

# Julian CHAN

✉ [julianchan8897@gmail.com](mailto:julianchan8897@gmail.com) | ☎ +447767766646 | 🌐 [github.com/julian-8897](https://github.com/julian-8897) | 🔗 [linkedin.com/in/julian-chan-a86b2921b](https://linkedin.com/in/julian-chan-a86b2921b)

## SUMMARY

Scientific ML research engineer and final-year PhD researcher developing GPU simulation software, differentiable models and interpretable dynamical equations. Experience spans JAX/PyTorch training, inverse problems and numerical validation.

## RESEARCH AND ENGINEERING PROJECTS

### Erebus: GPU-accelerated N-body Simulation Software 2026

- Developing a Python/CuPy N-body code that couples **fast multipole gravity** to chain-regularised black-hole subsystems through Bridge integration, with optional post-Newtonian dynamics through 3.5PN.
- Integrated GPU tree construction and traversal kernels using AI-assisted development.
- Added direct CPU-reference comparisons, conservation and restart regressions, HDF5 checkpointing and configurable SLURM workflows for simulation campaigns.

### Neural Surrogates and Symbolic Regression 2024 – 2026

*PhD research, University of Surrey*

- Developed parameter-conditioned Neural ODE surrogates for N-body dynamics, achieving approximately **1% median fractional error on held-out trajectories**. Integrated predictions into a semi-analytical model, giving merger times within 1–24% of direct simulations. [First-author paper accepted to MNRAS](#).
- Combined a pretrained neural surrogate with dimensionally constrained PhySO symbolic regression to infer compact hardening and eccentricity-growth equations for black-hole binaries. Evaluated integrated evolution on simulations excluded from neural training, symbolic fitting and coefficient calibration. *First-author manuscript submitted*.

### Differentiable Simulation and Inverse Methods (Turbulence) (Inverse methods) 2026

*Collaboration with Pasteur Labs; Honourable Mention, Tesseract Hackathon 2025.*

- Trained a PyTorch CNN closure through a JAX/Exponax spectral solver in-process, reducing 500-step rollout vorticity MSE by **71.7% versus the solver alone** and 15.7% versus a matched offline-trained CNN, with lower error on all 32 test seeds at every evaluated horizon. Separately demonstrated training through two Tesseract services using VJPs.
- Recovered Burgers viscosity through one JAX/Optax loop using either the solver adjoint (about 1–2% error) or interchangeable JAX/PyTorch PINNs behind a shared gradient interface.
- Trained a flow-matching posterior over viscosity and initial-condition parameters; checked calibration with simulation-based calibration and TARP coverage. Packaged the components as versioned containers with checkpointing and CI.

### Research Brief Agent (GitHub) 2026

- Built a streaming FastAPI research agent using Qdrant/SPECTER2 retrieval, full-text papers and web evidence to produce cited technical briefs. Added citation checks and scenario evaluations in CI, bounded tool use and Langfuse tracing.

## SELECTED PUBLICATION

### Prior or Feedback? What an LLM Uses When Adapting Neural Operators

*Accepted to the NeurIPS AI4Science Workshop.*

## SKILLS

**ML / SciML:** Python, PyTorch, JAX, differentiable programming

**Computing:** CuPy, SLURM/HPC, Docker

## EDUCATION

**PhD Astrophysics & Machine Learning**, University of Surrey 2023 – Present

**MSc Computer Science** *Distinction*, Swansea University 2021 – 2022

**MSci Physics with Particle Physics & Cosmology**, University of Birmingham 2016 – 2020