Transfer Learning for Finetuning Large Language Models

Tobias Strangmann[1], Lennart Purucker[1], Jörg K.H. Franke[1], Ivo Rapant[1], Fabio Ferreira[1], Frank Hutter[1,2]

[1]University of Freiburg, [2]ELLIS Institute Tübingen

Abstract

As the landscape of large language models expands, efficiently finetuning for specific tasks becomes increasingly crucial. At the same time, the landscape of parameter-efficient finetuning methods rapidly expands. Consequently, practitioners face a multitude of complex choices when searching for an optimal finetuning pipeline for large language models. To reduce the complexity for practitioners, we investigate transfer learning for finetuning large language models and aim to transfer knowledge about configurations from related finetuning tasks to a new task. In this work, we transfer learn finetuning by meta-learning performance and cost surrogate models for grey-box meta-optimization from a new meta-dataset. Counter-intuitively, we propose to rely only on transfer learning for new datasets. Thus, we do not use task-specific Bayesian optimization but prioritize knowledge transferred from related tasks over task-specific feedback. We evaluate our method on eight synthetic question-answer datasets and a meta-dataset consisting of 1,800 runs of finetuning Microsoft's Phi-3. Our transfer learning is superior to zero-shot, default finetuning, and meta-optimization baselines. Our results demonstrate the transferability of finetuning to adapt large language models more effectively.

1 Introduction

The landscape of large language models (LLMs) rapidly expands to a zoo of models (Team, 2024a; Abdin et al., 2024; Liu et al., 2024; DeepSeek-AI et al., 2024; Dubey et al., 2024; Jiang et al., 2023; Mistral AI, 2024; Team, 2024b; Yang et al., 2024), where different models exhibit varying strengths on specific tasks (Wei et al., 2022). At the same time, the landscape of parameter-efficient finetuning methods rapidly expands (Hu et al., 2021; Dettmers et al., 2023; Poth et al., 2023; Hayou et al., 2024).

Consequently, practitioners face a multitude of complex choices for finetuning LLMs. To support practitioners and reduce complexity, we investigate transfer learning of deep-learning pipelines for an LLM and specifications for the finetuning process, including all associated hyperparameters. We aim to transfer knowledge about pipelines from related finetuning tasks to a new task. Thus enabling practitioners to adapt LLMs more effectively to new tasks.

In this work, we transfer learn finetuning by meta-learning performance and cost surrogate models for grey-box meta-optimization from a new meta-dataset. We implement grey-box meta-optimizing by adjusting the Quick-Tune algorithm (Arango et al., 2024). Quick-Tune, was introduced for image classification and supports meta-learning surrogate models. In our version, we propose to rely only on the meta-learned surrogate models trained from scratch. That is, we do not use task-specific Bayesian optimization because *we do not refit the surrogate models* for a new dataset. In other words, our version of Quick-Tune can be understood as a dataset-aware portfolio builder (Xu et al., 2010). While counter-intuitive, we hypothesize that disabling Bayesian optimization leads to better generalization.

NeurIPS 2024 Workshop on Adaptive Foundation Models

We verify the effectiveness of our method for large language models by generating a meta-dataset based on a synthetic question-answer dataset and 1,800 runs of pipelines for finetuning Microsoft's Phi-3 model (Abdin et al., 2024). Our results show that transfer learning finetuning is superior to random search, DEHB (Awad et al., 2021), and Quick-Tune with Bayesian optimization. Moreover, meta-optimizing finetuning is, as expected, better than zero-shot and default LoRa (Hu et al., 2021).

Our Contributions. To make LLMs more easily adaptable and facilitate future studies, we contribute **(1)** synthetic datasets that serve a dual purpose: a) to create a meta-dataset for transfer learning and b) as an evaluation framework for LLM models; **(2)** a version of Quick-Tune for LLM finetuning adapted from the image to language domain; and **(3)** a novel counter-intuitive yet effective approach to finding the optimal pipeline for finetuning LLMs through transfer learning.

2 Related Work

Synthetic NLP Datasets & Meta-dataset. Question-answer datasets are scarce, with only a few notable examples such as TriviaQA, SQuAD, NaturalQuestions, and PubMedQA (Joshi et al., 2017; Rajpurkar et al., 2016; Kwiatkowski et al., 2019; Jin et al., 2019). Collecting large-scale question-answer datasets is resource-intensive, prompting researchers to explore synthetic generation methods to reduce annotation costs (Yang et al., 2017; Nayak et al., 2024; Lee et al., 2023; Puri et al., 2020; Ovadia et al., 2024). A recent approach by Mecklenburg et al. (2024) utilized *GPT-4* (OpenAI et al., 2024) as an LLM teacher to extract facts from Wikipedia articles and generate question-answer pairs. We use a similar method but apply it to arXiv papers with Llama-3.1-70b (Dubey et al., 2024).

Optimizing Finetuning Many finetuning methods with many hyperparameters exist, cf. (Hu et al., 2021; Dettmers et al., 2023; Li et al., 2023; Hayou et al., 2024; Poth et al., 2023; Liu et al., 2022; Wu et al., 2024). Likewise, many other hyperparameters of the finetuning pipeline exist, such as the choice of optimizer Shazeer and Stern (2018); Loshchilov and Hutter (2019); Franke et al. (2023); Chen et al. (2023). To address the multitude of choices for finetuning, recent work proposed (automated) meta-optimization to determine the optimal combination of finetuning method, optimizer, and hyperparameters. Methods like AutoGluon Multimodal (Tang et al., 2024), AutoPEFT (Zhou et al., 2024), AutoLoRa (Xu et al., 2023), and Quick-Tune (Rapant et al., 2024). However, these methods do not support finetuning LLMs for text generation, which is the focus of our work.

Transfer Learning Finetuning. In general, Quick-Tune (Arango et al., 2024) and its abstraction Quick-Tune-Tool (Rapant et al., 2024), building on earlier frameworks such as Öztürk et al. (2022), focus on transfer learning finetuning pipelines during meta-optimization. However, these prior works are limited to image classification. Our work extends Quick-Tune to finetuning LLMs and proposes a novel algorithmic adjustment. For LLMs, Zhang et al. (2024) introduced a meta-learning-related method for LoRA Hu et al. (2021). This method, however, does not transfer knowledge from related tasks to a new task. Instead, it performs a bi-level optimization for the LoRA rank and weights for one task. In other words, it is comparable to meta-optimizing only the rank of LoRA. In contrast, our work transfers knowledge between tasks via meta-learning. Likewise, all methods we consider can meta-optimize all hyperparameters of a finetuning pipeline.

3 Method

Our method, illustrated in Figure 1, consist of three steps: **A)** create synthetic NLP datasets from scientific papers, **B)** create a meta-dataset by training and evaluating finetuning pipelines; and **C)** transfer learning by pre-training our version of Quick-Tune on our meta-dataset. We then apply pre-trained Quick-Tune to find the optimal finetuning pipeline for new, related NLP tasks. The complete computational resources used for this method are listed in Section E. Limitations of our method can be found in appendix F.

A) Synthetic NLP Datasets. We follow Mecklenburg et al. (2024) to generate synthetic question-answer datasets from scientific papers from arxiv.org. In detail, we crawl papers and convert them to plain text papers with mathematical formulas translated to LaTeX. Next, we use a self-hosted version of *Llama-3.1-70B Instruct* (L3-70B) (AI@Meta, 2024) to extract atomic facts from each chapter of a paper. Then, we generate a set of 12 question-answer pairs for each fact. We add ten to training, one to validation, and one to testing data. Finally, our new question-answer dataset consists of training, validation, and test question-answer pairs for all facts. Appendix A details our prompt templates.

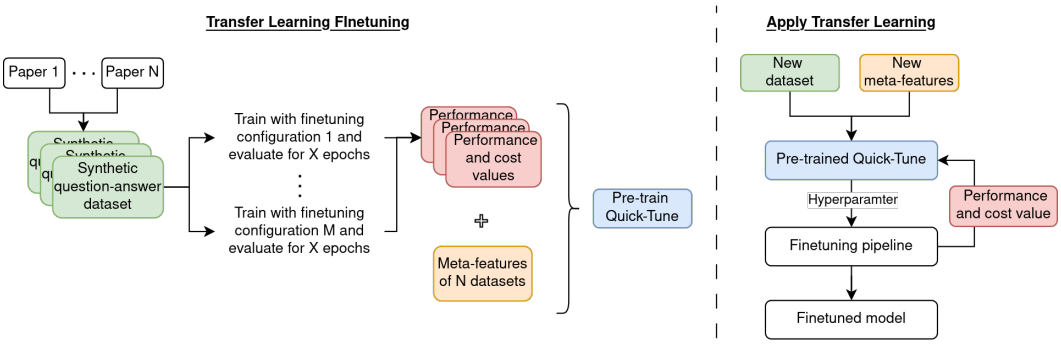

Figure 1: **Method Overview.** We generate new NLP datasets from scientific papers and then create a meta-dataset, which we use for transfer learning to finetune by pre-training Quick-Tune (left). For a new dataset, we compute meta-features and then apply the pre-trained Quick-Tune (right).

B) Our Meta-dataset. We create a meta-dataset by collecting meta-features, performance, and cost values for finetuning pipelines on synthetic datasets. Therefore, we create question-answer datasets from 30 papers. Then, for each paper, we train 60 finetuning pipelines with the training and validation question-answer pairs and evaluate them on the test pairs, producing 1,800 runs in total. Finally, we compute meta-features for each paper; see Appendix B for an overview. We visualize an overview of all runs in our meta-dataset in Figure 2.

For each paper, we randomly sample finetuning pipelines from a search space based on hyperparameters for LoRA (Hu et al., 2021), optimizers (AdamW (Loshchilov and Hutter, 2019) or AdamCPR (Franke et al., 2023)), and the learning rate scheduler. We also include a default finetuning pipeline as a baseline. We detail the search space in Appendix D.

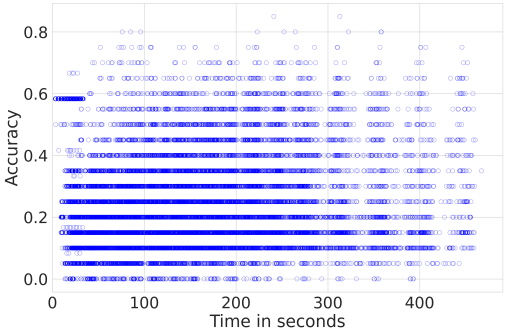

Figure 2: **Our Meta-Dataset.** For each run stored in our meta-dataset, represented by a blue circle, we present the accuracy and finetuning time in seconds.

After each epoch, we evaluate the finetuned models in the form of a student with L3-70B as a teacher. Given a finetuned model's answer to a question, L3-70B evaluates whether the generated answer is correct (0 or 1). Thus, L3-70B assess whether the student model learned to answer new questions about facts in papers after being finetuned on question-answer pairs about these facts. See Appendix C for the prompt template and an example of this process.

We use four meta-features to characterize each synthetic question-answer dataset: the total number of tokens, average sample length, vocabulary size, and the ratio of question-to-answer lengths.

C) Transfer Learning Finetuning with Quick-Tune. We use the performance metrics and meta-features stored in our meta-dataset to pre-train Quick-Tune, implemented in Quick-Tune-Tool (Rapant et al., 2024). That is, we meta-train the Gaussian Process-based surrogate models of Quick-Tune. This allows Quick-Tune to start with a strong prior for the performance and cost of finetuning pipeline on a new dataset, transferring knowledge across tasks. By default, the surrogate models are continuously refitted during optimization to facilitate Bayesian optimization.

In our version of Quick-Tune, we disable Bayesian optimization by disabling refitting. We hypothesize that disabling Bayesian optimization leads to better generalization by relying more on the knowledge transferred from related tasks than task-specific noise. In other words, while Bayesian optimization exploits the most promising pipeline on validation data, only relying on the prior from transfer learning could lead the meta-optimizer to find better, more general pipelines.

From a broader perspective, our version of Quick-Tune can be understood as a *dataset-aware* portfolio builder. Portfolios (Xu et al., 2010) are known as robust transfer learning methods (Feurer et al., 2022; Salinas and Erickson, 2023).

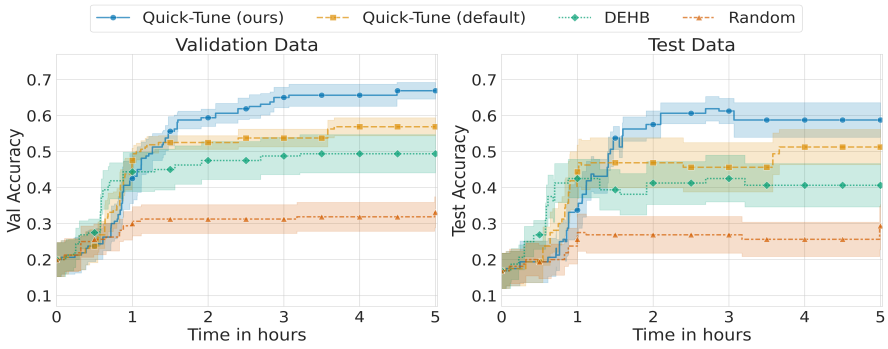

Figure 3: **Optimizer Performance Over Time.** We visualize the average validation (left) and test (right) performance across the eight datasets over time. At each time point, we evaluated the best pipeline found so far. We observe that DEHB and Quick-Tune (default) stagnant after 1 to 1.5 hours, with little progress on test scores afterward. Quick-Tune (ours) only stagnates after 3 hours.

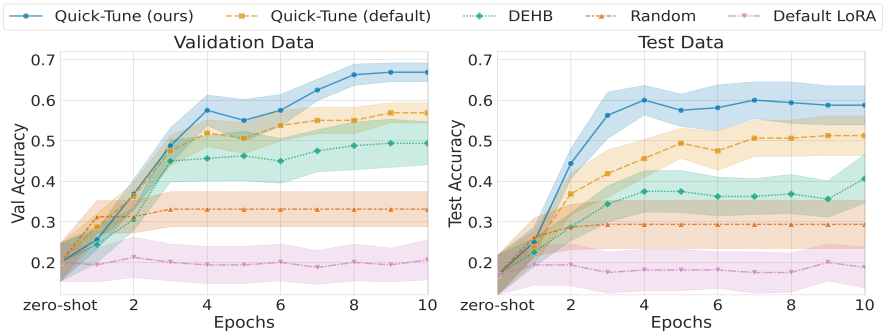

Figure 4: **Final Performance.** We show the validation (left) and test (right) learning curve of the best pipeline returned by the optimizers after 5 hours, averaged across eight datasets. The finetuning pipeline returned by Quick-Tune (ours) performs best.

4 Results

Experimental Setup. We experiment with finetuning Phi 3 Mini Instruct (3.8B parameters) (Abdin et al., 2024) on eight newly generated synthetic question-answer datasets (see Appendix B). We employ random search, DEHB (Awad et al., 2021), default Quick-Tune (Arango et al., 2024), and our version of Quick-Tune to meta-optimize the finetuning pipeline. Furthermore, we evaluate a default finetuning pipeline and zero-shot performance. Each optimizer is given a five-hour time budget. We again use our *Llama-3.1-70B* teacher for evaluation.

HYPOTHESIS: TRANSFER LEARNING LEADS TO BETTER GENERALIZATION.

Figure 3 presents the performance over time of the meta-optimizers for validation and test data. Figure 4 shows the performance of the best pipeline, see Appendix G for configuration details. The error bars in both figures represent the standard error of the mean. Note the initial performance represents the zero-shot performance of Phi 3. We observe that Quick-Tune (default) and DEHB get stuck after 1.5 hours during meta-optimization and fail to find a significantly better finetuning pipeline afterward. In contrast, Quick-Tune (ours), which relies only on transfer learning, further improves test performance. A similar trend manifests when training the best pipeline found by each meta-optimizer. The pipeline found by Quick-Tune (ours) generalizes best to test data.

Conclusion. In this study, we demonstrated that relying only on transfer learning for finetuning yields better performance than alternative methods, challenging conventional approaches and potentially simplifying the process of adapting large language models to specific tasks. In future work, we plan to understand this phenomenon in more detail and to generalize it to a meta-optimization method. Thus allowing us to effectively manage the zoo of and the plethora of methods for adapting large language models to specific tasks.

Acknowledgments

This work was carried out at the HoreKa Cluster, which is funded by the Baden-Württemberg Ministry of Science, Research and the Arts, and the Federal Ministry of Education and Research. The authors would also like to thank the state of Baden-Württemberg for the support provided by the bwHPC and the German Research Foundation (DFG) for funding through INST 35/1597-1 FUGG. We acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under SFB 1597 (SmallData), grant numbers 499552394 and 417962828. Frank Hutter acknowledges the financial support of the Hector Foundation.

References

Gemma Team. Gemma. 2024a. doi: 10.34740/KAGGLE/M/3301. URL https://www.kaggle.com/m/3301.

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Liyuan Liu, Young Jin Kim, Shuohang Wang, Chen Liang et al. Grin: Gradient-informed moe, 2024. URL https://arxiv.org/abs/2409.12136.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024. URL https://arxiv.org/abs/2405.04434.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford et al. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Mistral AI. Mistral nemo, 2024. URL https://mistral.ai/news/mistral-nemo/. Accessed: 2024-09-24.

Qwen Team. Qwen2.5: A party of foundation models, September 2024b. URL https://qwenlm.github.io/blog/qwen2.5/.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel et al. Emergent abilities of large language models, 2022. URL https://arxiv.org/abs/2206.07682.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu et al. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023. URL https://arxiv.org/abs/2305.14314.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha et al. Adapters: A unified library for parameter-efficient and modular transfer learning, 2023. URL https://arxiv.org/abs/2311.11077.

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

Sebastian Pineda Arango, Fabio Ferreira, Arlind Kadra, Frank Hutter, and Josif Grabocka. Quick-tune: Quickly learning which pretrained model to finetune and how. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tqh1zdXIra.

Lin Xu, Holger Hoos, and Kevin Leyton-Brown. Hydra: Automatically configuring algorithms for portfolio-based selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 210–216, 2010.

Noor Awad, Neeratyoy Mallik, and Frank Hutter. Dehb: Evolutionary hyperband for scalable, robust and efficient hyperparameter optimization. *arXiv preprint arXiv:2105.09821*, 2021.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017. URL https://arxiv.org/abs/1705.03551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016. URL https://arxiv.org/abs/1606.05250.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins et al. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering, 2019. URL https://arxiv.org/abs/1909.06146.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William W. Cohen. Semi-supervised qa with generative domain-adaptive nets, 2017. URL https://arxiv.org/abs/1702.02206.

Nihal V. Nayak, Yiyang Nan, Avi Trost, and Stephen H. Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation, 2024. URL https://arxiv.org/abs/2402.18334.

Seongyun Lee, Hyunjae Kim, and Jaewoo Kang. Liquid: A framework for list question answering dataset generation, 2023. URL https://arxiv.org/abs/2302.01691.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Training question answering models from synthetic data, 2020. URL https://arxiv.org/abs/2002.09599.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in llms, 2024. URL https://arxiv.org/abs/2312.05934.

Nick Mecklenburg, Yiyou Lin, Xiaoxiao Li, Daniel Holstein et al. Injecting new knowledge into large language models via supervised fine-tuning, 2024. URL https://arxiv.org/abs/2404.00213.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He et al. Loftq: Lora-fine-tuning-aware quantization for large language models, 2023. URL https://arxiv.org/abs/2310.08659.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, 2022. URL https://arxiv.org/abs/2205.05638.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger et al. Reft: Representation finetuning for language models, 2024. URL https://arxiv.org/abs/2404.03592.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost, 2018. URL https://arxiv.org/abs/1804.04235.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Jörg K. H. Franke, Michael Hefenbrock, Gregor Koehler, and Frank Hutter. Constrained parameter regularization, 2023. URL https://arxiv.org/abs/2311.09058.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real et al. Symbolic discovery of optimization algorithms, 2023. URL https://arxiv.org/abs/2302.06675.

Zhiqiang Tang, Haoyang Fang, Su Zhou, Taojiannan Yang et al. Autogluon-multimodal (automm): Supercharging multimodal automl with foundation models. *arXiv preprint arXiv:2404.16233*, 2024.

Han Zhou, Xingchen Wan, Ivan Vulić, and Anna Korhonen. Autopeft: Automatic configuration search for parameter-efficient fine-tuning, 2024. URL https://arxiv.org/abs/2301.12132.

Xilie Xu, Jingfeng Zhang, and Mohan Kankanhalli. Autolora: A parameter-free automated robust fine-tuning framework, 2023. URL https://arxiv.org/abs/2310.01818.

Ivo Rapant, Lennart Purucker, Fabio Ferreira, Sebastian Pineda Arango et al. Quick-tune-tool: A practical tool and its user guide for automatically finetuning pretrained models. In *AutoML Conference 2024 (Workshop Track)*, 2024.

E. Öztürk, F. Ferreira, H. S. Jomaa, L. Scmidth-Thieme et al. Zero-shot automl with pretrained models. In *Proc. of ICML'22*, pages 1128–1135, 2022.

Ruiyi Zhang, Rushi Qiang, Sai Ashish Somayajula, and Pengtao Xie. Autolora: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. *arXiv preprint arXiv:2403.09113*, 2024.

AI@Meta. Llama 3.1 model card. 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md.

Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: Hands-free automl via meta-learning. *Journal of Machine Learning Research*, 23(261):1–61, 2022.

David Salinas and Nick Erickson. Tabrepo: A large scale repository of tabular model evaluations and its automl applications. *arXiv preprint arXiv:2311.02971*, 2023.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL `https://arxiv.org/abs/2205.14135`.

Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL `https://doi.org/10.21105/joss.03021`.

Appendix

A Prompt Templates To Generate Our Synthetic NLP Dataset

We follow the prompt (Facts generation) to extract atomic facts out of unlabeled text. Our self-hosted version of L3-70B extracts as many as possible facts out of reprocessed approximately 2k token long text fragments. For each fact, we generate 12 question-answers pairs by using Q & A generation prompt , skipping facts that are too general or insufficiently specific to the article's topic (generated by Key topic generation). We aim to generate as many questions and answers as possible that explicitly relate to the fact, then paraphrase them to achieve the required 12 pairs.

Facts generation prompt (Mecklenburg et al., 2024)

System: "You are an AI assistant who knows about current artificial intelligence. Be precise but concise in your answer."
User: "Please break down the following snippet from an article about {key_topic} into atomic facts.\nGoal 1: The atomic facts should be as simple as possible, if it's a compound sentence, break down one more time.\nGoal 2: For clarity, avoid using pronouns like 'it', 'he', 'she', 'this', 'that' etc., and instead use the full names or titles.\nGoal 3: Output in the format: 1.fact_1\n\n{passage}\n\n1."

Q & A generation prompt (Mecklenburg et al., 2024)

System: "You are an AI assistant who knows about factual information about the paper with the title: {paper title}. Be precise but concise in your answer."
User: "Write 12 pairs of questions and answers probing the facts and statistics the given fact {fact} about {key_topic}.\nConsider first generating questions and answers that are very relevant and explicit to the fact, then paraphrase those questions and answers to reach the desired 12 Q&A pairs. If the fact is too broad or not specific enough to theme, you may reply with only with 'SKIP' and be done.\nEXAMPLE:\nFACT: 14 million viewers tuned in to the opening game of the series.\n1. Q: How many viewers watched the first game? A: 14 million people watched the first game of the series.\n\nEXAMPLE:\nFACT: The rose is red.\nSKIP\n\nFACT: fact['fact']\n1. "

Key topic generation prompt

System: "You are given a summary of the scientific paper. Return the key topic of this paper an nothing else"
User: {paper summary}

Atomic fact example

"Masked Image Modeling (MIM) is a learning framework that derives visual representations from unlabeled image data."

Q & A example

Question: "What does Masked Image Modeling (MIM) derive from unlabeled image data?"
Answer: "Masked Image Modeling (MIM) derives visual representations from unlabeled image data."

B Synthetic Datasets Details

We list our meta-features from our meta-dataset in Table 1 and the meta-dataset used for our experiments in Table 2.

Table 1: Meta-features trainings dataset

Dataset	token size	sample length	ratio q/a length	vocab size
2407.15849v1	46913	137.63	1.43	1530
2407.15847v1	82307	144.92	1.55	2570
2407.15845v1	75410	145.55	1.51	2330
2407.15843v1	117247	139.03	1.72	3840
2407.15839v1	59966	146.83	1.57	1900
2407.15837v1	83873	139.39	2.08	2720
2406.18451v2	91480	161.24	1.4	2520
2407.15835v1	87863	134.18	1.66	2940
2407.15831v1	3874	157.48	1.19	120
2405.04657v3	65048	144.11	1.3	2070
2407.15820v1	73764	164.01	1.51	1980
2402.16822v2	131833	141.71	1.57	4190
2401.00009v3	87762	131.66	2.42	2740
2407.15815v1	69078	142.61	1.41	2210
2407.15814v1	76705	149.07	1.55	2540
2403.20262v2	93673	131.59	1.91	2930
2407.13044v2	27154	129.01	1.8	920
2307.15220v3	146050	142.7	1.55	4840
2407.15786v1	109720	143.94	1.8	3410
2407.15784v1	44928	151.83	1.44	1460
2405.17814v4	88773	144.15	1.65	2720
2407.15771v1	84305	139.16	1.67	2680
2407.15762v1	133882	139.16	1.62	4030
2407.15748v1	136205	140.48	1.57	4260
2407.15739v1	94869	145.1	1.74	2990
2407.15738v1	143443	137.99	1.52	4570
2407.15734v1	144566	131.11	1.57	5010
2407.04856v2	147437	141.79	1.48	4600
2402.07370v2	64881	134.8	1.57	2100
2403.07805v3	87032	140.16	1.8	2810

Table 2: Meta-features HPO comparison

Dataset	token size	sample length	ratio q/a length	vocab size
2407.15723v1	54923	139.66	1.67	1840
2407.15720v1	157268	147.32	1.49	4740
2407.15719v1	86733	148.17	1.41	2570
2407.15708v1	45482	139.93	1.70	1390
2407.15656v1	124420	145.04	1.72	3900
2407.15617v1	82637	142.93	1.57	2580
2407.15600v1	89996	139.59	1.49	2970
2401.04152v2	42769	147.14	1.53	1280

C LLM Model Evaluation Details

For the evaluation, we continue to use our in-house hosted L3-70B model implemented with llama.cpp, leveraging it for both performance and resource efficiency. We make a small adjustment to our configuration, setting llama.cpp to process 128 parallel sequences and limiting the context size to 500 tokens, which is sufficient for our evaluation needs. To ensure efficient processing, we limit the maximum number of new tokens to 50 for each generated answer. Given that a comprehensive evaluation of the entire validation and test datasets would be time-prohibitive, we opted to select 20 random, fixed validation and test indices per paper (dataset) for this study.
Based on Mecklenburg et al. (2024) we use Evaluation prompt to generate our evaluation score.

Resulting to either a positive (Positive evaluation response) or negative (Negative evaluation response) result.

Evaluation prompt (based on Mecklenburg et al. (2024))

System: "You are a high school teacher grading student's responses for questions about {key_topic}. These responses are either correct or incorrect."
User: "Please evaluate the correctness of a sentence in answering the question: "{question}".\nThe correct answer is: "{sample_answer}"\nThe student response is: "{gen_answer}".\nYour grading is binary. Give 0 if the sentence is incorrect, give 1 if the sentence is correct, based on the given correct answer and the question.\n"Please note that your output is either 0 or 1, with the corresponding justification as python dict in the following format and nothing else:\n r"{'rating': <rating>, 'justification': <justification>}"

Positive evaluation response

Question: what does imitation learning (il) rely on to learn?
Generated answer: imitation learning (il) relies on expert demonstrations to learn.
Sample answer: il learns from expert guidance.
Decision: 'rating': 1, 'justification': 'The student response is correct because it conveys the same meaning as the correct answer, which is that imitation learning relies on some form of expert input, whether it is called "guidance" or "demonstrations".'

Negative evaluation response

Question: do agents and equipped functions work together in taskgen?
Generated answer: yes, they work together as part of the hybrid approach.
Sample answer: no, agents and equipped functions operate independently.
Decision: 'rating': 0, 'justification': 'The student response is incorrect because it states that agents and equipped functions work together, whereas the correct answer is that they operate independently.'

D Search Space Details

We employ AdamW and AdamCPR optimizers (Table 3) as well cosine schedulers (Table 4) with varying warmup steps (as a percentage of training set length) and decay factors. LoRA configurations (Table 5) include different ranks, alpha values, and dropout rates, with target modules being either query, key, and value; only the output layer; or all linear layers.

While we train 10 epochs, the batch size is fixed at 32, with gradient accumulation steps of 2, 4, or 8 to achieve mini-batch sizes of 64, 128, or 256. We utilize the Hugging Face tokenizer's chat template for Phi 3 Instruct to maintain consistency with the model's original template during training. An additional configuration option is the return_assistant_mask, which generates an attention mask excluding "user" and "system" segments, focusing the model's learning on "assistant" responses.

Fixed settings across all configurations include:

- torch.nn.CrossEntropyLoss as the loss function
- Gradient clip value of 1.0
- torch.bfloat16 precision
- Flash Attention 2 (Dao et al., 2022)
- Left-side padding (due to Flash Attention requirements)

To ensure all samples in the train set are used, we augment the dataset with random samples to make it divisible by the product of batch size and gradient accumulation steps. The number of additional samples (asc) is calculated as:

$$asc = (\lceil ltrain/bg \rceil * bg) - ltrain \tag{1}$$

where bg is the product of batch size and gradient accumulation steps, and $ltrain$ is the length of the train dataset.

Default values used for "Default LoRA" in Figure 4 are marked in bold in Tables 3, 4, and 5. A gradient accumulation step of 2 was used.

Table 3: Optimizer configuration space

	parameter	
optimizer	**AdamW**	AdamCPR
learning_rate	**1e-6**, 1e-5.5, 1e-5, 1e-4.5, 1e-4, 1e-3.5, 1e-3	
weight_decay	1e-0.5, 1e-1, 1e-1.5, **1e-2**, 1e-3, 1e-4	———
kappa_init_method	———	warm_start
kappa_init_param	———	warmup_steps x (1,2,4)

Table 4: Scheduler hyperparameter

	parameter
schedule	cosine
warmup_steps %	**10**, 20, 30, 40, 50
decay_factor	0, 0.1, **0.01**

Table 5: Lora configuration space

With q = query, k = key, v = value, o = output.
all-linear = q, k, v.

	parameter
target_modules	[q, k, v], o, **all-linear**
rank	**8**, 16, 32, 64
alpha	**16**, 32
dropout	**0**, 0.1

E Experiments Compute Resources

It took 900 compute hours to run all 1800 configurations for our meta-dataset and 170 compute hours for the experiments on a single NVIDIA A100 GPU.
Each run for the meta-dataset and experiments was allocated 8 CPU cores and 16 GB RAM.
Concurrently, we utilized two NVIDIA A6000 GPUs in parallel to run our self-hosted L3-70B model.

F Limitations Of Our Method

Although our method shows promising results compared to alternative methods, our meta-features are not based on an importance analysis. Furthermore, the evaluation does not take into account whether the model to be fine-tuned might start hallucinating during training and add further invented facts to the correct answer. Furthermore, at the current state we have too little data to understand why we achieve better performance when we only do transfer learning without Bayesian optimization. Another limitation is that we do not know how our finetuning generalizes with real tasks, i.e. not with synthetic data and without a teacher model.

G Results Configuration Details

The best pipeline configurations found by the individual optimizers, listed below. Resulting configurations by Quick-Tune (ours), Quick-Tune (default), DEHB, and random optimizer in Table 6, 7, 8, and 9.

Table 6: Quick-Tune (Ours) Found Configurations

With batch size = batch size 32 and gradient accumulation step [2, 4, 8].

Dataset	batch size	decay factor	fidelity	kappa init param	lora alpha	lora dropout	lora layer	lora rank	lr	warmup steps %	optimizer	return assistant mask	weight decay
2407.15723v1	64	1.0	4	nan	16	0.0	all-linear	64	1e-3	10	AdamW	False	1e-0.5
2407.15720v1	64	0.01	7	4.0	32	0.0	o	32	1e-3	10	AdamCPR	False	1e-0.5
2407.15719v1	64	1.0	8	nan	16	0.0	all-linear	64	1e-3	10	AdamW	False	1e-0.5
2407.15708v1	64	0.01	4	4.0	32	0.0	o	32	1e-3	10	AdamCPR	False	1e-0.5
407.15656v1	64	0.1	9	nan	32	0.0	o	16	1e-3	10	AdamW	False	1e-0.5
2407.15617v1	64	1.0	8	nan	16	0.0	all-linear	64	1e-3	10	AdamW	False	1e-0.5
2407.15600v1	128	0.01	8	nan	16	0.1	o	64	1e-3	10	AdamW	True	1e-3
2401.04152v2	64	0.01	4	4.0	32	0.0	o	32	1e-3	10	AdamCPR	False	1e-0.5

Table 7: Quick-Tune (Default) Found Configurations

With batch size = batch size 32 and gradient accumulation step [2, 4, 8].

Dataset	batch size	decay factor	fidelity	kappa init param	lora alpha	lora dropout	lora layer	lora rank	lr	warmup steps %	optimizer	return assistant mask	weight decay
2407.15723v1	64	0.01	2	1.0	32	0.1	all-linear	8	1e-3	10	AdamCPR	True	1e-2
2407.15720v1	64	0.01	5	4.0	32	0.0	o	32	1e-3	10	AdamCPR	False	1e-0.5
2407.15719v1	64	0.01	3	4.0	32	0.0	o	32	1e-3	10	AdamCPR	False	1e-0.5
2407.15708v1	64	0.01	2	4.0	32	0.0	o	32	1e-3	10	AdamCPR	False	1e-0.5
2407.15656v1	128	0.01	2	nan	16	0.1	o	64	1e-3	10	AdamW	True	1e-3
2407.15617v1	64	0.01	4	4.0	32	0.0	o	32	1e-3	10	AdamCPR	False	1e-0.5
2407.15600v1	128	0.01	1	4.0	16	0.1	all-linear	16	1e-4.5	30	AdamCPR	False	1e-4
2401.04152v2	64	0.01	2	4.0	32	0.0	o	32	1e-3	10	AdamCPR	False	1e-0.5

Table 8: DEHB Found Configurations

With batch size = batch size 32 and gradient accumulation step [2, 4, 8].

Dataset	batch size	decay factor	fidelity	kappa init param	lora alpha	lora dropout	lora layer	lora rank	lr	warmup steps %	optimizer	return assistant mask	weight decay
2407.15723v1	128	1.0	3	nan	32	0.0	o	16	1e-3	10	AdamW	True	1e-0.5
2407.15720v1	64	1.0	10	2.0	32	0.0	o	32	1e-3	10	AdamCPR	True	1e-0.5
2407.15719v1	256	1.0	3	4.0	16	0.0	o	16	1e-3.5	10	AdamCPR	True	1e-0.5
2407.15708v1	128	0.1	1	2.0	32	0.0	o	32	1e-06	30	AdamCPR	True	1e-0.5
2407.15656v1	64	0.01	10	nan	16	0.0	all-linear	16	1e-3	30	AdamW	False	1e-1.5
2407.15617v1	128	0.1	10	nan	32	0.1	o	32	1e-3	10	AdamW	True	1e-0.5
2407.15600v1	64	0.1	3	4.0	16	0.0	all-linear	16	1e-06	20	AdamCPR	False	1e-1.5
2401.04152v2	128	1.0	3	2.0	16	0.0	all-linear	32	1e-3	20	AdamCPR	True	1e-2

Table 9: Random Found Configurations

With batch size = batch size 32 and gradient accumulation step [2, 4, 8].

Dataset	batch size	decay factor	fidelity	kappa init param	lora alpha	lora dropout	lora layer	lora rank	lr	warmup steps %	optimizer	return assistant mask	weight decay
2407.15723v1	256	0.10	1	4.0	16	0.1	o	16	1e-5	10	AdamCPR	True	1e-4
2407.15720v1	64	0.01	1	NaN	16	0.0	o	16	1e-3.5	40	AdamW	False	1e-3
2407.15719v1	64	1.00	1	NaN	32	0.0	o	64	1e-3	20	AdamW	True	1e-1.5
2407.15708v1	256	0.01	1	4.0	16	0.0	qkv	32	1e-5	40	AdamCPR	True	1e-0.5
2407.15656v1	128	0.01	3	2.0	32	0.1	o	64	1e-3	10	AdamCPR	False	1e-2
2407.15617v1	64	0.01	1	NaN	32	0.1	all-linear	32	1e-3	10	AdamW	True	1e-1.5
2407.15600v1	64	0.10	1	4.0	32	0.0	qkv	16	1e-3.5	50	AdamCPR	False	1e-4
2401.04152v2	64	0.01	1	NaN	16	0.1	o	64	1e-3	10	AdamW	True	1e-2

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We create a meta-dataset from our synthetic data for transfer learning and present a novel counter-intuitive approach of finding the optimal pipeline for finetuning LLMs through transfer learning.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations can be found in appendix F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We have empirical findings rather than theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Despite the fact that we do not publish any code, we have described our method (3) in detail. All necessary hyperparameters (D), meta-features, tools, prompts (A, C) and paper names (B) as well as models are mentioned.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We describe our method in section 3. The Quick-Tune-Tool (Rapant et al., 2024) is publicly available, to run similar experiments. The code is not executable with one click as it requires the setup of the server for the evaluation, as well as additional steps such as the setup of SSH keys and connections.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: See section 3 (A) and (B), Appendix D and A for information about the dataset and hyperparameters details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: In section 4 we state that we use standard error of the mean over eight datasets in both of our main plots 3 and 4, calculated with Seaborn (Waskom, 2021).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Code of Ethics was observed to the best of our knowledge and belief.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Even if we change Quick-Tune so that we don't do Bayesian optimization, we haven't designed a new tool. We show that better performance can be achieved in the language domain. For potential positive as well as negative social influences, further experiments are needed.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper proposes no risk, as the finetuned models are trained on scientific papers and will not be published either way.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The only non-own code that was used is Quick-Tune-Tools and DEHB, which is cited. The models (llama3.1 and phi3) are cited as well. Contents of arXiv e-prints are free to use for research purposes (Terms of Use for arXiv APIs).

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.