Hamiltonian Monte Carlo sampler for compact binary sources

Marc Arène

Yann Bouffanais, Edward Porter

08/10/2018

GdR OG-ISIS

Outline

I. Bayesian inference for GW data analysis

II. Currently used algorithms

III. Hamiltonian Monte Carlo

IV. Tuning the algorithm

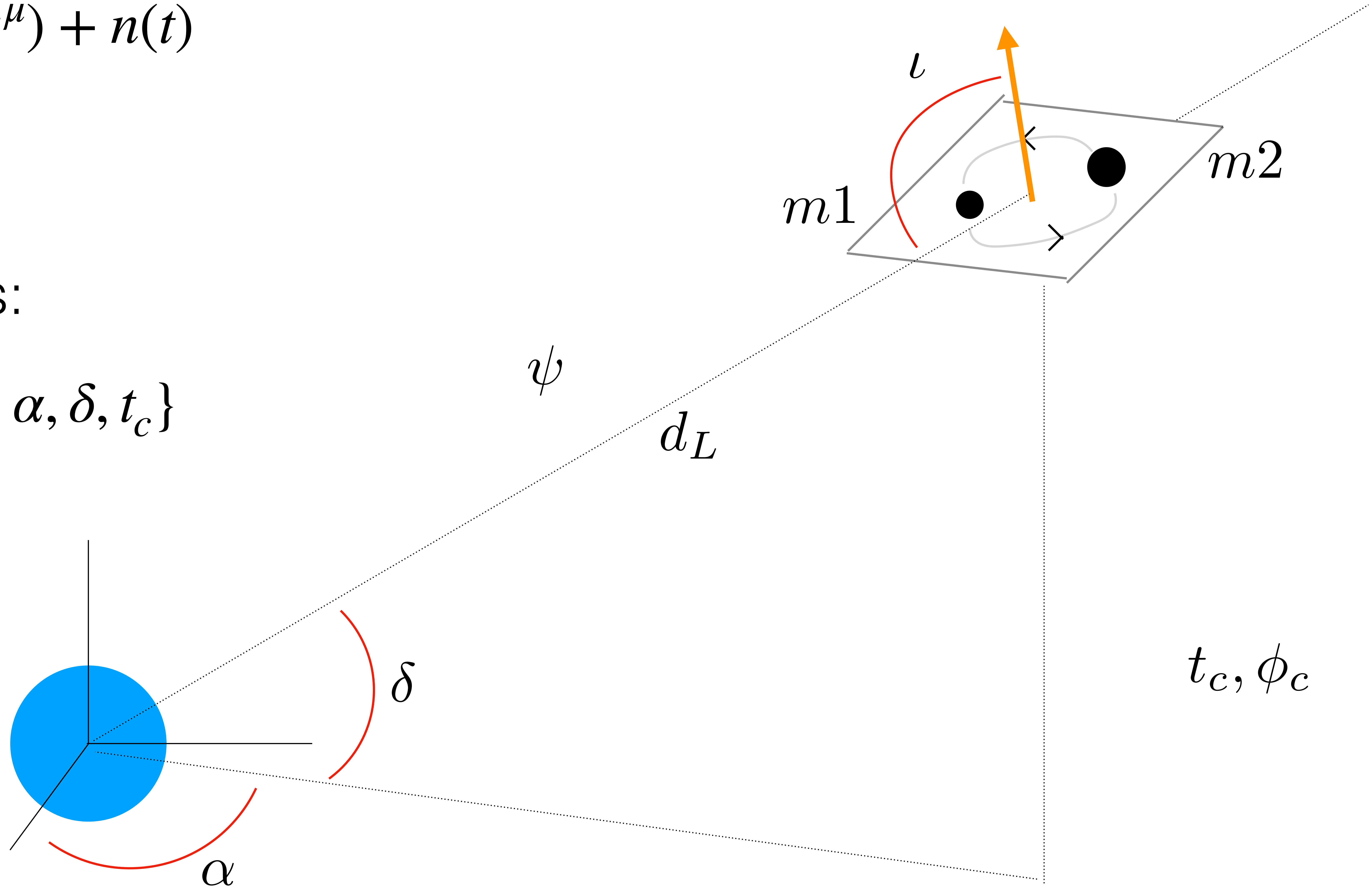
I. Bayesian inference for GW data analysis

- Data model: $s(t) = h(t; q^{\mu}) + n(t)$

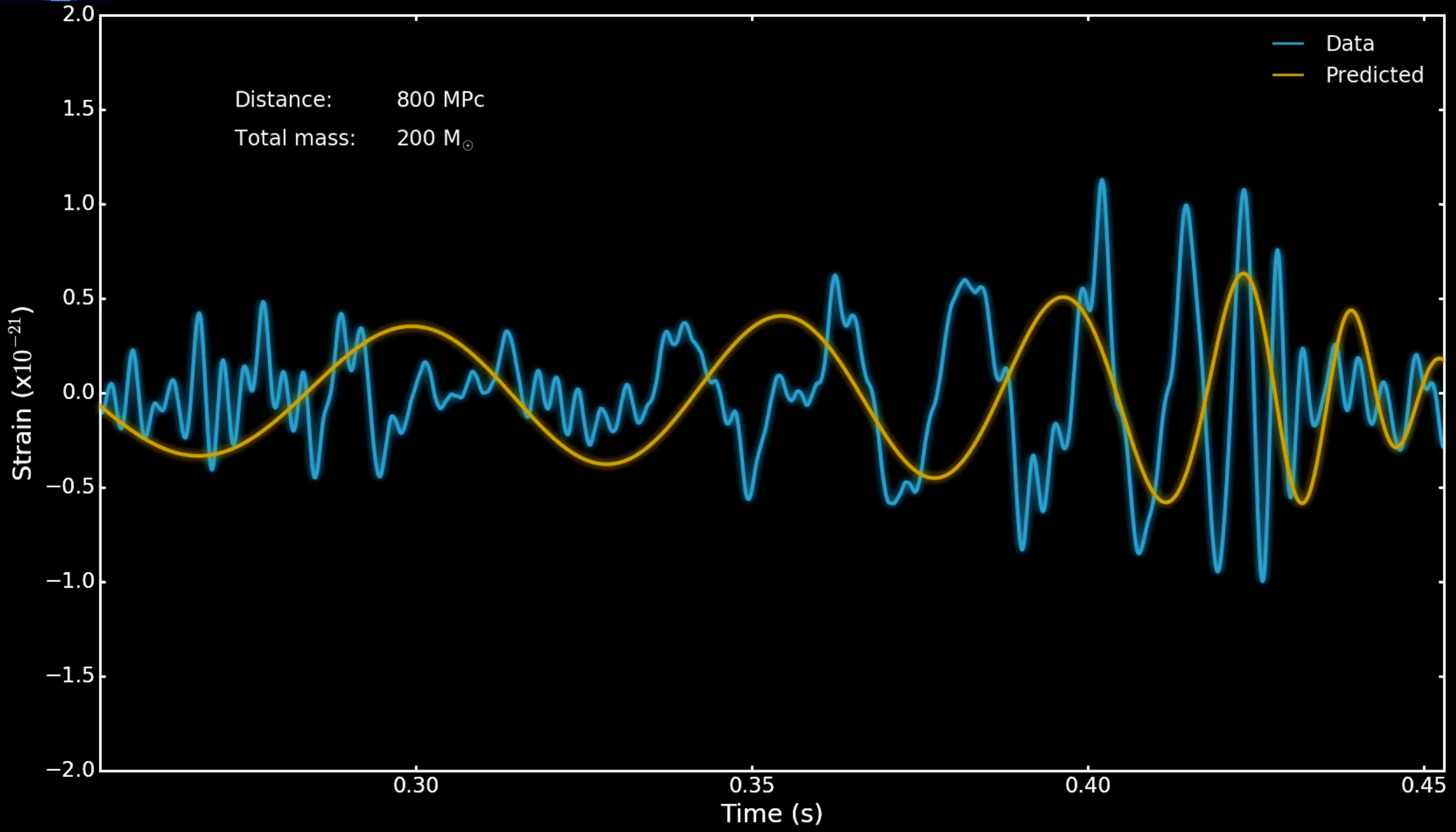
- Astrophysical parameters:

$$q^{\mu} = \{\iota, \phi_c, \psi, D_L, m_1, m_2, \alpha, \delta, t_c\}$$

- Noise model:
  - Gaussian + stationary



I. Bayesian inference for GW data analysis



Data & Best-fit Waveform: LIGO Open Science Center (losc.ligo.org); Prediction & Animation: C.North/M.Hannam (Cardiff University)

I. Bayesian inference for GW data analysis

- Bayesian framework

$$p(q^\mu \,|\, s, M) = \frac{\mathscr{L}(q^\mu)\pi(q^\mu)}{Z}$$

- Likelihood expression:

$$ln\mathscr{L}(q^\mu) = -\frac{1}{2}\sum_{d\in\{H,L,V\}}\left\langle \tilde{s}_d(f) - \tilde{h}_d(f; q^\mu) \,|\, \tilde{s}_d(f) - \tilde{h}_d(f; q^\mu)\right\rangle$$

- Noise weighted inner product:

$$\langle h \,|\, g\rangle = 2\int_0^\infty \frac{\tilde{h}(f)\tilde{g}^*(f) + \tilde{h}^*(f)\tilde{g}(f)}{S_n(f)}df$$

II. Currently used algorithms

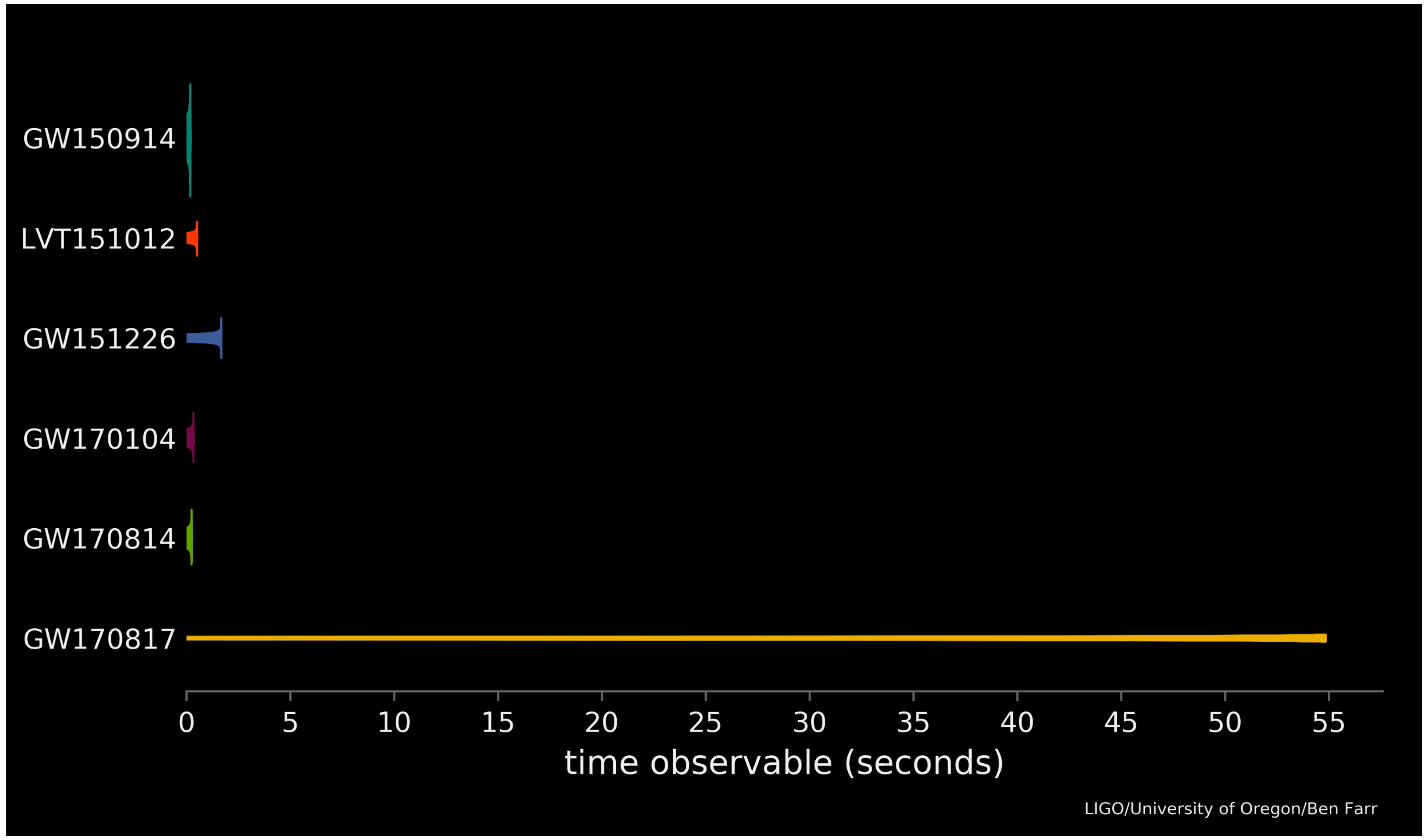
- Currently parameter estimation analysis is carried on using **parallel tempered MCMC** and **Nested sampling** algorithms

- Both belong to a family of random walk samplers

- Duration of those analysis:

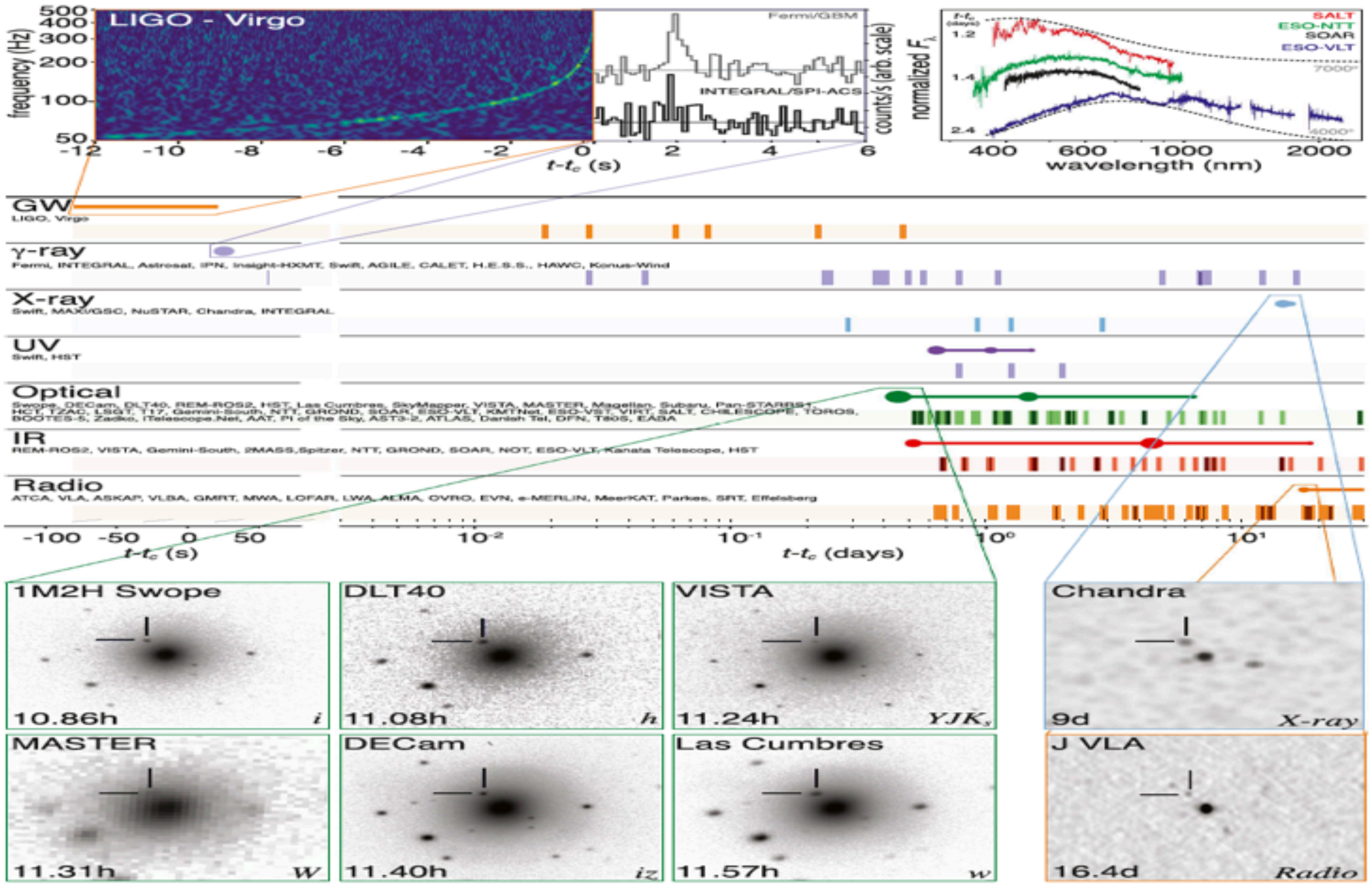
  - BBH(signal<1sec): few days

  - BNS(signal ~ 100sec): few weeks



- Likelihood is evaluated ~ 10^7 times (for MCMC) to get an effective sample size of 10^4 uncorrelated samples.

II. Electromagnetic follow-up



II. Currently used algorithms

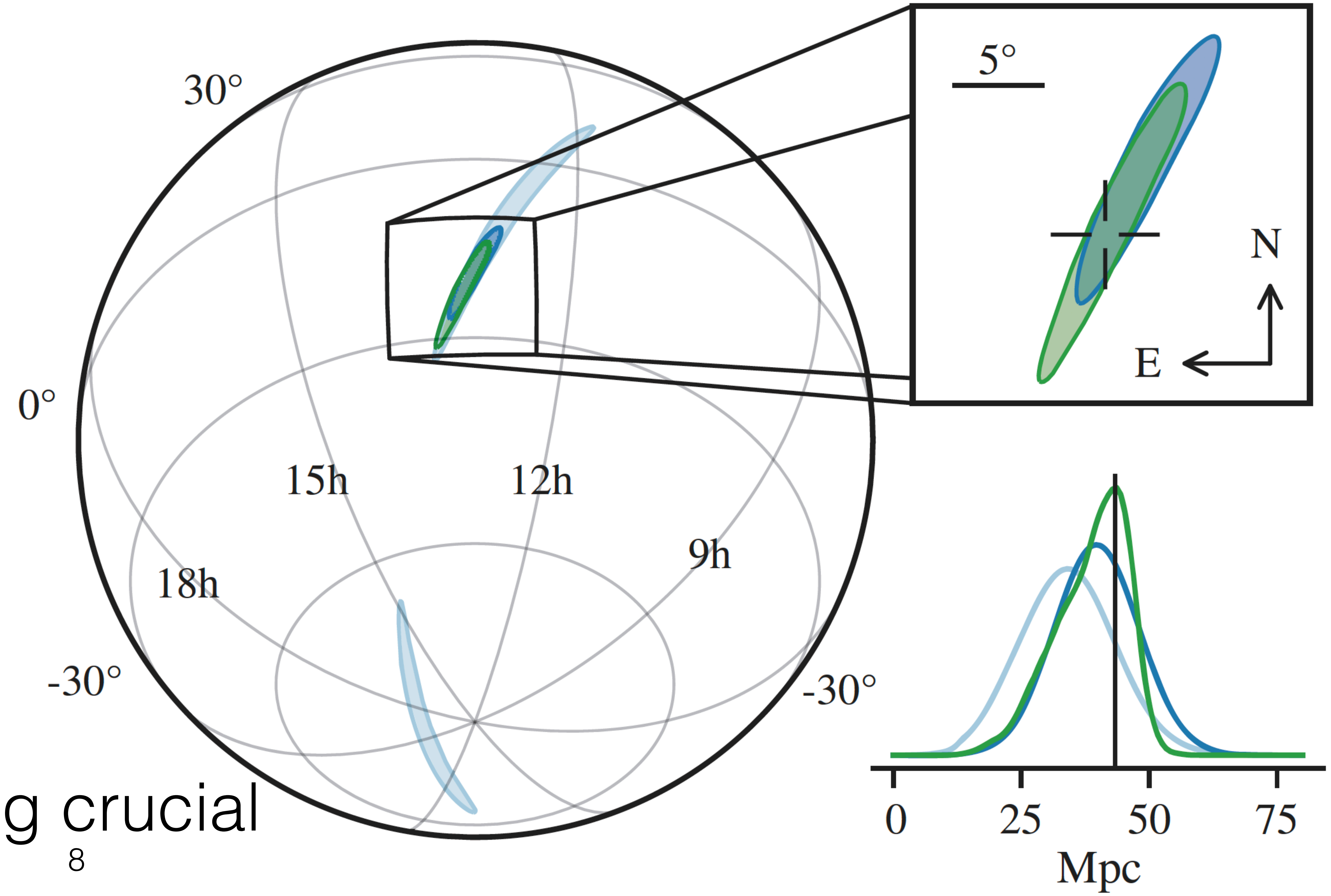
- Rates for O3:

(C. Pankow, 2018)

BBH: ~ 1-2 / week **BNS**: 1 / month **NSBH**: 1 / year

(Physical Review Letters, https://doi.org/10.1103/PhysRevLett.119.161101)

- Need for rapid sky maps for electromagnetic follow-up: <1h



- Low latency algorithms are becoming crucial

III. Hamiltonian Monte Carlo

- Non-random walk sampler
- Instead of proposing random jumps in parameter space, we use Hamiltonian dynamics to explore the posterior distribution
- Has been shown to be **D** times more efficient than MCMC samplers

  (A. Hajian, Phys. Rev. D75, 083525 (2007), astro-/0608679)

  D = dimensionality of parameter space
- Is better than MCMC at exploring multi-modal distributions

III. Hamiltonian Monte Carlo

- Define Hamiltonian

$$\mathscr{H}(q^\mu, p^\mu) = \mathscr{U}(q^\mu) + \mathscr{K}(p^\mu)$$
$$= -\, ln[\mathscr{L}(q^\mu)\pi(q^\mu)] + \frac{1}{2} M^{-1}_{\mu\nu} p^\mu p^\nu$$

- Canonical distribution

$$\Pi(q^\mu, p^\mu) \propto e^{-\mathscr{H}(q^\mu, p^\mu)}$$
$$\propto e^{-\mathscr{U}(q^\mu)}\ e^{-\mathscr{K}(p^\mu)}$$
$$\Pi(q^\mu, p^\mu) \propto \mathscr{L}(q^\mu)\pi(q^\mu)\ e^{-\frac{(p^\mu)^2}{2m^\mu}}$$

- If the momenta are drawn from a normal distribution, the marginal distribution for q^μ gives a sample set that asymptotically comes from the posterior distribution

III. Hamiltonian Monte Carlo

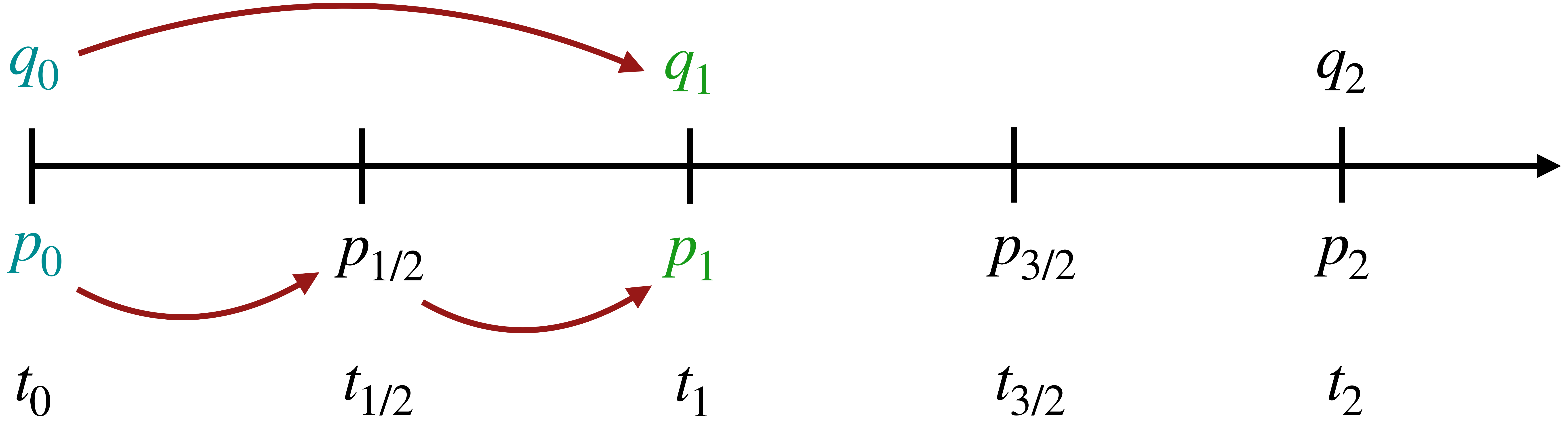
- Trajectories in phase space are found solving Hamilton's equations

$$\begin{cases} \dfrac{dp^\mu}{dt} = -\dfrac{\partial \mathscr{U}}{\partial q^\mu} \\ \dfrac{dq^\mu}{dt} = \dfrac{\partial \mathscr{K}}{\partial p^\mu} \end{cases} \longrightarrow \begin{cases} \dfrac{dp^\mu}{dt} = \dfrac{\partial ln[\mathscr{L}(q^\mu)\pi(q^\mu)]}{\partial q^\mu} \\ \dfrac{dq^\mu}{dt} = M^{-1}_{\mu\nu} p^\nu = (m^\mu)^{-1} p^\mu \end{cases}$$

- Can't be solved analytically
- We need an integrator that
  - Conserves the Hamiltonian
  - Is time reversible
  - Conserves phase space volume

Leap frog algorithm

III. Hamiltonian Monte Carlo



1/2 step in momenta

$$p^{\mu}(t+\frac{\epsilon}{2}) = p^{\mu}(t) + \frac{\epsilon}{2}\frac{\partial ln[\mathscr{L}(q^{\mu})\pi(q^{\mu})]}{\partial q^{\mu}}\Big|_{q^{\mu}(t)}$$

Full step in position

$$q^{\mu}(t+\epsilon) = q^{\mu}(t) + \frac{\epsilon}{m^{\mu}}p^{\mu}(t+\frac{\epsilon}{2})$$

1/2 step in momenta

$$p^{\mu}(t+\epsilon) = p^{\mu}(t+\frac{\epsilon}{2}) + \frac{\epsilon}{2}\frac{\partial ln[\mathscr{L}(q^{\mu})\pi(q^{\mu})]}{\partial q^{\mu}}\Big|_{q^{\mu}(t+\epsilon)}$$

III. Hamiltonian Monte Carlo

- Leapfrog algorithm

$$ln[\mathscr{L}(q^\mu)] \sim \langle s - h(q^\mu) \,|\, s - h(q^\mu) \rangle$$

Main computational bottleneck

$$\begin{cases} p^\mu(t + \frac{\epsilon}{2}) = p^\mu(t) + \frac{\epsilon}{2} \frac{\partial ln[\mathscr{L}(q^\mu)\pi(q^\mu)]}{\partial q^\mu}\Big|_{q^\mu(t)} \\ q^\mu(t + \epsilon) = q^\mu(t) + \frac{\epsilon}{m^\mu} p^\mu(t + \frac{\epsilon}{2}) \\ p^\mu(t + \epsilon) = p^\mu(t + \frac{\epsilon}{2}) + \frac{\epsilon}{2} \frac{\partial ln[\mathscr{L}(q^\mu)\pi(q^\mu)]}{\partial q^\mu}\Big|_{q^\mu(t+\epsilon)} \end{cases}$$

Each step along the trajectory: 2D waveform generations required

III. Hamiltonian Monte Carlo

- Leapfrog algorithm

$$
\begin{cases}
p^\mu(t+\frac{\epsilon}{2}) = p^\mu(t) + \frac{\epsilon}{2}\frac{\partial ln[\mathscr{L}(q^\mu)\pi(q^\mu)]}{\partial q^\mu}\Big|_{q^\mu(t)} \\
q^\mu(t+\epsilon) = q^\mu(t) + \frac{\epsilon}{m^\mu}p^\mu(t+\frac{\epsilon}{2}) \\
p^\mu(t+\epsilon) = p^\mu(t+\frac{\epsilon}{2}) + \frac{\epsilon}{2}\frac{\partial ln[\mathscr{L}(q^\mu)\pi(q^\mu)]}{\partial q^\mu}\Big|_{q^\mu(t+\epsilon)}
\end{cases}
$$

Time estimation to get 10^6 samples with numerical gradients:

10^6 x 200 x 9 x 2 x 1ms ~ 1 month !

- 10^6: Nb of trajectories
- 200: Nb of leapfrog steps per trajectory
- 9: Nb of parameters
- 2: Central differencing
- 1ms: One waveform generation

IV. Tuning the algorithm

- "Out of the box" leapfrog

$$
\begin{cases}
p^\mu(t+\frac{\epsilon}{2}) = p^\mu(t) + \frac{\epsilon}{2}\frac{\partial ln[\mathscr{L}(q^\mu)\pi(q^\mu)]}{\partial q^\mu}\Big|_{q^\mu(t)} \\
q^\mu(t+\epsilon) = q^\mu(t) + \frac{\epsilon}{m^\mu}p^\mu(t+\frac{\epsilon}{2}) \\
p^\mu(t+\epsilon) = p^\mu(t+\frac{\epsilon}{2}) + \frac{\epsilon}{2}\frac{\partial ln[\mathscr{L}(q^\mu)\pi(q^\mu)]}{\partial q^\mu}\Big|_{q^\mu(t+\epsilon)}
\end{cases}
$$

$$\epsilon^\mu = s^\mu \epsilon$$

$$s^\mu = {m_\mu}^{-1/2}$$

IV. Tuning the algorithm

- Scaled leapfrog

$$\begin{cases} \tilde{p}^{\mu}\left(t+\frac{\epsilon^{\mu}}{2}\right) = \tilde{p}^{\mu}(t) + \frac{\epsilon^{\mu}}{2}\frac{\partial ln[\mathscr{L}(q^{\mu})\pi(q^{\mu})]}{\partial q^{\mu}}\Big|_{q^{\mu}(t)} \\ q^{\mu}(t+\epsilon^{\mu}) = q^{\mu}(t) + \epsilon^{\mu}\tilde{p}^{\mu}\left(t+\frac{\epsilon^{\mu}}{2}\right) \\ \tilde{p}^{\mu}(t+\epsilon^{\mu}) = \tilde{p}^{\mu}(t+\frac{\epsilon^{\mu}}{2}) + \frac{\epsilon^{\mu}}{2}\frac{\partial ln[\mathscr{L}(q^{\mu})\pi(q^{\mu})]}{\partial q^{\mu}}\Big|_{q^{\mu}(t+\epsilon^{\mu})} \end{cases}$$

$$\epsilon^{\mu} = s^{\mu}\epsilon$$

$$s^{\mu} = m_{\mu}{}^{-1/2}$$

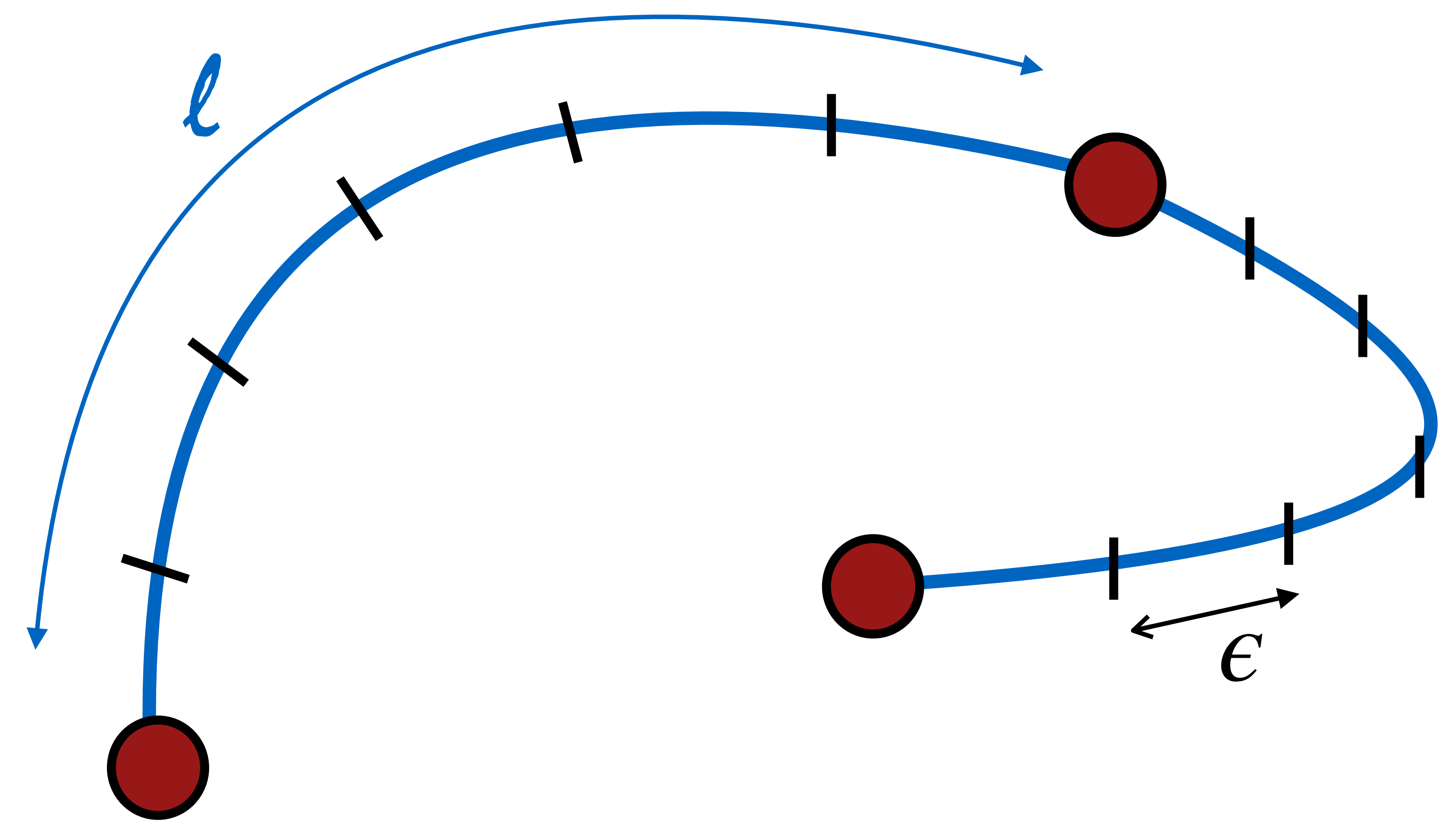
$$\tilde{p}^{\mu} = s^{\mu}p^{\mu}$$

IV. Tuning the algorithm

1. Free parameters

- $\epsilon \sim \mathcal{N}(5 \cdot 10^{-3},\ 1.5 \cdot 10^{-3})$

  $l \sim \mathcal{U}(50,\ 100)$



- Mass matrix: $M_{\mu\nu} = \begin{cases} m_\mu & if \quad \mu = \nu \\ 0 & if \quad \mu \neq \nu \end{cases}$

  Derived from a corrected Fisher Information Matrix

IV. Tuning the algorithm

2. Cubic fit approximation

- At each step of the leapfrog algorithm, replace numerical gradient by a cubic fit approximation (Carré and Porter, 2009)

$$f(q^\mu) = \sum_{i=1}^{D} a_i q^i + \sum_{j=1}^{D} \sum_{k=j}^{D} a_{jk} q^j q^k + \sum_{l=1}^{D} \sum_{v=l}^{D} \sum_{w=v}^{D} a_{lvw} q^l q^v q^w$$

- 220 coefficients per dimension

- How to derive the cubic fit approximation ?

IV. Tuning the algorithm

2. Cubic fit approximation

- Phase I **3-5 hours**
  - Run ~1000 numerical gradient trajectories
  - If trajectory is accepted, keep q^μ and $\frac{\partial ln[\mathscr{L}(q^\mu)\pi(q^\mu)]}{\partial q^\mu}$ at each leapfrog step
- Phase II **~ 10 min**
  - Derive the 220 coefficients of the cubic fit for each dimension, using a QR decomposition
- Phase III **20-25 hours**
  - Replace numerical gradients with approximate gradients from the cubic fit

x 3000 faster

IV. Tuning the algorithm

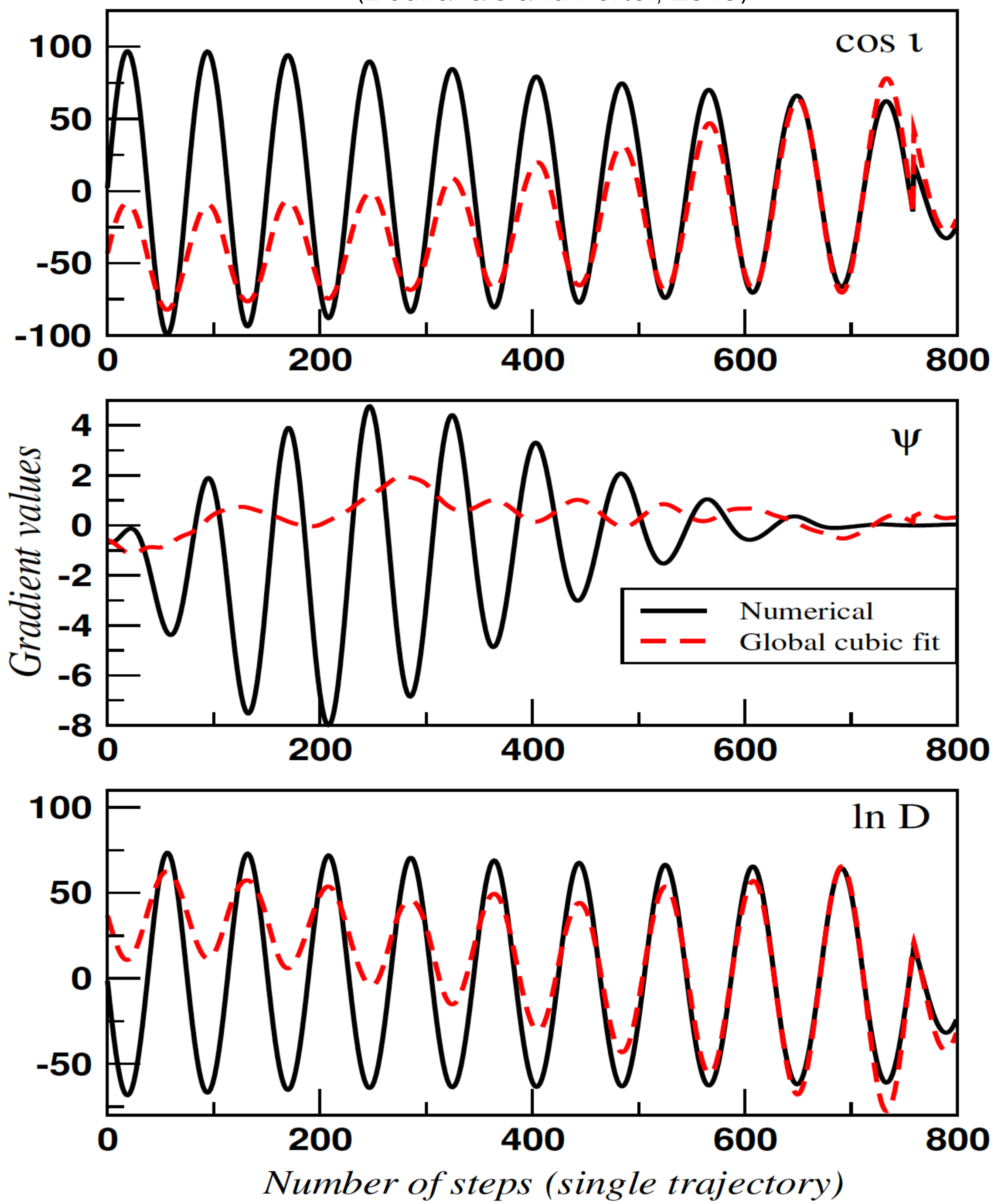
2. Cubic fit approximation

- Problem for BNS systems !
- Trajectories get rejected as soon as Phase III starts
- Cubic fit does not work for 3 parameters:

$$\{cos\iota, \psi, lnD_L\}$$

- These 3 parameters are the only bi-modal

(Bouffanais and Porter, 2018)



IV. Tuning the algorithm

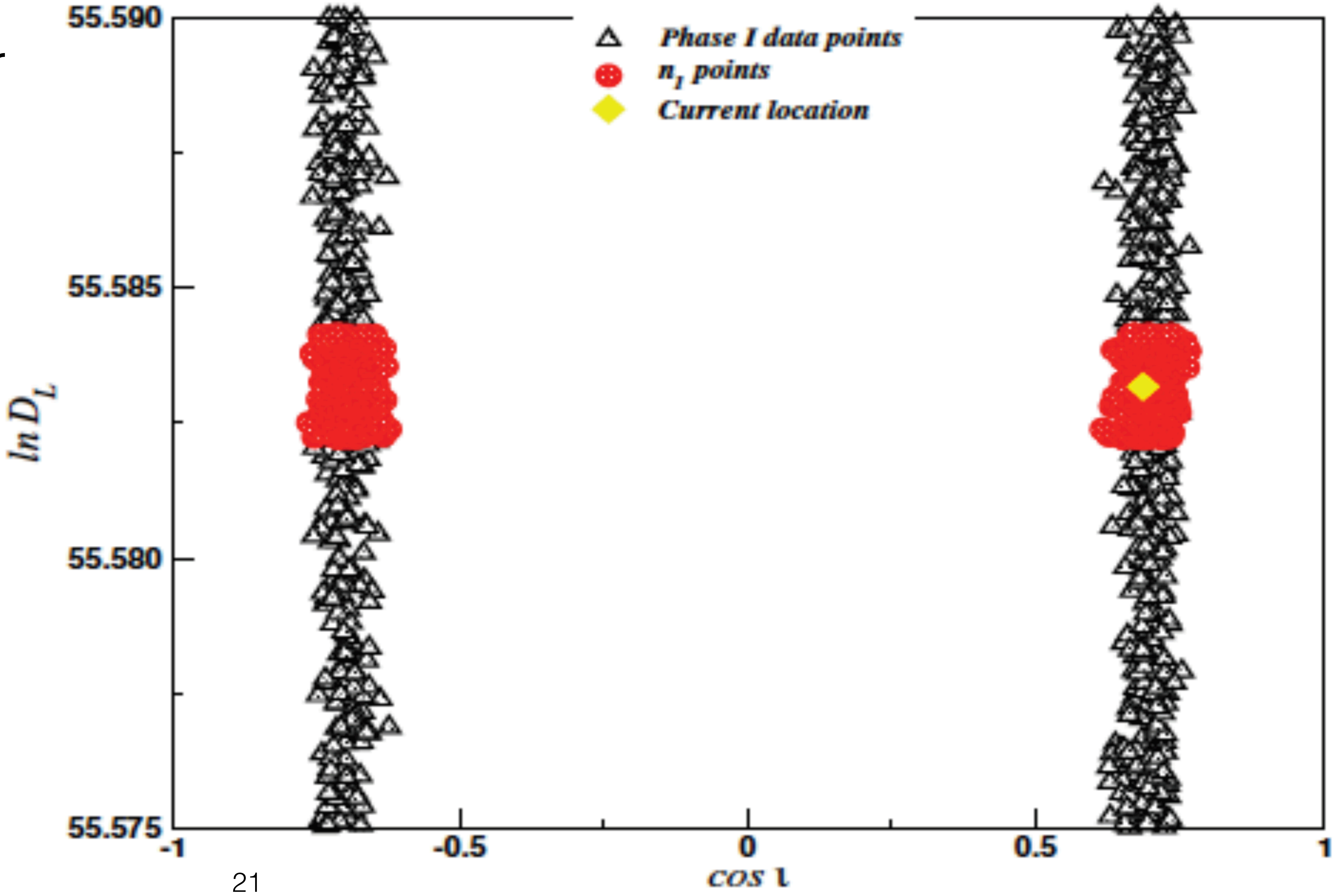
3. Ordered Look Up Tables

- So we decided to use Ordered Look-Up Tables for each of the 3 troublesome parameters

- Select the **n1** closest points to the point of interest

- Problem !!

(Bouffanais and Porter, 2018)



IV. Tuning the algorithm

3. Ordered Look Up Tables

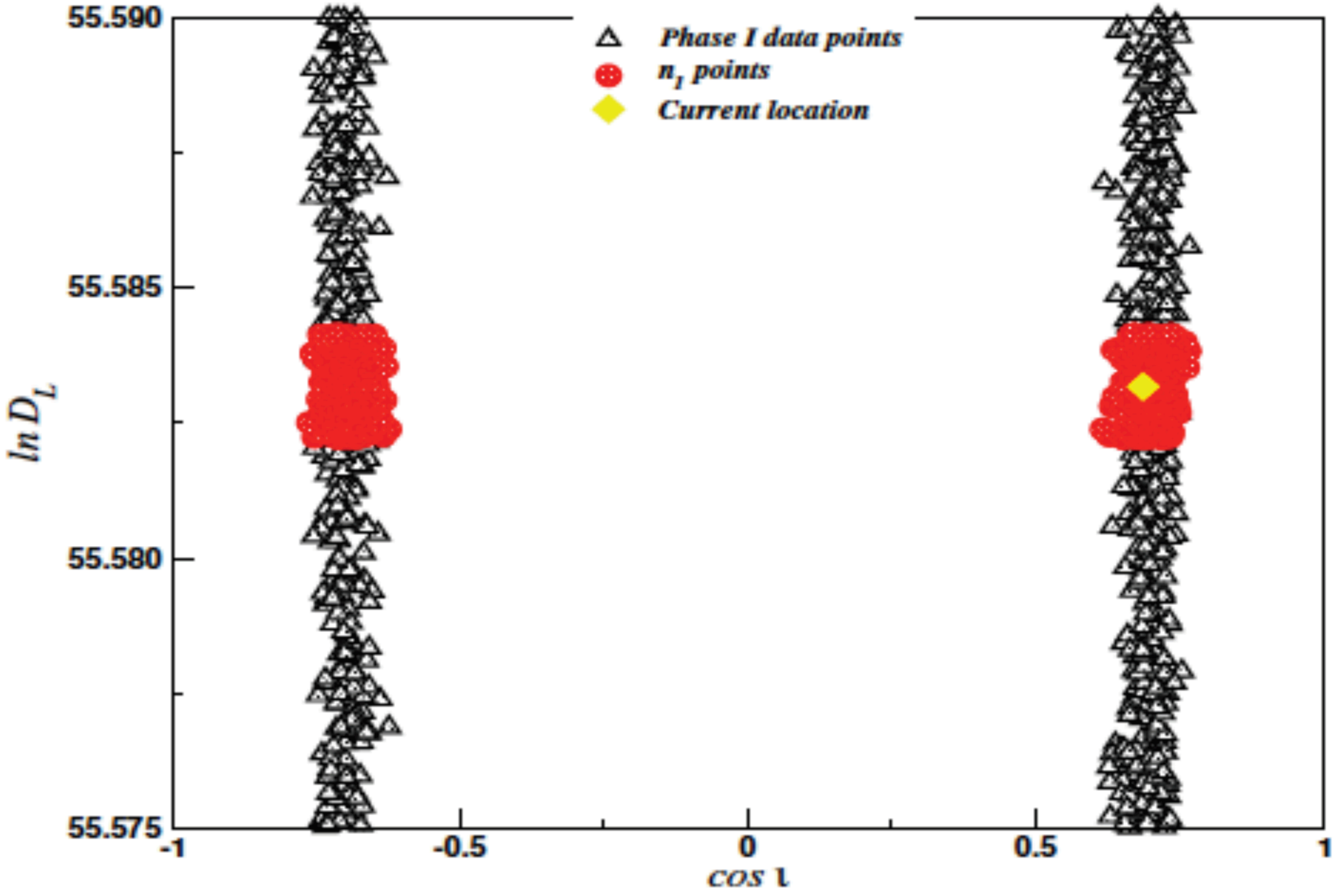
- Solution is to use a scaled Euclidian distance to sub-select the **n2** points that are truly local

- Usually

  - n1 = 2000

  - n2 = 200

(Bouffanais and Porter, 2018)

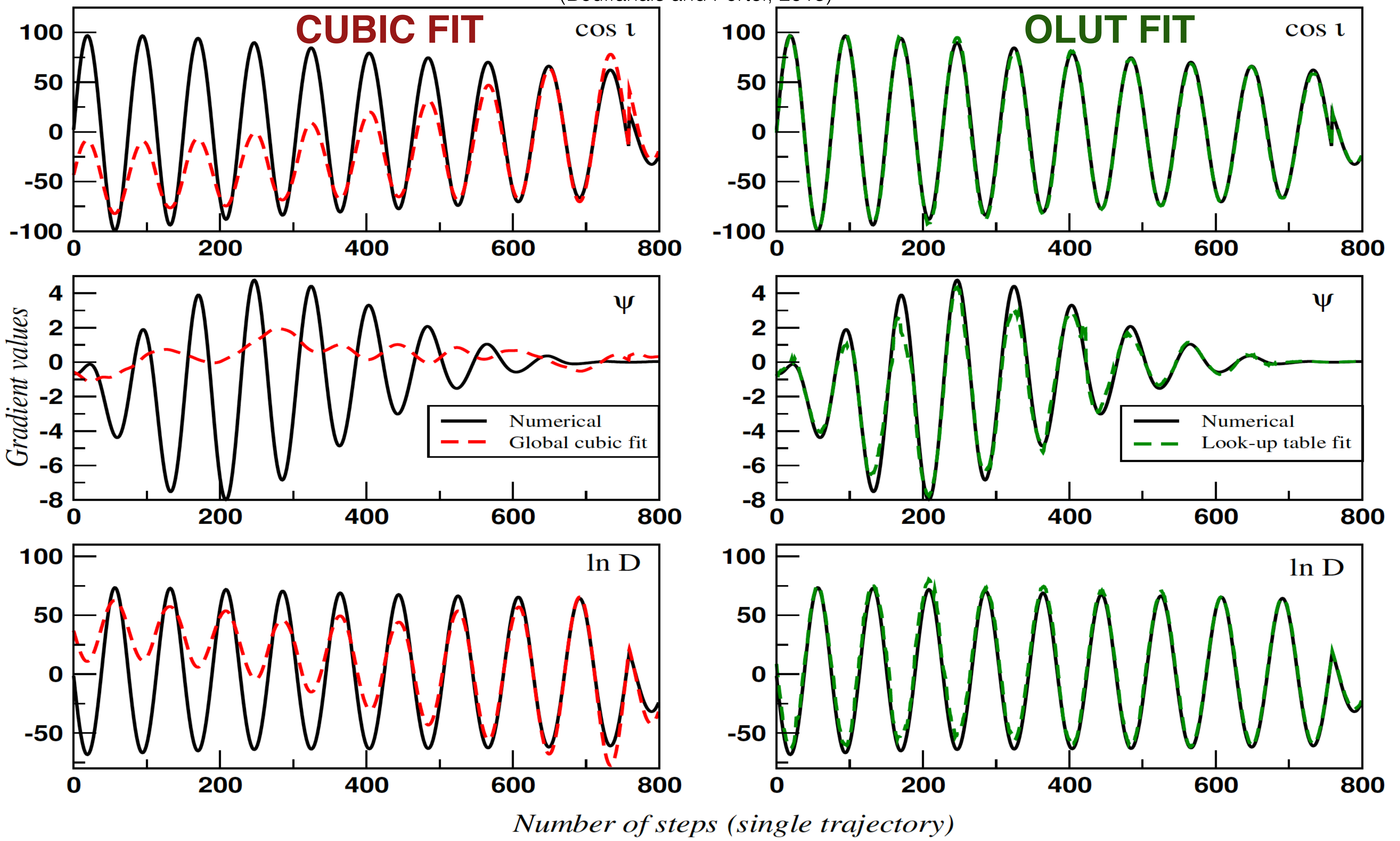


IV. Tuning the algorithm

3. Ordered Look Up Tables

(Bouffanais and Porter, 2018)

BAD ESTIMATE

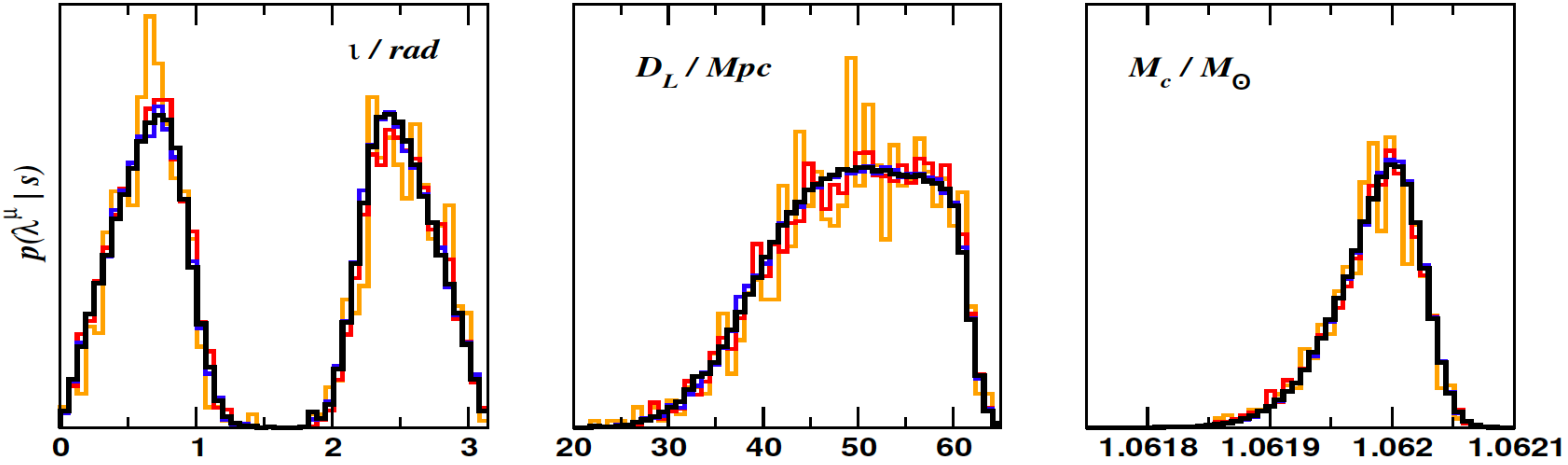


GOOD ESTIMATE

IV. Tuning the algorithm

4. Convergence analysis

Convergence after **10^3, 10^4, 10^5, 10^6** trajectories



(Bouffanais and Porter, 2018)

IV. Tuning the algorithm

4. Convergence analysis

(Bouffanais and Porter, 2018)

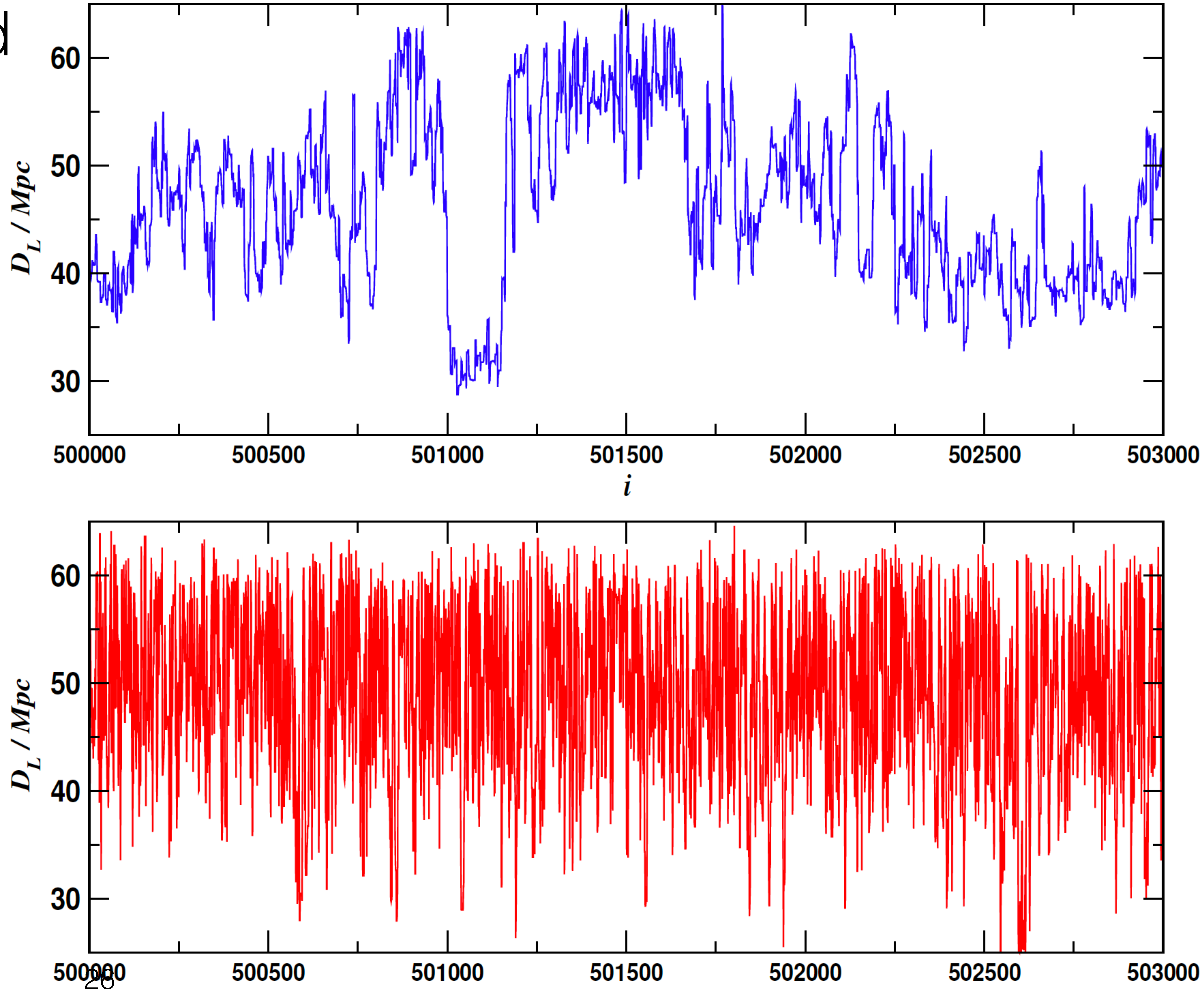
Autocorrelation falls off more rapidly

IV. Tuning the algorithm

5. Effective sample size

- Effective sample size is increased
  - 10^6 MCMC => 10^2 - 10^3
  - 10^6 HMC => 10^4 - 10^5
- Chain exploration is much more efficient

(Bouffanais and Porter, 2018)



Conclusion

- Non-random walk sampler
- Tuning + $\frac{\partial ln \mathscr{L}(q^\mu)}{\partial q^\mu}$ computation problem solved
- Larger effective sample size with shorter chains compared to MCMC
- Ongoing work
    - C to python
    - Include spins and tidal deformations
    - Main goal: introduce HMC in **L**SC **A**lgorithm **L**ibrary
    - Galactic Binaries, spinning SMBHBs, EMRIs for LISA

Thank you for your attention !