ZHIYU NI

Email: zhiyuni@berkeley.edu Tel: (+1)323-633-1447

EDUCATION

University of California, Berkeley (UC Berkeley) *Berkeley, CA*
PhD student, *Computer Science*, Advisor: Pierluigi Nuzzo *2025.1 - Present*

University of Southern California (USC) *Los Angeles, CA*
PhD student, *Computer Engineering* *2022.8 - 2024.12*

University of Science and Technology of China (USTC) *Hefei, China*
B.S. *Physics*, Outstanding Graduates *2018.8 - 2022.6*

PUBLICATIONS

Analyzing Adversarial Vulnerabilities of Graph Lottery Tickets (ICASSP 2024 **Oral**)
Zhiyu Ni[*], Subhajit Dutta Chowdhury[*], Qingyuan Peng, Souvik Kundu, Pierluigi Nuzzo

Finding Adversarially Robust Graph Lottery Tickets (TMLR)
Zhiyu Ni[*], Subhajit Dutta Chowdhury[*], Qingyuan Peng, Souvik Kundu, Pierluigi Nuzzo

Accelerating Grokking via Embedding Transfer from a Weaker Model (ICLR)
Zhiwei Xu[*], **Zhiyu Ni**[*], Yixin Wang[◇], Wei Hu[◇]

RESEARCH EXPERIENCE

Adversarially Robust Graph Lottery Ticket (github)

- Analyzed the robustness of pruned graph neural networks (GNNs) against adversarial attacks.
- Developed self-training techniques and a loss function that improved sparse models' robustness, achieving state-of-the-art (SOTA) robust GNNs with a 90% reduction in computational cost.

LLMs for Anomaly Detection (github)

- Investigated capabilities of LLMs (e.g., ChatGPT, Llama) in anomaly detection and designed in-context learning flows.
- Achieved SOTA precision on GPT-3.5-Turbo compared with GNN-based methods.

Accelerating Grokking via Embedding Transfer from a Weaker Model (github)

- Developed a method to accelerate the grokking phenomenon in neural networks by transferring embeddings from smaller, weaker models to larger target models.
- Demonstrated much faster generalization on target models compared with sota methods.

WORK EXPERIENCE

NLP Intern (iFLYTEK) *2022.2 - 2022.6*

- Trained large-scale multi-layer transformers for machine translation.
- Achieved 15% higher BLEU score compared to Google Translate.

Technical Intern (Synopsys) *2025.5 - 2025.8(expected)*

- Fine-tune LLMs to improve the quality of RTL code generation.
- Accelerate the verification process of circuit design using RL.

SKILLS

Python, C, PyTorch, Linux