FROM GENERALIST TO SPECIALIST: EVALUATING QLORA'S PRACTICALITY

Kshitij Alwadhi [*1]

ABSTRACT

This work examines the generalization capabilities of QLoRA, a parameter-efficient fine-tuning technique, across different complexity levels of domain-specific and algorithmic tasks. Through evaluations on the MMLU benchmark and k-digit addition tasks, we found that QLoRA performs well in low-complexity domains but struggles with more challenging knowledge and reasoning requirements. The study reveals marginal improvements in complex knowledge domains and significant performance degradation in arithmetic tasks involving longer numbers. Our findings highlight QLoRA's limitations in generalizing across diverse task complexities and underscore the need for more nuanced evaluation metrics and advanced fine-tuning approaches.

1 INTRODUCTION

The rapid advancement of Large Language Models (LLMs) has significantly changed how we used to train and deploy task-specific ML models. Previously, we used to train task-specific ML models, but these days, the LLMs have become general purpose in the sense that they have demonstrated unprecedented capabilities across various domains (Kaplan et al., 2020; Brown et al., 2020; Chowdhery et al., 2022). To further improve their performance on domain specific-tasks, we started to perform fine-tuning of our models (Howard & Ruder, 2018). However, full-fine tuning is not always possible due to the computational complexity and the resource-intensive nature of that task. So we started to leverage optimizations like Low-Rank Adaptation (LoRA) and its quantized variant QLoRA to address these limitations (Hu et al., 2021; Dettmers et al., 2023).

Despite the growing interest in parameter-efficient fine-tuning (PEFT) methods, the effectiveness of these methods on performing very domain-specific knowledge transfer remains largely unexplored. In the work that introduced us to LoRA and QLoRA, the evaluation was done in such a sense that the dataset being used for fine-tuning was a very general dataset like Guannaco (OASST) (Köpf et al., 2023) and FLAN-V2 (Chung et al., 2022). While the fine-tuning did show marginal improvements on the MMLU benchmark, the evaluation did not show any light on the distinction between genuine learning and memorization from the fine-tuning.

This research addresses these critical gaps by conducting an investigation into the generalization capabilities of QLoRA across different domains and task complexities. Our primary contribution includes:

1. A novel methodology for assessing the knowledge transfer effectiveness in quantized fine-tuning
2. Empirical analysis of QLoRA's performance across domain-specific tasks.

The evaluations done in this work offer critical insights for researchers and practitioners seeking to develop more efficient and effective model adaptation techniques. Our findings highlight the need for more sophisticated approaches to knowledge integration, particularly as the demand for computationally efficient AI solutions continues to grow.

The remainder of this paper is organized as follows: Section 2 talks about the problem we are solving in detail, Section 3 provides a comprehensive review of related work, Section 4 details our methodology, Section 5 presents our experimental results and analysis, and Section 6 concludes with a discussion of our findings.

2 PROBLEM DESCRIPTION

Our research addresses a critical limitation in current fine-tuning approaches: their effectiveness across generalizing knowledge across different domains and task complexities. Specifically, we try to answer the following research questions: (1) Can QLoRA effectively transfer and integrate domain-specific knowledge into pre-trained langauge models? and (2) What are the inherent limitations of current fine-tuning approaches in terms of genuine learning vs memorization? Particularly, we use QLoRA for answering these

[1]Department of Computer Science, Purdue. Correspondence to: Kshitij Alwadhi <kshitijalwadhi@gmail.com>.

Preliminary work. Under review by the Machine Learning and Systems (MLSys) Conference. Do not distribute.

questions as it is the most commonly used and not very resource expensive for fine tuning. Our investigation focuses on understanding the boundary conditions of knowledge integration, exploring how a fine-tuning technique like QLoRA can impact model's ability to learn and generalize across varying levels of task complexity.

3 Related Work

Parameter-efficient fine-tuning (PEFT) techniques have emerged as a critical area of research in the domain of Large Language Models (LLMs), addressing the computational and resource-intensive challenges of traditional model adaptation (Howard & Ruder, 2018). Full fine-tuning, the conventional approach, requires updating all model parameters, resulting in substantial computational overhead and memory requirements (Zaken et al., 2021). This limitation has prompted the development of more efficient adaptation strategies.

3.1 Full Fine-Tuning Approaches

Traditional fine-tuning methods involve updating all parameters of a pre-trained model (Howard & Ruder, 2018). While comprehensive, these approaches suffer from several critical drawbacks: very high computational costs (multiple GPUs required for training models with billions of parameters), massive storage requirements, and potential catastrophic forgetting of pre-trained knowledge (Kirkpatrick et al., 2016). Techniques like BitFit (Zaken et al., 2021) attempted to mitigate these issues by fine-tuning only bias terms, but still failed to address the fundamental computational challenges.

3.2 Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) was proposed in (Hu et al., 2021) as a parameter-efficient alternative to full fine-tuning. The core idea behind LoRA is to introduce learnable low-rank matrices into the model architecture to adapt pre-trained weights for downstream tasks without altering the original parameters. This technique involves the following steps:

1. **Decomposing Weight Updates into Low-Rank Matrices:** LoRA assumes that the weight updates ΔW required for task-specific fine-tuning can be expressed as the product of two low-rank matrices:

$$\Delta W = A \cdot B$$

   Here, $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times k}$ are the trainable low-rank matrices, where $r \ll \min(d, k)$. This decomposition significantly reduces the number of parameters compared to full fine-tuning.

2. **Parameter Injection:** During forward passes, the modified weights are computed as:

$$W' = W + \Delta W = W + A \cdot B$$

   where W are the frozen original model weights. By keeping W fixed, LoRA ensures that the pre-trained knowledge remains intact while only the low-rank matrices A and B are updated.

The low-rank structure drastically reduces computational and memory requirements. For example, a model with billions of parameters may require fine-tuning only a few million parameters using LoRA. LoRA has been applied to Transformer-based architectures, such as GPT and BERT, and has demonstrated competitive performance across tasks like text classification, summarization, and translation while requiring a fraction of the resources of full fine-tuning.

3.3 Quantized Fine-Tuning Approaches

Building on the foundation of LoRA, (Dettmers et al., 2023) introduced Quantized LoRA (QLoRA) to further optimize fine-tuning by leveraging quantization techniques. QLoRA enhances the efficiency of LoRA through the following mechanisms:

1. **4-Bit Quantization of Model Weights:** QLoRA reduces the memory footprint of the base model by applying 4-bit quantization to its weights. This quantization represents each parameter using only 4 bits, significantly compressing the model size without incurring substantial performance degradation.

2. **Hybrid Dequantization for Computation:** During inference and training, QLoRA dequantizes the compressed weights into higher precision (e.g., 16-bit floating-point) for computations. This dynamic dequantization ensures that the model retains numerical stability and accuracy during forward and backward passes.

3. **Integration with Low-Rank Adaptation:** Similar to LoRA, QLoRA employs low-rank matrices (A and B) to adapt the quantized model for task-specific needs. The updates are added to the dequantized weights during fine-tuning:

$$W' = \text{Dequantize}(W_{\text{quantized}}) + A \cdot B$$

4. **Gradient Quantization:** To further optimize memory usage, QLoRA quantizes gradients during training. This step ensures that even large-scale models can be fine-tuned on hardware with limited memory, such as GPUs with 16 GB or less of VRAM.

By combining 4-bit quantization with low-rank adaptation, QLoRA achieves a remarkable reduction in memory requirements while enabling efficient training on commodity hardware. For instance, QLoRA can fine-tune models with over 100 billion parameters on a single consumer-grade GPU. Despite aggressive compression, QLoRA maintains competitive performance on benchmark datasets, often matching or exceeding the results of full fine-tuning. Its ability to preserve performance while reducing resource consumption makes it a practical choice for real-world applications.

3.4 Other Relevant Works

There have been multiple other approaches for PEFT, with the most prominent ones being: Prefix Tuning (Li & Liang, 2021), which prepends trainable token embeddings to the input sequence, Adapter Methods (Houlsby et al., 2019), which insert small, trainable neural modules within Transformer layers, and Prompt Tuning (Lester et al., 2021), which learns continuous prompt representations for specific tasks.

Moreover, there have been some very recent works tackling similar problems related to our work. In (Biderman et al., 2024), the authors show that in standard low-rank settings, LoRA substantially underperforms full finetuning in tasks like programming and mathematics. In (Shuttleworth et al., 2024), work on answering the question *are the solutions learned by LoRA and full-finetuning really equivalent?*. They show how LoRA learns an intruder dimension, and despite achieving similar performance to full fine-tuning, they become a worse model of their pre-training distribution and also lack generality to other tasks.

4 Methodology

To systematically evaluate the effectiveness of QLoRA for domain-specific knowledge transfer and varying levels of task complexity, we designed a set of experiments. This section outlines the datasets, experimental setup, and implementation details utilized in our investigation.

Our methodology comprises two distinct approaches to investigate the knowledge transfer and generalization capabilities of QLoRA fine-tuning: (1) domain knowledge integration using the MMLU benchmark and (2) arithmetic task performance evaluation. This dual approach allows us to probe both the breadth of knowledge transfer across diverse domains and the depth of reasoning capabilities in a structured task environment. Figure 1 gives an overview of our approach.

4.1 Domain Knowledge Integration using MMLU

Task and Benchmark The MMLU benchmark, introduced by (Hendrycks et al., 2020), serves as a comprehensive test of general knowledge across 57 diverse subjects, ranging from humanities and social sciences to STEM fields. This breadth makes MMLU an ideal testbed for assessing the efficacy of knowledge transfer in language models.

Our approach introduces a novel knowledge generation strategy leveraging GPT-4o (OpenAI, 2024) as a teacher model for creating fine-tuning data for LLaMA (Touvron et al., 2023) (basically performing knowledge transfer)

Algorithm 1 Generative QnA Pair Generation

```
Teacher ← GPT-4o
MMLU_Domains ← Machine Learning, ...
for each domain in MMLU_Domains do
    Student ← LLaMA
    Prompt ← Generate comprehensive domain prompt
    QnA_Pairs ← Teacher.generate_pairs(Prompt)
    Filtered_QnA ← Quality_Filter(QnA_Pairs)
    Student.fine_tune(Filtered_QnA)
    Student.save_model(domain)
end for
```

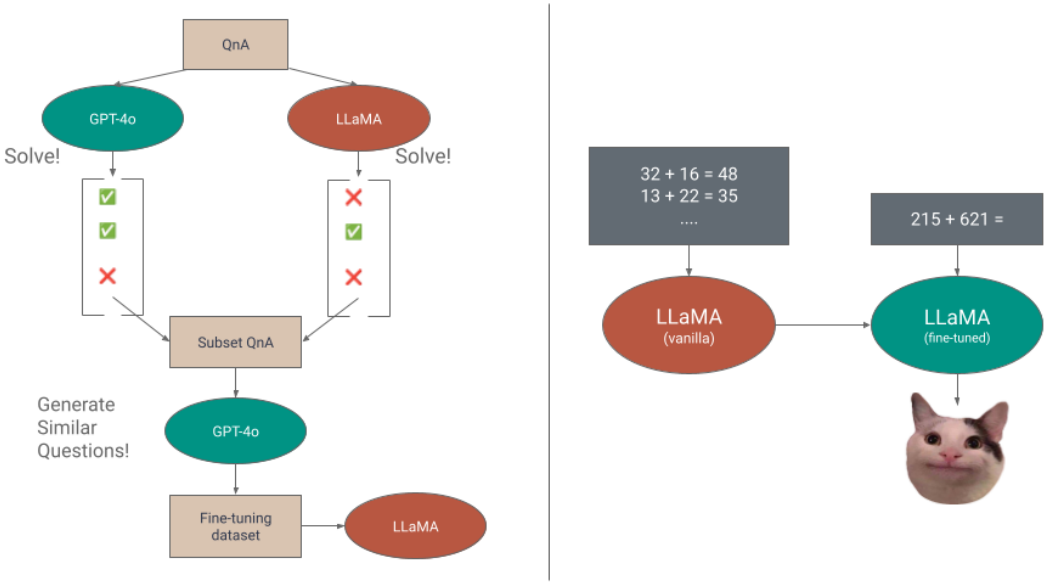
Method 1: Generative Data Approach The generative approach, as described in Algorithm 1, involves crafting domain-specific prompts for GPT-4o, generating question-answer pairs across MMLU domains, and applying a multi-stage quality filtering mechanism. This mechanism includes semantic relevance scoring, redundancy elimination, and retaining only solvable questions.

Algorithm 2 Targeted Knowledge QnA Generation

```
Teacher ← GPT-4o
Test_Set ← MMLU
for each question in Test_Set do
    Student ← LLaMA
    Teacher_Response ← Teacher.solve(question)
    Student_Response ← Student.solve(question)
    if Teacher_Response ≠ Student_Response then
        Knowledge_Gap ← Analyze_Difference
        New_QnA_Pairs ← Generate_Targeted_Pairs(
                Knowledge_Gap)
        Student.fine_tune(New_QnA_Pairs)
    end if
end for
```

Method 2: Targeted Knowledge Transfer The targeted approach, as outlined in Algorithm 2, identifies specific knowledge gaps where LLaMA exhibits deficiencies (LLaMA solves incorrectly and GPT-4o solves correctly), generating targeted question-answer pairs using GPT-4o as a teacher model, and fine-tuning the model with the newly generated data.

Figure 1. Overview of the two experiments we designed. Left: Fine tuning dataset creation process for targeted knowledge transfer, Right: Length generalizability of fine-tuned models on k-digit addition task.



The key distinction between these approaches lies in the specificity of knowledge transfer. While the first method generates a broad range of domain-specific QnA pairs, the second method focuses on addressing precise knowledge gaps identified through comparative performance analysis. It's worth noting that the second approach intentionally introduces a degree of data leakage, about which we will talk in our evaluation and discussion.

4.2 Arithmetic Task Generalization

To further evaluate the generalization capabilities of QLoRA, we conducted an in-depth analysis of the model's arithmetic performance, by testing it on k-digit addition task. This structured task environment allows us to probe the model's ability to learn and generalize algorithmic reasoning.

Data Generation We use the following methodology for generating arithmetic training (fine-tuning) and evaluation datasets:

1. **Uniform random sampling:** We generate k-digit numbers using uniform random sampling to ensure a diverse range of arithmetic problems.
2. **Precise digit length control:** Our generation process allows for exact control over the number of digits in each operand, enabling us to create datasets of varying complexity.

4.2.1 Baseline Performance

We first establish a baseline by evaluating the vanilla LLaMA model's performance on k-digit addition tasks. This initial assessment provides a benchmark for measuring the impact of fine-tuning.

4.2.2 Fine-tuning Procedure

The fine-tuning procedure involves generating random k-digit addition samples, fine-tuning the LLaMA model using these samples, and evaluating the resulting performance improvements on arithmetic tasks.

4.2.3 Generalization Assessment

The next aspect of our methodology is assessing length generalization by fine-tuning the model on k-digit addition, testing its performance on k+1 and higher digit additions, and analyzing its ability to generalize beyond the training distribution.

5 Evaluation

5.1 Experimental Setup and Preprocessing

5.1.1 Model Architecture

We utilize the LLaMA 2 7B parameter model as our base model, chosen for its balance between computational efficiency and performance. The model was initialized with

pre-trained weights from the HuggingFace Transformers library, using the default configuration without any initial modifications.

5.1.2 Evaluation Settings

- MMLU subsets: Machine learning and Professional Psychology
- Length generalization: Trained on 1-7 digit addition and tested on all (1-12 digits)
- GPU used: 1 x T4

5.1.3 Fine-tuning Configuration

QLoRA fine-tuning was implemented with the following hyperparameters:

Table 1. LoRA Configuration Parameters

PARAMETER	VALUE
RANK OF LORA (r)	32
LORA DROPOUT	0.1
QUANTIZATION	4-BIT NF4
NUMBER OF EPOCHS	1

5.2 Domain Knowledge Integration using MMLU

First, we go over the performance of our models on the ML subset of the MMLU benchmark. Table 2 shows the results from our experiments. The GPT-4o model outperforms the vanilla LLaMA performance, and hence can be rightfully used as a teacher model. Here, we can see how the domain-specific fine-tuning approach, particularly the "ML FT" model, outperforms both the vanilla LLaMA and the general fine-tuning (General FT) approach. However, the improvement that we see is not really what you would expect from leaking the test data to fine-tuning set. One could attribute that increasing the number of epochs would improve the performance, however based on our experiments, the performance actually started to degrade after the first epoch.

Similarly, Table 3 illustrates the results for the Professional Psychology subset of the MMLU benchmark. Here, the targeted Psychology FT model does not show a similar level of improvement (instead, the performance degrades here), achieving an accuracy of 0.30, much lower than the vanilla model's accuracy of 0.42 and the General FT's accuracy of 0.39. This result suggests that the transfer of domain-specific knowledge might vary significantly based on the task complexity and the inherent knowledge structure within the domain.

Table 2. Accuracy on MMLU (Machine Learning) dataset

MODEL	ACCURACY
GPT-4O	0.78
LLAMA (VANILLA)	0.30
LLAMA (GENERAL FT)	0.31
LLAMA (ML FT)	0.36
LLAMA (PSYCHOLOGY FT)	0.29

Table 3. Accuracy on MMLU (Professional Psychology) dataset

MODEL	ACCURACY
GPT-4O	0.88
LLAMA (VANILLA)	0.42
LLAMA (GENERAL FT)	0.39
LLAMA (PSYCHOLOGY FT)	0.30
LLAMA (ML FT)	0.37

Discussion of Results Based on the performance on the two subsets of MMLU in our experimental setting, we can conclude the following:

1. *Domain-Specific Fine-Tuning Effectiveness:* In domains with highly structured and well-defined knowledge bases, such as Machine Learning, targeted fine-tuning shows marked improvements. However, in more nuanced and less deterministic domains like Professional Psychology, the gains are not as pronounced, suggesting that the nature of the domain significantly impacts the effectiveness of fine-tuning strategies.

2. *Generalization vs. Memorization*: The General FT model performs comparably across both domains, indicating that general fine-tuning leads to a balanced but unspecialized improvement. However, this approach does not excel in either domain, which aligns with the hypothesis that general fine-tuning may prioritize broader generalization over domain-specific knowledge integration.

3. *Teacher-Student Knowledge Transfer*: The GPT-4o teacher model consistently outperforms all LLaMA variants, underscoring the limitations of fine-tuning approaches in fully bridging the gap between pre-trained and state-of-the-art models. While the generative and targeted approaches improve student performance, they fall short of achieving parity with the teacher model's understanding, highlighting inherent limitations in knowledge transfer, atleast when done through QLoRA.

4. *Task Complexity and Performance Variance*: The observed differences in accuracy across domains also suggest that task complexity plays a significant role in fine-tuning results. For instance, the structured nature of Machine Learning tasks might make them more suitable to improvement through fine-tuning, whereas the interpretative and subjec-

tive nature of Professional Psychology tasks could limit the efficacy of similar strategies.

5.3 Arithmetic Task Generalization

To evaluate the generalization capabilities of QLoRA on arithmetic tasks, we conducted experiments on k-digit addition, with the dataset ranging from 1-digit to 12-digit addition problems (carry-over allowed in the solutions). This evaluation enables us to systematically probe the model's ability to handle varying levels of task complexity and assess its generalization to unseen tasks.

Baseline and Fine-Tuning Performance Figure 2 illustrates the model's accuracy across different k-digit addition tasks. The performance of the vanilla LLaMA model (without fine-tuning) reveals significant limitations, with accuracy degrading sharply for addition tasks beyond 6 digits. This result highlights the inherent challenges of arithmetic reasoning in pre-trained language models without domain-specific fine-tuning.

Following this baseline assessment, we fine-tuned the model on k-digit addition tasks for $k \in [1, 7]$. The fine-tuned model demonstrated a notable improvement in accuracy, achieving higher performance for up to 8-digit addition tasks. However, two critical observations come up from the results:

1. *k-digit addition does not length generalize*: Despite fine-tuning on tasks up to 7 digits, the model's performance beyond 9-digit addition remains significantly impaired, with accuracy approaching 0%. This finding indicates that the model struggles to extrapolate beyond the patterns observed during training, underscoring a key limitation in its generalization ability. While the overall performance is still low, it can be improved by following the techniques used in (Zhou et al., 2023).

2. *Signs of emergent ability*: For tasks involving more than 9-digit addition, the model's near-zero accuracy raises the question of whether the model is exhibiting emergent abilities. Prior work by (Wei et al., 2022) highlighted that emergent abilities might manifest when model size or task complexity crosses a certain threshold. However, (Schaeffer et al., 2023) argued that such observations are often artifacts of the chosen evaluation metric.

Refined Metrics for Emergent Ability Analysis To address this, we replaced the binary accuracy metric with two continuous metrics: - *Digit Accuracy*: Measures the proportion of correctly predicted digits in the final answer. - *Edit Distance (Levenshtein Distance)*: Captures the number of edits required to transform the model's predicted answer into the correct answer, providing a graded measure of performance.

Figure 3 presents the results based on these refined metrics. Unlike the binary accuracy metric, which indicated a sharp drop in performance, these metrics reveal a more nuanced picture of the model's capabilities. The absence of clear discontinuities in digit accuracy and edit distance supports the argument from (Schaeffer et al., 2023), suggesting that signs of emergent abilities are artifacts of overly discrete evaluation criteria. The results with digit accuracy as the metric were very similar.

Insights and Implications The findings from this evaluation highlight both the strengths and limitations of QLoRA in arithmetic task generalization: (1) Fine-tuning improves performance within the range of observed training data but fails to generalize to tasks of greater complexity. (2) Continuous metrics provide a deeper understanding of model behavior, revealing gradual improvements rather than abrupt emergent abilities.

6 Conclusion

In this work, we explored the capabilities and limitations of QLoRA as a parameter-efficient fine-tuning approach for integrating domain-specific knowledge (through MMLU dataset) and generalizing on algorithmic tasks.

The findings reveal insights into the functional complexity limitations of QLoRA. While it excels in low-complexity, narrowly defined domains such as modifying chatbot behavior or instruction-following tasks, it struggles with more complex, knowledge-intensive fields. This limitation is evident in our MMLU evaluation, where fine-tuned models demonstrated only marginal improvements over their vanilla counterparts despite test data leakage during fine tuning, highlighting the difficulty in effectively introducing new domain-specific knowledge.

Similarly, our arithmetic evaluation underscores a fundamental challenge in length generalization. Fine-tuning LLaMA on k-digit addition significantly improved performance within the training range but showed clear degradation for higher-digit tasks. This limitation suggests a lack of inherent algorithmic reasoning capabilities, which could only be partially mitigated through fine-tuning (perhaps the improvement was just because of memorization which is highly possible in a task like addition). Moreover, the apparent emergent behaviors observed during evaluation were debunked as artifacts of discrete accuracy metrics, emphasizing the importance of adopting more nuanced, continuous evaluation methods such as digit accuracy and Levenshtein distance.

Our analysis further highlights the memorization patterns in fine-tuned models, suggesting that current fine-tuning techniques often optimize for reproducing specific training pat-

Figure 2. k-digit Addition Accuracy

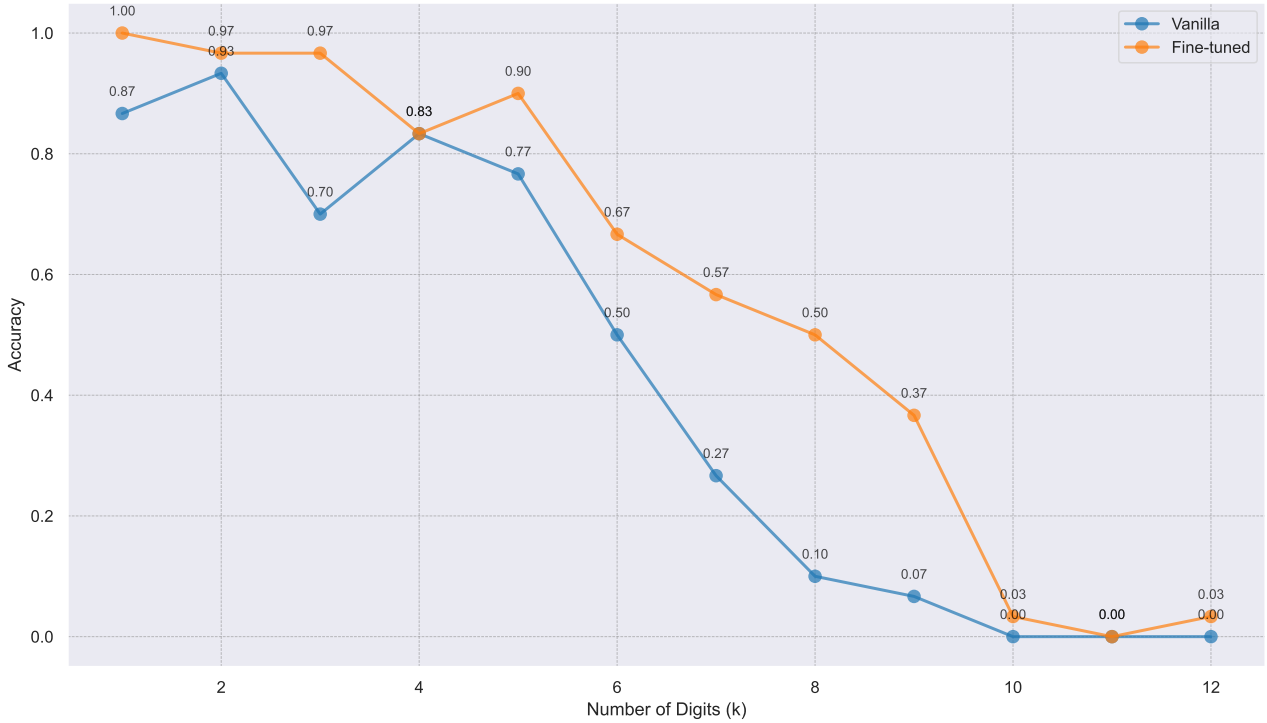
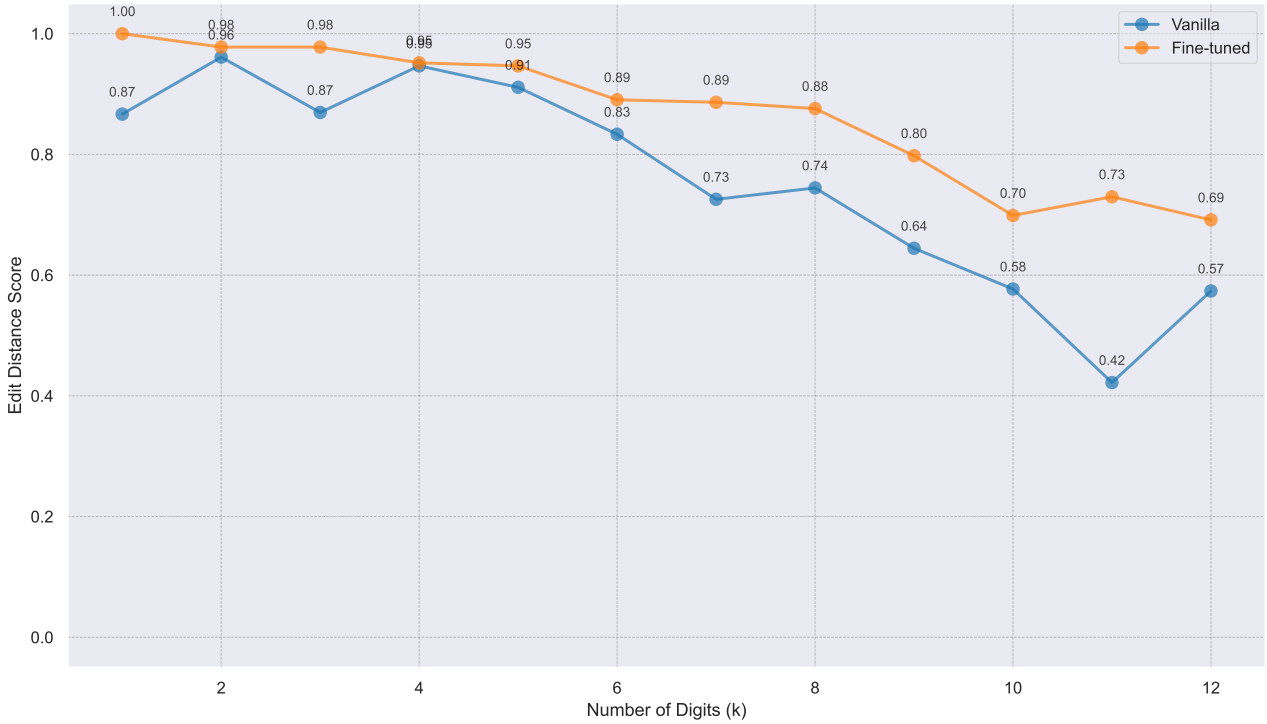


Figure 3. k-digit Addition (Levenshtein Distance metric)



terns rather than developing robust, transferable knowledge. This finding aligns with our identification of boundary conditions for effective knowledge integration, where QLoRA is more successful in narrow, well-structured domains but faces significant challenges in broader, complex domains requiring genuine understanding and generalization.

In conclusion, while QLoRA remains a practical and resource-efficient solution for many tasks, its efficacy diminishes as functional and domain complexity increases. Future work should focus on designing fine-tuning methods that balance efficiency with the ability to generalize and integrate knowledge in complex and dynamic settings.

REFERENCES

Biderman, D., Portes, J., Ortiz, J. J. G., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., Chiley, V., Frankle, J., Blakeney, C., and Cunningham, J. P. Lora learns less and forgets less, 2024. URL https://arxiv.org/abs/2405.09673.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms, 2023. URL https://arxiv.org/abs/2305.14314.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300, 2020. URL https://arxiv.org/abs/2009.03300.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. *CoRR*, abs/1902.00751, 2019. URL http://arxiv.org/abs/1902.00751.

Howard, J. and Ruder, S. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018. URL http://arxiv.org/abs/1801.06146.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. URL http://arxiv.org/abs/1612.00796.

Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z.-R., Stevens, K., Barhoum, A., Duc, N. M., Stanley, O., Nagyfi, R., ES, S., Suri, S., Glushkov, D., Dantuluri, A., Maguire, A., Schuhmann, C., Nguyen, H., and Mattick, A. Openassistant conversations – democratizing large language model alignment, 2023. URL https://arxiv.org/abs/2304.07327.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691, 2021. URL https://arxiv.org/abs/2104.08691.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *CoRR*, abs/2101.00190, 2021. URL https://arxiv.org/abs/2101.00190.

OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Schaeffer, R., Miranda, B., and Koyejo, S. Are emergent abilities of large language models a mirage?, 2023. URL https://arxiv.org/abs/2304.15004.

Shuttleworth, R., Andreas, J., Torralba, A., and Sharma, P. Lora vs full fine-tuning: An illusion of equivalence, 2024. URL https://arxiv.org/abs/2410.21228.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. Emergent abilities of large language models, 2022. URL https://arxiv.org/abs/2206.07682.

Zaken, E. B., Ravfogel, S., and Goldberg, Y. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *CoRR*, abs/2106.10199, 2021. URL https://arxiv.org/abs/2106.10199.

Zhou, H., Bradley, A., Littwin, E., Razin, N., Saremi, O., Susskind, J., Bengio, S., and Nakkiran, P. What algorithms can transformers learn? a study in length generalization, 2023. URL https://arxiv.org/abs/2310.16028.

A PROMPTS USED

```
You are an expert in {format_subject(dataset_name)}. You will be presented with a question and four possible answers. Choose the correct answer. Do not output any explanation, only choose the correct option.
Question: ...
Answer:
```

Listing 1. Prompt for evaluating GPT-4o/LLaMA performance on MMLU

```
You are generating data which will be used to train a machine learning model.\n\nYou will be given a high-level description of the model we want to train, and from that, you will generate data samples, each with a prompt/response pair.\n\nYou will do so in this format:\n```\n<prompt>prompt</prompt>\n<response>response_goes_here</response>\n```\n\nOnly one prompt/response pair should be generated per turn.\n\nFor each turn, make the example slightly more complex than the last, while ensuring diversity.\n\nMake sure your samples are unique and diverse, yet high-quality and complex enough to train a well-performing model.\n\nHere is the type of model we want to train:\n`A model that takes in a question in the field of 'professional medicine' and is given with a list of possible answers. The models produces an answer that is most likely to be correct. The model should just generate the answer and not provide any explanation or reasoning.`. Now, generate a prompt/response pair. Do so in the exact format requested:\n```\n<prompt>prompt</prompt>\n<response>response_goes_here</response>\n```\n\nOnly one prompt/response pair should be generated per turn.
```

Listing 2. Prompt for generating general dataset for fine-tuning by GPT-4o

```

You are an advanced AI model tasked with assisting in creating training data for smaller AI models.

The task of the smaller model is to take in a question in the field of {dataset_name} and a list of possible answers, and produce an answer that is most likely to be correct, without providing any explanation or reasoning.

When given a question, you should:

1. Analyze the knowledge or reasoning required to answer it correctly.

2. Generate a new Q&A pair that tests the same knowledge or reasoning skills.

The objective is that the smaller model should be able to answer the original question correctly if it can answer the new question correctly.
```

```
    Ensure that:

    - The information present in the
    original question is also present in
    the new question.
    - The new question has exactly 4 answer
     options (A, B, C, D). Mention these
    options in the question.
    - The correct answer is clearly
    indicated in the format shown below.

    Use the following format strictly:
    ```
    <question>Question text with options</
    question>
    <answer>Correct answer (e.g., A or B)</
    answer>
    ```

    Do not include explanations, reasoning,
     or any additional text.

    Here is a question that was correctly
    answered by a large model but
    incorrectly answered by a smaller model
    :
    {question} and its answer: {answer}

    Please generate a new Q&A pair
    following the instructions. Only one
    question/answer pair should be
    generated per turn.
```

Listing 3. Prompt for targetted dataset for fine-tuning by GPT-4o

```
You are an expert in solving math problems.
     You will be given two numbers and you
    need to add them. Do not produce any
    output other than the sum of the two
    numbers. You will also be given
    examples to help you understand the
    task.
Question: ...
Answer:
```

Listing 4. Prompt for evaluating GPT-4o/LLaMA performance on arithmetic