

Pengyu Nie

Research Interests

Software Engineering (SE), Artificial Intelligence (AI), AI4SE and SE4AI

The goal of my research is to improve developers' productivity during the design, testing, and maintenance of software and AI systems. My recent work focuses on (1) execution-guided and evolution-aware machine learning models for SE, and (2) AI4SE-assisted maintenance of AI systems.

Employments

2023–present **Assistant Professor**, *University of Waterloo*, Cheriton School of Computer Science, Waterloo, Ontario, Canada

Education

2017–2023 **M.Sc. & Ph.D. in ECE**, *The University of Texas at Austin*, USA

Advisor: Prof. Milos Gligoric

Thesis: Machine Learning for Executable Code in Software Testing and Verification

2013–2017 **B.Sc. in Physics**, *The University of Science and Technology of China*, China

Advisors: Prof. Ya Wang and Prof. Jiangfeng Du

Thesis: Long Distance Entanglement of NV Center Qubits via Ferromagnet

Publications

- [39] Bihui Jin, Kaiyuan Wang, and **Pengyu Nie**. Automated Modernization of Machine Learning Engineering Notebooks for Reproducibility. In International Symposium on Software Testing and Analysis (ISSTA), to appear. 2026. Oakland, USA.
- [38] Jiale Amber Wang, Kaiyuan Wang, and **Pengyu Nie**. TestEvo-Bench: An Executable and Live Benchmark for Test and Code Co-Evolution. In International Conference on AI-powered Software (Alware), ArXiv Track @ FSE. 2026. Montreal, Canada.
- [37] Wentao Zhang, Liliana Hotsko, Woojeong Kim, **Pengyu Nie**, Stuart Shieber, and Yuntian Deng. Program-as-Weights: A Programming Paradigm for Fuzzy Functions. In International Conference on AI-powered Software (Alware), ArXiv Track @ FSE. 2026. Montreal, Canada.
- [36] Yixin Li, Yuntian Deng, and **Pengyu Nie**. TokDrift: When LLM Speaks in Subwords but Code Speaks in Grammar. In Annual Meeting of the Association for Computational Linguistics (ACL), to appear. 2026. San Diego, USA.
- [35] Xiaoyu Cheng, Kundi Yao, **Pengyu Nie**, and Weiyi Shang. When AI Coding Assistants Leak Training Data: A Study of LLM Memorization in Code Generation. In International Conference on AI-powered Software (Alware) @ FSE, to appear. 2026. Montreal, Canada.
- [34] Bihui Jin, Heng Li, **Pengyu Nie**, and Ying Zou. Energy-Efficient Software Development: A Multi-dimensional Empirical Analysis of Stack Overflow. In International Conference on Software Engineering (ICSE). 2026. Rio de Janeiro, Brazil. doi:10.1145/3744916.3773177.
- [33] Yinxi Li, Jiaao Chen, Fang Wu, Jiakai Yu, Heli Qi, Weihao Xuan, Haokai Zhao, **Pengyu Nie**, Di Jin, and Xiangru Tang. Learning Multi-step Reasoning via Persistent Latent State Propagation. In

Workshop on Latent & Implicit Thinking—Going Beyond CoT Reasoning (LIT) @ ICLR. 2026. Rio de Janeiro, Brazil.

- [32] Xiaohui Wang, Kundi Yao, Lizhi Liao, **Pengyu Nie**, and Weiyi Shang. World of Logs: A Dataset of Logs from Online Documents. In International Conference on Mining Software Repositories (MSR), Data and Tool Showcase Track, to appear. 2026. Rio de Janeiro, Brazil.
- [31] Liliana Hotsko, Yinxi Li, Yuntian Deng, and **Pengyu Nie**. Code2LoRA: Hypernetwork-Generated Adapters for Code Language Models under Software Evolution. arXiv preprint (arXiv). 2026.
- [30] Chi-en Amy Tai, **Pengyu Nie**, Lukasz Golab, and Alexander Wong. NL in the Middle: Code Translation with LLMs and Intermediate Representations. In International Conference on Collaborative Advances in Software and Computing (CASCON), pages 295–300. 2025. Toronto, Canada. doi:10.1109/cascon66301.2025.00037.
- [29] Bihui Jin*, Jiayue Wang*, and **Pengyu Nie**. Learning to Edit Interactive Machine Learning Notebooks. In International Conference on the Foundations of Software Engineering (FSE), Ideas, Visions and Reflections Track, pages 681–685. 2025. Trondheim, Norway. doi:10.1145/3696630.3728523.
- [28] Linghan Zhong, Samuel Yuan, Jiyang Zhang, Yu Liu, **Pengyu Nie**, Junyi Jessy Li, and Milos Gligoric. A Tool for Generating Exceptional Behavior Tests With Large Language Models. In International Conference on the Foundations of Software Engineering (FSE), Demonstrations Track, pages 1193–1197. 2025. Trondheim, Norway. doi:10.1145/3696630.3728608.
- [27] Yifan Zong, Yuntian Deng, and **Pengyu Nie**. Mix-of-Language-Experts Architecture for Multilingual Programming. In International Workshop on Large Language Models for Code (LLM4Code), pages 200–208. 2025. Ottawa, Canada. doi:10.1109/llm4code66737.2025.00030.
- [26] Kaihang Jiang, Bihui Jin, and **Pengyu Nie**. CoUpJava: A Dataset of Code Upgrade Histories in Open-Source Java Repositories. In International Conference on Mining Software Repositories (MSR), Data and Tool Showcase Track, pages 441–445. 2025. Ottawa, Canada. doi:10.1109/msr66628.2025.00075.
- [25] Jiyang Zhang, Yu Liu, **Pengyu Nie**, Junyi Jessy Li, and Milos Gligoric. exLong: Generating Exceptional Behavior Tests with Large Language Models. In International Conference on Software Engineering (ICSE), pages 1462–1474. 2025. Ottawa, Canada. doi:10.1109/icse55347.2025.00176.
- [24] Marcos Macedo, Yuan Tian, **Pengyu Nie**, Filipe R. Cogo, and Bram Adams. InterTrans: Leveraging Transitive Intermediate Translations to Enhance LLM-based Code Translation. In International Conference on Software Engineering (ICSE), pages 1153–1164. 2025. Ottawa, Canada. doi:10.1109/icse55347.2025.00236.
- [23] Tongwei Zhang, Kundi Yao, Hanyang Hu, **Pengyu Nie**, Krishna Koravadi, and Weiyi Shang. Detecting DTC Requirement-Implementation Inconsistencies Using LLMs: An Experience Report. IEEE Software (IEEE Softw.), pages 103–111. 2025. doi:10.1109/ms.2025.3631386.
- [22] Saarang Agarwal, **Pengyu Nie**, and Meiyappan Nagappan. What Inputs Drive Effective LLM-based Unit Test Generation?. IEEE Software (IEEE Softw.), Special Issue on Alware in the FM Era, pages 48–56. 2025. doi:10.1109/ms.2025.3621625.
- [21] Jiale Amber Wang*, Kaiyuan Wang*, and **Pengyu Nie**. Efficient Incremental Code Coverage Analysis for Regression Test Suites. In International Conference on Automated Software Engineering (ASE), pages 1882–1894. 2024. Sacramento, USA. doi:10.1145/3691620.3695551.
- [20] Jiyang Zhang, **Pengyu Nie**, Junyi Jessy Li, and Milos Gligoric. Multilingual Code Co-evolution using Large Language Models. In Joint Meeting on European Software Engineering Conference and

- Symposium on the Foundations of Software Engineering (ESEC/FSE), pages 695–707. 2023. San Francisco, USA. doi:10.1145/3611643.3616350.
- [19] **Pengyu Nie**. Machine Learning for Executable Code in Software Testing and Verification. PhD Dissertation, The University of Texas at Austin. 2023. Austin, USA. This dissertation won a **Margarida Jacome Dissertation Award**.
- [18] Yu Liu, **Pengyu Nie**, Anna Guo, Milos Gligoric, and Owolabi Legunsen. Extracting Inline Tests from Unit Tests. In International Symposium on Software Testing and Analysis (ISSTA), pages 1458–1470. 2023. Seattle, USA. doi:10.1145/3597926.3598149.
- [17] Yu Liu, Jiyang Zhang, **Pengyu Nie**, Milos Gligoric, and Owolabi Legunsen. More Precise Regression Test Selection via Reasoning about Semantics-Modifying Changes. In International Symposium on Software Testing and Analysis (ISSTA), pages 664–676. 2023. Seattle, USA. doi:10.1145/3597926.3598086. This paper won an **ACM Distinguished Paper Award**.
- [16] **Pengyu Nie**, Rahul Banerjee, Junyi Jessy Li, Raymond J. Mooney, and Milos Gligoric. Learning Deep Semantics for Test Completion. In International Conference on Software Engineering (ICSE), pages 2111–2123. 2023. Melbourne, Australia. doi:10.1109/ICSE48619.2023.00178.
- [15] Yu Liu, Zachary Thurston, Alan Han, **Pengyu Nie**, Milos Gligoric, and Owolabi Legunsen. pytest-inline: An Inline Testing Tool for Python. In International Conference on Software Engineering (ICSE), Tool Demonstrations Track, pages 161–164. 2023. Melbourne, Australia. doi:10.1109/ICSE-Companion58688.2023.00046.
- [14] Yu Liu, **Pengyu Nie**, Owolabi Legunsen, and Milos Gligoric. Inline Tests. In International Conference on Automated Software Engineering (ASE), pages 57:1–13. 2022. Rochester, USA. doi:10.1145/3551349.3556952.
- [13] Jiyang Zhang, Sheena Panthaplackel, **Pengyu Nie**, Junyi Jessy Li, and Milos Gligoric. CodiT5: Pretraining for Source Code and Natural Language Editing. In International Conference on Automated Software Engineering (ASE), pages 22:1–12. 2022. Rochester, USA. doi:10.1145/3551349.3556955.
- [12] **Pengyu Nie**, Jiyang Zhang, Junyi Jessy Li, Raymond J. Mooney, and Milos Gligoric. Impact of Evaluation Methodologies on Code Summarization. In Annual Meeting of the Association for Computational Linguistics (ACL), pages 4936–4960. 2022. Dublin, Ireland.
- [11] **Pengyu Nie**, Karl Palmskog, Junyi Jessy Li, and Milos Gligoric. Roosterize: Suggesting Lemma Names for Coq Verification Projects using Deep Learning. In International Conference on Software Engineering (ICSE), Tool Demonstrations Track, pages 21–24. 2021. Madrid, Spain. doi:10.1109/ICSE-Companion52605.2021.00026.
- [10] Jiyang Zhang, Sheena Panthaplackel, **Pengyu Nie**, Junyi Jessy Li, Raymond J. Mooney, and Milos Gligoric. Leveraging Class Hierarchy for Code Comprehension. In Workshop on Computer Assisted Programming (CAP). 2020. Virtual, USA.
- [9] **Pengyu Nie**, Marinela Parovic, Zhiqiang Zang, Sarfraz Khurshid, Aleksandar Milicevic, and Milos Gligoric. Unifying Execution of Imperative Generators and Declarative Specifications. In Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA), pages 217:1–217:26. 2020. Chicago, USA.
- [8] Jaeseong Lee*, **Pengyu Nie***, Junyi Jessy Li, and Milos Gligoric. On the Naturalness of Hardware Descriptions. In Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), pages 530–542. 2020. Sacramento, USA.
- [7] **Pengyu Nie**, Ahmet Celik, Matthew Coley, Aleksandar Milicevic, Jonathan Bell, and Milos Gligoric. Debugging the Performance of Maven’s Test Isolation: Experience Report. In International Symposium

on Software Testing and Analysis (ISSTA), pages 249–259. 2020. Los Angeles, USA.

- [6] Sheena Panthaplackel, **Pengyu Nie**, Milos Gligoric, Junyi Jessy Li, and Raymond J. Mooney. Learning to Update Natural Language Comments Based on Code Changes. In Annual Meeting of the Association for Computational Linguistics (ACL), pages 1853–1868. 2020. Seattle, USA. doi:10.18653/v1/2020.acl-main.168.
- [5] **Pengyu Nie**, Karl Palmiskog, Junyi Jessy Li, and Milos Gligoric. Deep Generation of Coq Lemma Names using Elaborated Terms. In International Joint Conference on Automated Reasoning (IJCAR), pages 97–118. 2020. Paris, France. doi:10.1007/978-3-030-51054-1_6.
- [4] **Pengyu Nie**, Karl Palmiskog, Junyi Jessy Li, and Milos Gligoric. Learning to Format Coq Code using Language Models. In The Coq Workshop (Coq). 2020. Paris, France.
- [3] Ahmet Celik, **Pengyu Nie**, Christopher J. Rossbach, and Milos Gligoric. Design, Implementation, and Application of GPU-based Java Bytecode Interpreters. In Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA), pages 177:1–177:28. 2019. Athens, Greece.
- [2] **Pengyu Nie**, Rishabh Rai, Junyi Jessy Li, Sarfraz Khurshid, Raymond J. Mooney, and Milos Gligoric. A Framework for Writing Trigger-Action Todo Comments in Executable Format. In Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), pages 385–396. 2019. Tallinn, Estonia. This paper won an **ACM SIGSOFT Distinguished Paper Award**.
- [1] **Pengyu Nie**, Junyi Jessy Li, Sarfraz Khurshid, Raymond J. Mooney, and Milos Gligoric. Natural Language Processing and Program Analysis for Supporting Todo Comments as Software Evolves. In Workshop on Natural Language Processing for Software Engineering (NL4SE), pages 775–778. 2018. New Orleans, USA.

Grants

- 2024–2030 NSERC Discovery Grant (with Supplemental Grant), CAD\$204,500; *Source*: Natural Sciences and Engineering Research Council of Canada
- 2023–2028 University of Waterloo Start-Up Funding, CAD\$150,000; *Source*: Cheriton School of Computer Science, University of Waterloo
- 2025–2026 Generating Executable Test Plans from Multi-Modal Software Requirements, USD\$10,000; *Source*: Dragon Testing, 5608 17th Ave NW. Suite 1512, Seattle WA 98107, USA

Awards

- 2023 Margarida Jacome Dissertation Award for [19]
- 2023 ACM SIGSOFT Distinguished Paper Award for [17] at ISSTA'23
- 2020–2021 The University of Texas at Austin Graduate School Continuing Fellowship
- 2019 ACM SIGSOFT Distinguished Paper Award for [2] at FSE'19
- 2017–2022 The University of Texas at Austin Graduate School Fellowship
- 2014, 2015, 2016 Outstanding Students Scholarship, USTC
- 2013 Outstanding Freshmen Scholarship, USTC

Research Community Service

■ Conference Peer Reviews

- ICSE 2027 Program Committee, International Conference on Software Engineering
- FSE 2027 Program Committee, International Conference on the Foundations of Software Engineering
- ICSE 2026 Program Committee, International Conference on Software Engineering
- ASE 2026 Program Committee, International Conference on Automated Software Engineering
- ISSTA 2026 Program Committee, International Symposium on Software Testing and Analysis
- ISSTA 2025 Program Committee, International Symposium on Software Testing and Analysis
- LLM4Code 2025 Program Committee, International Workshop on Large Language Models for Code
- ASE 2024 Program Committee, International Conference on Automated Software Engineering
- ISSTA 2024 Program Committee, International Symposium on Software Testing and Analysis
- LLM4Code 2024 Program Committee, International Workshop on Large Language Models for Code
- ASE-SRC 2024 Program Committee, International Conference on Automated Software Engineering, Student Research Competition
- AAAI 2021 Program Committee, AAAI Conference on Artificial Intelligence
- NLP4Prog 2021 Program Committee, Workshop on Natural Language Processing for Programming
- AIST 2021 Program Committee, Artificial Intelligence in Software Testing Workshop
- ICLR 2025 Reviewer, International Conference on Learning Representations
- ACL 2025 Reviewer: ACL Rolling Review February (ACL'25)
- ACL 2024 Reviewer: ACL Rolling Review February (ACL'24), June (EMNLP'24), and December (ACL'25); Emergency Area Chair of ARR December

■ Journal Peer Reviews

- TOSEM ACM Transactions on Software Engineering and Methodology
- TSE IEEE Transactions on Software Engineering
- EMSE Empirical Software Engineering
- TOPLAS ACM Transactions on Programming Languages and Systems
- TACL Transactions of the Association for Computational Linguistics
- EAAI Engineering Applications of Artificial Intelligence
- IST Information and Software Technology
- JSS Journal of Systems and Software

Professional Activities

- Co-organizer Joint UT-Cornell Software Engineering Seminar (2022–2023)
- Co-organizer NLP+Programming Reading Group at UT Austin (2018–2022)
- Committee Graduate and Industry Networking (GAIN) at UT Austin (2022)
- Mentor ECE Partner Program at UT Austin (2022)

Teaching Experience

- Jan–Apr 2025 **Instructor**, UWaterloo, CS446/CS646/ECE452 Software Design and Architecture - <https://pengyunie.github.io/cs446-1251/>
- Sep–Dec 2024 **Instructor**, UWaterloo, CS846 Advanced Topics in Software Engineering: Machine Learning for Software Engineering - <https://pengyunie.github.io/cs846mlse-1249/>
- Jan–Apr 2024 **Instructor**, UWaterloo, CS446/CS646/ECE452 Software Design and Architecture - <https://pengyunie.github.io/cs446-1241/>
- Aug–Dec 2021 **Teaching assistant**, UT Austin, ECE382V Programming Paradigms

Mentoring Experience

■ Current Students

- Bihui Jin PhD 2024–2028
- Daniel Phan MMath 2024–2025
- Jiale Amber Wang MMath 2024–2025
- Saarang Agarwal MMath 2024–2025; co-supervised with Prof. Meiyappan Nagappan
- Larry Yinxi Li MMath 2025–2026
- Chaitanya Shahane Master's @ UVA 2024–2025; co-supervised with Prof. Wenxi Wang
- Liliana Hotsko Undergrad Internship 2025; co-supervised with Prof. Yuntian Deng

■ Previous Students

- Yifan Zong URF/URA 2024–2025; co-supervised with Prof. Yuntian Deng; first job: xAI
- Angel Liu URF/URA 2024–2025; grad school: NYU
- Kaihang Jiang URF 2024; grad school: Caltech
- Jiayue Wang URF/URA 2024–2025; grad school: CMU

■ Students Mentored during My PhD

- Yu Liu PhD 2021–2023; first job: Meta
- Jiyang Zhang PhD 2019–2023; first job: Google
- Rahul Banerjee Undergrad 2022–2023; first job: Amazon
- Kush Jain Undergrad 2019–2021; grad school: CMU
- Rishabh Rai Undergrad 2017–2018; first job: Salesforce

Presentations

- May 2025 *exLong: Generating Exceptional Behavior Tests with Large Language Models* [25], at ICSE 2025, Ottawa, Canada
- Oct 2024 *Execution-Guided LLMs for Test Completion and Generation*, guest lecture at Cornell CS6158 Software Engineering in the Era of Machine Learning, Ithaca, USA
- Sep 2024 *Software Testing & Machine Learning*, guest lecture at UVA CS6501 Machine Learning for Software Reliability, Charlottesville, USA
- Jun-Sep 2024 *Software Development, Evolution, and Testing in the Era of Large Language Models*, at CSER 2024 Spring, Kingston, Canada; CanSTARE 2024, The Blue Mountains, Canada
- May 2023 *Learning deep semantics for test completion* [16], at ICSE 2023, Melbourne, Australia
- May 2023 *pytest-inline: An inline testing tool for Python* [15], at ICSE 2023, Melbourne, Australia

- Jan–Mar 2023 *Execution-guided learning for software development, testing, and maintenance*, at HKUST (Guangzhou Campus), Guangzhou, China; ISU, Ames, USA; UIC, Chicago, USA; UVA, Charlottesville, USA; NUS, Singapore; UIUC, Urbana, USA; CUHK, Hong Kong, China; UWaterloo, Waterloo, Canada; UTD, Dallas, USA
- Dec 2022 *Execution-guided learning for evolving software*, at UTD, Dallas, USA; UTSA, San Antonio, USA
- Nov 2022 *A framework for writing trigger-action todo comments in executable format*, guest lecture at UT Austin ECE382V Engineering Program Analysis, Austin, USA
- Aug 2022 *Demo: Using the huggingface/transformers library*, guest lecture at UT Austin ECEw382V Machine Programming, Austin, USA
- Jul 2022 *On the naturalness of hardware descriptions*, guest lecture at UT Austin ECEw382V Machine Programming, Austin, USA
- Jun 2022 *Unifying execution of imperative generators and declarative specifications* [9], at PLDI 2022 (SIGPLAN Track), San Diego, USA
- May 2022 *PyKokkos: Performance portable kernels in Python*, at ICSE 2022, Pittsburgh, USA
- May 2022 *Impact of evaluation methodologies on code summarization* [12], at ACL 2022, Dublin, Ireland
- Sep 2021 *Improving proof assistant users' productivity using language models* [4, 5], at ITU, Copenhagen, Denmark
- May 2021 *Roosterize: Suggesting lemma names for Coq verification projects using deep learning* [11], at ICSE 2021
- Nov 2020 *Suggesting Coq lemma names using deep learning* [5], at UIUC, Urbana, USA
- Nov 2020 *Unifying execution of imperative generators and declarative specifications* [9], at OOPSLA 2020
- Nov 2020 *On the naturalness of hardware descriptions* [8], at FSE 2020
- Jul 2020 *Debugging the performance of Maven's test isolation: Experience report* [7] at ISSTA 2020
- Jul 2020 *Learning to format Coq code using language models* [4], at The Coq Workshop 2020
- Jun 2020 *Deep generation of Coq lemma names using elaborated terms* [5], at IJCAR 2020
- Nov 2019 *Mutation analysis for Coq*, at ASE 2019, San Diego, USA
- Aug 2019 *A framework for writing trigger-action todo comments in executable format* [2], at SUSTech, Shenzhen, China
- Aug 2019 *A framework for writing trigger-action todo comments in executable format* [2], at FSE 2019, Tallinn, Estonia
- Aug 2019 *A framework for writing trigger-action todo comments in executable format* [2], at LIP6, Paris, France
- Feb 2018 *Natural language processing and program analysis for supporting todo comments as software evolves* [1], at NL4SE 2018 (long presentation), New Orleans, USA

Industry Experience

- May–Aug 2021 **Software Engineer Intern**, *Facebook*, remotely from Austin
Manager: Georgios Gousios *Peers:* Edward Yao
 Detecting anomaly states in multivariate timeseries with variational auto-encoders
- Jun–Aug 2020 **Software Engineer Intern**, *Facebook*, remotely from Austin
Manager: Vijayaraghavan Murali *Peers:* Lee Gross, Vivek Nair
 Identifying bug-introducing commits given the bug reports and bug-fixing commits, by using weak supervision framework to combine both ML-based and heuristics-based approaches

Jul–Aug 2016 **Core Technology Researcher (Internship)**, *iFlytek Research*, Hefei
Manager: Pu Yang *Peers*: Feiwu Li
Music rhythm extraction using Hidden Markov Model

Open Source Contributions

- 2025 **MoLE**, mix-of-language-experts architecture for multilingual programming - <https://github.com/uw-swag/mole> (paper [27])
- 2025 **CoUpJava**, a dataset of code upgrade histories in open-source Java repositories - <https://github.com/uw-swag/CoUpJava> (paper [26])
- 2025 **exLong**, a tool for generating exceptional behavior tests with large language models - <https://github.com/EngineeringSoftware/exLong> (paper [25])
- 2024 **iJaCoCo**, an efficient incremental code coverage analysis tool for Java - <https://github.com/uw-swag/ijacoco> (paper [21])
- 2023 **TeCo**, ML+execution model for test completion - <https://github.com/EngineeringSoftware/teco> (paper [16])
- 2022 **time-segmented-evaluation**, a tool for setting up time-segmented evaluation for code summarization ML models - <https://github.com/EngineeringSoftware/time-segmented-evaluation> (paper [12])
- 2022 **pytest-inline**, a full-fledged inline testing framework in Python; a plugin of pytest - <https://github.com/pytest-dev/pytest-inline> (paper [15])
- 2022 **I-Test**, the first inline testing framework in Python and Java - <https://github.com/EngineeringSoftware/inlinetest> (paper [14])
- 2022 **CoditT5**, pretrained LLM for editing code and natural language - <https://github.com/EngineeringSoftware/CoditT5> (paper [13])
- 2021 **Roosterize-vscode**, VSCode extension for Roosterize (suggesting lemma names in Coq) - <https://github.com/EngineeringSoftware/roosterize-vscode> (paper [11])
- 2020 **HDLP**, data used to measure naturalness of hardware descriptions, and deep learning models for assignment completion in VHDL - <https://github.com/EngineeringSoftware/hdlp> (paper [8])
- 2020 **Roosterize**, a tool for suggesting lemma names for the Coq proof assistant - <https://github.com/EngineeringSoftware/roosterize> (paper [5])
- 2020 **MathComp Corpus**, a corpus of Coq verification projects' source code along with compile-time and runtime machine-readable representations, derived from MathComp - <https://github.com/EngineeringSoftware/math-comp-corpus> (paper [5])
- 2020 **LearningToUpdateNLComments**, a ML model for updating natural language comments based on code changes - <https://github.com/panthap2/LearningToUpdateNLComments> (paper [6])
- 2019 **seutil**, a Python library supporting ML/NLP + SE research - <https://github.com/pengyunie/seutil>