Yichen Xu

Email: yichen.x@outlook.com
GitHub: github.com/linyxus
Personal Site: www.yichenxu.me

Education Background

Bachelor of Computer Science — Beijing, China
at Beijing University of Posts and Telecommunications — Sep 2018–Jun 2022, expected

1. *Overall GPA:* 3.85/4.0, 92.74/100 (ranking **1st**/379).
2. *Awards and honours:* National Scholarship (two consecutive years; only the top 1% students will be awarded the scholarship); Merit Student (two consecutive years); 1st Prize in NSCSCC 2020 (MIPS CPU Design Contest); 1st Prize in National Mathematics Contest.

Publications

[1] **Y. Xu**, A. Boruch-Gruszecki, and L. Parreaux, "Implementing Path-Dependent GADT Reasoning for Scala 3", in *Scala Symposium 2021*, 2021.

[2] **Y. Xu**, Y. Zhu, F. Yu, Q. Liu, and S. Wu, "Disentangled Self-Attentive Neural Networks for Click-Through Rate Prediction", CIKM '21, Nov. 2021.

[3] Y. Zhu, **Y. Xu**, H. Cui, C. Yang, Q. Liu, and S. Wu, "Structure-Aware Hard Negative Mining for Heterogeneous Graph Contrastive Learning", in *KDD Workshop on Deep Learning on Graphs: Method and Applications*, 2021.

[4] Y. Zhu, **Y. Xu**, Q. Liu, and S. Wu, "An Empirical Study of Graph Contrastive Learning", in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 34, Curran Associates, Inc., 2021.

[5] Y. Zhu, **Y. Xu**, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph Contrastive Learning with Adaptive Augmentation", in *Proceedings of The Web Conference 2021*, ser. WWW '21, Ljubljana, Slovenia: Association for Computing Machinery, Apr. 2021, ISBN: 9781450370233.

[6] Y. Zhu, **Y. Xu**, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep Graph Contrastive Representation Learning", in *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

[7] Y. Zhu, **Y. Xu**, F. Yu, S. Wu, and L. Wang, "CAGNN: Cluster-Aware Graph Neural Networks for Unsupervised Graph Representation Learning", *arXiv.org*, Sep. 2020. arXiv: 2009.01674v1 [cs.LG].

Research Experience

Research Intern — Lausanne, Switzerland
at Programming Methods Laboratory, Ecole Polytechnique Fédérale de Lausanne — Jan 2021–Current

- Improving Generalized Algebraic Datatypes (GADTs) in dotty, the Scala 3 compiler.
- Learned a lot about programming language theory, type theory and modern compiler design. Had much experience in participating on large-scale open source projects with collaboration toolchains.
- Fixed bugs in the compiler: fixing constraint derivation and type variable instantiation in the constraint solver.
- Improved the compiler's type system with respect to GADTs: deriving more constraints when upcasting, adding support for GADT reasoning on refined types, path-dependent types and singleton types.

Research Intern — Beijing, China
at Institute of Automation, Chinese Academy of Sciences — Jun 2019–Current

Major research interests lie in the fields of graph representation learning (with an emphasis on self supervised learning and contrastive learning).

- **Project 1: Cluster-Aware Graph Neural Networks.** Utilize clustering labels to train graph neural networks in a self-supervised manner. In this project, I surveyed related papers, developed the ideas, coded the model and wrote the final paper in collaboration with others. The paper is currently in submission to ACM TIST.
- **Project 2: Graph Contrastive Learning.** Generate graph views with graph augmentations and learn representations with contrastive objectives. I did surveying, developed and experimented ideas, and writing research papers in this project. Developed two papers as a co-first author, one of them accepted to GRL+ ICML Workshop, the other accepted to WWW 2021.
- **Project 3: CTR Prediction with Automatic Feature Interaction.** Implement CTR prediction using automatic feature interaction with a disentangled attention module. I wrote code and did experimenting in this project.

Selected Projects

fscala2c `linyxus/fscala2c`

- A compiler that compiles a subset of Scala to C. Basic functional programming and object-oriented programming functionalities and main language features are included in the subset. Implemented the Hindley-Milner type system for type checking and inferencing.
- Used Scala to develop the compiler. Learned about closure conversions to compile function values into C. Practiced my knowledge in type theory, compiler construction and functional programming.

Sircle `linyxus/CoordML`

- A functional and interpreted language, inspired by Haskell and Scala.
- Used Scala to develop its interpreter. Practiced my knowledge in programming language theory.

meow `linyxus/meow`

- A functional programming toolkit for Scala. It includes common algebraic structures, useful monads and some interesting tools and toys such as currying helpers and a typed tagless final language embedded in Scala.
- Used Scala to develop the library, and had a deeper understanding of functional programming, category theory and programming language theory during the progress.

MipsHike `linyxus/mips-hike`

- A virtual machine that implements a subset of MIPS assembly language. Also includes a toy compiler compiling a C-like language to MipsHike assembly.
- Used Haskell to develop both the virtual machine and the compiler. Practiced my knowledge in computer architecture and compiler construction.

EasterCache `easter-mips/EasterCache`

- A full-featured and high-performance MIPS cache, communicating with memory via AXI bus in wrap mode. It includes a victim cache and supports write buffering, with all parameters configurable.
- Used Chisel3 to implement it. Dived deeply into digital hardware design (especially for CPUs), FPGAs and computer architecture. Part of our team's CPU at a MIPS CPU design contest, NSCSCC 2020. Our team got **1st** prize in the contest.

Skills

- **Programming** Especially experienced in Haskell, Scala, Python, C++, Verilog and C; comfortable with JavaScript, Coq, Agda and Clojure. Experienced in programming under Unix-like environments (macOS and Linux). **An experienced Emacs user. Experienced and interested in Functional Programming.**
- **Researching** Acquired good research skills during my research internship. Experienced in academic reading & writing, surveying and experimenting.

- **Programming Language Theory and Formal Methods** Understand fundamental concepts in type theory. Understand basic theories in software verification (like formal logic, Hoare logic and operational semantics). Learned the first and the second volume of Software Foundations by myself (Logical Foundations and Programming Language Foundations).
- **Machine learning** Experienced in PyTorch, PyTorch Geometric, NNI, Scikit-Learn and NumPy; comfortable with Tensorflow. Have a good knowledge of theories and methods about machine learning on graphs, and understand common and important concepts in other domains as well.
- **Hardware Design and Computer Architecture** Experienced in FPGA programming with Verilog, SystemVerilog and Chisel3. Familiar with CPU design; experienced in cache design.

Other Information

- **English Proficiency:** TOEFL 109 (R 30, L 28, S 24, W 27)
- **Social Functions:** A member of Microsoft Student Technology Club, a Microsoft Learn Student Ambassador, a peer reviewer at ECML-PKDD 2020 and NeurIPS 2020.