

Table 1. List of Included Primary Studies

ID	Year	Title	Venue	Reference
1	2020	Evaluating representation learning of code changes for predicting patch correctness in program repair	ASE	[1]
2	2021	Applying CodeBERT for Automated Program Repair of Java Simple Bugs	MSR	[2]
3	2021	CURE Code-Aware Neural Machine Translation for Automatic Program Repair	ICSE	[3]
4	2021	Generating Bug-Fixes Using Pretrained Transformers	PLDI	[4]
5	2021	Is this Change the Answer to that Problem? Correlating Descriptions of Bug and Code Changes for Evaluating Patch Correctness	ASE	[5]
6	2021	TFix-Learning to Fix Coding Errors with a Text-to-Text Transformer	PMLR	[6]
7	2022	An Analysis of the Automatic Bug Fixing Performance of ChatGPT	APR	[7]
8	2022	An empirical study of deep transfer learning-based program repair for Kotlin projects	FSE/ESEC	[8]
9	2022	Can OpenAI’s codex fix bugs?: an evaluation on QuixBugs	APR	[9]
10	2022	CIRCLE: Continual Repair across Programming Languages	ISSTA	[10]
11	2022	DEAR A Novel Deep Learning-based Approach for Automated Program Repair	ICSE	[11]
12	2022	DeepDev-PERF: A Deep Learning-Based Approach for Improving Software Performance	FSE/ESEC	[12]
13	2022	Fix Bugs with Transformer through a Neural-Symbolic Edit Grammar	ICLR	[13]
14	2022	Framing Program Repair as Code Completion	APR	[14]
15	2022	Less training, more repairing please: revisiting automated program repair via zero-shot learning	FSE/ESEC	[15]
16	2022	Patch Generation with Language Models: Feasibility and Scaling Behavior	ICLR	[16]
17	2022	SYNSHINE: improved fixing of Syntax Errors	TSE	[17]
18	2022	Towards JavaScript program repair with Generative Pre-trained Transformer (GPT-2)	APR	[18]
19	2022	VulRepair: A T5-Based Automated Software Vulnerability Repair	FSE/ESEC	[19]
20	2023	A Chain of AI-based Solutions for Resolving FQNs and Fixing Syntax Errors in Partial Code	arxiv	[20]
21	2023	A critical review of large language model on software engineering: An example from chatgpt and automated program repair	arxiv	[21]
22	2023	Agentcoder: Multi-agent-based code generation with iterative testing and optimisation	arxiv	[22]
23	2023	An Empirical Study of Adoption of ChatGPT for Bug Fixing among Professional Developers	ITA	[23]
24	2023	An Empirical Study on Fine-Tuning Large Language Models of Code for Automated Program Repair	ASE	[24]
25	2023	An Evaluation of the Effectiveness of OpenAI’s ChatGPT for Automated Python Program Bug Fixing using QuixBugs	iSEMANTIC	[25]
26	2023	An Extensive Study on Model Architecture and Program Representation in the Domain of Learning-based Automated Program Repair	APR	[26]
27	2023	Automated Code Editing with Search-Generate-Modify	TSE	[27]
28	2023	Automated Program Repair in the Era of Large Pre-trained Language Models	ICSE	[28]
29	2023	Automated Program Repair Using Generative Models for Code Infilling	AIED	[29]
30	2023	Automated Repair of Programs from Large Language Models	ICSE	[30]
31	2023	Baldur: Whole-Proof Generation and Repair with Large Language Models	FSE/ESEC	[31]
32	2023	Can LLMs Patch Security Issues	arxiv	[32]
33	2023	Coffee: Boost Your Code LLMs by Fixing Bugs with Feedback	arxiv	[33]

Continued on next page

ID	Year	Title	Venue	Reference
34	2023	Copiloting the Copilots: Fusing Large Language Models with Completion Engines for Automated Program Repair	FSE/ESEC	[34]
35	2023	Enhancing Automated Program Repair through Fine-tuning and Prompt Engineering	arxiv	[35]
36	2023	Enhancing Code Language Models for Program Repair by Curricular Fine-tuning Framework	ICSME	[36]
37	2023	Evaluating ChatGPT for Smart Contracts Vulnerability Correction	COMPSAC	[37]
38	2023	Evaluating Pre-trained Language Models for Repairing API Misuses	arxiv	[38]
39	2023	Examining zero-shot vulnerability repair with large language models	S&P	[39]
40	2023	Exploring the Limits of ChatGPT in Software Security Applications	arxiv	[40]
41	2023	FixEval: Execution-based Evaluation of Program Fixes for Programming Problems	APR	[41]
42	2023	Fully Autonomous Programming with Large Language Models	GECCO	[42]
43	2023	Gamma: Revisiting Template-Based Automated Program Repair Via Mask Prediction	ASE	[43]
44	2023	GPT-3-Powered Type Error Debugging: Investigating the Use of Large Language Models for Code Repair	SLE	[44]
45	2023	How Effective Are Neural Networks for Fixing Security Vulnerabilities	ISSTA	[45]
46	2023	Impact of Code Language Models on Automated Program Repair	ICSE	[46]
47	2023	Improving Automated Program Repair with Domain Adaptation	TOSEM	[47]
48	2023	InferFix: End-to-End Program Repair with LLMs over Retrieval-Augmented Prompts	FSE/ESEC	[48]
49	2023	Invalidator: Automated Patch Correctness Assessment via Semantic and Syntactic Reasoning	TSE	[49]
50	2023	Is ChatGPT the Ultimate Programming Assistant-How far is it?	arxiv	[50]
51	2023	Is Self-Repair a Silver Bullet for Code Generation?	ICLR	[51]
52	2023	LLM4SecHW: Leveraging domain-specific large language model for hardware debugging	AsianHOST	[52]
53	2023	Neural Program Repair with Program Dependence Analysis and Effective Filter Mechanism	arxiv	[53]
54	2023	PATCH: Empowering Large Language Model with Programmer-Intent Guidance and Collaborative-Behavior Simulation for Automatic Bug Fixing	TOSEM	[54]
55	2023	Pre-trained Model-based Automated Software Vulnerability Repair: How Far are We?	TDSC	[55]
56	2023	RAP-Gen: Retrieval-Augmented Patch Generation with CodeT5 for Automatic Program Repair	FSE/ESEC	[56]
57	2023	Refining ChatGPT-Generated Code: Characterizing and Mitigating Code Quality Issues	TOSEM	[57]
58	2023	Repair Is Nearly Generation: Multilingual Program Repair with LLMs	AAAI	[58]
59	2023	Resolving Crash Bugs via Large Language Models: An Empirical Study	arxiv	[59]
60	2023	Retrieval-based prompt selection for code-related few-shot learning	ICSE	[60]
61	2023	Self-Edit: Fault-Aware Code Editor for Code Generation	ACL	[61]
62	2023	SelfEvolve: A Code Evolution Framework via Large Language Models	arxiv	[62]
63	2023	Self-Refine: Iterative Refinement with Self-Feedback	NeurIPS	[63]
64	2023	SkipAnalyzer: An Embodied Agent for Code Analysis with Large Language Models	arxiv	[64]
65	2023	The Best of Both Worlds: Combining Learned Embeddings with Engineered Features for Accurate Prediction of Correct Patches	TOSEM	[65]
66	2023	The Plastic Surgery Hypothesis in the Era of Large Language Models	ASE	[66]
67	2023	The Right Prompts for the Job: Repair Code-Review Defects with Large Language Model	arxiv	[67]
68	2023	Towards Generating Functionally Correct Code Edits from Natural Language Issue Descriptions	ICSE	[68]

Continued on next page

ID	Year	Title	Venue	Reference
69	2023	Training Language Models for Programming Feedback Using Automated Repair Tools	AIED	[69]
70	2023	Vision Transformer-Inspired Automated Vulnerability Repair	TOSEM	[70]
71	2023	What makes good in-context demonstrations for code intelligence tasks with llms?	ASE	[71]
72	2023	ZeroLeak: Using LLMs for Scalable and Cost Effective Side-Channel Patching	ESORICS	[72]
73	2024	A Case Study of LLM for Automated Vulnerability Repair: Assessing Impact of Reasoning and Patch Validation Feedback	AIware	[73]
74	2024	A Deep Dive into Large Language Models for Automated Bug Localization and Repair	FSE/ESEC	[74]
75	2024	A Study of Vulnerability Repair in JavaScript Programs with Large Language Models	WWW	[75]
76	2024	An Empirical Evaluation of Pre-trained Large Language Models for Repairing Declarative Formal Specifications	EMSE	[76]
77	2024	APPT Boosting Automated Patch Correctness Prediction via Pre-trained Language Model	TSE	[77]
78	2024	Assessing the Latent Automated Program Repair Capabilities of Large Language Models using Round-Trip Translation	TOSEM	[78]
79	2024	AutoCodeRover: Autonomous Program Improvement	ISSTA	[79]
80	2024	Automated Program Repair for Introductory Programming Assignments	TLT	[80]
81	2024	Automated Program Repair via Conversation: Fixing 162 out of 337 bugs for \$0.42 each using chatgpt	ISSTA	[81]
82	2024	Automated Test Case Repair Using Language Models	TSE	[82]
83	2024	Benchmarking Automated Program Repair: An Extensive Study on Both Real-World and Artificial Bugs	ISSTA	[83]
84	2024	Chain-of-Thought Prompting of Large Language Models for Discovering and Fixing Software Vulnerabilities	arxiv	[84]
85	2024	CigaR: Cost-efficient Program Repair with LLMs	arxiv	[85]
86	2024	Code repair with llms gives an exploration-exploitation tradeoff	NeurIPS	[86]
87	2024	CodeEditorBench: Evaluating Code Editing Capability of Large Language Models	ICLR	[87]
88	2024	CodeR: Issue Resolving with Multi-Agent and Task Graphs	arxiv	[88]
89	2024	CodeScope: An Execution-based Multilingual Multitask Multidimensional Benchmark for Evaluating LLMs on Code Understanding and Generation	ACL	[89]
90	2024	ConDefects: A New Dataset to Address the Data Leakage Concern for LLM-based Fault Localization and Program Repair	FSE/ESEC	[90]
91	2024	CORE: Resolving Code Quality Issues using LLMs	FSE/ESEC	[91]
92	2024	CraftRTL: High-quality Synthetic Data Generation for Verilog Code Models with Correct-by-Construction Non-Textual Representations and Targeted Code Repair	ICLR	[92]
93	2024	CREF: An LLM-based Conversational Software Repair Framework for Programming Tutors	ISSTA	[93]
94	2024	CYCLE: Learning to Self-Refine the Code Generation	OOPSLA	[94]
95	2024	DebugBench: Evaluating Debugging Capability of Large Language Models	ACL	[95]
96	2024	Divide-and-Conquer: Automating Code Revisions via Localization-and-Revision	TOSEM	[96]
97	2024	Domain Knowledge Matters: Improving Prompts with Fix Templates for Repairing Python Type Errors	ICSE	[97]
98	2024	DrPlanner: Diagnosis and Repair of Motion Planners Using Large Language Models	RAL	[98]
99	2024	Enhancing Automated Program Repair with Solution Design	ASE	[99]

Continued on next page

ID	Year	Title	Venue	Reference
100	2024	Exploring and Lifting the Robustness of LLM-powered Automated Program Repair with Metamorphic Testing	arxiv	[100]
101	2024	Exploring ChatGPT's Capabilities on Vulnerability Management	USENIX Security	[101]
102	2024	Exploring Experiences with Automated Program Repair in Practice	ICSE	[102]
103	2024	Exploring Parameter-Efficient Fine-Tuning of Large Language Model on Automated Program Repair	ASE	[103]
104	2024	Exploring the Potential of ChatGPT in Automated Code Refinement: An Empirical Study	ICSE	[104]
105	2024	FastFixer: An Efficient and Effective Approach for Repairing Programming Assignments	ASE	[105]
106	2024	FlakyFix: Using Large Language Models for Predicting Flaky Test Fix Categories and Test Code Repair	TSE	[106]
107	2024	From Code to Correctness: Closing the Last Mile of Code Generation with Hierarchical Debugging	arxiv	[107]
108	2024	Frustrated with Code Quality Issues? LLMs can Help!	FSE/ESEC	[108]
109	2024	GitBug-Java: A Reproducible Benchmark of Recent Java Bugs	MSR	[109]
110	2024	HDLdebugger: Streamlining HDL debugging with Large Language Models	TODAES	[110]
111	2024	How Far Can We Go with Practical Function-Level Program Repair?	arxiv	[111]
112	2024	Improved Program Repair Methods using Refactoring with GPT Models	SIGCSE TS	[112]
113	2024	Investigating Large Language Models Capabilities for Automatic Code Repair in Python	Cluster Computing	[113]
114	2024	LDB: A Large Language Model Debugger via Verifying Runtime Execution Step by Step	ACL	[114]
115	2024	Leveraging Large Language Model for Automatic Patch Correctness Assessment	TSE	[115]
116	2024	Leveraging Print Debugging to Improve Code Generation in Large Language Models	arxiv	[116]
117	2024	Lost in Translation: A Study of Bugs Introduced by Large Language Models while Translating Code	ICSE	[117]
118	2024	MarsCode Agent: AI-native Automated Bug Fixing	arxiv	[118]
119	2024	MASAI: Modular Architecture for Software-engineering AI Agents	NeurIPS	[119]
120	2024	On Hardware Security Bug Code Fixes By Prompting Large Language Models	TIFS	[120]
121	2024	One Size Does Not Fit All: Multi-granularity Patch Generation for Better Automated Program Repair	ISSTA	[121]
122	2024	OpenCodeInterpreter: Integrating Code Generation with Execution and Refinement	ACL	[122]
123	2024	Out of Context: How important is Local Context in Neural Program Repair?	ICSE	[123]
124	2024	Out of Sight, Out of Mind: Better Automatic Vulnerability Repair by Broadening Input Ranges and Sources	ICSE	[124]
125	2024	Peer-aided Repairer: Empowering Large Language Models to Repair Advanced Student Assignments	arxiv	[125]
126	2024	PyDex: Repairing Bugs in Introductory Python Assignments using LLMs	OOPSLA	[126]
127	2024	PyTy: Repairing Static Type Errors in Python	ICSE	[127]
128	2024	RePair: Automated Program Repair with Process-based Feedback	ACL	[128]
129	2024	RetypeR: Integrated Retrieval-based Automatic Program Repair for Python Type Errors	ICSME	[129]
130	2024	Revisiting Unnaturalness for Automated Program Repair in the Era of Large Language Models	arxiv	[130]
131	2024	RTLFixer: Automatically Fixing RTL Syntax Errors with Large Language Models	DAC	[131]

Continued on next page

ID	Year	Title	Venue	Reference
132	2024	SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering	NeurIPS	[132]
133	2024	SWE-bench: Can Language Models Resolve Real-World GitHub Issues?	ICLR	[133]
134	2024	T5APR: Empowering Automated Program Repair across Languages through Checkpoint Ensemble	JSS	[134]
135	2024	Teaching Large Language Models to Self-Debug	ICLR	[135]
136	2024	Thinkrepair: Self-directed automated program repair	ISSTA	[136]
137	2024	VulAdvisor: Natural Language Suggestion Generation for Software Vulnerability Repair	ASE	[137]
138	2024	When Large Language Models Confront Repository-Level Automatic Program Repair: How Well They Done?	ICSE	[138]
139	2025	A New Era in Software Security: Towards Self-Healing Software via Large Language Models and Formal Verification	AST	[139]
140	2025	A study on Prompt Design, Advantages and Limitations of ChatGPT for Deep Learning Program Repair	AUSE	[140]
141	2025	ACFIX: Guiding LLMs with Mined Common RBAC Practices for Context-Aware Repair of Access Control Vulnerabilities in Smart Contracts	TSE	[141]
142	2025	Adversarial Reasoning for Repair Based on Inferred Program Intent	ISSTA	[142]
143	2025	Alibaba LingmaAgent: Improving Automated Issue Resolution via Comprehensive Repository Exploration	FSE-Companion	[143]
144	2025	Aligning the Objective of LLM-based Program Repair	ICSE	[144]
145	2025	APPATCH: Automated Adaptive Prompting Large Language Models for Real-World Software Vulnerability Patching	USENIX Security	[145]
146	2025	APRMCTS: Improving LLM-based Automated Program Repair with Iterative Tree Search	ASE	[146]
147	2025	Closing the Gap: A User Study on the Real-world Usefulness of AI-powered Vulnerability Detection & Repair in the IDESecurityArtifact-FunctionalArtifact-AvailableArtifact-Reusable	ICSE	[147]
148	2025	Combining Logic and Large Language Models for Assisted Debugging and Repair of ASP Programs	ICST	[148]
149	2025	ContrastRepair: Enhancing Conversation-Based Automated Program Repair via Contrastive Test Case Pairs	TOSEM	[149]
150	2025	Counterexample Guided Program Repair Using Zero-Shot Learning and MaxSAT-based Fault Localization	AAAI	[150]
151	2025	Demystifying LLM-based Software Engineering Agents	FSE/ESEC	[151]
152	2025	Demystifying Memorization in LLM-based Program Repair via a General Hypothesis Testing Framework	FSE/ESEC	[152]
153	2025	DesignRepair: Dual-Stream Design Guideline-Aware Frontend Repair with Large Language Models	ICSE	[153]
154	2025	Error Delayed Is Not Error Handled: Understanding and Fixing Propagated Error-Handling Bugs	FSE/ESEC	[154]
155	2025	Explainable Automated Debugging via Large Language Model-driven Scientific Debugging	EMSE	[155]
156	2025	FixAgent: Hierarchical Multi-Agent Framework for Unified Software Debugging	arxiv	[156]
157	2025	Guiding ChatGPT to Fix Web UI Tests via Explanation-Consistency Checking	ICST	[157]
158	2025	HapRepair: Learn to Repair OpenHarmony Apps	FSE/ESEC	[158]
159	2025	Hierarchical Knowledge Injection for Improving LLM-based Program Repair	ASE	[159]
160	2025	Hybrid Automated Program Repair by Combining Large Language Models and Program Analy	TOSEM	[160]

Continued on next page

ID	Year	Title	Venue	Reference
161	2025	Integrating Various Software Artifacts for Better LLM-based Bug Localization and Program Repair	TOSEM	[161]
162	2025	Investigating the Transferability of Code Repair for Low-Resource Programming Languages	NAACL	[162]
163	2025	Knowledge-Enhanced Program Repair for Data Science Code	ICSE	[163]
164	2025	Less is More: Adaptive Program Repair with Bug Localization and Preference Learning	AAAI	[164]
165	2025	MAGIS: LLM-Based Multi-Agent Framework for GitHub Issue Resolution	NeurIPS	[165]
166	2025	MORepair: Teaching LLMs to Repair Code via Multi-Objective Fine-tuning	TOSEM	[166]
167	2025	NIODebugger: A Novel Approach to Repair Non-Idempotent-Outcome Tests with LLM-Based Agent	ICSE	[167]
168	2025	OmniGIRL: A Multilingual and Multimodal Benchmark for GitHub Issue Resolution	ISSTA	[168]
169	2025	OpenHands: An Open Platform for AI Software Developers as Generalist Agents	ICLR	[169]
170	2025	PATCHAGENT: A Practical Program Repair Agent Mimicking Human Expertise	USENIX Security	[170]
171	2025	RAPGen: An Approach for Fixing Code Inefficiencies in Zero-Shot	ICSE	[171]
172	2025	Repair Ingredients Are All You Need: Improving Large Language Model-Based Program Repair via Repair Ingredients Search	ICSE	[172]
173	2025	RepairAgent: An Autonomous, LLM-Based Agent for Program Repair	ICSE	[173]
174	2025	RepairLLaMA: Efficient Representations and Fine-Tuned Adapters for Program Repair	TSE	[174]
175	2025	RepoGraph: Enhancing AI Software Engineering with Repository-level Code Graph	ICLR	[175]
176	2025	RustAssistant: Using LLMs to Fix Compilation Errors in Rust Code	ICSE	[176]
177	2025	Seeing is Fixing: Cross-Modal Reasoning with Multimodal LLMs for Visual Software Issue Fixing	ASE	[177]
178	2025	SWE-bench Multimodal: Do AI Systems Generalize to Visual Software Domains?	ICLR	[178]
179	2025	SWE-GPT: A Process-Centric Language Model for Automated Software Improvement	ISSTA	[179]
180	2025	SWE-RL: Advancing LLM Reasoning via Reinforcement Learning on Open Software Evolution	NeurIPS	[180]
181	2025	SWE-Search: Enhancing Software Agents with Monte Carlo Tree Search and Iterative Refinement	ICLR	[181]
182	2025	SWT-Bench: Testing and Validating Real-World Bug-Fixes with Code Agents	NeurIPS	[182]
183	2025	Teaching AI the ‘Why’ and ‘How’ of Software Vulnerability Fixes	FSE/ESEC	[183]
184	2025	Template-Guided Program Repair in the Era of Large Language Models	ICSE	[184]
185	2025	The Art of Repair: Optimizing Iterative Program Repair with Instruction-Tuned Models	EASE	[185]
186	2025	The Fact Selection Problem in LLM-Based Program Repair	ICSE	[186]
187	2025	The Impact of Fine-tuning Large Language Models on Automated Program Repair	ICSME	[187]
188	2025	Towards Detecting Prompt Knowledge Gaps for Improved LLM-guided Issue Resolution	MSR	[188]
189	2025	When Fine-Tuning LLMs Meets Data Privacy: An Empirical Study of Federated Learning in LLM-Based Program Repair	TOSEM	[189]

REFERENCES

- [1] Haoye Tian, Kui Liu, Abdoul Kader Kaboré, Anil Koyuncu, Li Li, Jacques Klein, and Tegawendé F Bisseyandé. Evaluating representation learning of code changes for predicting patch correctness in program repair. In *Proceedings of the 35th*

IEEE/ACM International Conference on Automated Software Engineering, pages 981–992, 2020.

- [2] Ehsan Mashhadi and Hadi Hemmati. Applying codebert for automated program repair of java simple bugs. In *Proceedings Companion of the 18th IEEE/ACM International Conference on Mining Software Repositories (MSR’21)*, pages 505–509, 2021.
- [3] Nan Jiang, Thibaud Lutellier, and Lin Tan. CURE: code-aware neural machine translation for automatic program repair. In *Proceedings of the 43rd IEEE/ACM International Conference on Software Engineering*, pages 1161–1173, 2021.
- [4] Dawn Drain, Chen Wu, Alexey Svyatkovskiy, and Neel Sundaresan. Generating bug-fixes using pretrained transformers. In *Proceedings of the 5th ACM SIGPLAN International Symposium on Machine Programming*, pages 1–8, 2021.
- [5] Haoye Tian, Xunzhu Tang, Andrew Habib, Shangwen Wang, Kui Liu, Xin Xia, Jacques Klein, and TegawendÉ F BissyandÉ. Is this change the answer to that problem? correlating descriptions of bug and code changes for evaluating patch correctness. In *37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–13. IEEE, 2022.
- [6] Berkay Berabi, Jingxuan He, Veselin Raychev, and Martin Vechev. Tfix: Learning to fix coding errors with a text-to-text transformer. In *International Conference on Machine Learning*, pages 780–791. PMLR, 2021.
- [7] Dominik Sobania, Martin Briesch, Carol Hanna, and Justyna Petke. An analysis of the automatic bug fixing performance of ChatGPT. In *2023 IEEE/ACM International Workshop on Automated Program Repair*, pages 23–30, 2023.
- [8] Misoo Kim, Youngkyoung Kim, Hohyeon Jeong, Jinseok Heo, Sungoh Kim, Hyunhee Chung, and Eunseok Lee. An empirical study of deep transfer learning-based program repair for kotlin projects. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1441–1452, 2022.
- [9] Julian Aron Prenner, Hlib Babii, and Romain Robbes. Can openai’s codex fix bugs? an evaluation on quixbugs. In *Proceedings of the Third International Workshop on Automated Program Repair*, pages 69–75, 2022.
- [10] Wei Yuan, Quanjun Zhang, Tieke He, Chunrong Fang, Nguyen Quoc Viet Hung, Xiaodong Hao, and Hongzhi Yin. CIRCLE: Continual repair across programming languages. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 678–690. ACM, 2022.
- [11] Yi Li, Shaohua Wang, and Tien N. Nguyen. DEAR: A novel deep learning-based approach for automated program repair. In *Proceedings of the 44th International Conference on Software Engineering*, pages 511–523, 2022.
- [12] Spandan Garg, Roshanak Zilouchian Moghaddam, Colin B Clement, Neel Sundaresan, and Chen Wu. DeepDev-PERF: A deep learning-based approach for improving software performance. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 948–958, 2022.
- [13] Yaojie Hu, Xingjian Shi, Qiang Zhou, and Lee Pike. Fix bugs with transformer through a neural-symbolic edit grammar. In *Deep Learning for Code Workshop*, 2022.
- [14] Francisco Ribeiro, Rui Abreu, and João Saraiva. Framing program repair as code completion. In *Proceedings of the Third International Workshop on Automated Program Repair*, pages 38–45. IEEE, 2022.
- [15] Chunqiu Steven Xia and Lingming Zhang. Less training, more repairing please: Revisiting automated program repair via zero-shot learning. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 959–971, 2022.
- [16] Sophia D Kolak, Ruben Martins, Claire Le Goues, and Vincent Josua Hellendoorn. Patch generation with language models: Feasibility and scaling behavior. In *International Conference on Learning Representations Deep Learning for Code Workshop*, 2022.
- [17] Toufique Ahmed, Noah Rose Ledesma, and Premkumar Devanbu. SynShine: Improved fixing of syntax errors. *IEEE Transactions on Software Engineering*, 49(4):2169–2181, 2022.
- [18] Márk Lajkó, Viktor Csuvik, and László Vidács. Towards javascript program repair with generative pre-trained transformer. In *2022 IEEE/ACM International Workshop on Automated Program Repair*, pages 61–68. IEEE, 2022.
- [19] Michael Fu, Chakkrit Tantithamthavorn, Trung Le, Van Nguyen, and Phung Dinh. Vulrepair: A t5-based automated software vulnerability repair. In *the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 935–947. ACM, 2022.
- [20] Qing Huang, Jiahui Zhu, Zhenchang Xing, Huan Jin, Changjing Wang, and Xiwei Xu. A chain of ai-based solutions for resolving fqns and fixing syntax errors in partial code. *arXiv preprint arXiv:2306.11981*, 2023.
- [21] Quanjun Zhang, Tongke Zhang, Juan Zhai, Chunrong Fang, Bowen Yu, Weisong Sun, and Zhenyu Chen. A critical review of large language model on software engineering: An example from ChatGPT and automated program repair. *arXiv preprint arXiv:2310.08879*, 2023.
- [22] Dong Huang, Qingwen Bu, Jie M Zhang, Michael Luck, and Heming Cui. Agentcoder: Multi-agent-based code generation with iterative testing and optimisation. *arXiv preprint arXiv:2312.13010*, 2023.

- [23] Haotong Ge and Yuemeng Wu. An empirical study of adoption of chatgpt for bug fixing among professional developers. *Innovation & Technology Advances*, 1(1):21–29, Jun. 2023.
- [24] Kai Huang, Jian Zhang, Xinlei Bao, Xu Wang, and Yang Liu. Comprehensive fine-tuning large language models of code for automated program repair. *IEEE Transactions on Software Engineering*, 51(4):904–928, 2025.
- [25] Marchel Christopher Wuisang, Marcel Kurniawan, Komang Andika Wira Santosa, Alexander Agung Santoso Gunawan, and Karen Etania Saputra. An evaluation of the effectiveness of openai’s ChatGPT for automated python program bug fixing using quixbugs. In *2023 International Seminar on Application for Technology of Information and Communication*, pages 295–300. IEEE, 2023.
- [26] Dániel Horváth, Viktor Csuvik, Tibor Gyimóthy, and László Vidács. An extensive study on model architecture and program representation in the domain of learning-based automated program repair. In *2023 IEEE/ACM International Workshop on Automated Program Repair (APR)*, pages 31–38, 2023.
- [27] Changshu Liu, Pelin Cetin, Yogesh Patodia, Saikat Chakraborty, Yangruibo Ding, and Baishakhi Ray. Automated code editing with search-generate-modify. *arXiv preprint arXiv:2306.06490*, 2023.
- [28] Chunqiu Steven Xia, Yuxiang Wei, and Lingming Zhang. Automated program repair in the era of large pre-trained language models. In *2023 IEEE/ACM 45th International Conference on Software Engineering*, pages 1482–1494. IEEE Computer Society, 2023.
- [29] Charles Koutchme, Sami Sarsa, Juho Leinonen, Arto Hellas, and Paul Denny. Automated program repair using generative models for code infilling. In *International Conference on Artificial Intelligence in Education*, pages 798–803. Springer, 2023.
- [30] Zhiyu Fan, 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*, pages 1469–1481. IEEE, 2023.
- [31] Emily First, Markus Rabe, Talia Ringer, and Yuriy Brun. 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*, pages 1229–1241, 2023.
- [32] Kamel Alrashedy and Abdullah Aljasser. Can llms patch security issues? *arXiv preprint arXiv:2312.00024*, 2023.
- [33] Seungjun Moon, Yongho Song, Hyungjoo Chae, Dongjin Kang, Taeyoon Kwon, Kai Tzu-iunn Ong, Seung-won Hwang, and Jinyoung Yeo. Coffee: Boost your code llms by fixing bugs with feedback. *arXiv preprint arXiv:2311.07215*, 2023.
- [34] Yuxiang Wei, Chunqiu Steven Xia, and Lingming Zhang. Copiloting the copilots: Fusing large language models with completion engines for automated program repair. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 172–184, 2023.
- [35] Rishov Paul, Md Mohib Hossain, Mohammed Latif Siddiq, Masum Hasan, Anindya Iqbal, and Joanna CS Santos. Enhancing automated program repair through fine-tuning and prompt engineering. *arXiv preprint arXiv:2304.07840*, 2023.
- [36] Sichong Hao, Xianjun Shi, Hongwei Liu, and Yanjun Shu. Enhancing code language models for program repair by curricular fine-tuning framework. In *2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 136–146. IEEE, 2023.
- [37] Emanuele Antonio Napoli and Valentina Gatteschi. Evaluating ChatGPT for smart contracts vulnerability correction. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1828–1833. IEEE, 2023.
- [38] Ting Zhang, Ivana Clairine Irsan, Ferdian Thung, David Lo, Asankhaya Sharma, and Lingxiao Jiang. Evaluating pre-trained language models for repairing api misuses. *arXiv preprint arXiv:2310.16390*, 2023.
- [39] Hammond Pearce, Benjamin Tan, Baleegh Ahmad, Ramesh Karri, and Brendan Dolan-Gavitt. Examining zero-shot vulnerability repair with large language models. In *2023 IEEE Symposium on Security and Privacy*, pages 2339–2356. IEEE, 2023.
- [40] Fangzhou Wu, Qingzhao Zhang, Ati Priya Bajaj, Tiffany Bao, Ning Zhang, Ruoyu Wang, Chaowei Xiao, et al. Exploring the limits of ChatGPT in software security applications. *arXiv preprint arXiv:2312.05275*, 2023.
- [41] Md Mahim Anjum Haque, Wasi Uddin Ahmad, Ismini Lourentzou, and Chris Brown. FixEval: Execution-based evaluation of program fixes for programming problems. In *2023 IEEE/ACM International Workshop on Automated Program Repair*, pages 11–18. IEEE, 2023.
- [42] Vadim Liventsev, Anastasiia Grishina, Aki Härmä, and Leon Moonen. Fully autonomous programming with large language models. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’23*, pages 1146–1155, 2023.
- [43] Quanjun Zhang, Chunrong Fang, Tongke Zhang, Bowen Yu, Weisong Sun, and Zhenyu Chen. Gamma: Revisiting template-based automated program repair via mask prediction. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering*, pages 535–547. IEEE, 2023.
- [44] Francisco Ribeiro, José Nuno Castro de Macedo, Kanae Tsushima, Rui Abreu, and João Saraiva. Gpt-3-powered type error debugging: Investigating the use of large language models for code repair. In *Proceedings of the 16th ACM*

SIGPLAN International Conference on Software Language Engineering, pages 111–124, 2023.

- [45] Yi Wu, Nan Jiang, Hung Viet Pham, Thibaud Lutellier, Jordan Davis, Lin Tan, Petr Babkin, and Sameena Shah. How effective are neural networks for fixing security vulnerabilities. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2023, pages 1282–1294. Association for Computing Machinery, 2023.
- [46] Nan Jiang, Kevin Liu, Thibaud Lutellier, and Lin Tan. Impact of code language models on automated program repair. In *Proceedings of the 45th International Conference on Software Engineering*, pages 1430–1442, 2023.
- [47] Armin Zirak and Hadi Hemmati. Improving automated program repair with domain adaptation. *ACM Transactions on Software Engineering and Methodology*, 33(3):1–43, 2024.
- [48] Matthew Jin, Syed Shahriar, Michele Tufano, Xin Shi, Shuai Lu, Neel Sundaresan, and Alexey Svyatkovskiy. Inferfix: End-to-end program repair with llms. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1646–1656. ACM, 2023.
- [49] Thanh Le-Cong, Duc-Minh Luong, Xuan Bach D Le, David Lo, Nhat-Hoa Tran, Bui Quang-Huy, and Quyet-Thang Huynh. Invalidator: Automated patch correctness assessment via semantic and syntactic reasoning. *IEEE Transactions on Software Engineering*, 49(06):3411–3429, 2023.
- [50] Haoye Tian, Weiqi Lu, Tsz On Li, Xunzhu Tang, Shing-Chi Cheung, Jacques Klein, and Tegawendé F Bissyandé. Is ChatGPT the ultimate programming assistant—how far is it? *arXiv preprint arXiv:2304.11938*, 2023.
- [51] Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. Is self-repair a silver bullet for code generation? In *The Twelfth International Conference on Learning Representations*, 2024.
- [52] Weimin Fu, Kaichen Yang, Raj Gautam Dutta, Xiaolong Guo, and Gang Qu. Llm4sechw: Leveraging domain-specific large language model for hardware debugging. In *2023 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, pages 1–6. IEEE, 2023.
- [53] Yuwei Zhang, Ge Li, Zhi Jin, and Ying Xing. Neural program repair with program dependence analysis and effective filter mechanism. *arXiv preprint arXiv:2305.09315*, 2023.
- [54] Yuwei Zhang, Zhi Jin, Ying Xing, and Ge Li. Steam: Simulating the interactive behavior of programmers for automatic bug fixing. *arXiv preprint arXiv:2308.14460*, 2023.
- [55] Quanjun Zhang, Chunrong Fang, Bowen Yu, Weisong Sun, Tongke Zhang, and Zhenyu Chen. Pre-trained model-based automated software vulnerability repair: How far are we? *IEEE Transactions on Dependable and Secure Computing*, 2023. Early Access, DOI: [10.1109/TDSC.2023.3308897](https://doi.org/10.1109/TDSC.2023.3308897).
- [56] Weishi Wang, Yue Wang, Shafiq Joty, and Steven CH Hoi. Rap-gen: Retrieval-augmented patch generation with codet5 for automatic program repair. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 146–158. ACM, 2023.
- [57] Yue Liu, Thanh Le-Cong, Ratnadira Widayarsi, Chakkrit Tantithamthavorn, Li Li, Xuan-Bach D. Le, and David Lo. Refining chatgpt-generated code: Characterizing and mitigating code quality issues. *ACM Transactions on Software Engineering and Methodology*, 2024. Just Accepted.
- [58] Harshit Joshi, José Cambronero Sanchez, Sumit Gulwani, Vu Le, Gust Verbruggen, and Ivan Radiček. Repair is nearly generation: Multilingual program repair with llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5131–5140, 2023.
- [59] Xueying Du, Mingwei Liu, Juntao Li, Hanlin Wang, Xin Peng, and Yiling Lou. Resolving crash bugs via large language models: An empirical study. *arXiv preprint arXiv:2312.10448*, 2023.
- [60] Noor Nashid, Mifta Sintaha, and Ali Mesbah. Retrieval-based prompt selection for code-related few-shot learning. In *Proceedings of the 45th International Conference on Software Engineering*, pages 2450–2462. IEEE, 2023.
- [61] Kechi Zhang, Zhuo Li, Jia Li, Ge Li, and Zhi Jin. Self-edit: Fault-aware code editor for code generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 769–787, 2023.
- [62] Shuyang Jiang, Yuhao Wang, and Yu Wang. Selfevolve: A code evolution framework via large language models. *arXiv preprint arXiv:2306.02907*, 2023.
- [63] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [64] Mohammad Mahdi Mohajer, Reem Aleithan, Nima Shiri Harzevili, Moshi Wei, Alvine Boaye Belle, Hung Viet Pham, and Song Wang. Skipanalyzer: An embodied agent for code analysis with large language models. *arXiv preprint arXiv:2310.18532*, 2023.
- [65] Haoye Tian, Kui Liu, Yinghua Li, Abdoul Kader Kaboré, Anil Koyuncu, Andrew Habib, Li Li, Junhao Wen, Jacques Klein, and Tegawendé F Bissyandé. The best of both worlds: Combining learned embeddings with engineered features for accurate prediction of correct patches. *ACM Transactions on Software Engineering and Methodology*, 1(1):1–1, 2022.

- [66] Chunqiu Steven Xia, Yifeng Ding, and Lingming Zhang. The plastic surgery hypothesis in the era of large language models. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering*, pages 522–534. IEEE, 2023.
- [67] Zelin Zhao, Zhaogui Xu, Jialong Zhu, Peng Di, Yuan Yao, and Xiaoxing Ma. The right prompts for the job: Repair code-review defects with large language model. *arXiv preprint arXiv:2312.17485*, 2023.
- [68] Sarah Fakhoury, Saikat Chakraborty, Madan Musuvathi, and Shuvendu K Lahiri. Towards generating functionally correct code edits from natural language issue descriptions. *arXiv preprint arXiv:2304.03816*, 2023.
- [69] Charles Koutchme. Training language models for programming feedback using automated repair tools. In *International Conference on Artificial Intelligence in Education*, pages 830–835. Springer, 2023.
- [70] Michael Fu, Van Nguyen, Chakkrit Tantithamthavorn, Dinh Phung, and Trung Le. Vision transformer-inspired automated vulnerability repair. *ACM Transactions on Software Engineering and Methodology*, 2023.
- [71] Shuzheng Gao, Xin-Cheng Wen, Cuiyun Gao, Wenxuan Wang, Hongyu Zhang, and Michael R Lyu. What makes good in-context demonstrations for code intelligence tasks with llms? In *2023 38th IEEE/ACM International Conference on Automated Software Engineering*, pages 761–773. IEEE, 2023.
- [72] M Caner Tol and Berk Sunar. Zeroleak: Using llms for scalable and cost effective side-channel patching. *arXiv preprint arXiv:2308.13062*, 2023.
- [73] Ummay Kulsum, Haotian Zhu, Bowen Xu, and Marcelo d’Amorim. A case study of llm for automated vulnerability repair: Assessing impact of reasoning and patch validation feedback. In *Proceedings of the 1st ACM International Conference on AI-Powered Software*, pages 103–111, 2024.
- [74] Soneya Binta Hossain, Nan Jiang, Qiang Zhou, Xiaopeng Li, Wen-Hao Chiang, Yingjun Lyu, Hoan Nguyen, and Omer Tripp. A deep dive into large language models for automated bug localization and repair. *Proceedings of the ACM on Software Engineering*, 1(FSE):1471–1493, 2024.
- [75] Tan Khang Le, Saba Alimadadi, and Steven Y Ko. A study of vulnerability repair in javascript programs with large language models. *arXiv preprint arXiv:2403.13193*, 2024.
- [76] Mohannad Alhanahnah, Md Rashedul Hasan, Lisong Xu, and Hamid Bagheri. An empirical evaluation of pre-trained large language models for repairing declarative formal specifications. *Empirical Software Engineering*, 30(5):1–38, 2025.
- [77] Quanjun Zhang, Chunrong Fang, Weisong Sun, Yan Liu, Tieke He, Xiaodong Hao, and Zhenyu Chen. Appt: Boosting automated patch correctness prediction via fine-tuning pre-trained models. *IEEE Transactions on Software Engineering*, 50(3):474–494, 2024.
- [78] Fernando Vallecillos Ruiz, Anastasiia Grishina, Max Hort, and Leon Moonen. A novel approach for automatic program repair using round-trip translation with large language models. *arXiv preprint arXiv:2401.07994*, 2024.
- [79] Yuntong Zhang, Haifeng Ruan, Zhiyu Fan, and Abhik Roychoudhury. Autocoderever: Autonomous program improvement. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 1592–1604, 2024.
- [80] Han Wan, Hongzhen Luo, Mengying Li, and Xiaoyan Luo. Automated program repair for introductory programming assignments. *IEEE Transactions on Learning Technologies*, 17:1705–1720, 2024.
- [81] Chunqiu Steven Xia and Lingming Zhang. Automated program repair via conversation: Fixing 162 out of 337 bugs for \$0.42 each using chatgpt. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 819–831, 2024.
- [82] Ahmadreza Saboor Yaraghi, Darren Holden, Nafiseh Kahani, and Lionel Briand. Automated test case repair using language models. *arXiv e-prints*, pages arXiv–2401, 2024.
- [83] Yicheng Ouyang, Jun Yang, and Lingming Zhang. Benchmarking automated program repair: An extensive study on both real-world and artificial bugs. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 440–452, 2024.
- [84] Yu Nong, Mohammed Aldeen, Long Cheng, Hongxin Hu, Feng Chen, and Haipeng Cai. Chain-of-thought prompting of large language models for discovering and fixing software vulnerabilities. *arXiv preprint arXiv:2402.17230*, 2024.
- [85] Dávid Hidvégi, Khashayar Etemadi, Sofia Bobadilla, and Martin Monperrus. CigaR: Cost-efficient program repair with llms. *arXiv preprint arXiv:2402.06598*, 2024.
- [86] Hao Tang, Keya Hu, Jin Zhou, Si Cheng Zhong, Wei-Long Zheng, Xujie Si, and Kevin Ellis. Code repair with llms gives an exploration-exploitation tradeoff. *Advances in Neural Information Processing Systems*, 37:117954–117996, 2024.
- [87] Jiawei Guo, Ziming Li, Xueling Liu, Kaijing Ma, Tianyu Zheng, Zhouliang Yu, Ding Pan, Yizhi Li, Ruibo Liu, Yue Wang, et al. Codeeditorbench: Evaluating code editing capability of large language models. *arXiv preprint arXiv:2404.03543*, 2024.
- [88] Dong Chen, Shaoxin Lin, Muhan Zeng, Daoguang Zan, Jian-Gang Wang, Anton Cheshkov, Jun Sun, Hao Yu, Guoliang Dong, Artem Aliev, et al. Coder: Issue resolving with multi-agent and task graphs. *arXiv preprint arXiv:2406.01304*,

- 2024.
- [89] Weixiang Yan, Haitian Liu, Yunkun Wang, Yunzhe Li, Qian Chen, Wen Wang, Tingyu Lin, Weishan Zhao, Li Zhu, Shuiguang Deng, et al. Codescope: An execution-based multilingual multitask multidimensional benchmark for evaluating llms on code understanding and generation. *arXiv preprint arXiv:2311.08588*, 2023.
 - [90] Yonghao Wu, Zheng Li, Jie M Zhang, and Yong Liu. ConDefects: A new dataset to address the data leakage concern for llm-based fault localization and program repair. *arXiv preprint arXiv:2310.16253*, 2023.
 - [91] Nalin Wadhwa, Jui Pradhan, Atharv Sonwane, Surya Prakash Sahu, Nagarajan Natarajan, Aditya Kanade, Suresh Parthasarathy, and Sriram Rajamani. Core: Resolving code quality issues using llms. *Proceedings of the ACM on Software Engineering*, 1(FSE):789–811, 2024.
 - [92] Mingjie Liu, Yun-Da Tsai, Wenfei Zhou, and Haoxing Ren. Crafttrl: High-quality synthetic data generation for verilog code models with correct-by-construction non-textual representations and targeted code repair. *arXiv preprint arXiv:2409.12993*, 2024.
 - [93] Boyang Yang, Haoye Tian, Weiguo Pian, Haoran Yu, Haitao Wang, Jacques Klein, Tegawendé F Bissyandé, and Shunfu Jin. Cref: An llm-based conversational software repair framework for programming tutors. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 882–894, 2024.
 - [94] Yangruibo Ding, Marcus J Min, Gail Kaiser, and Baishakhi Ray. Cycle: Learning to self-refine the code generation. *arXiv preprint arXiv:2403.18746*, 2024.
 - [95] Runchu Tian, Yining Ye, Yujia Qin, Xin Cong, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Debugbench: Evaluating debugging capability of large language models. *arXiv preprint arXiv:2401.04621*, 2024.
 - [96] Shangwen Wang, Bo Lin, Liqian Chen, and Xiaoguang Mao. Divide-and-conquer: Automating code revisions via localization-and-revision. *ACM Transactions on Software Engineering and Methodology*, 34(3):1–26, 2025.
 - [97] Yun Peng, Shuzheng Gao, Cuiyun Gao, Yintong Huo, and Michael Lyu. Domain knowledge matters: Improving prompts with fix templates for repairing python type errors. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, pages 1–13, 2024.
 - [98] Yuanfei Lin, Chenran Li, Mingyu Ding, Masayoshi Tomizuka, Wei Zhan, and Matthias Althoff. DrPlanner: Diagnosis and repair of motion planners using large language models. *arXiv preprint arXiv:2403.07470*, 2024.
 - [99] Jiuang Zhao, Donghao Yang, Li Zhang, Xiaoli Lian, Zitian Yang, and Fang Liu. Enhancing automated program repair with solution design. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 1706–1718, 2024.
 - [100] Pengyu Xue, Linhao Wu, Zhen Yang, Zhongxing Yu, Zhi Jin, Ge Li, Yan Xiao, Shuo Liu, Xinyi Li, Hongyi Lin, et al. Exploring and lifting the robustness of llm-powered automated program repair with metamorphic testing. *arXiv preprint arXiv:2410.07516*, 2024.
 - [101] Peiyu Liu, Junming Liu, Lirong Fu, Kangjie Lu, Yifan Xia, Xuhong Zhang, Wenzhi Chen, Haiqin Weng, Shouling Ji, and Wenhai Wang. How ChatGPT is solving vulnerability management problem. *arXiv preprint arXiv:2311.06530*, 2023.
 - [102] Fairuz Nower Meem, Justin Smith, and Brittany Johnson. Exploring experiences with automated program repair in practice. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)*, pages 870–870. IEEE Computer Society, 2024.
 - [103] Guochang Li, Chen Zhi, Jialiang Chen, Junxiao Han, and Shuiguang Deng. Exploring parameter-efficient fine-tuning of large language model on automated program repair. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 719–731, 2024.
 - [104] Qi Guo, Junming Cao, Xiaofei Xie, Shangqing Liu, Xiaohong Li, Bihuan Chen, and Xin Peng. Exploring the potential of chatgpt in automated code refinement: An empirical study. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)*, pages 379–391. IEEE Computer Society, 2023.
 - [105] Fang Liu, Zhenwei Liu, Qianhui Zhao, Jing Jiang, Li Zhang, Zian Sun, Ge Li, Zhongqi Li, and Yuchi Ma. Fastfixer: an efficient and effective approach for repairing programming assignments. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 669–680, 2024.
 - [106] Sakina Fatima, Hadi Hemmati, and Lionel C Briand. Flakyfix: Using large language models for predicting flaky test fix categories and test code repair. *IEEE Transactions on Software Engineering*, 50(12):3146–3171, 2024.
 - [107] Yuling Shi, Songsong Wang, Chengcheng Wan, Min Wang, and Xiaodong Gu. From code to correctness: Closing the last mile of code generation with hierarchical debugging. *arXiv preprint arXiv:2410.01215*, 2024.
 - [108] Nalin Wadhwa, Jui Pradhan, Atharv Sonwane, Surya Prakash Sahu, Nagarajan Natarajan, Aditya Kanade, Suresh Parthasarathy, and Sriram Rajamani. Frustrated with code quality issues? llms can help! *arXiv preprint arXiv:2309.12938*, 2023.
 - [109] André Silva, Nuno Saavedra, and Martin Monperrus. GitBug-Java: A reproducible benchmark of recent java bugs. *arXiv preprint arXiv:2402.02961*, 2024.

- [110] Xufeng Yao, Haoyang Li, Tsz Ho Chan, Wenyi Xiao, Mingxuan Yuan, Yu Huang, Lei Chen, and Bei Yu. Hdldebugger: Streamlining hdl debugging with large language models. *arXiv preprint arXiv:2403.11671*, 2024.
- [111] Jiahong Xiang, Xiaoyang Xu, Fanchu Kong, Mingyuan Wu, Zizheng Zhang, Haotian Zhang, and Yuqun Zhang. How far can we go with practical function-level program repair? *arXiv preprint arXiv:2404.12833*, 2024.
- [112] Ryosuke Ishizue, Kazunori Sakamoto, Hironori Washizaki, and Yoshiaki Fukazawa. Improved program repair methods using refactoring with gpt models. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pages 569–575, 2024.
- [113] Safwan Omari, Kshitiz Basnet, and Mohammad Wardat. Investigating large language models capabilities for automatic code repair in python. *Cluster Computing*, 27(8):10717–10731, 2024.
- [114] Li Zhong, Zilong Wang, and Jingbo Shang. Ldb: A large language model debugger via verifying runtime execution step-by-step. *arXiv preprint arXiv:2402.16906*, 2024.
- [115] Xin Zhou, Bowen Xu, Kisub Kim, Dong Gyun Han, Hung Huu Nguyen, Thanh Le-Cong, Junda He, Bach Le, and David Lo. Leveraging large language model for automatic patch correctness assessment. *IEEE Transactions on Software Engineering*, 50(11):2865–2883, 2024.
- [116] Xueyu Hu, Kun Kuang, Jiankai Sun, Hongxia Yang, and Fei Wu. Leveraging print debugging to improve code generation in large language models. *arXiv preprint arXiv:2401.05319*, 2024.
- [117] Rangeet Pan, Ali Reza Ibrahimzada, Rahul Krishna, Divya Sankar, Lambert Pouguem Wassi, Michele Merler, Boris Sobolev, Raju Pavuluri, Saurabh Sinha, and Reyhaneh Jabbarvand. Lost in translation: A study of bugs introduced by large language models while translating code. In *2024 IEEE/ACM 46th International Conference on Software Engineering*, pages 866–866. IEEE Computer Society, 2024.
- [118] Yizhou Liu, Pengfei Gao, Xinchun Wang, Jie Liu, Yexuan Shi, Zhao Zhang, and Chao Peng. Marscode agent: Ai-native automated bug fixing. *arXiv preprint arXiv:2409.00899*, 2024.
- [119] Daman Arora, Atharv Sonwane, Nalin Wadhwa, Abhav Mehrotra, Saiteja Utpala, Ramakrishna Bairi, Aditya Kanade, and Nagarajan Natarajan. Masai: Modular architecture for software-engineering ai agents. *arXiv preprint arXiv:2406.11638*, 2024.
- [120] Baleegh Ahmad, Shailja Thakur, Benjamin Tan, Ramesh Karri, and Hammond Pearce. On hardware security bug code fixes by prompting large language models. *IEEE Transactions on Information Forensics and Security*, 19:4043–4057, 2024.
- [121] Bo Lin, Shangwen Wang, Ming Wen, Liqian Chen, and Xiaoguang Mao. One size does not fit all: Multi-granularity patch generation for better automated program repair. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 1554–1566, 2024.
- [122] Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhui Chen, and Xiang Yue. Open-codeinterpreter: Integrating code generation with execution and refinement. *arXiv preprint arXiv:2402.14658*, 2024.
- [123] Julian Aron Prenner and Romain Robbes. Out of context: How important is local context in neural program repair? *arXiv preprint arXiv:2312.04986*, 2023.
- [124] Xin Zhou, Kisub Kim, Bowen Xu, DongGyun Han, and David Lo. Out of sight, out of mind: Better automatic vulnerability repair by broadening input ranges and sources. In *2024 IEEE/ACM 46th International Conference on Software Engineering*, pages 872–872. IEEE Computer Society, 2024.
- [125] Qianhui Zhao, Fang Liu, Li Zhang, Yang Liu, Zhen Yan, Zhenghao Chen, Yufei Zhou, Jing Jiang, and Ge Li. Peer-aided repairer: Empowering large language models to repair advanced student assignments. *arXiv preprint arXiv:2404.01754*, 2024.
- [126] Jialu Zhang, José Pablo Cambronero, Sumit Gulwani, Vu Le, Ruzica Piskac, Gustavo Soares, and Gust Verbruggen. Pydex: Repairing bugs in introductory python assignments using llms. *Proceedings of the ACM on Programming Languages*, 8(OOPSLA1):1100–1124, 2024.
- [127] Yiu Wai Chow, Luca Di Grazia, and Michael Pradel. PyTy: Repairing static type errors in python. In *Proceedings of the 46th International Conference on Software Engineering*, pages 871–871, 2024.
- [128] Yuze Zhao, Zhenya Huang, Yixiao Ma, Rui Li, Kai Zhang, Hao Jiang, Qi Liu, Linbo Zhu, and Yu Su. Repair: Automated program repair with process-based feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16415–16429, 2024.
- [129] Sichong Hao, Xianjun Shi, and Hongwei Liu. Retyper: Integrated retrieval-based automatic program repair for python type errors. In *2024 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 199–210. IEEE, 2024.
- [130] Aidan ZH Yang, Sophia Kolak, Vincent J Hellendoorn, Ruben Martins, and Claire Le Goues. Revisiting unnaturalness for automated program repair in the era of large language models. *arXiv preprint arXiv:2404.15236*, 2024.
- [131] YunDa Tsai, Mingjie Liu, and Haoxing Ren. Rtlfixer: Automatically fixing rtl syntax errors with large language models. *arXiv preprint arXiv:2311.16543*, 2023.

- [132] John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.
- [133] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024.
- [134] Reza Gharibi, Mohammad Hadi Sadreddini, and Seyed Mostafa Fakhrahmad. T5apr: Empowering automated program repair across languages through checkpoint ensemble. *Journal of Systems and Software*, 214:112083, 2024.
- [135] Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*, 2024.
- [136] Xin Yin, Chao Ni, Shaohua Wang, Zhenhao Li, Limin Zeng, and Xiaohu Yang. Thinkrepair: Self-directed automated program repair. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 1274–1286, 2024.
- [137] Jian Zhang, Chong Wang, Anran Li, Wenhan Wang, Tianlin Li, and Yang Liu. Vuladvisor: Natural language suggestion generation for software vulnerability repair. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 1932–1944, 2024.
- [138] Yuxiao Chen, Jingzheng Wu, Xiang Ling, Changjiang Li, Zhiqing Rui, Tianyue Luo, and Yanjun Wu. When large language models confront repository-level automatic program repair: How well they done? In *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, pages 459–471, 2024.
- [139] Yiannis Charalambous, Norbert Tihanyi, Ridhi Jain, Youcheng Sun, Mohamed Amine Ferrag, and Lucas C Cordeiro. A new era in software security: Towards self-healing software via large language models and formal verification. *arXiv preprint arXiv:2305.14752*, 2023.
- [140] Jialun Cao, Meiziniu Li, Ming Wen, and Shing-chi Cheung. A study on prompt design, advantages and limitations of ChatGPT for deep learning program repair. *arXiv preprint arXiv:2304.08191*, 2023.
- [141] Lyuye Zhang, Kaixuan Li, Kairan Sun, Daoyuan Wu, Ye Liu, Haoye Tian, and Yang Liu. Acfix: Guiding llms with mined common rbac practices for context-aware repair of access control vulnerabilities in smart contracts. *arXiv preprint arXiv:2403.06838*, 2024.
- [142] He Ye, Aidan ZH Yang, Chang Hu, Yanlin Wang, Tao Zhang, and Claire Le Goues. Adverintent-agent: Adversarial reasoning for repair based on inferred program intent. *Proceedings of the ACM on Software Engineering*, 2(ISSTA):1398–1420, 2025.
- [143] Yingwei Ma, Qingping Yang, Rongyu Cao, Binhua Li, Fei Huang, and Yongbin Li. Alibaba lingmaagent: Improving automated issue resolution via comprehensive repository exploration. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, pages 238–249, 2025.
- [144] Junjielong Xu, Ying Fu, Shin Hwei Tan, and Pinjia He. Aligning the objective of llm-based program repair. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 2548–2560. IEEE, 2025.
- [145] Yu Nong, Haoran Yang, Long Cheng, Hongxin Hu, and Haipeng Cai. {APPATCH}: Automated adaptive prompting large language models for {Real-World} software vulnerability patching. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 4481–4500, 2025.
- [146] Haichuan Hu, Congqing He, Hao Zhang, Xiaochen Xie, and Quanjun Zhang. Aprmcts: Improving llm-based automated program repair with iterative tree search. *arXiv preprint arXiv:2507.01827*, 2025.
- [147] Benjamin Steenhoek, Kalpathy Sivaraman, Renata Saldivar Gonzalez, Yevhen Mohylevskyy, Roshanak Zilouchian Moghaddam, and Wei Le. Closing the gap: A user study on the real-world usefulness of ai-powered vulnerability detection & repair in the ide. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 663–663. IEEE Computer Society, 2025.
- [148] Ricardo Brancas, Vasco Manquinho, and Ruben Martins. Combining logic and large language models for assisted debugging and repair of asp programs. In *2025 IEEE Conference on Software Testing, Verification and Validation (ICST)*, pages 646–657. IEEE, 2025.
- [149] Jiaolong Kong, Mingfei Cheng, Xiaofei Xie, Shangqing Liu, Xiaoning Du, and Qi Guo. ContrastRepair: Enhancing conversation-based automated program repair via contrastive test case pairs. *arXiv preprint arXiv:2403.01971*, 2024.
- [150] Pedro Orvalho, Mikoláš Janota, and Vasco M Manquinho. Counterexample guided program repair using zero-shot learning and maxsat-based fault localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 649–657, 2025.
- [151] Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. Demystifying llm-based software engineering agents. *Proceedings of the ACM on Software Engineering*, 2(FSE):801–824, 2025.
- [152] Jiaolong Kong, Xiaofei Xie, and Shangqing Liu. Demystifying memorization in llm-based program repair via a general hypothesis testing framework. *Proceedings of the ACM on Software Engineering*, 2(FSE):2712–2734, 2025.

- [153] Mingyue Yuan, Jieshan Chen, Zhenchang Xing, Aaron Quigley, Yuyu Luo, Tianqi Luo, Gelareh Mohammadi, Qinghua Lu, and Liming Zhu. Designrepair: Dual-stream design guideline-aware frontend repair with large language models. *arXiv preprint arXiv:2411.01606*, 2024.
- [154] Haoran Liu, Shanshan Li, Zhouyang Jia, Yuanliang Zhang, Linxiao Bai, Si Zheng, Xiaoguang Mao, and Xiangke Liao. Error delayed is not error handled: Understanding and fixing propagated error-handling bugs. *Proceedings of the ACM on Software Engineering*, 2(FSE):2571–2593, 2025.
- [155] Sungmin Kang, Bei Chen, Shin Yoo, and Jian-Guang Lou. Explainable automated debugging via large language model-driven scientific debugging. *arXiv preprint arXiv:2304.02195*, 2023.
- [156] Cheryl Lee, Chunqiu Steven Xia, Longji Yang, Jen-tse Huang, Zhouruixin Zhu, Lingming Zhang, and Michael R Lyu. A unified debugging approach via llm-based multi-agent synergy. *arXiv preprint arXiv:2404.17153*, 2024.
- [157] Zhuolin Xu, Yuanzhang Lin, Qiushi Li, and Shin Hwei Tan. Guiding ChatGPT to fix web ui tests via explanation-consistency checking. *arXiv preprint arXiv:2312.05778*, 2023.
- [158] Zhihao Lin, Mingyi Zhou, Wei Ma, Chi Chen, Yun Yang, Jun Wang, Chunming Hu, and Li Li. Haprepair: Learn to repair openharmony apps. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, pages 319–330, 2025.
- [159] Ramtin Ehsani, Esteban Parra, Sonia Haiduc, and Preetha Chatterjee. Bug fixing with broader context: Enhancing llm-based program repair via layered knowledge injection. *arXiv preprint arXiv:2506.24015*, 2025.
- [160] Fengjie Li, Jiajun Jiang, Jiajun Sun, and Hongyu Zhang. Hybrid automated program repair by combining large language models and program analysis. *ACM Transactions on Software Engineering and Methodology*, 34(7):1–28, 2025.
- [161] Qiong Feng, Xiaotian Ma, Jiayi Sheng, Ziyuan Feng, Wei Song, and Peng Liang. Integrating various software artifacts for better llm-based bug localization and program repair. *ACM Transactions on Software Engineering and Methodology*, 2025.
- [162] Kyle Wong, Alfonso Amayuelas, Liangming Pan, and William Yang Wang. Investigating the transferability of code repair for low-resource programming languages. *arXiv preprint arXiv:2406.14867*, 2024.
- [163] Shuyin Ouyang, Jie M Zhang, Zeyu Sun, and Albert Merono Penuela. Knowledge-enhanced program repair for data science code. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 898–910. IEEE, 2025.
- [164] Zhenlong Dai, Bingrui Chen, Zhuoluo Zhao, Xiu Tang, Sai Wu, Chang Yao, Zhipeng Gao, and Jingyuan Chen. Less is more: Adaptive program repair with bug localization and preference learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 128–136, 2025.
- [165] Wei Tao, Yucheng Zhou, Yanlin Wang, Wenqiang Zhang, Hongyu Zhang, and Yu Cheng. Magis: Llm-based multi-agent framework for github issue resolution. *Advances in Neural Information Processing Systems*, 37:51963–51993, 2024.
- [166] Boyang Yang, Haoye Tian, Jiadong Ren, Hongyu Zhang, Jacques Klein, Tegawendé F Bissyandé, Claire Le Goues, and Shunfu Jin. Morepair: Teaching llms to repair code via multi-objective fine-tuning. *arXiv preprint arXiv:2404.12636*, 2024.
- [167] Kaiyao Ke. Niodebugger: A novel approach to repair non-idempotent-outcome tests with llm-based agent. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 762–762. IEEE Computer Society, 2025.
- [168] Lianghong Guo, Wei Tao, Runhan Jiang, Yanlin Wang, Jiachi Chen, Xilin Liu, Yuchi Ma, Mingzhi Mao, Hongyu Zhang, and Zibin Zheng. Omnigirl: A multilingual and multimodal benchmark for github issue resolution. *Proceedings of the ACM on Software Engineering*, 2(ISSTA):24–46, 2025.
- [169] Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741*, 2024.
- [170] Zheng Yu, Ziyi Guo, Yuhang Wu, Jiahao Yu, Meng Xu, Dongliang Mu, Yan Chen, and Xinyu Xing. Patchagent: A practical program repair agent mimicking human expertise. In *Proceedings of the 34th USENIX Security Symposium (USENIX Security’25)*, Seattle, WA, USA, 2025.
- [171] Spandan Garg, Roshanak Zilouchian Moghaddam, and Neel Sundaresan. Rapgen: An approach for fixing code inefficiencies in zero-shot. *arXiv preprint arXiv:2306.17077*, 2023.
- [172] Jiayi Zhang, Kai Huang, Jian Zhang, Yang Liu, and Chunyang Chen. Repair ingredients are all you need: Improving large language model-based program repair via repair ingredients search. *arXiv preprint arXiv:2506.23100*, 2025.
- [173] Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. Repairagent: An autonomous, llm-based agent for program repair. *arXiv preprint arXiv:2403.17134*, 2024.
- [174] André Silva, Sen Fang, and Martin Monperrus. RepairLLaMA: Efficient representations and fine-tuned adapters for program repair. *arXiv preprint arXiv:2312.15698*, 2023.
- [175] Siru Ouyang, Wenhao Yu, Kaixin Ma, Zilin Xiao, Zhihan Zhang, Mengzhao Jia, Jiawei Han, Hongming Zhang, and Dong Yu. Repograph: Enhancing ai software engineering with repository-level code graph. *arXiv preprint arXiv:2410.14684*, 2024.

- [176] Pantazis Deligiannis, Akash Lal, Nikita Mehrotra, and Aseem Rastogi. Fixing rust compilation errors using llms. *arXiv preprint arXiv:2308.05177*, 2023.
- [177] Kai Huang, Jian Zhang, Xiaofei Xie, and Chunyang Chen. Seeing is fixing: Cross-modal reasoning with multimodal llms for visual software issue fixing. *arXiv preprint arXiv:2506.16136*, 2025.
- [178] John Yang, Carlos E Jimenez, Alex L Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriel Synnaeve, Karthik R Narasimhan, et al. Swe-bench multimodal: Do ai systems generalize to visual software domains? *arXiv preprint arXiv:2410.03859*, 2024.
- [179] Yingwei Ma, Rongyu Cao, Yongchang Cao, Yue Zhang, Jue Chen, Yibo Liu, Yuchen Liu, Binhua Li, Fei Huang, and Yongbin Li. Swe-gpt: A process-centric language model for automated software improvement. *Proceedings of the ACM on Software Engineering*, 2(ISSTA):2362–2383, 2025.
- [180] Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025.
- [181] Antonis Antoniadis, Albert Örwall, Kexun Zhang, Yuxi Xie, Anirudh Goyal, and William Wang. Swe-search: Enhancing software agents with monte carlo tree search and iterative refinement. *arXiv preprint arXiv:2410.20285*, 2024.
- [182] Niels Mündler, Mark Müller, Jingxuan He, and Martin Vechev. Swt-bench: Testing and validating real-world bug-fixes with code agents. *Advances in Neural Information Processing Systems*, 37:81857–81887, 2024.
- [183] Amiao Gao, Zenong Zhang, Simin Wang, Liguang Huang, Shiyi Wei, and Vincent Ng. Teaching ai the ‘why’ and ‘how’ of software vulnerability fixes. *Proceedings of the ACM on Software Engineering*, 2(FSE):2006–2029, 2025.
- [184] Kai Huang, Jian Zhang, Xiangxin Meng, and Yang Liu. Template-guided program repair in the era of large language models. In *ICSE*, pages 1895–1907, 2025.
- [185] Fernando Vallecillos Ruiz, Max Hort, and Leon Moonen. The art of repair: Optimizing iterative program repair with instruction-tuned models. *arXiv preprint arXiv:2505.02931*, 2025.
- [186] Nikhil Parasaram, Huijie Yan, Boyu Yang, Zineb Flahy, Abriele Qudsi, Damian Ziaber, Earl T Barr, and Sergey Mechtaev. The fact selection problem in llm-based program repair. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 2574–2586. IEEE, 2025.
- [187] Roman Macháček, Anastasiia Grishina, Max Hort, and Leon Moonen. The impact of fine-tuning large language models on automated program repair. *arXiv preprint arXiv:2507.19909*, 2025.
- [188] Ramtin Ehsani, Sakshi Pathak, and Preetha Chatterjee. Towards detecting prompt knowledge gaps for improved llm-guided issue resolution. In *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*, pages 699–711. IEEE, 2025.
- [189] Wenqiang Luo, Jacky Keung, Boyang Yang, He Ye, Claire Le Goues, Tegawende F Bissyande, Haoye Tian, and Xuan Bach D Le. When fine-tuning llms meets data privacy: An empirical study of federated learning in llm-based program repair. *ACM Transactions on Software Engineering and Methodology*, 2024.