

Jiaqi Gan

Pittsburgh, PA | jig77@pitt.edu

Physics Ph.D. candidate in biophysics, with research experience in interpretable machine learning and small language model post-training. Develops models grounded in biophysical constraints to explain single-cell dynamics. Builds training and evaluation systems for AI call agents, spanning dataset development, LoRA fine-tuning, and behavioral benchmarking.

EDUCATION

University of Pittsburgh	Ph.D. Candidate in Physics, Biophysics Research	Expected Apr 2027
University of Pittsburgh	M.S. in Physics	2023
Shanghai University	B.S. in Applied Physics	2020

RESEARCH EXPERIENCE

Xing Lab | *Graduate Researcher, University of Pittsburgh* Aug 2022–Present

LogicDynamo | Learning and Explaining Cell-State Changes

- **Framework development:** Developed LogicDynamo in PyTorch to infer cellular dynamics from single-cell RNA and RNA-velocity data, implementing the model-fitting workflow that connects biological measurements to dynamical-system analysis.
- **Interpretable modeling:** Designed a generalized additive ODE model that separates individual regulatory effects from pairwise interactions, using saturating dose-response functions and sparse regulation to make the learned dynamics interpretable.
- **Validation:** Evaluated the framework across 2 simulated dynamical systems and 5+ real single-cell datasets; implemented ensemble uncertainty estimation and computational gene perturbations to assess noise sensitivity and produce regulator-effect reports for experimental follow-up.

GraphVelo | Multimodal Cell Dynamics · [Code](#) · [Nature Communications, 2025](#)

- **Software development:** Co-developed GraphVelo, an open-source Python package for estimating the direction and speed of cell-state changes; implemented local tangent-space projection to refine noisy RNA-velocity inputs and support mapping between RNA and chromatin-accessibility measurements.
- **Benchmarking:** Performed ground-truth tests across 3 simulated trajectory types (linear, cyclic, and branching), evaluating direction, magnitude, and sign with cosine similarity, RMSE, and sign accuracy; demonstrated retention of speed information lost by cosine-kernel projection.
- **Sensitivity analysis:** Tested robustness to noise and dimensionality reduction, comparing velocity estimates with known simulated dynamics; found better speed agreement in PCA than UMAP, identifying limits on quantitative interpretation of cell-state visualizations.

INDUSTRY EXPERIENCE

ScamAI | *ML Research Intern* Jul 2026–Present

Small Language Model Post-Training for AI Call Agents

- **Data generation:** Built a pipeline that produced 3,395 synthetic calls and 15,102 training targets. Enforced prefix-only reply inputs and separate dialogue and summary reviews.
- **Post-training:** Fine-tuned six 0.6-4B Qwen3, Llama, and Gemma models with one rank-16 LoRA adapter per model for replies and summaries. Balanced both tasks in each update and applied loss only to target tokens.
- **Evaluation:** Evaluated 4,608 outputs on 192 held-out calls with a custom five-criterion LLM judge. Improved reply qualification by 43-74 percentage points and summary qualification by 43-68 points across all six models.

CallScreenBench and Real-Call Risk Prediction · [Preprint](#)

- **Benchmark design:** Built CallScreenBench end to end to evaluate call screening and owner notes across 22 call archetypes; implemented stateful caller simulations, 5 quality-measure groups, and 3 guardedness diagnostics, evaluating 6 locally run language models.
- **Metric validation:** Stress-tested evaluation metrics with 6 scripted policies and estimated uncertainty with paired bootstrap intervals; exposed a scoring loophole when a hangup-and-copy agent achieved 100% caller-detail recall without meaningful dialogue, demonstrating the need for counter-metrics.
- **Predictive modeling:** Analyzed 10,211 real inbound calls and 330,956 transcribed turns to predict later sensitive-information requests; a bag-of-words baseline reached 0.87 ROC-AUC from the first 8 caller lines on 1,276 calls held out by phone number. [Preprint](#)

TECHNICAL SKILLS

Languages: Python, C++, R, MATLAB | **Systems:** Git, Linux, Slurm, CUDA

ML Frameworks: PyTorch, Hugging Face Transformers, MLX, scikit-learn