Bridging the Stellar Mass Gap: N-body Simulations of Intermediate Mass Black Hole Formation in Globular Clusters

Abstract

Intermediate-mass black holes (IMBHs; $10^{2-5} M_\odot$) bridge the gulf between stellar-mass and super-massive black holes yet their birth channel is hotly debated. We will run star-by-star N-body simulations of 10^{4-6} stars in globular-cluster potentials, coupling GPU-accelerated NBODY6++GPU with 2.5-PN terms to track relativistic inspirals. Varying cluster density, initial black-hole retention and metallicity, we follow 10 Gyr of dynamical evolution and record hierarchical mergers that climb past the pair-instability mass gap. The project will deliver merger trees, IMBH occupation fractions and synthetic observables, providing first-principles predictions for upcoming LISA and ELT surveys.

Project Description

Scientific motivation

Recent observations hint at $\sim 10^3 M_\odot$ Black Holes (BHs) in 47 Tucanae (Gültekin. 2017; Kızıltan et al. 2017)and M15 (Huang et al. 2025). Direct N-body and Monte-Carlo studies show two promising growth paths: 1) Run-away stellar collisions in young, dense cores (Fujii et al. 2024); 2) Hierarchical mergers of retained stellar BHs (Torniamenti et al. 2024). Yet past runs either stopped at $M_{\rm BH} < 500 M_\odot$ because of gravitational-wave recoil or used over-simplified dynamics. Our proposal pushes beyond both limits with **post-Newtonian-accurate, million-timestep integrations** on modern GPUs.

Key questions

1. Can GC dynamics alone build IMBHs above the recoil barrier ($\gtrsim 10^3 M_\odot$)?

2. Which cluster birth conditions (mass, half-mass radius, metallicity, BH retention) maximise IMBH yield?
3. What present-day observables (tip of velocity-dispersion, ejected runaway stars) tag clusters that host IMBHs?

Answering these will resolve the mystery mass gap of BHs.

Method

There are two the-state-of-art N-body simulation codes available for this project. **NBODY6++GPU (Beijing branch)**—already well-developed and production-tested on supercomputers and including spin-dependent recoil kicks and 2.5-Post-Newtonian (PN) correction (Wang et al. 2016), available at Github. An alternative option is **PeTar**, a higher effiency code than NBODY6++GPU with up to 10^7 particles support and identical PN support (Wang et al. 2020), available at Github.

To explore realistic GC environments and initial conditions as much as possible, we provide the parameter gird as below:

PARAMETER	BASELINE	RANGE EXPLORED	RATIONALE
N_*	10^5	10^{4-6}	Typical massive GC masses
Density profile	?	?	Matches Milky-Way GCs
Half-mass radius	2.5 pc	1.5–6 pc	Captures observed spread
Metallicity Z	0.0002	0.0002–0.02	Sets BH natal masses
Initial BHs retention	50 %	10–80 %	Tracks fallback & kicks

Here we assume a Chabrier (2003) initial mass function (IMF). We evolve stars self-consistently via the Binary-Star Evolution (BSE) module embedded in the code, producing BH remnants whose distribution naturally reflects metallicity-dependent winds. Close encounters use the AR-CHAIN integrator with PN corrections; gravitational-wave captures and recoil kicks are modelled following recent prescriptions (Preto et al. 2008; Wang et al. 2016; Wang et al. 2020). Every merger stores masses, spins and kick velocity for constructing full merger trees.

A single 2×10^5-star run needs $\approx 4{\times}10^{14}$ **floating-point operations**. On $4 \times$ NVIDIA A100 GPUs this equates to $\sim$ **10 wall-clock days** (benchmarks from Wang et al. 2016 test suite) for 10 Gyr evolution. We request **720 runs** (parameter grid above): $\approx$ **70 million GPU-core-hours**, comfortably within a typical 2 MSU allocation.

Expected outcome

- **IMBH occupation fraction** vs. cluster mass and metallicity.
- **Merger-tree catalogue** for LISA rate forecasts.
- **Synthetic kinematic maps** and ejected-star velocity distributions to link with JWST and ELT.

All data products will be deposited in DataCentral, enabling community cross-checks and electromagnetic follow-ups.

Risks mitigation

We have two optional code to run.

Reference

Gültekin. 2017: https://www.nature.com/articles/542175a

Kızıltan et al. 2017: https://www.nature.com/articles/nature21361

Huang et al. 2025: https://academic.oup.com/nsr/article/12/2/nwae347/7810597

Fujii et al. 2024: https://www.science.org/doi/10.1126/science.adi4211

Torniamenti et al. 2024: https://www.aanda.org/articles/aa/full_html/2024/08/aa49272-24/aa49272-24.html

Wang et al. 2016: https://academic.oup.com/mnras/article/450/4/4070/990854

Wang et al. 2020: https://academic.oup.com/mnras/article/497/1/536/5867779

Chabrier. 2003: https://iopscience.iop.org/article/10.1086/376392

Preto et al. 2008: https://iopscience.iop.org/article/10.1088/1742-6596/154/1/012049