

Filip Skogh

📍 Zürich, Switzerland

✉️ filipskogh99@hotmail.com

☎️ +41 76 269 4039



PROFILE

With 3 years of experience in C++, I'm enthusiastic about writing and optimizing code related to machine learning and security.

EDUCATION

ETH Zürich SEMP student, M.Sc. Computer Science

Zürich, Switzerland
Sep 2022 - Sep 2023

- **Thesis:** Weakly Supervised Video Object Segmentation
- **Activities:** ETH Analytics Club
- **Courses:** Natural Language Processing, Advanced Machine Learning, Computer Vision Project

Chalmers University of Technology M.Sc. Data Science and AI

Gothenburg, Sweden
Aug 2021 - Sep 2023

- **GPA:** 4.8/5.0
- **Courses:** Non-linear Optimization, Stochastic Processes and Bayesian Inference, Algorithms Computer Vision, Reinforcement Learning, Image Analysis, Large-scale Distributed Computation

Nanyang Technological University Exchange Student, B.Sc. Computer Science

Singapore
Jan-Jun 2020

- **Courses:** Digital Signal Processing, Cryptography, Operating Systems, Computer Networks

Luleå University of Technology B.Sc. Computer Science and Engineering

Luleå, Sweden
Aug 2018 - Jun 2021

- **GPA:** 5.0/5.0

EXPERIENCE

Machine Learning Intern IBM Research

Zürich, Switzerland
May-Current 2024

- Vision Transformer in Pytorch researching different types of normalization layers.
- Multi-modal foundation models for drug discovery.
- Open source contributions to drug discovery project GT4SD and huggingface spaces.

Machine Learning Engineer Logiblox AG

Zürich, Switzerland
Oct 2023 - Apr 2024

- Increased token generation speed by over 300% on LLM inference server by implementing quantization, parallelization and deciding on the optimal hardware.
- Set up from scratch a dockerized server with CI/CD pipeline running llama.cpp for LLM inference.
- Optimized (caching and operation merging) compiler that converts visual no-code representation in to executable python code.
- Set up a database and code-sandbox for an user facing LLM app that clean datasets.

Master Thesis Student Computer Vision Lab ETH Zürich

Zürich, Switzerland
Feb-Sep 2023

- **Supervisors:** Prof. Fisher Yu and Dr. Martin Danelljan
- **Objective:** Reduce the annotation burden for video object segmentation.
- **Solution:** Wrote algorithm in PyTorch to calculate spatial and temporal information between video frames to derive a consistency based loss.
- Achieved a 90% relative performance to the fully-supervised model without having access to the video mask annotations.

Optimization Research University of Massachusetts

Massachusetts, United States
Jun-Sep 2022

- Our paper received the Best Paper Runner-up award at the 14th IEEE IGSC 2023 conference in Toronto, Canada.
- Mixed-integer optimization in a distributed systems setting.
- Developed a server load scheduler that route requests to data-centers such that carbon is minimized while satisfying latency constraint.

Security Software Engineer Orange Cyberdefense

Stockholm, Sweden
Summers 2019 - 2021

- Developed automated threat response scripts in Python that block ransomware, C&C servers and take snapshots for forensics. Deployed globally on 50.000+ end points in 70+ countries.
- Offensive security: metasploit, kerberos security, email security.
- Network security, ssh and SMB server security, developed software to periodically scan network using nmap and masscan.

PROJECTS

Packet intercept proxy: C++ project developed continuously for three years by hooking Windows socket API `send` and `recv` using the Detours C++ library. Large-scale reverse engineering of project with 100k+ loc. Reverse engineered encryption protocols and ciphers to intercept traffic at packet level.

Packet fuzzer: Wrote a custom fuzzer in C++ to search op-code ranges and packet payload data to find server-side exploits for an MMORPG game.

Teaching Transformers arithmetic: Trained a GPT-style Transformer in PyTorch to learn addition with different tokenizers. Showed that if the digits are tokenized in reverse order the problem is easier.

Neural network certification: Developed custom network layers in PyTorch to propagate intervals through a network allowing us to deterministically prove properties about robustness, certain fairness guarantees and that adversarial attacks are not possible. This project was part of the course Reliable and Trustworthy AI at ETH.

Transformer inference: Implemented the decoder-only Transformer in PyTorch with a KV-cache for improved inference speed.

Blockchain implementation: Implemented parts of the Bitcoin protocol from scratch to create, (i) a wallet address derived from an elliptic curve public key, (ii) a signed transaction which can be broadcasted to the network, and (iii) a block verifier.

Google Developer Student Club: Built an interactive 3D learning game in Unity3D C# by building a hospital simulator. The project idea was conceived by medical professors prompted by the pandemic and was aimed to simulate medical students' practicum. During the project I worked in close contact with medical professionals and translated medical procedures into implementable scenarios in-game.

RANSAC: Iterative parameter estimation using RANSAC with optimal hypothesis testing that minimizes the number of tests performed. The Project was motivated by the scarcity of available implementation and was based on the original white paper.

TECHNICAL SKILLS

Languages:	Python, C++, Java, C, Matlab, English, Swedish
Frameworks:	PyTorch, AFL++, Hugging Face, Map-Reduce, PySpark, OpenCV, Flask, Java Native Interface, MySQL
Miscellaneous:	Vim, Slurm, FastAPI, Docker, Git, Regex