

Gan Xu

3425 Motor Ave, Los Angeles, CA, 90034 | 984-888-6951 | gan.xu@icloud.com | ganxu.science | Github

EDUCATION

- MS, Computer Science, Washington University in St. Louis 09/2018- 05/2021
- BS, Computer Science & Mathematics, University of North Carolina at Chapel Hill 08/2015- 12/2017

TECHNICAL SKILLS

- **Programming languages:** Java/Kotlin, Python, TypeScript, C/C++, Shell, HTML, CSS
- **Framework and Tools:** Android, React, Git, Kafka, Docker, Protobuf
- **Related Coursework:** Algorithms, Data Structures, Databases, Operating System, Computer Networks, High Performance Computer System, System Security, Artificial Intelligence, Machine Learning, Bayesian Methods in Machine Learning etc.

WORK EXPERIENCE

Snap Inc. **Santa Monica, CA**
Software Engineer - Messaging Client *June 2021 - Now*

- Design, develop, modify applications and systems to Messaging and related functions on **Android** Snapchat app. Contributed to some highly impactful features including:
 - Chat-Reply, allow users to send a reply with quoted view of original message. The project involves message schema update, UI update and many cross-team collaboration on on-boarding various message types. The long requested by user feature now sees **200+ millions Daily Message Sent**.
 - Voice Note Revamp(**3rd most sent message type** on Snapchat worldwide), fully revamped UI with waveform, scrubbing to adjust progress, and variable playback speed supported. Also migrated audio player to **Exoplayer** from Android MediaPlayer. **9% note playback and 18% listen duration growth** after launch.
 - New Chat Page v2, revamp the page for selecting and creating 1-1/group chat/call, the new page is implemented with cross-platform framework and aligns the experience between iOS and Android. AB results brings more page visit(**+16%**) and more group creation and more calling/video chat (**+14.5%**).
 - Family Center, **lead** the android development of Snapchat's first **parental control** tool, which allows parents and guardians to keep tabs on who their teens message with on the app and report the suspicious users. (**N.Y.Times Report**: [link]) The project is still in rollout process but has received very positive feedbacks from public.
- **Cross-platform UI** development in **Typescript** with Snap's framework (comparable to **React Native**) on iOS and Android.
- Collaborate with back-end engineers to implement large scale **full-stack** projects, (e.g data sync, message schema...)
- Design and set up user **metric reporting** and **analyze, present** results from **AB studies** for **5+** major new features.
- **Cross-team** and **cross-function** collaboration with designers, program managers and data scientists on new **feature design**, implementation, integration and AB analysis.

Washington University **Saint Louis, MO**
Graduate Research Assistant *Feb 2019 - Nov 2020*

- Proposed methods and ran experiment to improve the communication performance over unreliable networks for **distributed AI algorithms**, including message partition and reconstruction, customized **Reliable UDP** protocol, forward error correction and etc. **50% run time reduction** over plain TCP for some extreme conditions.
- Collaborated with **Raytheon** BBN Technologies on DAPRA funded **distributed AI** project, details available upon approval.

PROJECTS

Distributed Agent Workflow Scheduling *May 2019 - Nov 2020*
Java, Maven, Kafka, Jenkins, SLF4J

- Mapped scheduling problems for agents to be solved by distributed constraint optimization(**DCOP**) framework.
- Built a **real-time messaging** system for distributed agents to coordinate with each other based on **Kafka**.
- Deployed maximum gain messaging(**MGM**) algorithm, an anytime algorithm allowing agents get results even if interrupted.
- Design and created APIs and schemas to integrate with other modules.

Multi-Room Chat Server(Web Application) **Github**
JavaScript, Node.js, HTML, CSS, MongoDB, Socket.IO *Jun 2020 - Aug 2020*

- Designed a real-time multi-room chat server using **Node.JS** and **Socket.IO**.
- Implemented both client-server and chat-server to realize the functions, saved chat history with **MongoDB**.
- Automated system deployment with **Docker**, and operated the online application on an **AWS EC2** Instance to improve the performance and make good management of the application.