Gravitational Wave Analysis: Algorithms and acceleration

John Veitch
University of Glasgow

Einstein Telescope e-Infrastructure Board Workshop
Geneva
26/10/2023

Overview

- Gravitational Wave Data
- Classical Analysis Strategies
  - Core algorithms
- Machine Learning
  - Replacement
  - Augmentation
- Discussion

Gravitational Wave Data - the basics

- Strain data: Float64 @ 16384 Hz = 4TB/year per detector raw data
- d(t) = h(t) = n(t)
  - Noise usually modelled as stationary gaussian process: Power Spectral Density
  - h(t) contains:
    - ~1 CBC / minute
    - + Pulsars
    - + Supernovae
    - + Stochastic background
    - + unknown??

Astrophysics Requires

- Identification (“searches”)
- Characterisation (“parameter estimation”)
- Cataloguing
- Population Analysis
- Confronting Theory (incl. GR)

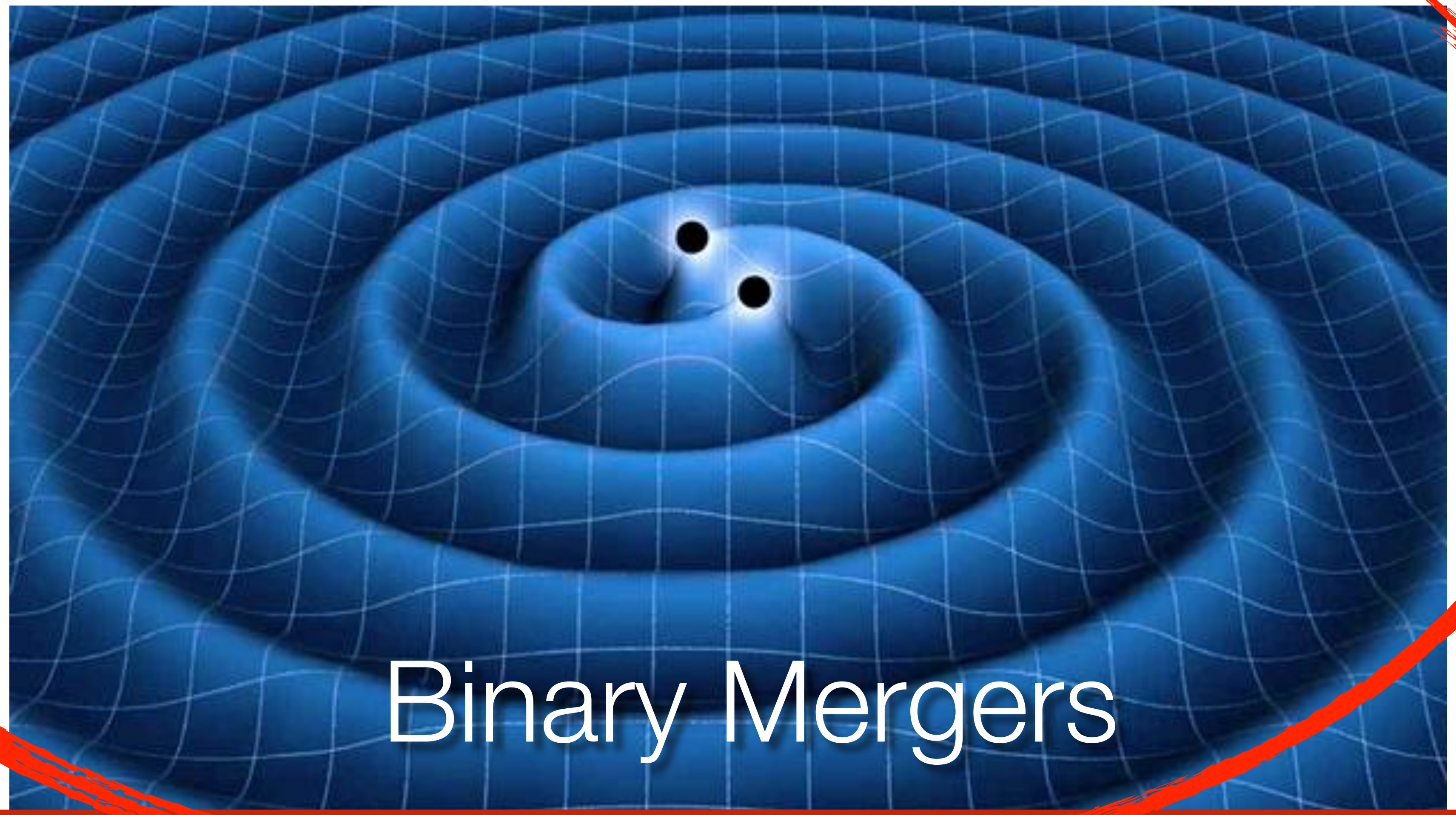
University of Glasgow

Categories of sources

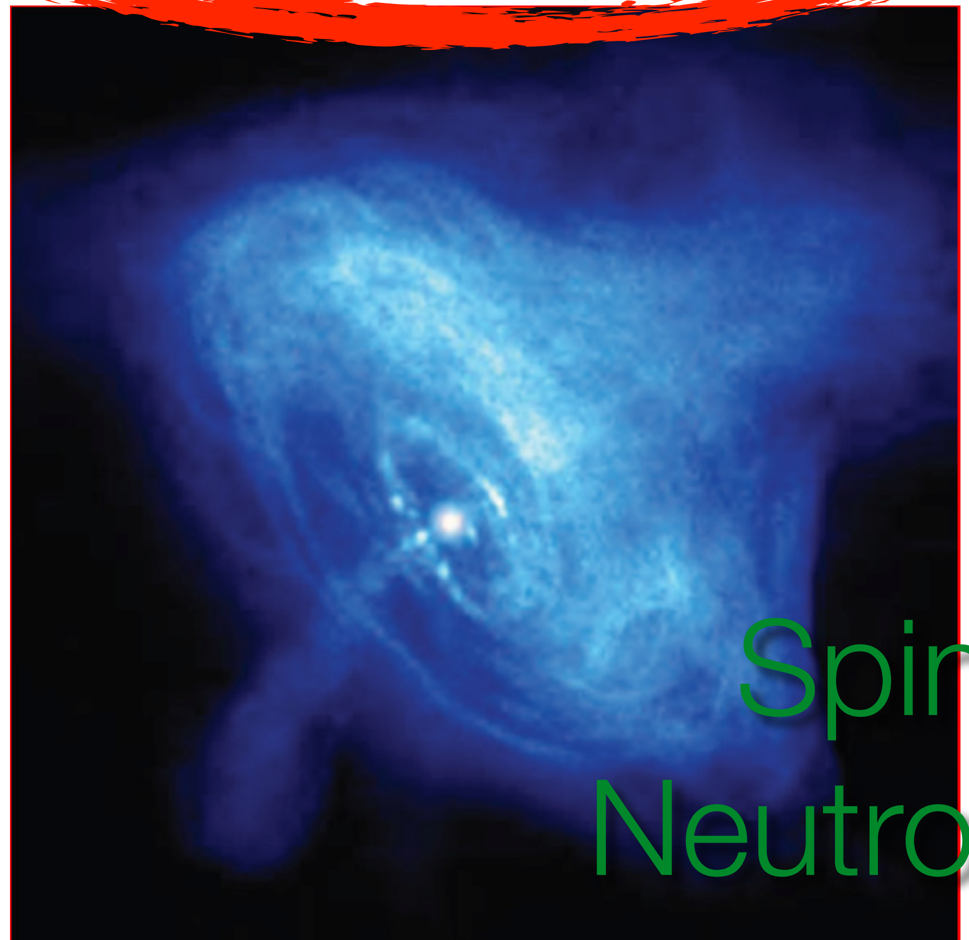
Modelled

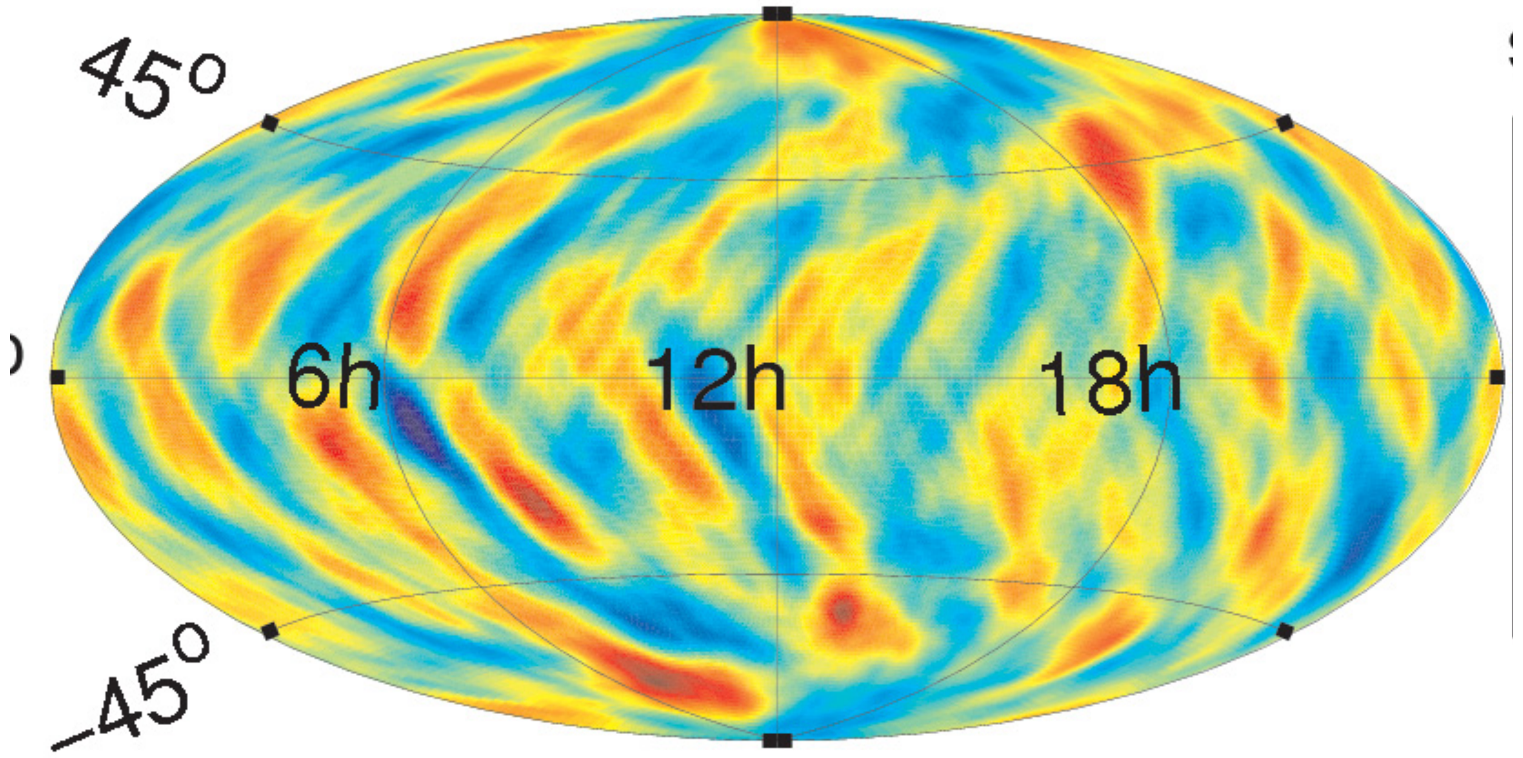
Unmodelled

Transient



Continuous

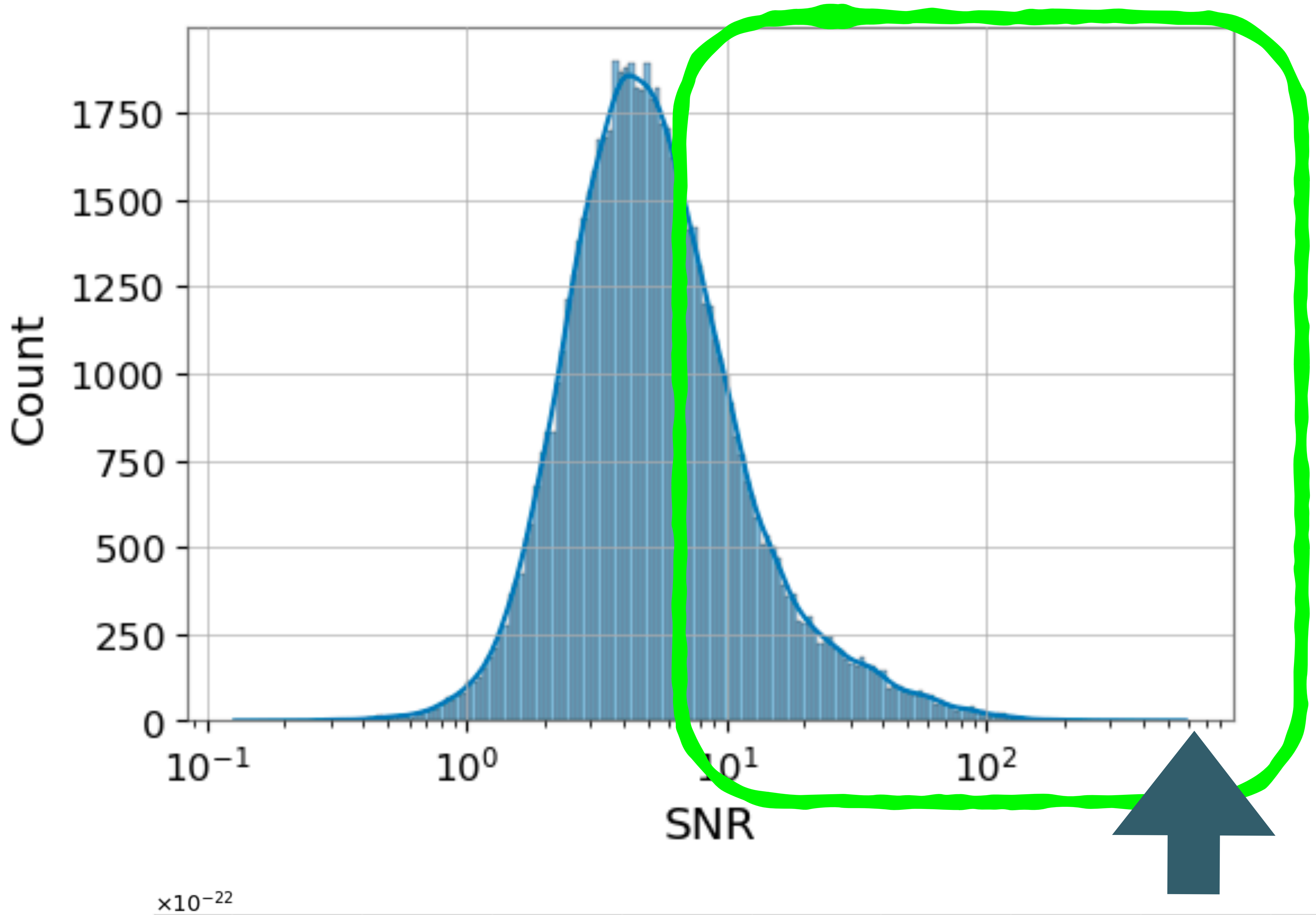


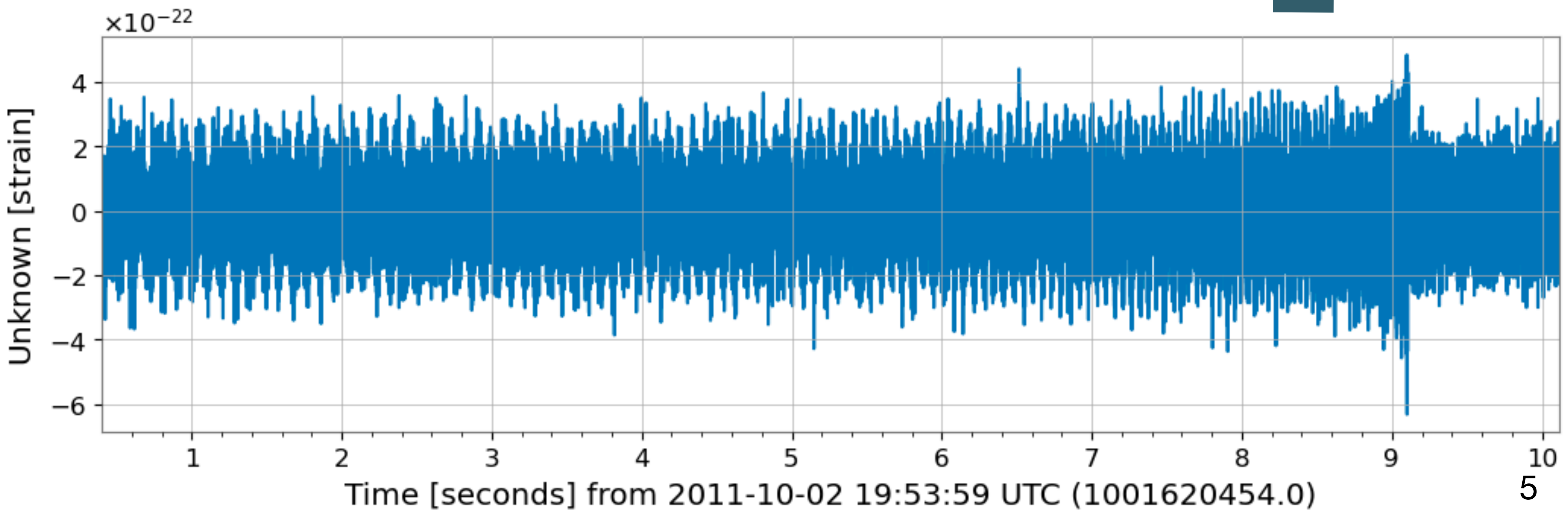


Stochastic Background

Compact Binaries

- **Signal to noise ratio** drives detectability and amount of extractable information
- ET will *detect* a CBC (SNR>6) every ~90s (MDC)
- Noise limited: most sources are quiet, similar distribution as we have with LIGO-Virgo-KAGRA.
    - But signals are much longer!
    - 1.4-1.4 $M_\odot$ from 5Hz: 107 mins vs ~3 mins from 20Hz
- Rate increases by ~3 orders of magnitude

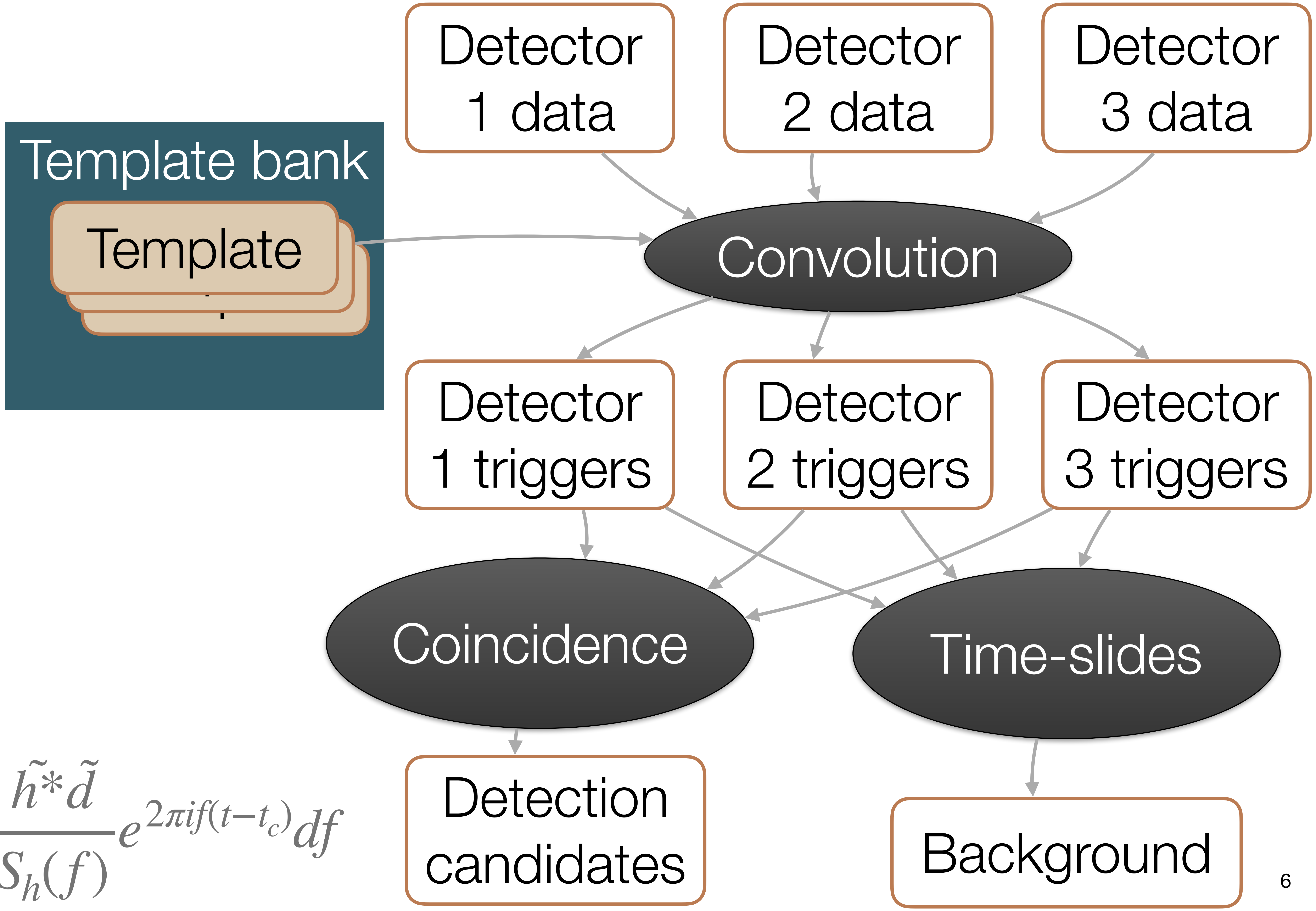




University of Glasgow

CBC Detection

- Matched filter pipelines: PyCBC, GSTLAL, MBTA, SPIIR
- Exhaustive search strategy
- Compute detection statistic for all templates, data, times
  - Massively parallel
- Maximise over extrinsic params
- Search over time with FFT



$$\rho(t) = \int \frac{\tilde{h}^* \tilde{d}}{S_h(f)} e^{2\pi i f (t - t_c)} df$$

University of Glasgow

Number of templates

- Fisher Information Metric on parameter space

$$\Gamma_{ij} = \left\langle \frac{\partial \hat{h}}{\partial \theta_i} \middle| \frac{\partial \hat{h}}{\partial \theta_j} \right\rangle \qquad < a|b >= \int_0^\infty \frac{a(f)^* b(f)}{S_h(f)} df$$

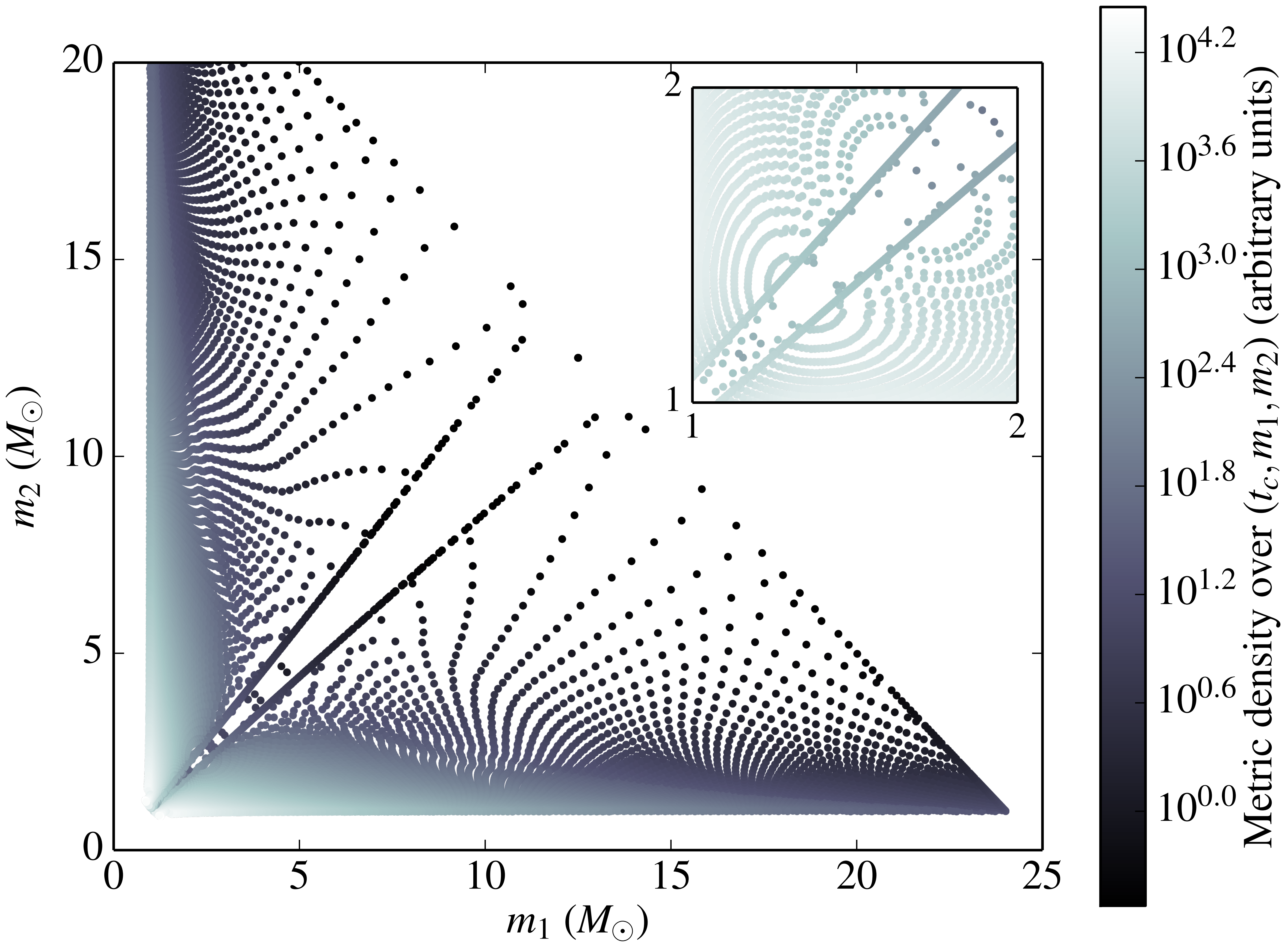
$$\log L(d|A,\vec{\theta}) \approx \frac{1}{2}\left[A_{ML}^2(1-\Gamma_{ij}(\theta_{ML})\Delta\theta_i\Delta\theta_j) - \Delta A^2 + \ldots\right]$$

- Template bank density $\propto \sqrt{\det \Gamma}$

- For chirp mass, $\dfrac{\Delta M_c}{M_c} \propto M_c^{5/3} \propto (\#\text{cycles})^{-1}$

$$\phi(f) = 2[8\pi M_c f]^{-5/3}$$

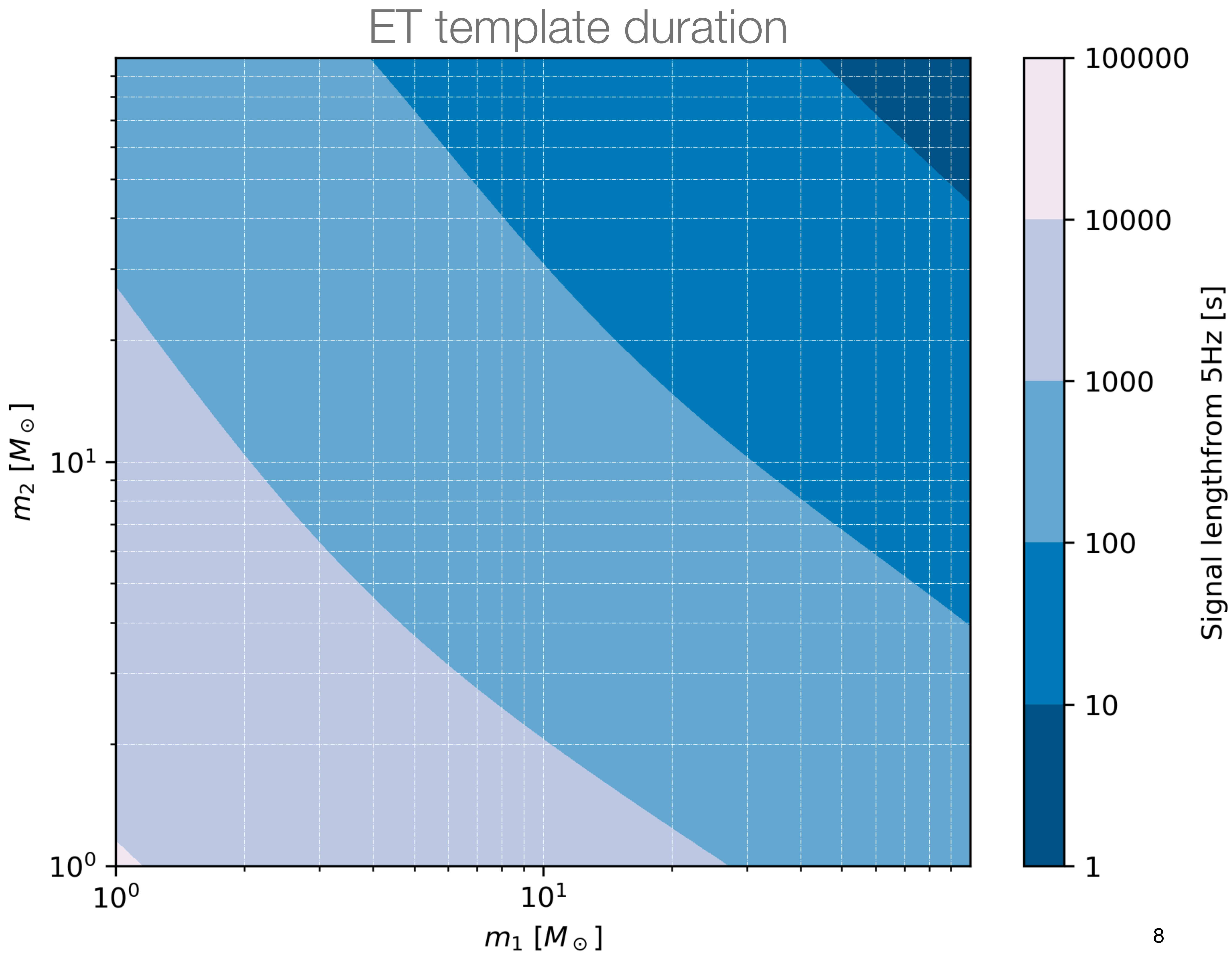
- About 1 order of magnitude more cycles from low freq 5Hz vs 20 Hz



University of Glasgow

Scaled up to Einstein Telescope

- More templates from
  - More cycles in band (10x)
- Longer templates
  - More data per filter (~100 x)
  - FFT scaling O(N log(N))
- More signals!
  - ~1000 x
- Search cost ~10 x 100 log(100) ≅2000 greater?
- But not all of these are essential to detect loudest sources!



Parameter Estimation

- Bayesian inference problem: quantify uncertainty on parameters $\vec{\theta}$ caused by noisy measurements (and other uncertainty e.g. calibration)
- Posterior probability distribution function

Prior Likelihood

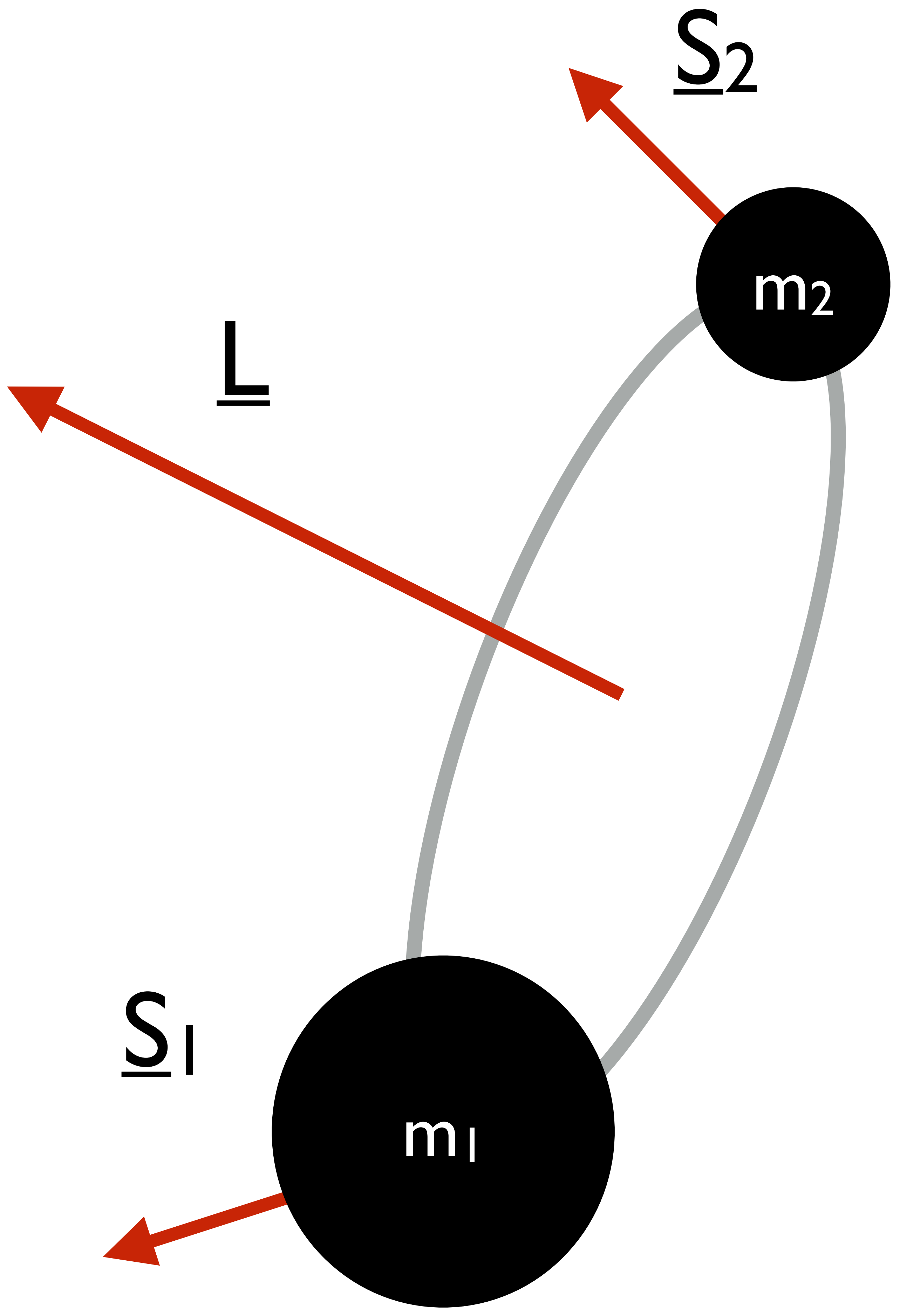
$$p(\vec{\theta} | d, H) = \frac{p(\vec{\theta} | H)\, p(d | \vec{\theta}, H)}{p(d | H)}$$

Evidence

- Produce *samples* from the posterior
  - Stochastic sampling algorithms (e.g. MCMC, Nested Sampling)
  - 1000s of final independent samples desirable
- Likelihood function also based on noise-weighted inner product

What is there to measure?

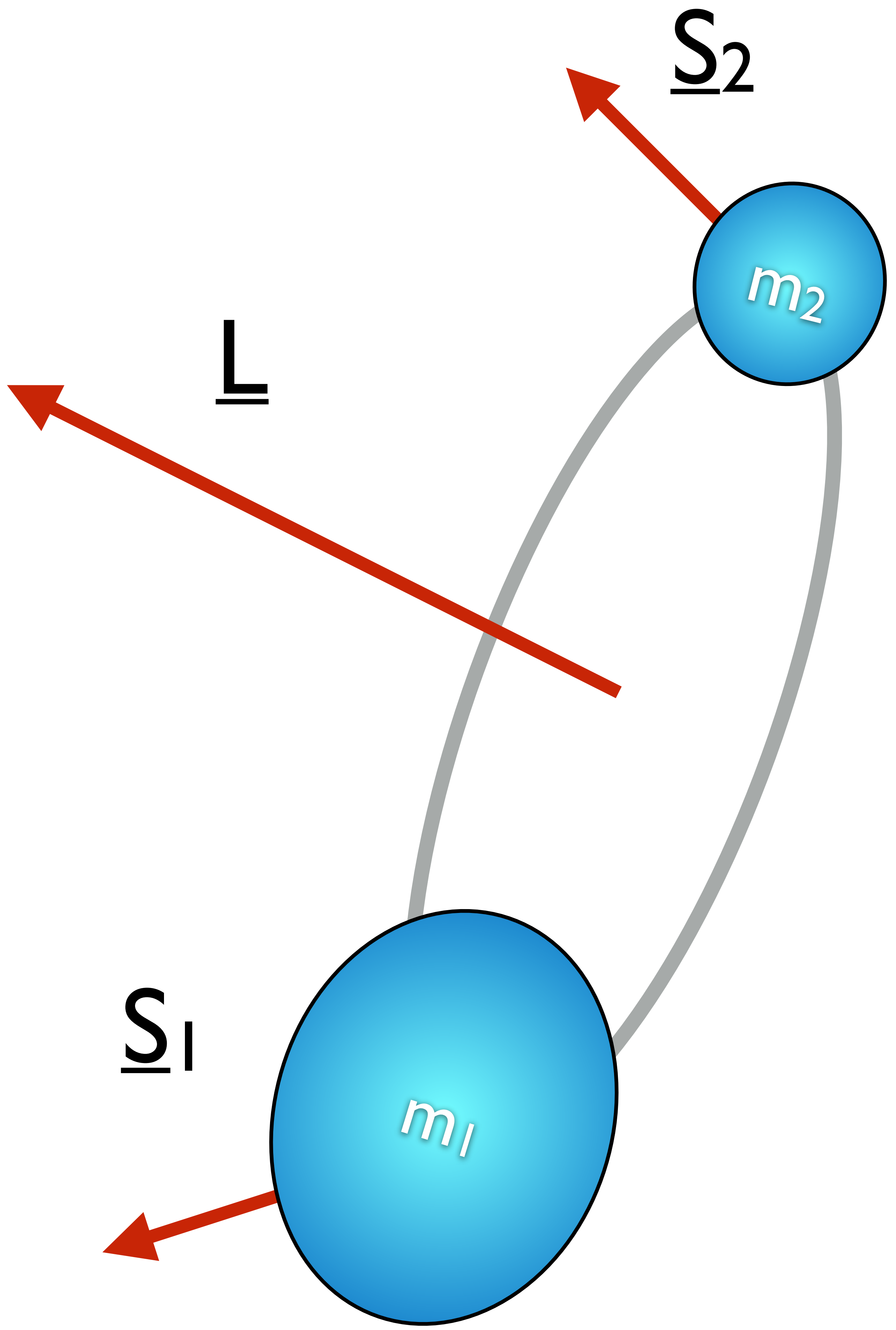
- Intrinsic Parameters
  - masses
  - spins
- Extrinsic Parameters
  - Inclination
  - Orientation
  - Polarisation
  - Sky position
  - luminosity distance
  - time



What is there to measure?

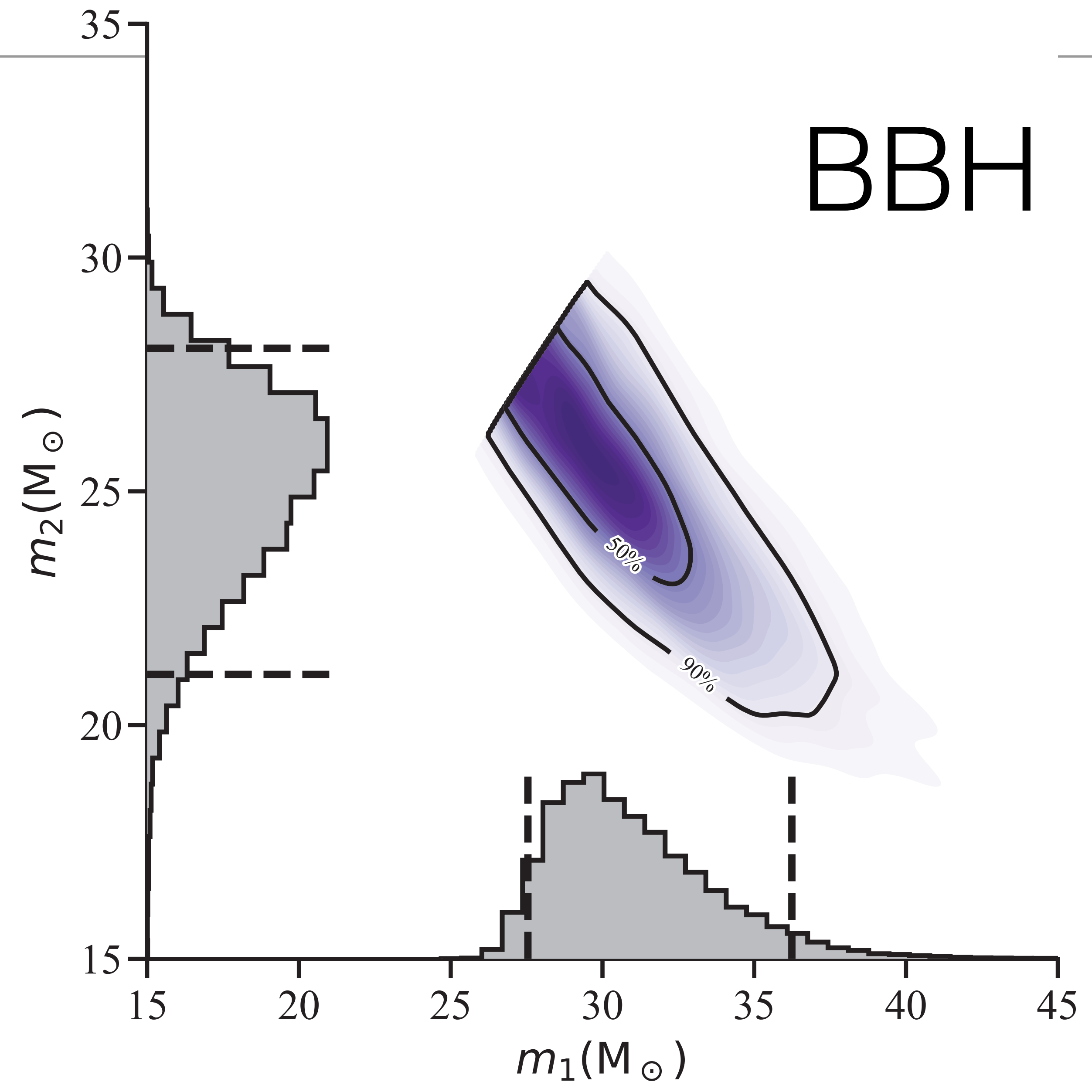
Subtler effects

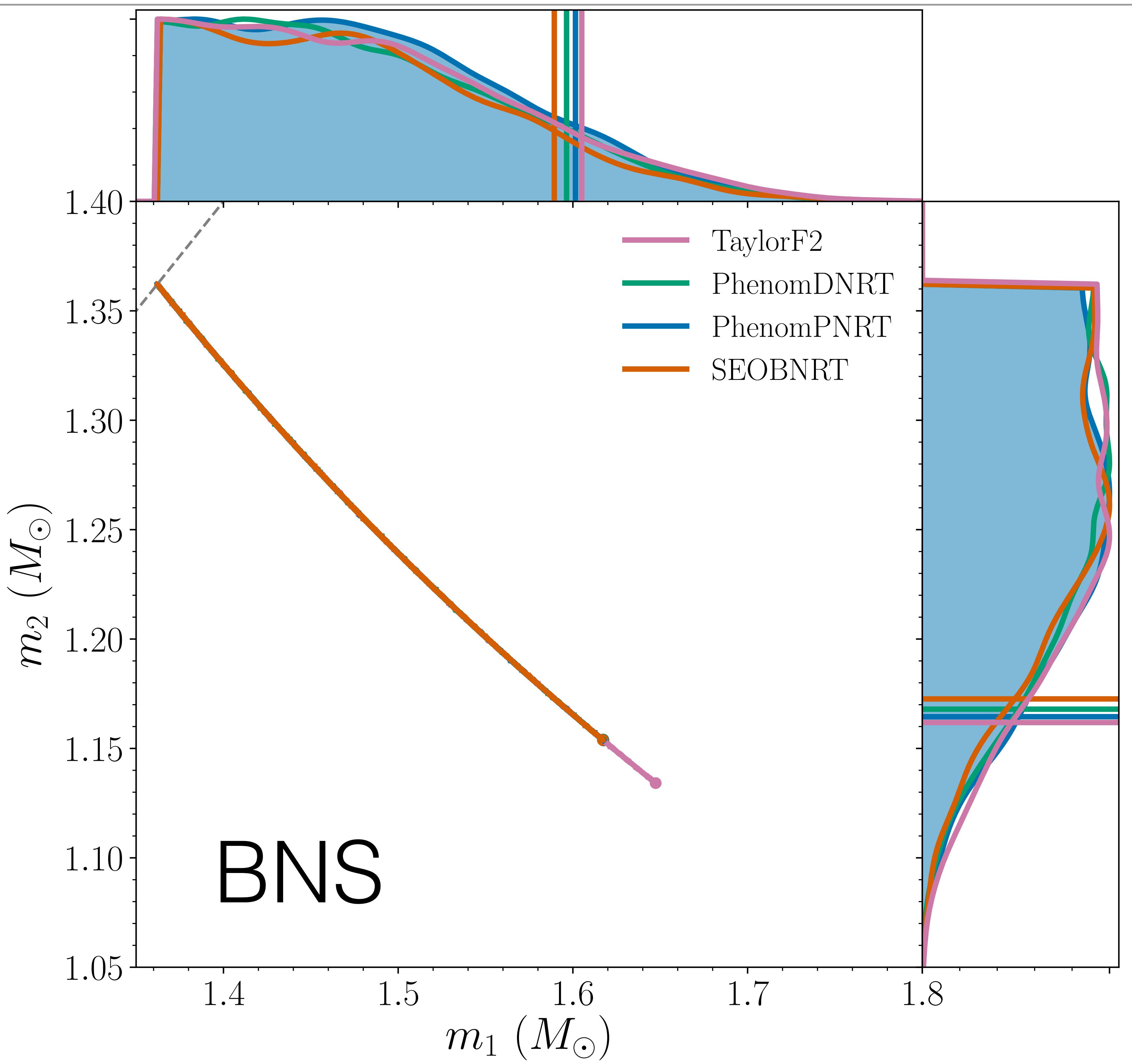
- NS Equation of state
  - tidal deformation
- Deviations from GR
- eccentricity

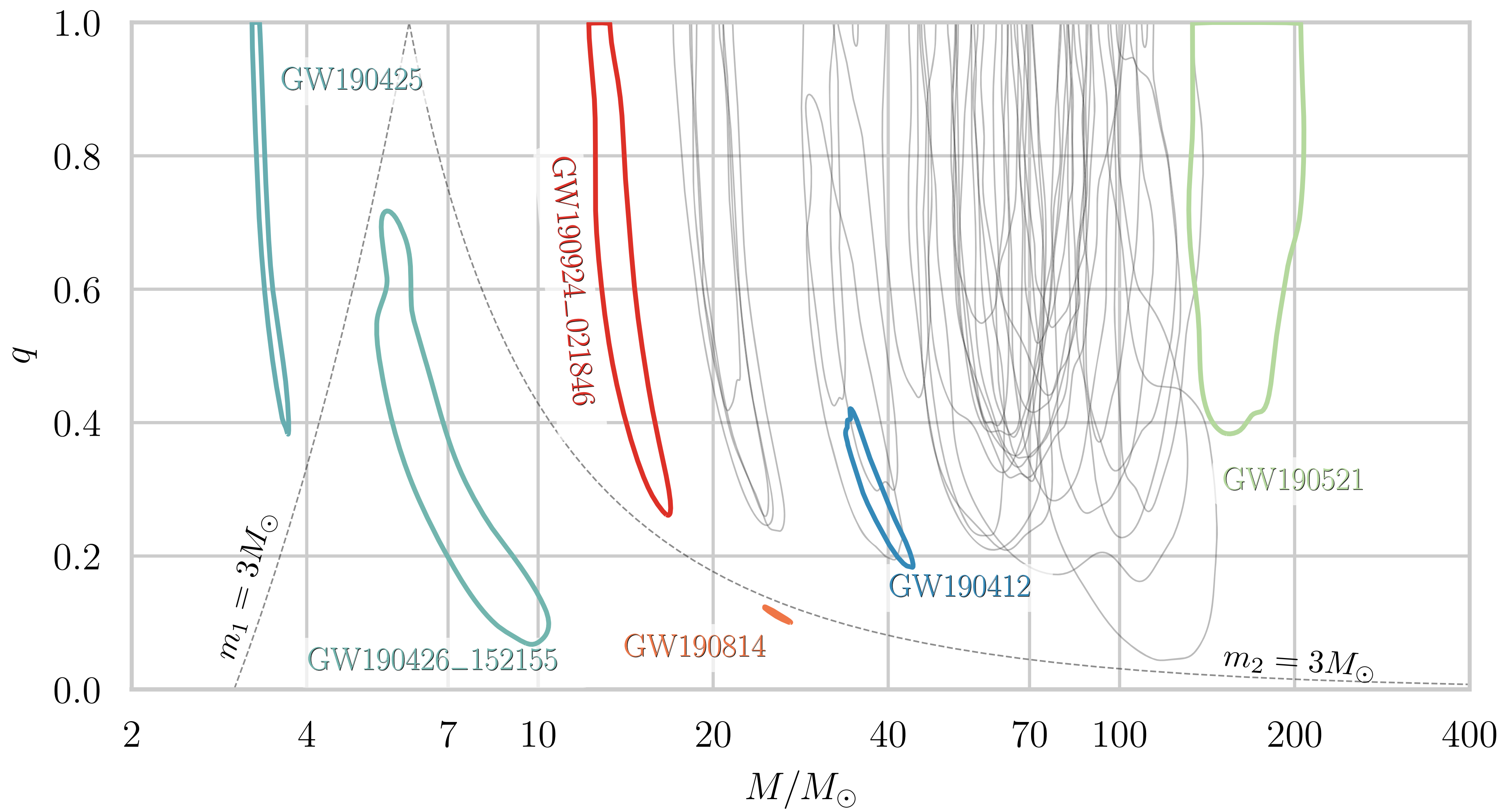


University of Glasgow

Parameter Estimation - masses







$$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{M^{1/5}} \simeq \frac{c^3}{G}\left[\frac{5}{96}\pi^{-8/3} f^{-11/3} \dot{f}\right]^{3/5}$$

Computational Cost

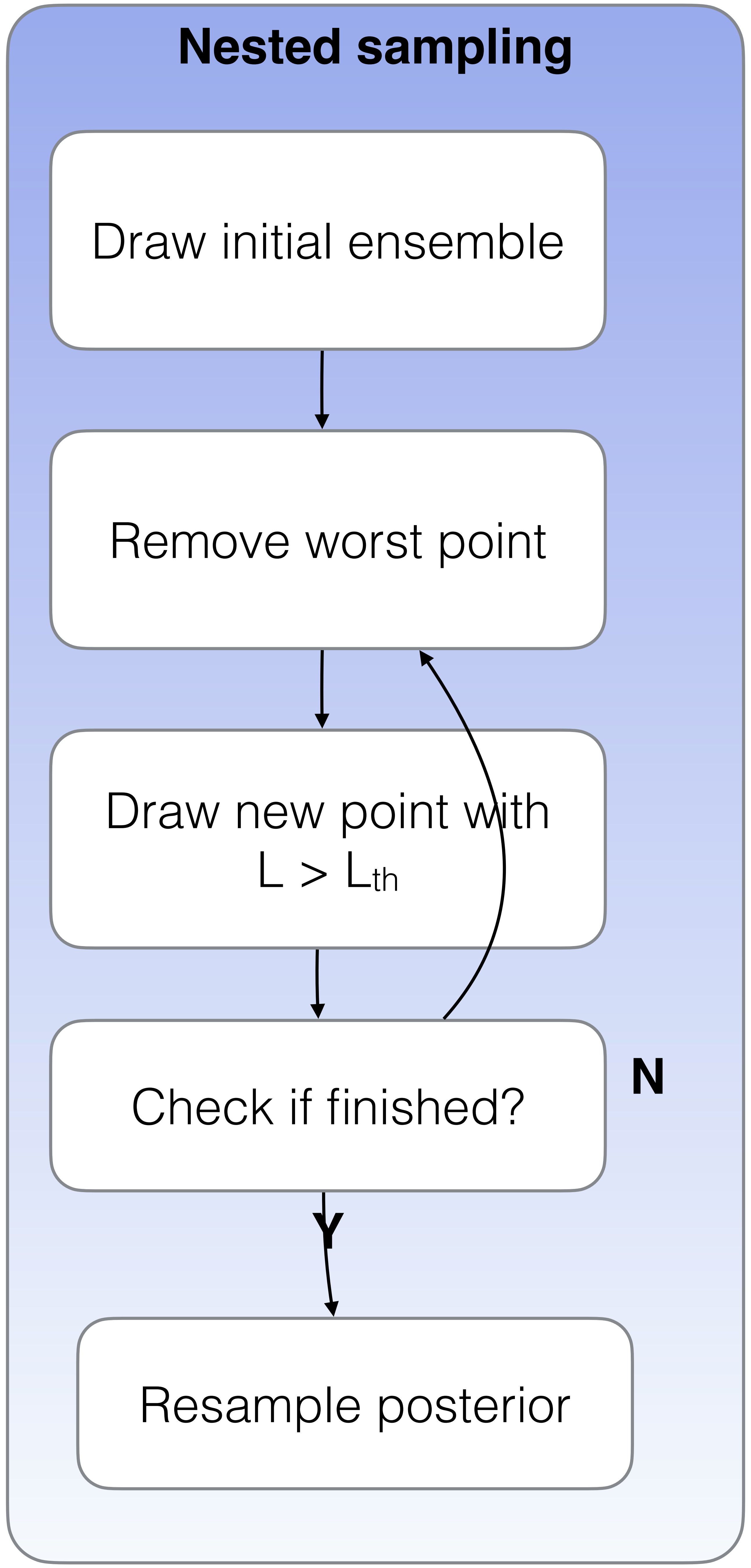
- For O3 LVK PE analyses, used tens of millions of cpu-hours for “production runs” (doesn’t include development + simulations)
  - 3 pipelines used:
    - LALInference, Bilby+dynesty: stochastic sampling
    - RIFT: Hybrid Grid+Monte Carlo
- Similar amounts again used for **testing GR**!
  - Mostly uses the same type of stochastic samplers, with more complex models
- This was actually slightly less than in O2!

Computational Scaling

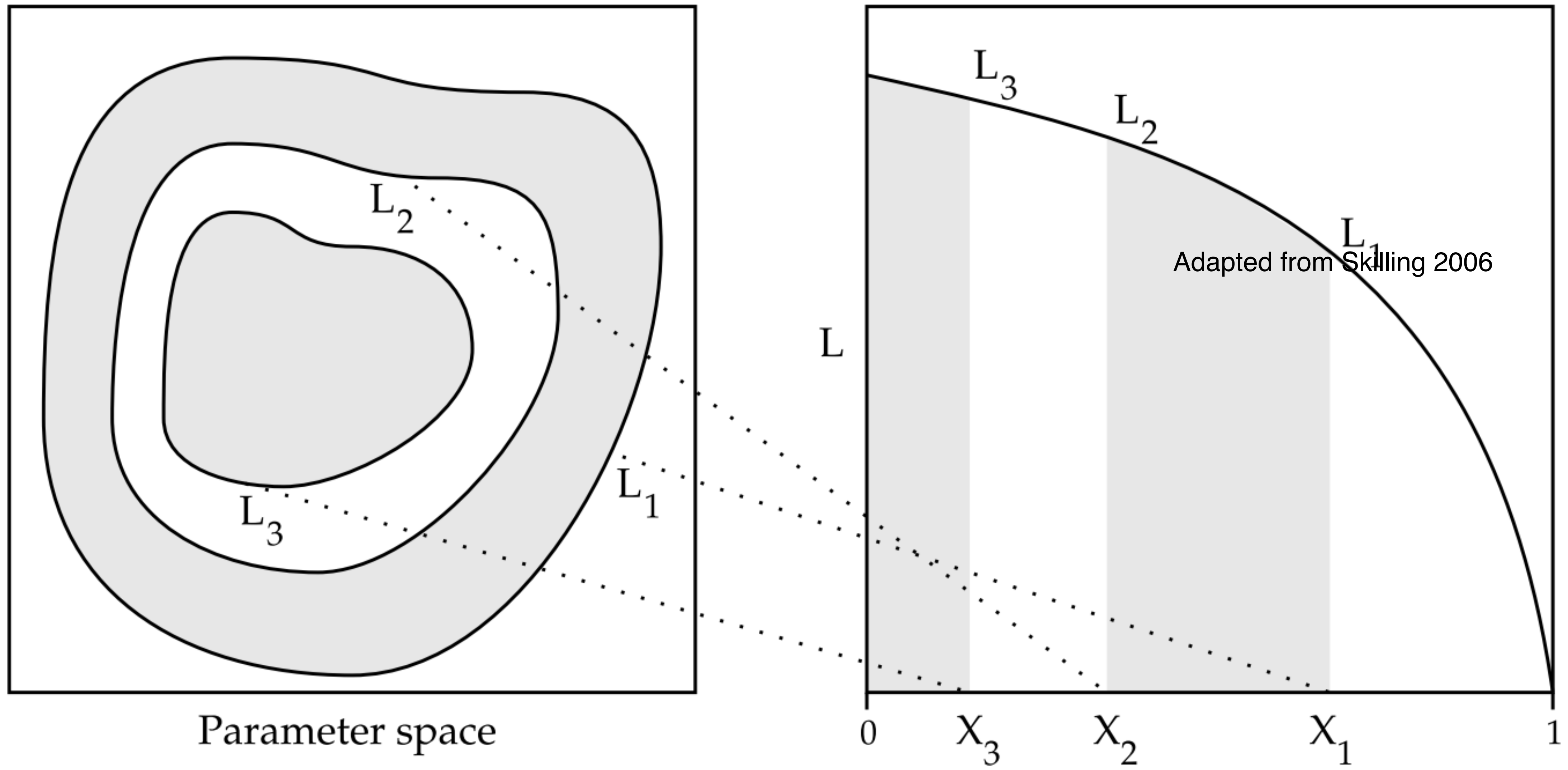
- How do computational costs scale in practice?
- Identify bottlenecks
  - Increase parallelism
  - Reduce inefficiencies
- Use example of nested sampling to break down the details, since this is my speciality.

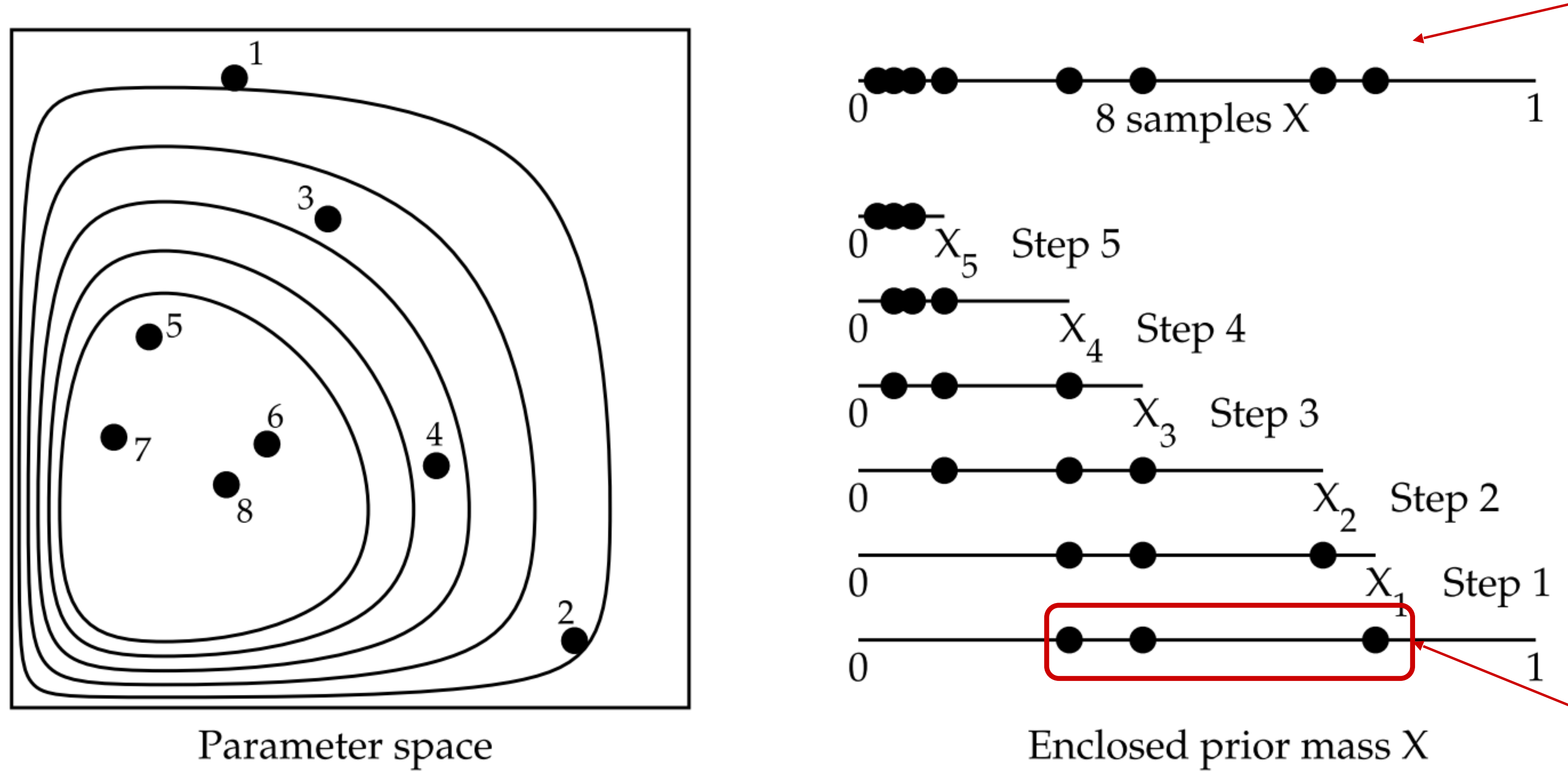
University of Glasgow

Nested Sampling



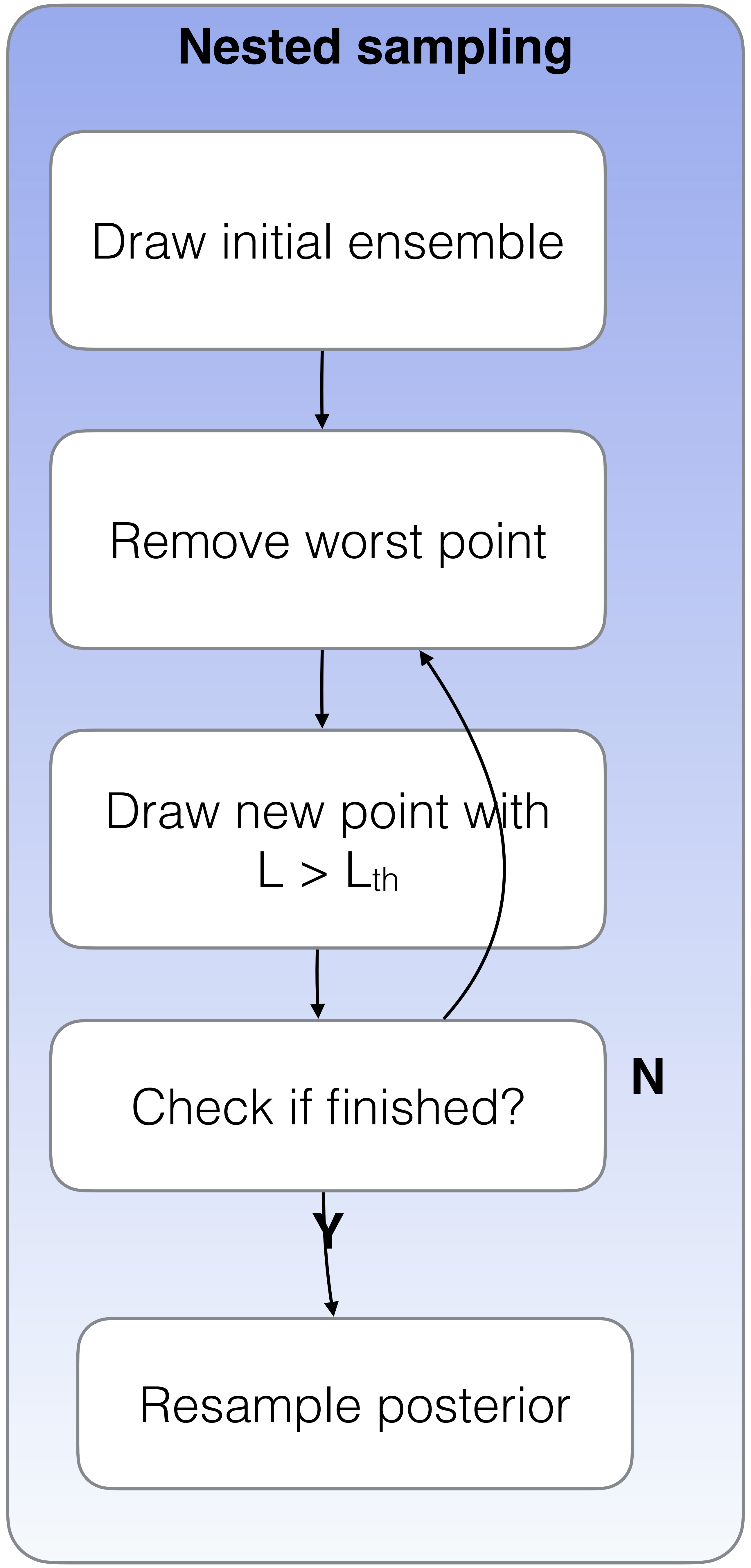
$$Z = \int_0^1 L(X)dX$$

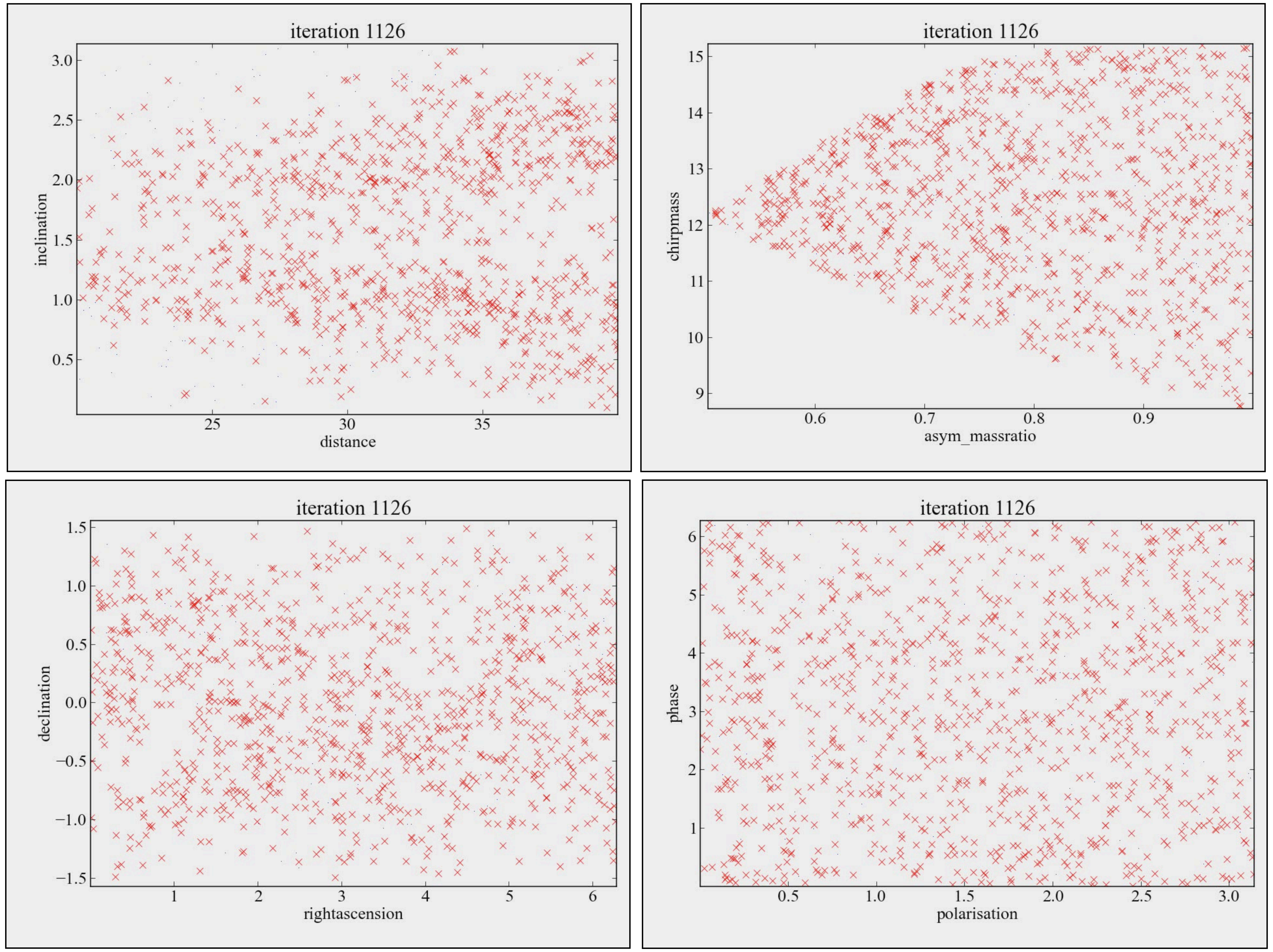




Used for the final evidence estimate

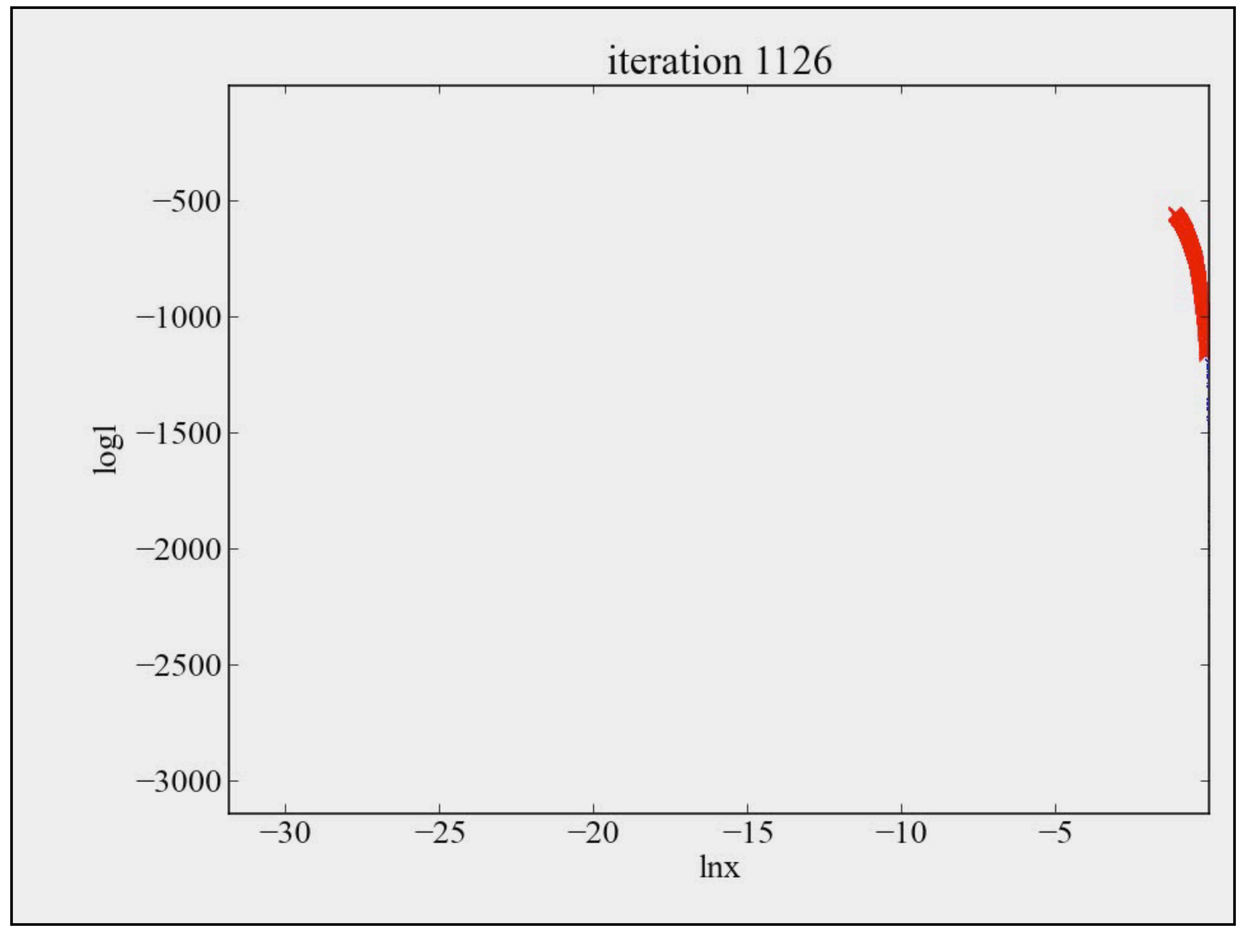
Initial live points

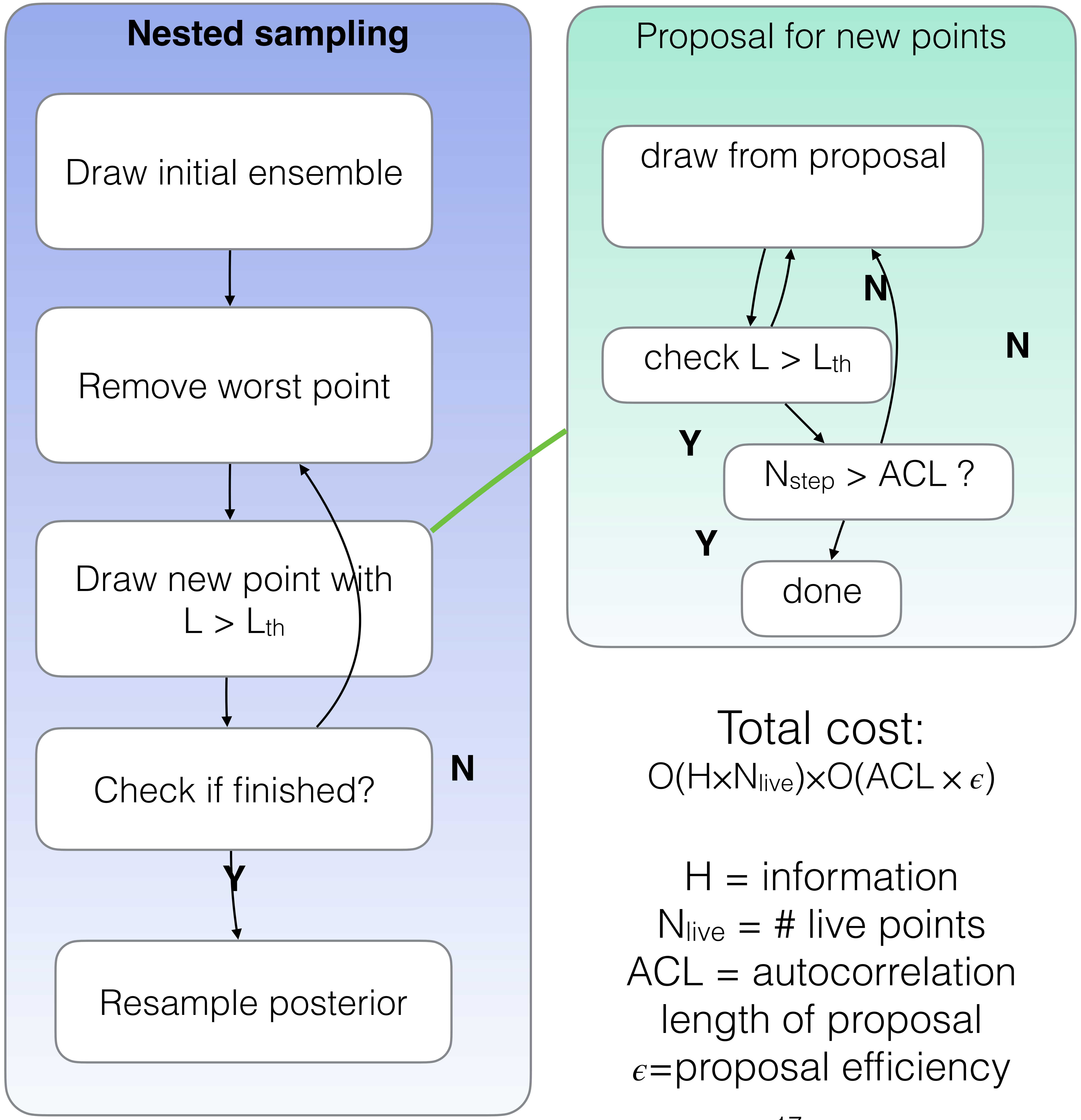




Total cost:
$O(H \times N_{live})$

H = information
N_{live} = # live points





Total cost:

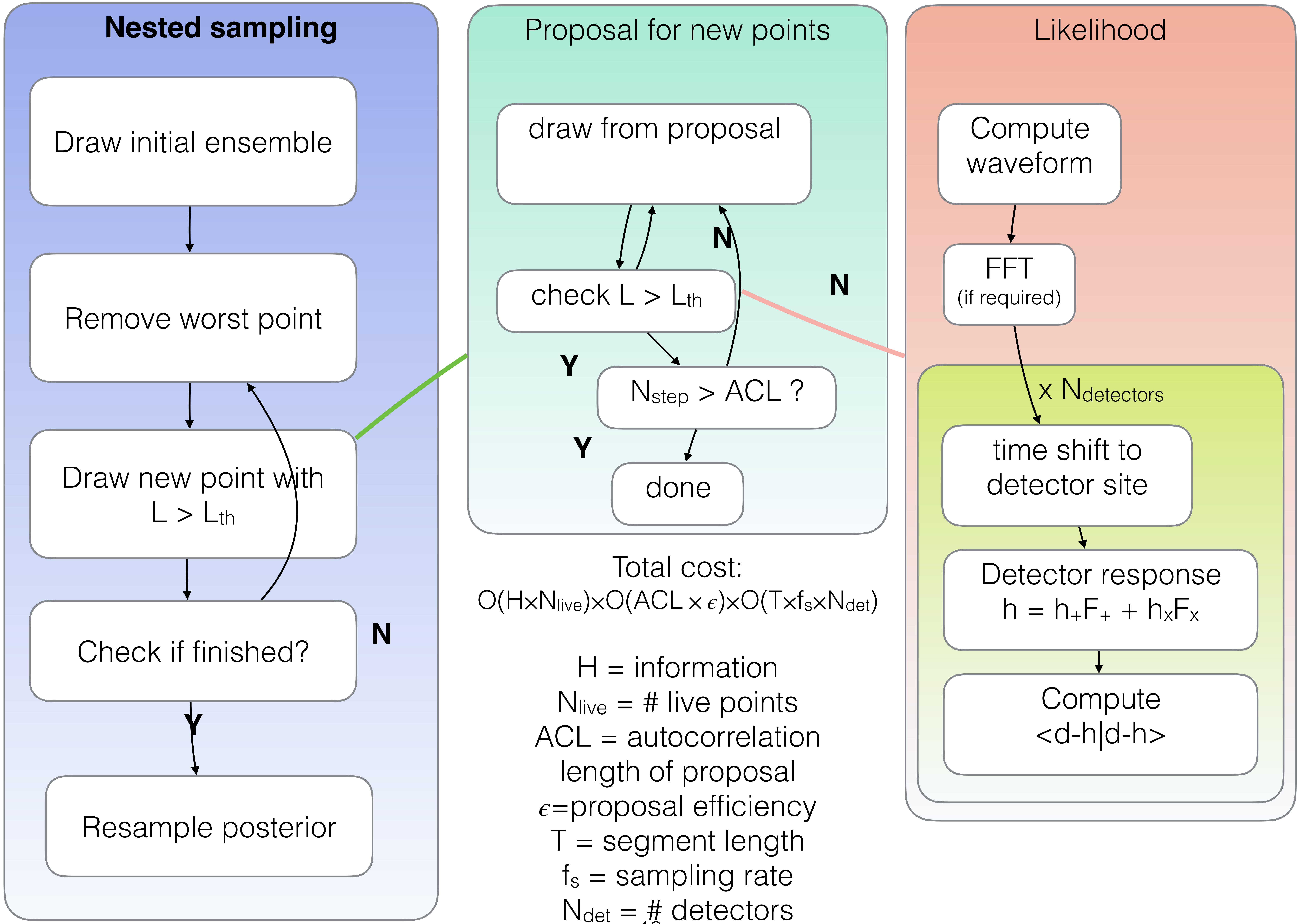
$O(H \times N_{live}) \times O(ACL \times \epsilon)$

H = information

N_{live} = # live points

ACL = autocorrelation length of proposal

ϵ=proposal efficiency



Nested sampling
Draw initial ensemble
Remove worst point
Draw new point with
L > Lth
Check if finished?
N
Y
Resample posterior
Proposal for new points
draw from proposal
check L > Lth
N
N
Y
Nstep > ACL ?
Y
done
Likelihood
Compute
waveform
FFT
(if required)
x Ndetectors
time shift to
detector site
Detector response
h = h+F+ + hxFx
Compute
<d-h|d-h>
Total cost:
O(H×Nlive)×O(ACL × ϵ)×O(T×fs×Ndet)
H = information
Nlive = # live points
ACL = autocorrelation
length of proposal
ϵ=proposal efficiency
T = segment length
fs = sampling rate
Ndet = # detectors

Waveform cost

BNS

f_s=4096 Hz
f_{min} = 20 Hz
T = 196 s
m1 = 1.4 $M_\odot$
m2 = 1.4 $M_\odot$

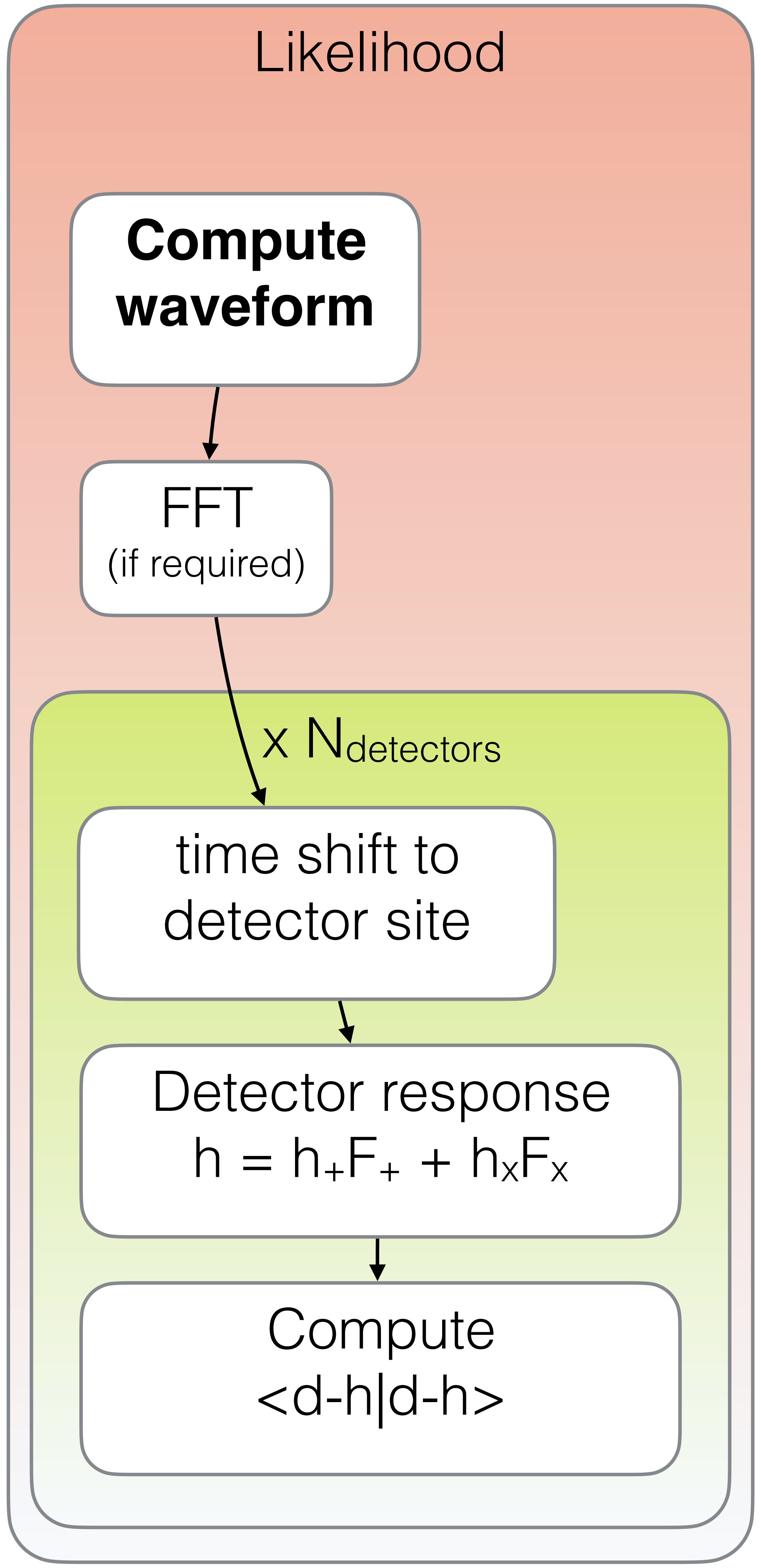
BBH

f_s=4096 Hz
f_{min} = 20 Hz
T = 4 s
m1 = 36 $M_\odot$
m2 = 30 $M_\odot$

	BNS	BBH
IMRPhenomPv2	433 ms	1 ms
IMRPhenomXPHM	578 ms	4 ms
SEOBNRv4PHM	??	5050 ms
SEOBNRv4_ROM		1.5 ms
rest of likelihood function	15 ms	<0.1 ms

(non-spinning, averaged over extrinsic params)
computed on 4.2GHz i7-7700K

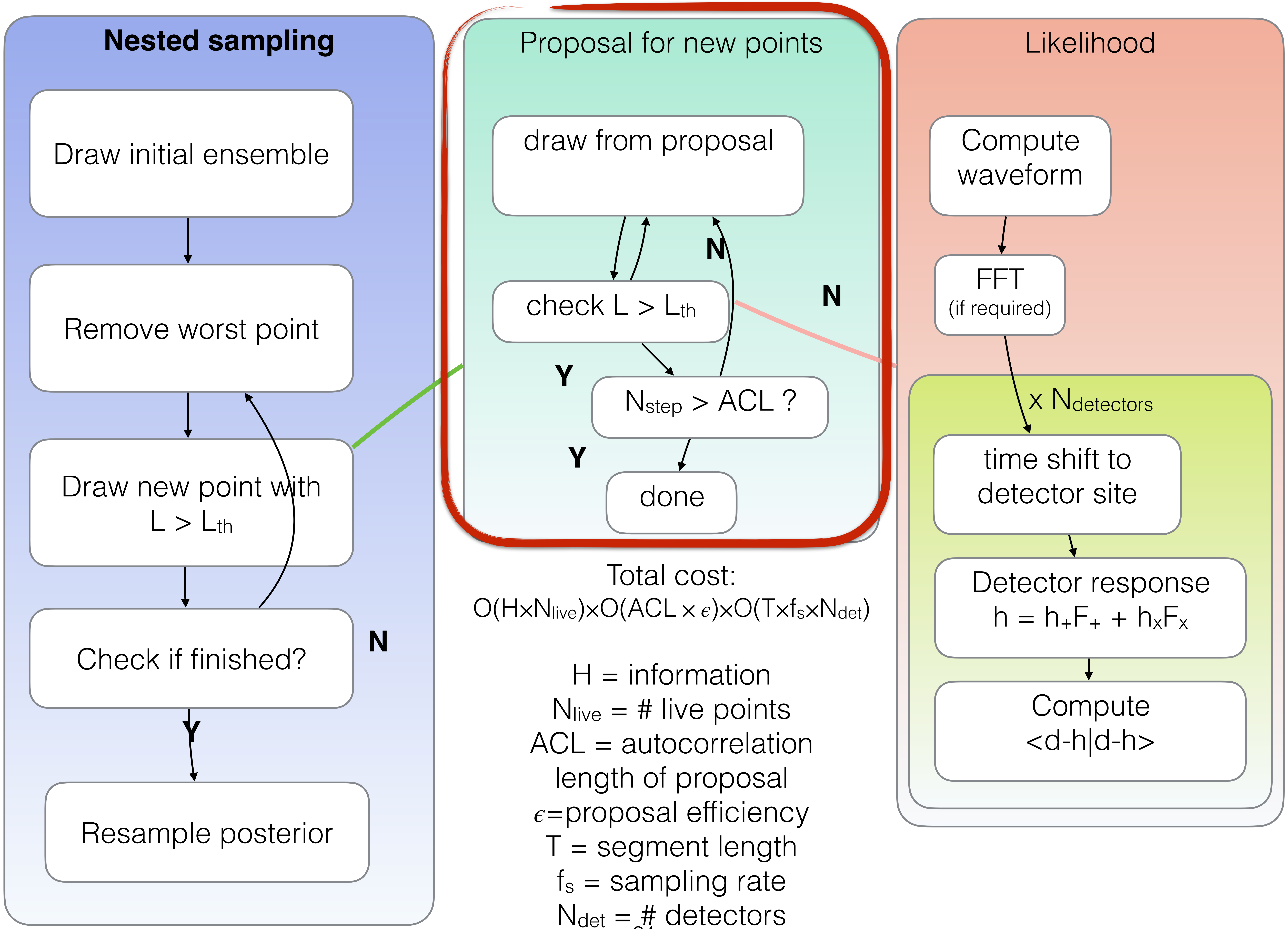
Typical PE run used ~10^7 waveforms
with O3 nested sampler



Waveform and Likelihood Acceleration

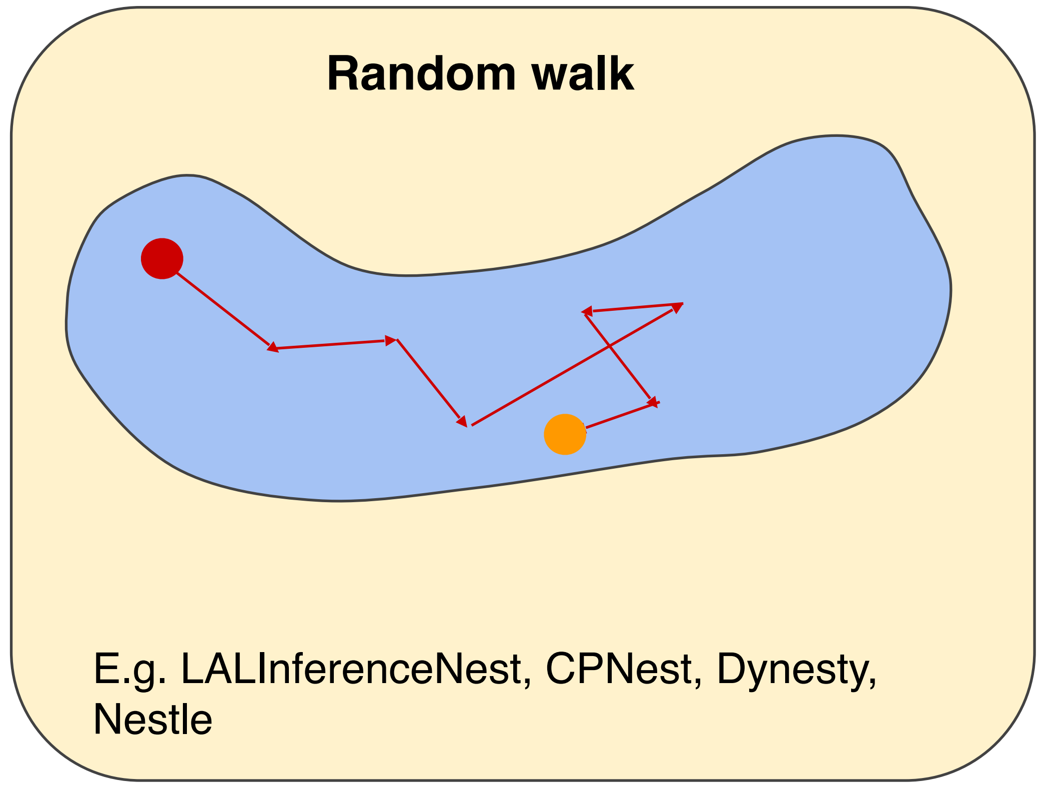
- Reduced Order Models (e.g. Puerrer arXiv:1512.02248, Cotesta+ arXiv:2003.12079)
  - Decompose waveform (A(f), ϕ(f)) into basis functions. Interpolate weights across q, $\vec{\chi}$
  - Bypasses time-domain PDEs (good for SEOB) and/or NR [Blackman+ arXiv:1701.00550])
  - Can make use of GPUs / ML methods for interpolation [e.g. Khan+ arXiv:2008.12932 , Barsotti+ arXiv:2110.08901]
  - >1000x speed up for very slow waveforms
- Reduced Order Quadrature [Canizares+ 1404.6284, Smith+1604.08253, Qi+ 2009.13812]
  - Replace inner product in freq domain with reduced basis
  - 10000x speed up!
  - Requires pre-computation of projection coefficients - narrow mass range or very large datasets
- Multi-band waveforms [Vinciguerra+ arXiv:1703.02062, Morisaki arXiv:2104.07813]
  - ~ 50x speed up for BNS but no precomputation
- Heterodyned likelihood (a.k.a Relative binning) [Cornish arXiv:1007.4820, arXiv:2109.02728, Zackay+ arXiv:1806.08792, Finstad+ arXiv:2009.13759]
  - Use difference between a reference waveform and proposed waveform to compute likelihood. Bandwidth of difference << full bandwidth of signal
  - Similar speed-up to ROQ for freq-domain waveforms but no pre-computation. Very powerful for BNS
  - Not (as) applicable to time-domain PDE based waveforms

University of Glasgow

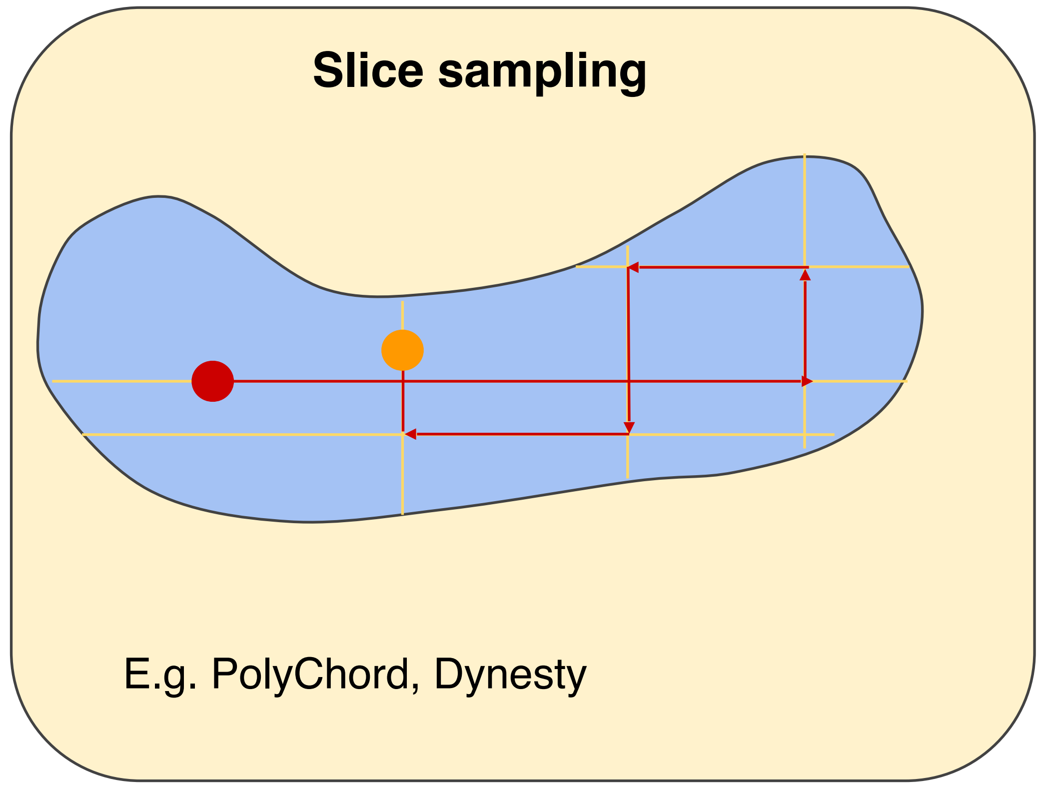


Total cost:
$O(H \times N_{live}) \times O(ACL \times \epsilon) \times O(T \times f_s \times N_{det})$

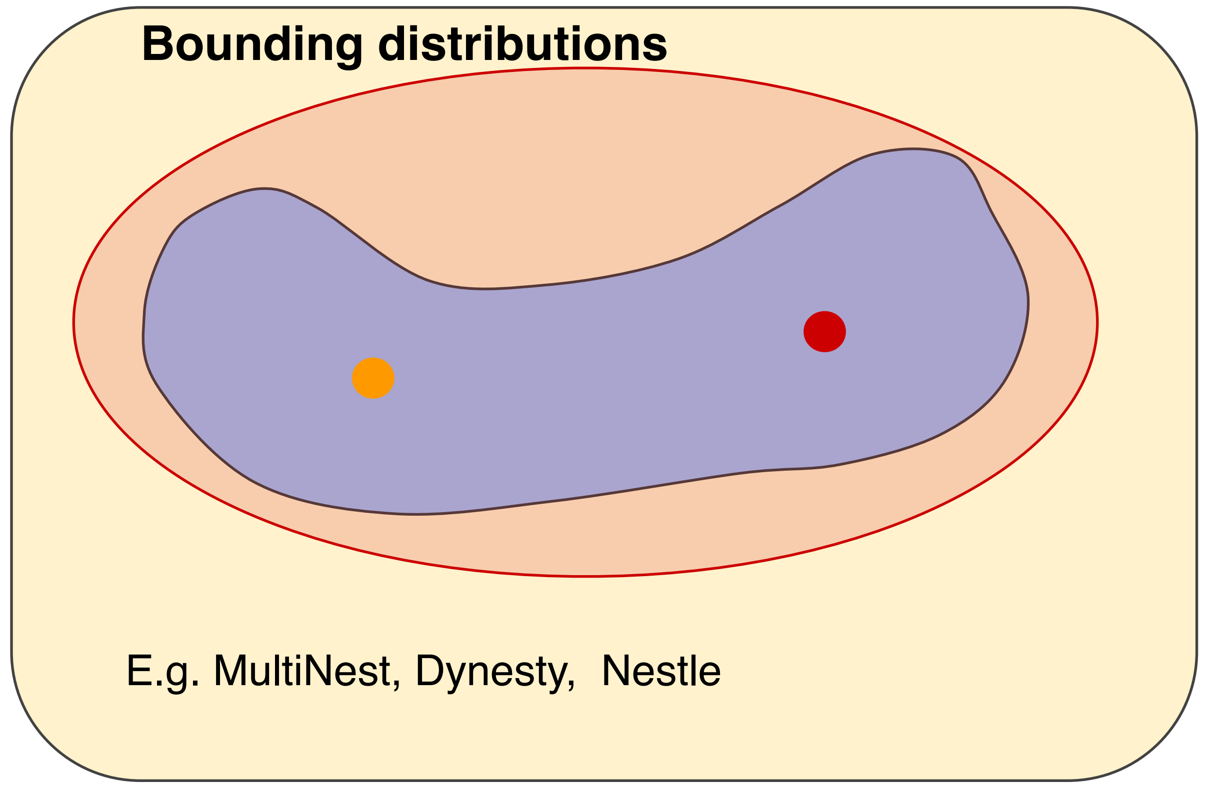
H = information
N_{live} = # live points
ACL = autocorrelation length of proposal
ϵ=proposal efficiency
T = segment length
f_s = sampling rate
N_{det} = # detectors



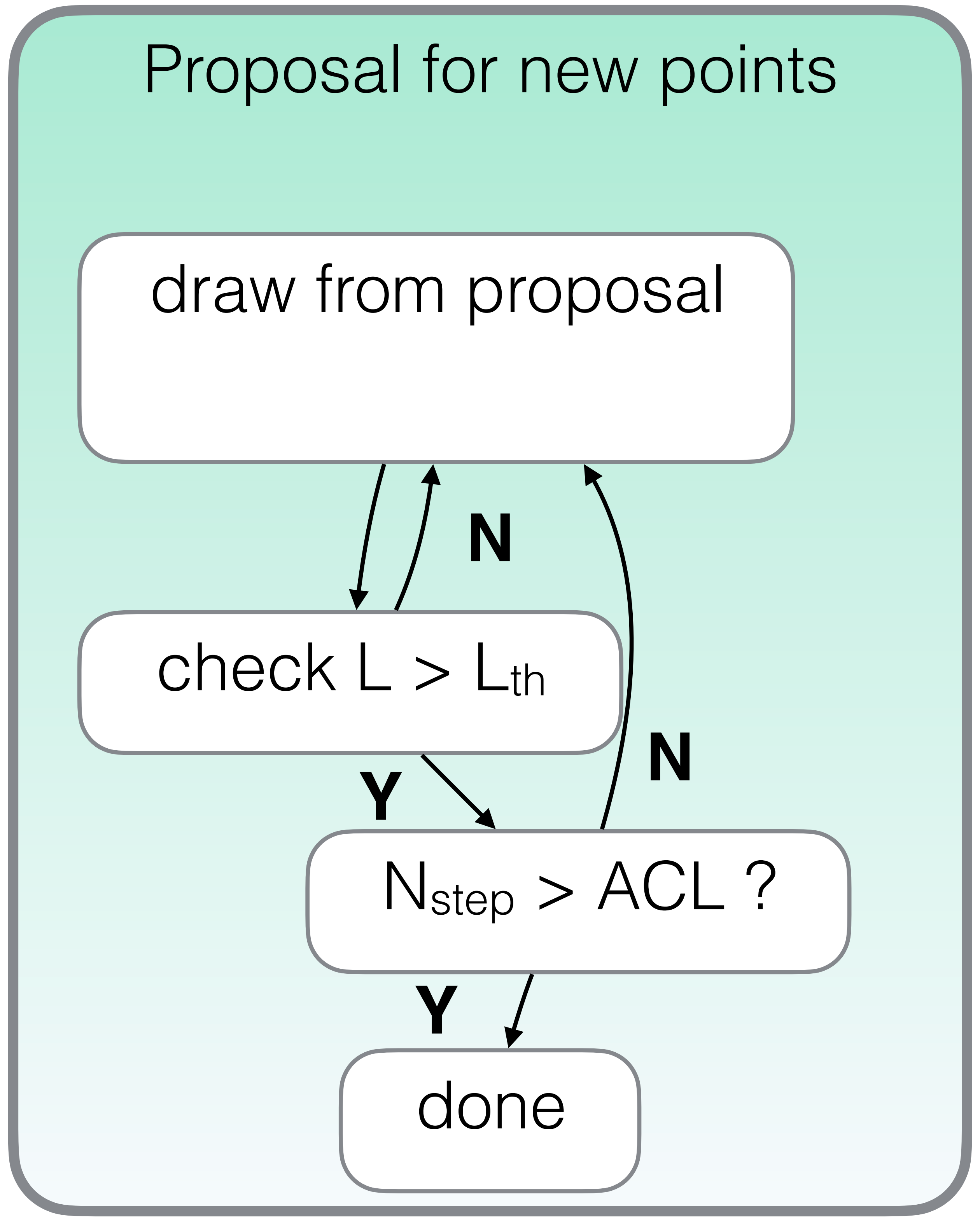
flexible
auto-tune for efficiency
long ACLs (5000 for CBC)



auto-tuning
long ACLs
multiple modes



zero ACL
poor efficiency in high dimensions



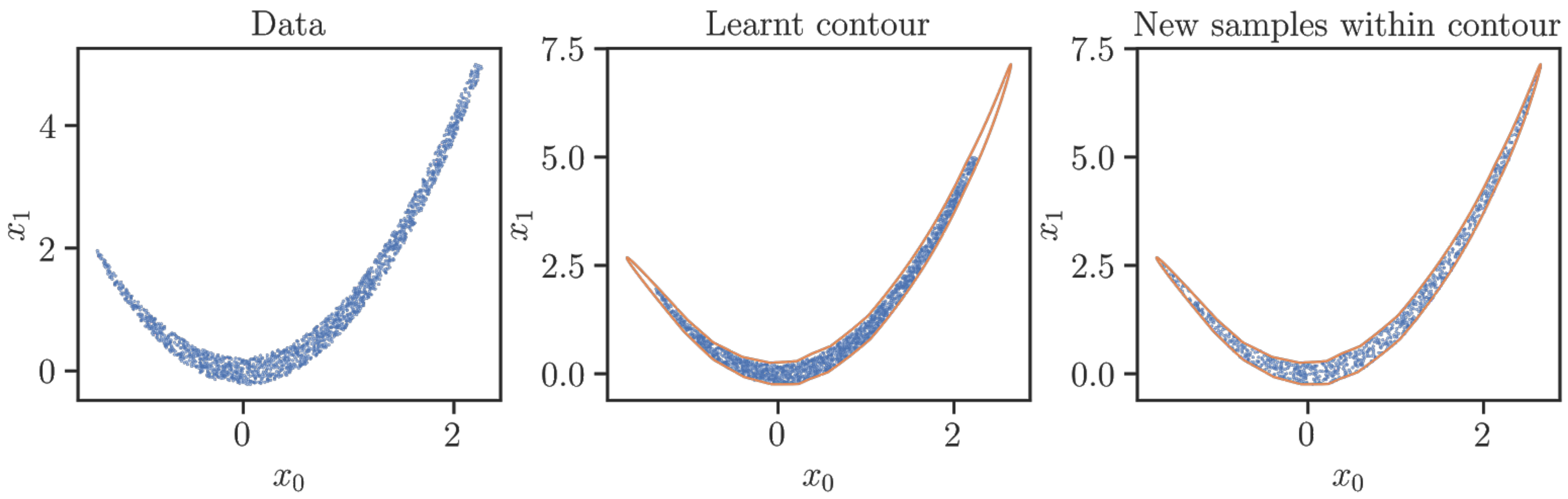
Total cost:

$$O(H \times N_{live}) \times O(ACL \times \epsilon) \times O(T \times f_s \times N_{det})$$

H = information
N_{live} = # live points
ACL = autocorrelation length of proposal
ϵ=proposal efficiency
T = segment length
f_s = sampling rate
N_{det} = # detectors

Example: Nested sampling with AI

Train a machine learning algorithm to learn iso-likelihood contours during nested sampling and then sample directly from those contours to produce new samples according to the prior.



Normalising flows

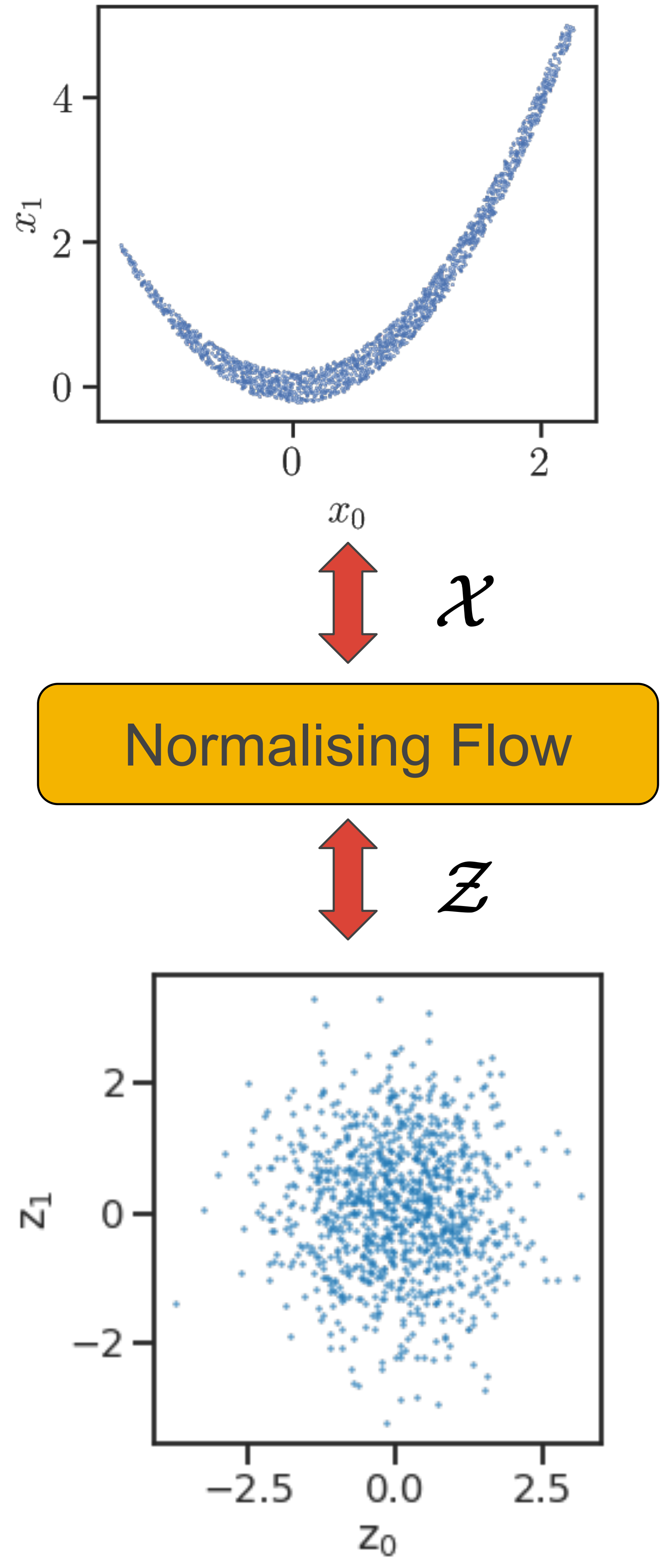
- They learn an invertible mapping (f) from a complex distribution in the physical space X to a simple distribution in the latent space Z

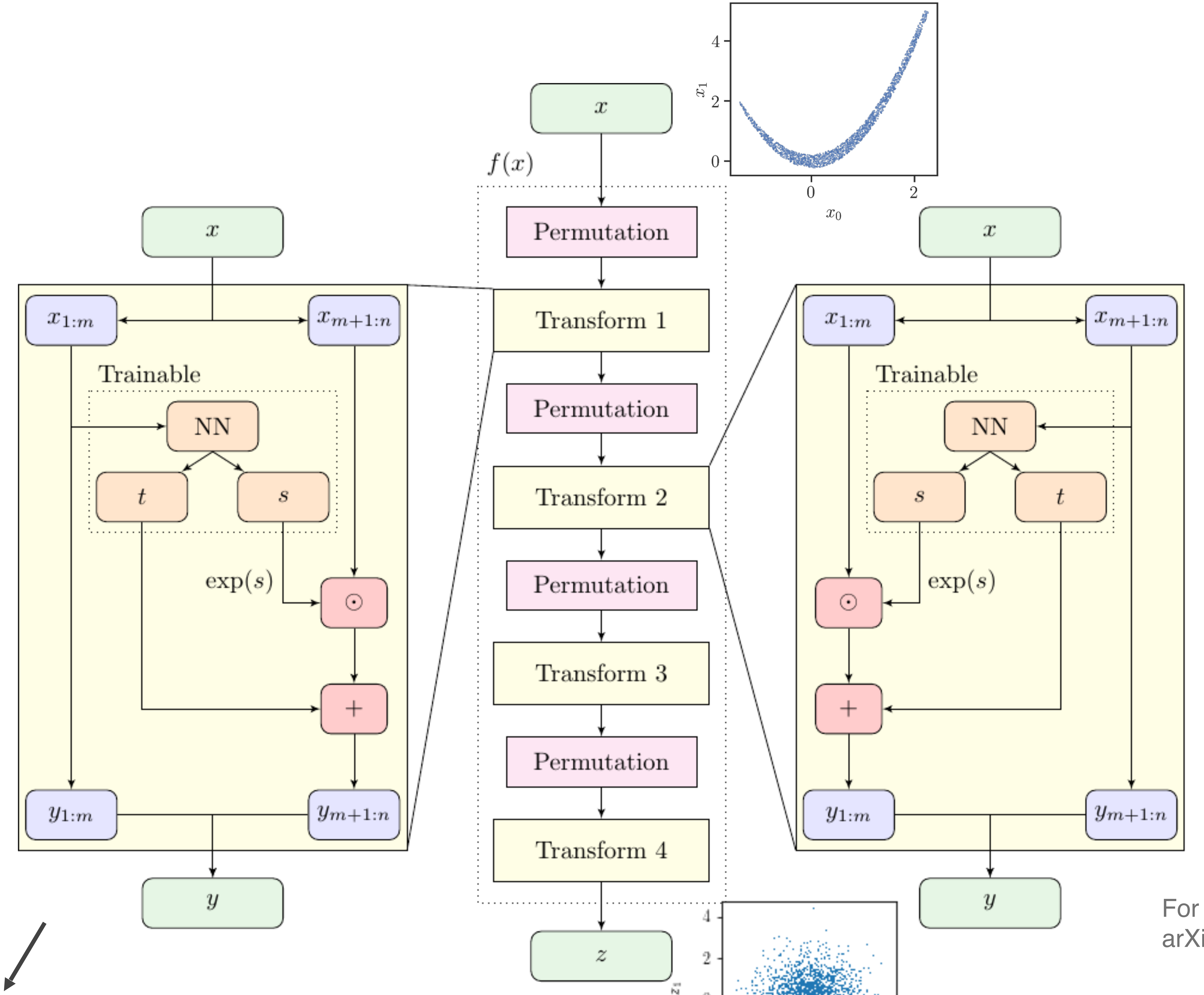
f

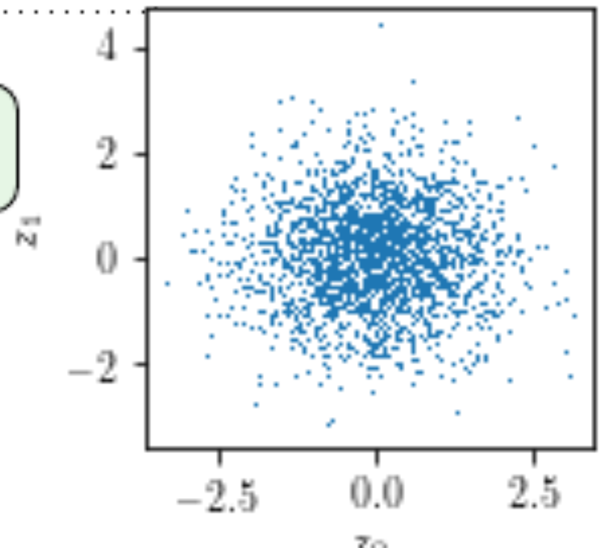
- The mapping has a tractable Jacobian so we can compute the probability of a sample in the physical space:

$$p_X(x) = p_Z(f(x)) \left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right| .$$

- There are different types, we choose to use a version based on affine coupling transforms



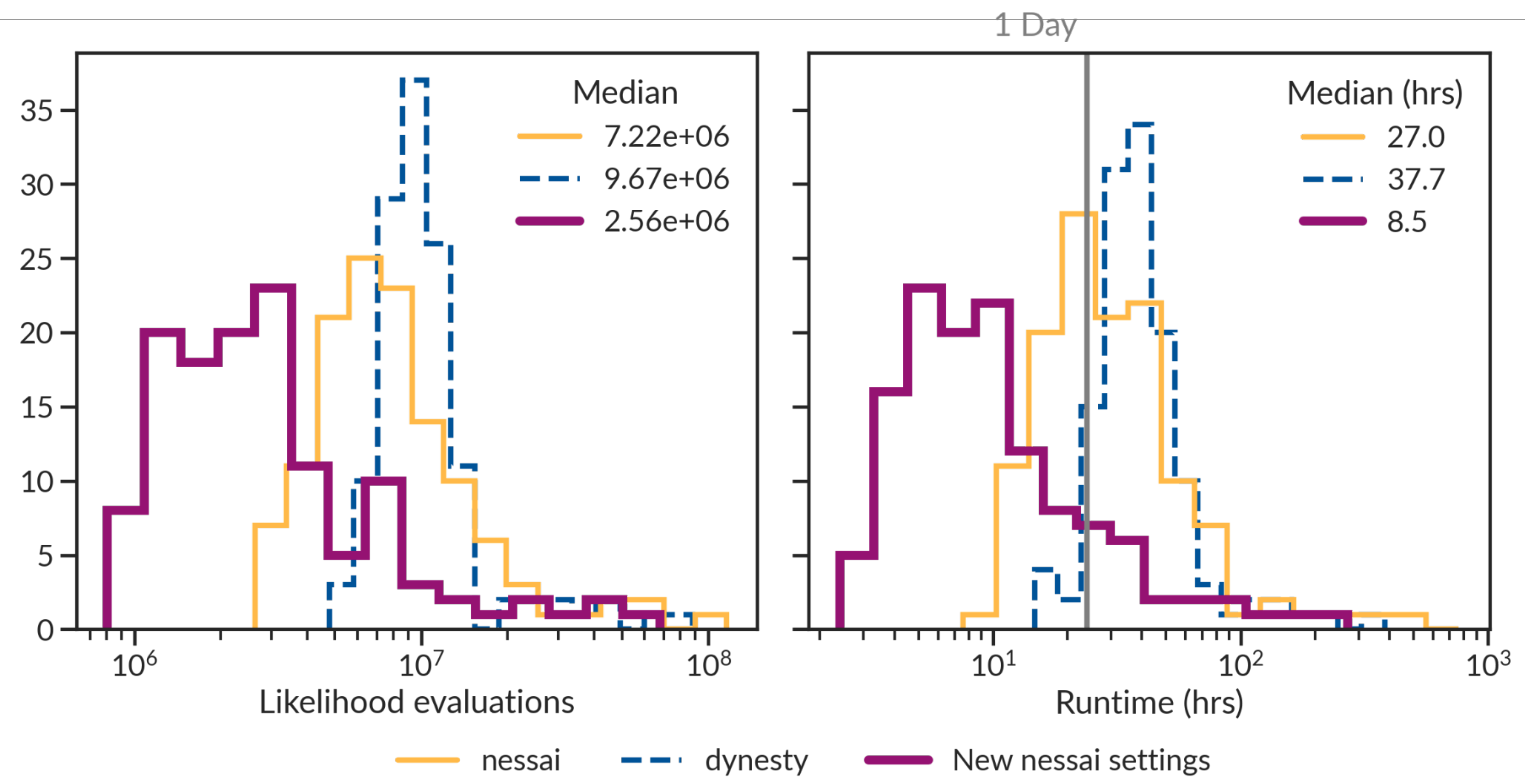




$$\begin{cases} y_{1:m} &= x_{1:m} \\ y_{m+1:n} &= x_{m+1:n} \odot \exp[s(x_{1:m})] + t(x_{1:m}) \end{cases}$$

For details, see arXiv:1605.08803

Acceleration (LVC O3 network)

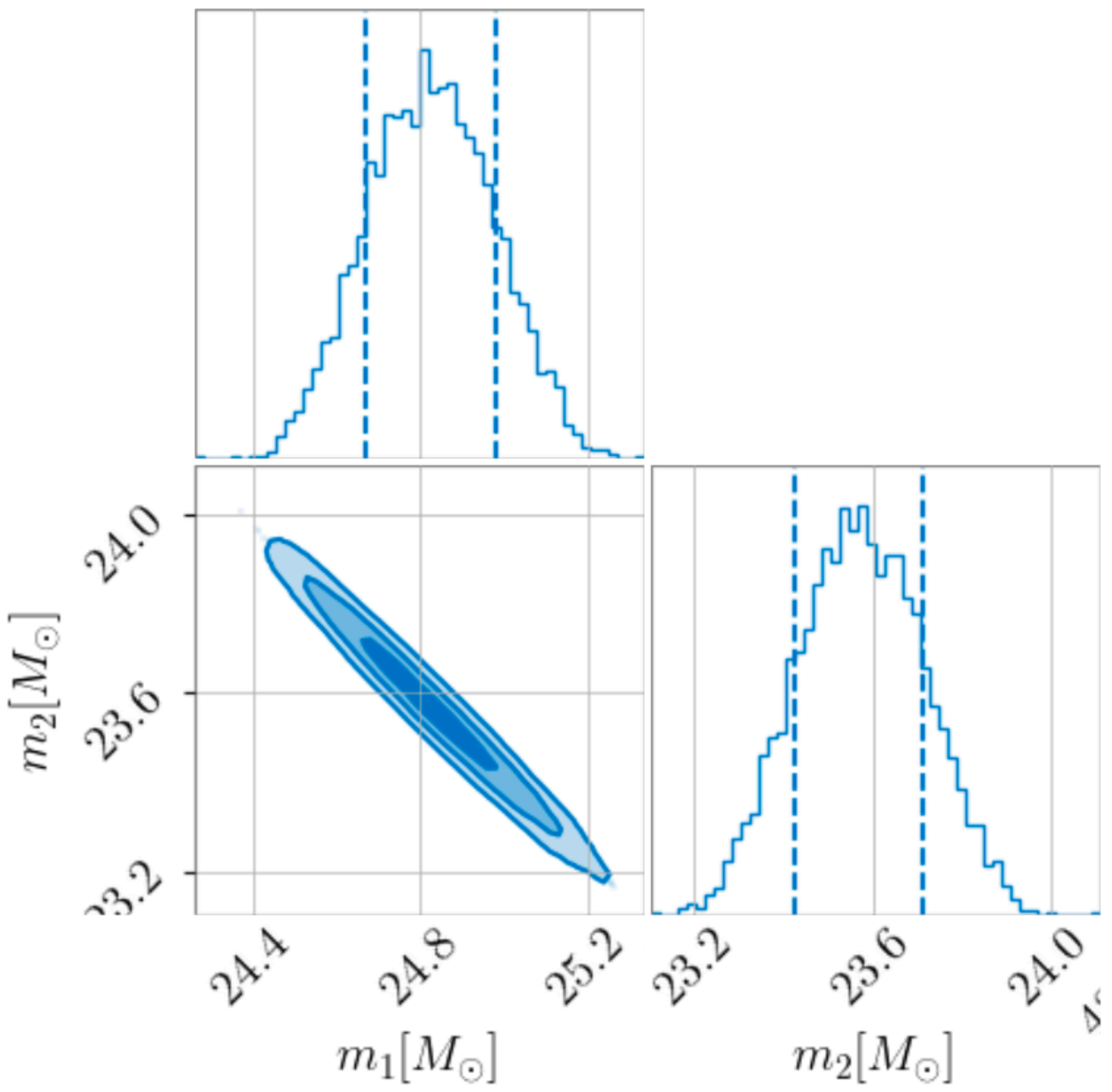


~3x faster than O3b Bilby+dynesty

University of Glasgow

ET example BBH analysis

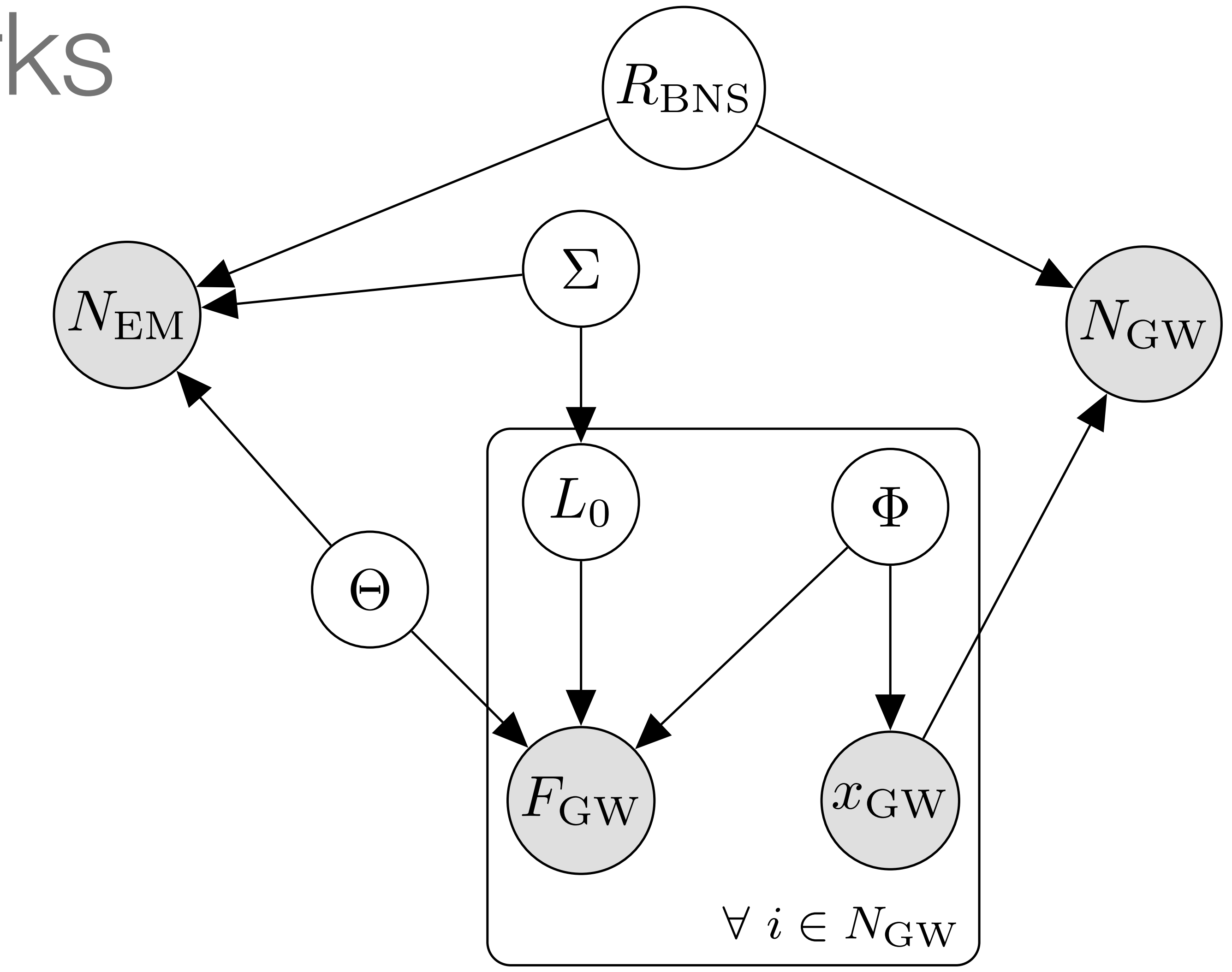
- Analysed MDC1 loudest BBH signal with Bilby+nessai 8
  core [results page]
    - 63s duration from 5Hz, standard likelihood used
    - SNR 588, 50 nats information
    - Run took 9 days 17.5 hours (actually better than expected!)
    - Algorithm slightly over-constrained signal (needs tunir
-



University of Glasgow

Population - level analyses

- Hierarchical inference problems, posed as bayesian networks
- Ingredients:
  - Selection function (estimated from injections)
  - Posterior samples from events
  - Astrophysical / population model (can be slow)
- May include multi-messenger data
- Stochastic sampling methods used here too



A DAG describing a multi messenger population analysis of sGRBs and BNSes
Hayes+ 2023

University of Glasgow

Novel methods

- Explosion in Machine Learning methods in last 5 years
  - CNNs, RNNs, CVAEs, GANs, Normalising flows, diffusion models, …
- Many off-the-shelf techniques work for images or text, but GW applications usually require some customisation
- Enabled by and enables GPU computing as a general tool
  - Tensorflow, PyTorch, JAX main toolkits used in GWs so far
  - Python-driven with CUDA/C/Fortran backend
  - Can offers speedups of 1000x for certain problems
  - Other problems can be re-cast into GPU-friendly forms

ML-Enhanced Analyses

- Emulation via neural network or similar
  - Waveforms [e.g. Thomas+2022]
  - Selection function [e.g. Gerosa+2020]
  - Background estimation for searches [Baker+ 2015, Kim+2015, Kapadia+2017, Kim+2020]
- ML to improve stochastic samplers
  - e.g. Nessai for nested sampling [Williams+ 2021]
  - MCMC w/ normalising flows [Ashton+ 2021, Wong+ 2023]
  - Variational Inference
- ML on output of searches
  - Random forest, simple NNs
- GPUification of core algorithms
  - ✅ FFTs for CBC, CW searches
  - Waveform generation with CUDA (not all waveforms are amenable)
  - GPU-based probability toolkits (e.g. tensorflow probability, Pyro) can accelerate population inference

ML-native analyses

- Deep learning classifiers for detection
  - DNNs/CNNs Gabbard+2018, George+2018, Huerta+2021, Schäfer+2022
  - Still mostly limited to short duration signals and high false alarm rates
- Bayesian Inference algorithms
  - Likelihood-free inference (e.g. Vitamin [Gabbard+2022], DINGO [Dax+2021])
  - (Conditional, continuous) normalising flows shown to work for overlapping signals
  - Training takes days - weeks, inference seconds or less!
- None of these have been shown to work for long signals (>8s?). Frontier needs pushed back for 3G analysis.
  - What is the actual limitation? SNR per sample? Dimensionality?

CBC Landscape

- Current algorithms can easily *detect* ET signals (see previous MDCs).
    - 3G sensitivity motivates expanded searches: spin precession, tides, eccentricity. **These are the most interesting systems**
    - Template banks explode with additional dimensionality and longer signals
    - **Optimally** detecting signals requires very long filters, accounting for Earth rotation. IMO exhaustive search probably not the way forward.
- ML methods more efficient, but not as sensitive (yet). Useless at long signals.
    - Matched filter compresses the SNR into a small number of d.o.f - use as convolutional input?
- Parameter estimation:
    - Techniques exist for long signals with sampling algorithm but haven't yet been tested on 3G data fully
    - Models need to be expanded to include additional physics
    - ML methods look very promising on several fronts, but none fully ready yet

Discussion

- Computing model: HTC model used so far, will it continue? Move to cloud?
- Custom hardware e.g. FPGA? Not much uptake for current analyses.
- Pinning down numbers - MDC to gather stats
- Astrophysics interface - what are the highest priorities, if we can't do everything?
- Numerical Relativity - requirements for better accuracy require vastly expensive simulations
- Looking even further ahead - new tech, e.g. quantum computing?