Chen, L., Guo, Q., Jia, H., Zeng, Z., Wang, X., Xu, Y., ... & Zhang, S. (2024). A survey on evaluating large language models in code generation tasks. *arXiv preprint arXiv:2408.16498*.

<https://doi.org/10.48550/arXiv.2408.16498>

Haque, M. A. (2025). Llms: A game-changer for software engineers?. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, 100204. <https://doi.org/10.1016/j.tbench.2025.100204>

Zhang, Y., Liang, G., Salem, T., & Jacobs, N. (2019, December). Defense-pointnet: Protecting pointnet against adversarial attacks. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 5654-5660). IEEE. <https://doi.org/10.1109/BigData47090.2019.9006307>

Deanda, D., Alsmadi, I., Guerrero, J., & Liang, G. (2025). Defending mutation-based adversarial text perturbation: a black-box approach. *Cluster Computing*, *28*(3), 196.

He, X., Huang, W., & Lv, C. (2024). Trustworthy autonomous driving via defense-aware robust reinforcement learning against worst-case observational perturbations. *Transportation Research Part C: Emerging Technologies*, *163*, 104632. <https://doi.org/10.1016/j.trc.2024.104632>

Deanda, D., Masupalli, Y. P., Yang, J., Lee, Y., Cao, Z., & Liang, G. (2025). Benchmarking robustness of contrastive learning models for medical image-report retrieval. *arXiv preprint arXiv:2501.09134*.

<https://doi.org/10.48550/arXiv.2501.09134>

Qiu, R., Zeng, W. W., Ezick, J., Lott, C., & Tong, H. (2024). How efficient is llm-generated code? a rigorous & high-standard benchmark. *arXiv preprint arXiv:2406.06647*.

<https://doi.org/10.48550/arXiv.2406.06647>

Coignion, T., Quinton, C., & Rouvoy, R. (2024, June). A performance study of llm-generated code on leetcode. In *Proceedings of the 28th international conference on evaluation and assessment in software engineering* (pp. 79-89). <https://doi.org/10.1145/3661167.3661221>

Huang, D., Qing, Y., Shang, W., Cui, H., & Zhang, J. M. (2024). Effibench: Benchmarking the efficiency of automatically generated code. *Advances in Neural Information Processing Systems*, *37*, 11506-11544.

Guo, L., Wang, Y., Shi, E., Zhong, W., Zhang, H., Chen, J., ... & Zheng, Z. (2024, September). When to stop? towards efficient code generation in llms with excess token prevention. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis* (pp. 1073-1085). <https://doi.org/10.1145/3650212.3680343>

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., ... & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, *43*(2), 1-55. <https://doi.org/10.1145/3703155>

Ehsani, R., Pathak, S., & Chatterjee, P. (2025, April). Towards detecting prompt knowledge gaps for improved llm-guided issue resolution. In *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)* (pp. 699-711). IEEE. <https://doi.org/10.1109/MSR66628.2025.00107>

Khojah, R., de Oliveira Neto, F. G., Mohamad, M., & Leitner, P. (2025). The Impact of Prompt Programming on Function-Level Code Generation. *IEEE Transactions on Software Engineering*. <https://doi.org/10.1109/TSE.2025.3587794>

Solovyeva, L., Weidmann, S., & Castor, F. (2025, April). Ai-powered, but power-hungry? energy efficiency of llm-generated code. In *2025 IEEE/ACM Second International Conference on AI Foundation Models and Software Engineering (Forge)* (pp. 49-60). IEEE. <https://doi.org/10.1109/Forge66646.2025.00012>

Bolón-Canedo, V., Morán-Fernández, L., Cancela, B., & Alonso-Betanzos, A. (2024). A review of green artificial intelligence: Towards a more sustainable future. *Neurocomputing*, *599*, 128096. <https://doi.org/10.1016/j.neucom.2024.128096>

Paul, D. G., Zhu, H., & Bayley, I. (2024, July). Benchmarks and metrics for evaluations of code generation: A critical review. In *2024 IEEE International Conference on Artificial Intelligence Testing (AITest)* (pp. 87-94). IEEE. <https://doi.org/10.1109/AITest62860.2024.00019>

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., ... & Zaremba, W. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., ... & Sutton, C. (2021). Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Du, M., Luu, A. T., Ji, B., Liu, Q., & Ng, S. K. (2024). Mercury: A code efficiency benchmark for code large language models. *Advances in Neural Information Processing Systems*, *37*, 16601-16622.

Hou, W., & Ji, Z. (2025). Comparing large language models and human programmers for generating programming code. *Advanced Science*, *12*(8), 2412279. <https://doi.org/10.1002/advs.202412279>

Dou, S., Jia, H., Wu, S., Zheng, H., Zhou, W., Wu, M., ... & Huang, X. (2024). What's wrong with your code generated by large language models? an extensive study. *arXiv preprint arXiv:2407.06153*.

<https://doi.org/10.48550/arXiv.2407.06153>

Niu, C., Zhang, T., Li, C., Luo, B., & Ng, V. (2024, April). On evaluating the efficiency of source code generated by llms. In *Proceedings of the 2024 IEEE/ACM First International Conference on AI Foundation Models and Software Engineering* (pp. 103-107). <https://doi.org/10.1145/3650105.3652295>

Dou, S., Jia, H., Wu, S., Zheng, H., Zhou, W., Wu, M., ... & Huang, X. (2024). What's wrong with your code generated by large language models? an extensive study. *arXiv preprint arXiv:2407.06153*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, 24824-24837.

Steiss, J., Tate, T., Graham, S., Cruz, J., Hebert, M., Wang, J., ... & Olson, C. B. (2024). Comparing the quality of human and ChatGPT feedback of students’ writing. *Learning and Instruction*, *91*, 101894. <https://doi.org/10.1016/j.learninstruc.2024.101894>

Alier, M., Peñalvo, F. J. G., & Camba, J. D. (2024). Generative Artificial Intelligence in Education: From Deceptive to Disruptive. *International Journal of interactive multimedia and artificial intelligence*, *8*(5), 5-14. <https://doi.org/10.9781/ijimai.2024.02.011>

Bozkurt, A. (2023). Generative artificial intelligence (AI) powered conversational educational agents: The inevitable paradigm shift. *Asian Journal of Distance Education*, *18*(1). <https://orcid.org/0000-0002-4520-642X>

Cao, L., & Dede, C. (2023). Navigating a world of generative AI: Suggestions for educators. *The next level lab at harvard graduate school of education*, *5*(2).

Pesovski, I., Santos, R., Henriques, R., & Trajkovik, V. (2024). Generative AI for customizable learning experiences. *Sustainability*, *16*(7), 3034. <https://doi.org/10.3390/su16073034>

Khazanchi, R. A. S. H. M. I., & Khazanchi, P. A. N. K. A. J. (2024). Generative AI to improve special education teacher preparation for inclusive classrooms. *Exploring new horizons: Generative artificial intelligence and teacher education*, *159*.

Doughty, J., Wan, Z., Bompelli, A., Qayum, J., Wang, T., Zhang, J., ... & Sakr, M. (2024, January). A comparative study of AI-generated (GPT-4) and human-crafted MCQs in programming education. In *Proceedings of the 26th Australasian Computing Education Conference* (pp. 114-123). <https://doi.org/10.1145/3636243.3636256>

Weisz, J. D., He, J., Muller, M., Hoefer, G., Miles, R., & Geyer, W. (2024, May). Design principles for generative AI applications. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (pp. 1-22). <https://doi.org/10.1145/3613904.3642466>

Dave, D. M., Mandvikar, S., & Engineer, P. A. (2023). Augmented intelligence: Human-AI collaboration in the era of digital transformation. *International Journal of Engineering Applied Sciences and Technology*, *8*(6), 24-33.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

Zhang, Z., Zhang, A., Li, M., & Smola, A. (2022). Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Walter, Y. (2024). Embracing the future of Artificial Intelligence in the classroom: the relevance of AI literacy, prompt engineering, and critical thinking in modern education. *International Journal of Educational Technology in Higher Education*, *21*(1), 15. <https://doi.org/10.1186/s41239-024-00448-3>

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, 24824-24837.

Zhang, Z., Zhang, A., Li, M., & Smola, A. (2022). Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*. <https://doi.org/10.48550/arXiv.2203.11171>

Zhao, X., Li, M., Lu, W., Weber, C., Lee, J. H., Chu, K., & Wermter, S. (2023). Enhancing zero-shot chain-of-thought reasoning in large language models through logic. *arXiv preprint arXiv:2309.13339*.

https://doi.org/10.48550/arXiv.2309.13339

Liu, X., Wang, J., Sun, J., Yuan, X., Dong, G., Di, P., ... & Wang, D. (2023). Prompting frameworks for large language models: A survey. *arXiv preprint arXiv:2311.12785*.

Lo, L. S. (2023). The CLEAR path: A framework for enhancing information literacy through prompt engineering. *The Journal of Academic Librarianship*, *49*(4), 102720. <https://doi.org/10.1016/j.acalib.2023.102720>

Rasnayaka, S., Wang, G., Shariffdeen, R., & Iyer, G. N. (2024, April). An empirical study on usage and perceptions of llms in a software engineering project. In *Proceedings of the 1st International Workshop on Large Language Models for Code* (pp. 111-118). <https://doi.org/10.1145/3643795.3648379>

Zeng, Z., Tan, H., Zhang, H., Li, J., Zhang, Y., & Zhang, L. (2022, July). An extensive study on pre-trained models for program understanding and generation. In *Proceedings of the 31st ACM SIGSOFT international symposium on software testing and analysis* (pp. 39-51). <https://doi.org/10.1145/3533767.3534390>

Xia, C. S., Wei, Y., & Zhang, L. (2022). Practical program repair in the era of large pre-trained language models. *arXiv preprint arXiv:2210.14179*. <https://doi.org/10.48550/arXiv.2210.14179>

Cao, J., Li, M., Wen, M., & Cheung, S. C. (2025). A study on prompt design, advantages and limitations of chatgpt for deep learning program repair. *Automated Software Engineering*, *32*(1), 1-29. <https://doi.org/10.1007/s10515-025-00492-x>

Nichols, B., Buttlar, D., & Farrell, J. (1996). *Pthreads programming: A POSIX standard for better multiprocessing*. " O'Reilly Media, Inc.".

Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., ... & Wu, Y. (2023). How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Chen, L., Lin, P. H., Vanderbruggen, T., Liao, C., Emani, M., & De Supinski, B. (2023, September). Lm4hpc: Towards effective language model application in high-performance computing. In *International Workshop on OpenMP* (pp. 18-33). Cham: Springer Nature Switzerland. <https://doi.org/10.1007/978-3-031-40744-4_2>

Cahyawijaya, S., Winata, G. I., Wilie, B., Vincentio, K., Li, X., Kuncoro, A., ... & Fung, P. (2021). IndoNLG: Benchmark and resources for evaluating Indonesian natural language generation. *arXiv preprint arXiv:2104.08200*. <https://doi.org/10.48550/arXiv.2104.08200>

Ahmed, T., & Devanbu, P. (2022, October). Few-shot training llms for project-specific code-summarization. In *Proceedings of the 37th IEEE/ACM international conference on automated software engineering* (pp. 1-5). [https://doi.org/10.1145/3551349.355955](https://doi.org/10.1145/3551349.3559555)

Ding, X., Chen, L., Emani, M., Liao, C., Lin, P. H., Vanderbruggen, T., ... & Du, W. (2023, November). Hpc-gpt: Integrating large language model for high-performance computing. In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis* (pp. 951-960). <https://doi.org/10.1145/3624062.3624172>

Yang, Y., Zhu, Y., Chen, S., & Jian, P. (2023). API comparison knowledge extraction via prompt-tuned language model. *Journal of Computer Languages*, *75*, 101200. <https://doi.org/10.1016/j.cola.2023.101200>

Huang, Q., Yuan, Z., Xing, Z., Xu, X., Zhu, L., & Lu, Q. (2022, October). Prompt-tuned code language model as a neural knowledge base for type inference in statically-typed partial code. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering* (pp. 1-13). <https://doi.org/10.1145/3551349.3556912>

Nashid, N., Sintaha, M., & Mesbah, A. (2023, May). Retrieval-based prompt selection for code-related few-shot learning. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)* (pp. 2450-2462). IEEE. <https://doi.org/10.1109/ICSE48619.2023.00205>

Ahmed, T., & Devanbu, P. (2022, October). Few-shot training llms for project-specific code-summarization. In *Proceedings of the 37th IEEE/ACM international conference on automated software engineering* (pp. 1-5). <https://doi.org/10.1145/3551349.3559555>

Shi, Z., Mathur, U., & Pavlogiannis, A. (2024, April). Optimistic prediction of synchronization-reversal data races. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (pp. 1-13). <https://doi.org/10.1145/3597503.3639099>

Serebryany, K., Potapenko, A., Iskhodzhanov, T., & Vyukov, D. (2011, September). Dynamic race detection with LLVM compiler: Compile-time instrumentation for ThreadSanitizer. In *International Conference on Runtime Verification* (pp. 110-114). Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-29860-8_9>

Liew, D., Cogumbreiro, T., & Lange, J. (2024). Sound and partially-complete static analysis of data-races in gpu programs. *Proceedings of the ACM on Programming Languages*, *8*(OOPSLA2), 2434-2461. <https://doi.org/10.1145/3689797>

Choi, J. D., Lee, K., Loginov, A., O'Callahan, R., Sarkar, V., & Sridharan, M. (2002, May). Efficient and precise datarace detection for multithreaded object-oriented programs. In *Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation* (pp. 258-269). <https://doi.org/10.1145/512529.512560>

Malakar, S., Haider, T. B., & Shahriar, R. (2024). RaceFixer--An Automated Data Race Fixer. *arXiv preprint arXiv:2401.04221*. <https://doi.org/10.48550/arXiv.2401.04221>

Kadosh, T., Schneider, N., Hasabnis, N., Mattson, T., Pinter, Y., & Oren, G. (2023, September). Advising openmp parallelization via a graph-based approach with transformers. In *International Workshop on OpenMP* (pp. 3-17). Cham: Springer Nature Switzerland. <https://doi.org/10.1007/978-3-031-40744-4_1>

Chen, L., Bhattacharjee, A., Ahmed, N., Hasabnis, N., Oren, G., Vo, V., & Jannesari, A. (2024, August). Ompgpt: A generative pre-trained transformer model for openmp. In *European Conference on Parallel Processing* (pp. 121-134). Cham: Springer Nature Switzerland. <https://doi.org/10.1007/978-3-031-69577-3_9>

Mišić, M., & Dodović, M. (2024). An assessment of large language models for OpenMP-based code parallelization: a user perspective. *Journal of Big Data*, *11*(1), 161. <https://doi.org/10.1186/s40537-024-01019-z>

Alsofyani, M., & Wang, L. (2024, August). Detecting data races in openmp with deep learning and large language models. In *Workshop Proceedings of the 53rd International Conference on Parallel Processing* (pp. 96-103). <https://doi.org/10.1145/3677333.3678160>