

# ANZE XIE

(608)-236-3958 | [a1xie@ucsd.edu](mailto:a1xie@ucsd.edu) | [AnzeXie.github.io](https://AnzeXie.github.io)

## EDUCATION

**University of California-San Diego**  
M.S. in Computer Science & Engineering

Sept. 2022 - Jun. 2024  
GPA: 3.9/4.0

**University of Wisconsin-Madison**  
B.S. in Computer Science & Statistics & Applied Mathematics

Sept. 2018 - Dec. 2021  
GPA: 3.99/4.00

## WORKING EXPERIENCE

- |   |                                     |
|---|-------------------------------------|
| <b>AI Engineering Intern</b>  | Institute of Foundation Models, USA |
| <b>Large-scale foundation model training infrastructure</b>   | May. 2025 - Aug. 2025               |
| • Conducted supervised finetuning based on Qwen2.5-72B using LLaMA-Factory  |                                     |
| • Optimized system parameters for efficient training of large-scale Mixture-of-Experts (MoE) models using Megatron-LM |                                     |
| • Assessed long-context training capabilities of Megatron-LM's context parallelism mechanism                          |                                     |
| <b>Pretraining data for coding capabilities</b>   | Aug. 2025 - Nov. 2025               |
| • Collected, deduplicated, and cleaned a large-scale dataset of coding problems for pretraining                       |                                     |
| • Built an efficient, scalable pipeline for generating synthetic coding problems using vLLM and Ray                   |                                     |
| • Generated and curated a dataset of more than 100 million high-quality synthetic coding problems                     |                                     |
| • Synthesized multi-turn traces of coding agents to enhance pretraining data richness                                 |                                     |

## RESEARCH EXPERIENCE

- |   |                       |
|---|-----------------------|
| <b>Research Assistant</b>   | UCSD, USA             |
| <i>Advisor: Prof. Hao Zhang</i>   | Mar. 2023 - May. 2025 |
| Area of Research: Long-context LLM training systems and LLM evaluation  |                       |
| <b>LongChat and LongEval</b>  |                       |
| • Curated a long-context dataset and proposed a finetuning method for extending LLM's context window  |                       |
| • Developed an evaluation toolset to assess LLM's long-context capabilities   |                       |
| • Finetuned LLaMA-7B and LLaMA-13B models and extended the context window by 8x   |                       |
| • Published on <a href="#">Instruction workshop at NeurIPS 2023</a>   |                       |
| <b>DistFlashAttn</b>  |                       |
| • Developed a state-of-the-art memory-efficient attention mechanism optimized for long-context LLM training   |                       |
| • Proposed a re-materialization-aware gradient checkpointing strategy   |                       |
| • Conducted ablation study on the efficiency of contemporary systems, including Megatron-LM, DeepSpeed Ulysses, FlashAttention, and Ring attention. |                       |
| • Enabled up to 8x longer sequences and achieved a 2x speedup in training   |                       |
| • Published on <a href="#">COLM 2024</a>  |                       |
| <b>GameArena</b>  |                       |
| • Developed an innovative platform that dynamically evaluates LLM reasoning capabilities through live computer games                                |                       |
| • Conducted retrospective analysis on LLM reasoning process to reveal specific LLM reasoning capabilities   |                       |
| • Published on <a href="#">ICLR 2025</a>  |                       |
| <b>Research Assistant</b>   | UW-Madison, USA       |
| <i>Advisor: Prof. Shivaram Venkataraman, Prof. Theodoros Rekatsinas</i>   | Feb. 2021 - May. 2022 |
| Area of Research: Systems for graph learning  |                       |
| <b>Data Mining Over Paleobiology Database</b>   |                       |
| • Extracted a knowledge graph from a paleobiology relational database with SQL and Python   |                       |
| • Trained graph embedding models over the knowledge graph and performed link prediction   |                       |
| • Provided insights on fact discovery for the team's paleobiologist   |                       |
| • Published in <a href="#">VLDB 2021</a>  |                       |

## PUBLICATIONS

---

Lanxiang Hu\*, Qiyu Li\*, **Anze Xie\***, Nan Jiang, Ion Stoica, Haojian Jin, and Hao Zhang. "GameArena: Evaluating LLM Reasoning through Live Computer Games." (ICLR 2025). Co-first authored.

Dacheng Li\*, Rulin Shao\*, **Anze Xie**, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. "DISTFLASHATTN: Distributed Memory-efficient Attention for Long-context LLMs Training." (COLM 2024).

Dacheng Li\*, Rulin Shao\*, **Anze Xie**, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. "How long can opensource llms truly promise on context length." (Instruction workshop @ NeurIPS 2023).

**Anze Xie**, Anders Carlsson, Jason Mohoney, Roger Waleffe, Shanan Peters, Theodoros Rekatsinas, and Shivaram Venkataraman. "Demo of marius: a system for large-scale graph embeddings." *Proceedings of the VLDB Endowment* 14, no. 12 (2021): 2759-2762.

## OPEN-SOURCE CONTRIBUTIONS

---

Contributed significantly to [LongChat and LongEval](#), a repository supports training and evaluation of long-context LLMs

Developed the data preprocessing, postprocessing, and rule-based configuration optimizer modules for [Marius](#) and [MariusGNN](#), a unified system for large-scale graph-learning tasks

## TEACHING EXPERIENCE

---

### Teaching Assistant

UCSD, USA

Scalable Data Systems

Jan. 2024 - Mar. 2024

ML Systems

Mar. 2024 - Jun. 2024

## AWARDS

---

UW-Madison Undergraduate Scholarship for Summer Study, 2020, 2021

UW-Madison Dean's List (7 semesters)

## SKILLS

---

**Languages:** Python, GO, C, C++, Java, SQL, Matlab, R, HTML/CSS, L<sup>A</sup>T<sub>E</sub>X

**Frameworks and Tools:** PyTorch, Tensorflow, Transformers, pytest, tox