#	Paper	Paper Type (Research / Education)	DOI/URL	SDF				SE								
				SDF -01	SDF -02	SDF -03	SDF -04	SE-01	SE-02	SE-03	SE-04	SE-05	SE-06	SE-07	SE-08	SE-09
1	First, E., Rabe, M., Ringer, T., & Brun, Y. (2023, November). Baldur: Whole-proof generation and repair with large language models. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1229-1241).	R	https://doi.org/10.1145/3611643.3616243				M						S			S
2	Grishina, A., Hort, M., & Moonen, L. (2023). The EarlyBIRD Catches the Bug: On Exploiting Early Layers of Encoder Models for More Efficient Code Classification. arXiv preprint arXiv:2305.04940.	R	https://arxiv.org/abs/2305.04940										S	S	S	
3	Zhang, J., Nie, P., Li, J. J., & Gligoric, M. (2023, November). Multilingual code co-evolution using large language models. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 695-707).	R	https://doi.org/10.1145/3611643.3616350											S	S	
4	Gao, H., Treude, C., & Zahedi, M. (2023, November). Evaluating Transfer Learning for Simplifying GitHub READMEs. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1548-1560).	R	https://doi.org/10.1145/3611643.3616291						M				M		M	
5	Zhang, Y., Xu, Z., Liu, C., Chen, H., Sun, J., Qiu, D., & Liu, Y. (2023, November). Software architecture recovery with information fusion. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1535-1547).	R	https://doi.org/10.1145/3611643.3616285								S			S		
6	Wang, S., Geng, M., Lin, B., Sun, Z., Wen, M., Liu, Y., ... & Mao, X. (2023, November). Natural Language to Code: How Far Are We?. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 375-387).	R	https://doi.org/10.1145/3611643.3616323	M								M				
7	Lin, B., Wang, S., Liu, Z., Liu, Y., Xia, X., & Mao, X. (2023). CCT5: A Code-Change-Oriented Pre-Trained Model. arXiv preprint arXiv:2305.10785.	R	https://arxiv.org/abs/2305.10785											S	M	
8	Li, H., Hao, Y., Zhai, Y., & Qian, Z. (2023, November). Assisting Static Analysis with Large Language Models: A ChatGPT Experiment. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 2107-2111).	R	https://doi.org/10.1145/3611643.3613078										S		M	
9	Gupta, P., Khare, A., Bajpai, Y., Chakraborty, S., Gulwani, S., Kanade, A., ... & Tiwari, A. (2023, November). Grace: Language Models Meet Code Edits. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1483-1495).	R	https://doi.org/10.1145/3611643.3616253				S							M	M	
10	Dong, J., Lou, Y., Hao, D., & Tan, L. (2023, May). Revisiting Learning-based Commit Message Generation. In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE) (pp. 794-805). IEEE.	R	https://doi.org/10.1109/ICSE48619.2023.00075					M	S					S		
11	Bouzenia, Islem, and Michael Pradel. "When to Say What: Learning to Find Condition-Message Inconsistencies." In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 868-880. IEEE, 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00081				S		M			S	S	S	S	
12	Lemieux, Caroline, Jeevana Priya Inala, Shuvendu K. Lahiri, and Siddhartha Sen. "CODAMOSA: Escaping coverage plateaus in test generation with pre-trained large language models." In International conference on software engineering (ICSE). 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00085										S		M	
13	Parasaram, Nikhil, Earl T. Barr, and Sergey Mechtaev. "Rete: Learning namespace representation for program repair." In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 1264-1276. IEEE, 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00112										S		M	
14	Ezzini, Saad, Sallam Abualhaija, Chetan Arora, and Mehrdad Sabetzadeh. "Ai-based question answering assistance for analyzing natural-language requirements." Proceedings of the 45th International Conference on Software Engineering May 2023Pages 1277–1289 (2023).	R	https://doi.org/10.1109/ICSE48619.2023.00113							S			S		M	
15	Fan, Zhiyu, Xiang Gao, Martin Mirchev, Abhik Roychoudhury, and Shin Hwei Tan. "Automated repair of programs from large language models. In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)." IEEE Computer Society, Los Alamitos, CA, USA (2023): 1469-1481.	R	https://doi.org/10.1109/ICSE48619.2023.00128										S		S	
16	Xia, Chunqiu Steven, Yuxiang Wei, and Lingming Zhang. "Automated program repair in the era of large pre-trained language models." In Proceedings of the 45th International Conference on Software Engineering (ICSE 2023). Association for Computing Machinery. 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00129										S		S	
17	Lee, Cheryl, Tianyi Yang, Zhuangbin Chen, Yuxin Su, Yongqiang Yang, and Michael R. Lyu. "Heterogeneous anomaly detection for software systems via semi-supervised cross-modal attention." arXiv preprint arXiv:2302.06914 (2023).	R	https://export.arxiv.org/abs/2302.06914										S		S	
18	Ahmed, Toufique, Supriyo Ghosh, Chetan Bansal, Thomas Zimmermann, Xuchao Zhang, and Saravan Rajmohan. "Recommending Root-Cause and Mitigation Steps for Cloud Incidents using Large Language Models." ICSE '23: Proceedings of the 45th International Conference on Software Engineering May 2023 Pages 1737–1749 (2023).	R	https://doi.org/10.1109/ICSE48619.2023.00149										S		S	
19	Nam, Daye, Brad Myers, Bogdan Vasilescu, and Vincent Hellendoorn. "Improving API Knowledge Discovery with ML: A Case Study of Comparable API Methods." In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 1890-1906. IEEE, 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00161				M				S	M				
20	Nie, Pengyu, Rahul Banerjee, Junyi Jessy Li, Raymond J. Mooney, and Milos Gligoric. "Learning Deep Semantics for Test Completion." 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)	R	https://doi.org/10.1109/ICSE48619.2023.00178										S		S	
21	Mastropaolo, Antonio, Luca Pascarella, Emanuela Guglielmi, Matteo Ciniselli, Simone Scalabrino, Rocco Oliveto, and Gabriele Bavota. "On the robustness of code generation techniques: An empirical study on github copilot." 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)	R	https://doi.org/10.1109/ICSE48619.2023.00181						S			S				
22	Shi, Ensheng, Yanlin Wang, Wenchao Gu, Lun Du, Hongyu Zhang, Shi Han, Dongmei Zhang, and Hongbin Sun. "Cocosoda: Effective contrastive learning for code search." In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 2198-2210. IEEE, 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00185									S		S		
23	Steenhoek, Benjamin, Md Mahbubur Rahman, Richard Jiles, and Wei Le. "An empirical study of deep learning models for vulnerability detection." In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 2237-2248. IEEE, 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00188									S	S			
24	Yuan, Bin, Yifan Lu, Yilin Fang, Yueming Wu, Deqing Zou, Zhen Li, Zhi Li, and Hai Jin. "Enhancing Deep Learning-based Vulnerability Detection by Building Behavior Graph Model." In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 2262-2274. IEEE, 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00190									S	S			
25	Yang, Xu, Shaowei Wang, Yi Li, and Shaohua Wang. "Does data sampling improve deep learning-based vulnerability detection? Yeas! and Nays!." In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 2287-2298. IEEE, 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00192									S	S			
26	Kang, Sungmin, Juyeon Yoon, and Shin Yoo. "Large language models are few-shot testers: Exploring llm-based general bug reproduction." In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 2312-2323. IEEE, 2023.	R	https://doi.org/10.1109/ICSE48619.2023.00194									S	S		M	
27	Griebl, Elisabeth, Benedikt Fein, Florian Obermüller, Gordon Fraser, and René Just. "On the Applicability of Language Models to Block-Based Programs." ICSE '23: Proceedings of the 45th International Conference on Software Engineering May 2023 Pages 2374–2386 (2023).	R	https://doi.org/10.1109/ICSE48619.2023.00199	S					S							
28	Tufano, Rosalia, Luca Pascarella, and Gabriele Bavota. "Automating Code-Related Tasks Through Transformers: The Impact of Pre-training." ICSE '23: Proceedings of the 45th International Conference on Software Engineering May 2023 Pages 2425–2437 (2023).	R	https://doi.org/10.1109/ICSE48619.2023.00203						S			S				
29	Le, V. H., & Zhang, H. (2023). Log Parsing with Prompt-based Few-shot Learning. ICSE '23: Proceedings of the 45th International Conference on Software Engineering	R	https://doi.org/10.1109/ICSE48619.2023.00204										S		S	
30	Nashid, N., Sintaha, M., & Mesbah, A. (2023). Retrieval-based prompt selection for code-related few-shot learning. In Proceedings of the 45th International Conference on Software Engineering (ICSE'23).	R	https://doi.org/10.1109/ICSE48619.2023.00205										S		S	
31	Nong, Y., Ou, Y., Pradel, M., Chen, F., & Cai, H. (2023). VULGEN: Realistic Vulnerability Generation Via Pattern Mining and Deep Learning. In IEEE/ACM 45th International Conference on Software Engineering(ICSE)	R	https://doi.org/10.1109/ICSE48619.2023.00211										S		S	

32	Malinka, K., Peresíni, M., Firc, A., Hujnák, O., & Janus, F. (2023, June). On the educational impact of chatgpt: Is artificial intelligence ready to obtain a university degree?. In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 47-53).	E	https://doi.org/10.1145/3587102.3588827		S	S	S										
33	Ouh, E. L., Gan, B. K. S., Shim, K. J., & Wlodkowski, S. (2023). ChatGPT, Can You Generate Solutions for my Coding Exercises? An Evaluation on its Effectiveness in an undergraduate Java Programming Course. arXiv preprint arXiv:2305.13680.	E	https://arxiv.org/abs/2305.13680v1		S	S	S						S				
34	Cipriano, B. P., & Alves, P. (2023, June). Gpt-3 vs object oriented programming assignments: An experience report. In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 61-67).	E	https://doi.org/10.1145/3587102.3588814		S	S	S										
36	Savelka, J., Agarwal, A., Bogart, C., Song, Y., & Sakr, M. (2023). Can Generative Pre-trained Transformers (GPT) Pass Assessments in Higher Education Programming Courses?. arXiv preprint arXiv:2303.09325.	E	https://arxiv.org/abs/2303.09325v1		S	S	M										
37	Leinonen, J., Denny, P., MacNeil, S., Sarsa, S., Bernstein, S., Kim, J., ... & Hellas, A. (2023). Comparing code explanations created by students and large language models. arXiv preprint arXiv:2304.03938.	E	https://arxiv.org/abs/2304.03938					S		S							
38	Balse, R., Valaboju, B., Singhal, S., Warriem, J. M., & Prasad, P. (2023, June). Investigating the potential of gpt-3 in providing feedback for programming assessments. In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 292-298).	E	https://doi.org/10.1145/3587102.3588852		S	M											
39	Reeves, B., Sarsa, S., Prather, J., Denny, P., Becker, B. A., Hellas, A., ... & Leinonen, J. (2023, June). Evaluating the performance of code generation models for solving Parsons problems with small prompt variations. In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 299-305).	E	https://doi.org/10.1145/3587102.3588805		S	S	S						M				
40	Speth, S., Meißner, N., & Becker, S. (2023, August). Investigating the Use of AI-Generated Exercises for Beginner and Intermediate Programming Courses: A ChatGPT Case Study. In 2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T) (pp. 142-146). IEEE.	E	https://doi.org/10.1109/CSEET58097.2023.00030		S	S	S										
41	Shau, A. C., Liang, Y. C., Hsieh, W. J., Lin, X. L., & Ma, S. P. (2023, August). PSAbot: A Chatbot System for the Analysis of Posts on Stack Overflow. In 2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T) (pp. 137-141). IEEE.	E	https://doi.org/10.1109/CSEET58097.2023.00029		S	S				S							
42	Berrezueta-Guzman, J., & Krusche, S. (2023, August). Recommendations to create programming exercises to overcome ChatGPT. In 2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T) (pp. 147-151). IEEE.	E	https://doi.org/10.1109/CSEET58097.2023.00031		S	S	S										
43	Chan, W. K., Yu, Y. T., Keung, J. W., & Lee, V. C. (2023, August). Toward AI-assisted Exercise Creation for First Course in Programming through Adversarial Examples of AI Models. In 2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T) (pp. 132-136). IEEE.	E	https://doi.org/10.1109/CSEET58097.2023.00028		M	M	M						M				
44	Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., ... & Vinyals, O. (2022). Competition-level code generation with alphacode. Science, 378(6624), 1092-1097.	E	https://doi.org/10.1126/science.abq1158										S				
45	Asare, O., Nagappan, M. & Asokan, N. Is GitHub's Copilot as bad as humans at introducing vulnerabilities in code?. Empir Software Eng 28, 129 (2023).	R	https://doi.org/10.1007/s10664-023-10380-1										M	S			
46	Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J. (2022, February). The robots are coming: Exploring the implications of openai codex on introductory programming. In Proceedings of the 24th Australasian Computing Education Conference (pp. 10-19).	E	https://doi.org/10.1145/3511861.3511863		M	M	M										
47	Sarsa, S., Denny, P., Hellas, A., & Leinonen, J. (2022, August). Automatic generation of programming exercises and code explanations using large language models. In Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1 (pp. 27-43).	E	https://doi.org/10.1145/3501385.3543957		S	S	S										
48	Farah, J. C., Spaenlehauer, B., Sharma, V., Rodríguez-Triana, M. J., Ingram, S., & Gillet, D. (2022, March). Impersonating chatbots in a code review exercise to teach software engineering best practices. In 2022 IEEE Global Engineering Education Conference (EDUCON) (pp. 1634-1642). IEEE.	E	https://doi.org/10.1109/EDUCON52537.2022.9766793							M			M	M			
49	Jalil, S., Rafi, S., LaToza, T. D., Moran, K., & Lam, W. (2023, April). Chatgpt and software testing education: Promises & perils. In 2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW) (pp. 4130-4137). IEEE.	E	https://doi.org/10.1109/ICSTW58534.2023.00078											S		S	
50	Leinonen, J., Hellas, A., Sarsa, S., Reeves, B., Denny, P., Prather, J., & Becker, B. A. (2023, March). Using large language models to enhance programming error messages. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (pp. 563-569).	E	https://doi.org/10.1145/3545945.3569770		M								M				
51	MacNeil, S., Tran, A., Hellas, A., Kim, J., Sarsa, S., Denny, P., ... & Leinonen, J. (2023, March). Experiences from using code explanations generated by large language models in a web software development e-book. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (pp. 931-937).	E	https://doi.org/10.1145/3545945.3569785					S									
52	MacNeil, S., Tran, A., Mogil, D., Bernstein, S., Ross, E., & Huang, Z. (2022, August). Generating diverse code explanations using the gpt-3 large language model. In Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 2 (pp. 37-39).	E	https://doi.org/10.1145/3501709.3544280					S									
53	Dakhel, A. M., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. C., & Jiang, Z. M. J. (2023). Github copilot ai pair programmer: Asset or liability?. Journal of Systems and Software, 203, 111734.	E	https://doi.org/10.1016/j.jss.2023.111734		S	S	S										
54	Sobania, D., Briesch, M., Hanna, C., & Petke, J. (2023). An analysis of the automatic bug fixing performance of chatgpt. arXiv preprint arXiv:2301.08653.	R	https://arxiv.org/abs/2301.08653					S						S	M		

	4	3	3	2	1	3	0	0	7	2	2	8	0
	12	10	8	6	0	6	1	2	10	22	7	12	1
	16	13	11	8	1	9	1	2	17	24	9	20	1
R	2	0	0	5	1	6	1	2	11	22	9	19	1
E	14	13	11	3	0	3	0	0	6	2	0	1	0

Legend: S Strong agreement (3+ out of 4) M Medium agreement (2 out of 4)

SDF-01	**SDF-Fundamentals:** Fundamental Programming Concepts and Practices: This knowledge unit aims to develop understanding of basic concepts, and ability to fluently use basic language constructs as well as modularity constructs. It also aims to familiarize students with the concept of common libraries and frameworks, including those to facilitate API-based access to resources.		**Fundamental Programming Concepts and Practices**
SDF-02	**SDF-DataStructures:** Fundamental Data Structures: This knowledge unit aims to develop core concepts relating to Data Structures and associated operations. Students should understand the important data structures available in the programming language or as libraries, and how to use them effectively, including choosing appropriate data structures while designing solutions for a given problem.		**Fundamental Data Structures**
SDF-03	**SDF-Algorithms:** Algorithms: This knowledge unit aims to develop the foundations of algorithms and their analysis. The KU should also empower students in selecting suitable algorithms for building modest-complexity applications.		**Algorithms**
SDF-04	**SDF-Practices:** Software Development Practices: This knowledge unit develops the core concepts relating to modern software development practices. Its aim is to develop student understanding and basic competencies in program testing, enhancing readability of programs, and using modern methods and tools including some general-purpose IDE.		**Software Development Practices**

SE-01	**SE-Teamwork:** Because of the nature of learning programming, most students in introductory SE have little or no exposure to the collaborative nature of SE. Practice (for instance in project work) may help, but lecture and discussion time spent on the value of clear, effective, and efficient communication and collaboration. are essential for Software Engineering.		**Teamwork**
SE-02	**SE-Tools:** Industry reliance on SE tools has exploded in the past generation, with version control becoming ubiquitous, testing frameworks growing in popularity, increased reliance on static and dynamic analysis in practice, and near-ubiquitous use of continuous integration systems. Increasingly powerful IDEs provide code searching and indexing capabilities, as well as small scale refactoring tools and integration with other SE tools. An understanding of the nature of these tools is broadly valuable - especially version control systems.		**Tools and Environments**
SE-03	**SE-Requirements:** Knowing how to build something is of little help if we do not know what to build.Product Requirements (aka Requirements Engineering, Product Design, Product Requirements solicitation, PRDs, etc.) introduces students to the processes surrounding the specification of the broad requirements governing development of a new product or feature.		**Product Requirements**
SE-04	**SE-Design:** While Product Requirements focuses on the user-facing functionality of a software system, Software Design focuses on the engineer-facing design of internal software components. This encompasses large design concerns such as software architecture, as well as small-scale design choices like API design.		**Software Design**
SE-05	**SE-Construction:** Software Construction focuses on practices that influence the direct production of software: use of tests, test driven development, coding style. More advanced topics extend into secure coding, dependency injection, work prioritization, etc.		**Software Construction**
SE-06	**SE-Validation:** Software Verification and Validation focuses on how to improve the value of testing - understand the role of testing, failure modes, and differences between good tests and poor ones.		**Software Verification and Validation**
SE-07	**SE-Refactoring:** Refactoring and Code Evolution focuses on refactoring and maintenance strategies, incorporating code health, use of tools, and backwards compatibility considerations.		**Refactoring and Code Evolution**
SE-08	**SE-Reliability:** Software Reliability aims to improve understanding of and attention to error cases, failure modes, redundancy, and reasoning about fault tolerance.		**Software Reliability**
SE-09	**SE-FormalMethods:** Formal Methods provides mathematically rigorous mechanisms to apply to software, from specification to verification. (Prerequisites: Substantial dependence on core material from the Discrete Structures area, particularly knowledge units DS/Basic Logic and DS/Proof Techniques.)		**Formal Methods**