From Microspheres to Supermassive Stars

An overview of the University of Idaho's Numerical Relativity group's research

---

Leonardo R. Werneck

Thiago Assumpção, David Boyer, Terrence Pierre-Jacques
Samuel Cupp, Samuel Tootle, Nishita Jadoo
Zachariah B. Etienne

In collaboration with:
P. Chang (UWM), B. Kelly (UMBC/GSFC), S. Gossan (Hofstra), J. Baker (GSFC), J. Wise (Georgia Tech)

Funding acknowledgement
NASA: 80NSSC24K0360
NSF: OAC-2227105, PHY-2110352, AST-{2108072,2108269,2227080}

2024 Idaho National Laboratory
Idaho Falls, ID, USA

Outline

- ➡ Part I — Tiny Levitating Spheres
  - Sphere diameter: $\sim 70\,\mu\mathrm{m} = 7 \times 10^{-5}\,\mathrm{m}$
  - A new experiment for measuring the gravitational constant *G*
  - Key software: RETINAS

- ➡ Part II — Colliding Spheres
  - Sphere diameter: $\sim 24\,\mathrm{km} = 2.4 \times 10^{3}\,\mathrm{m}$
  - Binary neutron star mergers
  - Key software: Einstein Toolkit, GRHayL, NRPy+

- ➡ Part III — Large Collapsing Spheres
  - Sphere diameter: $\sim 1\,\mathrm{Mkm} = 1 \times 10^{9}\,\mathrm{m}$
  - Gravitational collapse of supermassive stars
  - Key software: Einstein Toolkit, ENZO Project, MESA

- ➡ Part IV — Key Software Overview

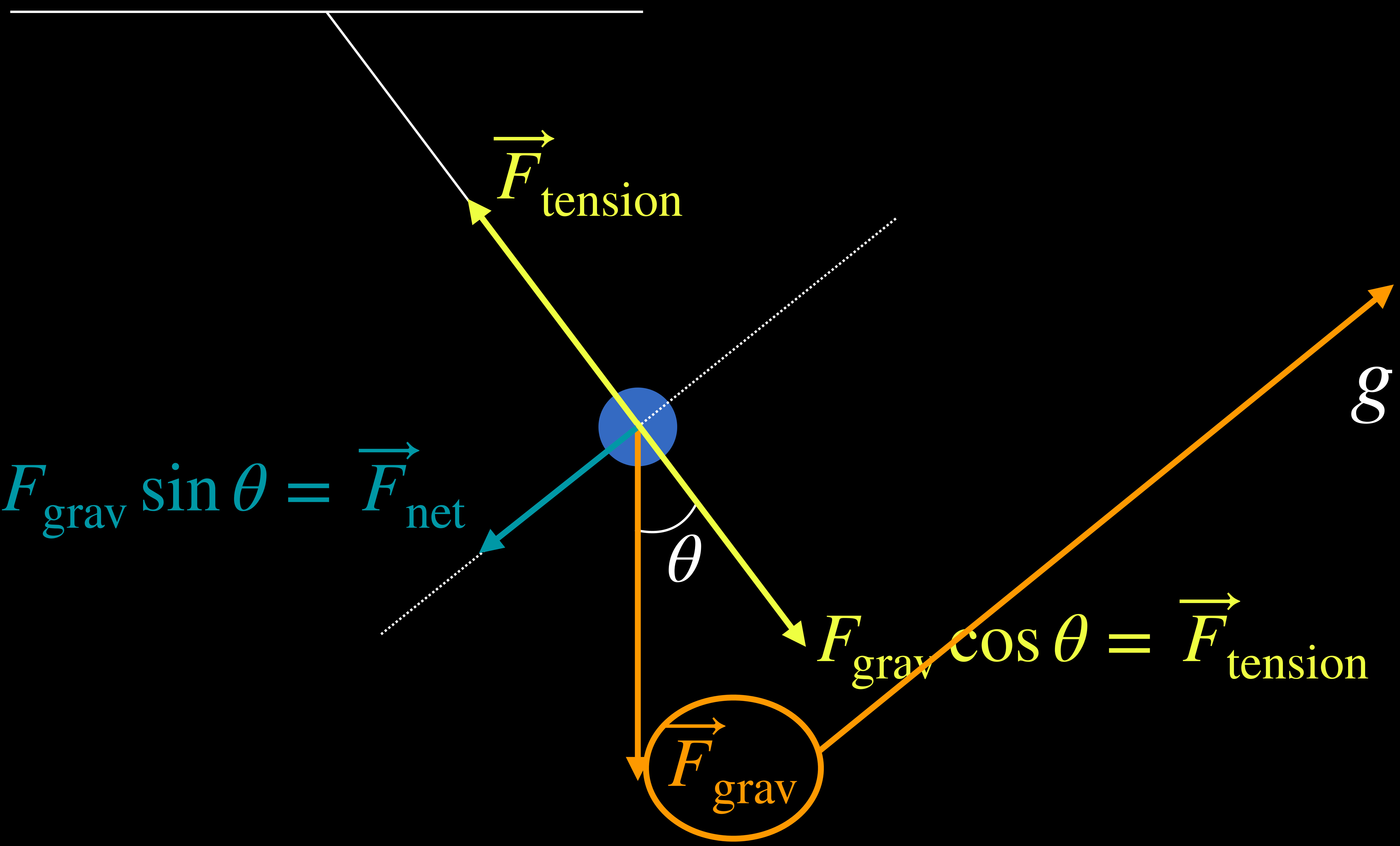
Part I — Tiny Levitating Spheres

The Gravitational Force

$$\vec{F}_{\text{net}} = m\vec{a}$$

$$\vec{F}_{\text{grav}} = m\vec{g}$$

$$g = 9.8\,\text{m/s}^2 = 32\,\text{ft/s}^2$$



$\vec{F}_{\text{tension}}$

$F_{\text{grav}} \sin\theta = \vec{F}_{\text{net}}$

θ

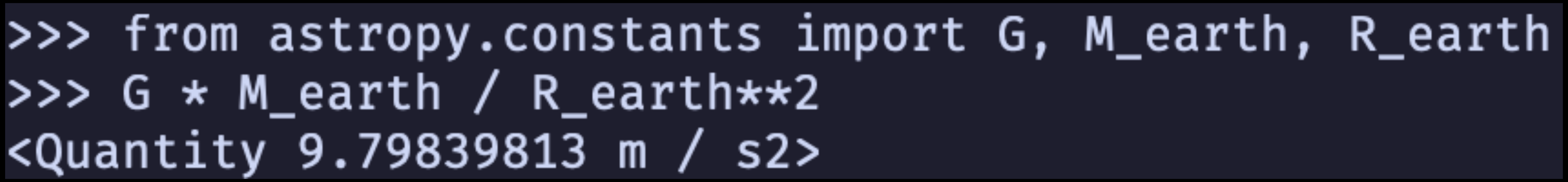
$F_{\text{grav}} \cos\theta = \vec{F}_{\text{tension}}$

$\vec{F}_{\text{grav}}$

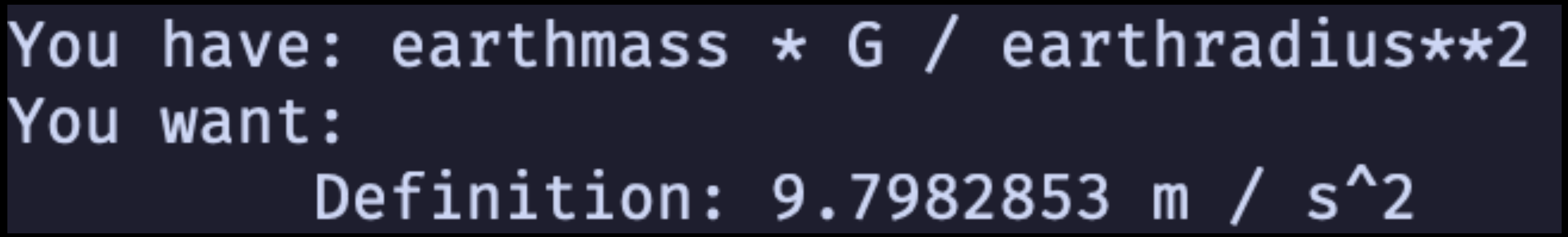
Where does *G* come from?

Where does $g = 9.8\,\mathrm{m/s^2} = 32\,\mathrm{ft/s^2}$ come from?

$$F_{\text{grav}} = \frac{GMm}{r^2} \approx \boxed{\frac{GM_{\text{Earth}}}{r_{\text{Earth}}^2}}\,m$$



```
>>> from astropy.constants import G, M_earth, R_earth
>>> G * M_earth / R_earth**2
<Quantity 9.79839813 m / s2>
```

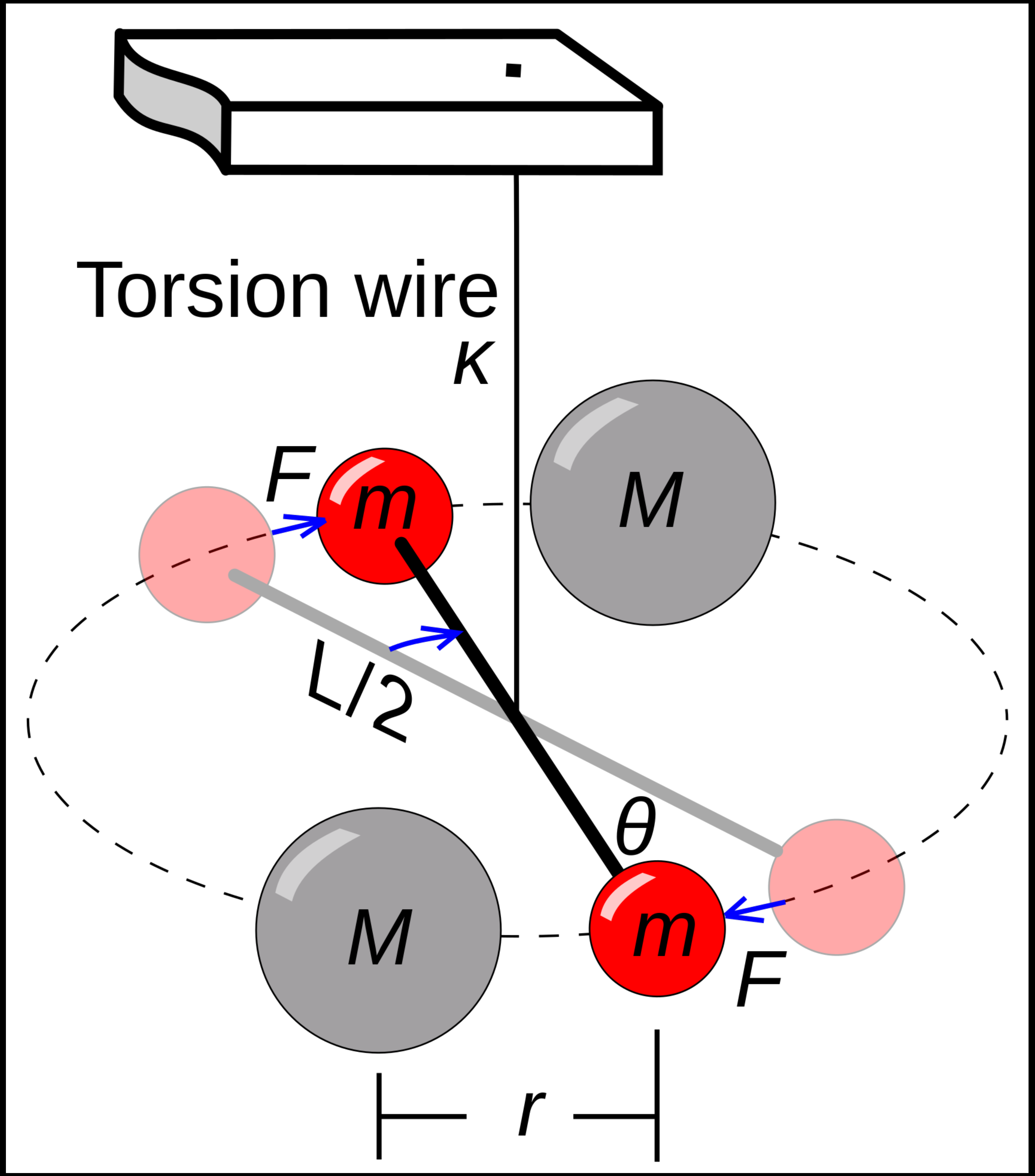


```
You have: earthmass * G / earthradius**2
You want:
        Definition: 9.7982853 m / s^2
```

But where does the value of *G* come from?

Measuring *G*

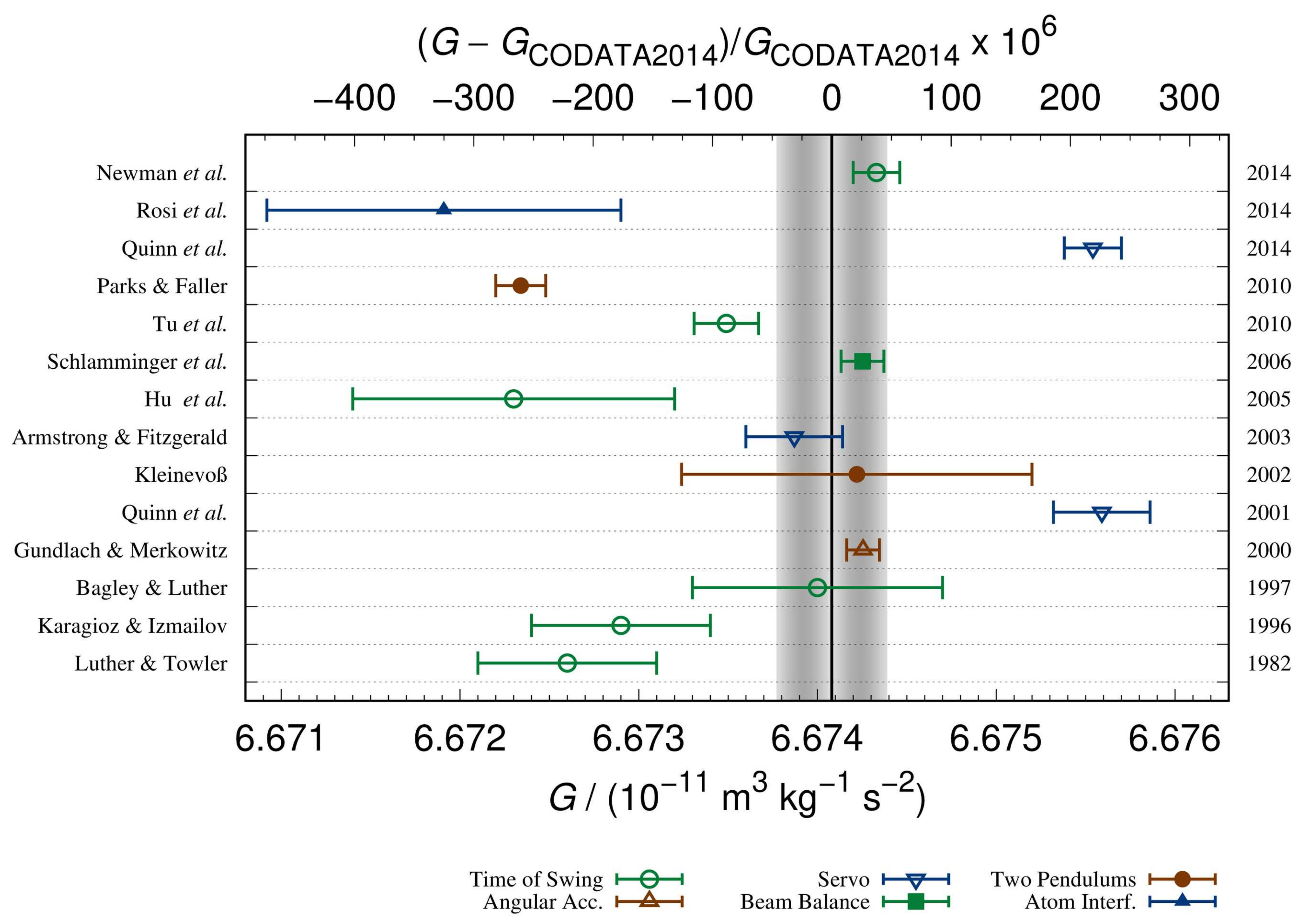
The Cavendish Experiment



Credit: Wikipedia

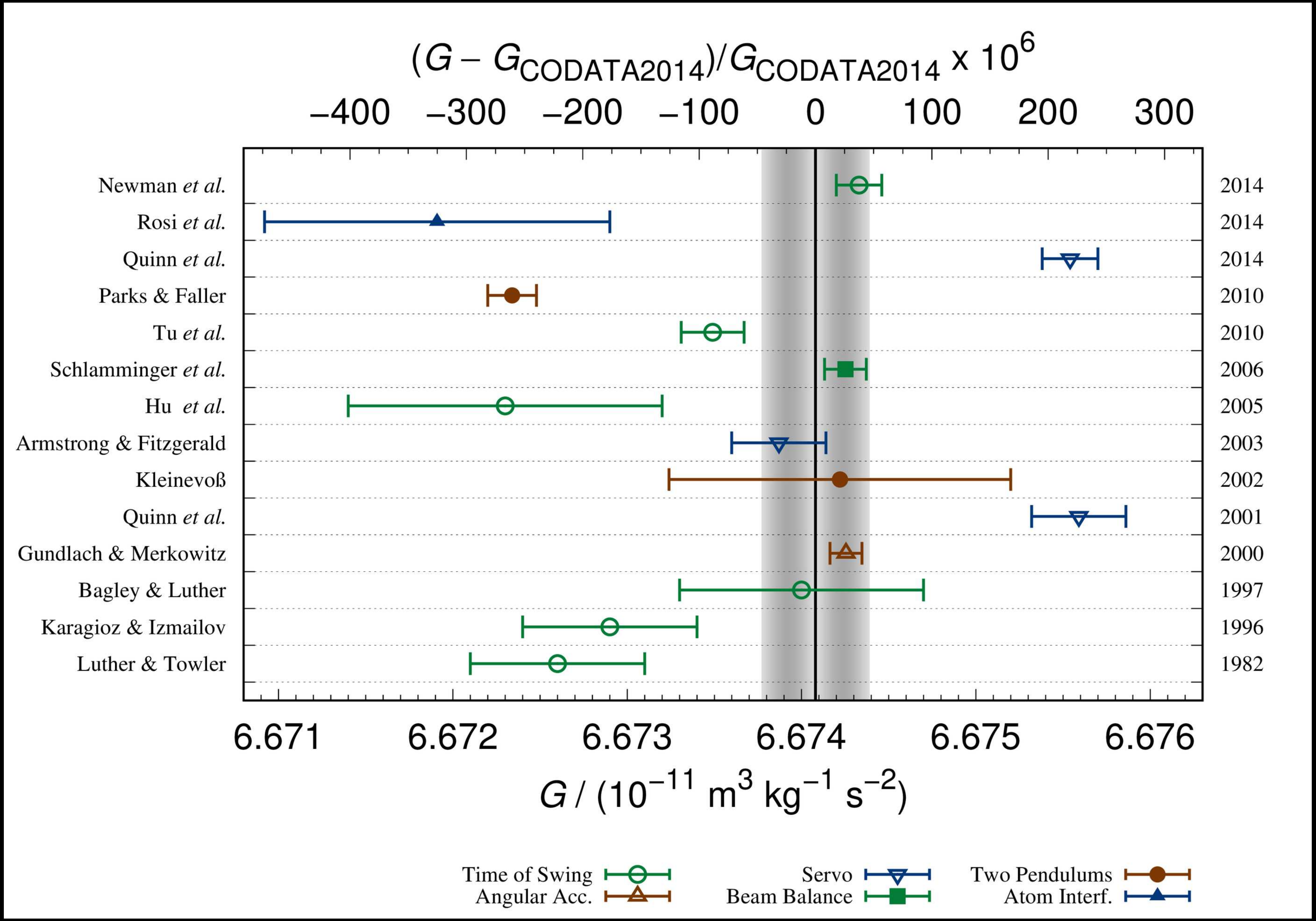
- Henry Cavendish, in 1978.
- Based on John Michell’s apparatus.
- Results within 1% of current values.
- Many modern experiments use variations of this method.

Measuring *G*



Credit: National Institute of Standards and Technology (NIST)

Measuring *G*



Credit: National Institute of Standards and Technology (NIST)

Gravity is Weak

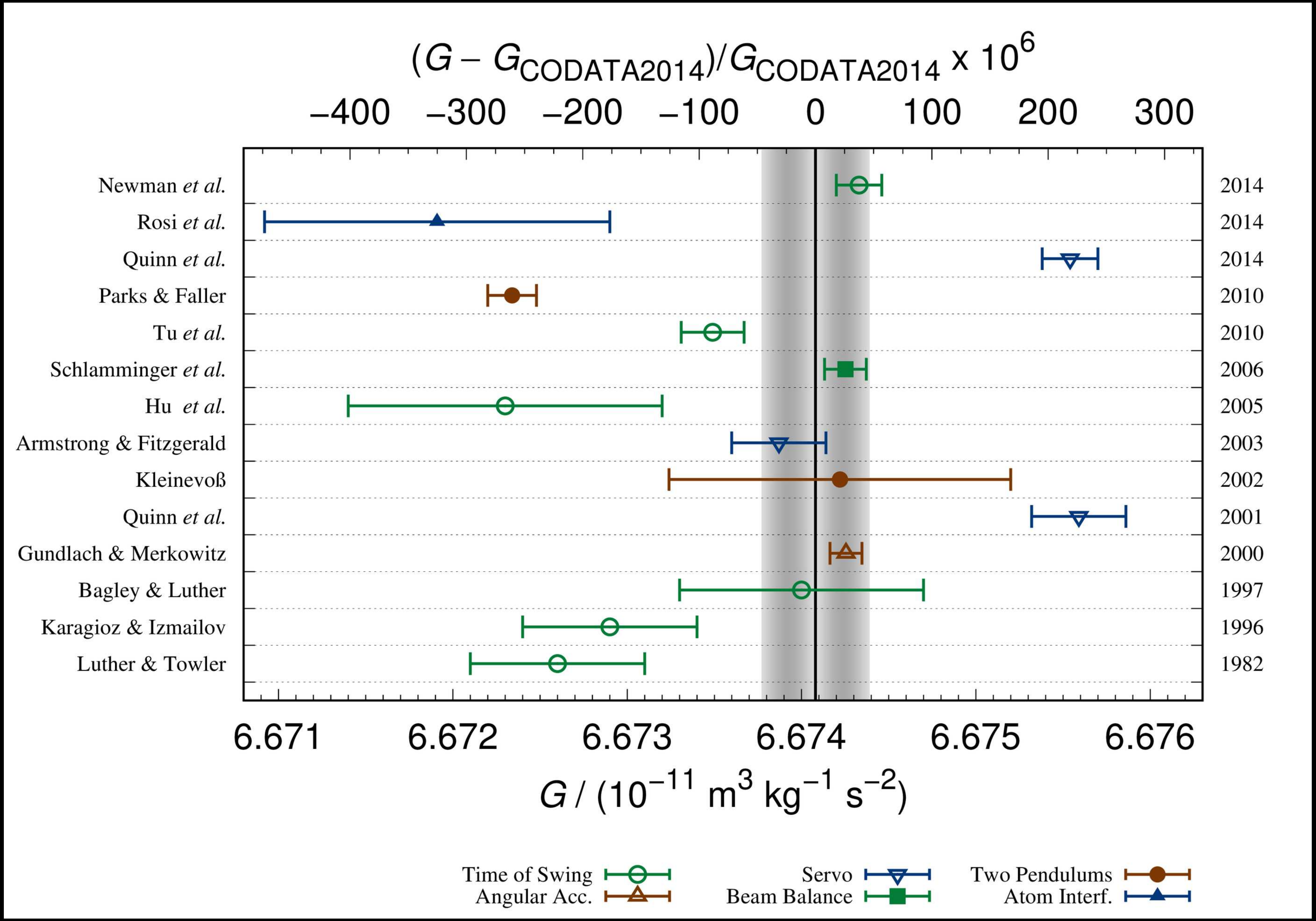
$$F_{\text{grav}} = G\frac{Mm}{r^2}$$

$$G \approx 6.67 \times 10^{-11}\,\mathrm{N\,m^2/kg^2}$$

$$F_{\text{Coulomb}} = k_C\frac{q_1 q_2}{r^2}$$

$$k_C \approx 8.99 \times 10^{9}\,\mathrm{N\,m^2/C^2}$$

Measuring *G*

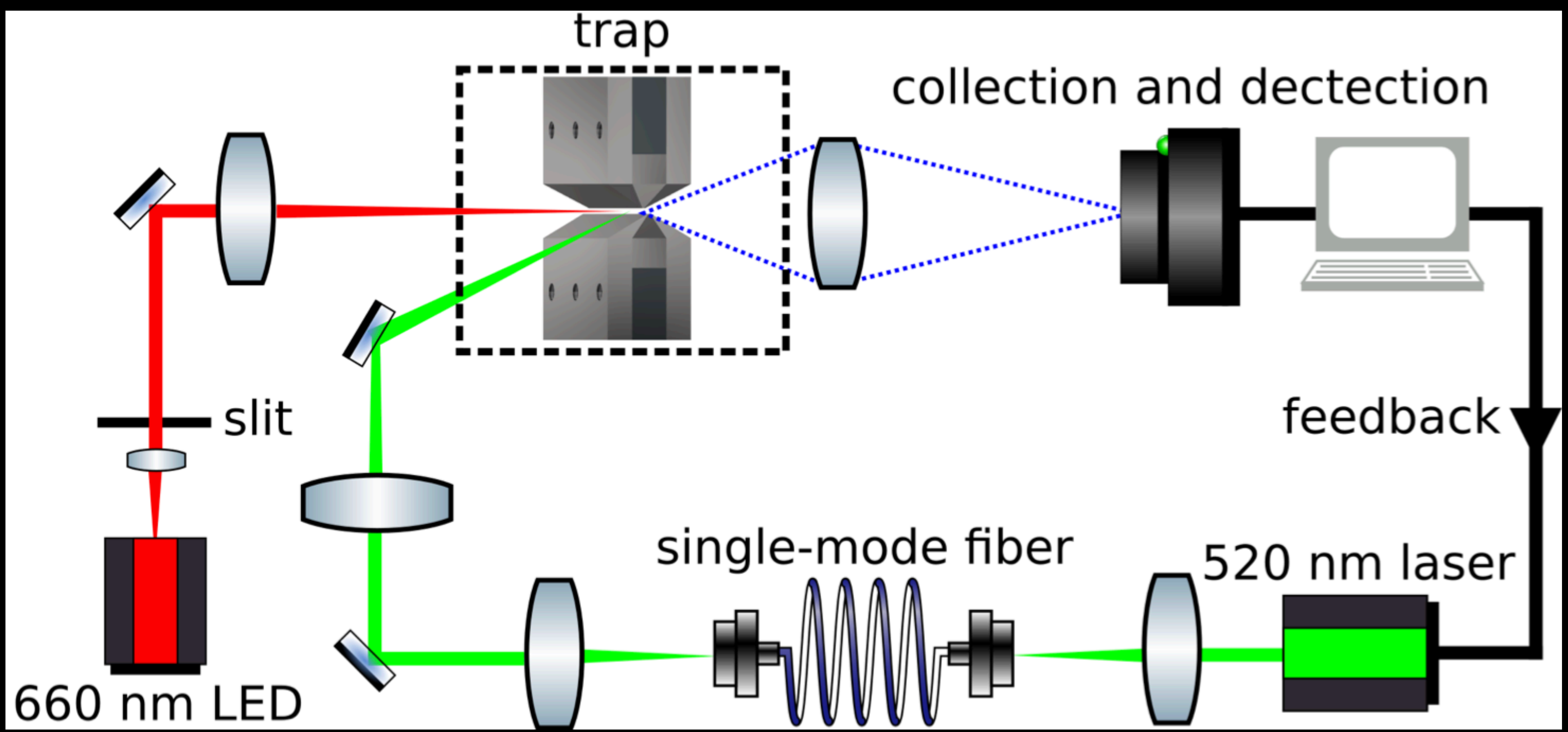


Credit: National Institute of Standards and Technology (NIST)

Systematic Errors

- Difficult to characterize.
- Most likely explanation.

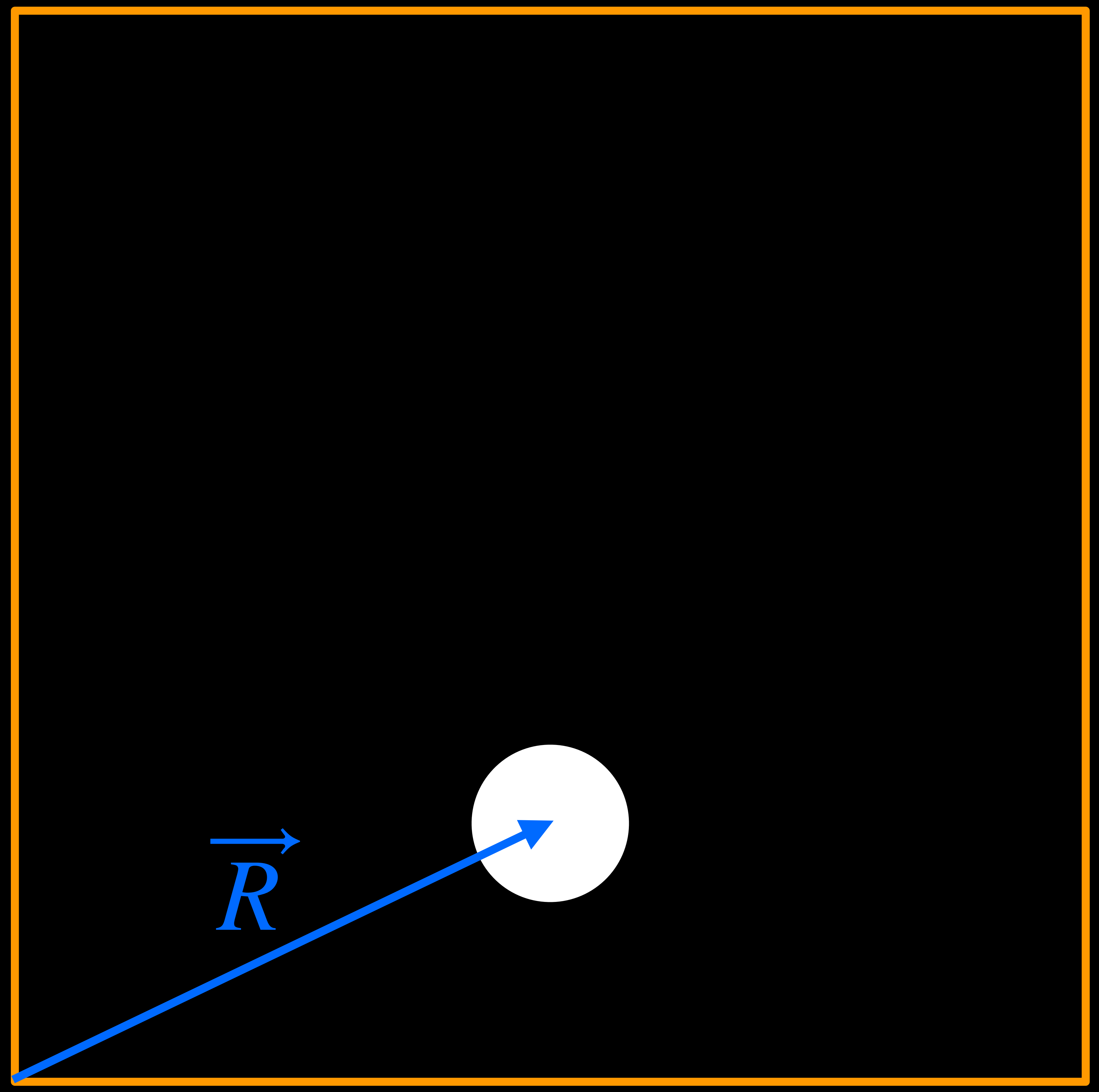
Measuring *G*



Lewandowski *et al.*, Phys. Rev. Applied **15**, 014050 (2021)

- Magneto-gravitational trap.
- Microspheres of diameter ~60 μm.
- High-sensitive, room temperature accelerometer.
- Uses feedback control to damp and cool the motion of the particle.

Particle Tracking



“Center of Mass” Tracking

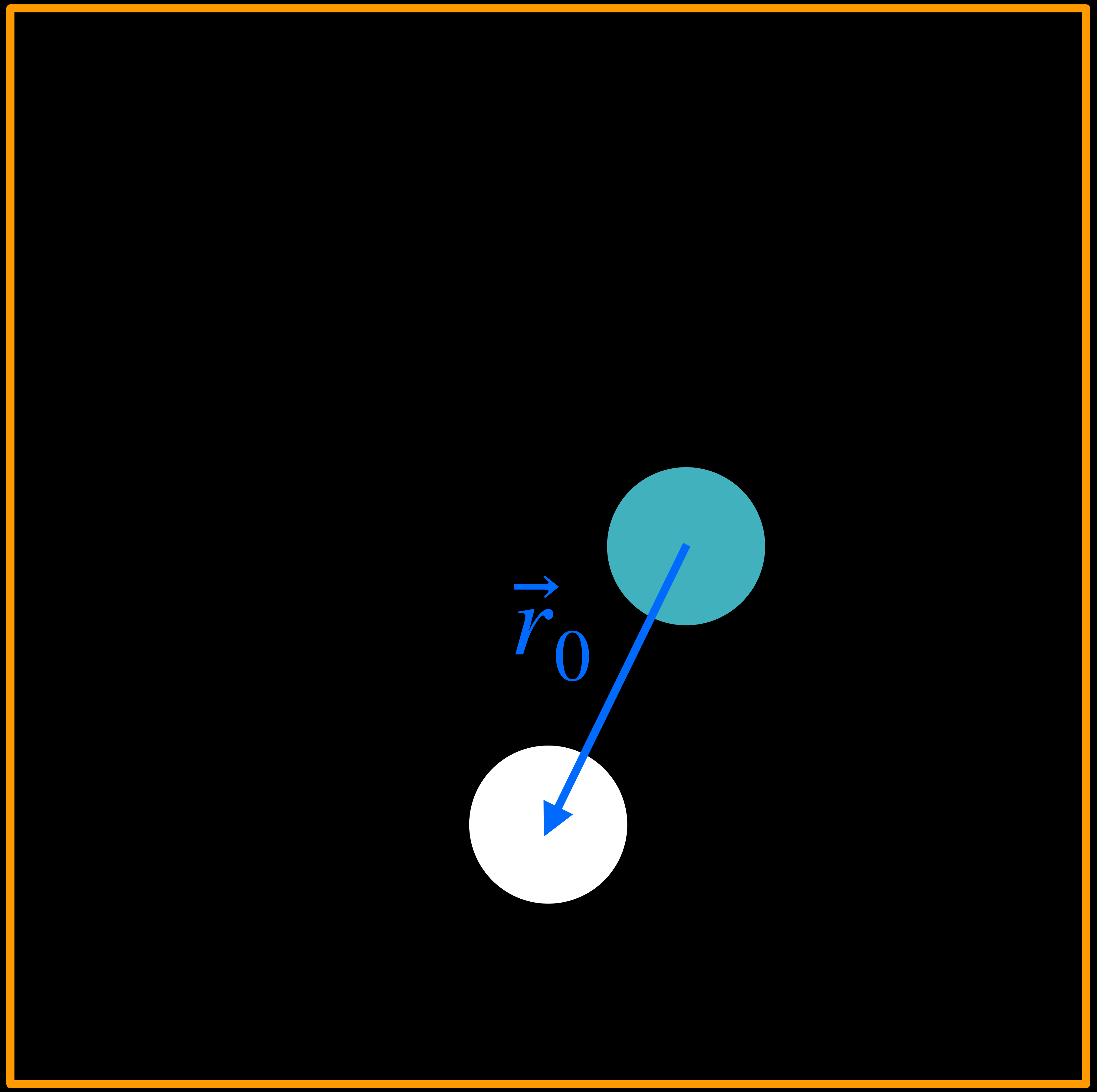
$$\vec{R} = \frac{\sum_{\vec{r}} \vec{r} I(\vec{r})}{\sum_{\vec{r}} I(\vec{r})}$$

✓ Computationally inexpensive.

✓ Easy to implement.

✗ Accuracy greatly affected by:
- Image boundaries
- Noise

Particle Tracking

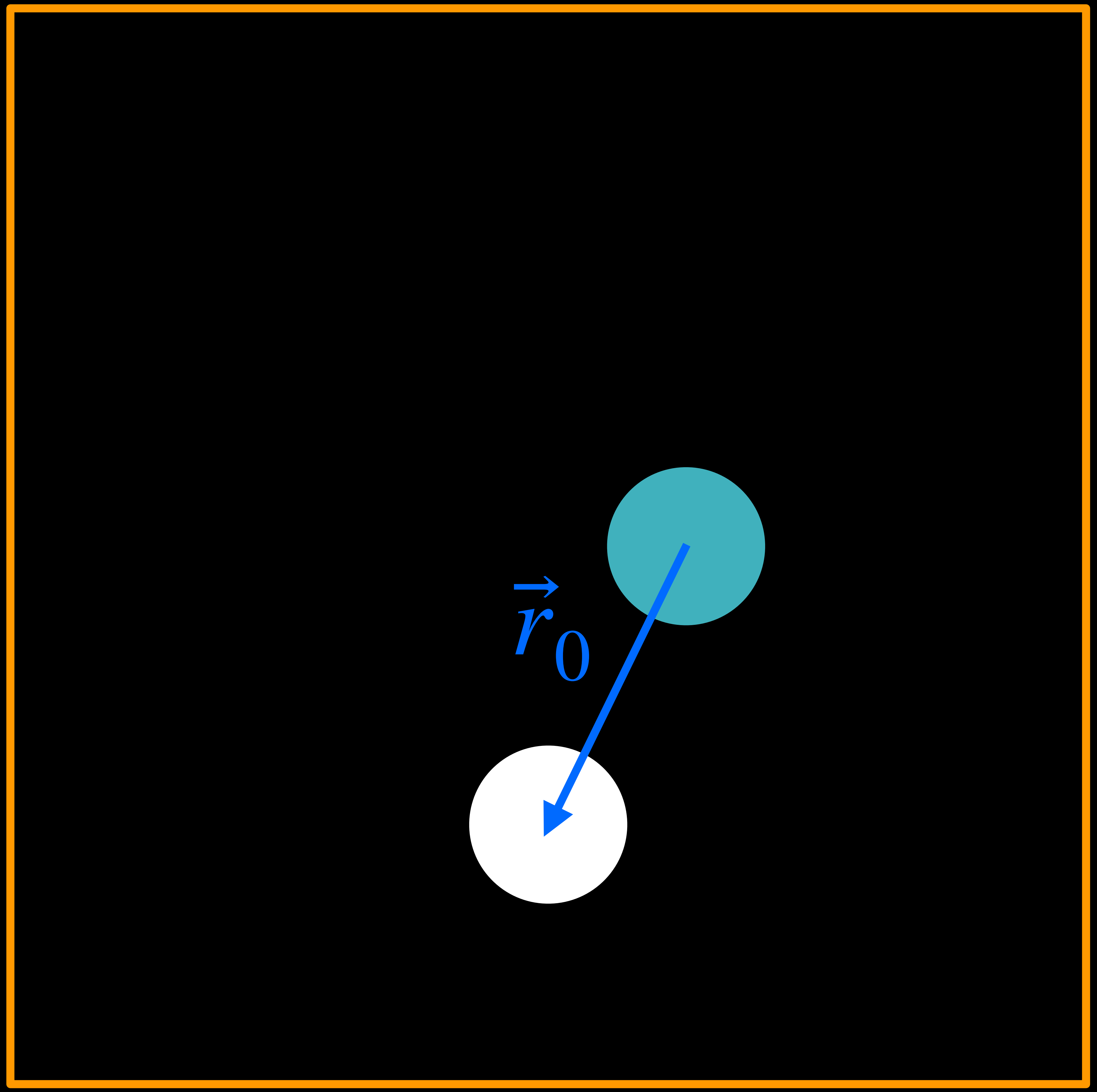


Maximum Likelihood Estimation

$$\chi^2 = \sum_{\vec{r}} \left[\frac{I(\vec{r} - \vec{r}_0) - E(\vec{r})}{\sigma(\vec{r}, \vec{r}_0)} \right]^2$$

- ✓ Implementations readily available.
- ✓ Maximum accuracy.
- ✗ Requires accurate particle model.
- ✗ Computationally intensive.

Particle Tracking



Uniformly-Weighted Cross-Correlation

$$\sigma(\vec{r},\vec{r}_0) = \sigma \implies \chi^2 \sim \sum_{\vec{r}} I(\vec{r}-\vec{r_0})E(\vec{r})$$

$$\sum_{\vec{r}} I(\vec{r}-\vec{r_0})E(\vec{r}) = \mathscr{F}^{-1}\left[\mathscr{F}[I] \otimes \overline{\mathscr{F}[E]}\right]$$

✓ Some implementations available.

✓ Can yield sub-pixel precision.

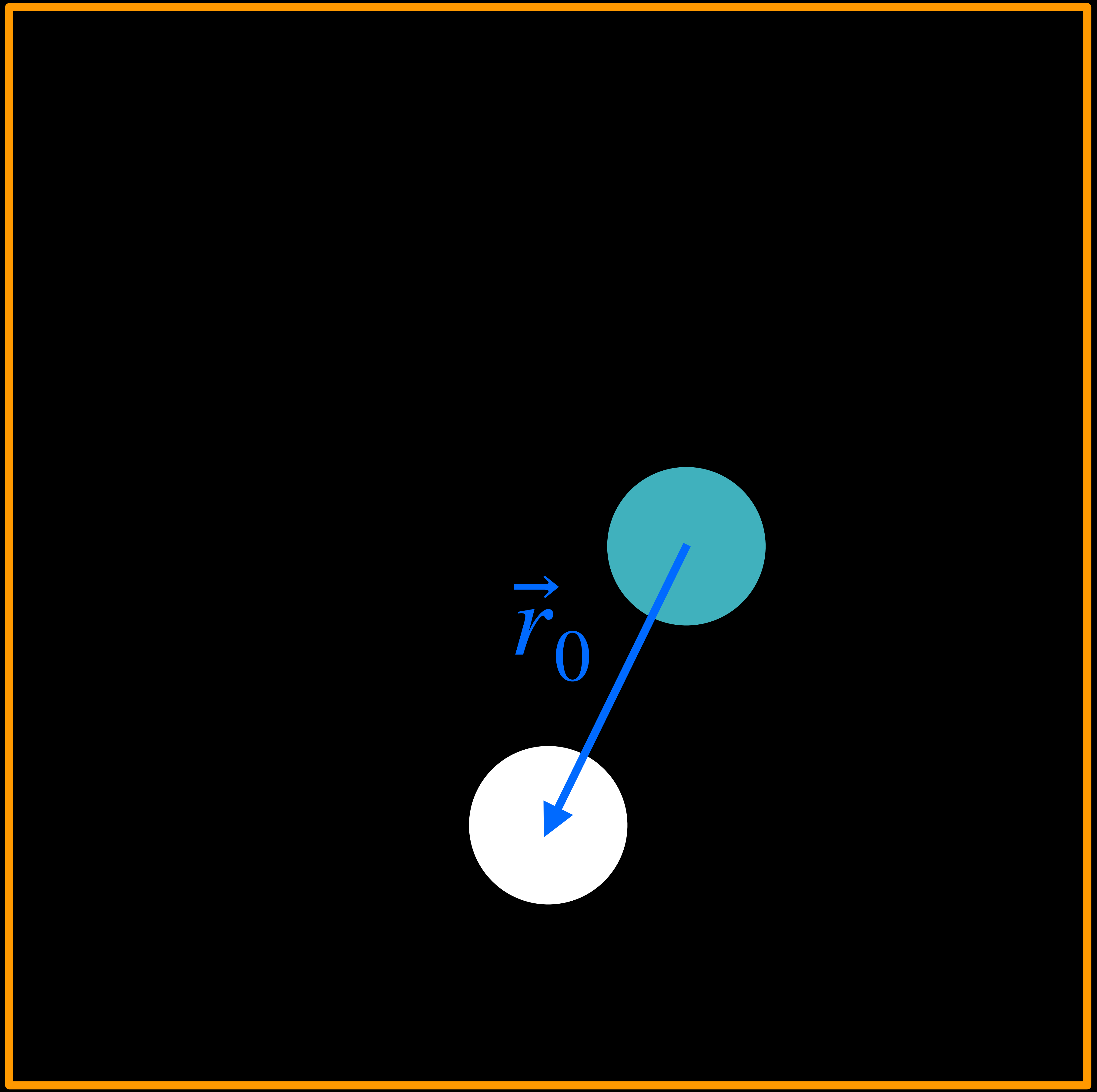
✓ Does not requires accurate particle model.

!? Less computationally intensive than MLE.

✗ Poorly models the noise in the images.

★ Lewandowski *et al.*, PRApplied **15**, 014050 (2021)

Particle Tracking



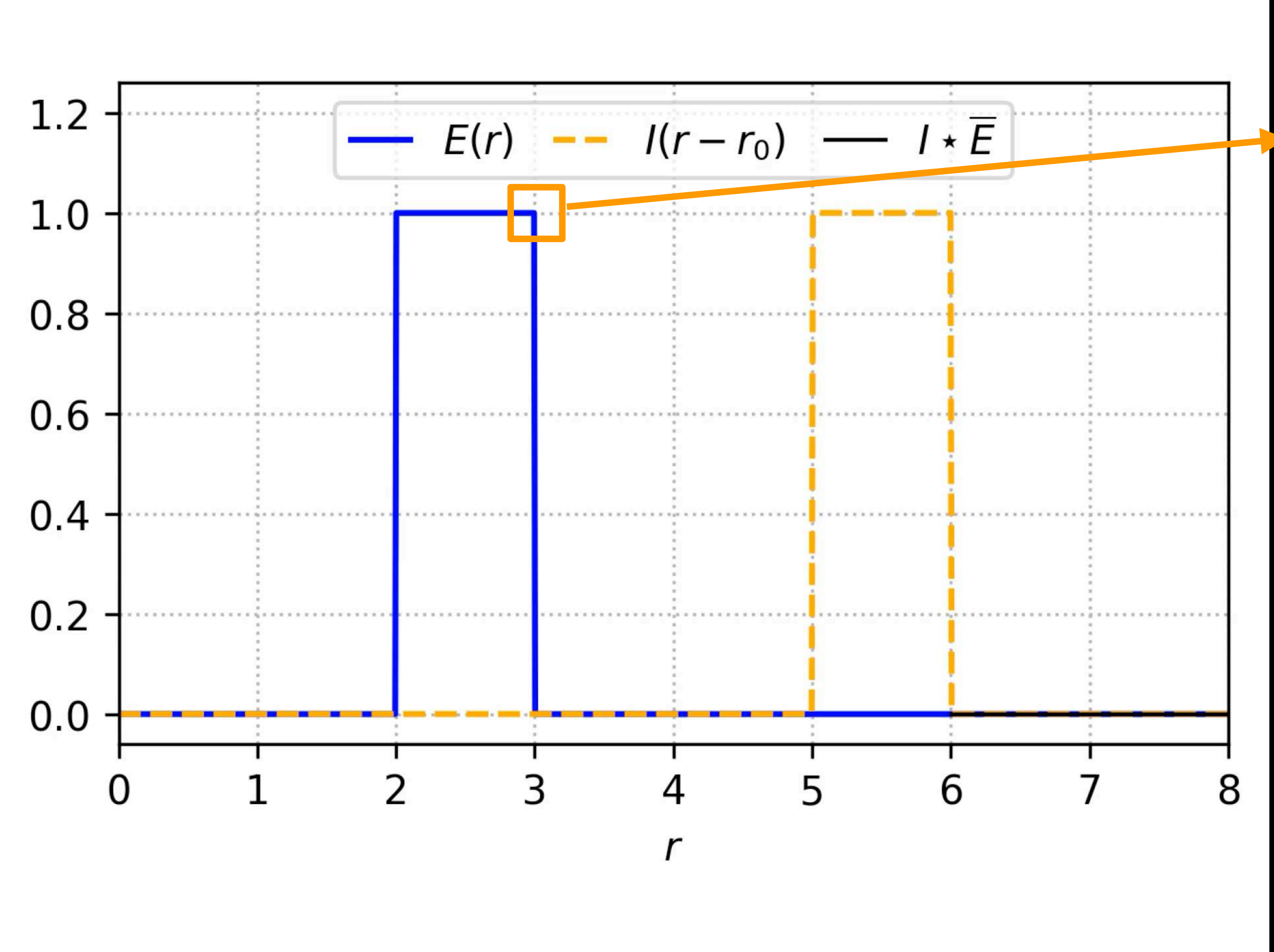
Shot-Noise-Weighted Cross-Correlation

$$\sigma(\vec{r},\vec{r}_0) = \sqrt{E(\vec{r})} \implies \chi^2 \sim \sum_{\vec{r}} \frac{\left[I(\vec{r}-\vec{r}_0)\right]^2}{E(\vec{r})}$$

$$\sum_{\vec{r}} \frac{\left[I(\vec{r}-\vec{r}_0)\right]^2}{E(\vec{r})} = \mathscr{F}^{-1}\left[\mathscr{F}[I^2] \otimes \overline{\mathscr{F}[1/E]}\right]$$

- ✓ No implementations available.
- ✓ Can yield sub-pixel precision.
- ✓ Does not requires accurate particle model.
- !? Less computationally intensive than MLE.
- ✓ Adequately models the noise in the images.
- ★ LRW++, submitted to RSI (2024)

Sub-Pixel Tracking: The Upsampling Algorithm



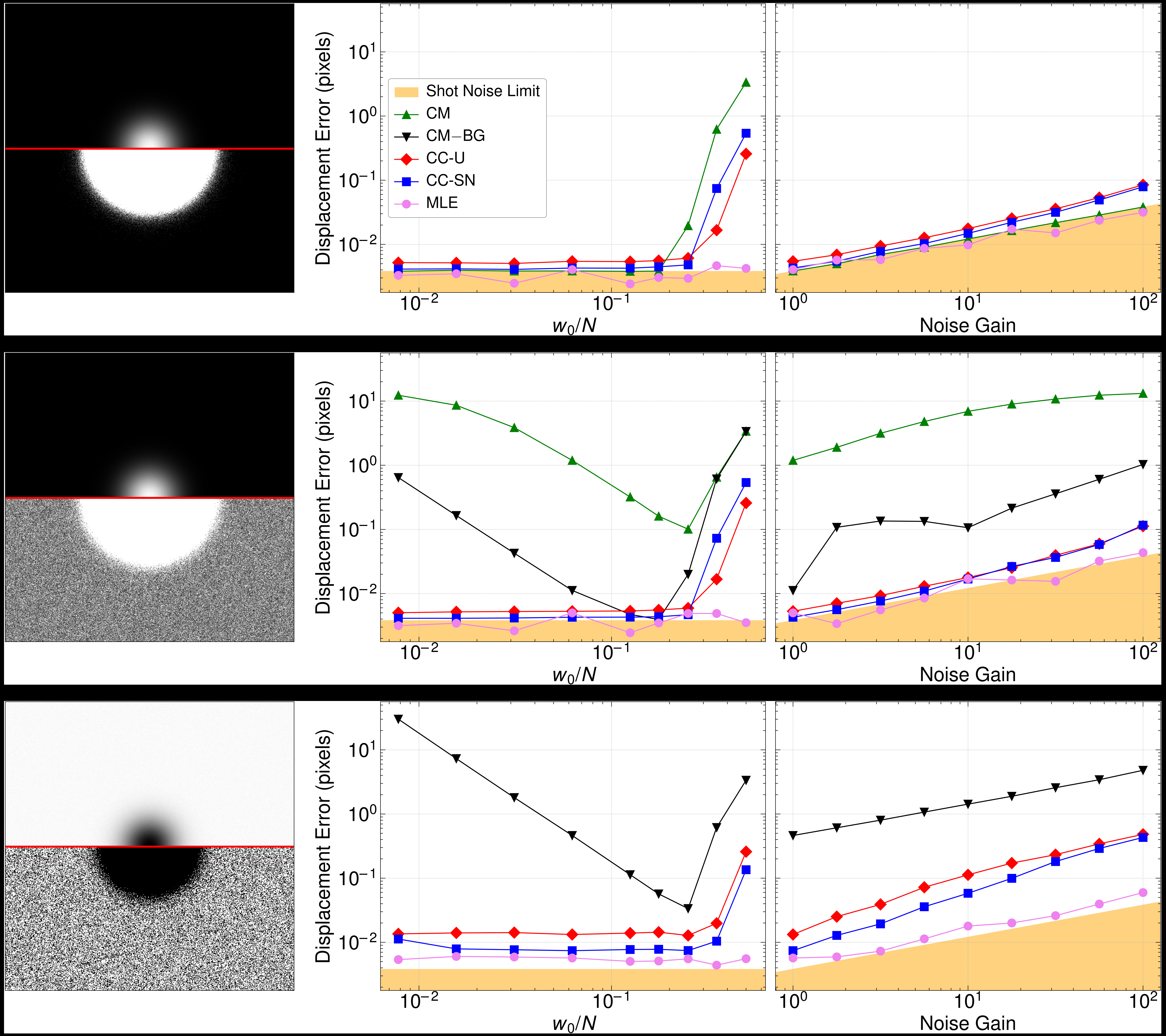
1) 1.5-pixel square around maximum.

2) Resolution: $u \times u$.

3) Yields upsampled CC.

4) Maximum of upsampled CC yields sub-pixel displacement estimate.

Particle Tracking

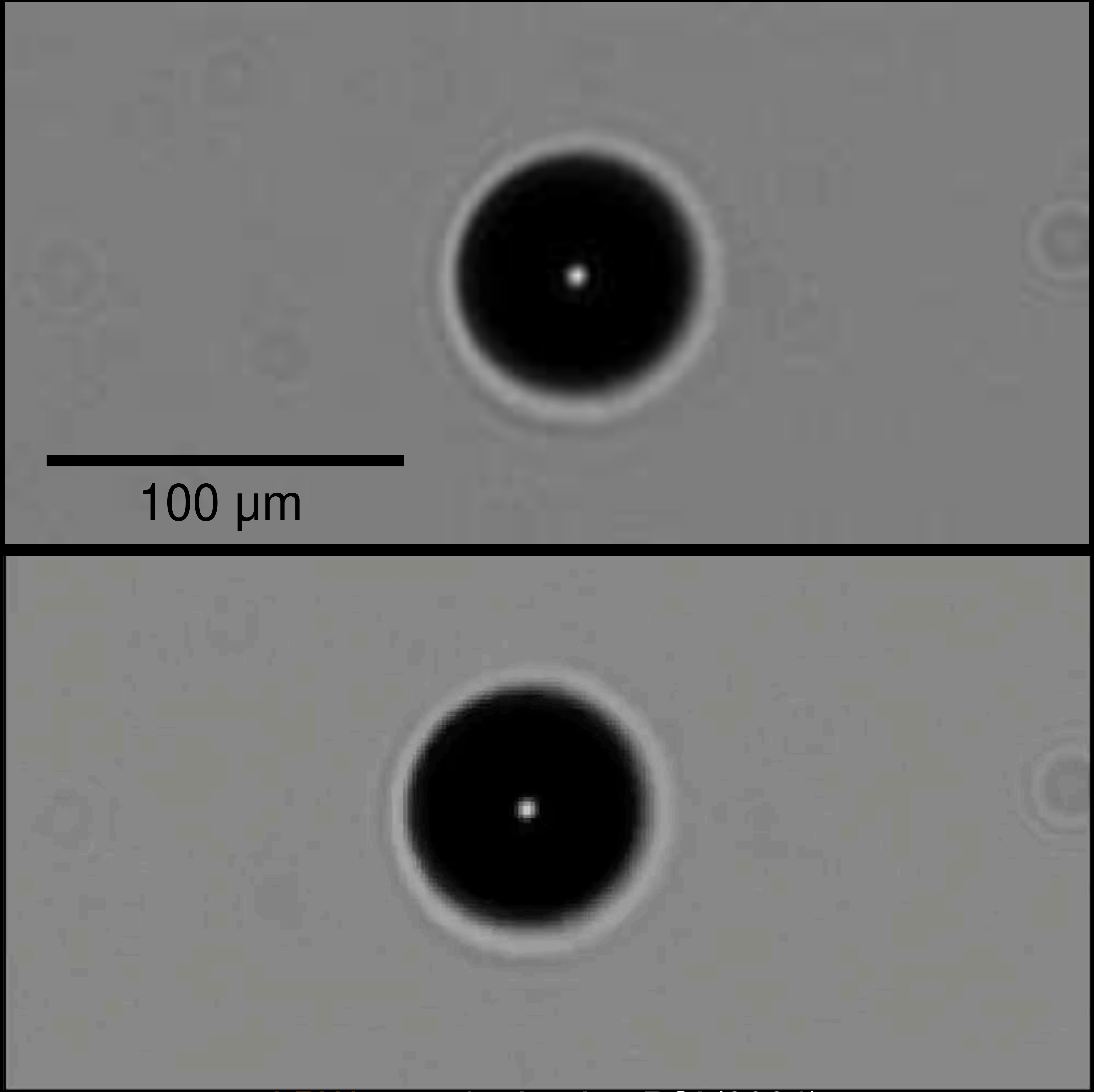


Tested against standard methods

- Fixed image size: 512 x 512.
- Fixed upsampling factor: 512.
- Comparable results to ideal Maximum Likelihood Estimation.
- Small fraction of the computational cost.

LRW++, *submitted to RSI* (2024)

Particle Tracking



LRW++, *submitted to RSI* (2024)

Real World Complications

- Fixed upsampling factor: 512.

!! Particle is no longer a simple Gaussian.

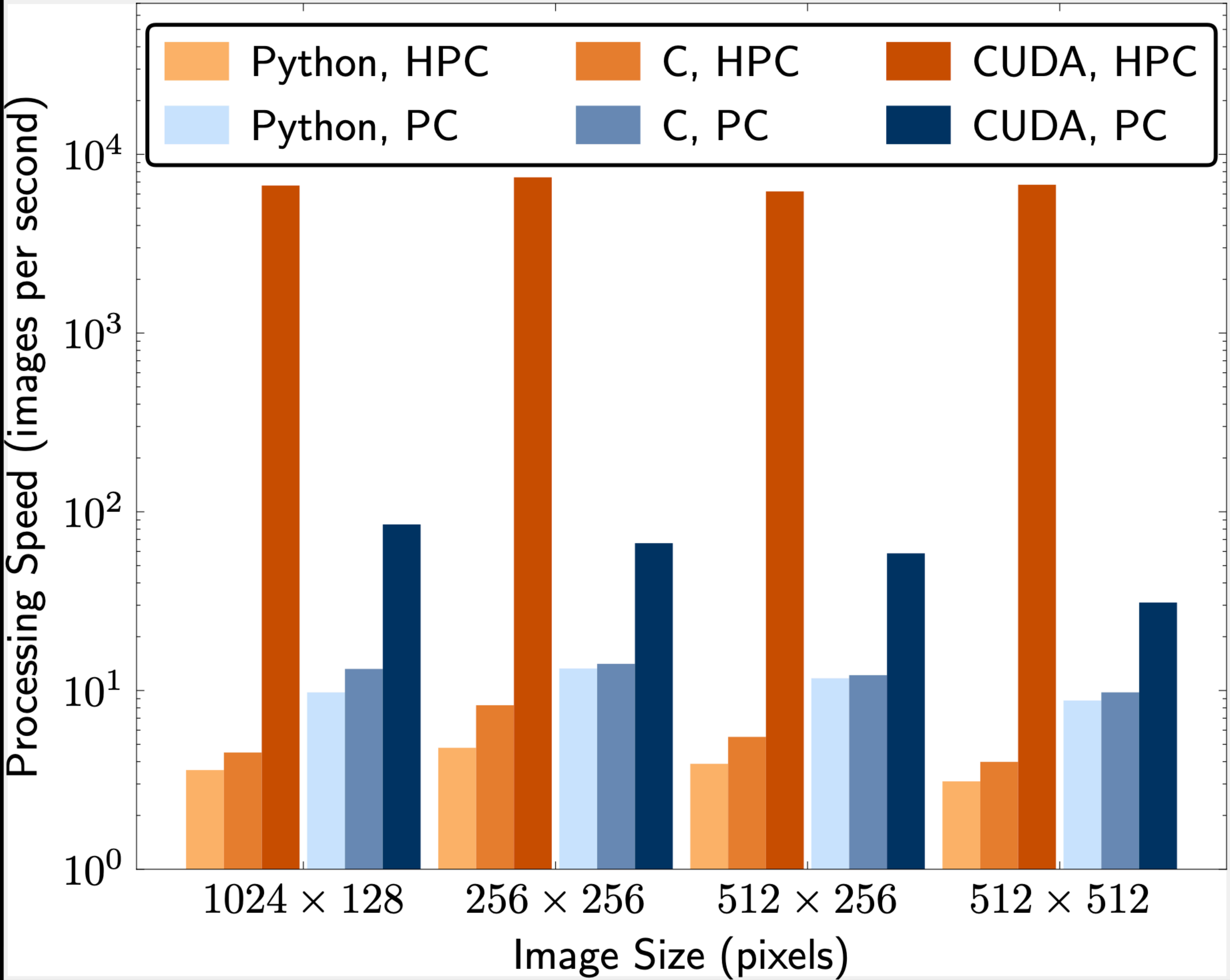
!! Two cameras.

  - Front: 256 x 128.
  - Side: 128 x 128.

!! Acquisition rate: 470 images/s/camera.

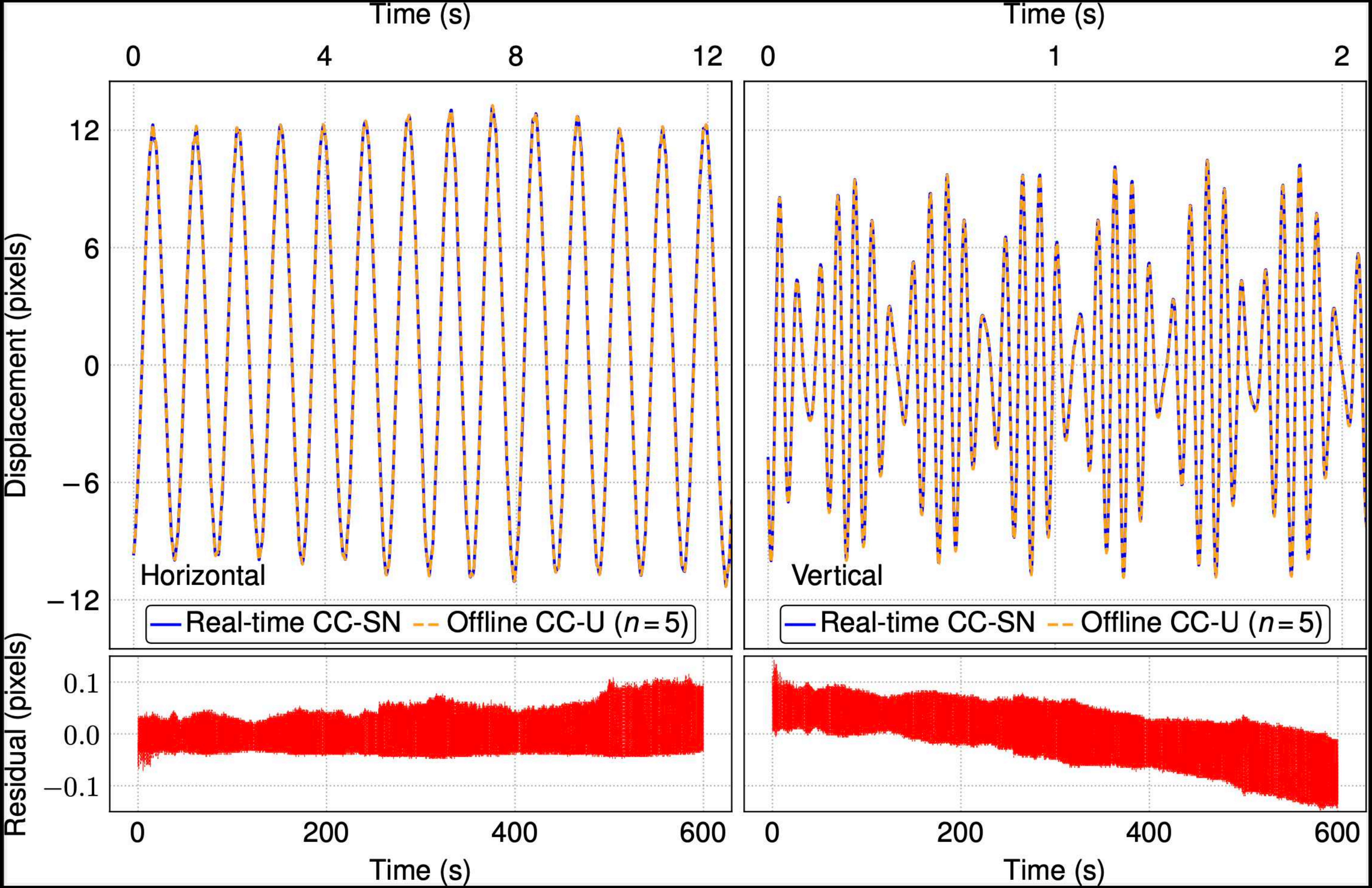
!! Need real-time analysis for feedback control.

RETINAS: The Real time Image Analysis Software



LRW++, *submitted to RSI* (2024)

RETINAS: The Real time Image Analysis Software



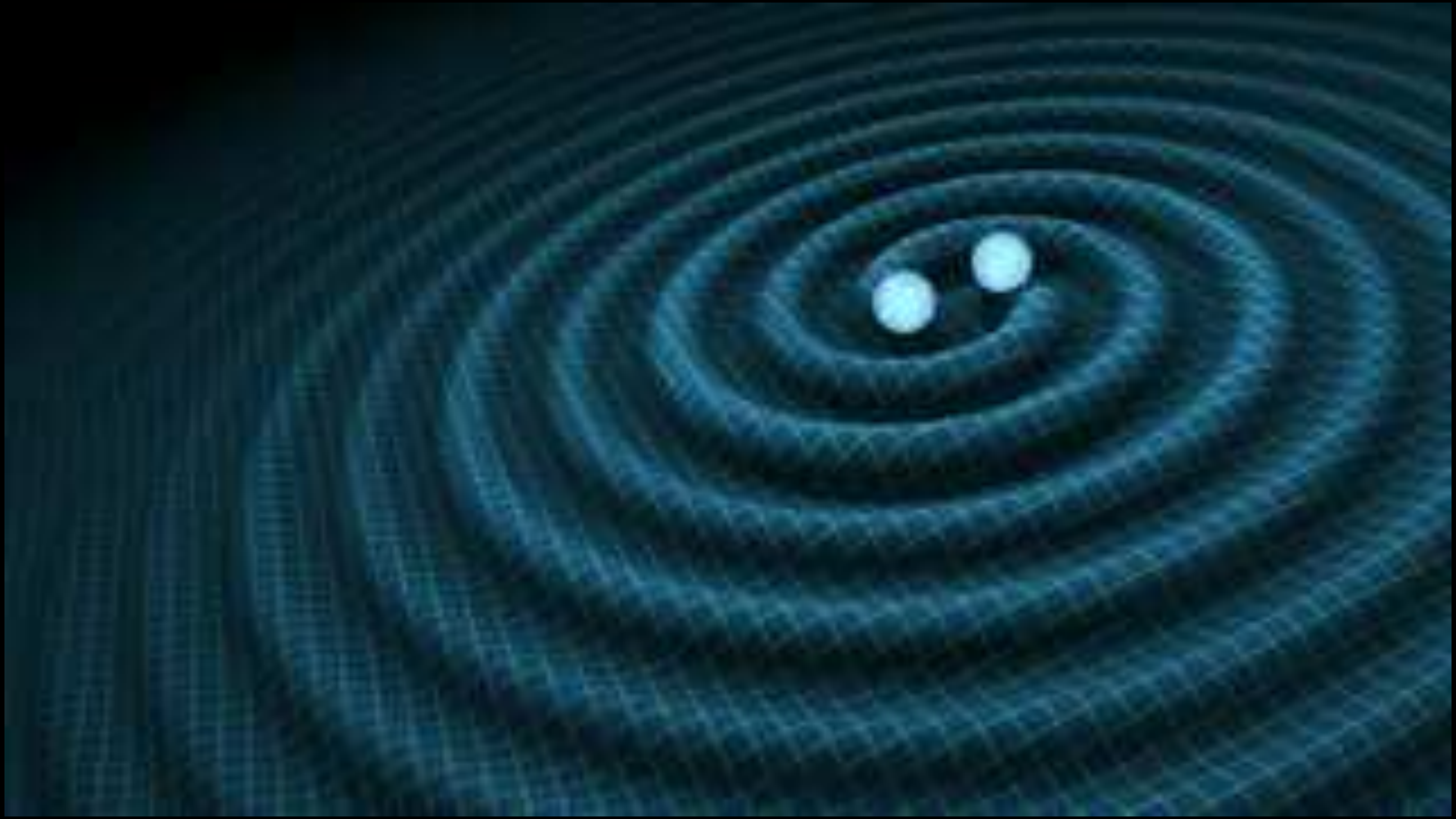
LRW++, *submitted to RSI* (2024)

Part II — Large Orbiting Stars

What are Gravitational Waves?

GWs are ripples in spacetime traveling at the speed of light, carrying source information.

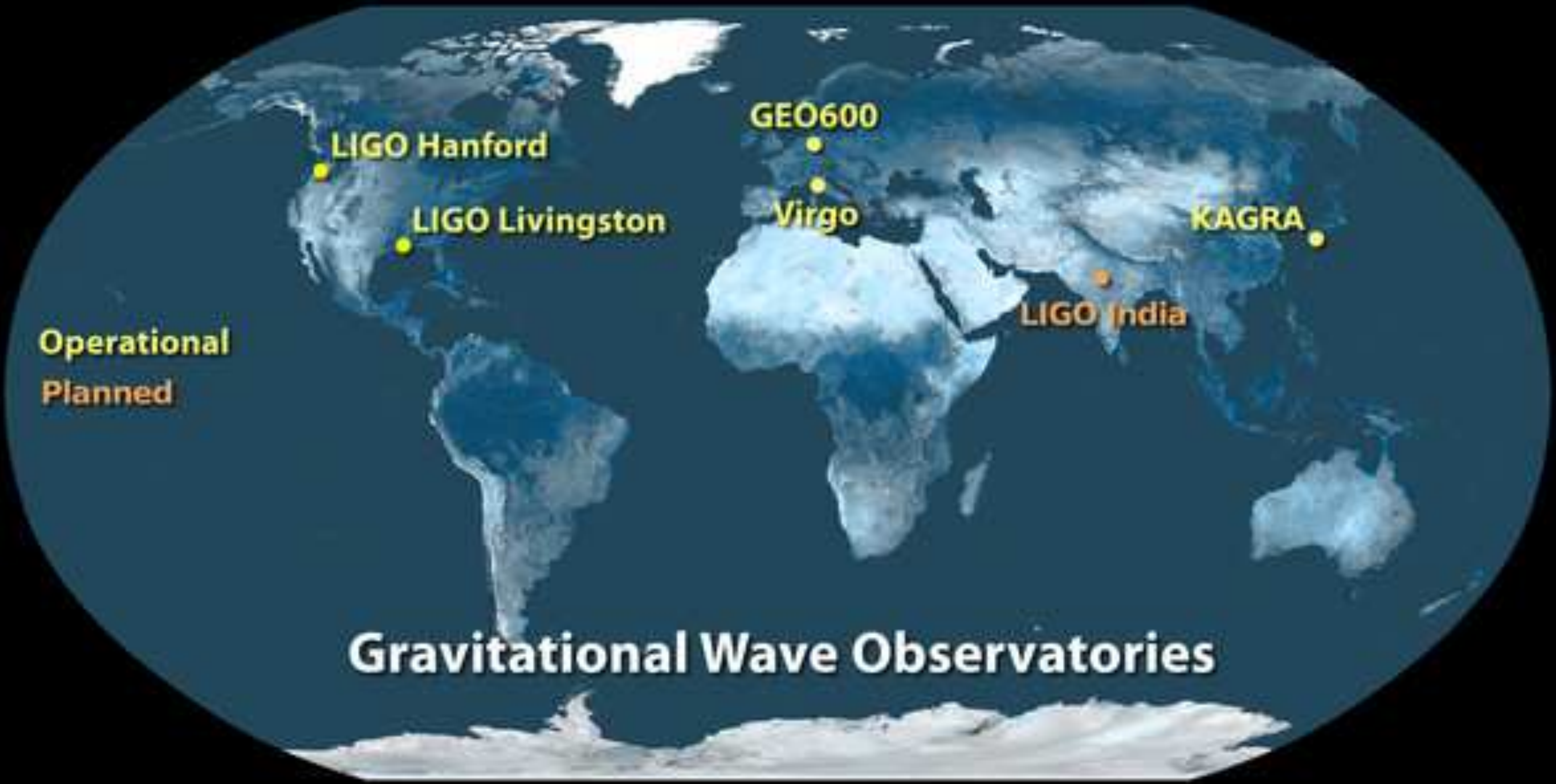
- Caused by motion or collapse of massive objects.
- Predicted by General Relativity, confirmed by detectors.



Credit: LIGO Caltech

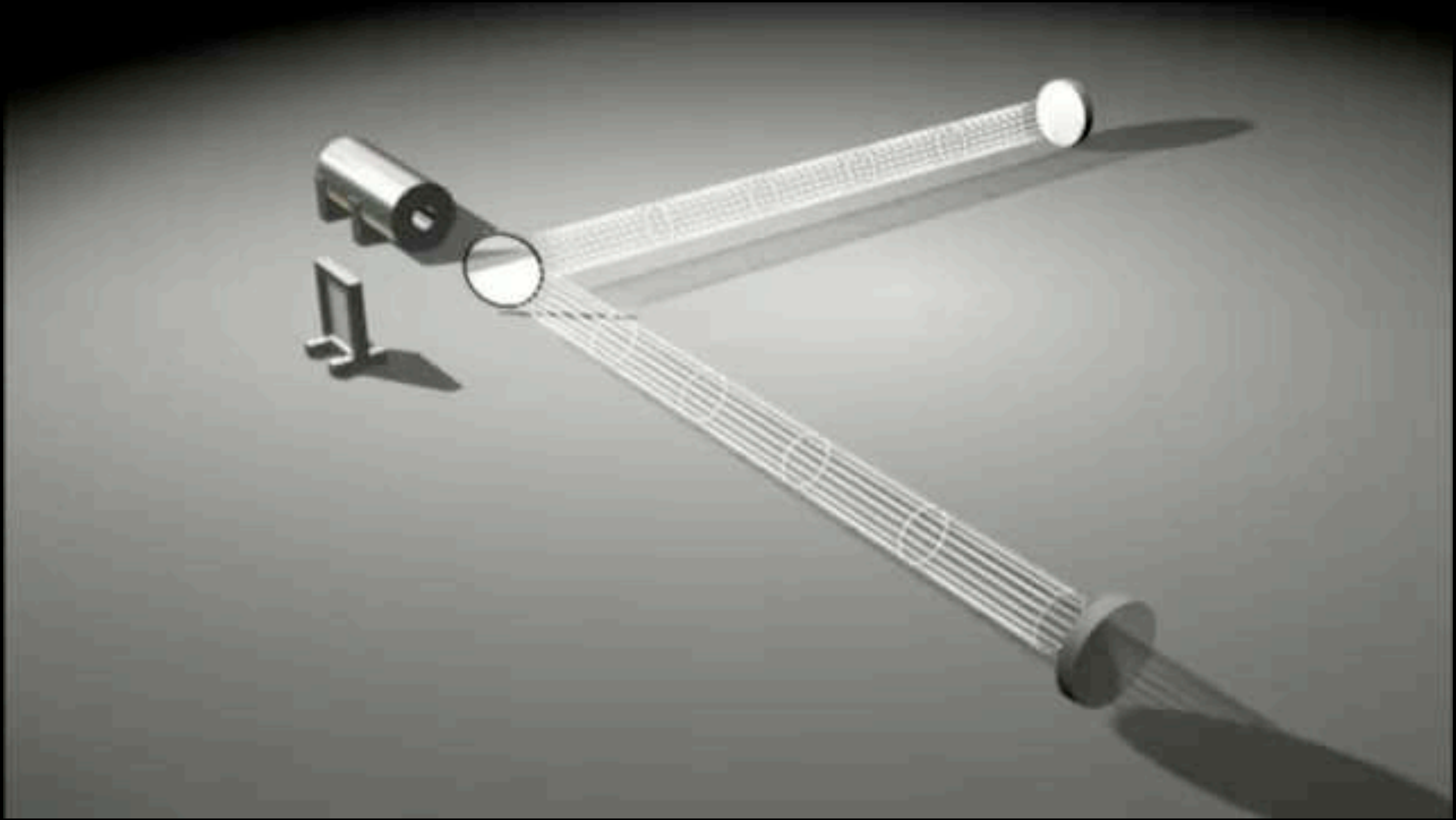
Detecting Gravitational Waves

- Tiny effects on matter ⇒ extremely difficult to detect.
  - Magnitude inversely proportional to source distance.
  - Earth length changes $\sim 10^{-19}\,\mathrm{m}$ (proton radius $\sim 8.33 \times 10^{-16}\,\mathrm{m}$).
- Specialized laser interferometers detect these changes.



Credit: LIGO Caltech

Detecting Gravitational Waves

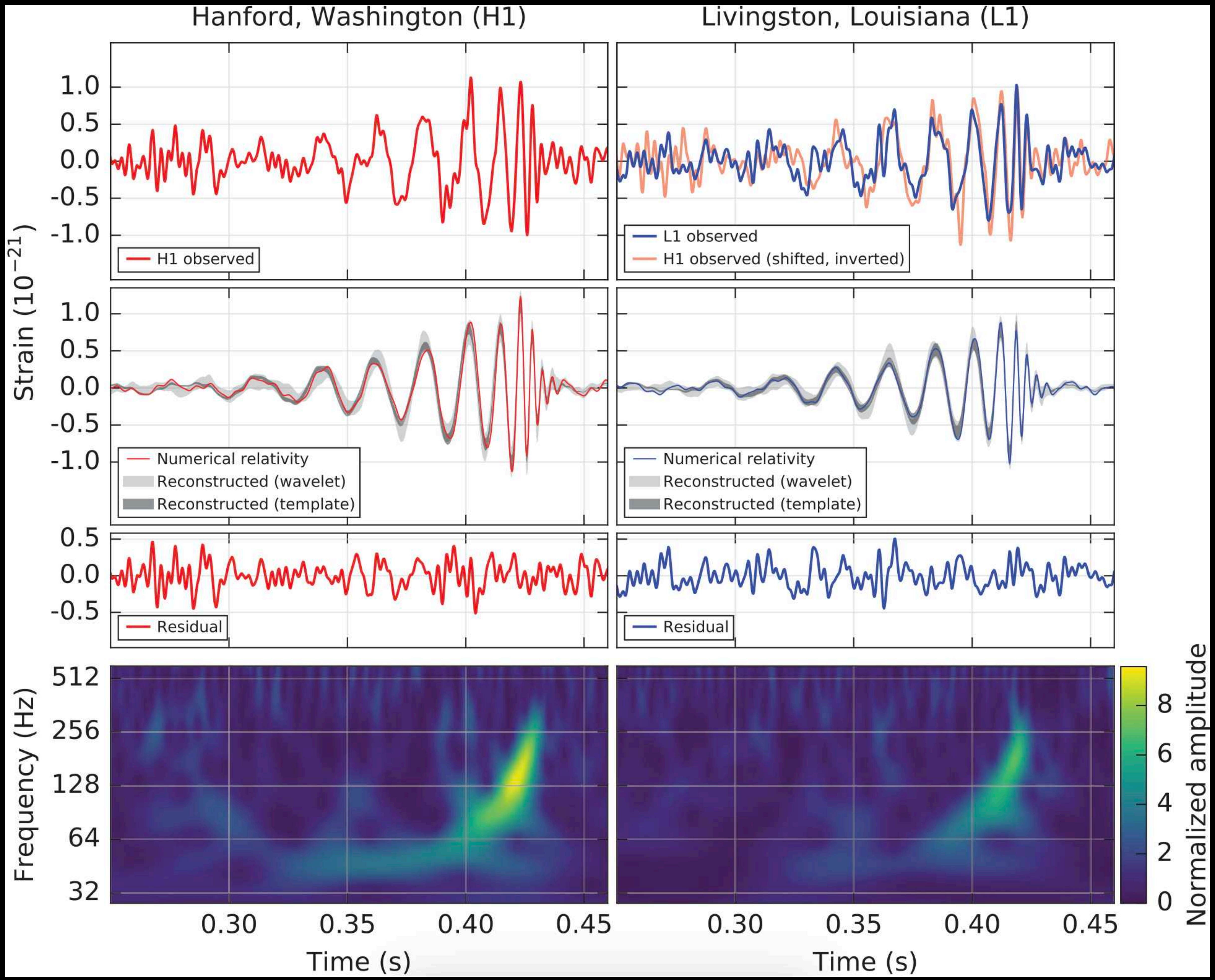


Credit: LIGO Caltech

Detecting Gravitational Waves

GW150914

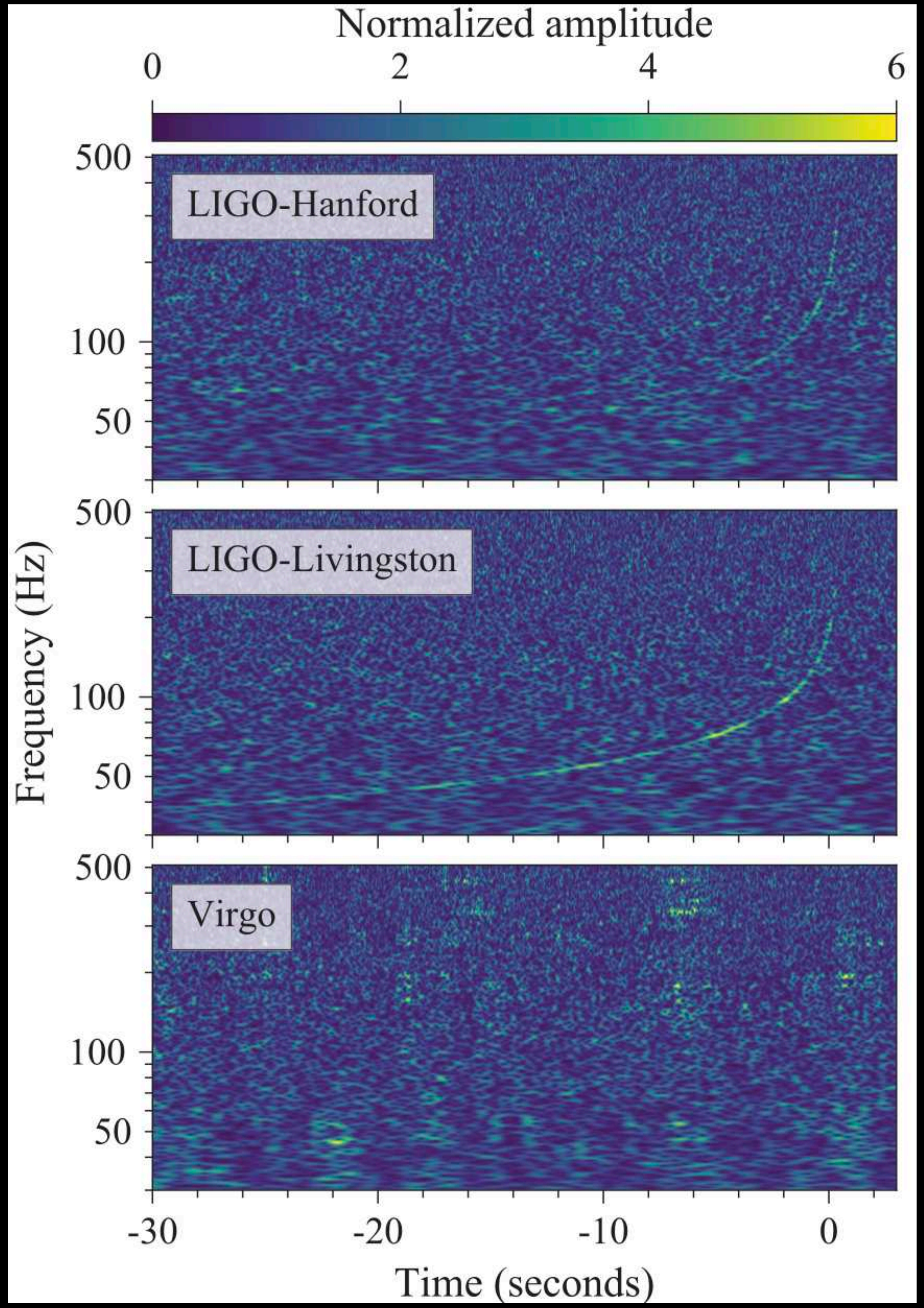
- Binary black holes.
- First detection ever.



Abbot *et al.*, PRL **116**, 061102 (2016)

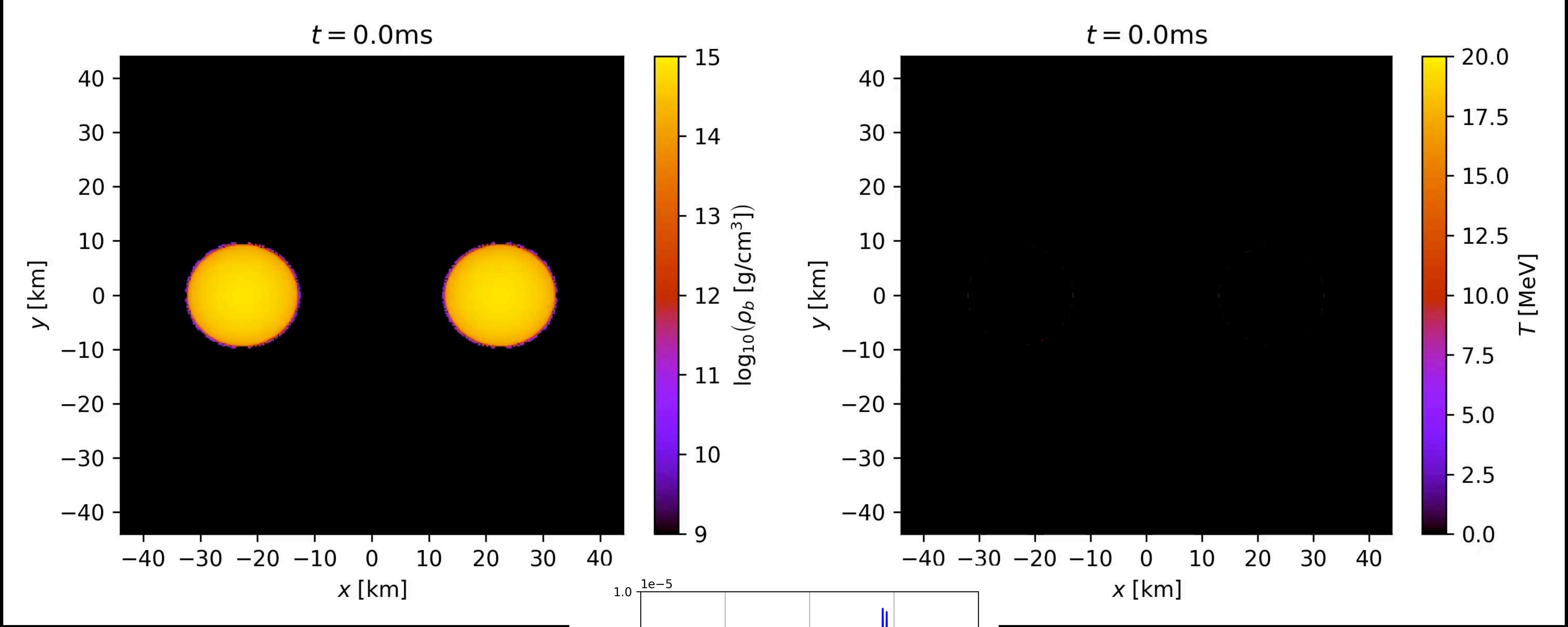
GW170817

- Binary neutron stars (BNS).
- First and only BNS detection to date.

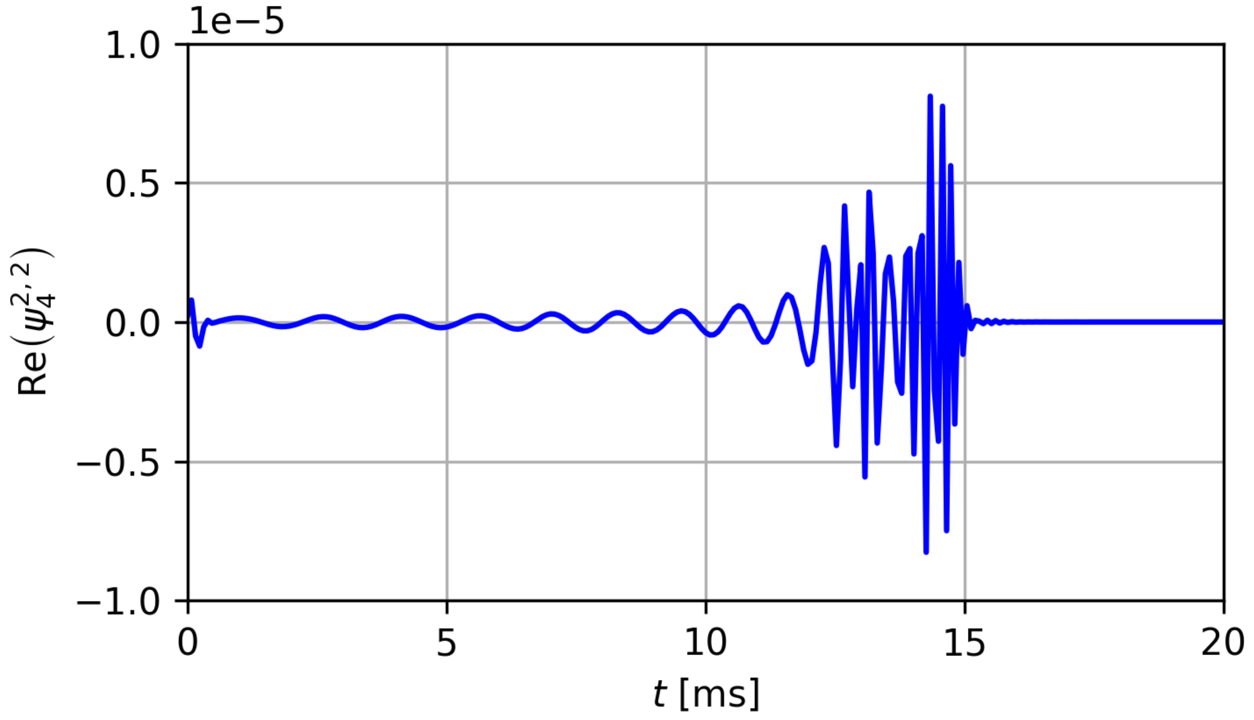


Abbot *et al.*, PRL **119**, 161101 (2017)

Binary Neutron Stars (BNS) Merger Simulations



- Equal-mass (1.39 solar masses)
- Magnetized
- Advanced EOS
- [O'Connor & Ott LS220 EOS]



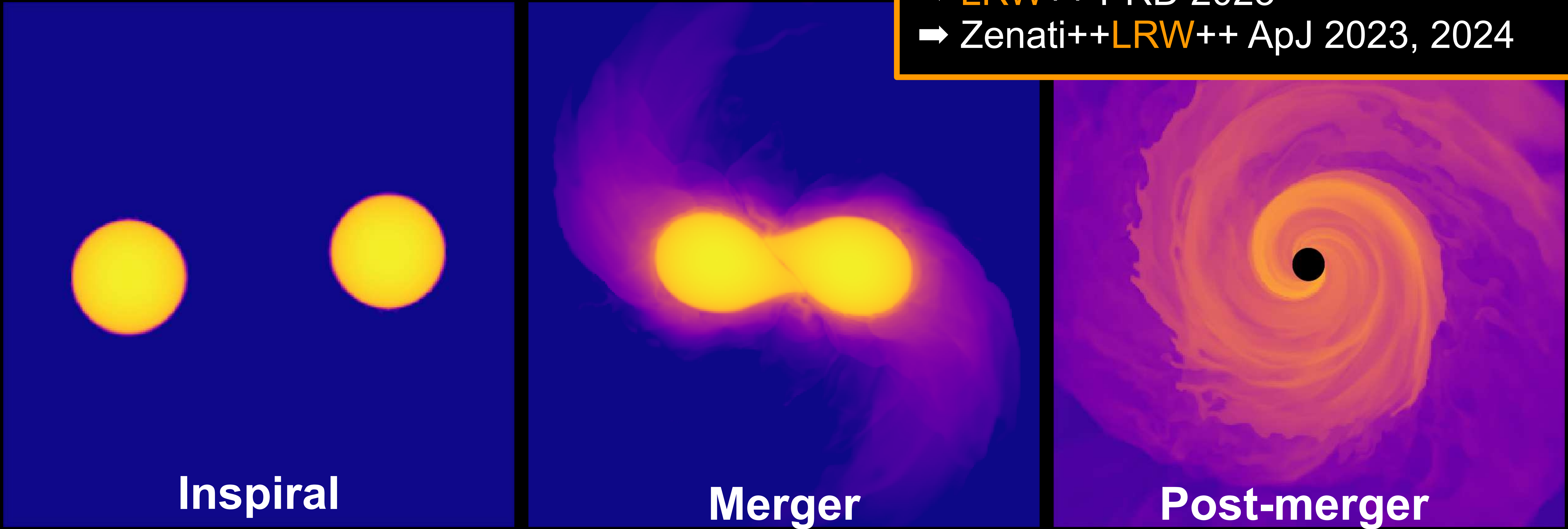
- Initial data produced by Tanmayee Gupte using LORENE (for more details see talks by T. Gupte and Josh Faber from 2021 TCAN Workshop)

Binary Neutron Stars (BNS) Merger Simulations

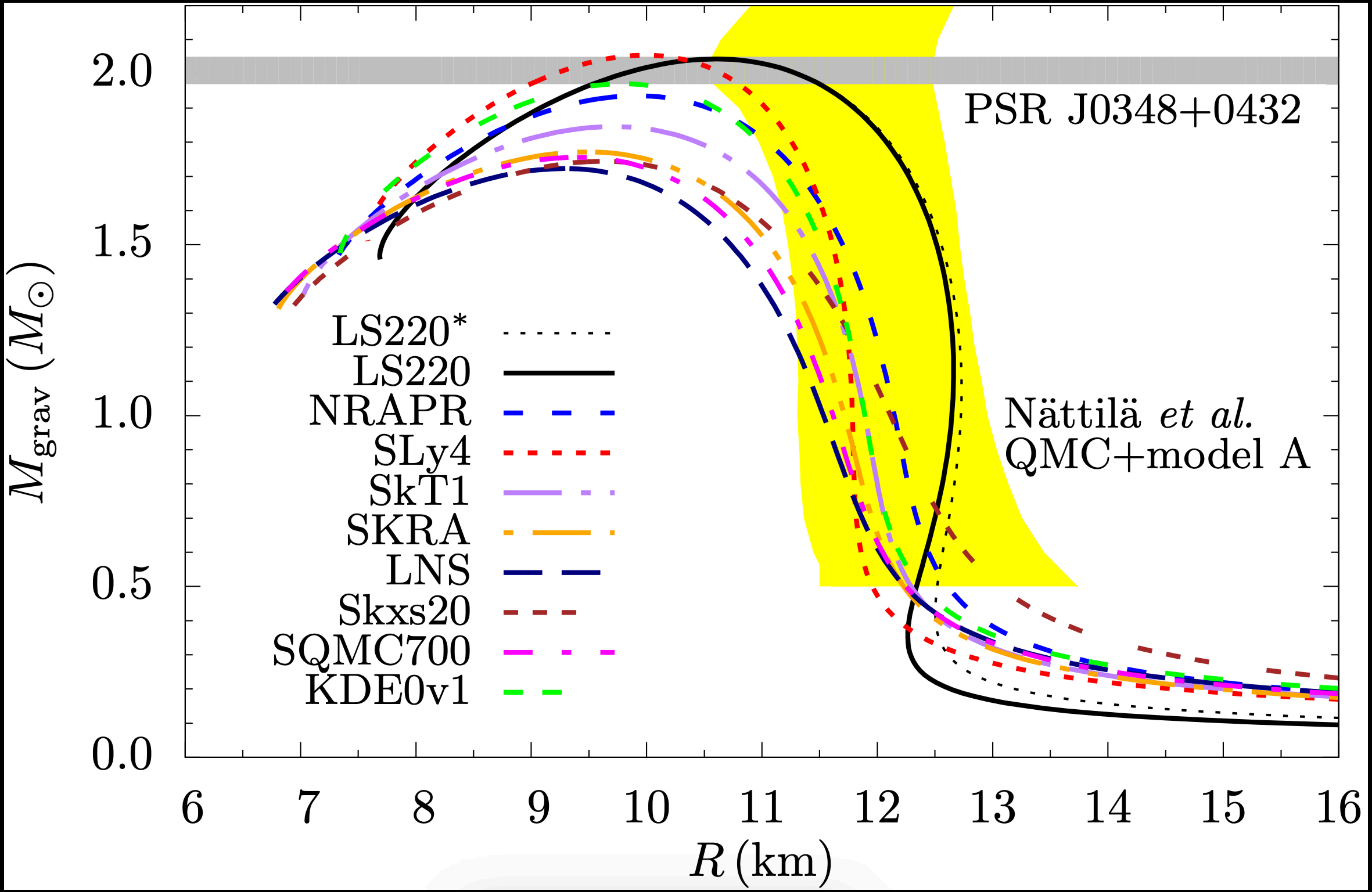
- GW170817 + GRB170817A + AT2017gfo: new frontier in multi messenger observations
- Current & future observations promise further insights
- Unanswered questions!
  - Equation of state of extreme nuclear matter?
  - How do dynamical ejecta lead to observed phenomena?

Much work has been done

- ➡ Living Reviews
  - ➡ Faber & Rasio, Springer 2012
    - ➡ Burns, Springer 2020
- ➡ Muguia-Berthier++LRW++ ApJ 2021
- ➡ Armengol++LRW++ PRD 2022
- ➡ LRW++ PRD 2023
- ➡ Zenati++LRW++ ApJ 2023, 2024

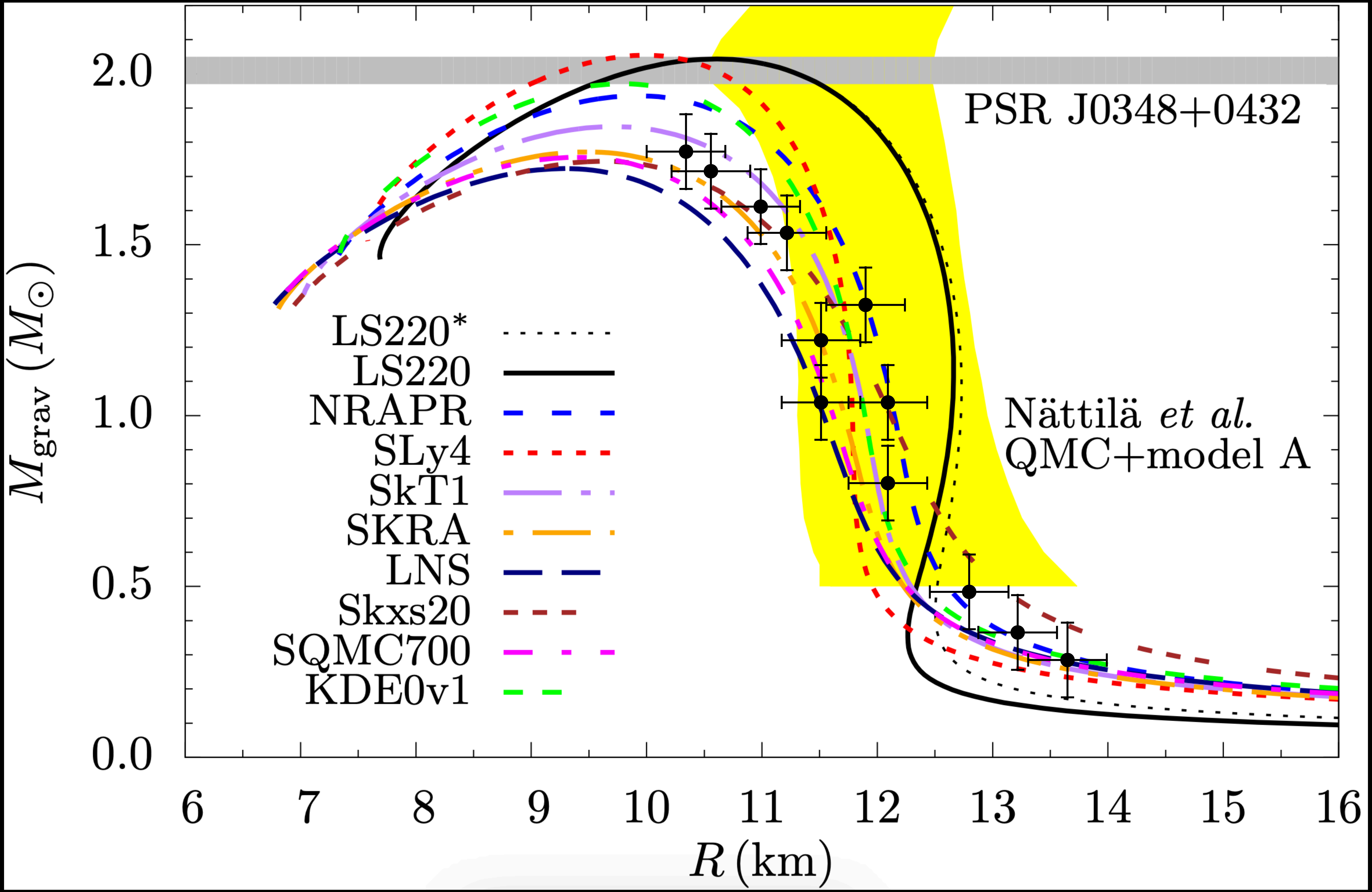


Binary Neutron Stars (BNS) Merger Simulations



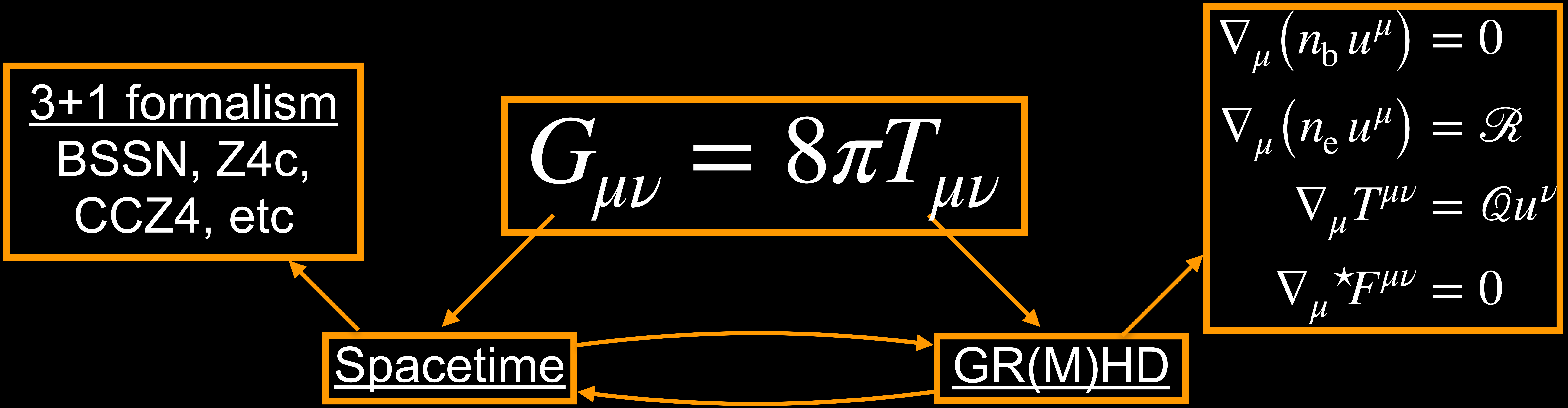
Schneider *et al.*, PRC **96**, 065802 (2017)

Binary Neutron Stars (BNS) Merger Simulations



Schneider *et al.*, PRC **96**, 065802 (2017)

How to Model BNS Numerically?



3+1 formalism
BSSN, Z4c,
CCZ4, etc

$$G_{\mu\nu} = 8\pi T_{\mu\nu}$$

$$\begin{aligned} \nabla_\mu \left(n_{\mathrm{b}}\, u^\mu \right) &= 0 \\ \nabla_\mu \left(n_{\mathrm{e}}\, u^\mu \right) &= \mathscr{R} \\ \nabla_\mu T^{\mu\nu} &= \mathscr{Q} u^\nu \\ \nabla_\mu {}^\star F^{\mu\nu} &= 0 \end{aligned}$$

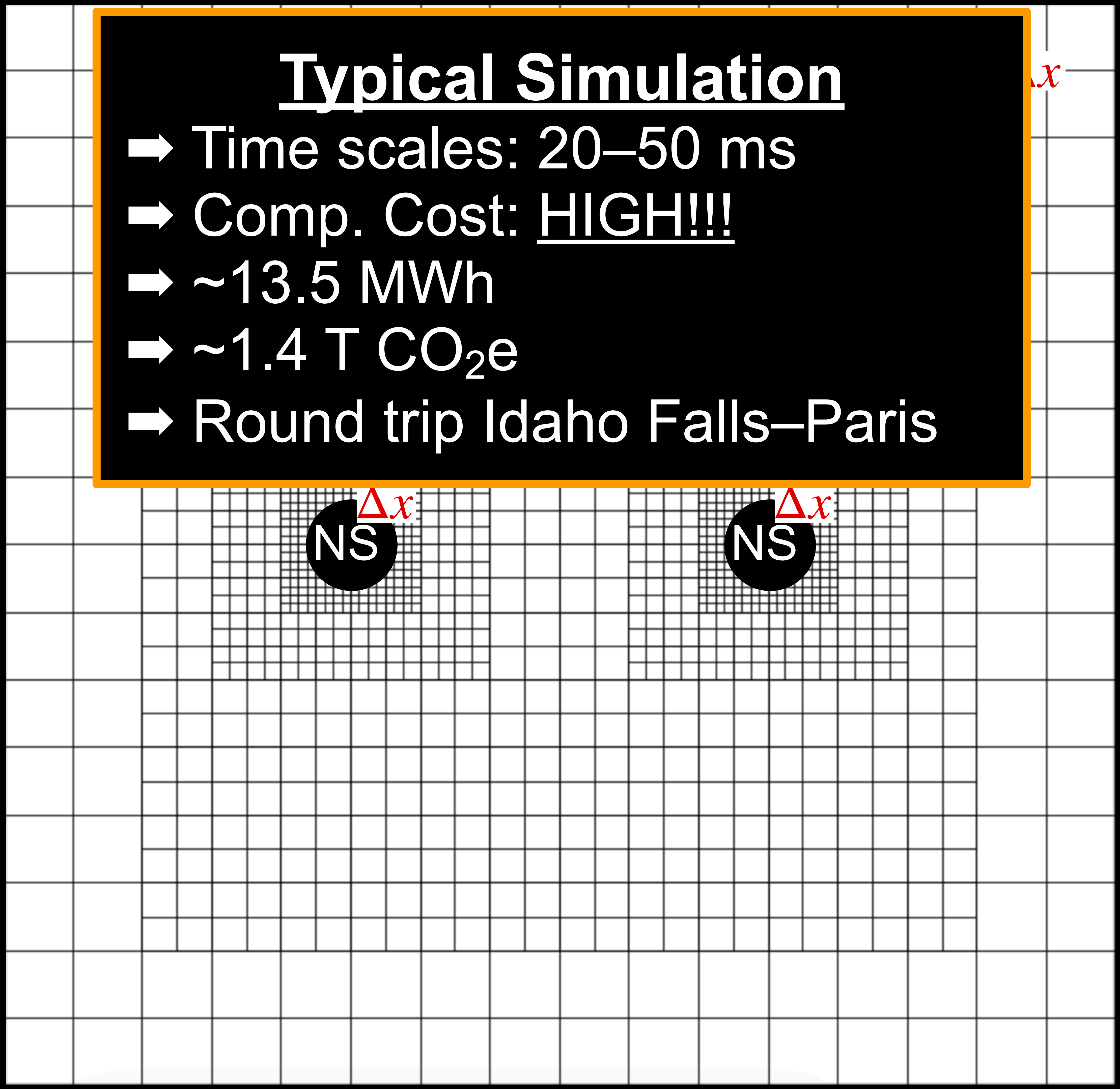
Spacetime

GR(M)HD

GR(M)HD Modeling — Grid-Based Schemes

- !! Prominent Example: Cartesian AMR
- ✓ Established, robust method
- ✓ Relatively straightforward implementation
- ✓ Effective at capturing shock features
- ✗ Discontinuous changes in resolution
- ✗ Strong numerical artifacts e.g., reflections
- ✗ Systems with near-symmetries: <u>highly inefficient</u>
- ✗ Eulerian method: inefficient for certain flows

$\approx 17 \times 10^9$ gridpoints



<u>Typical Simulation</u>

- ➡ Time scales: 20–50 ms
- ➡ Comp. Cost: <u>HIGH!!!</u>
- ➡ ~13.5 MWh
- ➡ ~1.4 T CO_2e
- ➡ Round trip Idaho Falls–Paris

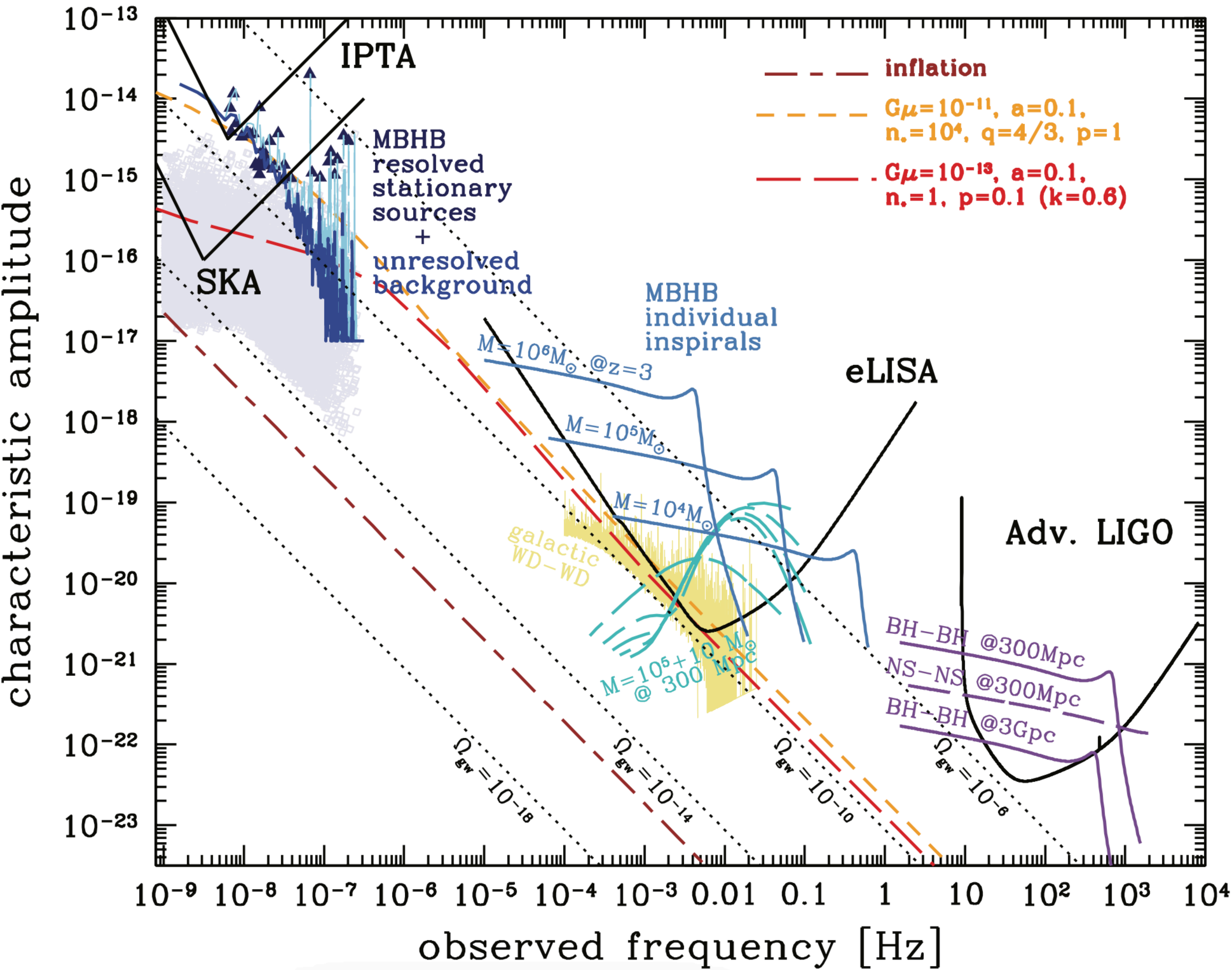
Box-in-box Cartesian AMR grids

Part III — Supermassive Stars

Gravitational Wave Sources Across the Spectrum

The GW spectrum is as wide as the EM spectrum, requiring multiple observatories:

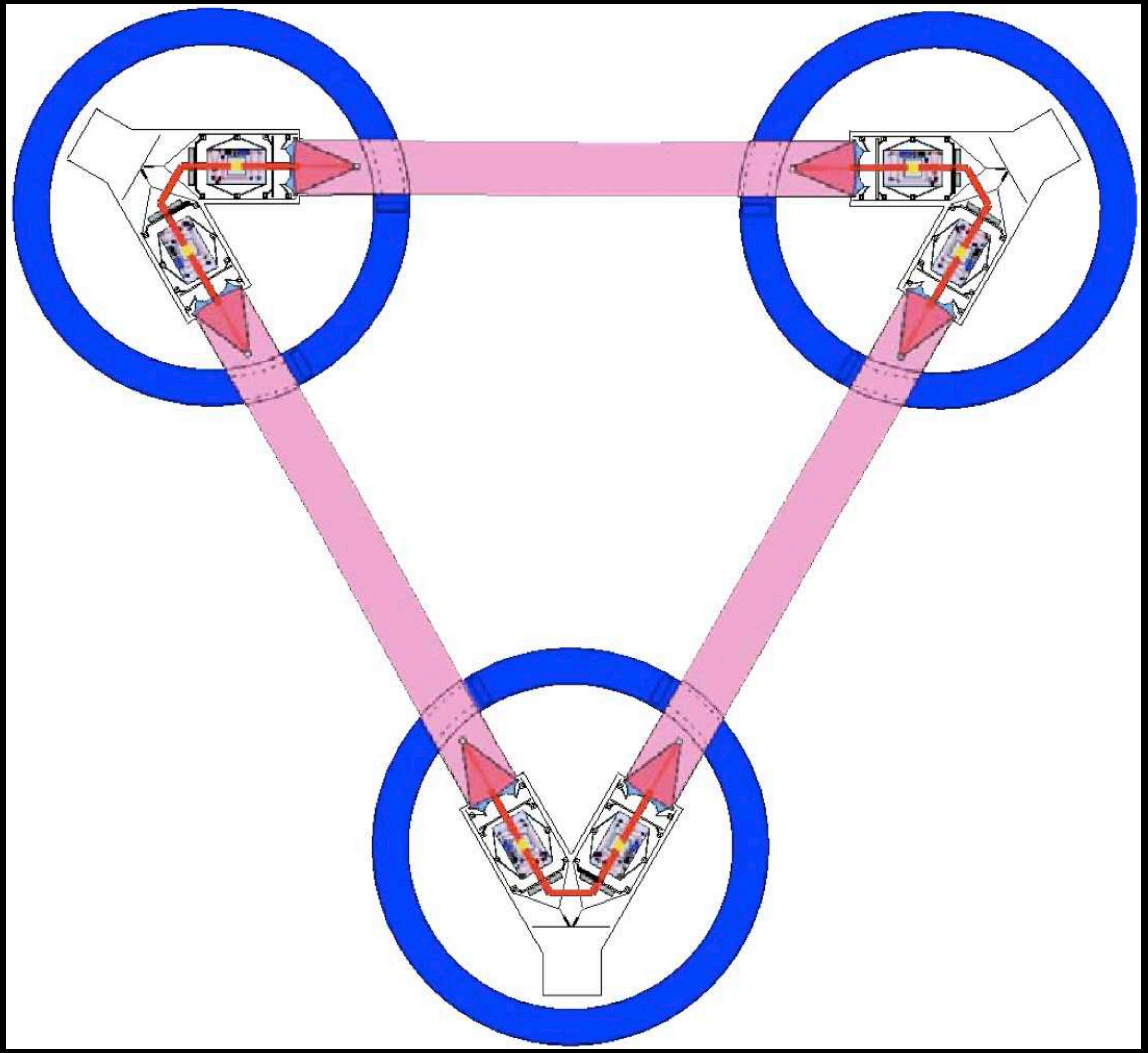
- **Hz range**: (advanced) LIGO, Virgo, KAGRA — detected many stellar black hole and some neutron star mergers.
- **nHz range**: Pulsar Timing Arrays (PTAs) — tracking stable pulsars for 15+ years, showing evidence of primordial GW background from supermassive black-hole binaries.
- **mHz range**: Laser Interferometer Space Antenna (LISA) — adopted by ESA, scheduled launch in mid-2030s.
- **In between**: other concepts, none actively in development.



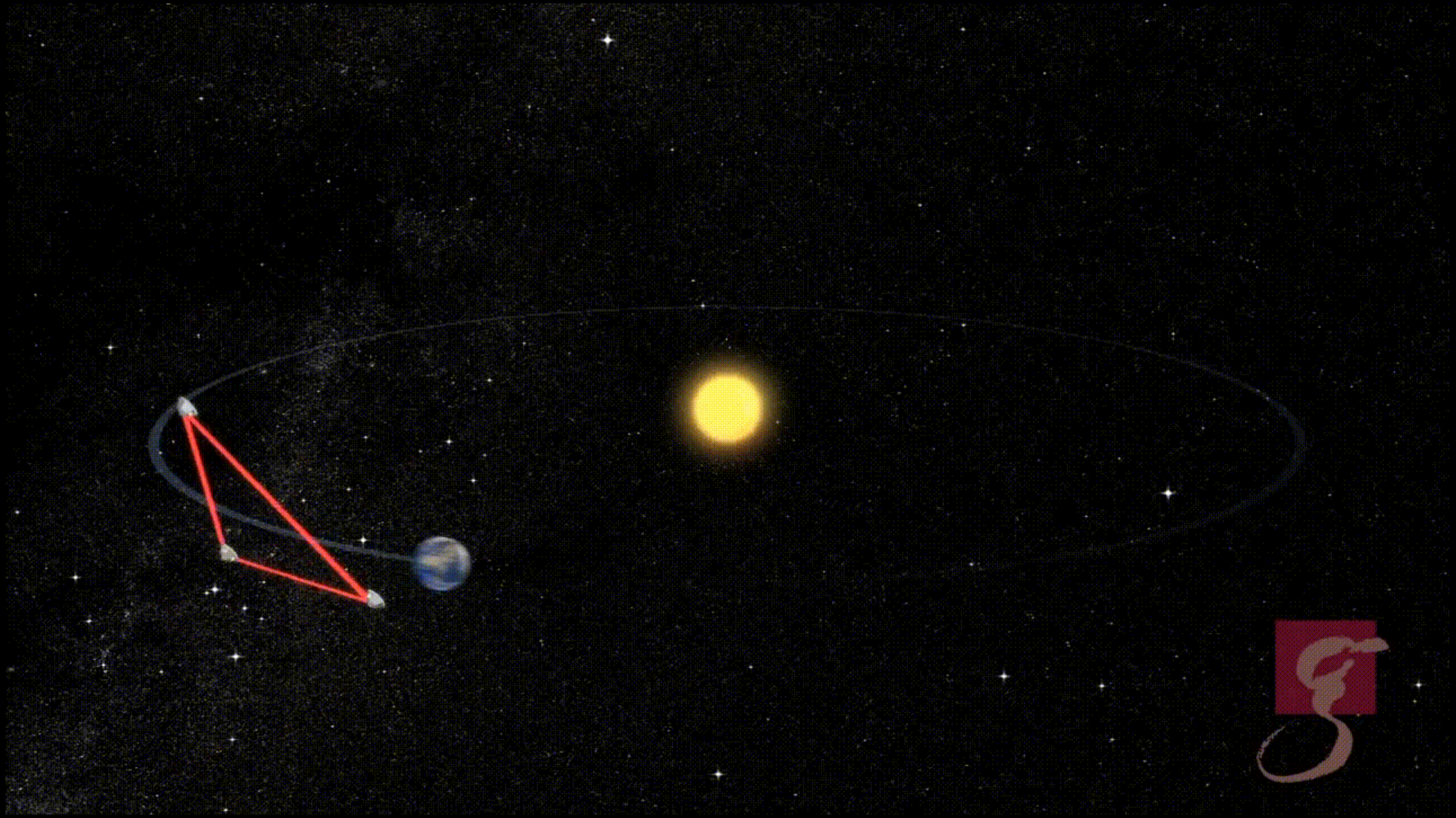
Lommen, Rep. Prog. Phys. **78** 124901 (2015)

The Laser Interferometer Space Antenna (LISA)

- Three spacecraft in an equilateral triangle, 2.5 million km apart.
- Similar to LIGO/Virgo/KAGRA, but much larger scale.
- Targets low-frequency GWs.
- Resolution of 20 pm over a million km.
- Scheduled to launch in the mid-2030s.



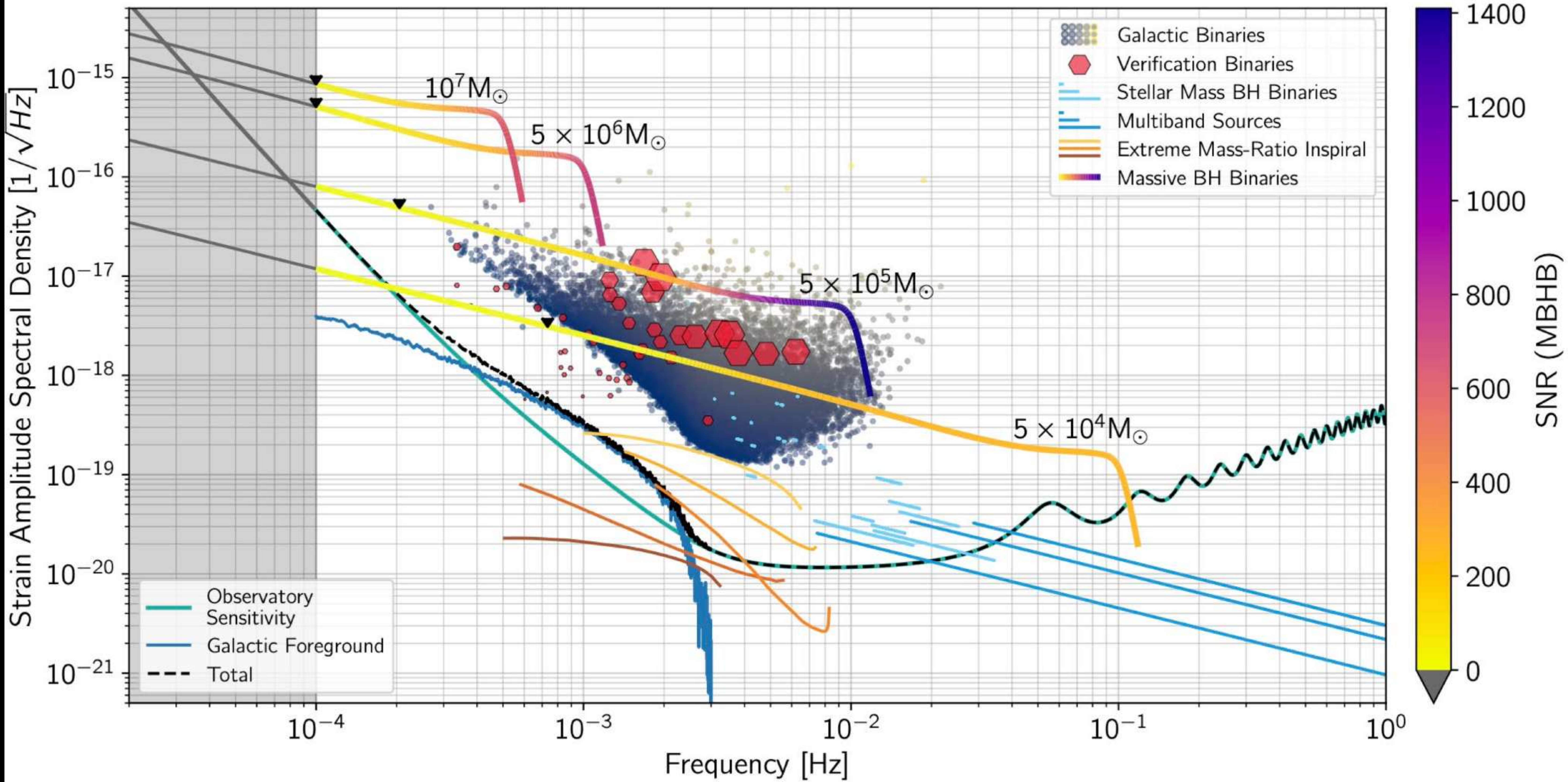
Credit: LISA Mission Concept



Credit: NASA

LISA Standard Gravitational Wave Sources

- Merging supermassive black holes.
- Extreme-mass-ratio inspirals.
- Galactic white dwarf binaries.
- All persistent binary sources with high accumulated SNR.



Credit: ESA LISA Definition Study Report

LISA Standard Gravitational Wave Sources

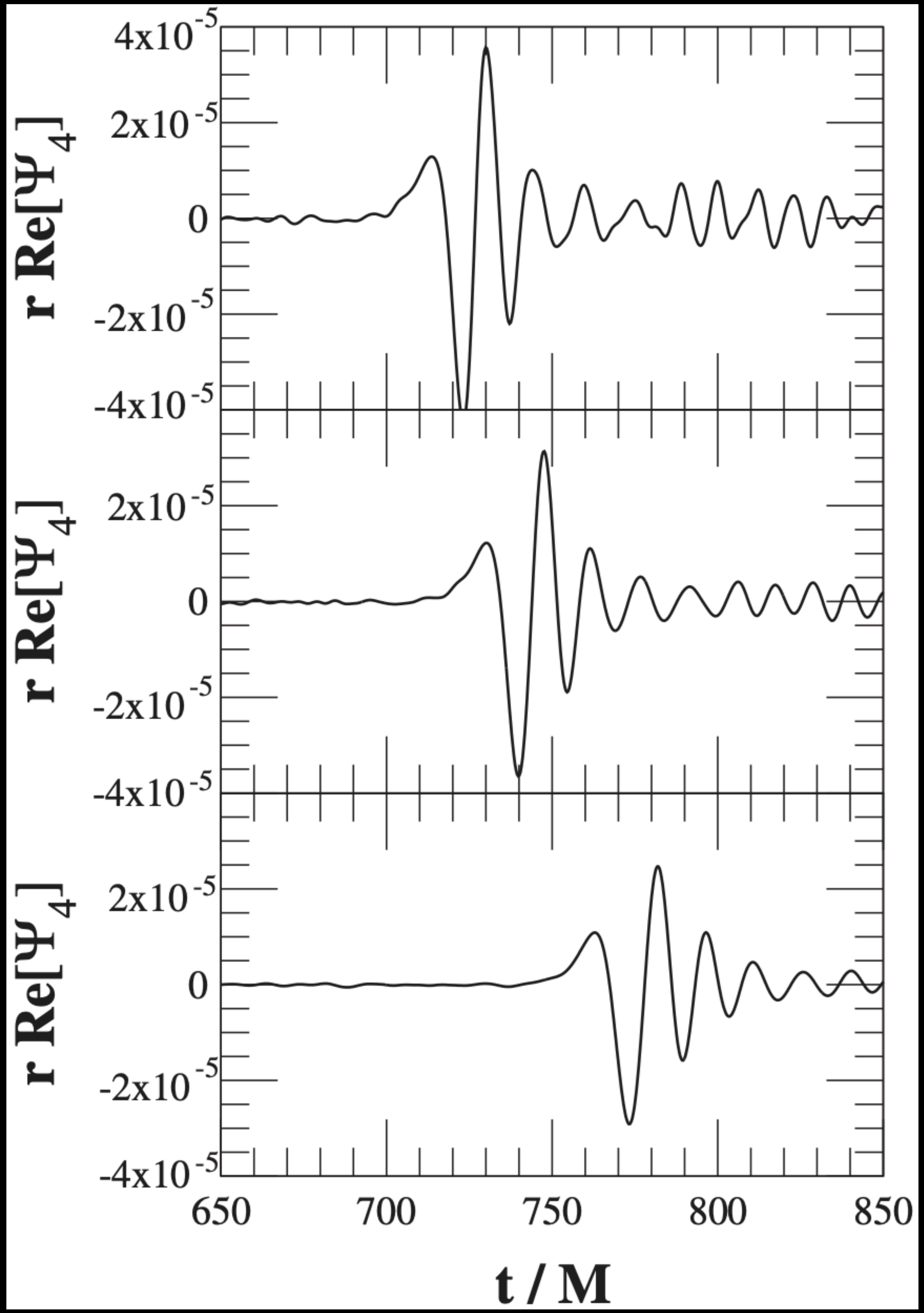
Origin of supermassive black holes:

- Merger of smaller stellar BHs;
- Rapid growth via accretion; and/or
- Direct collapse of gas to massive seeds

Investigating DCBH formation is crucial for understanding early Universe population.

Potential significant GW source for LISA, if waveforms are known.

Currently, only a few numerical waveforms and crude analytical estimates.

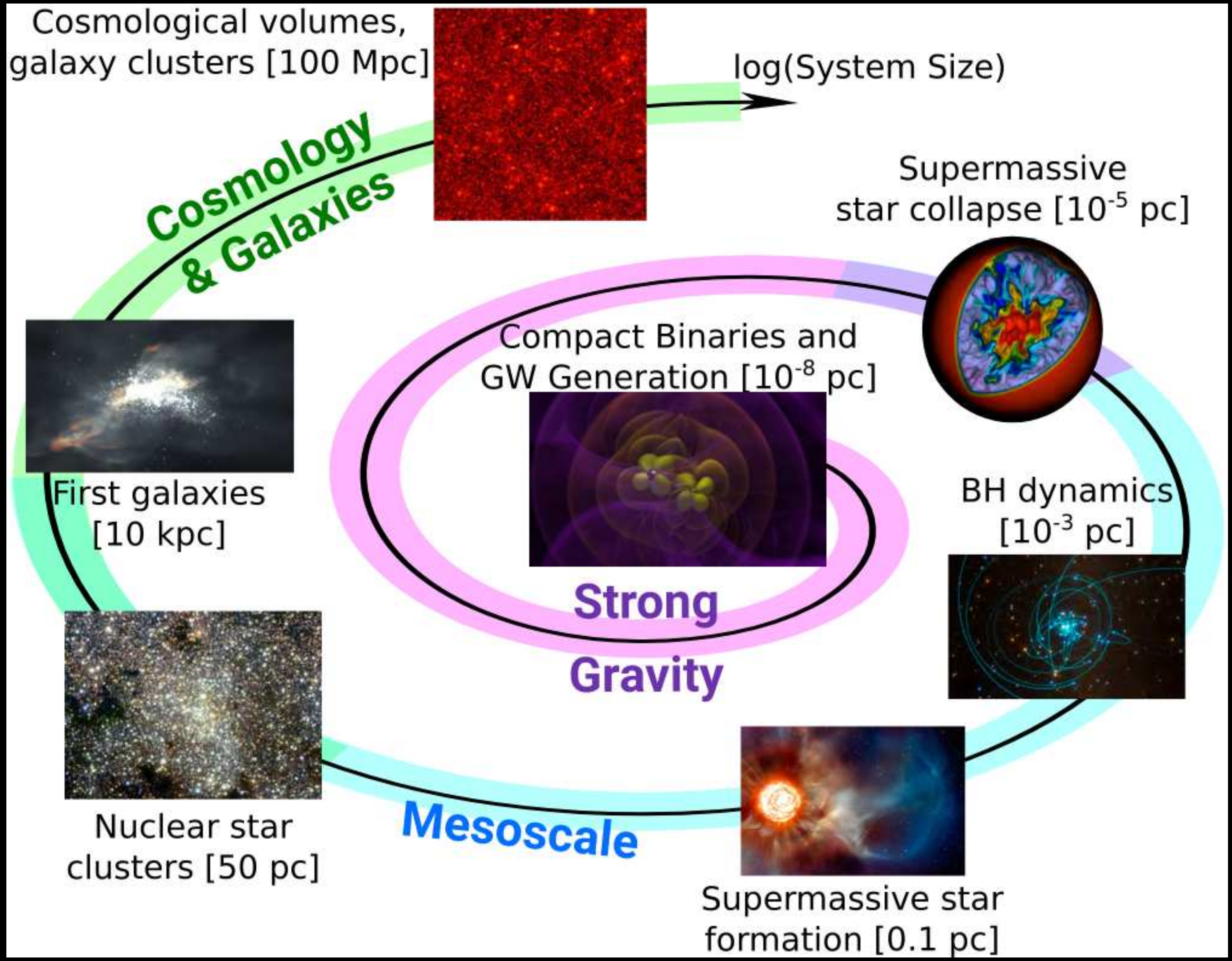


Saijo & Hawke, PRD **80**, 064001 (2009)

A Multi-Code Approach to DCBH Formation

The Enzo Project

MESA



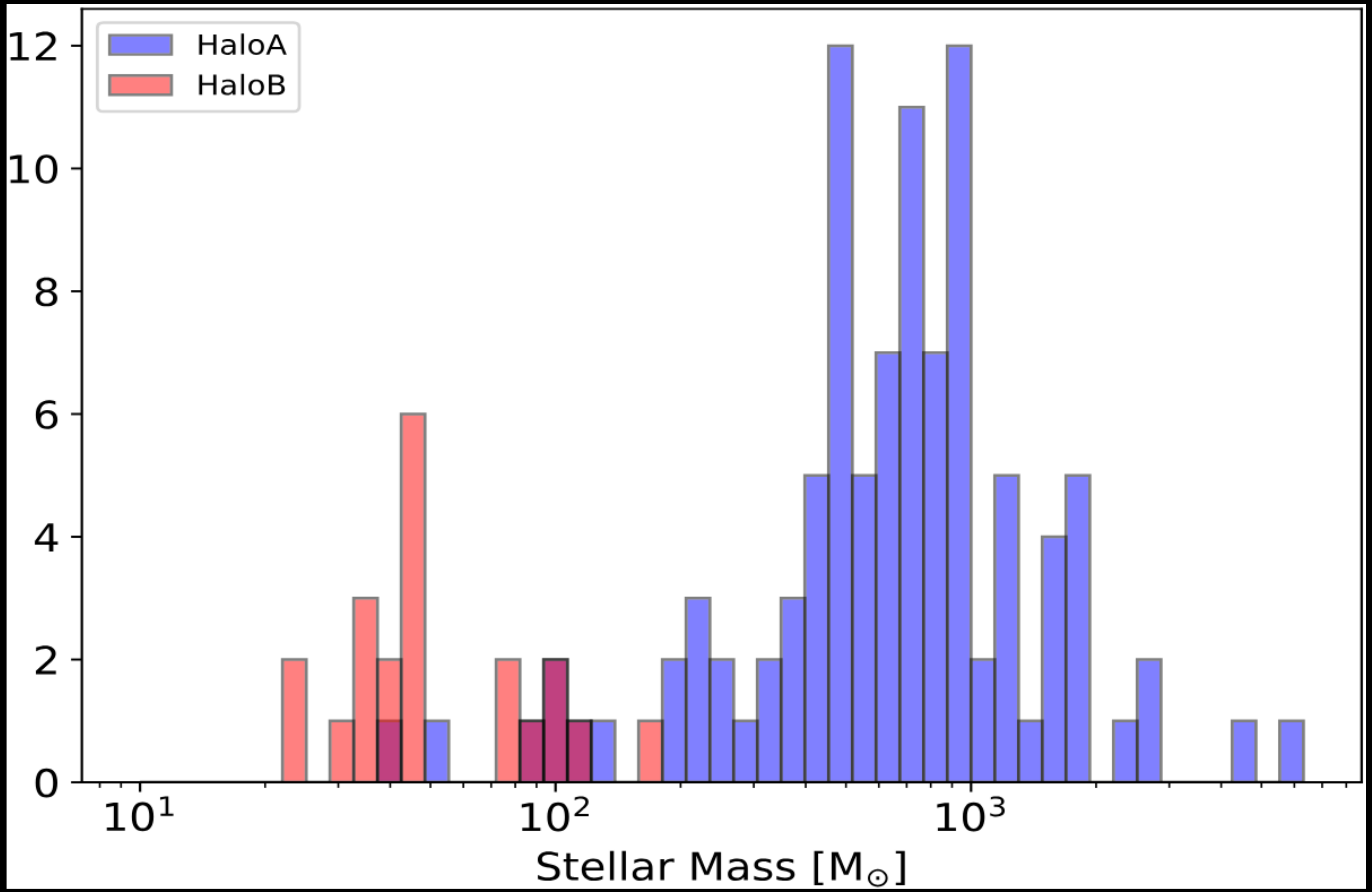
IDEA: Derive initial conditions from large-scale simulations instead of random/idealized configurations:

1. Cosmological simulations like Renaissance.
2. Newtonian protostellar evolution code like MESA, up to strong-field regime.
3. Full Numerical Relativity with the Einstein Toolkit for strong gravity scenarios.

The Cosmological Scale

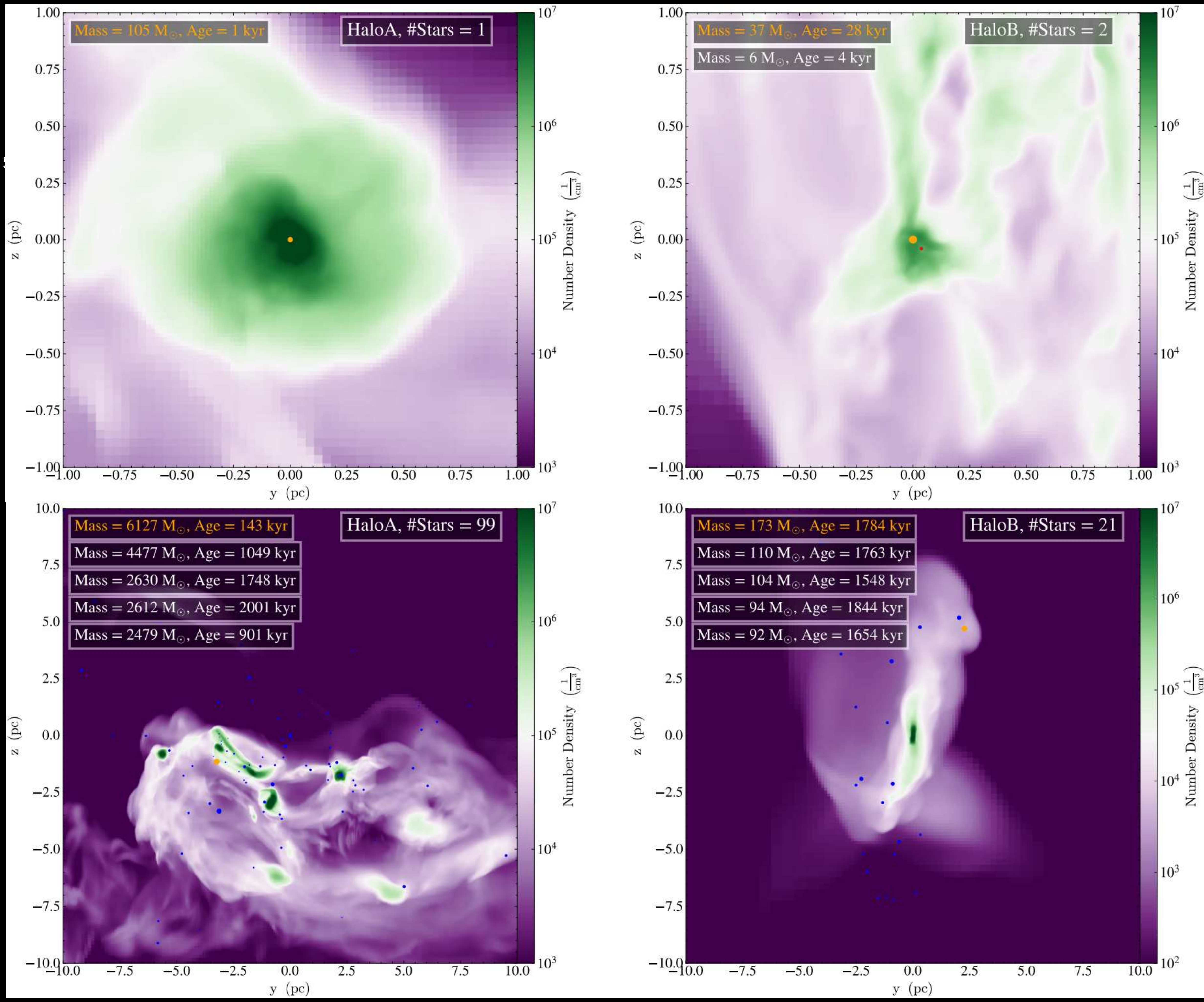
The Renaissance Simulations (RenSims; rensimlab.github.io) were conducted in 2013-2015 using Enzo on the Blue Waters supercomputer.

Output: regions likely to collapse into supermassive stars (SMS), with density, pressure, angular momentum profiles, etc.



Likely SMS masses identified from halos

Regan et al., OJA **3** (2020)



Initial and final density halos from RenSims

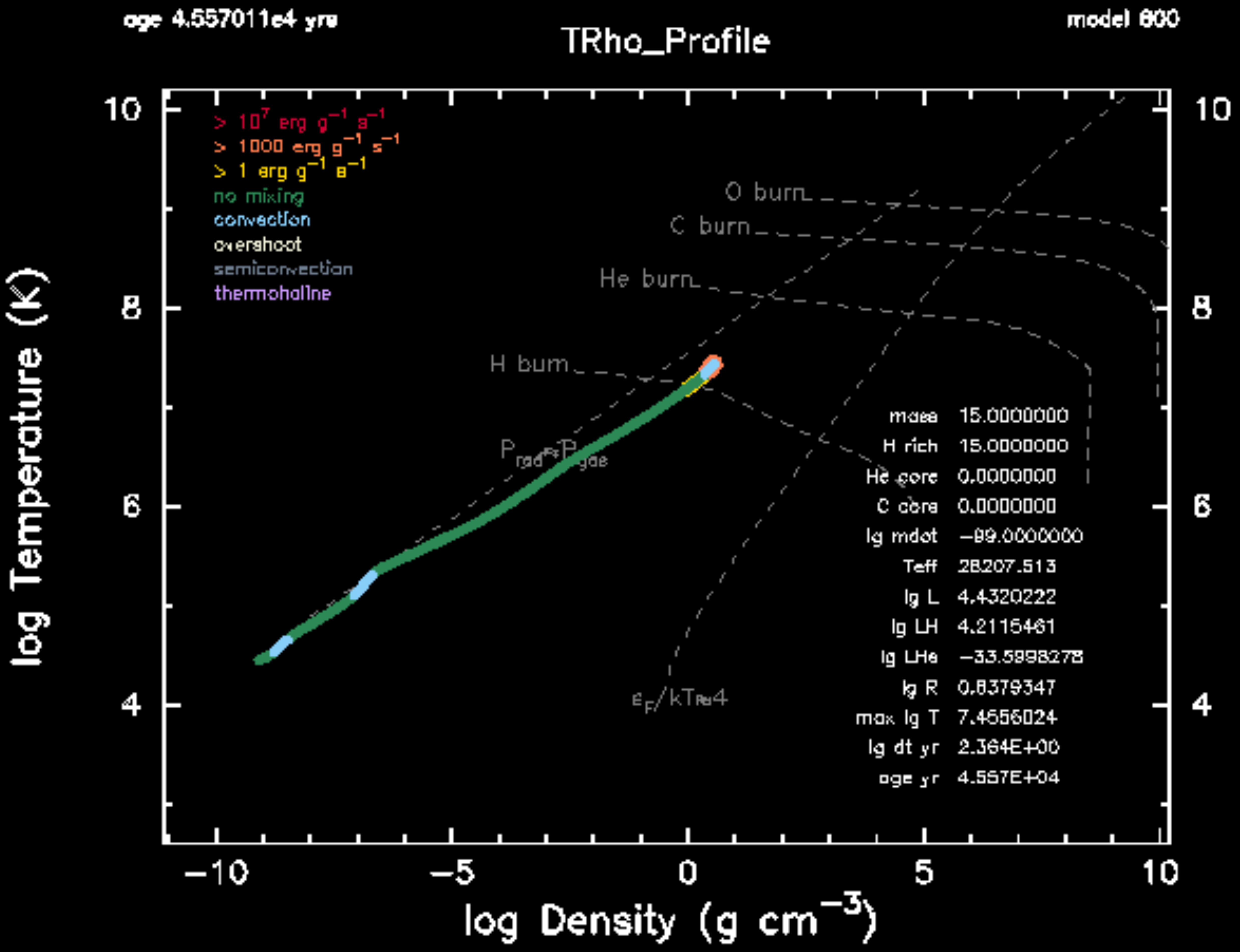
The Mesoscale

Modules for Experiments in Stellar Astrophysics (MESA; docs.mesastar.org) is a 1D stellar evolution code with rotation support.

Performs full stellar evolutions with arbitrary equations of state and nuclear processes.

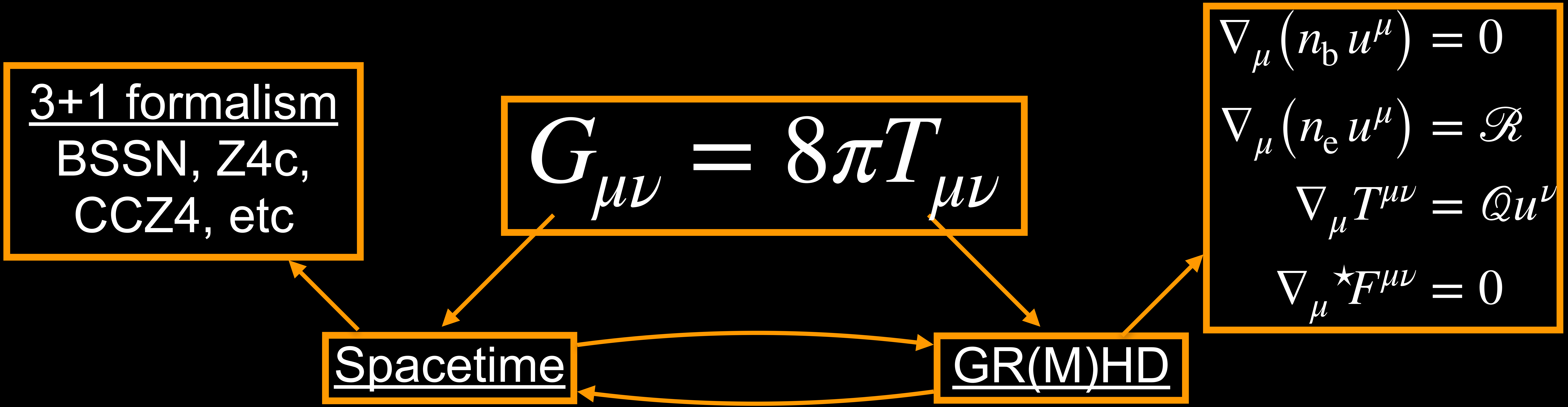
Input: Bulk information (fluid density, rotation info)

Output: 1D hydrodynamic field profiles, approximate equation of state



Density vs Temperature evolution from sample MESA run

The Strong Gravity Scale



3+1 formalism
BSSN, Z4c,
CCZ4, etc

$$G_{\mu\nu} = 8\pi T_{\mu\nu}$$

$$\nabla_\mu \left(n_\mathrm{b}\, u^\mu\right) = 0$$
$$\nabla_\mu \left(n_\mathrm{e}\, u^\mu\right) = \mathscr{R}$$
$$\nabla_\mu T^{\mu\nu} = \mathcal{Q} u^\nu$$
$$\nabla_\mu {}^\star F^{\mu\nu} = 0$$

Spacetime

GR(M)HD

Part IV — Key Software Overview

The Einstein Toolkit

The Einstein Toolkit

- Simulate cutting edge science.
- Use latest numerical methods.
- Make use of latest hardware.
  - Cache.
  - Vectorization.
  - Accelerators.
  - Scale to many cores.
  - Scale to many nodes.

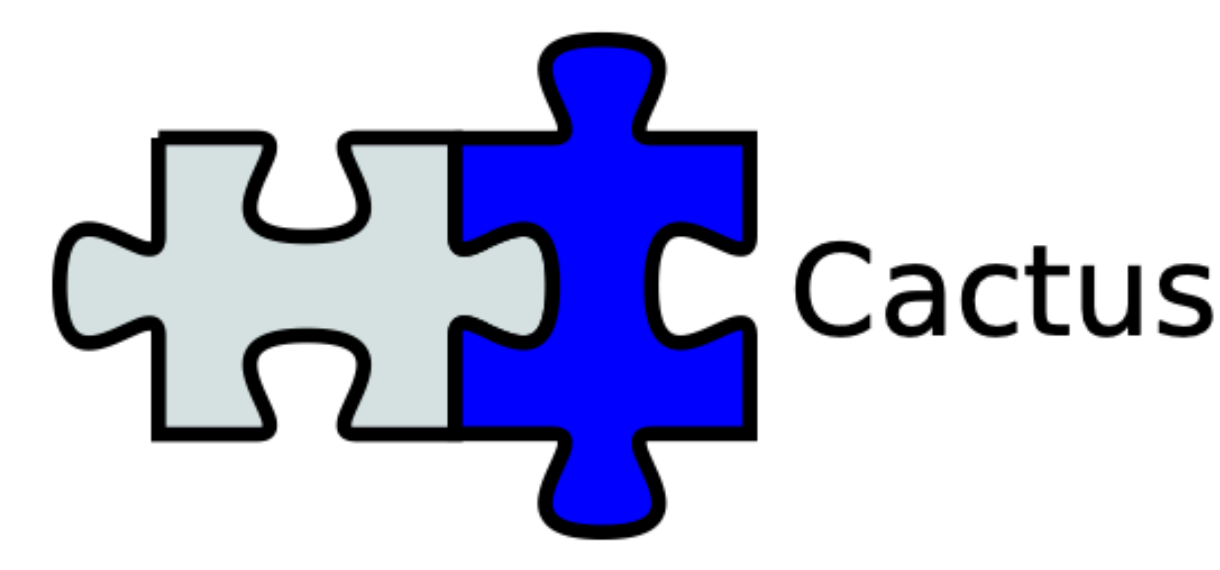
- “Mundane” tasks
  - Efficient I/O
  - Checkpoint/Restart
  - Parameter Parsing
  - Visualization
  - Analysis
  - Steering

- Efficient use of all hardware is complex.
- Requires:
  - Experts from different disciplines.
  - Careful design to ensure extensibility, portability, reproducibility, and longevity.

…

The Einstein Toolkit

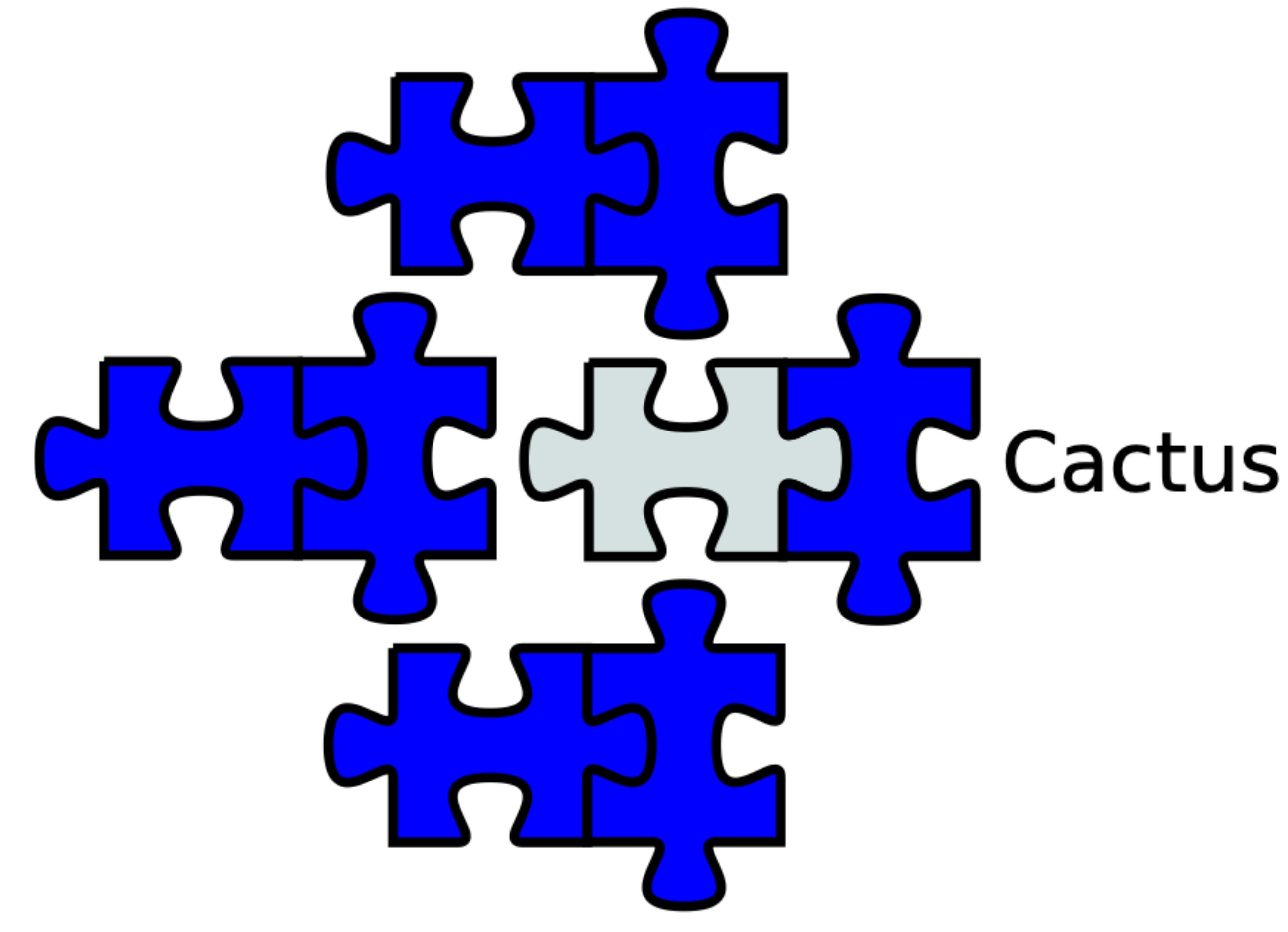
- Initially: some infrastructure, some application code.



Credit: Roland Haas

The Einstein Toolkit

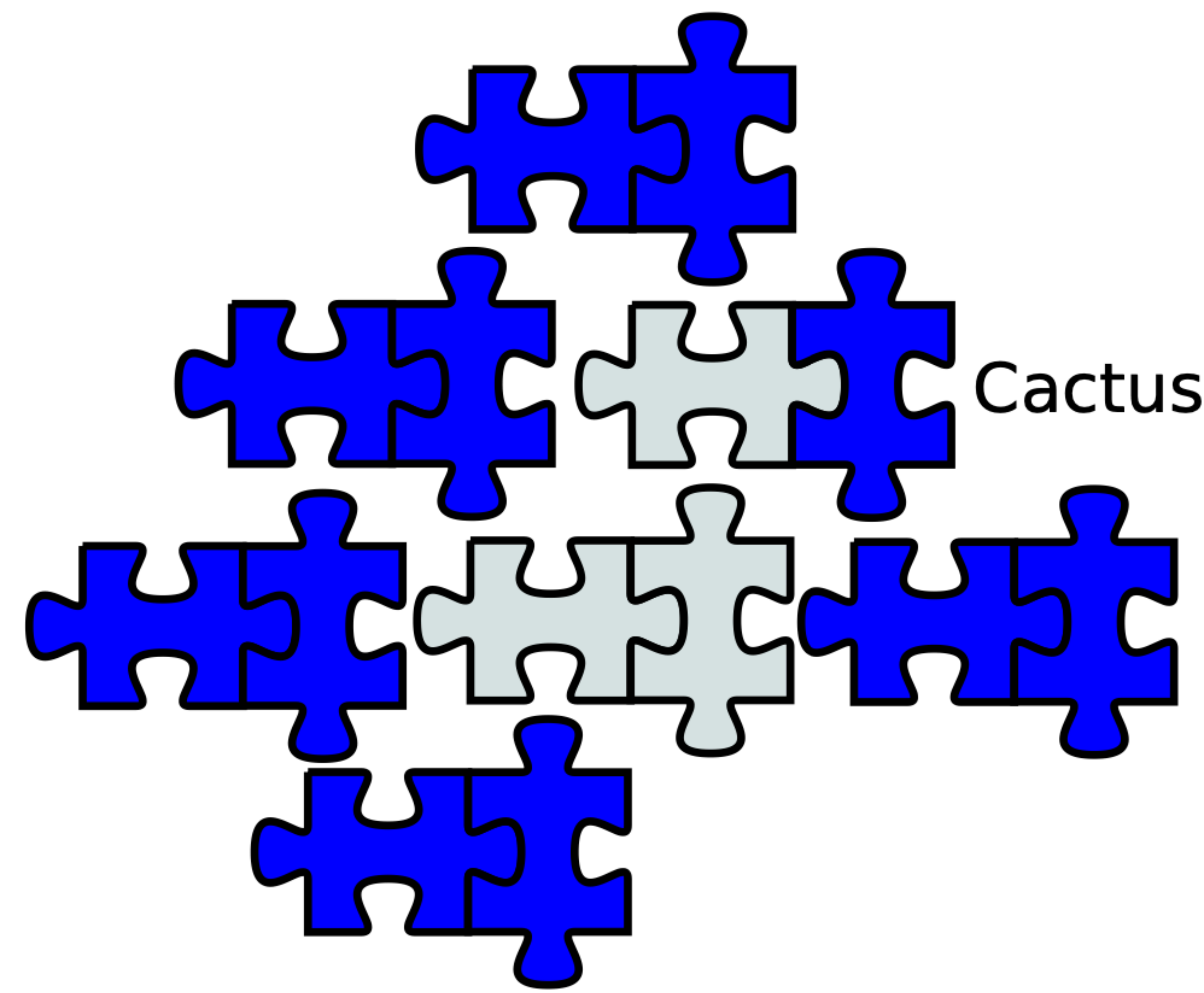
- Growing application suite.



Credit: Roland Haas

The Einstein Toolkit

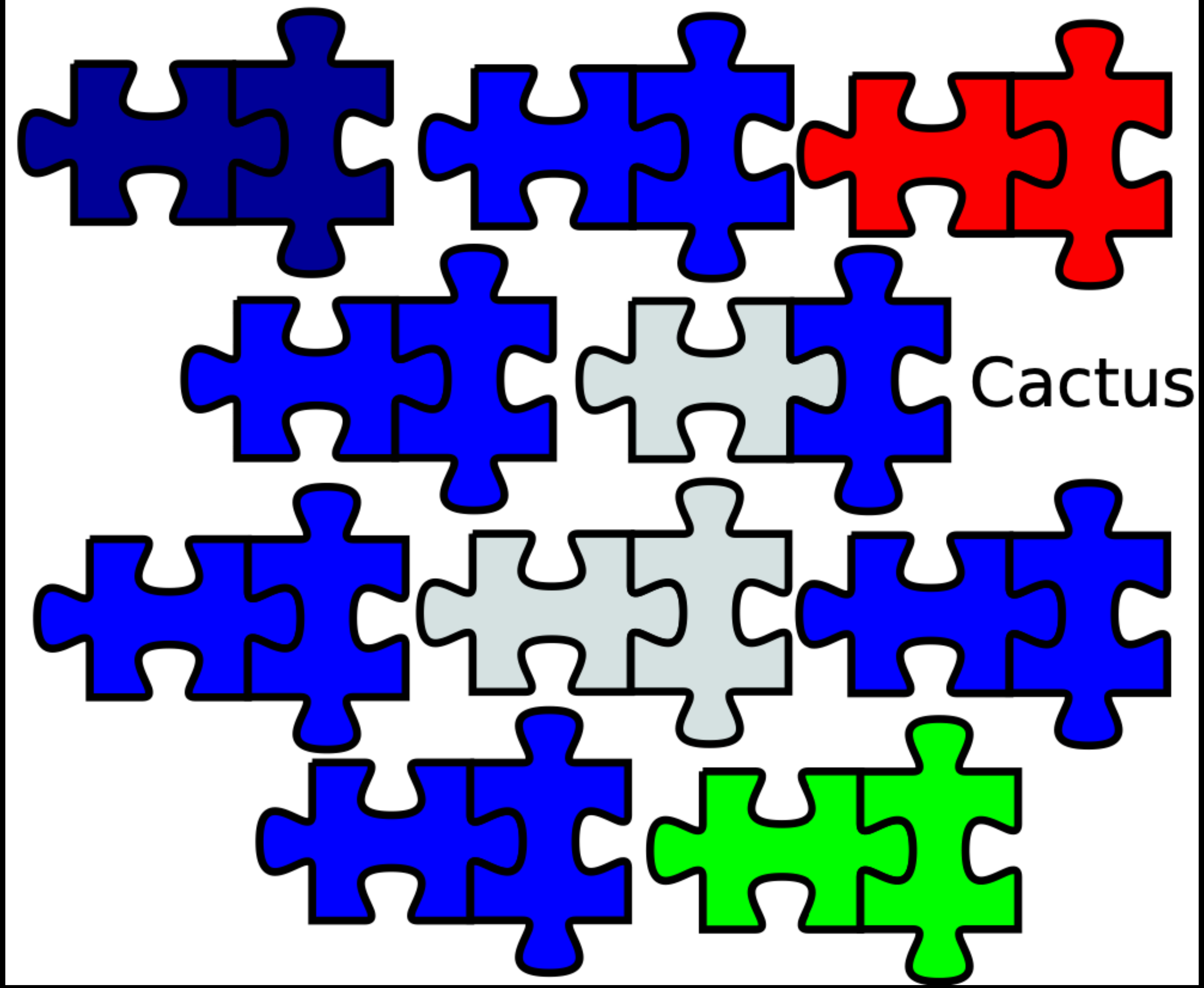
- Growing infrastructure “return”.



Credit: Roland Haas

The Einstein Toolkit

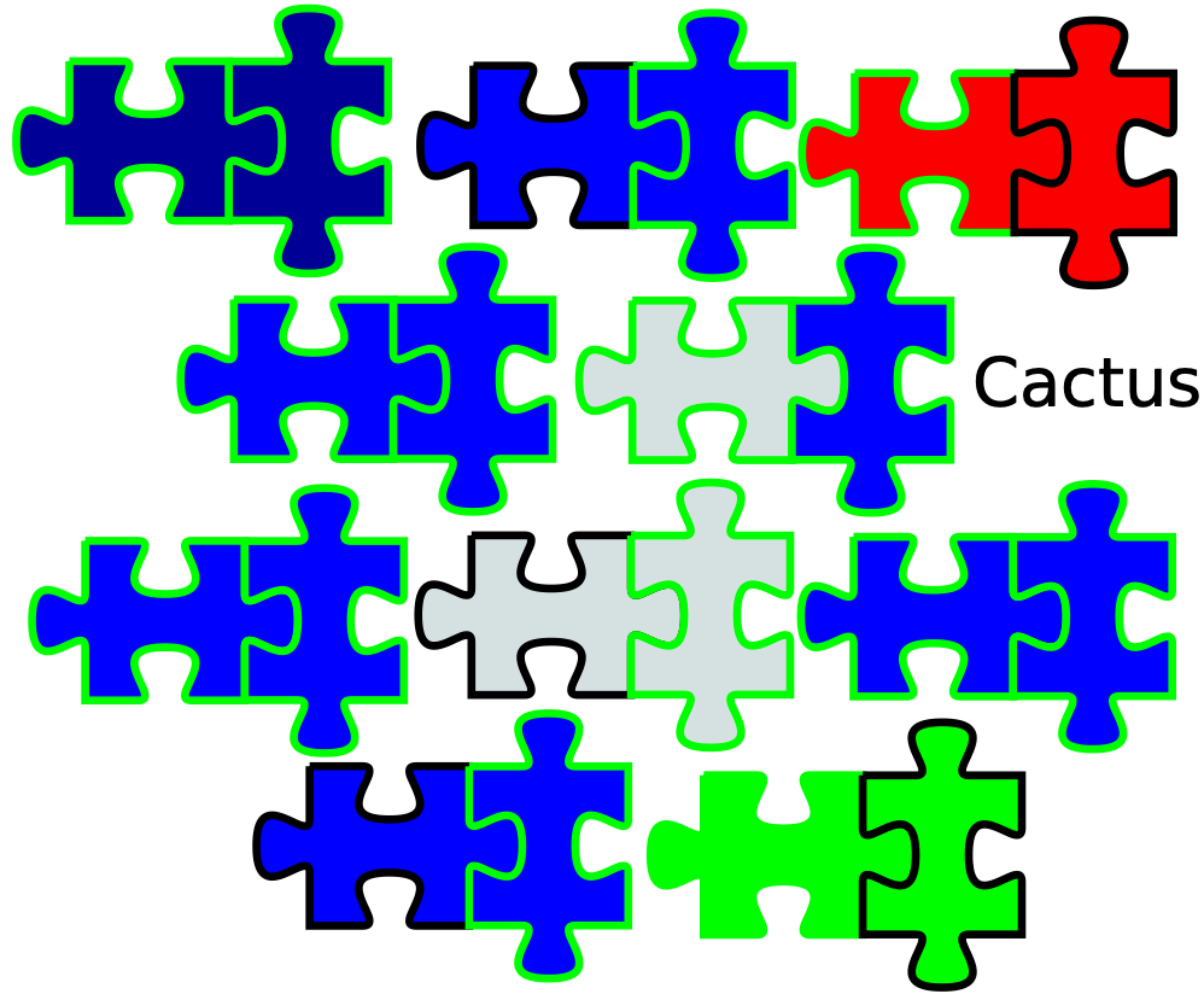
- Users from more research fields.



Credit: Roland Haas

The Einstein Toolkit

- Most modules are open-source, but not necessarily all.



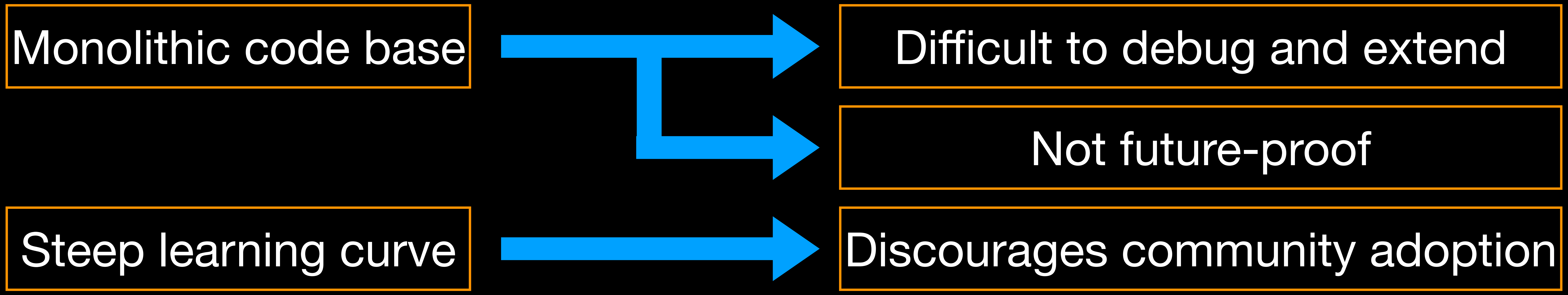
Credit: Roland Haas

The Einstein Toolkit

- Open, community-driven software development.
- Separation of physics software and computational infrastructure.
- Stable, extensible interfaces.
- Doing science » Running a simulation
  - Students need to know a lot about physics (meaningful initial conditions, numerical stability, accuracy/resolution, have patience, have curiosity, develop a “gut feeling” for what is right ...)
  - The Einstein Toolkit cannot give that, however: Open codes that are easy to use allow to concentrate on these things!

The General Relativistic Hydrodynamics Library

GRHayL: Motivation



Monolithic code base → Difficult to debug and extend

Monolithic code base → Not future-proof

Steep learning curve → Discourages community adoption

Credit: https://www.teepublic.com/tapestry/3141846-tangled-octopus

GRHayL: Streamlined New User Pipeline

New students must learn:

- Physics
- Mathematics
- Computer science
- Astronomy

Weakest link!

Solution:

- Small, modular code pieces
- No obscure language features
- **Extensive documentation**

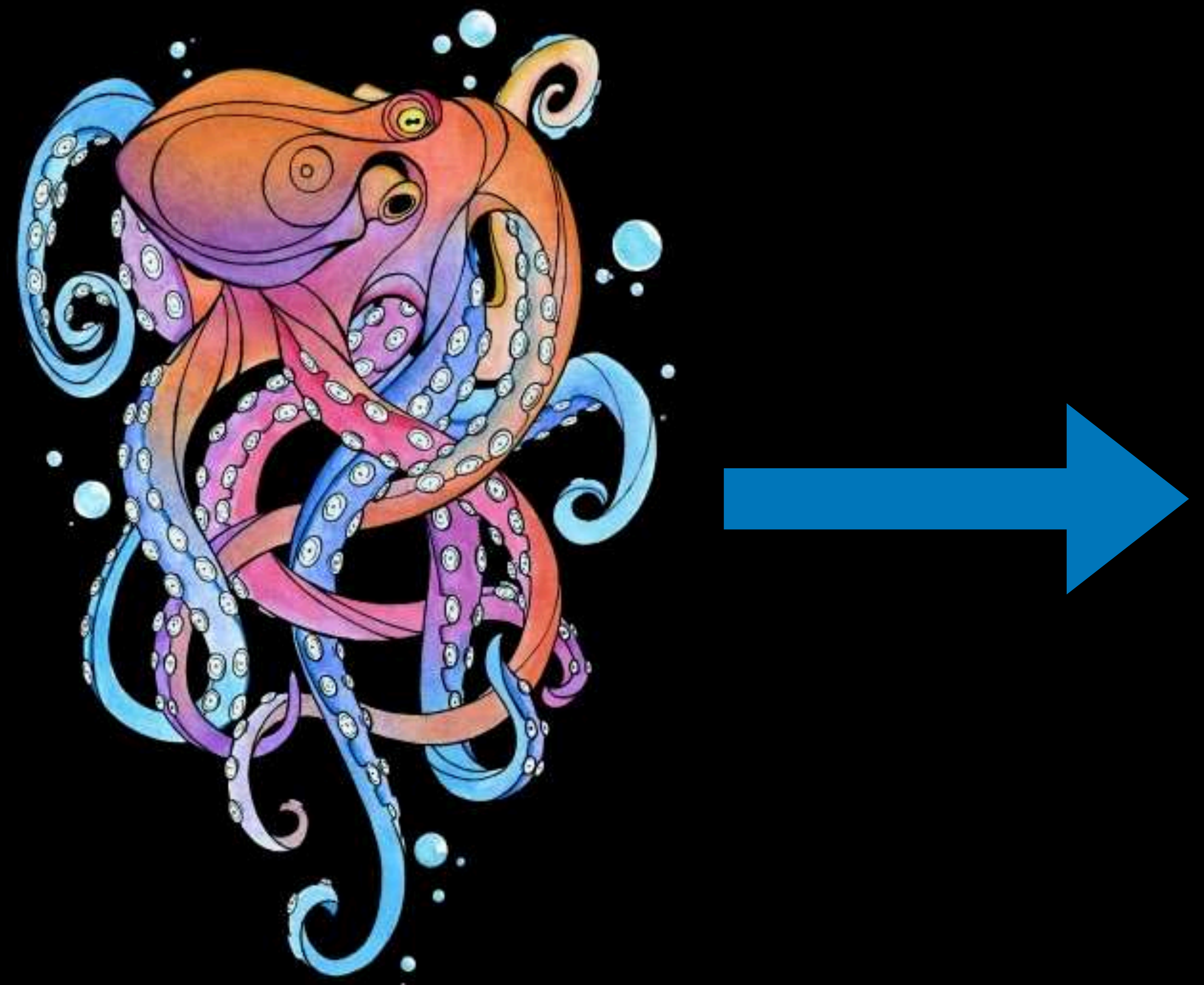
GRHayL: Streamlined New User Pipeline

Small, modular pieces → Self-container modules (**gems**)

No obscure language features → C with minimum dependencies

Good, extensive documentation → Wiki pages for all gems and functions

GRHayL: Modular & Infrastructure-Agnostic



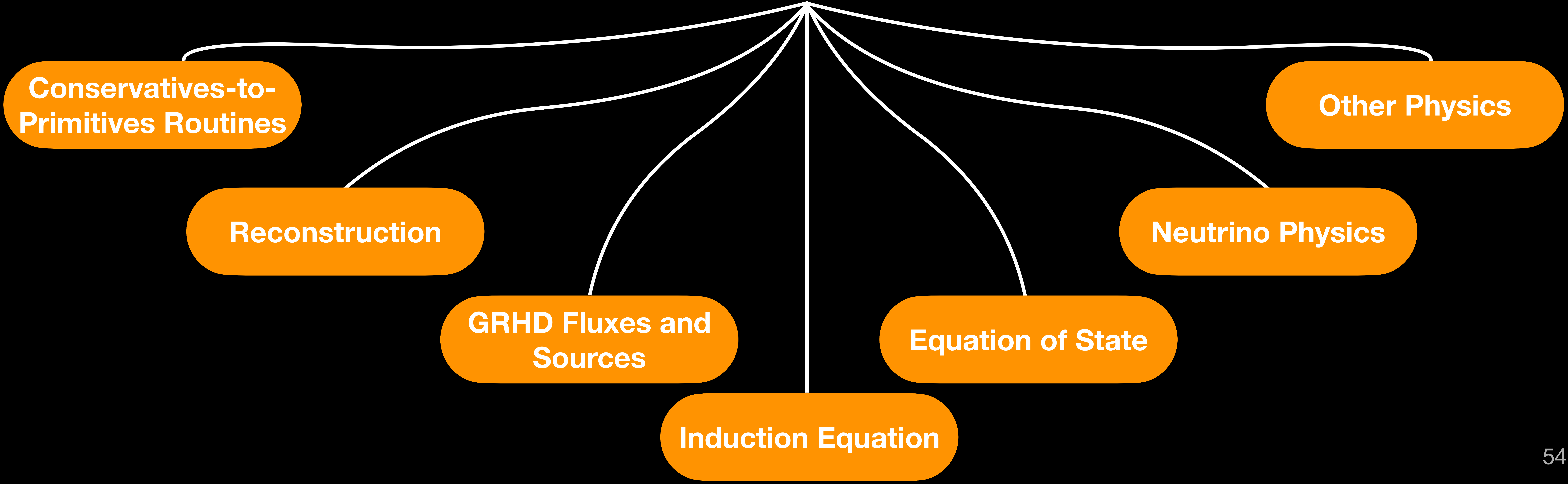
Credit: https://www.deviantart.com/sylviaritter/art/Cosmic-Cuttlefish-766515479

GRHayL: Modular & Infrastructure-Agnostic


© 2018 Sylvia Ritter

Core Code Infrastructure
Cactus/Einstein Toolkit
NRPy+/BlackHoles@Home
Your Infrastructure/Code

C structs pass data between infrastructure & gems



GRHayL: Continuous Integration & Code Coverage

Automated continuous integration (CI) with GitHub Actions

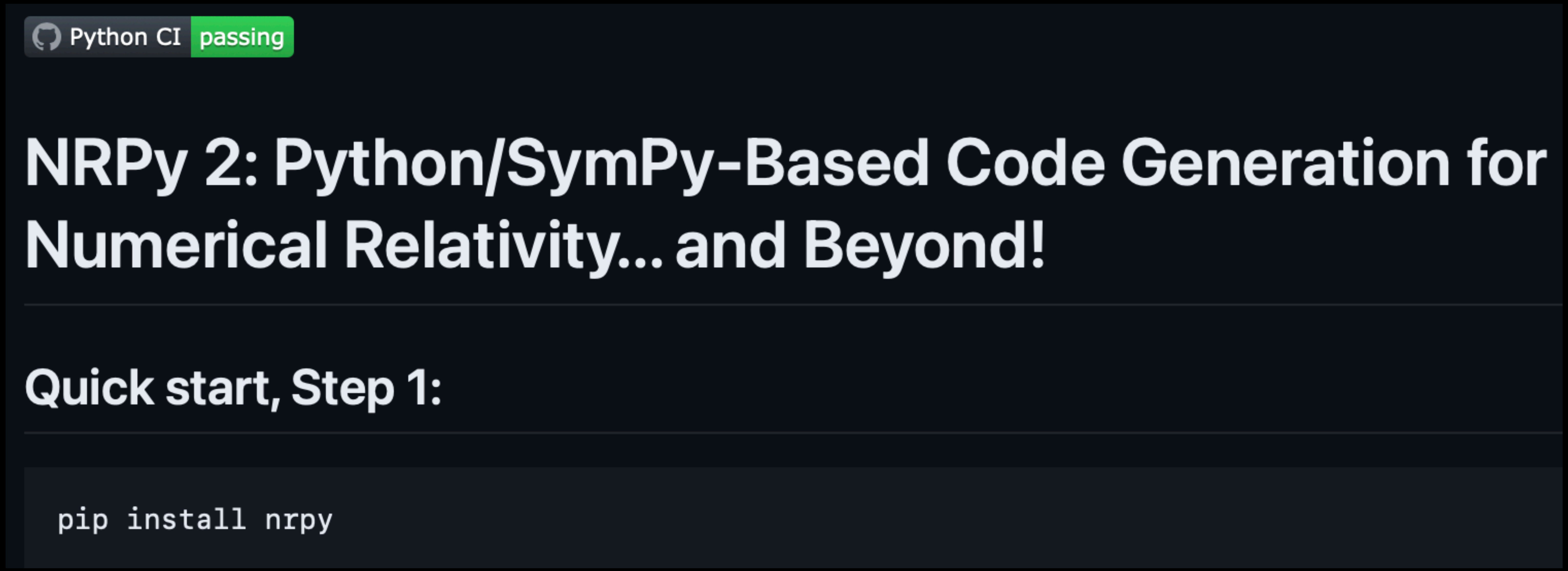
- Multiple OS/compiler combinations
- Uses trusted output to validate test output
- Core functions have individual unit tests

GRHayL

Credit: https://github.com/GRHayL/GRHayL

Python-based code generation for numerical relativity... and beyond!

NRPy+



NRPy 2: Python/SymPy-Based Code Generation for Numerical Relativity... and Beyond!

Quick start, Step 1:

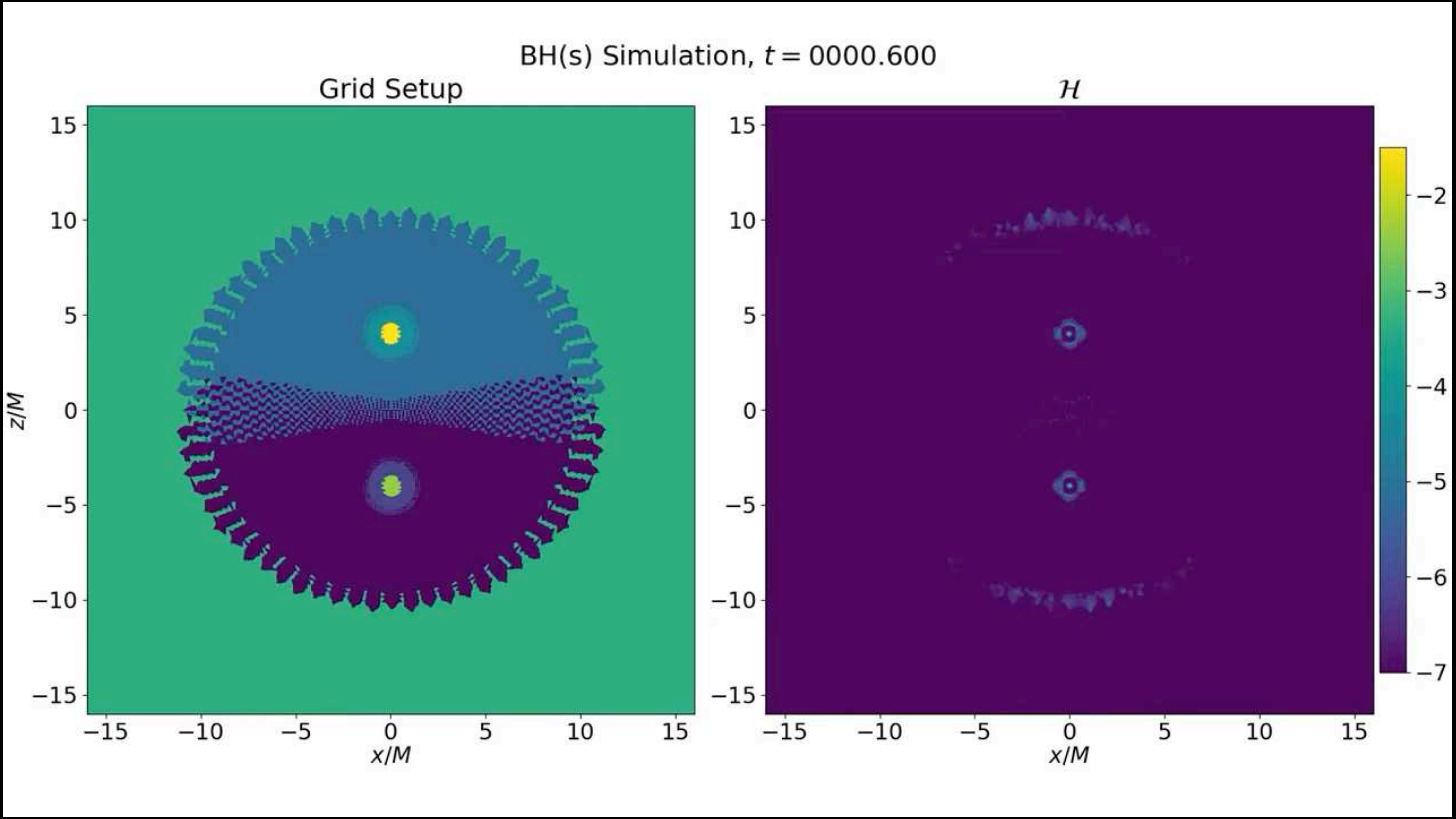
```
pip install nrpy
```

- Inspired by Kranc, but no Mathematica/Maple license required.
- Python/Sympy-based C/C++/Charm++/CUDA code generator.
  ★ Similar effort by LBNL/LLNL’s AMReX code generator.
- Arbitrary-order finite-differences for time-integration of PDEs.
- Has it’s own infrastructure called BlackHoles@Home.
- Supports Cartesian-like, cylindrical-like, and spherical-like coordinates.

BlackHoles@Home

Credit: Zach Etienne

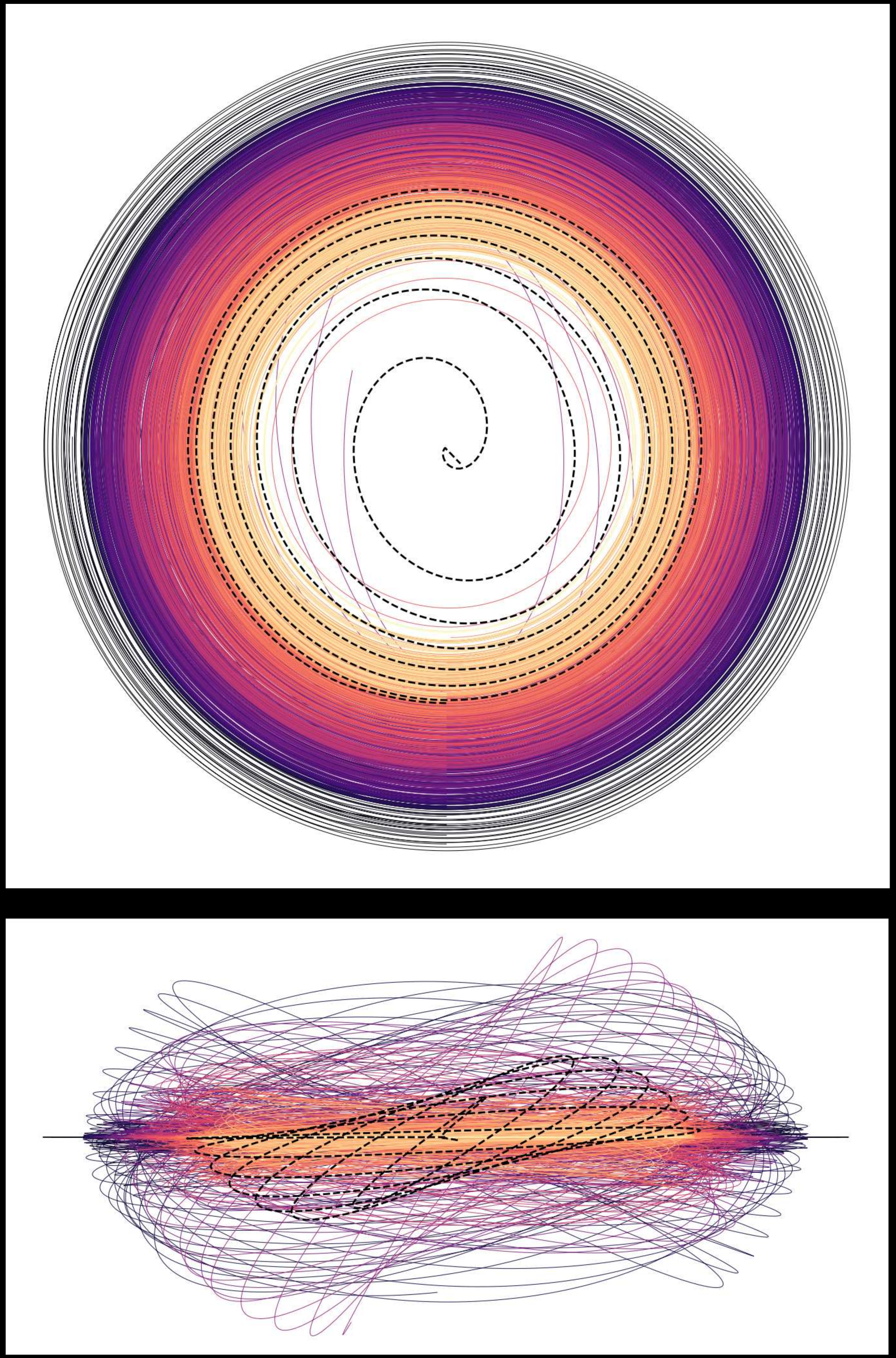
BlackHoles@Home



Credit: Zach Etienne

BlackHoles@Home

115 Binary Black Hole Simulations



Credit: Zach Etienne

- BlackHoles@Home: volunteer computing project.
- Goal: largest state-of-the-art BBH GW catalog.
- Challenge: must fit a ~90 Gb BBH simulation in a desktop computer.
- With NRPy+, this is possible! In fact, as a proof-of-principle, Zach was able to fit it in a cellphone!
- Efficient grids reduced memory required to ~3 Gb!
- Stay tuned for its launch!