

# Pengyu Nie

Assistant Professor  
Cheriton School of Computer Science, University of Waterloo

200 University Avenue West, DC 3348  
Waterloo, Ontario N2L 3G1, Canada  
📞 +1 519 503 6636  
✉️ pynie@uwaterloo.ca  
🌐 pengyunie.github.io

## Research Interests

### Software Engineering, Machine Learning, Natural Language Processing

My focus is on improving developers' productivity during software development, testing, and maintenance. Specific topics include execution-guided machine learning models for testing and verification, learning to evolve code and comments, and frameworks for executable comments and specifications.

## Education

2017–2023 **M.Sc. & Ph.D.**, *The University of Texas at Austin*  
Advisor: Prof. Milos Gligoric

2013–2017 **B.Sc.**, *The University of Science and Technology of China (USTC)*  
Advisor: Prof. Ya Wang and Prof. Jiangfeng Du

## Publications

- [22] 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)*. 2025.
- [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)*, 1882–1894. 2024.
- [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)*, 695–707. 2023.
- [19] **Pengyu Nie**. *Machine Learning for Executable Code in Software Testing and Verification*. Ph.D. thesis, The University of Texas at Austin, 2023.
- [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)*, 1458–1470. 2023.
- [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)*, 664–676. 2023.
- [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)*, 2111–2123. 2023.
- [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, Tool Demonstrations Track (ICSEDemo)*, 161–164. 2023.

- [14] Yu Liu, **Pengyu Nie**, Owolabi Legunsen, and Milos Gligoric. Inline Tests. In *International Conference on Automated Software Engineering (ASE)*, 57:1–13. 2022.
- [13] Jiyang Zhang, Sheena Panthaplackel, **Pengyu Nie**, Junyi Jessy Li, and Milos Gligoric. CoditT5: Pretraining for Source Code and Natural Language Editing. In *International Conference on Automated Software Engineering (ASE)*, 22:1–12. 2022.
- [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)*, 4936–4960. 2022.
- [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, Tool Demonstrations Track (ICSEDemo)*, 21–24. 2021.
- [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 @ NeurIPS*. 2020.
- [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)*, 217:1–217:26. 2020.
- [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)*, 530–542. 2020.
- [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)*, 249–259. 2020.
- [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)*, 1853–1868. 2020.
- [5] **Pengyu Nie**, Karl Palmskog, Junyi Jessy Li, and Milos Gligoric. Deep Generation of Coq Lemma Names using Elaborated Terms. In *International Joint Conference on Automated Reasoning (IJCAR)*, 97–118. 2020.
- [4] **Pengyu Nie**, Karl Palmskog, Junyi Jessy Li, and Milos Gligoric. Learning to Format Coq Code using Language Models. In *The Coq Workshop @ IJCAR*. 2020.
- [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)*, 177:1–177:28. 2019.
- [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)*, 385–396. 2019.
- [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 @ AACL*, 775–778. 2018.

## Grants Received

- 2024–2030 NSERC Discovery Grant, \$150,000
- 2024–2025 NSERC Discovery Supplemental Grant, \$12,500
- 2023–2028 University of Waterloo Start-Up Funding, \$150,000

## Scholarships and 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

### ■ Program Committee

- ICSE 2026 International Conference on Software Engineering
- ISSTA 2025 International Symposium on Software Testing and Analysis
- ICLR 2025 International Conference on Learning Representations
- LLM4Code 2025 Workshop on Large Language Models for Code @ ICSE
- ASE 2024 International Conference on Automated Software Engineering
- ISSTA 2024 International Symposium on Software Testing and Analysis
- EMNLP 2024 Empirical Methods in Natural Language Processing (ARR June)
- ACL 2024 Annual Meeting of the Association for Computational Linguistics (ARR February)
- LLM4Code 2024 Workshop on Large Language Models for Code @ ICSE
- ASE SRC 2024 Student Research Competition @ ASE
- AAAI 2021 AAAI Conference on Artificial Intelligence
- NLP4Prog 2021 Workshop on Natural Language Processing for Programming @ ACL-IJCNLP
- AIST 2021 Artificial Intelligence in Software Testing Workshop @ ICST

### ■ Journal Review

TOSEM (2023, 2024), TSE (2022, 2023, 2024), EMSE (2024), IST (2024), TAACL (2023), EAAI (2023), JSS (2023), TOPLAS (2022)

### ■ Sub-Review

ICSE (2019, 2020, 2022, 2023), ASE (2018, 2021), TSE (2021), ISSTA (2019, 2020), COLING (2020), ISSRE (2020), IJCAI (2019), FSE (2018)

---

## Professional Activities

- Seminar Organizer Organizer/Co-organizer of the Joint UT-Cornell Software Engineering Seminar 2022–2023, and the NLP+Programming Reading Group at UT Austin 2018–2022
- Proposal Writing As a Ph.D. student, helped with preparing proposals for NSF and DARPA
- GAIN Committee Member of Graduate and Industry Networking (GAIN) at UT Austin 2022, updated and maintained event website
- Grad Mentor ECE Partner Program at UT Austin 2022 Fall, mentored a group of six first-year graduate students

---

## Teaching Experience

- Jan–Apr 2025 **Instructor**, UWaterloo, CS446/CS646/ECE452 Software Design and Architecture
- 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 Grad Students

- Bihui Jin PhD 2024–2028 (expected)
- Daniel Phan MMath 2024–2025 (expected)
- Jiale Amber Wang MMath 2024–2025 (expected); URF 2024 Winter (Jan–Apr)
- Saarang Agarwal MMath 2024–2025 (expected)

### ■ Current Undergrad Students

- Yifan Zong since Jan 2024; URF 2024 Winter (Jan–Apr)
- Angel Liu since Jan 2024; URF 2024 Spring (May–Aug)
- Kaihang Jiang since May 2024; URF 2024 Spring (May–Aug)
- Jiayue Wang since May 2024

### ■ Students Mentored during My PhD

- Yu Liu PhD 2021–2023; co-authored [14, 15, 17, 18, 22]; first job: Meta
- Jiyang Zhang PhD 2019–2023; co-authored [10, 13, 12, 17, 20, 22]
- Rahul Banerjee Undergrad 2022–2023; co-authored [16]; first job: Amazon
- Kush Jain Undergrad 2019–2021; grad school: CMU
- Rishabh Rai Undergrad 2017–2018; co-authored [2]; first job: Salesforce

## Presentations

- 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

Worked with Georgios Gousios and Edward Yao on detecting anomaly states in multivariate timeseries with variational auto-encoders

Jun–Aug 2020 **Software Engineer Intern**, *Facebook*, remotely from Austin

*Manager*: Vijayaraghavan Murali

Worked with Vijayaraghavan Murali, Lee Gross, and Vivek Nair on 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

Worked with Feiwu Li and Pu Yang on music rhythm extraction using Hidden Markov Model

---

## Open Source Contributions

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>