inspired a thriving ecosystem. This includes efficiency-focused methods like QLoRA (Dettmers et al., 2024) and, critically, new architectural variants that question LoRA's core mechanism. For instance, recent works like DoRA (Liu et al., 2024) and PiSSA (Meng et al., 2024) propose decomposing the pre-trained weights themselves to refine the adaptation process. Despite these advances, the community's central hypothesis has remained anchored to a single principle, inspired by findings on intrinsic dimensionality (Aghajanyan et al., 2020): that the weight update matrix (ΔW) can be effectively captured by a low-rank approximation. This foundational assumption, however, has been largely accepted without rigorous empirical validation.

In this work, we argue that focusing solely on the low-rank component, as illustrated by the principle in Figure 1, is a critical limitation. We posit that the low-rank hypothesis is systematically incomplete. We find that the true weight update, ΔW, is not purely low-rank. Instead, it is a composite of a dominant low-rank matrix and a substantial, highly structured residual that prior methods inherently fail to capture. We term this component the **delta profile**, representing what has been the conceptual **"missing profile"** in prior adaptation frameworks. While Figure 1 visualizes the assumption we challenge, the extensive experiments in the subsequent sections provide the definitive empirical evidence for this structural gap and the power of modeling it.

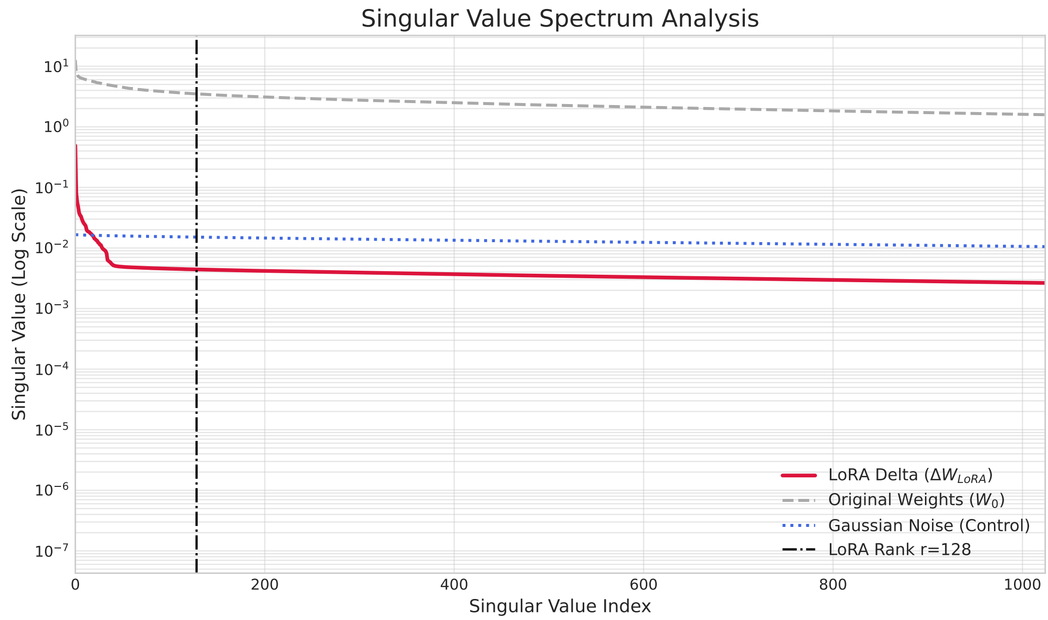

Figure 1: Singular value spectrum of the Low-Rank Adaptation (LoRA) principle, comparing a pre-trained weight matrix (W_0) with its update matrix (ΔW_{LoRA}). The spectrum of ΔW_{LoRA} confirms its explicitly low-rank nature by decaying sharply at the specified rank ($r = 128$), in stark contrast to the high-rank profile of the original weights. This visualization exemplifies the foundational low-rank hypothesis that our work re-examines and extends.

This discovery recasts the problem: the goal is not merely to refine the low-rank update, but to model the complete decomposition of the weight delta itself. To this end, we introduce **Pro**file-**D**ecomposed **A**daptation (**ProDA**), a framework that captures and synergistically integrates both components. ProDA first learns the missing profile using an extremely efficient parameterization. More critically, it moves beyond a simple additive patch by introducing a multiplicative, input-dependent mechanism. This allows the profile to act as a dynamic gate, modulating the low-rank update on a per-token basis and enabling a genuine synergistic collaboration.

Our contributions are threefold:

- We are the first to systematically demonstrate that the true weight update is a composite structure, revealing the incompleteness of the conventional low-rank hypothesis and identifying the **delta profile** as the key overlooked component.
- We propose **ProDA**, a novel PEFT method that directly models this decomposition of the weight update, synergistically integrating the profile and the low-rank component via a computationally efficient multiplicative mechanism.
- We conduct extensive experiments showing that ProDA establishes a new state-of-the-art across diverse benchmarks, surpassing strong baselines, including LoRA (Hu et al., 2021), DoRA (Liu et al., 2024), and PiSSA (Meng et al., 2024), thereby validating our thesis that this previously overlooked component is not noise, but a rich, exploitable signal.

2 Related Work

Parameter-Efficient Fine-Tuning (PEFT) has emerged as a critical paradigm for adapting foundation models without the prohibitive costs of full fine-tuning. The field has progressed through three key approaches. First, **additive methods** insert lightweight modules, such as Adapters (Houlsby et al., 2019), between a model's frozen transformer blocks. Second, **prompt-based tuning** freezes the entire model and optimizes continuous "soft prompts" prepended to the input sequence (Lester et al., 2021; Li & Liang, 2021). The third and arguably most influential paradigm, **reparameterizing**

weight updates, is pioneered by Low-Rank Adaptation (LoRA) (Hu et al., 2021)—the central focus of our work. LoRA is predicated on the observation that the weight update (ΔW) has a low intrinsic rank; by approximating it with a low-rank decomposition, LoRA achieves a compelling trade-off between performance and efficiency that has established it as a foundational technique.

Our work is situated within a recent wave of research that, acknowledging the limitations of the original low-rank hypothesis, seeks to enhance LoRA's expressiveness. One significant line of research focuses on improving the core low-rank decomposition itself, with methods like DoRA (Liu et al., 2024) improving stability and SVD-inspired approaches like PiSSA (Meng et al., 2024) finding a better low-rank basis. Other directions introduce more dynamism, such as AdaLoRA (Zhang et al., 2023) which adaptively allocates ranks, or aim to better align the training dynamics with full fine-tuning, as explored in LoRA-GA (Wang et al., 2024b) and LoRA-Pro (Wang et al., 2024c). The third key direction, which our work advances, involves augmenting the LoRA update. While antecedent methods like LoRA+ (Hayou et al., 2024a) propose simple scalar corrections, ProDA's contribution is fundamentally different. Rather than adding a simple scalar, we identify and model the *structured residual* inherent in the LoRA approximation—the "delta profile." By parameterizing this profile with efficient row and column vectors, inspired by early vector-based methods (Zaken et al., 2021; Liu et al., 2022), and crucially, by engineering a synergistic, multiplicative interaction between this profile and the low-rank update, ProDA provides a more holistic and principled model of the true weight delta.

3 METHODOLOGY: A SYNERGISTIC VIEW OF PROFILE-DECOMPOSED ADAPTATION

Our methodology is built upon a key insight: the weight update matrix ΔW in fine-tuning exhibits a structure that is not fully captured by the low-rank hypothesis alone. We argue that ΔW can be decomposed into a dominant low-rank component, which forms the basis of LoRA (Hu et al., 2022), and a structured component we term the **delta profile**. This section first revisits LoRA to ground our discussion, then progressively develops our method, **ProDA**, by first modeling this profile as a simple additive correction and subsequently evolving this concept into a synergistic formulation where the profile dynamically modulates the low-rank adaptation process itself.

3.1 PRELIMINARIES: REVISITING LOW-RANK ADAPTATION (LORA)

Parameter-Efficient Fine-Tuning (PEFT) methods adapt large pre-trained models by training only a small fraction of their parameters. Among these, Low-Rank Adaptation (LoRA) (Hu et al., 2022) is motivated by the observation that the weight update matrix, ΔW, for a pre-trained weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, often has a low intrinsic rank (Aghajanyan et al., 2020). Consequently, LoRA approximates ΔW with a low-rank decomposition, $\Delta W \approx BA$, where $B \in \mathbb{R}^{d_{\text{out}} \times r}$, $A \in \mathbb{R}^{r \times d_{\text{in}}}$, and the rank $r \ll \min(d_{\text{out}}, d_{\text{in}})$. During adaptation, W_0 remains frozen while only A and B are trained. The forward pass is modified as:

$$y = W_0 x + s \cdot BAx \tag{1}$$

where $x \in \mathbb{R}^{d_{\text{in}}}$, $y \in \mathbb{R}^{d_{\text{out}}}$, and s is a scaling hyperparameter. A key advantage of LoRA is its inference efficiency; the learned matrices B and A can be merged into W_0 ($W' = W_0 + s \cdot BA$), introducing zero additional latency.

3.2 PRINCIPLE 1: THE DELTA PROFILE AS AN ADDITIVE CORRECTION

Our central hypothesis is that the low-rank approximation $\Delta W \approx BA$ is incomplete, leaving a structured residual we call the **delta profile**, P. Thus, the true update can be more accurately represented as $\Delta W = BA + P$. As our first principle, we model this profile in its most direct form: a global structural offset. We hypothesize this offset can be efficiently parameterized by axis-aligned components, namely a column vector $b_c \in \mathbb{R}^{d_{\text{out}}}$ and a row vector $b_r \in \mathbb{R}^{d_{\text{in}}}$. While other methods also employ vector-based adaptations (Zaken et al., 2021; Liu et al., 2022), our formulation is distinct as it is explicitly derived from modeling the residual of the low-rank hypothesis. The resulting additive profile is:

$$P_{\text{add}} = b_c \mathbf{1}_{d_{\text{in}}}^T + \mathbf{1}_{d_{\text{out}}} b_r^T \tag{2}$$

We opt for this rank-2 outer product formulation for two primary reasons. **First, parameter efficiency:** this structure captures global, axis-aligned biases using only $d_{\text{in}} + d_{\text{out}}$ parameters, which is a highly efficient method for modeling row-wise and column-wise corrective signals. **Second, structural intuition:** this form can be interpreted as learning a global "vertical" and "horizontal" adjustment for the entire weight matrix W_0, correcting for systematic shifts that the low-rank update BA inherently neglects. It serves as a powerful yet simple first-order approximation of the structured residual error.

Combining this with LoRA provides a baseline that applies a static correction for the global error components missed by the low-rank update:

$$y_{\text{static}} = W_0 x + s \cdot BAx + (b_c \mathbf{1}_{d_{\text{in}}}^T + \mathbf{1}_{d_{\text{out}}} b_r^T)x \tag{3}$$

This additive model provides a first-order correction for LoRA's approximation error. However, it treats the low-rank update and the profile correction as two independent processes. This raises a fundamental question: Should the profile's role be confined to a static offset, or can it play a more integral part in the adaptation process?

3.3 Principle 2: The Profile as a Synergistic Modulator

We posit that a more expressive and powerful adaptation model must capture the interdependence between the low-rank and profile components. Instead of only providing a static correction, the delta profile P should also **dynamically modulate** the low-rank update BA. This principle of conditional computation, where one component's output shapes the behavior of another, has proven effective in contexts such as feature modulation (Perez et al., 2018).

To achieve this, we formulate a synergistic update rule where the profile contributes both additively and multiplicatively. This leads to our final **Profile-Decomposed Adaptation (ProDA)** formulation:

$$\Delta W_{\text{ProDA}} = \underbrace{BA}_{\text{Low-Rank Update}} + \underbrace{P}_{\text{Additive Profile}} + \underbrace{\gamma \odot (BA \odot P)}_{\text{Modulated Interaction}} \tag{4}$$

where $P = b_c \mathbf{1}_{d_{\text{in}}}^T + \mathbf{1}_{d_{\text{out}}} b_r^T$, $\odot$ denotes the Hadamard product, and γ is a learnable gating mechanism that controls the strength of the modulation. This gate combines a global scalar γ_{global} with an input-dependent term: $\gamma = \gamma_{\text{global}} + \sigma(\text{Controller}(x))$. To minimize parameter overhead, the 'Controller(x)' is implemented as a lightweight network. Specifically, it consists of a two-layer MLP with a down-projection to a small bottleneck dimension d_{bottle} and an up-projection back to a scalar output:

$$\text{Controller}(x) = W_{\text{up}} \cdot \text{ReLU}(W_{\text{down}} x) \tag{5}$$

where $W_{\text{down}} \in \mathbb{R}^{d_{\text{bottle}} \times d_{\text{in}}}$ and $W_{\text{up}} \in \mathbb{R}^{1 \times d_{\text{bottle}}}$. The complete forward pass for ProDA is a unified computation:

$$y_{\text{ProDA}} = (W_0 + \Delta W_{\text{ProDA}})x \tag{6}$$

This formulation elevates the delta profile from a simple corrective term to an integral, dynamic modulator of the fine-tuning process. It enables a richer and more expressive adaptation than what is achievable with independent components. For inference, the static components of ProDA—namely, the low-rank update BA and the additive profile P—can be merged into the original weights. The modulated interaction term introduces a minimal, input-dependent computational path, a trade-off we find favorable given the significant gains in expressiveness and performance.

Parameter Analysis. For a weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA introduces $N_{\text{LoRA}} = r(d_{\text{in}} + d_{\text{out}})$ trainable parameters. Our ProDA method extends this by adding $N_{\text{profile}} = d_{\text{out}} + d_{\text{in}}$ parameters for the profile vectors and $N_{\text{controller}} = d_{\text{in}} d_{\text{bottle}} + d_{\text{bottle}} + 1$ parameters for the controller (defined in Eq. 5). The total parameter count is thus $N_{\text{ProDA}} = N_{\text{LoRA}} + N_{\text{profile}} + N_{\text{controller}}$. Given that r and d_{bottle} are small, N_{ProDA} remains significantly smaller than the $d_{\text{in}} \times d_{\text{out}}$ parameters of the full matrix, preserving the core efficiency of PEFT methods.

4 Experiments

This section presents a comprehensive evaluation of our proposed method, ProDA. Our experiments are designed to validate three central claims: 1) ProDA demonstrates superior performance compared to state-of-the-art PEFT methods across a diverse set of challenging language tasks. 2) Our core scientific premise—the ”low-rank + profile” structure of the weight delta—is empirically sound and is the primary source of ProDA’s effectiveness. 3) Each component within the ProDA framework contributes meaningfully to the final performance, as demonstrated through rigorous ablation studies.

4.1 Experimental Setup

Models and Datasets To validate ProDA’s versatility, we conduct experiments across both decoder-only and encoder-only architectures. For decoder models, we use **LLaMA-2-7B** (AI@Meta, 2023) and its successor **LLaMA-3-8B** (AI@Meta, 2024), which are prominent open-source autoregressive models. To assess generalization to natural language understanding (NLU), we employ the widely-used encoder model, **RoBERTa-base** (Liu et al., 2019).

Our evaluation spans three distinct benchmark suites designed to test a wide range of capabilities: (1) **Commonsense Reasoning:** We use a broad collection of eight datasets to measure general reasoning, including 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 (Clark et al., 2018), and OBQA (Mihaylov et al., 2018). (2) **Natural Language Understanding:** We use the standard **GLUE** benchmark (Wang et al., 2018) for a comprehensive NLU evaluation. (3) **Complex Generative Tasks:** To test performance on more demanding generative reasoning, we use **MT-Bench** (Zheng et al., 2024) for evaluating conversational abilities, **GSM8K** (Cobbe et al., 2021) for multi-step mathematical reasoning, and **HumanEval** (Chen et al., 2021) for code generation.

Compared Methods We benchmark ProDA against a comprehensive set of baselines. Our practical upper bound is **Full Fine-Tuning (Full FT)**, which updates all model parameters. The foundational baseline is **LoRA** (Hu et al., 2021), upon which our work is built. We also compare against a diverse set of state-of-the-art LoRA variants. These include **DoRA** (Liu et al., 2024), which decomposes weights into magnitude and direction; **PiSSA** (Meng et al., 2024), which uses principal singular vectors for initialization; **AdaLoRA** (Zhang et al., 2023), which adaptively allocates parameter budgets; **Delta-LoRA** (Zi et al., 2023), which re-parameterizes the update; and other strong competitors such as **DyLoRA** (Valipour et al., 2022), **MELoRA** (Ren et al., 2024), **rsLoRA** (Kalajdzievski, 2023), **LoRA+** (Hayou et al., 2024b), and **LoRA-GA** (Wang et al., 2024a).

Implementation Details We implemented our experiments in PyTorch using the Hugging Face Transformers library. All models were fine-tuned on NVIDIA L40 GPUs. Across all experiments, we used the AdamW optimizer (Loshchilov & Hutter, 2019) with a linear learning rate scheduler featuring a 10% warmup phase. Following standard practice, we applied ProDA’s adaptation modules to all linear layers within the transformer blocks (i.e., the query, key, value, and output projections). For ProDA, the default LoRA rank was set to $r = 8$ with a scaling factor $\alpha = 16$ and a dropout of 0.05. The additive profile vectors and the global modulation scalar were zero-initialized, while the controller’s weights were Kaiming-initialized.

We tailored hyperparameters for different model families to ensure optimal performance. **For RoBERTa-base (Liu et al., 2019) on GLUE,** we used a learning rate of 2×10^{-4} and trained for 3 epochs with a batch size of 32 and a maximum sequence length of 512. **For LLaMA models (AI@Meta, 2023; 2024) on generative tasks,** we used a lower learning rate of 3×10^{-5} for greater stability. To accommodate the larger model size, we employed a per-device batch size of 4 with 8 gradient accumulation steps, achieving an effective batch size of 32. Training was conducted for 3 epochs with a sequence length of 2048.

To ensure a fair comparison, all baseline methods were trained under identical settings for each task family. All reported results are the mean and standard deviation ($\pm$ std. dev.) from three runs with different random seeds to guarantee statistical robustness.

4.2 Experiments and Analysis

We conduct a comprehensive set of experiments to validate ProDA and answer three central questions. First, what is the empirical evidence for the ”delta profile” that motivates our work? Second, how does ProDA perform against state-of-the-art PEFT methods across diverse tasks and model architectures? Finally, what is the individual contribution of each component in our proposed synergistic design? We address these questions in the subsequent sections, providing a thorough validation of our approach.

Table 1: Main results on eight commonsense reasoning benchmarks for the LLaMA family. We compare ProDA against strong PEFT baselines. The best-performing method is in **bold** and the second-best is <u>underlined</u>. LoRA and DoRA results are from Liu et al. (2024).

Model	**Method**	**BoolQ**	**PIQA**	**SIQA**	**Hella Swag**	**Wino Grande**	**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
LLaMA2-7B	LoRA	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
	DoRA	71.8	83.7	76.0	89.1	82.6	83.7	68.2	82.4	79.7
	PiSSA	<u>75.0</u>	**87.0**	<u>81.6</u>	<u>95.0</u>	<u>86.5</u>	<u>88.5</u>	<u>75.9</u>	<u>86.4</u>	<u>84.5</u>
	ProDA	**75.7**	<u>86.9</u>	**83.2**	**95.8**	**87.8**	**89.2**	**76.9**	**88.1**	**85.5**
LLaMA3-8B	LoRA	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	DoRA	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	85.2
	PiSSA	**77.2**	<u>90.0</u>	<u>82.9</u>	<u>96.6</u>	<u>88.4</u>	**93.6**	<u>82.4</u>	<u>87.4</u>	<u>87.3</u>
	ProDA	<u>77.1</u>	**90.5**	**83.3**	**97.2**	**89.6**	<u>93.4</u>	**83.9**	**89.2**	**88.0**

4.2.1 Main Results on Commonsense Reasoning

We first evaluate ProDA on a diverse suite of eight challenging commonsense reasoning benchmarks. The results, presented in Table 1, demonstrate that ProDA sets a new state-of-the-art for parameter-efficient fine-tuning. On the LLaMA-2-7B model, ProDA achieves an average score of **85.5**, decisively outperforming the strong PiSSA (Meng et al., 2024) baseline by 1.0 point and the foundational LoRA (Hu et al., 2021) method by a significant **7.9** points. This substantial margin provides a resounding validation of our central thesis: explicitly modeling the structured delta profile is not an incremental tweak, but a critical and previously overlooked component for effective adaptation.

Crucially, this performance advantage is not an isolated finding but is consistently reinforced on the more advanced LLaMA-3-8B architecture, underscoring the robustness and scalability of our approach. Here, ProDA secures the top average score of **88.0**, maintaining a clear advantage over both the highly competitive PiSSA (Meng et al., 2024) (+0.7) and DoRA (Liu et al., 2024) (+2.8) baselines. This consistent superiority across model generations strongly suggests that ProDA's architectural principle—modeling the synergistic interplay between the low-rank update and its structural residual—is a more fundamental and generalizable approach than existing methods. By capturing and leveraging this intricate relationship, ProDA consistently unlocks a higher performance ceiling, establishing a new and compelling standard for the field.

Table 2: Performance comparison on the GLUE benchmark using RoBERTa-base. ProDA is evaluated against Full Fine-Tuning and other PEFT methods. For each task, the best result is in **bold** and the second-best is <u>underlined</u>. All baseline results are sourced from Ren et al. (2024).

Method	**MRPC**	**RTE**	**CoLA**	**STS-B**	**SST-2**	**QQP**	**QNLI**	**MNLI**	**Avg.**
Full FT	88.2	84.1	<u>64.6</u>	90.6	94.3	**92.0**	92.7	<u>87.5</u>	86.8
LoRA	89.9	85.9	62.4	91.4	94.4	90.8	92.6	86.9	86.8
DyLoRA	89.5	84.5	61.1	91.1	94.3	90.2	92.2	86.3	86.2
AdaLoRA	90.2	85.2	61.6	91.2	94.5	90.1	93.1	87.3	86.7
Delta-LoRA	90.2	<u>87.0</u>	63.8	91.6	95.1	90.9	93.1	<u>87.5</u>	<u>87.5</u>
MELoRA	<u>90.9</u>	86.6	64.1	<u>91.9</u>	<u>95.4</u>	90.8	<u>93.2</u>	87.2	<u>87.5</u>
ProDA	**91.7**	**88.1**	**65.6**	**92.4**	**95.7**	<u>91.5</u>	**93.9**	**87.6**	**88.3**

4.2.2 Generalization to Encoder Architectures and NLU Tasks

To test the generality of our approach beyond decoder-only models, we evaluated ProDA on the GLUE benchmark using a RoBERTa-base (Liu et al., 2019) encoder architecture. The results, presented in Table 2, are striking. ProDA achieves an average score of **88.3**, not only outperforming all other parameter-efficient baselines, including the strong MELoRA and Delta-LoRA (Zi et al., 2023) methods (+0.8 points), but more remarkably, surpassing full fine-tuning by a significant **1.5-point margin**. This counter-intuitive result suggests ProDA acts not merely as a parameter-efficient proxy for full adaptation, but as a **potent regularizer**, potentially preventing the overfitting that can occur during full fine-tuning.

A deeper look at the per-task breakdown reveals the source of this dominant performance. ProDA's superiority does not stem from a high variance across tasks, but from broad-based excellence. It achieves the top score on **seven of the eight** GLUE tasks, spanning diverse capabilities from semantic similarity (MRPC) to natural language inference (RTE) and grammatical acceptability (CoLA). The only exception is QQP, where full fine-tuning's access to the complete parameter space confers a slight advantage. This pattern of consistent, top-tier performance strongly supports our central hypothesis. The synergistic interplay between the low-rank update and the structural delta profile provides a more robust and universally applicable adaptation trajectory, enhancing a wide range of NLU capabilities simultaneously. By constraining the adaptation to a decomposed, low-dimensional manifold, ProDA effectively filters out noise and task-specific artifacts, forcing the model to learn more generalizable features. This demonstrates that the principles of profile-decomposed adaptation are foundational, leading to a more fundamentally capable model.

Table 3: Evaluating ProDA's capabilities in generative reasoning and coding. This table compares PEFT methods against the full fine-tuning of LLaMA-2-7B on three key benchmarks: dialogue simulation (MT-Bench), mathematical problem-solving (GSM8K), and code synthesis (HumanEval Pass@1). We also investigate ProDA's performance scalability by increasing its rank capacity. Scores are the mean ($\pm$ std. dev.) of three runs, with the top two results highlighted in **bold** and with an <u>underline</u>.

Method	MT-Bench	GSM8K	HumanEval	Avg.
Full FT	$5.30_{\pm 0.11}$	$\mathbf{59.36}_{\pm 0.85}$	$\mathbf{35.31}_{\pm 2.13}$	**33.32**
LoRA ($r = 8$)	$5.61_{\pm 0.10}$	$42.08_{\pm 0.04}$	$14.76_{\pm 0.17}$	20.82
DoRA ($r = 8$)	$5.97_{\pm 0.02}$	$53.07_{\pm 0.75}$	$19.75_{\pm 0.41}$	26.26
AdaLoRA ($r = 8$)	$5.57_{\pm 0.05}$	$50.72_{\pm 1.39}$	$17.80_{\pm 0.44}$	24.70
PiSSA ($r = 8$)	$5.30_{\pm 0.02}$	$44.54_{\pm 0.27}$	$16.02_{\pm 0.78}$	21.95
rsLoRA ($r = 8$)	$5.25_{\pm 0.03}$	$45.62_{\pm 0.10}$	$16.01_{\pm 0.79}$	22.29
LoRA+ ($r = 8$)	$5.71_{\pm 0.08}$	$52.11_{\pm 0.62}$	$18.17_{\pm 0.52}$	25.33
LoRA-GA ($r = 8$)	$5.95_{\pm 0.16}$	$53.60_{\pm 0.30}$	$19.81_{\pm 1.46}$	26.45
LoRA-GA ($r = 32$)	$5.79_{\pm 0.09}$	$55.12_{\pm 0.30}$	$20.18_{\pm 0.19}$	27.03
LoRA-GA ($r = 128$)	$6.13_{\pm 0.07}$	$55.07_{\pm 0.18}$	$23.05_{\pm 0.37}$	28.08
ProDA ($r = 8$)	$6.12_{\pm 0.28}$	$54.35_{\pm 0.52}$	$21.14_{\pm 0.33}$	27.20
ProDA ($r = 32$)	$\underline{6.32_{\pm 0.24}}$	$55.58_{\pm 0.47}$	$21.23_{\pm 0.64}$	27.71
ProDA ($r = 128$)	$\mathbf{6.42}_{\pm 0.37}$	$\underline{56.43_{\pm 0.76}}$	$\underline{23.28_{\pm 0.40}}$	<u>28.71</u>

4.2.3 Performance on Complex Generative Reasoning Tasks

We now assess ProDA on a suite of complex generative reasoning tasks, with results detailed in Table 3. The findings confirm ProDA's decisive superiority in this demanding arena. Even at a constrained rank of 8, ProDA achieves an average score of **27.20**, the highest among all rank-8 PEFT methods. This is driven by strong performance across all tasks, including the challenging conversational and instruction-following capabilities measured by MT-Bench. The remarkable **+6.38** point average lead over the standard LoRA (Hu et al., 2021) is a powerful testament to our core thesis: for intricate generative tasks, merely capturing a low-rank update is insufficient. ProDA's explicit modeling of the global delta profile provides the critical expressive power that these tasks demand.

More compellingly, ProDA exhibits exceptional performance scaling, beginning to challenge the dominance of full fine-tuning. As the rank capacity is increased to 128, ProDA's average score climbs to **28.71**. This dramatically narrows the performance gap to Full Fine-Tuning to just **4.61** points—a stark contrast to the **12.5**-point chasm for the standard LoRA (Hu et al., 2021). This impressive scaling, particularly on the difficult GSM8K and HumanEval benchmarks, showcases that ProDA is not simply an efficient proxy but a potent adaptation framework in its own right. It provides a highly effective and scalable path toward achieving near full fine-tuning performance on complex generative tasks, without incurring the prohibitive costs of full model training.

4.2.4 Ablation Study of ProDA Components

To isolate the contributions of ProDA's core components, we conducted a rigorous ablation study, with results presented in Table 4. This analysis validates our two-principle design by progressively building from a LoRA (Hu et al., 2021) baseline to the full ProDA model.

Table 4: Detailed ablation analysis of ProDA's components on LLaMA-2-7B (rank 8). The study starts with a LoRA baseline and sequentially introduces the additive profile, followed by the synergistic interaction, to quantify the performance contribution of each. Both components are shown to be critical for achieving the final performance.

Method Configuration	**Commonsense Avg.**	**GSM8K**
(1) LoRA (Baseline)	77.6	42.1
(2) + Additive Profile	84.2	51.9
(3) **+ Synergistic Interaction (Full ProDA)**	**85.5**	**54.4**

The results unequivocally demonstrate the impact of each proposed component. First, incorporating only the additive profile (Row 2) yields a substantial performance leap over the LoRA baseline (Row 1), boosting the Commonsense Reasoning average by 6.6 points and the GSM8K score by a notable 9.8 points. This large gain confirms that modeling the structural residual is critical. Second, the subsequent introduction of the synergistic interaction term (Row 3) delivers further crucial gains, adding another 1.3 points to the Commonsense average and 2.5 points to GSM8K, thereby cementing the full ProDA model's superior performance. This two-step improvement provides compelling evidence for our central thesis: while the additive profile corrects for a major deficiency in LoRA, it is the dynamic, synergistic modulation that unlocks the model's full potential.

5 Conclusion

In this work, we challenge the prevailing low-rank hypothesis in parameter-efficient fine-tuning, positing that it provides an incomplete picture of the weight update process. We identify and empirically validate the existence of a systematic structural error in the widely-used LoRA approximation—a component we term the "delta profile." Our central contribution, ProDA, is a new PEFT framework that moves beyond merely augmenting LoRA to offer a more principled and complete model of the true weight delta. ProDA achieves this by first learning this profile directly, and critically, by modeling the synergistic, multiplicative interaction between this profile and the low-rank update.

Our extensive experiments empirically validate ProDA's superiority, establishing a new state-of-the-art for PEFT across diverse architectures and a wide spectrum of tasks, from commonsense reasoning to demanding generative reasoning. Strikingly, ProDA's performance can even surpass that of full fine-tuning, suggesting it acts as a powerful regularizer. The core message of this work is that the future of parameter-efficient adaptation lies not in refining the low-rank component in isolation, but in holistically modeling the complete structure of the fine-tuning delta. The "missing profile" is not a peripheral error to be minimized, but a rich, structured signal to be synergistically exploited.

REFERENCES

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

AI@Meta. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/arXiv.2307.09288. URL `https://doi.org/10.48550/arXiv.2307.09288`.

AI@Meta. Llama 3 model card, 2024. URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 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.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024a.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models, 2024b. URL `https://arxiv.org/abs/2402.12354`.

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, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora, 2023. URL `https://arxiv.org/abs/2312.03732`.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 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. *arXiv preprint arXiv:2402.09353*, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *arXiv preprint arXiv:2404.02948*, 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. *arXiv preprint arXiv:1809.02789*, 2018.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Jiahuan Pei. Melora: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.17263*, 2024.

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. *arXiv preprint arXiv:1904.09728*, 2019.

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.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. *arXiv preprint arXiv:2407.05000*, 2024a.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. *arXiv preprint arXiv:2407.05000*, 2024b.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized?, 2024c. URL `https://arxiv.org/abs/2407.18242`.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

Bojia Zi, Xianbiao Qi, Lingzhi Wang, Jianan Wang, Kam-Fai Wong, and Lei Zhang. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. *arXiv preprint arXiv:2309.02411*, 2023.

6 APPENDIX

6.1 REPRODUCIBILITY STATEMENT

To ensure full reproducibility, we will release all source code, fine-tuned ProDA adapters, and experiment scripts under an Apache 2.0 license on GitHub upon publication. Our implementation is built upon PyTorch, Hugging Face Transformers, and PEFT, and was run on NVIDIA L40 GPUs. All experiments were conducted on publicly available models from the Hugging Face Hub (e.g., `LLaMA-2/3`, `RoBERTa-base`) and standard benchmarks (e.g., GLUE, GSM8K), using their official data splits. Key hyperparameters are detailed in Section 4.1. All reported metrics are the mean and standard deviation from three independent runs with different random seeds to ensure statistical robustness.

LLM USAGE STATEMENT

In refining the prose of this manuscript, we employed a large language model (LLM) for assistance. Its application was restricted to proofreading and improving the clarity of our writing. We affirm that all intellectual contributions—from the formulation of the central hypothesis and the design of ProDA, to the analysis of the results—are entirely our own. We take full accountability for the content and findings of this research.