

# Chia-Yi Su

574-303-5218 | csu3@nd.edu | <https://chiayisu.github.io/> | <https://github.com/chiayisu>

## EDUCATION

### University of Notre Dame

*Ph.D. in Computer Science and Engineering*

Proposal: Context-based Language Models for Source Code Summarization

Advisor: Collin McMillan

Notre Dame, IN

Aug 2022 – present

### National Kaohsiung University of Science and Technology

*M.S. in Electronic Engineering*

Thesis: Interactive Dialog System for Disease Information Retrieval

Advisor: Tsong-Yi Chen

Kaohsiung, Taiwan

August 2020 – June 2022

## PROFESSIONAL EXPERIENCE

### Research Assistant, Computer Science and Engineering

*University of Notre Dame*

August 2022 – present

Notre Dame, IN

- Led applied **machine learning** research in software engineering resulting in **10 research articles**
- Designing a **reward function** based on human-attention to improve various software engineering tasks
- Integrated **agentic AI** with **LangChain** for memory bug localization in C projects based on human behaviors
- Prompted and fine-tuned **LLMs (CodeLlama, GPT2, GPT4, Gemini)** with novel process for source code summarization that reduces **81% of the tokens** and shows **10% improvement in USE**
- Implemented a **dual encoder Transformer** with **Tensorflow** for context-aware source code summarization that results in **3% improvement in METEOR** and **1% improvement in USE** in Java dataset
- Implemented a **semantic loss function** based on **RLHF** by using **Pytorch/Tensorflow** to improve both **LLMs (LLaMA)** and **Transformer** for source code summarization
- Conducted empirical studies on **LLM reasoning** capabilities across code intelligence tasks identifying limitations in **LLMs** with **Gemini, GPT4o, CodeLlama, and GPT2** as examples

### Research Assistant, Electronic Engineering

*National Kaohsiung University of Science and Technology*

August 2019 – June 2022

Kaohsiung, Taiwan

- Enabled users to retrieve disease information with chat media and **natural language interaction**
- Trained BERT for sequence labeling and domain classification jointly by using **PyTorch** that achieves **92%** accuracy
- Implemented a supervised learning method to warm up dialog policy network for reinforcement learning models that reached **95% accuracy in dialog policy making** and **94% accuracy in token prediction**

### Software Engineer Intern

*Hewlett Packard Enterprise*

July 2019 – June 2020

Taipei, Taiwan

- Automated **100+ test cases** with Robot Framework, reducing test time from **10 days to 4 hours**
- Proposed and developed test automation functions to **lower the difficulty of test automation**
- Developed web to enable users to retrieve information front-end

## TECHNICAL SKILLS

**Programming Languages:** Python, C/C++, SQL, Java

**Machine learning:** Reinforcement learning, RLHF, LLM reasoning, Knowledge distillation, LLM post-training, Agentic AI, Transformer

**Libraries:** PyTorch, TensorFlow, Huggingface transformers, LangChain, NumPy, Pandas, Scikit-learn, Matplotlib, NLTK, Flask

**LLMs:** CodeLlama, LLaMA, StarCoder, nanoGPT, Gemini, GPT

**LLMOps:** QLoRA, LoRA, Instruct fine-tuning

**Others:** Web development, Docker, Static analysis, AWS, Git, Vim, Joern, Doxygen, Code intelligence, Software agents

## SELECTED PROJECTS

### A Reward Model based on Human Attention for Software Engineering Tasks

*University of Notre Dame*

September 2025 – present

Notre Dame, IN

- **Concepts:** LLMs, human attention, reward function, and LLM post-training, LLM alignment, model attention
- Designing **reward models** based on **human attention** for various software engineering tasks
- Aligning **model attention** with **human attention** for various software engineering tasks

<b>Do Code LLMs Do Static Analysis?</b> University of Notre Dame	June 2024 – May 2025 Notre Dame, IN
• <b>Concepts:</b> LLMOps, Prompt engineering, LLM reasoning, Static analysis, Code intelligence, and LLM post-training	
• Assessed <b>reasoning capabilities of LLMs</b> based on what human programmers do for code intelligence tasks	
<b>Context-aware Code Summary Generation</b> University of Notre Dame	December 2023 – June 2024 Notre Dame, IN
• <b>Concepts:</b> LLMOps, Prompt engineering, LLM reasoning, Static analysis, Code intelligence, and LLM post-training	
• Fine-tuned and prompted <b>LLMs ( GPT, Gemini, CodeLlama, and GPT-2)</b> with a novel process that <b>reduces 81% of tokens</b> and <b>shows 10% improvement over USE</b>	
<b>Distilled GPT for Source Code Summarization</b> University of Notre Dame	July 2023 – December 2023 Notre Dame, IN
• <b>Concepts:</b> Knowledge distillation, LLM post-training, and Prompt engineering	
• Distilled knowledge of source code summarization from <b>GPT3.5</b> with <b>GPT-2</b> and <b>StarCoder</b> to demonstrate <b>secure</b> and <b>efficient</b> open-source summarization alternatives	
<b>Semantic Similarity Loss for Neural Source Code Summarization</b> University of Notre Dame	January 2024 – July 2024 Notre Dame, IN
• <b>Concepts:</b> RLHF, Loss function, Optimization, Human ratings and feedback, and LLM post-training	
• Implemented a semantic similarity loss for <b>LLMs (LLaMA)</b> and <b>Transformer</b> with <b>PyTorch/TensorFlow</b> using <b>RLHF</b> to improve source code summarization	

## PUBLICATION

---

- [1] **C. Su**, A. Bansal, Y. Huang, T. Li, C. McMillan, "Context-aware Code Summary Generation," in Journal of Systems and Software (JSS), accepted, July 2025.
- [2] **C. Su**, A. Bansal, C. McMillan, "Revisiting File Context for Source Code Summarization", in Automated Software Engineering Journal (ASE Journal), Volume 31, article 62, 2024.
- [3] **C. Su** and C. McMillan, "Semantic Similarity Loss for Neural Source Code Summarization", in Journal of Software Evolution and Process (JSEP), 2024.
- [4] **C. Su** and C. McMillan, "Distilled GPT for Source Code Summarization", in Automated Software Engineering Journal (ASE Journal), Volume 31, article 22, 2024
- [5] **C. Su**, A. Bansal, V. Jain, S. Ghanavati, C. McMillan, "A Language Model of Java Methods with Train/Test Deduplication", in 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tool Demos (ESEC/FSE '23), San Francisco, California, USA, December 3-9, 2023.
- [6] A. Bansal, **C. Su**, Zachary Karas, Y. Zhang, Y. Huang, T. Li, C. McMillan, "Modeling Programmer Attention as Scanpath Prediction", in 38th IEEE/ACM International Conference on Automated Software Engineering, New Ideas and Emerging Results (ASE'23 NIER), September 11 - 15, 2023.
- [7] **C. Su** and C. McMillan, "Do Code LLMs Do Static Analysis?" (Under review at Empirical Software Engineering)
- [8] **C. Su** and C. McMillan, "CMind: An AI Agent for Localizing C Memory Bugs" (Under review at ICPC'26)
- [9] M. Dhakal, **C. Su**, R. Wallace, C. Fakhimi, A. Bansal, T. Li, Y. Huang, C. McMillan, "A Grounded Theory Study to Guide AI-Driven Code Comment Improvement" (Under review at JSEP)
- [10] **C. Su**, A. Bansal, V. Jain, S. Ghanavati, S. Peddinti, C. McMillan, "Which Code Statements Implement Privacy Behaviors in Android Applications?" (Preprint)

## PRESENTATION

---

- Conference presentation:** Distilled GPT for Source Code Summarization", in 39th IEEE/ACM International Conference on Automated Software Engineering. (Journal-first presentation)
- Conference presentation:** A Language Model of Java Methods with Train/Test Deduplication," in ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'23)
- Conference presentation:** Modeling Programmer Attention as Scanpath Prediction," in IEEE/ACM International Conference on Automated Software Engineering (ASE'23)

## SERVICE

---

- Student volunteer:** ASE'24
- Program committee:** ASE'25 NIER
- Reviewer:** ACM Transactions on Software Engineering and Methodology, Springer Automated Software Engineering Journal