Nicolas Smits

nsmits@andrew.cmu.edu | linkedin.com/in/nicolas-smits | github.com/nicksmits1

Education

Carnegie Mellon University — Pittsburgh, PA
Doctor of Philosophy in Chemical Engineering — *Jan 2027 - Present*
Advisor: Carl D. Laird | **Research:** Optimisation, Process Systems Engineering, Stochastic Programming, ML, RL

Carnegie Mellon University — Pittsburgh, PA
Master of Science in Artificial Intelligence Engineering - Chemical Engineering — *Dec 2026*
GPA: 3.99/4.00 | **Teaching Assistant:** Systems and Toolchains for AI in Engineering (06-763), Fall 2026

McGill University — Montreal, Canada
Bachelor of Chemical Engineering, Minor in Applied Artificial Intelligence — *May 2025*
GPA: 3.73/4.00

Technical Skills

Programming Languages: Python, Java, MATLAB
Machine Learning: PyTorch, scikit-learn, TensorFlow/Keras, JAX, Hugging Face, Ray/RLlib
Data & Systems: Flask, PostgreSQL, SQL, Apache Spark, Kafka, Gunicorn, Neo4j, Hadoop, REST APIs, pytest,
Libraries: PyTorch, TensorFlow, Scikit-Learn, Keras, Gymnasium, Numpy, Pandas, Matplotlib, MosaicML, Seaborn, Ray, RLlib
Agentic AI Engineering: Agent harness design, tool calling, MCP, agent skills, CLI automation, agent orchestration
Cloud & MLOps: Docker (Compose), Git, GitHub Actions, MLflow, Weights & Biases, Prometheus, Grafana, AWS, GCP
Languages: English (Native), French (Fluent), Dutch (Fluent), German (Intermediate)

Experience

Center for Advanced Process Decision Making (CAPD) — Pittsburgh, PA
Graduate Research Assistant — *Jun 2026 - Aug 2026*

- Developed and deployed the open source Simultaneous Neural Differential Algebraic Equation (SiNDAE) Python package based on novel Hybrid Differential Algebraic Equations training methods developed at CAPD, allowing data-driven neural network surrogates to be embedded in process models and trained using Non-Linear Program (NLP) solvers.
- Wrote detailed technical documentation of package functionality and application examples spanning process control, bioprocess kinetics, and dynamical systems, deploying automated docs via Jupyter Book and CI/CD github workflows.
- Integrated and tested additional NLP solvers using Test-Driven-Development (TDD) and Agentic development workflows, shortening development time while ensuring codebase robustness, maintainability. AIChE 2026; JOSS pending.

HydrateTech Laboratory — Montreal, Canada
Computational Research Assistant — *May 2024 - Jul 2024*

- Implemented a Python tool and adapted GRADE hydrate cage identification algorithm with custom Depth-First Search (DFS), allowing partially formed molecular cage recognition, enabling entropy analysis for CO2 sequestration studies.
- Optimized geometric validation routines across diverse hydrate-host systems, achieving a 97% detection accuracy.

Lhoist Group — Nivelles, Belgium
R&D Engineer Intern (Flue Gas Treatment) — *Jun 2022 - Jul 2022 & Jun 2023 - Aug 2023*

- Developed a FGT kinetic process model using Python (scipy) for Lhoist's Sorbacal®sorbents for a scrubbing pilot reactor, capturing previously not modeled second order reaction dynamics and stoichiometric ratio vs conversion relationship.
- Improved upon simple legacy process model to include HCl, SO2, and CO2 secondary reactions, and temperature dependence, improving model robustness, enabling 50% MSE reduction, validated against pilot reactor data.
- Tuned stiff ODE solvers and built an Excel (openpyxl) front-end for a Python back-end, facilitating real-time simulations and rate constant estimation for engineers and experimentalists.

Projects

Graph Reinforcement Learning for Resilient Supply Chain Network Design | *Thesis* — *CMU | Jan 2026 - Present*

- Designed a graph RL agent (GATv2 encoder with Double DQN, PyTorch Geometric) that learns which transport links to add to a supply chain network so that customer demand is still met when plants, warehouses, or routes fail.
- Developed a size-independent Q-network that scores each candidate link from its endpoint embeddings, allowing a policy trained on small networks to design networks 4×larger without retraining.
- Built the multi-echelon inventory simulator, failure-scenario evaluation, and a parallel, resumable experiment pipeline on a 32-core server, with held-out test networks and statistical acceptance criteria fixed before any results were examined.
- First-author abstract accepted at ESCAPE 37 (Trondheim, 2027); codebase developed test-first with agentic workflows.

Production Movie Recommendation Service with LLM Cold Start *CMU | Sep 2026 - Present*

- Developed and deployed a containerized recommendation service (Flask, Gunicorn, Docker Compose) on a Linux AWS VM for a movie-streaming platform, serving 5.2M requests in its first monitored week, 99.98% within 500 ms.
- Designed an asynchronous LLM cold-start pipeline in which a Kafka consumer submits each new user's free-text preferences to TypeSafe's Jev model to extract a structured taste profile, before a vectorized NumPy ranker converts into recommendations in 1.4 ms, personalizing results before the median user's first request at an LLM cost of $1/day.
- Benchmarked content-based filtering, SVD collaborative filtering, and LightGBM LambdaMART on a chronological split, deploying LightGBM for a 2.2x NDCG@20 improvement over SVD at 139 ms P95 latency.

Multimodal RAG Pipeline for Grounded Document QA *CMU | Jan 2026 - Present*

- Built a multimodal RAG pipeline using a frozen ColQwen3 retriever for page-image retrieval and a Qwen3-VL generator fine-tuned with LoRA on teacher-distilled, evidence-conditioned supervision.
- Evaluated 3 configurations (vanilla VLM, VLM+RAG, VLM+RAG+LoRA fine-tuning) on the REAL-MM-RAG benchmark (8k pages, 5k queries) across answer quality (BERTScore, LLM-as-a-Judge), and groundedness metrics.
- Designed ablations on training signal quality and input modality (OCR text vs. page images) to isolate the effects of supervision and visual grounding on generation performance.

FIFA Player ETL Platform, Streaming Analytics & Cloud ML Pipeline *CMU | Sep 2025 - Dec 2025*

- Developed an end-to-end pipeline on the Sofifa Kaggle dataset (2015–2022) integrating PySpark ETL with PostgreSQL.
- Implemented ML models including Random Forest Regressor and Gradient Boosting Tree (SparkML), as well as both deep and shallow MLPs (PyTorch) to predict player overall rating from player skills, achieving $R^2 \geq 96\%$.
- Containerized services and deployed inference on Google Cloud Platform, connecting a containerized PostgreSQL instance on a cloud VM to a Dataproc Hadoop Cluster, ensuring reproducibility, version control, and modular pipeline structure.

Deep Reinforcement Learning for Sepsis Treatment in the ICU *McGill | Jan 2025 - Apr 2025*

- Implemented multiple DQN variants (Double, Dueling, Prioritized, Multi-Step, Rainbow) in the ICU-Sepsis-v2 environment, modeling patient treatment as an MDP to improve treatment decision making.
- Benchmarked DQN algorithms across 300k training episodes and 10 random seeds, revealing that Dueling DQN achieved the highest final return with improved convergence speeds over alternatives.
- Analyzed learning dynamics, showing that prioritized replay accelerated convergence by 100k episodes, and other DQN variants' sensitivity to hyperparameters

Energy-Efficient Day-Ahead Electricity Price Forecasting *McGill | Jan 2025 - Apr 2025*

- Developed (in a team of 4) SARIMA, RNN, LSTM, hybrid CNN-LSTM, and Transformer models trained on 4 years of ERCOT market and NOAA weather data to forecast day-ahead electricity prices, documented in NeurIPS-style report.
- Demonstrated tuned CNN-LSTM outperformed baselines SARIMA model by 2$\times$ in RMSE and MAE, capturing volatile price spikes and improving generalization capacity while using few input features.
- Applied compound scaling to CNN, reducing training energy consumption by 12%/epoch with accuracy maintained.