hand, the fine-tuning model with the CodeExercise-Python-27k dataset resulted in slight performance drops, with scores decreasing by 4.2% on HumanEval and 2.4% on HumanEval+. This contrast suggests that the success of fine-tuning largely depends on the underlying properties of the dataset being used.

Despite CodeExercise-Python-27k covering 27,000 exercises across a range of coding topics—from beginner syntax to machine learning—the content was generated by a teacher model and Camel, without much human oversight. This approach is likely to provide incorrect answers in response to given questions. Such issues may have restricted the model's ability to generalize effectively, increasing the risk of overfitting and reducing its flexibility. These findings highlight the critical role of high-quality, well-structured, and consistently formatted datasets in the fine-tuning of large language models.

IV. Conclusion

This study presents a parameter-efficient fine-tuning approach using QLoRA to improve the performance of compact large language models for code generation tasks in resource-constrained environments. By applying QLoRA to the Qwen2.5-Coder-0.5B-Instruct model, we demonstrated the potential of lightweight fine-tuning methods to significantly enhance model adaptability and accuracy, while maintaining low memory usage and computational efficiency. The findings emphasize the importance of high-quality, instruction-aligned datasets in maximizing the benefits of such fine-tuning strategies. These contributions support the broader applicability of compact LLMs in domains like AI-assisted learning, educational platforms, and low-resource development settings. Future work may explore the impact of additional hyperparameter tuning and the integration of more diverse, human-validated datasets to improve generalization and robustness across coding tasks further.

References

[1] U. Durrani *et al.*, "A Decade of Progress: A Systematic Literature Review on the Integration of AI in Software Engineering Phases and Activities (2013-2023)," *IEEE Access*, p. 1, 2024, doi: 10.1109/access.2024.3488904.

[2] K. Qiu, N. Puccinelli, M. Ciniselli, and L. Di Grazia, "From Today's Code to Tomorrow's Symphony: The AI Transformation of Developer's Routine by 2030," *ACM Transactions on Software Engineering and Methodology*, 2024, doi: 10.1145/3709353.

[3] L. Fattorini, "Artificial Intelligence Index Report 2023," *Artificial Intelligence Index Report*, 2023.

[4] K. Misiejuk, R. Kaliisa, and J. Scianna, "Augmenting assessment with AI coding of online student discourse: A question of reliability," *Computers & Education: Artificial Intelligence*, vol. 6, p. 100216, 2024, doi: 10.1016/j.caeai.2024.100216.

[5] H. Ghaemi, Z. Alizadehsani, A. Shahraki, and J. M. Corchado, "Transformers in source code generation: A comprehensive survey," *Journal of Systems Architecture*, p. 103193, 2024, doi: 10.1016/j.sysarc.2024.103193.

[6] M. R. Lyu, B. Ray, A. Roychoudhury, S. H. Tan, and P. Thongtanunam, "Automatic Programming: Large Language Models and Beyond," *ACM Transactions on Software Engineering and Methodology*, 2024, doi: 10.1145/3708519.

[7] S. Pooja, C. B. Chandrakala, and L. K. Raju, "Developer's Roadmap to Design Software Vulnerability Detection Model Using Different AI Approaches," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3191115.

[8] S. Martínez-Fernández *et al.*, "Software Engineering for AI-Based Systems: A Survey," *ACM Transactions on Software Engineering and Methodology*, vol. 31, no. 2, 2022, doi: 10.1145/3487043.

[9] B. Rozière *et al.*, "Code Llama: Open Foundation Models for Code," pp. 1–48, 2023, [Online]. Available: http://arxiv.org/abs/2308.12950

[10] B. Hui *et al.*, "Qwen2.5-Coder Technical Report," pp. 1–23, 2024, [Online]. Available: http://arxiv.org/abs/2409.12186

[11] Z. Feng *et al.*, "CodeBERT: A pre-trained model for programming and natural languages," in *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, 2020. doi: 10.18653/v1/2020.findings-emnlp.139.

[12] M. A. K. Raiaan *et al.*, "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," *IEEE Access*, doi: 10.1109/access.2024.3365742.

[13] X. Chen, C. Gao, C. Chen, G. Zhang, and Y. Liu, "An Empirical Study on Challenges for LLM Application Developers," *ACM Transactions on Software Engineering and Methodology*, 2025, doi: 10.1145/3715007.

[14] D. Cotroneo, A. Foggia, C. M. Improta, P. Liguori, and R. Natella, "Automating the Correctness Assessment of AI-generated Code for Security Contexts," *Journal of Systems and Software*, vol. abs/2310.18834, 2023, doi: 10.48550/arxiv.2310.18834.

[15] M. Weyssow, X. Zhou, K. Kim, D. Lo, and H. Sahraoui, "Exploring Parameter-Efficient Fine-Tuning Techniques for Code Generation with Large Language Models," *ACM Transactions on Software Engineering and Methodology*, Jan. 2025, doi: 10.1145/3714461.

[16] Z. R. K. Rostam and S. Szénási, "Achieving Peak Performance for Large Language Models: A Systematic Review," *IEEE Access*, p. 1, 2024, doi: 10.1109/access.2024.3424945.

[17] E. Hu *et al.*, "LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS," in *ICLR 2022 - 10th International Conference on Learning Representations*, 2022.

[18] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," May 2023, [Online]. Available: http://arxiv.org/abs/2305.14314

[19] E. Nijkamp *et al.*, "CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.13474

[20] Y. Wang, H. Le, A. D. Gotmare, N. D. Q. Bui, J. Li, and S. C. H. Hoi, "CodeT5+: Open Code Large Language Models for Code Understanding and Generation," May 2023, [Online]. Available: http://arxiv.org/abs/2305.07922

[21] S. Ren *et al.*, "CodeBLEU: a Method for Automatic Evaluation of Code Synthesis," Sep. 2020, [Online]. Available: http://arxiv.org/abs/2009.10297

[22] M. Chen *et al.*, "Evaluating Large Language Models Trained on Code," Jul. 2021, [Online]. Available: http://arxiv.org/abs/2107.03374

[23] J. Liu, C. S. Xia, Y. Wang, and L. Zhang, "Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, in NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2023.

[24] B. Liu *et al.*, "MFTCoder: Boosting Code LLMs with Multitask Fine-Tuning," Nov. 2023, [Online]. Available: http://arxiv.org/abs/2311.02303

[25] N. Mejia Petit, "Vezora/Tested-22k-Python-Alpaca," Hugging Face, [Online]. Available: https://huggingface.co/datasets/Vezora/Tested-22k-Python-Alpaca

[26] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "CAMEL: Communicative Agents for 'Mind' Exploration of Large Language Model Society," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.17760

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, in ACL '02. USA: Association for Computational Linguistics, 2002, pp. 311–318. doi: 10.3115/1073083.1073135.