A Language-Guided Bayesian Optimization for Efficient LoRA Hyperparameter Search

Anonymous Authors[1]

Abstract

Fine-tuning Large Language Models (LLMs) with Low-Rank Adaptation (LoRA) is resource-efficient, but its performance is highly sensitive to hyperparameter choices, making exhaustive search expensive. To address this, we propose a framework that integrates pre-trained LLM knowledge into Bayesian Optimization (BO) for efficient LoRA hyperparameter optimization. Our method uses an LLM as a discrete-to-continuous mapping module that converts hyperparameter configurations and domain-aware prompts into continuous embeddings, where BO is performed. The prompts describe the roles and relationships of LoRA hyperparameters, while an additional learnable token captures information not easily expressed in text. We further introduce proxy evaluation on a data subset, exploiting its strong correlation with full-data training to reduce evaluation cost. Experiments show that our method finds strong hyperparameters within about 30 iterations, achieving over 20% improvement over standard hyperparameters selected from roughly 45,000 combinations.

1. Introduction

Large Language Models (LLMs) have become strong foundation models that can be adapted to diverse downstream tasks. However, full fine-tuning is computationally expensive because it requires updating billions of parameters (Brown et al., 2020). Parameter-Efficient Fine-Tuning (PEFT) methods address this issue by adapting models with substantially fewer trainable parameters. Among them, Low-Rank Adaptation (LoRA) (Hu et al., 2022) is widely adopted, as it freezes pre-trained weights and introduces lightweight trainable low-rank adapters for efficient task adaptation. Despite its efficiency, LoRA performance is highly sensitive to hyperparameter choices (Sengupta et al., 2024; Biderman et al., 2024; Mao et al., 2025). Since the search space is large and each evaluation requires costly LLM fine-tuning, exhaustive or naïve search is impractical (Sun et al., 2024; Meo et al., 2024; Bini et al., 2025). This motivates the need for a LoRA-specific and sample-efficient hyperparameter optimization framework.

Bayesian optimization (BO) is well suited for LoRA HPO because it can optimize expensive black-box functions with relatively few evaluations (Korovina et al., 2020; Ranković & Schwaller, 2023; 2025). However, applying standard BO to LoRA is not straightforward. LoRA hyperparameters are discrete and structured, whereas BO is typically designed for continuous, smooth spaces. In addition, standard BO does not naturally incorporate domain knowledge (Yan et al., 2025), and its surrogate model can be unreliable when only a few observations are available.

To address these challenges, we propose an efficient BO-based HPO framework tailored to LoRA. Our method uses an LLM to encode each hyperparameter configuration as a structured text template that describes its names, values, roles, and interactions. The template is combined with a learnable token and mapped into a continuous embedding space, where a surrogate model guides BO toward promising configurations. We also use proxy training evaluation to reduce the cost of each evaluation.

In summary, our contributions are as follows:

- **LLM-guided BO for LoRA HPO.** We propose the first framework that combines an LLM with BO for LoRA hyperparameter optimization, enabling domain-aware and sample-efficient search.
- **Efficient optimization design.** We introduce structured text representations, a learnable token, a projection layer, and a proxy evaluation protocol to improve optimization efficiency and reduce computational cost.
- **Broad empirical validation.** We demonstrate consistent improvements across LoRA variants and model architectures, showing the effectiveness and generalizability of the proposed framework.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Related Work

Low-Rank Adaptation (LoRA) and hyperparameter sensitivity in LoRA. LoRA (Hu et al., 2022) is one of the most widely used parameter-efficient fine-tuning methods for Large Language Models (LLMs), enabling task-specific adaptation by adding trainable low-rank adapters to frozen pre-trained models. Numerous variants (Liu et al., 2024b; Kalajdzievski, 2023; Meng et al., 2024) have been proposed to improve LoRA's stability, convergence, and performance. Despite these advances, LoRA performance remains highly sensitive to hyperparameters such as rank, scaling factor, learning rate, batch size, and dropout rate (Zhang et al., 2024; Liu et al., 2025; Jin et al., 2023; Marek et al., 2025). Since effective configurations often depend on the dataset and base model (Rajabzadeh et al., 2024; Yan et al., 2025), systematic and efficient hyperparameter optimization for LoRA remains an important but underexplored problem.

Hyperparameter optimization for LoRA. Prior studies have explored black-box optimization methods (Inouye et al., 2024; Tribes et al., 2024; Oliver & Wang, 2024) and efficient search strategies (Yan et al., 2025) for LoRA hyperparameter selection. However, these approaches often require many costly evaluations and provide limited mechanisms for incorporating LoRA-specific domain knowledge. This is particularly problematic because hyperparameter optimization generally benefits from expert knowledge (Wu et al., 2019; Shawki et al., 2021; Bowler et al., 2022), and LoRA introduces adapter-specific design choices that further complicate the search process (Halfon et al., 2024; Yan et al., 2025). These limitations motivate a more informed and sample-efficient optimization framework.

Bayesian optimization for hyperparameter optimization. Bayesian optimization (BO) is a widely used approach for expensive hyperparameter optimization, as it can identify promising configurations with relatively few evaluations by combining a surrogate model with an acquisition function (Shahriari et al., 2015). However, applying BO to discrete and structured spaces remains challenging (Deshwal & Doppa, 2021; Chu et al., 2024). Recent work suggests that combining LLMs with BO can improve search by leveraging natural language priors, embeddings, or agent-based mechanisms (Zhang et al., 2023; Ramos et al., 2023; Agarwal et al., 2025; Liu et al., 2024a;c). Building on this direction, we propose a framework that integrates BO with LLMs for LoRA hyperparameter optimization by constructing a LoRA-aware embedding space and performing BO.

3. Method

We propose a framework that combines a Large Language Model (LLM) with Bayesian Optimization (BO) to discover appropriate hyperparameters for LoRA tuning. We obtain continuous embeddings from the LLM and use them as inputs to the surrogate model, enabling a BO process tailored to LoRA Hyperparameter Optimization (HPO). The LLM in our framework not only encodes rich prior knowledge through large-scale pretraining, but also provides a convenient interface for injecting additional knowledge in textual form. Furthermore, to reduce cost, we introduce proxy training evaluation, which estimates the performance of a full-dataset model using a model trained on a subset of the data. With these components, our framework improves not only the sample efficiency of BO but also the overall computational efficiency of LoRA HPO.

Algorithm 1 Pseudo code for our framework

Require: Candidate pool $\mathcal{X}_{\text{cand}}$, observed dataset $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$, budget N, parameters ω (GP), θ (Projection layer), ψ (Learnable token), LLM ϕ, acquisition function a, feature extractor $g(\cdot;\theta,\psi)$
Initialize: parameters ω, θ, ψ; $\mathcal{D}_0 \leftarrow \emptyset$; Choose initial candidate $x_1 \in \mathcal{X}_{\text{cand}}$
for $n = 1$ **to** N **do**
  $y_n \leftarrow \textit{Proxy}(x_n)$
  $\mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup \{(x_n, y_n)\}$
  Remove x_n from $\mathcal{X}_{\text{cand}}$
  while not convergence **do**
    for all $x_i \in \mathcal{D}_n$ **do**
      $t_i \leftarrow \text{Template}(x_i)$
      $\mathbf{z}_i \leftarrow g(\mathbf{x}_i;\theta,\psi) = P(\phi(t_i,\psi);\theta)$
    end for
    Compute $\log p(\mathbf{y}|\mathbf{Z},\omega,\theta,\psi)$
    Update ω, θ, ψ
  end while
  for all $x_j \in \mathcal{X}_{\text{cand}}$ **do**
    $t_j \leftarrow \text{Template}(x_j)$
    $\mathbf{z}_j \leftarrow g(\mathbf{x}_j;\theta,\psi) = P(\phi(t_j,\psi);\theta)$
    Compute $a(\mathbf{z}_j;\omega,\theta,\psi)$
  end for
  $j' = \arg\max_j a(\mathbf{z}_j;\omega,\theta,\psi)$
  Suggest next evaluation point $x_{n+1} \leftarrow x_{j'}$
end for
$(x^*, y^*) \leftarrow \arg\max_{(x,y)\in\mathcal{D}} y$
return x^*

3.1. Proposed Framework

Overview. Our framework repeats four steps at each BO iteration: proxy training evaluation, embedding extraction, surrogate model update, and next-point selection. Given a hyperparameter configuration x_n, we first obtain its proxy performance y_n by fine-tuning LoRA on a small subset of the training data. We then convert x_n into a domain-aware text template t_n and feed it, together with a learnable token ψ, into a frozen LLM ϕ and projection layer $P(\cdot;\theta)$

Table 1. **Hyperparameter search range.**

Hyperparameters	Search Range	Count
Rank (r)	1 ∼256 (2^n)	9
Scaling Factor (α)	$\frac{r}{2} \sim 128r$ ($2^n r$)	9
Batch Size	2 ∼256 (2^n)	8
Learning Rate	1e-6 ∼ 5e-3	10
Dropout Rate	0.0 ∼ 0.3 ($0.05 \times n$)	7

to obtain the embedding $\mathbf{z}_n = P(\phi(t_n, \psi); \theta)$. The surrogate model is updated using the observed pair $(\mathbf{z}n, y_n)$, and the next configuration is selected by applying the acquisition function to the candidate pool $\mathcal{X}cand$. Algorithm 1 summarizes the full procedure.

Domain-aware prompting. To incorporate LoRA-specific knowledge into the embedding space, we represent each hyperparameter configuration as a structured text template. Instead of using only simple name-value pairs, such as $t = \text{name}, \text{value}$ (Ranković & Schwaller, 2025), we use $t = \text{explanation}, \text{name}, \text{value}$. The explanation describes the role of each hyperparameter and its interactions with others, such as the relationship between rank and the scaling factor.

Learnable token and projection layer. We introduce a learnable token ψ and a projection layer $P(\cdot; \theta)$ to adapt LLM embeddings for BO. The learnable token is appended to the domain-aware template and processed by the frozen LLM, while the projection layer maps the extracted representation into a BO-friendly feature space. The final embedding is defined as $\mathbf{z} = P(\phi(t, \psi); \theta)$. During optimization, the LLM remains frozen, and only the learnable token and projection layer are trained.

Bayesian optimization with LLM. We perform BO in the LLM-based embedding space rather than directly on the raw hyperparameter space. Since LoRA hyperparameters are discrete and structured, directly applying a standard GP can be ineffective. To address this, we use the embedding function $g(\mathbf{x}; \theta, \psi) = P(\phi(t, \psi); \theta)$ and define the GP kernel over the resulting embeddings:

$$k(\mathbf{x}, \mathbf{x}'|\omega) \rightarrow k(g(\mathbf{x}; \theta, \psi), g(\mathbf{x}'; \theta, \psi)|\omega, \theta, \psi). \quad (1)$$

The GP kernel parameters ω, projection layer parameters θ, and learnable token ψ are jointly optimized by maximizing the marginal log-likelihood:

$$\Phi^* = \arg\max_{\Phi} \mathcal{L}(\Phi), \quad (2)$$

where $\Phi = \omega, \theta, \psi$. After updating the surrogate model, we evaluate the acquisition function over the candidate pool and select the next hyperparameter configuration to test.

Proxy training evaluation. Since evaluating every LoRA configuration with full-data fine-tuning is expensive, we use proxy training evaluation to reduce cost. Instead of training on the entire dataset, we fine-tune LoRA on a randomly selected subset and use its performance as a proxy for full-training performance. Prior work has shown that subset-based training can strongly correlate with full-data results (Klein et al., 2017; Oliver & Wang, 2024), and we empirically find that using only 10% of the data is sufficient in our setting. This substantially reduces evaluation time and allows more BO iterations under the same budget.

4. Experiments

4.1. Experimental Setting

We define the hyperparameter candidate pool in Table 1.

Tasks. Following prior work (Meng et al., 2024), we evaluate on three tasks: math reasoning, code generation, and conversation. For math reasoning, models are fine-tuned on MetaMathQA and evaluated on GSM8K and MATH using Accuracy (%). For code generation, models are fine-tuned on CodeFeedback and evaluated on HumanEval and MBPP using Pass@1. For conversation, models are fine-tuned on WizardLM-Evol-Instruct and evaluated on MT-Bench. Each training dataset contains 100K samples, and we use a 10K subset for proxy training evaluation.

Baselines. We benchmark our framework against several HPO methods: random search, Optuna, standard Bayesian optimization (BO), latent Bayesian optimization (LBO), and NOMAD (Tribes et al., 2024).

4.2. Experimental Results

Hyperparameter optimization for PiSSA and various models. We evaluate our framework on LoRA, PiSSA, and multiple backbone models. Table 2 shows that our method consistently improves performance over the corresponding baseline settings, demonstrating that the proposed HPO framework can serve as a plug-and-play module for both LoRA variants and different LLM architectures. We also observe several meaningful hyperparameter trends: smaller batch sizes are often preferred, consistent with prior work (Marek et al., 2025); applying dropout frequently improves performance; and strong results sometimes appear when the scaling factor α is 16 or 32 times the rank r. This last finding goes beyond common guidelines that set α to twice the rank (Diehl, 2024) or use rank-based or fixed scaling values (Sun et al., 2024; Liu et al., 2025), suggesting that broader exploration of rank–scaling combinations can reveal stronger LoRA configurations.

Comparison with various HPO methods. We compare our framework with widely used HPO baselines and a LoRA-specific HPO (NOMAD) under the same optimization bud-

Table 2. **Generalization results across LoRA variants and model architectures.**

(a) Results across LoRA variants

Strategy	Ours	Accuracy (%)		Pass@1		GPT-Score
		GSM8K	MATH	HumanEval	MBPP	MT-Bench
LoRA (Hu et al., 2022)	✗	41.47	5.24	16.31	35.47	7.181
	✓	62.93 (+21.46)	12.88 (+7.64)	30.49 (+14.18)	42.59 (+7.12)	7.350 (+0.169)
PiSSA (Meng et al., 2024)	✗	52.46	7.34	22.56	40.48	7.200
	✓	60.88 (+8.42)	12.06 (+4.72)	31.71 (+9.15)	41.53 (+1.05)	7.475 (+0.275)

(b) Results across model architectures

Model	Ours	Accuracy (%)		Pass@1		GPT-Score
		GSM8K	MATH	HumanEval	MBPP	MT-Bench
Mistral-7B-v0.1 (Jiang et al., 2023)	✗	69.90	19.96	45.73	61.90	8.425
	✓	74.07 (+4.17)	23.46 (+3.50)	54.27 (+8.54)	65.08 (+3.18)	8.688 (+0.263)
Gemma-7B (Team et al., 2024)	✗	75.51	29.44	49.39	63.23	8.363
	✓	78.77 (+3.26)	30.24 (+0.80)	53.05 (+3.66)	67.46 (+4.23)	8.488 (+0.125)

Table 3. **Comparison against existing HPO methods.**

Search Method	Accuracy (%)		Pass@1	
	GSM8K	MATH	HumanEval	MBPP
Random	59.14	10.51	23.17	36.77
Optuna	54.13	10.50	27.44	38.62
BO	57.32	11.42	20.12	35.19
LBO	59.51	11.88	26.83	37.83
NOMAD	52.160	9.12	24.39	37.30
Ours	**62.93**	**12.88**	**30.49**	**42.59**

Table 4. **Ablation studies of each component.**

Projection Layer	Domain-aware Prompting	Learnable Token	GSM8K	MATH
✗	✗	✗	47.76	8.72
✓	✗	✗	53.98	9.16
✓	✓	✗	61.41	12.46
✓	✓	✓	**62.93**	**12.88**

get. As shown in Table 3, our method outperforms all baselines, including BO-based methods, showing that LLM-based domain knowledge improves both search efficiency and effectiveness for LoRA HPO.

Ablation studies. We ablate the three main components of our framework: domain-aware prompting, the projection layer, and the learnable token. As shown in Table 4, each component improves BO performance, with domain-aware prompting providing a particularly large gain, highlighting the importance of explicit LoRA-specific knowledge. We also find that vanilla BO often explores only a limited region, especially similar learning rates, while the full framework searches a broader range of configurations, showing that our components enable more effective exploration and better hyperparameter selection.

Qualitative analysis of the effect of our framework. We visualize the embeddings $\mathbf{z}$ of hyperparameter configurations in Figure 1 to examine the effect of each component.

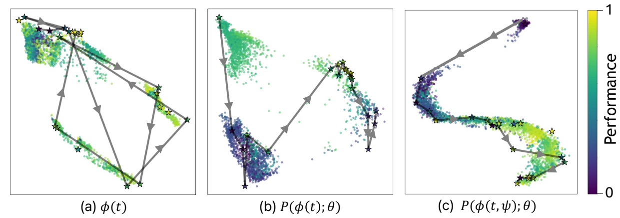

Figure 1. **Changes of optimization paths.** (a) Frozen LLM ϕ; (b) Frozen LLM with projection layer $P(\cdot;\theta)$; and (c) Frozen LLM with both the projection layer and learnable token ψ. Dots denote all possible points, and stars denote visited points during optimization. The arrows show selected optimization trajectories.

With frozen LLM embeddings, high- and low-performing configurations are entangled, making BO unstable and poorly directed. Adding the projection layer produces a more structured space, while further incorporating the learnable token organizes embeddings more clearly according to performance. The BO trajectories also show that calibrated embeddings guide the search toward high-performing regions, whereas frozen embeddings lack a clear direction. These results suggest that the projection layer and learnable token make the BO landscape smoother and more discriminative, improving search efficiency and final performance.

5. Conclusion

We propose an LLM-guided Bayesian optimization framework for LoRA hyperparameter optimization. Our method injects LoRA-specific knowledge through domain-aware prompting and calibrates LLM embeddings with a learnable token and projection layer for effective BO. To reduce cost, we use proxy evaluation on a training data subset. Experiments show that our framework efficiently finds strong hyperparameter configurations, improves performance across LoRA variants, model architectures, and model scales, and outperforms existing HPO methods in both efficiency and effectiveness. Beyond LoRA, it can serve as a practical baseline for broader fine-tuning strategies.

References

Agarwal, D., Arivazhagan, M. G., Das, R., Swamy, S., Khosla, S., and Gangadharaiah, R. Searching for optimal solutions with llms via bayesian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Biderman, D., Portes, J., Ortiz, J. J. G., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., Chiley, V., Frankle, J., et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.

Bini, M., Girrbach, L., and Akata, Z. Delora: Decoupling angles and strength in low-rank adaptation. *arXiv preprint arXiv:2503.18225*, 2025.

Bowler, A. L., Pound, M. P., and Watson, N. J. A review of ultrasonic sensing and machine learning methods to monitor industrial processes. *Ultrasonics*, 124:106776, 2022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chu, J., Park, J., Lee, S., and Kim, H. J. Inversion-based latent bayesian optimization. *Advances in Neural Information Processing Systems*, 37:68258–68286, 2024.

Deshwal, A. and Doppa, J. Combining latent space and structured kernels for bayesian optimization over combinatorial spaces. *Advances in neural information processing systems*, 34:8185–8200, 2021.

Diehl, S. Introduction to fine-tuning large language models, 2024. Available at https://www.stephendiehl.com/posts/training_llms/.

Halfon, A., Gretz, S., Arviv, O., Spector, A., Toledo-Ronen, O., Katz, Y., Ein-Dor, L., Shmueli-Scheuer, M., and Slonim, N. Stay tuned: An empirical study of the impact of hyperparameters on llm tuning in real-world applications. *arXiv preprint arXiv:2407.18990*, 2024.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Inouye, D., Lindo, L., Lee, R., and Allen, E. Applied autotuning on lora hyperparameters, 2024.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Jin, H., Wei, W., Wang, X., Zhang, W., and Wu, Y. Rethinking learning rate tuning in the era of large language models. In *2023 IEEE 5th International Conference on Cognitive Machine Intelligence (CogMI)*, pp. 112–121. IEEE, 2023.

Kalajdzievski, D. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

Klein, A., Falkner, S., Bartels, S., Hennig, P., and Hutter, F. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial intelligence and statistics*, pp. 528–536. PMLR, 2017.

Korovina, K., Xu, S., Kandasamy, K., Neiswanger, W., Poczos, B., Schneider, J., and Xing, E. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3393–3403. PMLR, 2020.

Liu, J., Kong, Z., Dong, P., Shen, X., Zhao, P., Tang, H., Yuan, G., Niu, W., Zhang, W., Lin, X., et al. Rora: Efficient fine-tuning of llm with reliability optimization for rank adaptation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Liu, S., Gao, C., and Li, Y. Large language model agent for hyper-parameter optimization. *arXiv preprint arXiv:2402.01881*, 2024a.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024b.

Liu, T., Astorga, N., Seedat, N., and van der Schaar, M. Large language models to enhance bayesian optimization. *arXiv preprint arXiv:2402.03921*, 2024c.

Mao, Y., Ge, Y., Fan, Y., Xu, W., Mi, Y., Hu, Z., and Gao, Y. A survey on lora of large language models. *Frontiers of Computer Science*, 19(7):197605, 2025.

Marek, M., Lotfi, S., Somasundaram, A., Wilson, A. G., and Goldblum, M. Small batch size training for language models: When vanilla sgd works, and why gradient accumulation is wasteful. *arXiv preprint arXiv:2507.07101*, 2025.

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

Meo, C., Sycheva, K., Goyal, A., and Dauwels, J. Bayesian-lora: Lora based parameter efficient fine-tuning using optimal quantization levels and rank values trough differentiable bayesian gates. *arXiv preprint arXiv:2406.13046*, 2024.

Oliver, M. and Wang, G. Crafting efficient fine-tuning strategies for large language models. *arXiv preprint arXiv:2407.13906*, 2024.

Rajabzadeh, H., Valipour, M., Zhu, T., Tahaei, M., Kwon, H. J., Ghodsi, A., Chen, B., and Rezagholizadeh, M. Qdylora: Quantized dynamic low-rank adaptation for efficient large language model tuning. *arXiv preprint arXiv:2402.10462*, 2024.

Ramos, M. C., Michtavy, S. S., Porosoff, M. D., and White, A. D. Bayesian optimization of catalysts with in-context learning. *arXiv preprint arXiv:2304.05341*, 2023.

Ranković, B. and Schwaller, P. Bochemian: Large language model embeddings for bayesian optimization of chemical reactions. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023.

Ranković, B. and Schwaller, P. Gollum: Gaussian process optimized llms–reframing llm finetuning through bayesian optimization. *arXiv preprint arXiv:2504.06265*, 2025.

Sengupta, A., Seth, V., Pathak, A., Verma, A., Raman, N., Gopalakrishnan, S., Chatterjee, N., and Chakraborty, T. Robust and efficient fine-tuning of llms with bayesian reparameterization of low-rank adaptation. *arXiv preprint arXiv:2411.04358*, 2024.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2015.

Shawki, N., Nunez, R. R., Obeid, I., and Picone, J. On automating hyperparameter optimization for deep learning applications. In *2021 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pp. 1–7. IEEE, 2021.

Sun, Y., Li, Z., Li, Y., and Ding, B. Improving lora in privacy-preserving federated learning. *arXiv preprint arXiv:2403.12313*, 2024.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Tribes, C., Benarroch-Lelong, S., Lu, P., and Kobyzev, I. Hyperparameter optimization for large language model instruction-tuning, 2024. URL https://arxiv.org/abs/2312.00949.

Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., and Deng, S.-H. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

Yan, M., Wang, Z., Jia, Z., Venkataraman, S., and Wang, Y. Plora: Efficient lora hyperparameter tuning for large models. *arXiv preprint arXiv:2508.02932*, 2025.

Zhang, M. R., Desai, N., Bae, J., Lorraine, J., and Ba, J. Using large language models for hyperparameter optimization. *arXiv preprint arXiv:2312.04528*, 2023.

Zhang, R., Qiang, R., Somayajula, S. A., and Xie, P. Autolora: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. *arXiv preprint arXiv:2403.09113*, 2024.