Atharva Naik

arnaik@andrew.cmu.edu atharva-naik Atharva Naik
https://atharva-naik.github.io/ Google Scholar

Education

2024 – · · · · **Ph.D. Language Technologies, Carnegie Mellon University**.
Advisors: Carolyn Rose, Daniel Fried
GPA: 4.14/4

2022 – 2024 **M.S. Language Technologies, Carnegie Mellon University**
GPA: 4.11/4

2018 – 2022 **B.Tech. Computer Science, Indian Institute of Technology, Kharagpur**
GPA: 9.66/10

Research

Research Interests: Developing synthetic data generation and post-training methods for easy-to-hard generalization and self-improving coding agents across software engineering (linting and code refactoring), data science, and computational linguistics.

Experience

2025 **Research Intern, Oracle Labs** MetaLint: Generalizable Idiomatic Code Quality Analysis with Instruction Following and Easy-to-Hard Generalization

2022 – 2024 **Research Assistant, Carnegie Mellon University** Reinforcement Learning for Code Quality & Review, Reasoning for AI Safety, Programming by Examples for Reasoning

2021 **Research Intern, Technische Universität Darmstadt** Neural Network Architecture for Faithful Interpretability in NLP.

Research Intern, Adobe RL agent for Creative Human-Human Collaboration.

Research Intern, University of Alberta Neuro-Symbolic Fuzzy Logic-based Reasoning for Explainable Natural Language Inference.

2019-2020 **Student Researcher, Autonomous Ground Vehicle (AGV) Group** Path Planning and Localization for Autonomous Driving.

Publications

Conference Publications

1. **A. Naik**, Y. Mathur, D. Agrawal, *et al.*, "Pbebench: A multi-step programming by examples reasoning benchmark inspired by historical linguistics," in ***ACL Findings***, 2026, pp. 8877–8918.
2. S. Vashistha, A. Bibhuti, **A. Naik**, M. Tutek, and S. Aditya, "Pragworld: A benchmark evaluating llms' local world model under minimal linguistic alterations and conversational dynamics," in ***AAAI***, 2026, pp. 33 323–33 331.
3. **A. Naik**, D. Agrawal, H. Sng, *et al.*, "Programming by example meets historical linguistics: A large language model based approach to sound law induction," in ***ACL***, 2025, pp. 29 628–29 647. URL: https://aclanthology.org/2025.acl-long.1432/.
4. **A. Naik**, M. Alenius, D. Fried, and C. Rose, "CRScore: Grounding automated evaluation of code review comments in code claims and smells," in *NAACL*, 2025, pp. 9049–9076. URL: https://aclanthology.org/2025.naacl-long.457/.

5. S. Gandhi, **A. Naik**, Y. Xie, and C. Rose, "An empirical study on strong-weak model collaboration for repo-level code generation," in ***EMNLP***, 2025.
6. **A. Naik**, J. R. Yin, A. Kamath, *et al.*, "Generating situated reflection triggers about alternative solution paths: A case study of generative ai for computer-supported collaborative learning," in ***AIED***, 2024.
7. **A. Naik**, J. R. Yin, A. Kamath, *et al.*, "Providing tailored reflection instructions in collaborative learning using large language models," in ***BJET***, vol. 56, 2024, pp. 531–550.
8. A. Rao, S. Vashistha, **A. Naik**, S. Aditya, and M. Choudhury, "Tricking LLMs into Disobedience: Understanding, Analyzing, and Preventing Jailbreaks," in ***LREC-COLING***, 2024.
9. **A. Naik**, S. Das, J. Vedurada, and S. Aditya, "Sync: A structurally guided hard negative curriculum for generalizable neural code search," in ***AACL***, 2023.
10. Z. Wu, Z. X. Zhang, **A. Naik**, Z. Mei, M. Firdaus, and L. Mou, "Weakly Supervised Explainable Phrasal Reasoning with Neural Fuzzy Logic," in ***ICLR***, 2023.
11. Y. Xie, **A. Naik**, D. Fried, and C. Rose, "CMTrans: Improving Code Translation with Comparable Corpora and Multiple References," in ***EMNLP Findings***, 2023.
12. S. Bv, J. A. Patel, **A. Naik**, Y. Butala, S. Sharma, and N. Chhaya, "Towards Enabling Synchronous Digital Creative Collaboration: Codifying Conflicts in Co-Coloring," in ***CHI Extended Abstracts***, 2022.
13. B. Santra, S. Roychowdhury, A. Mandal, *et al.*, "Representation Learning for Conversational Data using Discourse Mutual Information Maximization," in ***NAACL***, 2022.
14. Y. Wang, S. Mishra, P. Alipoormolabashi, *et al.*, "Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks," in ***EMNLP***, 2022.
15. R. Mukherjee, **A. Naik**, S. Poddar, S. Dasgupta, and N. Ganguly, "Understanding the Role of Affect Dimensions in Detecting Emotions from Tweets: A Multi-task Approach," in ***SIGIR***, 2021.

Preprints

1. **A. Naik**, Y. Mathur, C. Rose, D. Mortensen, *et al.*, "Reacomp: Compiling llm reasoning into symbolic solvers for efficient program synthesis," *arXiv preprint arXiv:2605.05485*, 2026.
2. S. Gandhi, Y. Xie, **A. Naik**, R. Zhu, and C. Rose, "Steer, don't solve: Training small critic models for large code agents," *arXiv preprint arXiv:2606.21811*, 2026.
3. M. N. Kapadnis, L. Baghel, **A. Naik**, and C. Rosé, "Charteditbench: Evaluating grounded multi-turn chart editing in multimodal language models," *arXiv preprint arXiv:2602.15758*, 2026.
4. **A. Naik**, L. Baghel, D. Govindarajan, D. Agrawal, D. Fried, and C. Rose, "Metalint: Generalizable idiomatic code quality analysis through instruction-following and easy-to-hard generalization," *arXiv preprint arXiv:2507.11687*, 2025.
5. **A. Naik**, A. Xie, A. Rao, *et al.*, "Secure and useful models are reasonable: Aligning code models via utility-preserving reasoning," 2025.
6. M. N. Kapadnis, **A. Naik**, and C. Rose, "Crscore++: Reinforcement learning with verifiable tool and ai feedback for code review," *arXiv preprint arXiv:2506.00296*, 2025.
7. **A. Naik**, "On the limitations of embedding based methods for measuring functional correctness for code generation," *arXiv preprint arXiv:2405.01580*, 2024.
8. **A. Naik**, K. Zhang, N. Robinson, *et al.*, "Can large language models code like a linguist?: A case study in low resource sound law induction," 2024. arXiv: `2406.12725 [cs.CL]`.

Projects

MetaDSPRM (Ongoing)

- Investigating **simplicity bias** in data science agents on DSGym Kaggle-style challenges, where models often under-explore and prematurely select suboptimal ML pipelines.
- Developing a **rubric-conditioned process reward model (PRM)** that uses natural-language rubrics to provide flexible, test-time process supervision over agent trajectories without task-specific retraining.
- Exploring **PRM training** on easy trajectories, and evaluating whether rubric-based process signals can support **easy-to-hard transfer** to harder data science workflows.

Evaluating Code Slop in Long-Horizon Coding Agents

- Studied **long-horizon coding agents** beyond task completion, focusing on *code slop*—excess code complexity and **maintainability debt** introduced during multi-step agent workflows.
- Built an evaluation framework on SWE-EVO that jointly measures functional success and static-analysis-based code quality changes across agent trajectories.
- Compared execution-time cleanup strategies and found that different policies can produce similar task success while inducing large variation in code maintainability, motivating **adaptive cleanup mechanisms** for agent workflows.

Steer, Don't Solve

- Developed a **post-training framework** that steers frozen code agents with a lightweight **critic** trained via supervised fine-tuning to provide **intra-trajectory feedback**.
- Demonstrated **transfer across agent families**, improving SWE-bench Verified performance by up to +5.2 points while using a critic 30–92× cheaper than a frontier model.
- Improved Qwen3-Next-80B-A3B accuracy from 20.8% to 25.2% while reducing inference cost by 64% through shorter agent trajectories.

ReaComp

- Developed ReaComp, a framework that compiles a small set of LLM **reasoning traces** into reusable **symbolic program synthesizers** using coding agents.
- Produced standalone **symbolic solvers** that outperform test-time-scaled LLMs on program synthesis (+16.3 points on PBEBench-Hard) while requiring zero LLM inference.
- Improved neuro-symbolic program synthesis by boosting PBEBench-Hard accuracy from 68.4% to 85.8% with 78% fewer tokens, while enabling zero-shot transfer to historical linguistics.

MetaLint

- Developed MetaLint, an **instruction-following framework** for detecting and correcting non-idiomatic Python and Java code. Used **instruction fine-tuning** and **direct preference optimization** for test-time adaptation to novel PEP and JEP idioms.
- Achieved state-of-the-art 70.43% idiom detection recall and 26.73% line-level localization on a PEP benchmark, matching models such as o3-mini with a fine-tuned 4B model.
- Demonstrated **robust generalization** across programming languages, model families, linters, and reasoning settings.

Projects (continued)

PBEBench

- Developed PBEBench, a scalable, **contamination-free benchmark generator** for evaluating LLM **inductive reasoning** and planning via knowledge-free, multi-step string rewrite tasks with **controllable difficulty**.
- Analyzed **reasoning bottlenecks** and **test-time scaling**, measuring performance saturation across state-of-the-art open- and closed-source models, including gpt-oss-120B and GPT-5.
- Showed that while state-of-the-art reasoning models outperform non-reasoning LLMs, none achieve expert human-level performance on realistic historical linguistics tasks.

Skills

Coding	Python (expert), C/C++, Bash (familiar), Javascript (novice)
Frameworks	vLLM, DeepSpeed, PyTorch, HuggingFace, Fairseq, NLTK, spaCy, Tensorflow, FastAPI, Flask, Django, PyQt5, Jupyterlab, OpenCV, Git

Achievements

- 2025 **Amazon Trusted AI Challenge Finalist**, Led the Carnegie Mellon University team to the finals as one of the top four defender teams for developing secure code LLMs.
- **ACL 2025 Oral Presentation**, One of 243 papers selected from over 3,000 accepted submissions.
- 2024 **Amazon Trusted AI Challenge Top 10 Team**, Selected among 90 applicant teams and awarded a $250K research grant as team lead for CMU to advance the development of secure code LLMs.
- **Best Paper & Best Student Paper Nominations**, Artificial Intelligence in Education (AIED 2024).
- 2022 **2nd Place**, Deep Learning Labs OpenAI GPT-3 Hackathon.
- 2021 **DAAD WISE Scholarship**.
- **MITACS Globalink Scholarship**.
- **Bronze**, Inter IIT Technology Meet (IIT Kharagpur contingent).
- 2019 **2nd Place**, Intelligent Ground Vehicle Competition (IGVC).
- 2018 All India Rank **1248** in JEE Advanced and **1618** in JEE Mains among 1M candidates.
- **Kishore Vaigyanik Protsahan Yojana (KVPY) Scholarship**.