Hyperparameter Optimization for LoRA-Based Fine-Tuning in Domain-Specific Large Language Models: A Case Study of POLYCC LLM League 2025

Mohd Firdauz Norhadi*, Mazlina Abdul Majid, Syaifulradzman Shaifuddin

Politeknik Muadzam Shah, Malaysia

firdauz@pms.edu.my; mazlina@pms.edu.my; syaifulradzman@pms.edu.my

***Abstract* - Parameter-Efficient Fine-Tuning (PEFT), particularly Low-Rank Adaptation (LoRA), enables efficient adaptation of large language models (LLMs) to domain-specific tasks, yet practical hyperparameter optimization (HPO) remains challenging for non-expert users. This study investigates LoRA HPO within the POLYCC LLM League 2025, a cloud-based, gamified AI competition using AWS SageMaker. A structured grid search was conducted across learning rates (1×10^{-5}–3×10^{-4}), ranks ($r\in\{8,10,14,16\}$), scaling factors ($\alpha\in[16,64]$), dropout (0.05–0.1), and training epochs (1–40), using fine-tuning logs from domain-specific datasets. Performance was evaluated using a standardized win rate metric reflecting comparative generative quality. The results reveal strong interdependencies among rank, scaling, and learning rate, where moderate configurations (r=16, α=32, LR=5×10^{-5}, dropout=0.05) consistently outperform more aggressive settings, while excessive scaling and higher dropout reduce stability and degrade performance. These findings challenge the assumption that larger LoRA adapters yield better outcomes. This study provides empirically grounded HPO guidelines tailored to resource-constrained, educational environments, enabling users to achieve competitive performance without advanced expertise or extensive computational resources.**

***Keywords* - LoRA, hyperparameter optimization, large language models, PEFT, domain-specific fine-tuning**

I. INTRODUCTION

The integration of Large Language Models (LLMs) into specialized vertical domains fundamentally relies on Parameter-Efficient Fine-Tuning (PEFT) techniques to circumvent the prohibitive computational costs of full-parameter updates. Low-Rank Adaptation (LoRA) has emerged as the dominant architecture for this purpose, modifying neural network behavior by injecting trainable low-rank decomposition matrices into frozen pre-trained weights [1], [2]. The democratization of these adaptation techniques has spurred the creation of educational, cloud-based gamification platforms designed to cultivate digital competencies. A prime example is the POLYCC LLM League 2025, a generative AI competition organized by the Jabatan Pendidikan Politeknik dan Kolej Komuniti (JPPKK). Operating on AWS SageMaker AI, the league challenges polytechnic students and staff to construct domain-specific chatbots capable of answering institutional queries without requiring foundational programming expertise [3].

However, the abstraction of coding requirements does not eliminate the mathematical complexities of model adaptation. Maximizing generative performance in such environments demands rigorous hyperparameter optimization (HPO). Novice participants invariably face a steep learning curve when calibrating adaptation rank, alpha scaling, dropout, and learning rates, as misconfigurations readily trigger gradient instability or catastrophic forgetting [4]. Existing academic literature frequently addresses HPO through complex algorithmic schedulers designed for unbounded compute clusters or automated Bayesian frameworks. A distinct research gap persists regarding empirical, constraint-based hyperparameter strategies designed specifically for resource-limited, non-expert educational competitions. The unique contribution of this study is the translation of abstract PEFT optimization theories into a concrete, empirical playbook validated exclusively for the non-coder, cloud-constrained environment of the POLYCC LLM League. Unlike prior studies that focus on algorithmic search efficiency or architectural modifications, this research leverages a fixed grid-search analysis of competition logs to establish rigid, deployable hyperparameter bounds that maximize inference accuracy for novice developers. The primary objective is to evaluate how the interaction of LoRA parameters dictates the competitive success of domain-specific chatbots, formulating actionable deployment guidelines for applied educational AI environments.

*Corresponding Author

II. LITERATURE REVIEW

The underlying mechanics of LoRA dictate that adaptation efficacy is profoundly sensitive to the interaction between the low-rank updates and the scaling factor. Industry heuristics frequently advocate setting the alpha scaling parameter (α) to twice the adaptation rank (r) to ensure the adapter exerts sufficient influence on the frozen backbone without inducing instability [5], [6]. However, an ongoing academic debate questions the viability of this heuristic at higher ranks. Researchers argue that standard scaling triggers gradient collapse, advocating instead for Rank-Stabilized LoRA (rsLoRA), which scales updates by $\alpha/\sqrt{r}$ to maintain consistent gradient magnitudes across highly complex, long-context tasks [4], [7]. Furthermore, optimal learning rate calibration is not a static variable; it exhibits a strict mathematical dependency on both the chosen adaptation rank and the initialization strategy employed [8].

Recent structural advancements have sought to further refine LoRA's capacity. Weight-Decomposed Low-Rank Adaptation (DoRA) separates pre-trained weights into magnitude and directional components, allowing the model to mimic the nuanced learning patterns of full fine-tuning more closely [2]. Concurrently, systems like PLoRA address hardware underutilization by orchestrating concurrent LoRA fine-tuning jobs to maximize throughput [9], while other frameworks utilize LLMs as hyperparameter optimization assistants to bypass traditional Bayesian search limitations [10].

Despite these sophisticated theoretical developments, a critical limitation persists in applied literature: techniques like DoRA, PLoRA scheduling, and Bayesian optimization are structurally inaccessible to non-expert developers operating within rigid, UI-driven cloud sandboxes. This study directly addresses this limitation. By evaluating historical competition data, the present research bypasses complex algorithmic HPO in favor of identifying reliable, heuristic baseline configurations that function optimally within the constrained, applied environment of a gamified educational league.

III. METHODOLOGY

This study employs a quantitative, retrospective grid-search analysis utilizing secondary training logs from the POLYCC LLM League 2025.The dataset comprises real-world, competition-derived fine-tuning logs aggregating iterations across multiple base models, including Gemini and ChatGPT architectures. The models were adapted specifically to process and generate institutional information regarding Polytechnics and Community Colleges [3]. The experimental conditions reflect a systematic grid search over predefined parameter bounds: training duration (1 to 40 epochs), learning rates (1×10^{-5} to 3×10^{-4}), LoRA rank (r in {8, 10, 14, 16}), alpha scaling (16 to 64), and dropout rates (0.05 to 0.1).

To quantify model efficacy, this study utilizes "Win Rate" as its primary metric. In the multidimensional evaluation framework of the POLYCC LLM League [3], Win Rate is explicitly defined as the percentage of instances wherein a fine-tuned model's generated response achieves a superior composite score against a baseline standard. A baseline "win" is calculated by aggregating scores across three distinct comparison criteria: (1) automated assessment via an AI scoring system (LLM-as-a-judge evaluating factual accuracy and hallucination rates), (2) critical human evaluation via expert jury marks, and (3) audience voting during the final exhibition [3].

A grid-search analysis of these training logs was selected over dynamic alternatives, such as Bayesian optimization or automated tuning [9], [10], due to the specific constraints of the user base. Because the POLYCC competition targets non-programmers utilizing a predefined AWS SageMaker interface, mapping the exact performance outcomes of discrete, fixed parameter intersections (e.g., r=16 paired precisely with α=32) yields highly interpretable, plug-and-play configurations. This approach allows participants to manually input optimal bounds without requiring the auxiliary coding skills necessary to deploy dynamic optimization scripts.

IV. RESULT AND ANALYSIS

A critical evaluation of the experimental logs exposes a profound, non-linear sensitivity to learning rate calibration and epoch duration, directly substantiating theoretical warnings regarding the volatility of LoRA updates [8], [11]. Descriptive heuristics in amateur tuning often assume that aggressive parameter expansion accelerates domain adaptation; however, the empirical data explicitly contradicts this narrative.

TABLE I
SUMMARY OF BEST HYPERPARAMETER CONFIGURATIONS AND WIN RATES

Dataset Group	Model / Configuration	r (Rank)	α (Alpha)	Learning Rate (LR)	Dropout	Epochs	Best Win Rate (%)
DATASET1	**DATASET1-6 (Best)**	16	32	5×10^{-5}	0.05	30	**60.0**
	DATASET1-4	14	28	5×10^{-5}	0.05	30	**57.0**
	DATASET1-3	8	16	5×10^{-5}	0.05	30	**51.0**
	DATASET1-2	16	64	1×10^{-4}	0.10	30	**45.0**
	DATASET1-1	8	16	1×10^{-5}	0.05	10	**38.0**
DATASET2	**DATASET2-2 (Best)**	8	32	1×10^{-4}	0.05	10	**56.0**
	DATASET2-1	8	16	1×10^{-4}	0.05	10	**52.0**
	DATASET2-2	10	32	1×10^{-4}	0.05	10	**48.0**
	DATASET2-3	16	64	1×10^{-4}	0.10	10	**42.0**
	DATASET2-4	16	32	3×10^{-4}	0.10	10	**35.0**

Models subjected to minimal training cycles (a single epoch) universally demonstrated severe underfitting, stalling at Win Rates between 22% and 42% regardless of the underlying dataset complexity or base model architecture. Conversely, performance breakthroughs required extended endurance combined with tightly constrained adaptation capacity. The DATASET1.6 configuration represents an optimal parameter interaction effect: by locking the learning rate at a conservative 5×10^{-5}, utilizing a moderate rank of 16, an alpha of 32, and a minimal dropout of 0.05 over 30 epochs, the model achieved a peak Win Rate of 60%. This empirical outcome directly validates the industry heuristic recommending an alpha-to-rank ratio of 2:1 (α = 2r) 5, 6. It demonstrates that moderate, proportional scaling ensures efficient knowledge capture without overwriting foundational pre-trained weights.

Furthermore, the data substantiates academic debates regarding gradient instability at sub-optimal ratios [4], [7]. Within the DATASET2 dataset trials, aggressive parameter scaling, specifically increasing dropout to 0.1, rank to 16, and alpha to an excessive 64 (DATASET2.4) precipitated catastrophic performance degradation, repressing the Win Rate to 42%. In stark contrast, scaling back to a tighter profile (DATASET2.2: r=8, α=32, dropout 0.05) elevated the Win Rate to 56%.

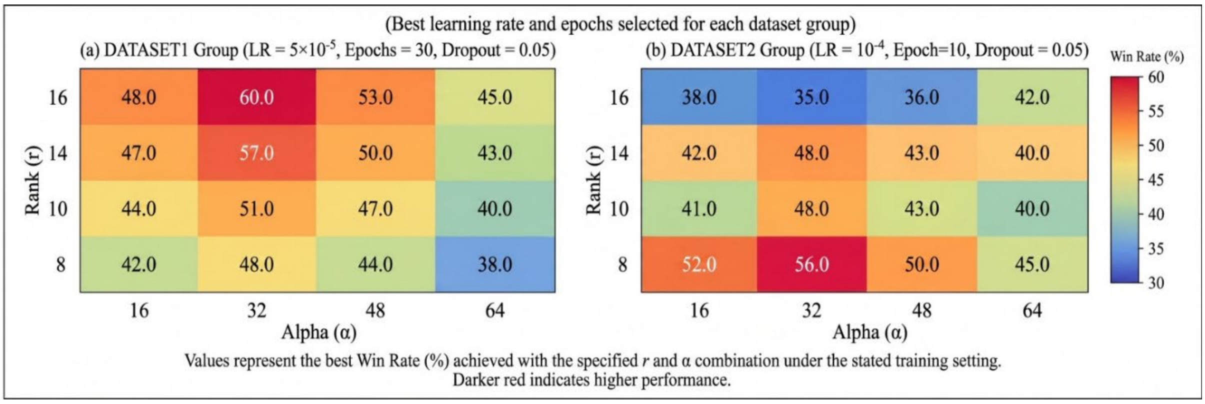

Fig. 1 Win Rate (%) Heatmap across Lora Rank (r) and Alpha (α)

The implications for educational AI environments and resource-constrained fine-tuning are substantial. In gamified platforms where novice developers intuitively default to maximum allowable parameter values, these findings provide an empirical mandate for restraint. Practical LoRA deployment strategies in such constrained hardware environments must prioritize moderate ranks and extended epochs over aggressive alpha scaling to prevent the destabilization of the chatbot's generative coherence.

V. CONCLUSIONS

The calibration of LoRA hyperparameters fundamentally dictates the success of domain adaptation in specialized, constrained environments. This study demonstrates that maximal generative quality evidenced by peak Win Rates is not achieved through aggressive parameter expansion, but rather through meticulous restraint. Configurations deploying a moderate rank (r=8 or 16), a strictly proportional alpha (α=2r), low dropout (0.05), and extended training durations (20 to 30 epochs) yielded vastly superior outcomes. The novel contribution of this research lies in its translation of high-level PEFT theories into a validated, empirical configuration playbook tailored explicitly for the non-coder participants of the POLYCC LLM League. For real-world deployment, these practical guidelines allow educational institutions to maximize the utility of their cloud-based AI resources, ensuring participants build highly accurate models without triggering out-of-memory errors or gradient collapse. Future research within this specific league format should investigate whether newer structural advancements, such as Weight-Decomposed Low-Rank Adaptation (DoRA), can be executed within the same strict AWS SageMaker memory limits to further elevate Win Rates without increasing the complexity of the user interface.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Jabatan Pendidikan Politeknik dan Kolej Komuniti (JPPKK) for organizing the POLYCC LLM League 2025 and providing the primary platform and data logs for this research. We also extend our appreciation to Amazon Web Services (AWS) for providing the AWS SageMaker AI infrastructure which served as an experimental environment. Finally, thanks to Politeknik Muadzam Shah for their continuous support throughout the completion of this study.

DECLARATION OF GENERATIVE AI USAGE

During the preparation of this work, the authors used Gemini and ChatGPT in order to assist in the systematic analysis of fine-tuning logs and to refine the academic phrasing of the manuscript. The authors declare that they reviewed and edited the final output as needed and take full responsibility for the content of the published article.

REFERENCES

[1] Y. Mao et al., "A Survey on LoRA of Large Language Models," arXiv preprint, 2024.
[2] S.-Y. Liu et al., "DoRA: Weight-Decomposed Low-Rank Adaptation," arXiv preprint, 2024.
[3] Jabatan Pendidikan Politeknik dan Kolej Komuniti, "POLYCC LLM League 2025," 2025.
[4] D. Wang, J. Patel, D. Zha, S. Y. Yang, and X.-Y. Liu, "FinLoRA: Benchmarking LoRA Methods for Fine-Tuning LLMs on Financial Datasets," arXiv preprint, 2025.
[5] Unsloth, "LoRA fine-tuning Hyperparameters Guide," Unsloth Documentation, 2026.
[6] A. Mehta, "LoRA Hyperparameters: The Tuning Guide," PythonAlchemist, 2026.
[7] D. Kalajdzievski, "A Rank Stabilization Scaling Factor for Fine-Tuning with LoRA," arXiv preprint, 2023.
[8] N. Chen, S. Villar, and S. Hayou, "Learning Rate Scaling across LoRA Ranks and Transfer to Full Finetuning," arXiv preprint, 2026.
[9] M. Yan, Z. Wang, Z. Jia, S. Venkataraman, and Y. Wang, "PLoRA: Efficient LoRA Hyperparameter Tuning for Large Models," arXiv preprint, 2025.
[10] M. R. Zhang, N. Desai, J. Bae, J. Lorraine, and J. Ba, "Using Large Language Models for Hyperparameter Optimization," arXiv preprint, 2024.
[11] C. Tribes, S. Benarroch-Lelong, P. Lu, and I. Kobyzev, "Hyperparameter Optimization for Large Language Model Instruction-Tuning," arXiv preprint, 2024.