OPT2024: 16th Annual Workshop on Optimization for Machine Learning

A Unified Convergence Theory for Large Language Model Efficient Fine-tuning

Zhanhong Jiang ZHJIANG@IASTATE.EDU
Iowa State University
Nastaran Saadati NSAADATI@IASTATE.EDU
Iowa State University
Aditya Balu BADITYA@IASTATE.EDU
Iowa State University
Minh Pham MP5847@NYU.EDU
New York University
Joshua R. Waite JRWAITE@IASTATE.EDU
Iowa State University
Nasla Saleem NASLA@IASTATE.EDU
Iowa State University
Chinmay Hegde CHINMAY.H@NYU.EDU
New York University
Soumik Sarkar SOUMIKS@IASTATE.EDU
Iowa State University

Abstract

Parameter efficient fine-tuning (PEFT) has gained considerable attention by manipulating model parameters, and low-rank adaptation (LoRA) is deemed the state-of-the-art technique in PEFT. Though LoRA has witnessed its great successes in numerous fields, there still exists a theoretical gap on how it enables convergence. More recently, representation fine-tuning (ReFT) was developed by turning the fine-tuning to hidden representations of a model that encode significant semantic information, seemingly yielding the better performance than LoRA. In this work, we first establish the connection between LoRA and ReFT and then unify them into a meta-algorithm, dubbed model efficient fine-tuning (MeFT). MeFT not only provably shows the best available convergence rate coinciding with that in existing algorithms, but also theoretically reveals the relationship between the low rank and the convergence error. Our analysis facilitates the theoretical understanding of how low-rank decomposition fine-tuning techniques drive LLMs and offers useful insights for more efficient future algorithm design.

1. Introduction

Fine-tuning [4, 39] has remained as the extremely popular and critical technique to expedite the adaptation of large language models (LLMs) for achieving desirable performance in downstream tasks, as it is memory-efficient and computationally tractable. Parameter efficient fine-tuning (PEFT) is now the workhorse to satisfy such need due to the emergence of low-rank adaptation (LoRA) [3, 10, 13, 20, 22, 33, 34], which remarkably reduces the number of trainable parameters by learning pairs of rank-decomposition matrices, while freezing the original weights. Nevertheless, all existing

attempts have primarily focused on applications and empirical performance, leaving their relevant convergence rates as a question mark. Additionally, a recent work [31] proposed another novel fine-tuning approach termed representation fine-tuning (ReFT), which instead schematically fine-tunes the hidden representations that encode rich semantic information from data, by conducting interventions on hidden layers. ReFT leads to the seemingly better model accuracy in different tasks compared to LoRA and establishes a new line of work in model fine-tuning. However, regardless of its stunning performance, there is still no rigorously theoretical guarantee provided from ReFT. Thereby, in this work, we try to bridge the gap between the practice and theory.

Contributions. To address the above issues, we tactically analyze the favored LoRA algorithm and the recently proposed ReFT algorithm, and surprisingly discover that both can be unified into a meta-algorithm, dubbed model efficient fine-tuning (MeFT). We then establish the theoretical convergence rate for MeFT by deriving the new smoothness constant for the objective loss with a few mild assumptions, under the scenarios of low-rank decomposition/projection. Specifically, the contributions we make in this work are outlined as follows: a) We propose MeFT to unify the parameter efficient fine-tuning and representation fine-tuning. The unification also naturally applies to other variants of LoRA; b) we establish the relationship between the convergence error bound for MeFT and the intrinsic rank, resorting to the smoothness condition of the non-convex objective with respect to the low-rank matrix. Please see Appendix for additional analysis and/or results.

2. Preliminaries and Problem Formulation

LoRA. The forward pass of a deep neural network involves weight matrix multiplications in numerous dense layers. Though these matrices are full-rank, when adapting the model to specific tasks, the authors in [1] have surprisingly shown that a pre-trained LLM has a low "intrinsic dimension", which maintains the efficient learning even if it is projected into a smaller subspace. Inspired by this, Hu et al. [10] proposed the vanilla LoRA, which implied that during fine-tuning, the updates to the weights also have a low "intrinsic rank". Since its emergence, LoRA has set the state-of-the-art. Considering a pre-trained model, we denote by $\mathbf{W}_0 \in \mathbb{R}^{d\times k}$ its original weight matrix. The updates to $\mathbf{W}_0$ is denoted as $\Delta\mathbf{W}$ and can be represented by a low-rank decomposition between two matrices, $\mathbf{A} \in \mathbb{R}^{r\times k}$ and $\mathbf{B} \in \mathbb{R}^{d\times r}$. r is the *rank* in this context, and it satisfies that $r \ll \min\{d, k\}$. Then the following relationship is obtained:

$$\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W} = \mathbf{W}_0 + \mathbf{BA}. \tag{1}$$

As LoRA only conducts fine-tuning, $\mathbf{W}_0$ is *frozen* and does not get involved in the gradient update during backpropagation. Only $\mathbf{A}$ and $\mathbf{B}$ matrices consist of trainable parameters. The initialization for these two matrices is critical as it should not change $\mathbf{W}$ at the beginning. Hence, each element of $\mathbf{A}_0$ is sampled from a Gaussian distribution, i.e., $a_{ij,0} \sim \mathcal{N}(0, \sigma^2)$, and each element of $\mathbf{B}_0$ is set 0, i.e., $b_{ij,0} = 0$.

ReFT. A technique that has only recently been proposed is ReFT, which intervenes hidden representations of a model to steer it towards successfully implementing downstream tasks during inference time. Two instantiations resorting to low-rank linear subspaces of representations were developed accordingly. We summarize them in the following. Denote by $\mathbf{h} \in \mathbb{R}^d$ the hidden representation and by $\mathbf{R} \in \mathbb{R}^{r\times d}$ and $\mathbf{U} \in \mathbb{R}^{r\times d}$ the low-rank projection matrices, where r is the rank of subspaces and d is the dimension of the hidden-state representation. Note that d in this context has different

physical meaning compared to that in LoRA and we keep using it for the purpose of unification in the latter section. Therefore, the low-rank linear subspace ReFT (LoReFT) can be expressed as

$$\Phi_{LoReFT}(\mathbf{h}) = \mathbf{h} + \mathbf{R}^\top(\mathbf{U}\mathbf{h} + \mathbf{b} - \mathbf{R}\mathbf{h}), \tag{2}$$

which edits the hidden representation in the r-dimensional subspace spanned by the orthonormal rows of $\mathbf{R}$, where $\mathbf{b} \in \mathbb{R}^r$ is the bias. This operation is called distributed interchange intervention (DII) defined in [7] and shown efficient in representation intervention. Essentially, $\Phi_{LoReFT}(\mathbf{h})$ can be deemed as a linear transformation on $\mathbf{h}$ if we simplify Eq. 2 by setting $\mathbf{b} = \mathbf{0}$. Another simplified version called DiReFT gets rid of the orthogonality constraint and difference operation in Eq. 2, leading to training time reduction. It is given by the following formula:

$$\Phi_{DiReFT}(\mathbf{h}) = \mathbf{h} + \mathbf{U}_2^\top(\mathbf{U}_1\mathbf{h} + \mathbf{b}), \tag{3}$$

where both $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{r\times d}$ are low-rank projection matrices. DiReFT is claimed to resemble LoRA [31] as it can be thought of as LoRA applied to hidden representations at certain positions.

Problem formulation. Consider fine-tuning an LLM that is parameterized by $\boldsymbol{\theta} \in \mathbb{R}^m$ in *full rank*, over a dataset $\mathcal{D}$. The cardinality of $|\mathcal{D}|$ is S. Note that in this context, we resort to a vector $\boldsymbol{\theta}$ instead of $\mathbf{W}$ (representing all weight matrices in multiple layers) for problem formulation and convergence analysis, unless $\mathbf{W}$ is used for a specific purpose somewhere in the analysis. $\mathbf{W}$ in practice can be flattened as $\boldsymbol{\theta}$. We have known that when m is extremely large, training the model is computationally expensive. Instead, denoting by $\boldsymbol{v}$ ($\boldsymbol{v} := [\mathbf{A}, \mathbf{B}]$ or $\boldsymbol{v} := [\mathbf{U}_1, \mathbf{U}_2, \mathbf{b}]$) the low-rank adapter (LoRA) or projection matrices (DiReFT), and by $\boldsymbol{\theta}_0$ the pre-trained model parameters, we have the following equation:

$$\min_{\boldsymbol{v}\in\mathbb{R}^n} f(\boldsymbol{v}, \boldsymbol{\theta}_0) := \frac{1}{S}\sum_{s\in\mathcal{D}} f_s(\boldsymbol{v}, \boldsymbol{\theta}_0), \tag{4}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ is the global loss, f_s is the loss corresponding to the sample s. Throughout the paper, we assume f is continuously differentiable, coercive, and bounded below, i.e., $f \geq f^* > -\infty$. We notice that the dimension of $\boldsymbol{v}$ is different from that of $\boldsymbol{\theta}_0$, since in Eq. 4 we only optimize the parameters of low-rank matrices. Another scrutiny is that Eq. 4 is suitable for either LoRA or ReFT as both keep pre-trained model parameters frozen during training.

3. Model Efficient Fine-tuning (MeFT)

In this section, we delve into the proposed MeFT to unify both LoRA and ReFT, followed by its concrete convergence analysis. In this work, we leverage DiReFT instead of LoReFT for simplicity and the analysis techniques for the former can be extended to the latter. Recall LoRA and DiReFT in the following:

$$\text{LoRA} : \mathbf{W} = \mathbf{W}_0 + \mathbf{B}\mathbf{A} \tag{5}$$

$$\text{DiReFT} : \Phi_{DiReFT}(\mathbf{h}) = \mathbf{h} + \mathbf{U}_2^\top(\mathbf{U}_1\mathbf{h} + \mathbf{b}) \tag{6}$$

We let $\mathbf{b} = \mathbf{0}$ and rewrite DiReFT as follows:

$$\Phi_{DiReFT}(\mathbf{h}) = (\mathbf{I} + \mathbf{U}_2^\top\mathbf{U}_1)\mathbf{h} \tag{7}$$

By equating $\mathbf{W}_0 = \mathbf{I}$, $\mathbf{B} = \mathbf{U}_2^\top$, and $\mathbf{A} = \mathbf{U}_1$, we retain LoRA in DiReFT mathematically, as shown in Figure 1 in Appendix A.3. This motivates us to unify these two schemes together. Thus, we propose MeFT as either parameters or representations belong to a model. To ease the analysis, throughout the rest of the paper, we only use the same notations as those in LoRA. One may argue that if $\mathbf{b}$ existing in DiReFT impedes the analysis. A recipe to this is to approximate $\mathbf{b} \approx \mathbf{U}_3\mathbf{h}$ such that the second part of DiReFT can be rewritten as $\mathbf{U}_2^\top(\mathbf{U}_1 + \mathbf{U}_3)\mathbf{h}$, which can be regarded as a linear transformation of the hidden representation. The simplification of $\mathbf{b}$ also follows from the traditionally theoretical analysis for neural network models.

Denote by $\boldsymbol{g} = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \nabla f_s(\boldsymbol{v}, \boldsymbol{\theta}_0)$ the stochastic gradient, where $\mathcal{S}$ is a mini-batch of $\mathcal{D}$. $\boldsymbol{g}$ is assumed to be an unbiased estimate of $\nabla f(\boldsymbol{v}, \boldsymbol{\theta}_0)$, i.e., $\nabla f(\boldsymbol{v}, \boldsymbol{\theta}_0) = \mathbb{E}[\boldsymbol{g}]$. Algorithm 1 shows the algorithmic framework for optimizing parameters $\boldsymbol{v}$ using SGD. Line 4 in Algorithm 1 manifests the

Algorithm 1: MeFT

Data: Input: pre-trained parameters $\boldsymbol{\theta}_0$, initial low-rank adapter $\boldsymbol{v}_0$, the number of epochs T, learning rate α, dataset $\mathcal{D}$

for $t = 1, 2, ..., T$ **do**

Sample a random mini-batch $\mathcal{S}$ from $\mathcal{D}$;

Calculate the stochastic gradient $\boldsymbol{g}_t = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \nabla f_s(\boldsymbol{v}_t, \boldsymbol{\theta}_0)$;

Fine-tune the low-rank parameters: $\boldsymbol{v}_{t+1} = \boldsymbol{v}_t - \alpha\boldsymbol{g}_t$;

end

Result: $\boldsymbol{v}_T$

mini-batch stochastic gradient, which is specifically with respect to $\boldsymbol{v}_t$ only. However, the frozen $\boldsymbol{\theta}_0$ is still necessitated to fulfill the loss value calculation. When low-rank decomposition techniques are applied in MeFT, we are not already aware of any existing results reporting the convergence error bound, or whether it can still maintain the similar convergence rate as $\mathcal{O}(\frac{1}{\sqrt{T}})$ [12]. We would also like to study if the rank r can impact the convergence error bound. To this end, we start with the following assumption.

Assumption 1 *There exists a constant $L > 0$ such that $\|\nabla f(\mathbf{W}_1) - \nabla f(\mathbf{W}_2)\|_F \leq L\|\mathbf{W}_1 - \mathbf{W}_2\|_F$, for all $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d\times k}$, where $\|\cdot\|_F$ is the Frobenius norm.*

This assumption implies that $f(\boldsymbol{\theta})$ is L-smooth but not with low-rank decomposition. We have to make sure the smoothness condition holds for $\mathbf{W} = \mathbf{W}_0 + \mathbf{BA}$ with the following two assumptions.

Assumption 2 *For any $\mathbf{A}$ and $\mathbf{B}$ matrices induced by Algorithm 1, their largest singular values i.e., $\sigma_1(\mathbf{A}_1), \sigma_1(\mathbf{B})$, satisfy the following condition: $0 < \max\{\sigma_1(\mathbf{A}), \sigma_1(\mathbf{B})\} \leq \sigma$, where $\sigma > 0$.*

Assumption 3 *1) There exists a constant $C > 0$ such that for any parameters $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{B}_1$, $\mathbf{B}_2$, $\|\mathbf{W}_1 - \mathbf{W}_2\|_F \leq C(\|\mathbf{A} - \mathbf{A}_2\|_F + \|\mathbf{B}_1 - \mathbf{B}_2\|_F)$; 2) the gradient of f on $\mathbf{W}$, $\nabla_{\mathbf{W}} f$, is bounded above by a constant $G > 0$, i.e., $\|\nabla_{\mathbf{W}} f\|_F \leq G$.*

Part 1 in Assumption 3 resembles a similar Triangle inequality among model parameters. We leverages two instances for justification. They are $\mathbf{A}_1 = \mathbf{A}_2$ and $\mathbf{B}_1 = \mathbf{B}_2$, respectively. In these two scenarios, either $\mathbf{A}$ or $\mathbf{B}$ is frozen. Then C can be quickly determined as $\sqrt{r}\sigma$ as for any matrix $\mathbf{Z} \in \mathbb{R}^{r\times k}$, $\|\mathbf{Z}\|_F = \sqrt{\sum_i \sigma_i^2(\mathbf{Z})} \leq \sqrt{r}\sigma_1(\mathbf{Z})$. In a generalized scenario, one can set a sufficiently

large constant to ensure the condition to hold, though it leads to a looser bound. Part 2 in Assumption 3 seems strong in this context. While it should be noted that though it has been relaxed in many existing works, for LLM fine-tuning, this is still required to make sure the objective loss f is also smooth with low-rank decomposition.

Lemma 1 *Let Assumptions 1, 2, 3 hold. Suppose that* $\mathbf{W}$ *satisfies MeFT, i.e.,* $\mathbf{W} = \mathbf{W}_0 + \mathbf{BA}$. *Then, we have the following relationship:*

$$\|\nabla f(\boldsymbol{v}_1, \boldsymbol{\theta}_0) - \nabla f(\boldsymbol{v}_2, \boldsymbol{\theta}_0)\| \leq (2LC\sqrt{r}\sigma + G)\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|, \tag{8}$$

for any given $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^m$, *where* $\|\cdot\|$ *is Euclidean norm.*

From Lemma 1, it can be observed that the loss f by using MeFT is *smooth* with a new constant $2LC\sqrt{r}\sigma + G$. Equivalently, $f(\boldsymbol{v}, \boldsymbol{\theta}_0)$ is $2LC\sqrt{r}\sigma + G$-smooth. Compared to the original smoothness constant L, with the low-rank decomposition and tuning both $\mathbf{A}$ and $\mathbf{B}$ yields a larger smoothness constant if $\sqrt{r} \geq \frac{1}{2C\sigma}$ (which can be satisfied empirically). This intuitively implies that the loss f with low-rank decomposition becomes less smooth, but instead the gradient may change faster. This can assist in convergence for the early phase of the optimization, particularly if the gradient is large at a point. Since a larger smoothness constant implies more dramatic changes in gradients, which benefits the gradient decaying. However, less smooth objective function may slow down the convergence as well, especially in the later phase, thus negatively affecting the convergence error.

Assumption 4 *There exists a constant* $\zeta > 0$ *such that* $\mathbb{E}[\|\boldsymbol{g} - \nabla f(\boldsymbol{v}, \boldsymbol{\theta}_0)\|^2] \leq \zeta^2$.

Theorem 2 *Let Assumptions 1 to 4 hold. Consider a sequence* $(\boldsymbol{v}_t)_{t\in\mathbb{N}}$ *generated by Algorithm 1, with a constant step size* $\alpha = \sqrt{\frac{D}{\hat{L}\zeta^2 T}}$, *where* $D = f(\boldsymbol{v}_1, \boldsymbol{\theta}_0) - f^*$, $\hat{L} = 2LC\sqrt{r}\sigma + G$. *Then for any* $T \geq \frac{4\hat{L}D}{\zeta^2}$, *it follows that*

$$\min_{t=1:T}\mathbb{E}[\|\nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2] \leq 2.5\sqrt{\frac{D\hat{L}\zeta^2}{T}}. \tag{9}$$

Convergence rate. Theorem 2 suggests that the convergence rate with MeFT is $\mathcal{O}(\sqrt{\frac{1}{T}})$, which resembles the best available rate [6]. The initialization error D and the variance of $\boldsymbol{g}$ also influence the error bound. Both a good initialization and a large batch are able to reduce the convergence error bound, making it reach an "approximate" critical point faster. Fine-tuning with large mini-batches aligns with the conclusion in [23], but this also practically slows down the convergence. Given an arbitrarily small constant $\varepsilon > 0$, we have that $T = \mathcal{O}(\varepsilon^{-2}) \Rightarrow \min_{t=1:T}\mathbb{E}[\|\nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2] = \mathcal{O}(\varepsilon)$. Substituting $L = 2LC\sqrt{r}\sigma + G$ into the conclusion of Theorem 2 implies that the convergence rate has a correlation with the rank r and the largest eigenvalue σ of low-rank decomposition matrices, i.e., $\min_{t=1:T}\mathbb{E}[\|\nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2] \leq \mathcal{O}\left(\frac{r^{1/4}\sigma^{1/2}}{T^{1/2}}\right)$. To the best of our knowledge, this is the first result to show how the convergence rate evolves with the "intrinsic rank" for a low-rank decomposition model efficient fine-tuning. We have also noticed that a recent result [38] reporting an information-theoretic generalization bound shows the similar relationship between the bound and the rank, i.e., $\mathcal{E} \propto \mathcal{O}(r^{1/2})$, where $\mathcal{E}$ signifies the generalization bound in this context. Preliminary results are presented to validate the relationship between the rank and the error bound in Appendix A.6.

Training vs. fine-tuning. Given the same initialization and variance, compared to training a model with full parameters, MeFT results in a worse convergence error bound. This intuitively makes sense since model updates have been approximated by using a low-rank decomposition, leading to the approximation error [25]. If r is extremely large, then T is required to be large as well to enable the convergence as more parameters are trainable. However, the convergence will be slow due to the relationship between α, T, and r. In this case, the error bound seems to be too pessimistic to explain the SGD performance. However, it should be noted that this result is for fine-tuning the model, instead of training from scratch entirely from a random initialization. Hence, D could be much smaller in this scenario, leading to a smaller convergence error bound for MeFT empirically. Another insight from the conclusion is that the learning rate has now become dependent on the model size, $\alpha \propto r^{-1/4}$ such that it may facilitate the LLM fine-tuning by allowing extrapolation from smaller models to large ones.

References

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

[2] Matan Avitan, Ryan Cotterell, Yoav Goldberg, and Shauli Ravfogel. What changed? converting representational interventions to natural language. *arXiv preprint arXiv:2402.11355*, 2024.

[3] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*, 2023.

[4] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

[5] Shangqian Gao, Ting Hua, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Adaptive rank selections for low-rank approximation of language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 227–241, 2024.

[6] Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

[7] Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pages 160–187. PMLR, 2024.

[8] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

[9] 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.

[10] 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.

[11] Yudong Huang, Hongyang Du, Xinyuan Zhang, Dusit Niyato, Jiawen Kang, Zehui Xiong, Shuo Wang, and Tao Huang. Large language models for networking: Applications, enabling techniques, and challenges. *IEEE Network*, 2024.

[12] Ahmed Khaled and Peter Richtárik. Better theory for sgd in the nonconvex world. *arXiv preprint arXiv:2002.03329*, 2020.

[13] Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki Markus Asano. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

[14] Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts. *arXiv preprint arXiv:2404.15159*, 2024.

[15] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[17] Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. Relora: High-rank training through low-rank updates. In *The Twelfth International Conference on Learning Representations*, 2024.

[18] Kaizhao Liang, Bo Liu, Lizhang Chen, and Qiang Liu. Memory-efficient llm training with online subspace descent. *arXiv preprint arXiv:2408.12857*, 2024.

[19] Sheng Liu, Lei Xing, and James Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*, 2023.

[20] 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.

[21] Zefang Liu and Jiahua Luo. Adamole: Fine-tuning large language models with adaptive mixture of low-rank adaptation experts. *arXiv preprint arXiv:2405.00361*, 2024.

[22] Zequan Liu, Jiawen Lyn, Wei Zhu, Xing Tian, and Yvette Graham. Alora: Allocating low-rank adaptation for fine-tuning large language models. *arXiv preprint arXiv:2403.16187*, 2024.

[23] Dang Nguyen, Wenhan Yang, Rathul Anand, Yu Yang, and Baharan Mirzasoleiman. Memory-efficient training of llms with larger mini-batches. *arXiv preprint arXiv:2407.19580*, 2024.

[24] Ipek Ozkaya. Application of large language models to software engineering tasks: Opportunities, risks, and implications. *IEEE Software*, 40(3):4–8, 2023.

[25] Hossein Rajabzadeh, Mojtaba Valipour, Tianshu Zhu, Marzieh Tahaei, Hyock Ju Kwon, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. Qdylora: Quantized dynamic low-rank adaptation for efficient large language model tuning. *arXiv preprint arXiv:2402.10462*, 2024.

[26] Shashwat Singh, Shauli Ravfogel, Jonathan Herzig, Roee Aharoni, Ryan Cotterell, and Ponnurangam Kumaraguru. Mimic: Minimally modified counterfactuals in the representation space. *arXiv preprint arXiv:2402.09631*, 2024.

[27] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

[28] 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.

[29] Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. *arXiv preprint arXiv:2205.12410*, 2022.

[30] Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*, 2024.

[31] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *arXiv preprint arXiv:2404.03592*, 2024.

[32] Xingyu Xie, Kuangyu Ding, Shuicheng Yan, Kim-Chuan Toh, and Tianwen Wei. Optimization hyper-parameter laws for large language models. *arXiv preprint arXiv:2409.04777*, 2024.

[33] Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*, 2023.

[34] Adam X Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. *arXiv preprint arXiv:2308.13111*, 2023.

[35] Rui Yang, Ting Fang Tan, Wei Lu, Arun James Thirunavukarasu, Daniel Shu Wei Ting, and Nan Liu. Large language models in health care: Development, applications, and challenges. *Health Care Science*, 2(4):255–263, 2023.

[36] Yifan Yang, Kai Zhen, Ershad Banijamal, Athanasios Mouchtaris, and Zheng Zhang. Adazeta: Adaptive zeroth-order tensor-train adaption for memory-efficient large language models fine-tuning. *arXiv preprint arXiv:2406.18060*, 2024.

[37] Changhai Zhou, Shijie Han, Shiyang Zhang, Shichao Weng, Zekai Liu, and Cheng Jin. Rankadaptor: Hierarchical dynamic low-rank adaptation for structural pruned llms. *arXiv preprint arXiv:2406.15734*, 2024.

[38] 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. *arXiv preprint arXiv:2402.16842*, 2024.

[39] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[40] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

Appendix A. Additional Analysis

In this section, we present additional analysis or missing proof in the main contents.

A.1. Methodology comparison

Table 1: Quantitative comparison among different methods

Method	Convergence	# of params
`Full model`	$\mathcal{O}(\frac{1}{\sqrt{T}})$	kd
`LoRA[10]`	N/A	$(d+k)r$
`DiReFT [31]`	N/A	$2dr$
`MeFT`	$\mathcal{O}(\sqrt{\frac{r^{1/2}}{T}})$	$2dr$ or $(k+d)r$

T: number of iterations; d: input dimension; k: output dimension; r: low rank;

A.2. Related Works

Parameter efficient fine-tuning (PEFT). In this context, we review related works on low-rank adaptation (LoRA) and refer interested reader in other PEFT methods such as adapter-based or prompt-based methods to [9, 16, 29]. Despite the fact that LoRA was just proposed in [10] a couple years ago, it has shown pronounced capabilities in fine-tuning LLMs, resulting in the more rapid applications of LLMs in various areas, e.g., medicine [27], networking [11], healthcare [35], and software engineering [24]. To continue reducing the computational overhead when fine-tuning LLMs, advanced LoRA variants have been proposed by manipulating low-rank matrices. Beyond the aforementioned works, Lialin et al. [17] developed ReLoRA which utilized low-rank updates to train high-rank networks by reducing the memory and improving the training speed. To address the issue of suboptimal performance when using vanilla LoRA with fixed rank, the authors from [37] introduced RankAdapter, which involves an end-to-end automatic optimization flow to determine different ranks during fine-tuning based on a light-weight performance model. This aims at reducing the performance gap caused by fixed rank. However, these two works only empirically evaluate their algorithms' performance without delivering any theoretical insights. Another recent work [36] focusing on memory-efficient LLM fine-tuning presented AdaZeta, leveraging the zeroth-order method to approximate gradients. Their analysis shows the explicit convergence rate, Nevertheless, the convergence error bound relies heavily on a quite complex constant, which could be difficult to determined in practice. Analogously, existing works [5, 25, 28] still pay more attention to the impact of dynamic ranks on the fine-tuning performance. More recently, LoRA was integrated with mixture of experts [14, 30] for improving the accuracy performance. The authors in [21] replaced a single LoRA in one layer with multiple LoRA experts, such that the most appropriate experts can be selected and activated based on the input context. Unfortunately, these works are still lack of theoretical foundations and haven't investigated the suitability of LoRA.

Representation fine-tuning (ReFT). ReFT is a new line of work which was developed recently [31] by switching the focus of fine-tuning from model parameters to hidden representations. They claimed that representations have encoded rich semantic information and suggested that editing representations may be a more powerful alternative to weight updates. The motivation originates from some recent works on activation steering and representation engineering [2, 15, 19, 26], showing that adding fixed or task-specific steering vectors or applying concept erasure to the residual stream ensures the decent adaptation of pre-trained LLMs to downstream tasks without computationally intensive fine-tuning. We briefly review these related works in this context for completeness. A technique called inference-time intervention [15] was developed to enhance the "truthfullness" of LLMs by shifting model activations during inference. Their resulting findings also suggested that LLMs may have an internal representation of the likelihood of something being true. To boost the transparency of AI systems, the authors in [40] called for the representation engineering comprising representation reading and representation control, which turned the action from model parameters to hidden representations. A recent work [26] defined the so-called representation-space counterfactuals and unified the linear erasure and steering vector interventions together by proposing an approach called minimally modified counterfactuals, which successfully mitigates bias in multi-class classification and reduces the generation of toxic language. However, the representation-space counterfactuals can be converted to natural language counterfactuals [2] for analyzing the linguistic alterations corresponding to a given representation-space intervention and interpreting the features utilized for encoding a specific concept. Additionally, interventional interpretability [31] has been another incentive for ReFT, implying that interventions on linear subspaces of representations evidently show that human-interpretable concepts are encoded linearly. Regardless of the convincing performance by ReFT, we are not aware of any reported results on the provable guarantees, thus resulting in a theoretical gap.

Convergence for LLMs. Distinct convergence rate for LLM fine-tuning is a fairly underexplored topic, particularly when a vast majority of works tends to highlight the astonishing empirical performance of LLMs. Only a couple works have attempted to address this issue, including [18] and [23]. Liang et al. [18] presented a memory-efficient LLM training method called online subspace descent and provided the convergence guarantee, instead of an explicit rate. Similarly, another work [23] advocated the adoption of large mini-batches for training LLMs, where they arrived at the same convergence rate as in [8]. However, this work is not applicable to low-rank decomposition matrix fine-tuning. A concurrent work [32] proposed Opt-Laws to capture the relationship between hyper-parameters and training outcomes by leveraging stochastic differential equations, retaining the sub-linear convergence rate of training LLMs. Though their analysis does not apply to fine-tuning, the convergence theoretically justifies that the error bound is proportional to the model size.

A.3. Schematic Diagram

Figure 1 shows the schematic diagrams of both LoRA and MeFT. Though what they fine-tune is different, the updates comply with each other. The interventions to hidden representations in low-dimensional space in ReFT can resemble the low-rank matrix adaption in LoRA.

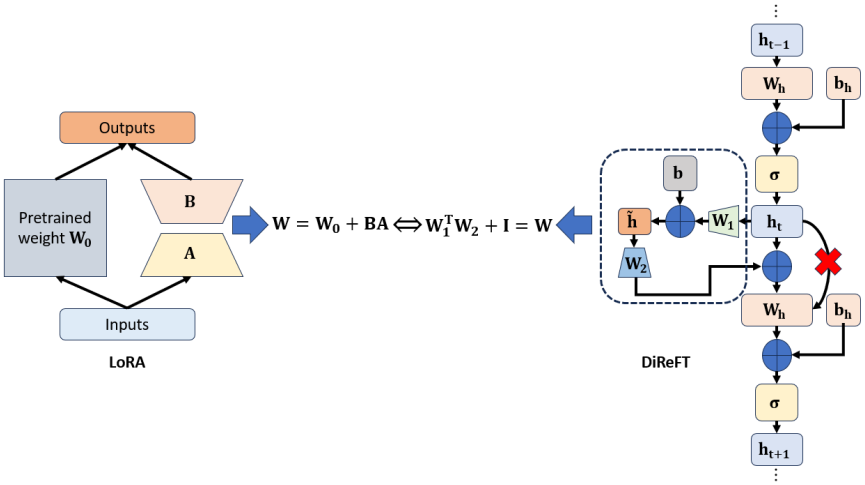


Figure 1: The operational equivalence between LoRA (left side) and ReFT (right side). A unified matrix operation can be applied to either parameters for LoRA or representations for ReFT.

A.4. Proof for Lemma 1

Lemma 1 Let Assumptions 1, 2, 3 hold. Suppose that $\mathbf{W}$ satisfies MeFT, i.e., $\mathbf{W} = \mathbf{W}_0 + \mathbf{BA}$. Then, we have the following relationship:

$$\|\nabla f(\boldsymbol{v}_1, \boldsymbol{\theta}_0) - \nabla f(\boldsymbol{v}_2, \boldsymbol{\theta}_0)\| \leq (2LC\sqrt{r}\sigma + G)\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|, \tag{10}$$

for any given $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^m$.

Proof Recall the gradient of f on $\mathbf{W}$, $\nabla_{\mathbf{W}} f$. Then we can obtain the gradient on $\boldsymbol{v}$ being $[\nabla_{\mathbf{A}} f(\mathbf{W}), \nabla_{\mathbf{B}} f(\mathbf{W})]$. Based on the chain rule, we have $\nabla_{\mathbf{A}} f(\mathbf{W}) = \mathbf{B}^\top \nabla_{\mathbf{W}} f$ and $\nabla_{\mathbf{B}} f(\mathbf{W}) = \nabla_{\mathbf{W}} f \mathbf{A}^\top$. Given this and the fact that $f(\mathbf{W}_i) := f(\mathbf{W}_0 + \mathbf{B}_i\mathbf{A}_i)$, $i = 1, 2$, we have the following relationship

$$\begin{aligned}
&\frac{\|\nabla f(\boldsymbol{v}_1, \mathbf{W}_0) - \nabla f(\boldsymbol{v}_2, \mathbf{W}_0)\|_F}{\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_F} \\
&= \frac{\|\mathbf{B}_1^\top \nabla_{\mathbf{W}_1} f - \mathbf{B}_2^\top \nabla_{\mathbf{W}_2} f\|_F + \|\nabla_{\mathbf{W}_1} f \mathbf{A}_1^\top - \nabla_{\mathbf{W}_2} f \mathbf{A}_2^\top\|_F}{\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_F} \\
&= \frac{\|\mathbf{B}_1^\top \nabla_{\mathbf{W}_1} f - \mathbf{B}_1^\top \nabla_{\mathbf{W}_2} f + \mathbf{B}_1^\top \nabla_{\mathbf{W}_2} f - \mathbf{B}_2^\top \nabla_{\mathbf{W}_2} f\|_F}{\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_F} \\
&+ \frac{\|\nabla_{\mathbf{W}_1} f \mathbf{A}_1^\top - \nabla_{\mathbf{W}_2} f \mathbf{A}_1^\top + \nabla_{\mathbf{W}_2} f \mathbf{A}_1^\top - \nabla_{\mathbf{W}_2} f \mathbf{A}_2^\top\|_F}{\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_F}
\end{aligned} \tag{11}$$

As

$$\mathbf{B}_1^\top \nabla_{\mathbf{W}_1} f - \mathbf{B}_1^\top \nabla_{\mathbf{W}_2} f + \mathbf{B}_1^\top \nabla_{\mathbf{W}_2} f - \mathbf{B}_2^\top \nabla_{\mathbf{W}_2} f = \mathbf{B}_1^\top (\nabla_{\mathbf{W}_1} f - \nabla_{\mathbf{W}_2} f) + (\mathbf{B}_1^\top - \mathbf{B}_2^\top) \nabla_{\mathbf{W}_2} f$$

According to Triangle inequality and Cauchy-Schwartz inequality, we can obtain

$$\begin{aligned}
&\|\mathbf{B}_1^\top \nabla_{\mathbf{W}_1} f - \mathbf{B}_1^\top \nabla_{\mathbf{W}_2} f + \mathbf{B}_1^\top \nabla_{\mathbf{W}_2} f - \mathbf{B}_2^\top \nabla_{\mathbf{W}_2} f\|_F \\
&\leq \|\mathbf{B}_1^\top (\nabla_{\mathbf{W}_1} f - \nabla_{\mathbf{W}_2} f)\|_F + \|(\mathbf{B}_1^\top - \mathbf{B}_2^\top) \nabla_{\mathbf{W}_2} f\|_F \\
&\leq \|\mathbf{B}_1^\top\|_F \|\nabla_{\mathbf{W}_1} f - \nabla_{\mathbf{W}_2} f\|_F + \|\mathbf{B}_1^\top - \mathbf{B}_2^\top\|_F \|\nabla_{\mathbf{W}_2} f\|_F.
\end{aligned} \tag{12}$$

Similarly, we have $\|\nabla_{\mathbf{W}_1} f\mathbf{A}_1^\top - \nabla_{\mathbf{W}_2} f\mathbf{A}_1^\top + \nabla_{\mathbf{W}_2} f\mathbf{A}_1^\top - \nabla_{\mathbf{W}_2} f\mathbf{A}_2^\top\|_F \leq \|\nabla_{\mathbf{W}_1} f - \nabla_{\mathbf{W}_2} f\|_F \|\mathbf{A}_1^\top\|_F + \|\nabla_{\mathbf{W}_2} f\|_F \|\mathbf{A}_1^\top - \mathbf{A}_2^\top\|_F$. With these relationship in hand, we have

$$\begin{aligned}
&\frac{\|\nabla f(\boldsymbol{v}_1, \mathbf{W}_0) - \nabla f(\boldsymbol{v}_2, \mathbf{W}_0)\|_F}{\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_F} \\
&\leq \frac{\|\nabla_{\mathbf{W}_1} f - \nabla_{\mathbf{W}_2} f\|_F (\|\mathbf{A}_1^\top\|_F + \|\mathbf{B}_1^\top\|_F) + \|\nabla_{\mathbf{W}_2} f\|_F (\|\mathbf{A}_1^\top - \mathbf{A}_2^\top\|_F + \|\mathbf{B}_1^\top - \mathbf{B}_2^\top\|_F)}{\|\mathbf{A}_1 - \mathbf{A}_2\|_F + \|\mathbf{B}_1 - \mathbf{B}_2\|_F} \\
&\leq \frac{L\|\mathbf{W}_1 - \mathbf{W}_2\|_F (\|\mathbf{A}_1^\top\|_F + \|\mathbf{B}_1^\top\|_F)}{\|\mathbf{A}_1 - \mathbf{A}_2\|_F + \|\mathbf{B}_1 - \mathbf{B}_2\|_F} + G \\
&\leq 2LC\sqrt{r}\sigma + G,
\end{aligned} \tag{13}$$

where the second inequality follows from Assumption 1 and the third inequality follows from Assumption 2 and Assumption 3. This completes the proof. ■

A.5. Proof for Theorem 2

Theorem 2 Let Assumptions 1 to 4 hold. Consider a sequence $(\boldsymbol{v}_t)_{t\in\mathbb{N}}$ generated by Algorithm 1, with a constant step size $\alpha = \sqrt{\frac{D}{\hat{L}\zeta^2 T}}$, where $D = f(\boldsymbol{v}_1, \boldsymbol{\theta}_0) - f^*$, $\hat{L} = 2LC\sqrt{r}\sigma + G$. Then for any $T \geq \frac{4\hat{L}D}{\zeta^2}$, it follows that

$$\min_{t=1:T} \mathbb{E}[\|\nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2] \leq 2.5\sqrt{\frac{D\hat{L}\zeta^2}{T}}. \tag{14}$$

Proof In light of the smoothness assumption, we can obtain the following relationship

$$f(\boldsymbol{v}_{t+1}, \boldsymbol{\theta}_0) - f(\boldsymbol{v}_t, \boldsymbol{\theta}_0) - \langle \nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0), \boldsymbol{v}_{t+1} - \boldsymbol{v}_t \rangle \leq \frac{\hat{L}}{2}\|\boldsymbol{v}_{t+1} - \boldsymbol{v}_t\|^2. \tag{15}$$

Substituting the update law into the last equation yields the following relationship:

$$f(\boldsymbol{v}_{t+1}, \boldsymbol{\theta}_0) - f(\boldsymbol{v}_t, \boldsymbol{\theta}_0) + \alpha\langle \nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0), \boldsymbol{g}_t \rangle \leq \frac{\hat{L}}{2}\alpha^2\|\boldsymbol{g}_t\|^2. \tag{16}$$

The right hand side of Eq. 16 can be rewritten as

$$\frac{\hat{L}}{2}\alpha^2\|\boldsymbol{g}_t\|^2 = \frac{\hat{L}}{2}\alpha^2\|\boldsymbol{g}_t - \nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0) + \nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2, \tag{17}$$

which leads to the following relationship

$$\frac{\hat{L}}{2}\alpha^2\|\boldsymbol{g}_t\|^2 \leq \hat{L}\alpha^2(\|\boldsymbol{g}_t - \nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2 + \|\nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2) \tag{18}$$

It follows from a basic inequality $\|\boldsymbol{a} + \boldsymbol{b}\|^2 \leq 2\|\boldsymbol{a}\|^2 + 2\|\boldsymbol{b}\|^2$. Substituting Eq. 18 to Eq. 16 and taking expectation on both sides produces:

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{v}_{t+1}, \boldsymbol{\theta}_0) - f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)] + \alpha\mathbb{E}[\|\nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2] \leq \\
&\hat{L}\alpha^2\mathbb{E}[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2 + \|\nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2].
\end{aligned} \tag{19}$$

By leveraging Assumption 4, Eq. 19 becomes

$$\mathbb{E}[f(\boldsymbol{v}_{t+1},\boldsymbol{\theta}_0)-f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)]+\alpha\mathbb{E}[\|\nabla f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)\|^2]\leq\hat{L}\alpha^2\zeta^2+\hat{L}\alpha^2\mathbb{E}[\|\nabla f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)\|^2] \tag{20}$$

As the step size $\alpha=\sqrt{\frac{D}{\hat{L}\zeta^2T}}$ and $T\geq\frac{4\hat{L}D}{\zeta^2}$, we can obtain that $\alpha\leq\frac{1}{2\hat{L}}$. We then have:

$$\mathbb{E}[f(\boldsymbol{v}_{t+1},\boldsymbol{\theta}_0)-f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)]\leq\hat{L}\alpha^2\zeta^2-\frac{\alpha}{2}\mathbb{E}[\|\nabla f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)\|^2]. \tag{21}$$

With some simple mathematical manipulations, the following inequality is obtained:

$$\mathbb{E}[\|\nabla f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)\|^2]\leq\frac{2(\mathbb{E}[f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)-f(\boldsymbol{v}_{t+1},\boldsymbol{\theta}_0)])}{\alpha}+\frac{\hat{L}\alpha\zeta^2}{2}. \tag{22}$$

Now we sum the above equation over 1 to T such that

$$\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)\|^2]\leq\frac{2(f(\boldsymbol{v}_1,\boldsymbol{\theta}_0)-f^*)}{\alpha}+\frac{T\hat{L}\alpha\zeta^2}{2}. \tag{23}$$

Dividing both sides by T, we have:

$$\min_{t=1:T}\mathbb{E}[\|\nabla f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)\|^2]\leq\frac{\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\boldsymbol{v}_t,\boldsymbol{\theta}_0)\|^2]}{T}\leq\frac{2D}{\alpha T}+\frac{\hat{L}\alpha\zeta^2}{2}. \tag{24}$$

Substituting the step size $\alpha=\sqrt{\frac{D}{\hat{L}\zeta^2T}}$ into the last inequality yields the desirable result. ■

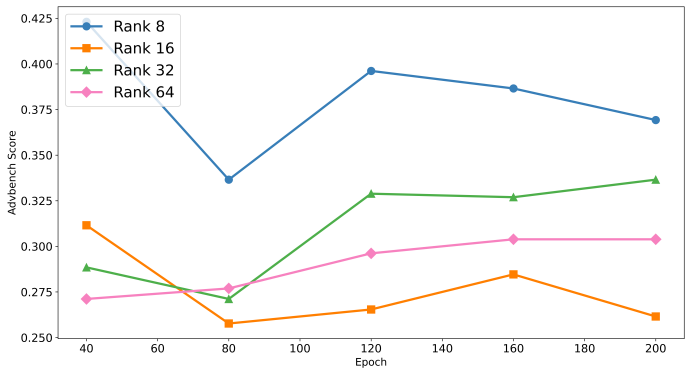
A.6. Experimental Results

In this study, we employed the `Llama-2-7b-hf` model with the `Alpaca-GPT4` dataset. Due to the extensive training time associated with LoRA (Low-Rank Adaptation), we utilized only a subset of the dataset to ensure computational efficiency without compromising the validity of our results. The model's performance was evaluated using two benchmark suites: `MT-Bench` and `AdvBench`, which assess the model across a range of tasks, providing a holistic view of its strengths and limitations. We specifically analyzed the effects of varying LoRA ranks (8, 16, 32, 64) on model performance, while maintaining a fixed batch size of 32 to balance memory usage and training stability.

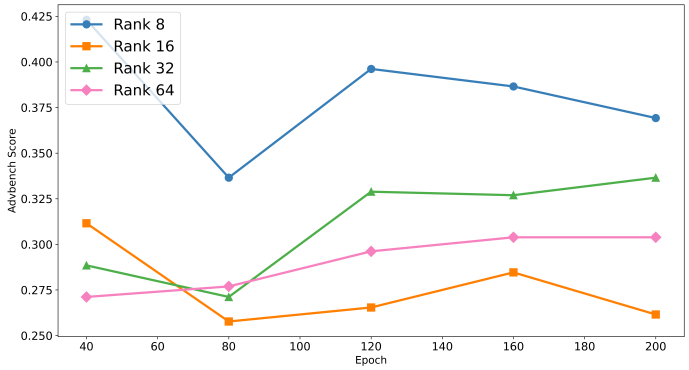
Evaluation Metrics:

- `MT-Bench`: Evaluates the model's machine translation capabilities, focusing on its ability to produce accurate and fluent translations across diverse languages.
- `AdvBench`: Assesses the model's robustness against adversarial inputs, testing its capacity to handle challenging scenarios with reliability and stability.

Due to the computationally intensive nature of the evaluations, results were collected from checkpoints taken between 40 and 200 epochs, which were randomly selected to provide a representative sample of the model's performance throughout training.



(*a*)



(*b*)

Figure 2: (a) Performance (the higher, the better) of the `meta-llama/Llama-2-7b-hf` model on `MT-Bench` using different LoRA ranks (8, 16, 32, 64) with a batch size of 32. Checkpoints from 40 to 200 epochs were randomly selected; (b) Performance (the higher, the better) of the `meta-llama/Llama-2-7b-hf` model on `AdvBench` using different LoRA ranks (8, 16, 32, 64) with a batch size of 32. Checkpoints from 40 to 200 epochs were randomly selected.

Figure 2 (a) demonstrates the model's performance on `MT-Bench`, illustrating the impact of different LoRA ranks on translation accuracy. Lower ranks generally yield better performance while also reducing computational cost.
Figure 2 (b) highlights the model's resilience in the face of adversarial inputs, as measured by `AdvBench`. Models with lower LoRA ranks tend to exhibit greater robustness, providing improved stability and reliability under challenging conditions.
Our experiments demonstrate that lower LoRA ranks not only result in better performance on both benchmarks, but also require fewer computational resources and shorter training times. These findings support our theoretical results ($\min_{t=1:T}\mathbb{E}[\|\nabla f(\boldsymbol{v}_t, \boldsymbol{\theta}_0)\|^2] \leq \mathcal{O}\left(\frac{r^{1/4}\sigma^{1/2}}{T^{1/2}}\right)$) and underscore the significance of choosing the right LoRA rank based on the specific goals and constraints of the deployment environment for practical application. We remark that the results shown here are preliminary and more extensive results will be presented in our future work for the thorough validation.