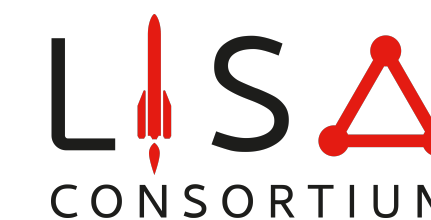
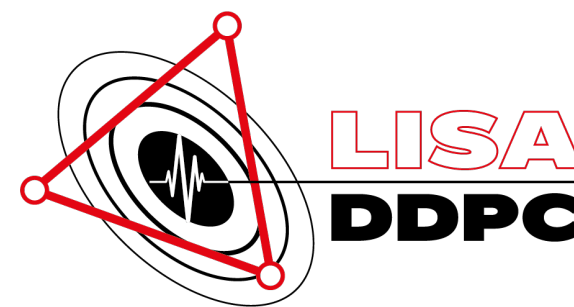


Rapid detection and parameter estimation of **Massive Black Hole Binaries** for **LISA**

Malvina Bellotti, Phd student at the Observatoire de la Côte d'Azur, Nice

Supervisor: **Natalia Korsakova**, Observatoire de la Côte d'Azur, Nice

Co-supervisor: **Jess McIver**, University of British Columbia, Vancouver

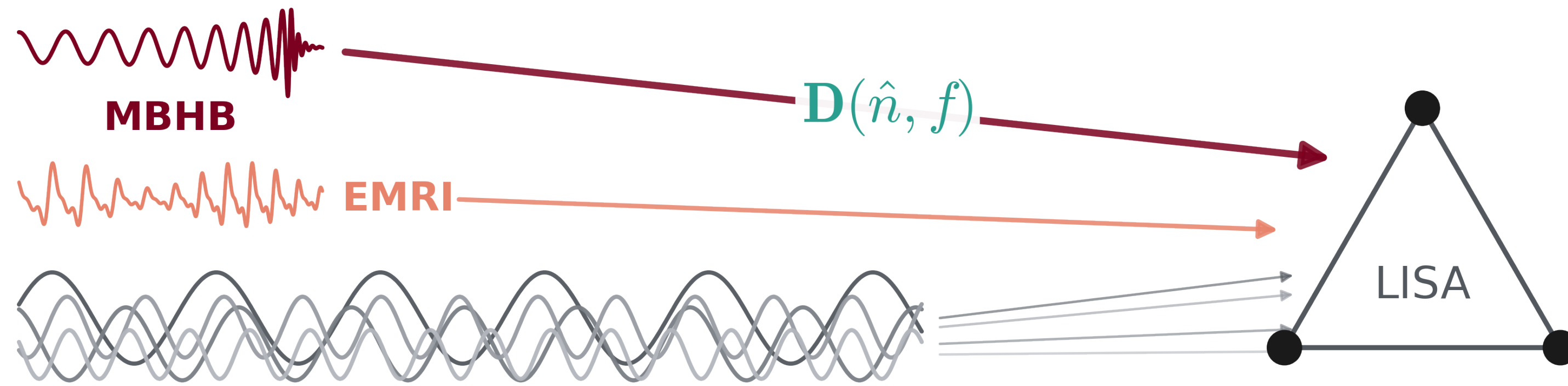
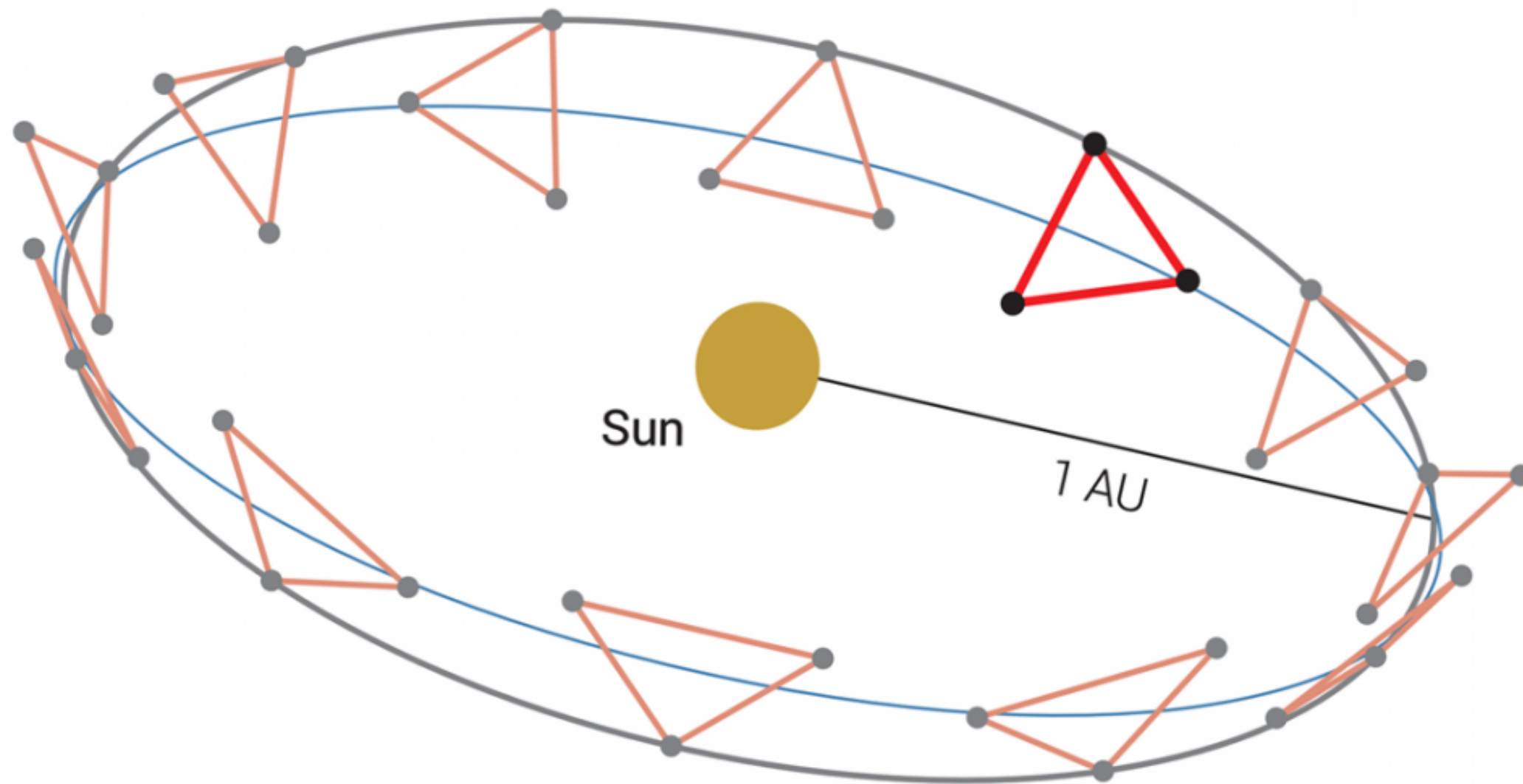


LISA and the cocktail party problem

ESA space-based detector:

- launch in 2035
- band: 0.1 mHz - 1 Hz
- Three spacecraft in triangular constellation, 2.5 million km arms, heliocentric orbit

Sources: MBHBs, EMRIs, stellar BH binaries, galactic binaries



galactic binaries: tens of millions

$$\underbrace{\tilde{d}_I(t, f)}_{\text{channel I (X, Y, Z)}} = \sum_k \underbrace{\mathbf{D}_I(\hat{n}_k, t, f)}_{\substack{\text{detector response:} \\ \text{sky position, frequency}}} : \underbrace{\tilde{\mathbf{h}}_k(t, f; \xi_k)}_{\substack{\text{source k,} \\ \text{parameters } \xi_k}} + \underbrace{\tilde{n}_I(t, f)}_{\text{noise}}$$

The mixing depends on sky position

Detector response changes with sky position and frequency, and with time during LISA orbit. Lets us localise the source.

Two ways to listen:

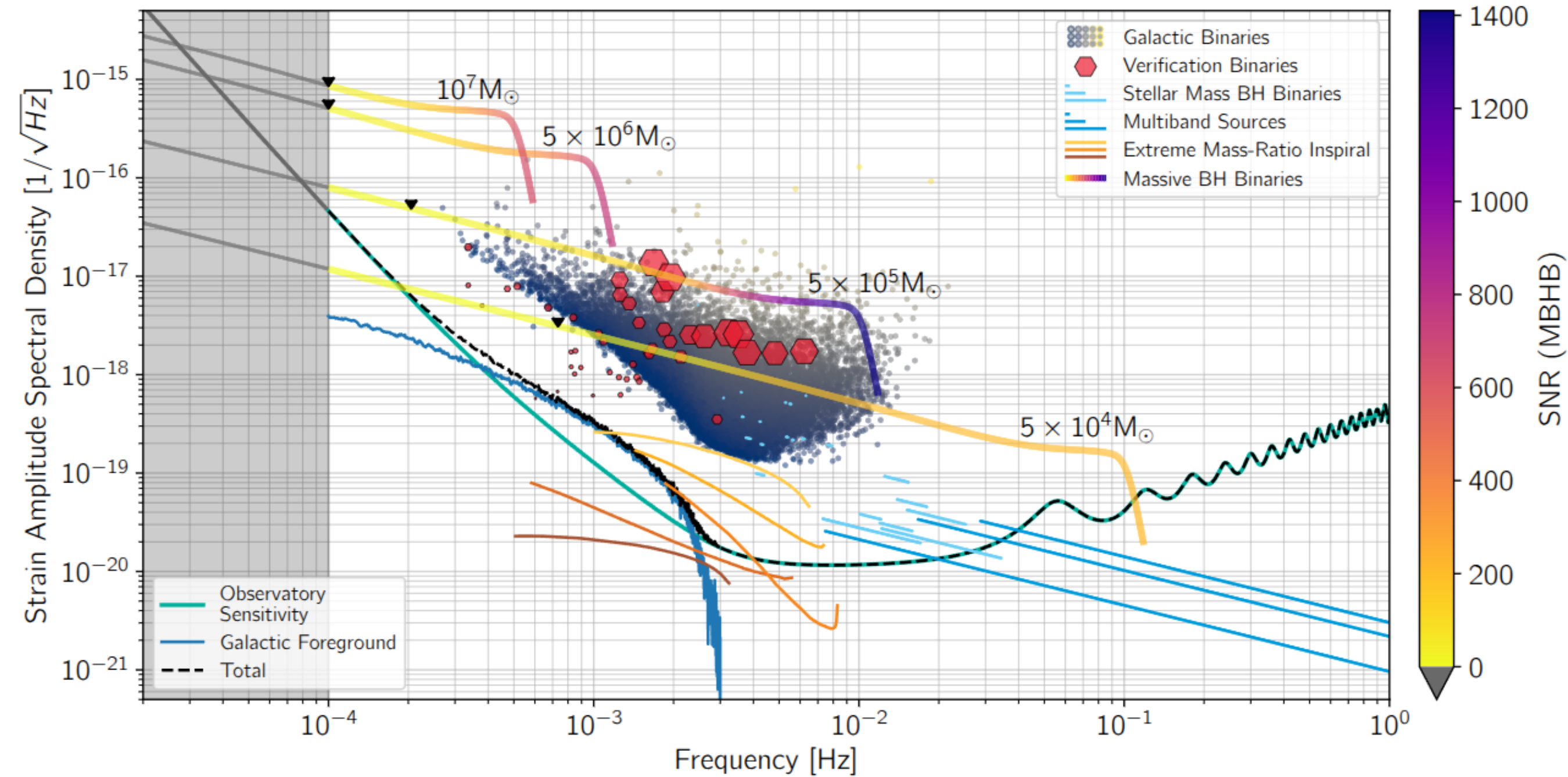
Global Fit: analyze every voice at once. Accurate, but **slow**.

Our networks: estimate MBHB, and treat the rest as background.

Massive Black Hole Binaries

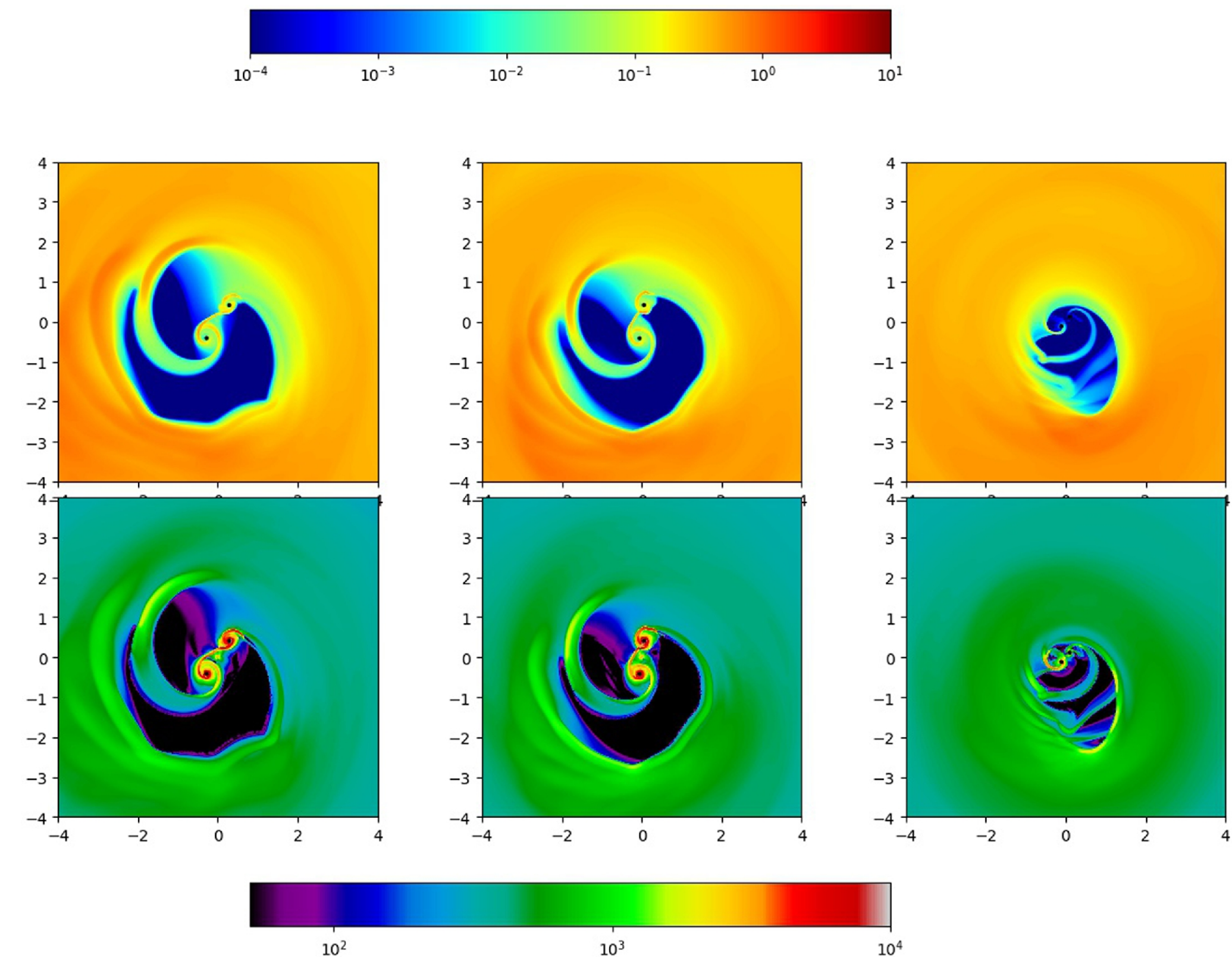
$$10^4 M_{\odot} < M < 10^7 M_{\odot}$$

[Colpi+ 2024]



- Found at the **centre of most galaxies**
- Hierarchical galaxy mergers → **binary** systems
- Enter LISA band **days to months** before merger
- **Loudest LISA sources** → test GR, probe cosmic history, constrain cosmology

In presence of sufficient amount of gas an EM counterpart can be triggered



Tang +18

Low Latency **Alert Pipeline** for **MBHBs**

Real-time detection essential for:

- Trigger **protected periods** in LISA data stream
- **Multi-messenger alerts**: precursor EM emission detection + afterglow surveys
- Trigger **complete parameter estimation** pipelines

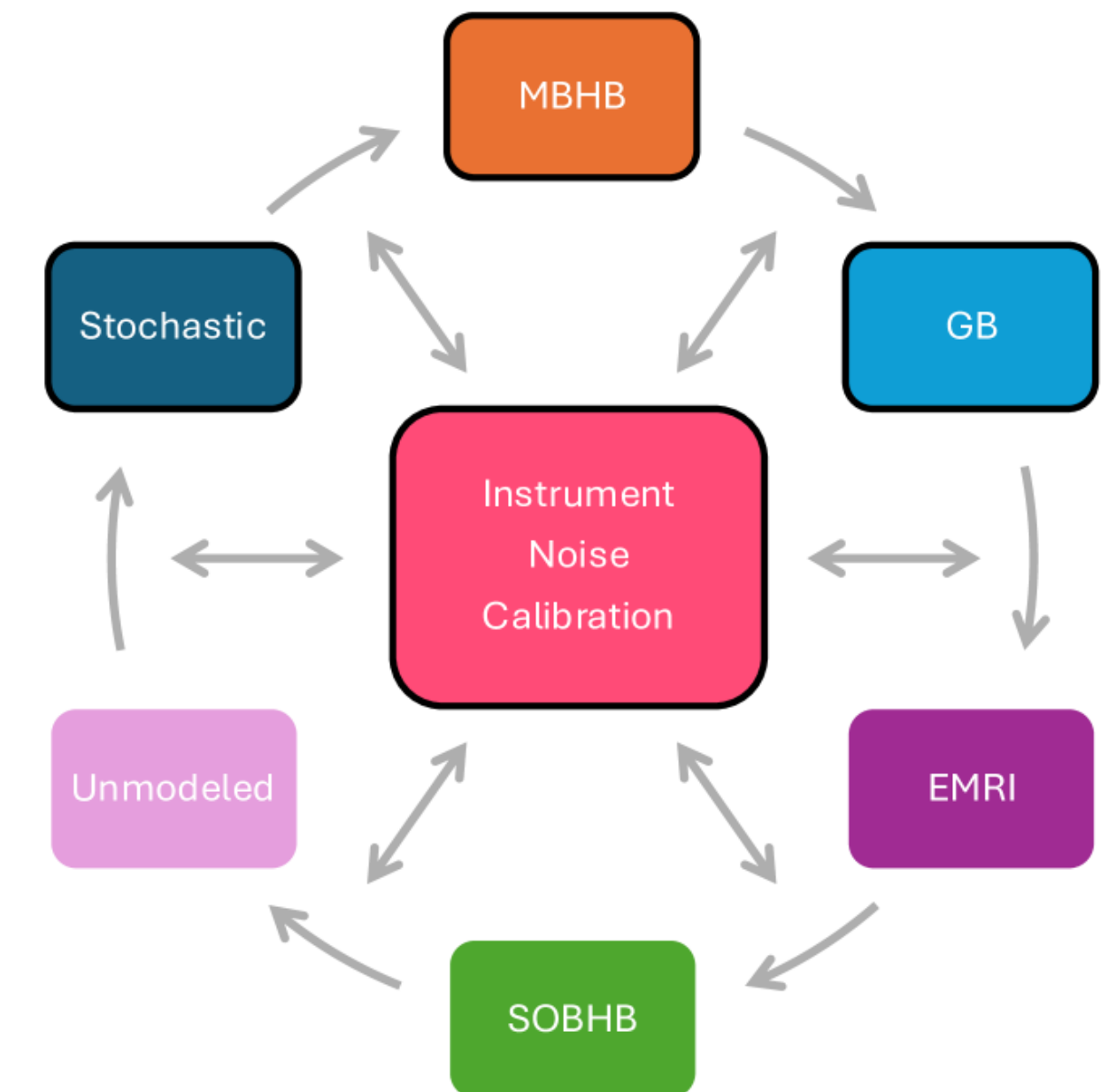
Goal:

analyze transients in **less than 30 minutes** from data reaching the ground

continuously **update parameters** (masses, distance, merger time, sky position)

Focus on **MBHB**! We have to perform analysis in the presence of other sources/realistic noise but we can marginalise over them!

Global Fit is **too slow** for real time



Katz+ 25

Simulation based inference

$$\underbrace{p(y|x)}_{\text{Posterior}} = \frac{\overbrace{p(x|y)}^{\text{Likelihood}} \underbrace{p(y)}_{\text{Prior}}}{\underbrace{p(x)}_{\text{Evidence}}}$$

$$p(y|x) = \frac{\overbrace{p(y,x)}^{\text{Joint Distribution}}}{p(x)}$$

Allows to avoid likelihood calculations!

simulate a training dataset of parameters from the prior and their corresponding observations

train a neural network on simulated data to learn the shape of posterior distribution

$$x = h(y) + \text{noise} + (\text{glitches/gaps})$$

Samples from joint distribution $p(y, x)$

We do:

Conditional posterior estimation with generative models.

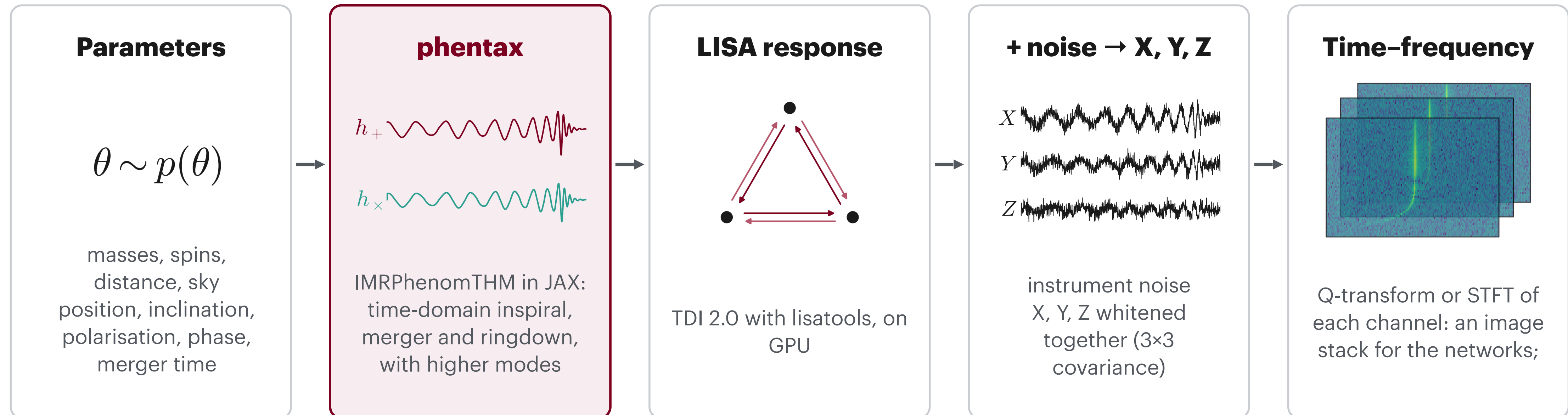
At the moment implementations for Normalising Flows and Flow Matching.

Condition inverted map on real data:

amortised inference $\rightarrow$ train once, then run **inference in seconds** when new data arrives.

Data production

Mojito-based waveforms



On the fly

Simulate during training:
every step new batch

Galactic binaries

GPU-accelerated GBGPU
waveforms

We use this data as input for two different pipelines

Point parameter estimation

To provide starting points for
Global Fit

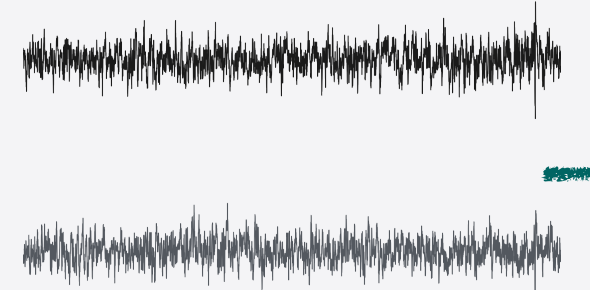
Conditional flow matching

full posteriors

Detection and point estimation pipeline

with Aryamann Rao & Jess McIver (UBC)

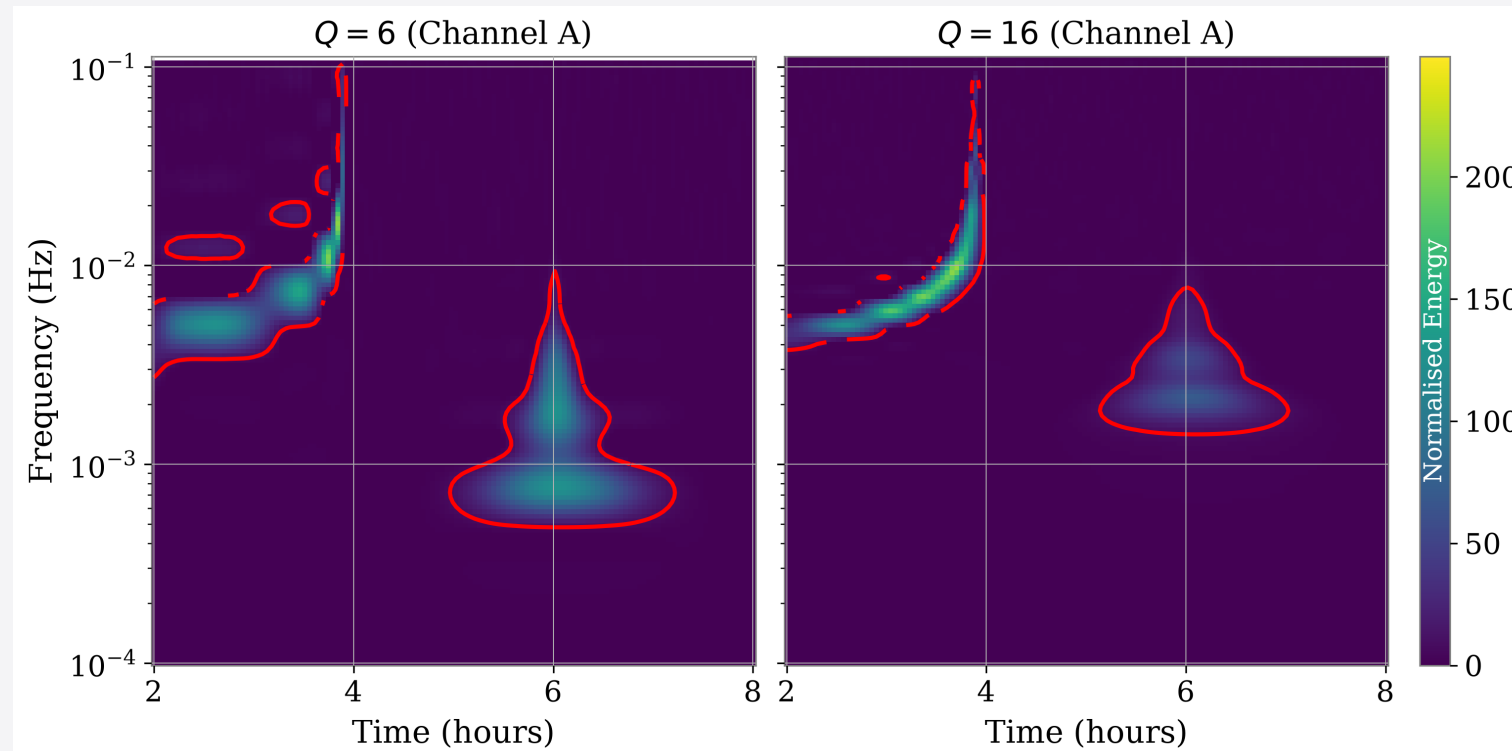
LISA data



TDI X,Y,Z
Mojito Light

1 • DETECTION

Excess power in the Q-transform



[Rao et al, in preparation]

first estimates of
chirp mass and
coalescence time

$\mathcal{M}_c, t_c$

**(2,2) chirp fit, with
Viterbi algorithm**

flags a candidate
MBHB

Look out!
Paper on search ad classification of glitches vs MBHBs
Rao et al. (with Bellotti):
On arxiv very soon!!

2 • PARAMETER ESTIMATION

Pick the CNN

CNN · mass range 1

CNN · mass range 2

⋮

CNN · mass range k

⋮

CNN · mass range K

one network per mass
range;
data cut around t_c

Point estimates

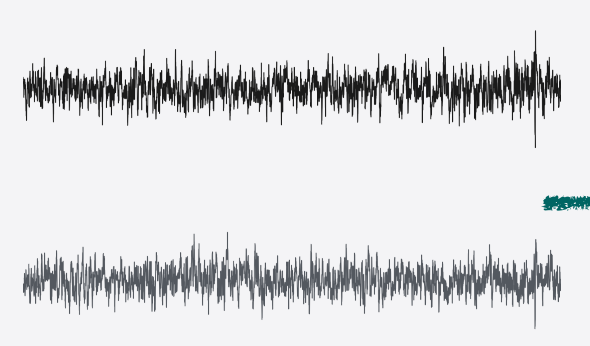
m_1, m_2, D_L, t_c
 $\lambda, \beta, \iota, \psi, \phi_c$

Parameters can
serve as initial
guess for the
Global Fit

Detection and point estimation pipeline

with Aryamann Rao & Jess McIver (UBC)

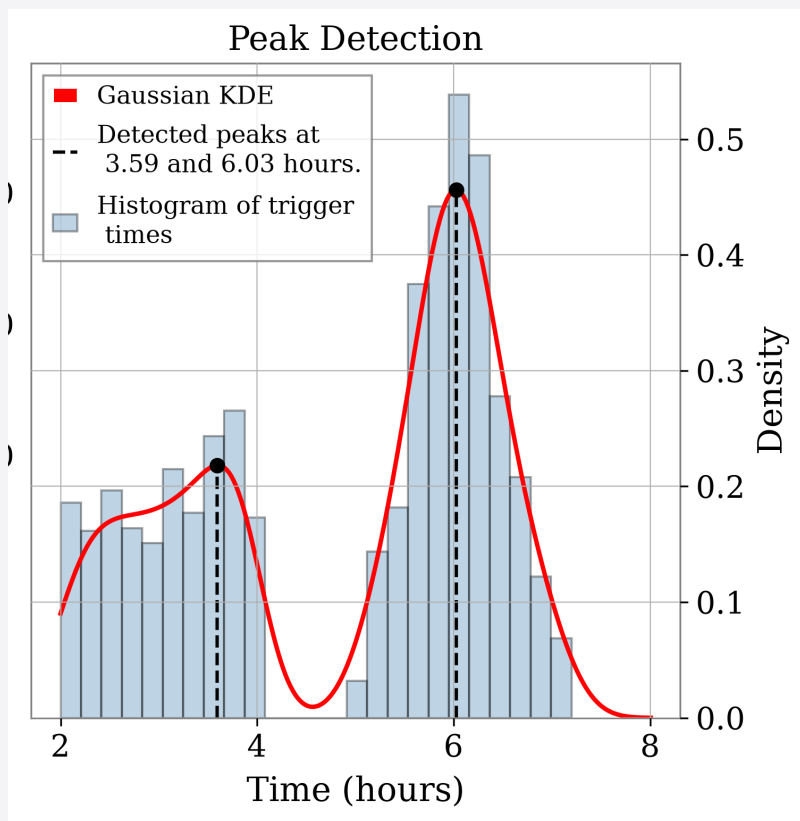
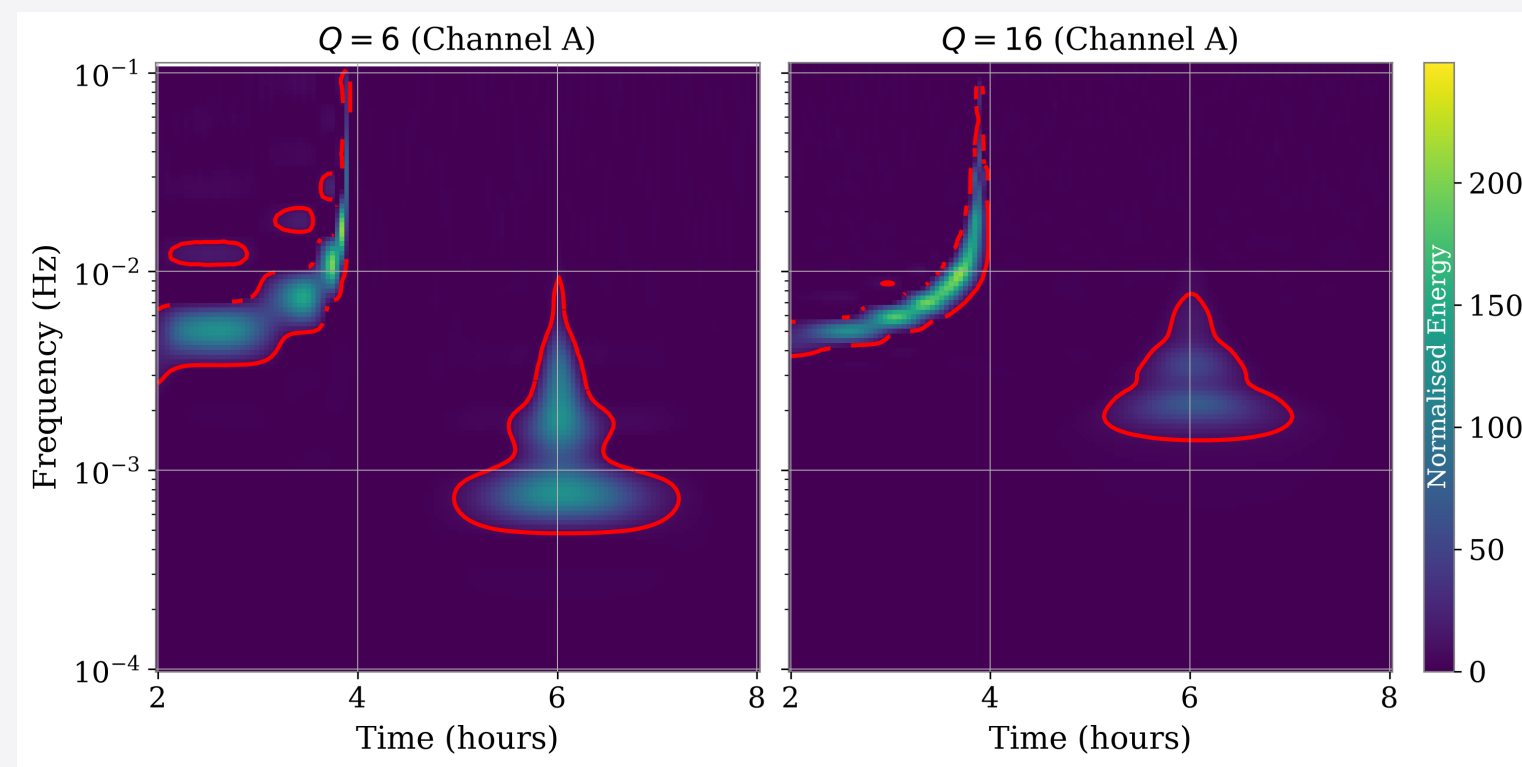
LISA data



TDI X,Y,Z
Mojito Light

1 • DETECTION

Excess power in the Q-transform



[Rao et al, in preparation]

flags a candidate
MBHB

(2,2) chirp fit, with Viterbi algorithm

$$\mathcal{M}_c, t_c$$

first estimates of
chirp mass and
coalescence time

2 • PARAMETER ESTIMATION

Pick the CNN

- CNN · mass range 1
- CNN · mass range 2
- ⋮
- CNN · mass range k**
- ⋮
- CNN · mass range K

one network per mass
range;
data cut around t_c

Point estimates

$$m_1, m_2, D_L, t_c$$
$$\lambda, \beta, \iota, \psi, \phi_c$$

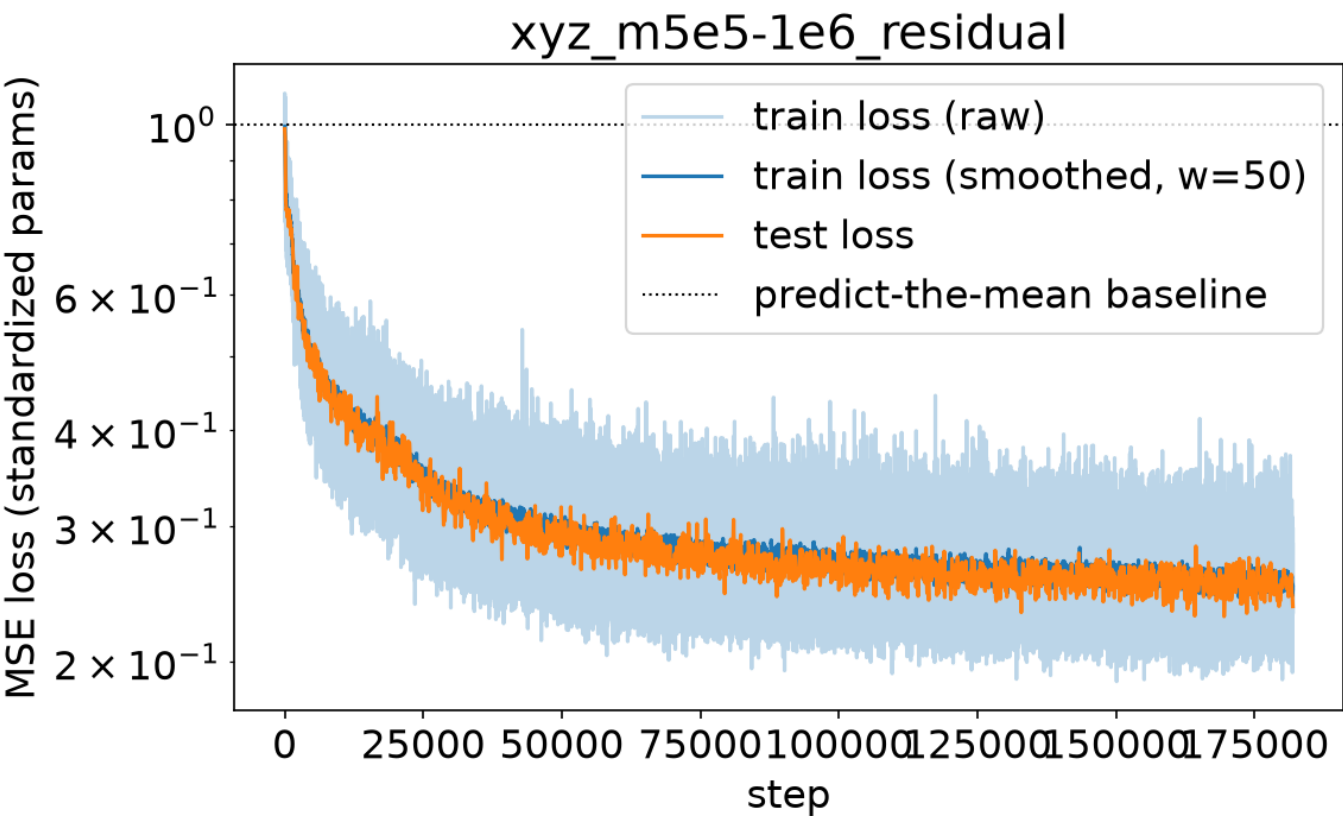
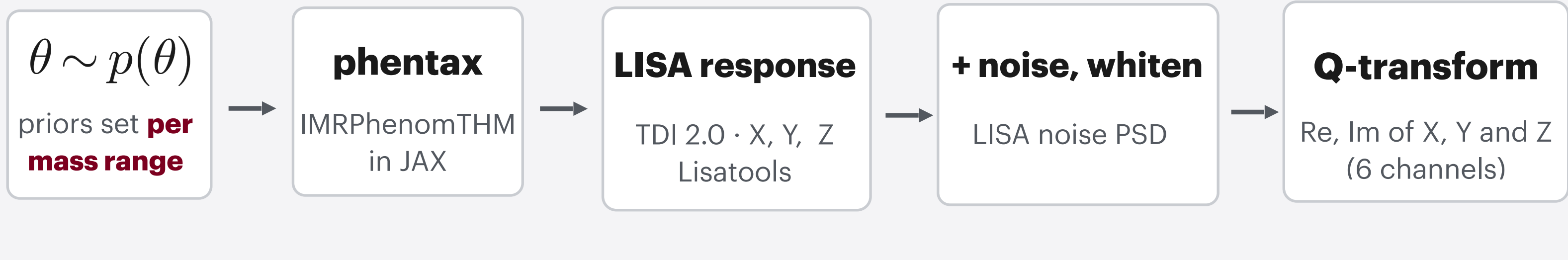
Parameters can
serve as initial
guess for the
Global Fit

Look out!

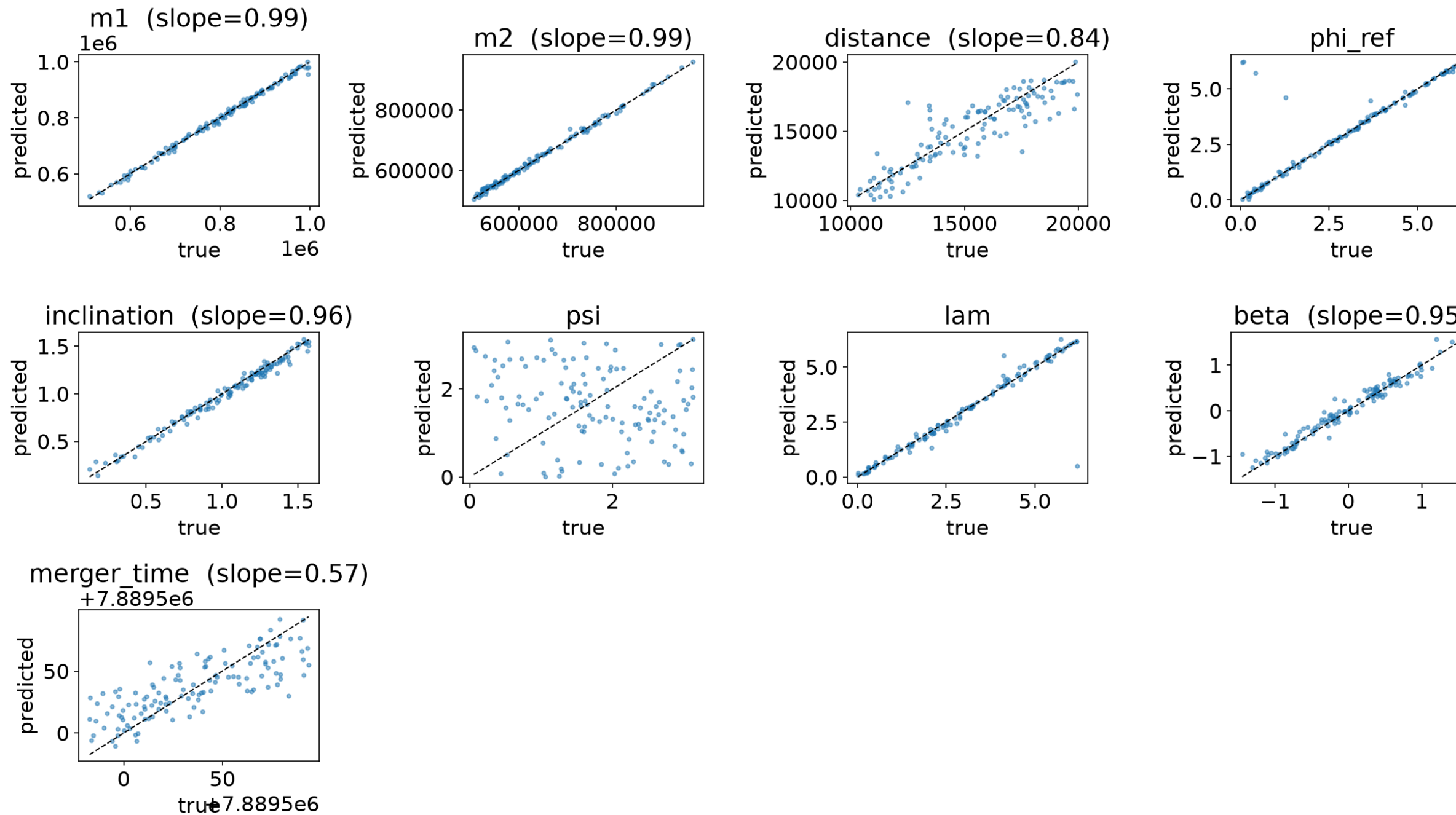
Paper on search ad classification of glitches vs MBHBs
Rao et al. (with Bellotti):
On arxiv very soon!!

Point estimation: a simple **CNN** trained on the fly

SIMULATE · ON GPU, A NEW BATCH AT EVERY STEP



LEARN



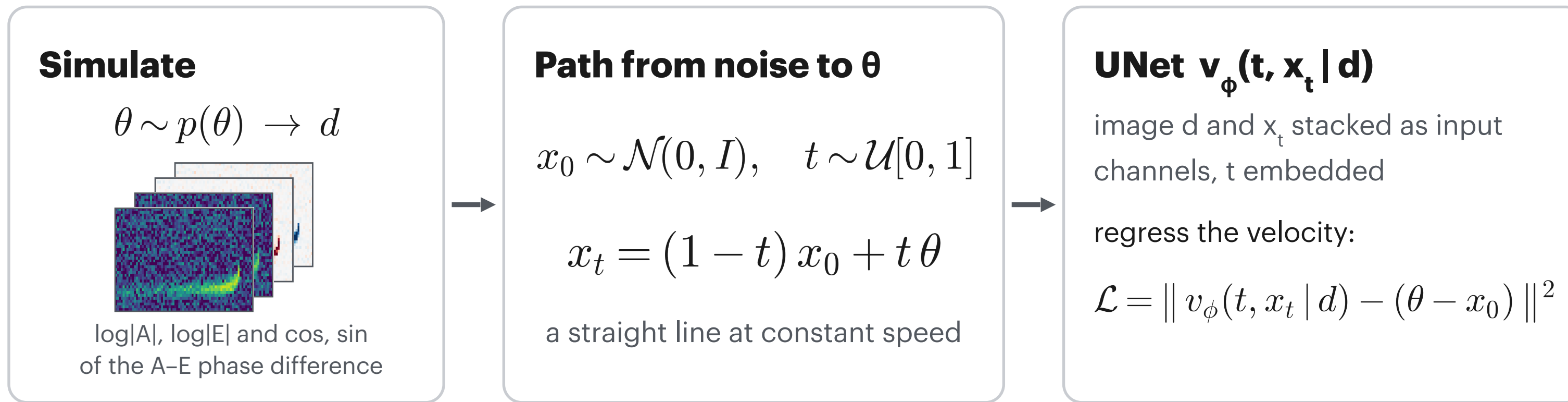
$\hat{\theta} \approx \mathbb{E}[\theta | d]$ MSE makes the CNN return the posterior mean: fast and simple, but no error bars, no correlations

Full posteriors with conditional flow matching

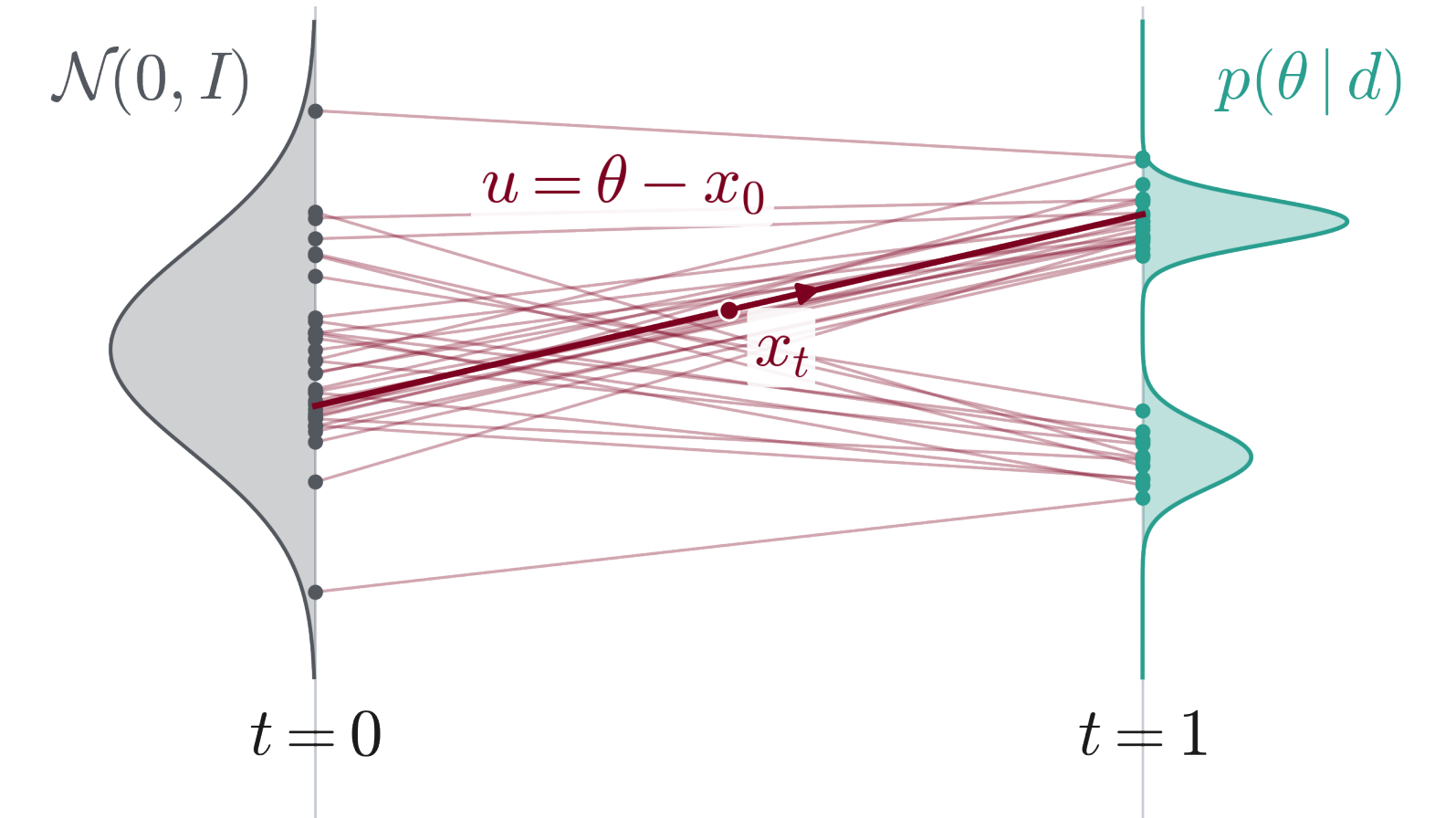
Flow matching trains a NN to predict the **velocity vector field**

describing a the path from an easy distribution into a complex target distribution (parameters' posteriors)

TRAINING · ONCE, OFFLINE

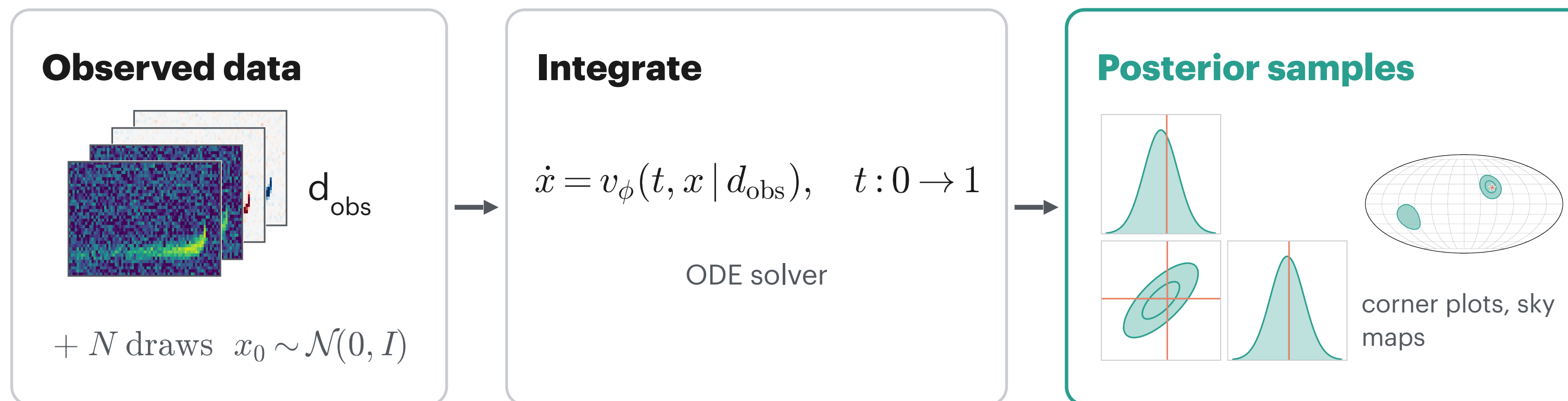


Training pairs: straight lines

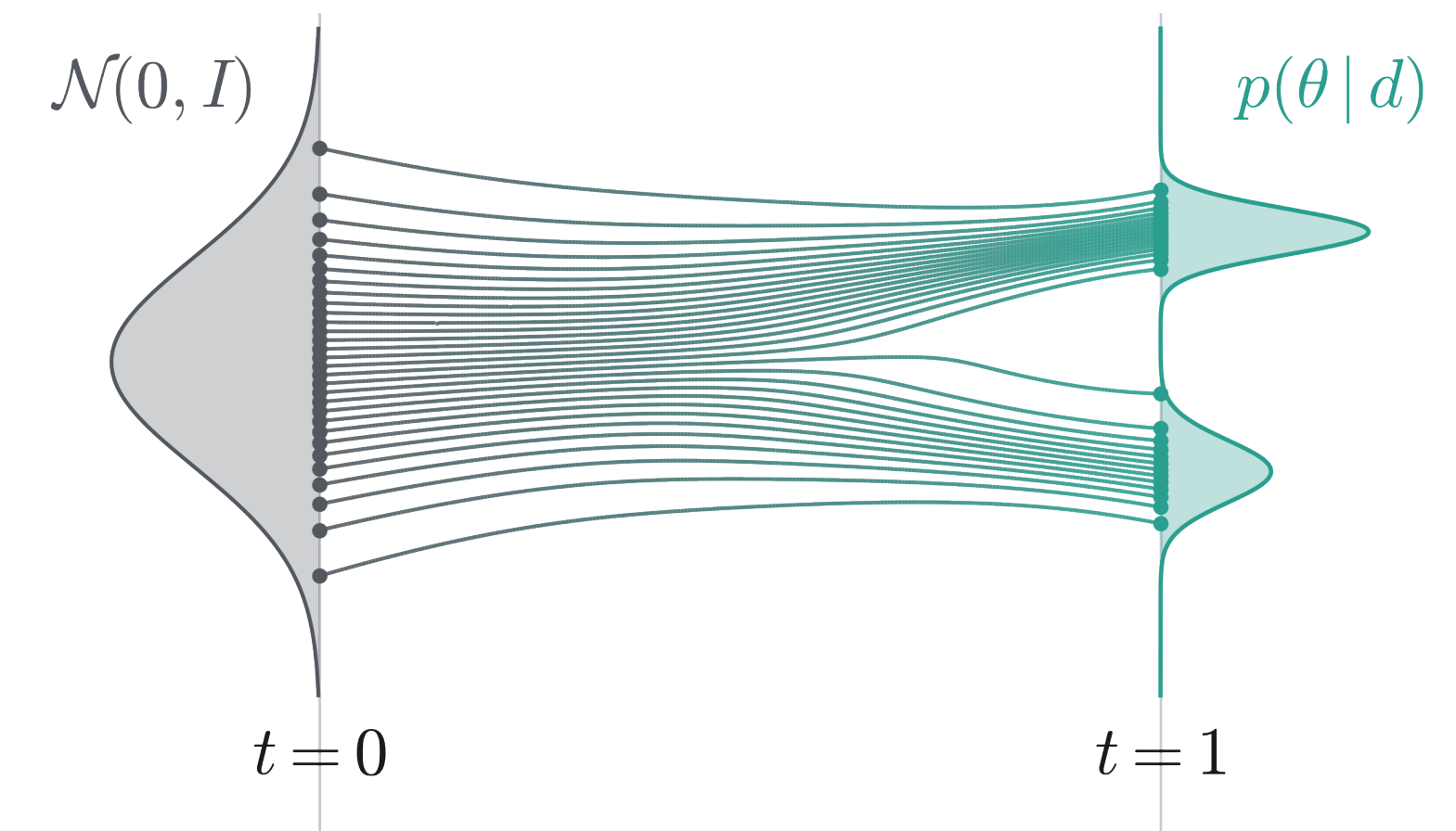


INFERENCE · EVERY NEW EVENT, NO RETRAINING

amortised inference

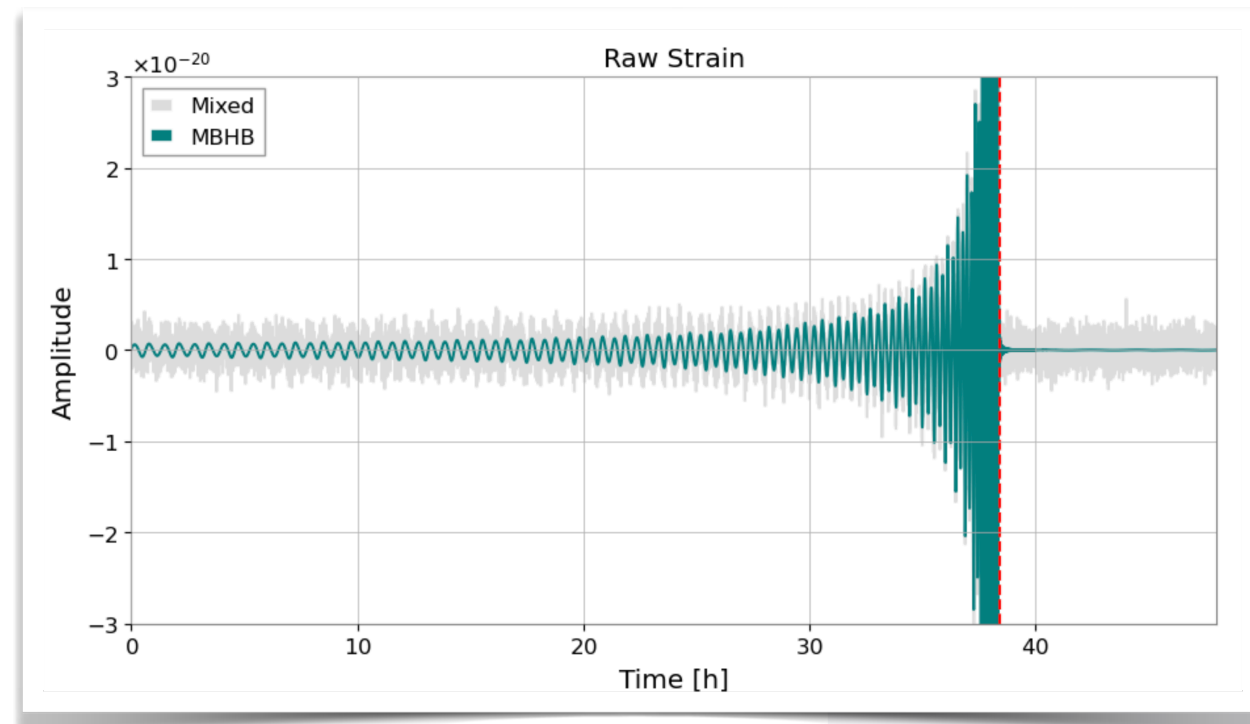


Learned flow:



Data representation

We train and evaluate our network on two data representations:



time series

Training data produced **on-the-fly**

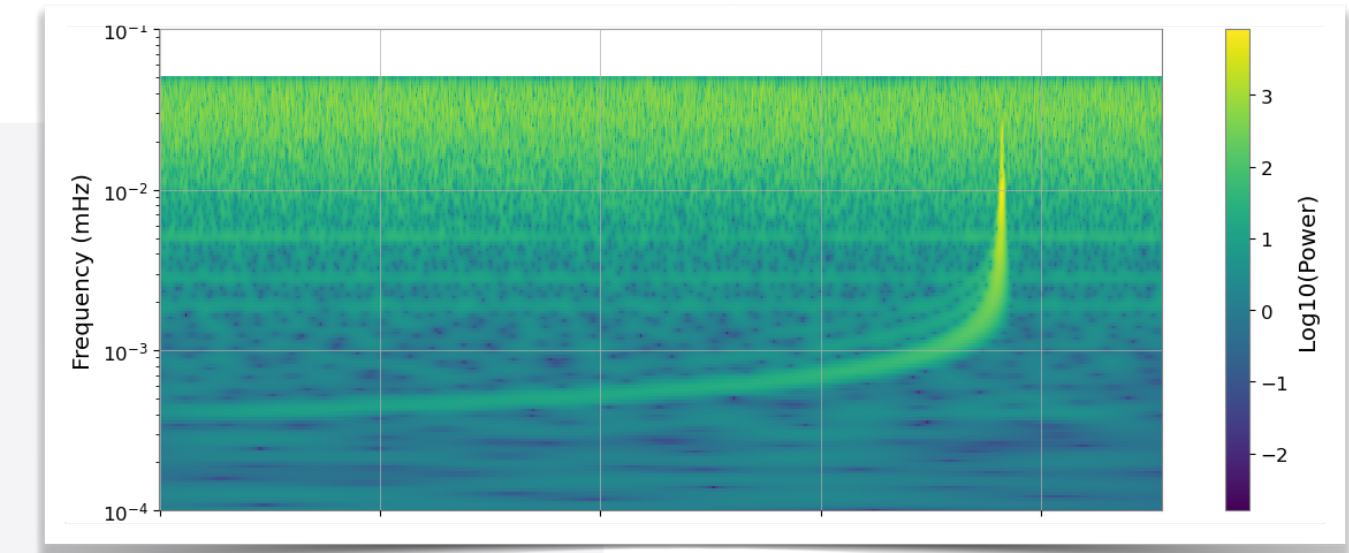
but

NNs could struggle to extract long temporal features from 1D sequences

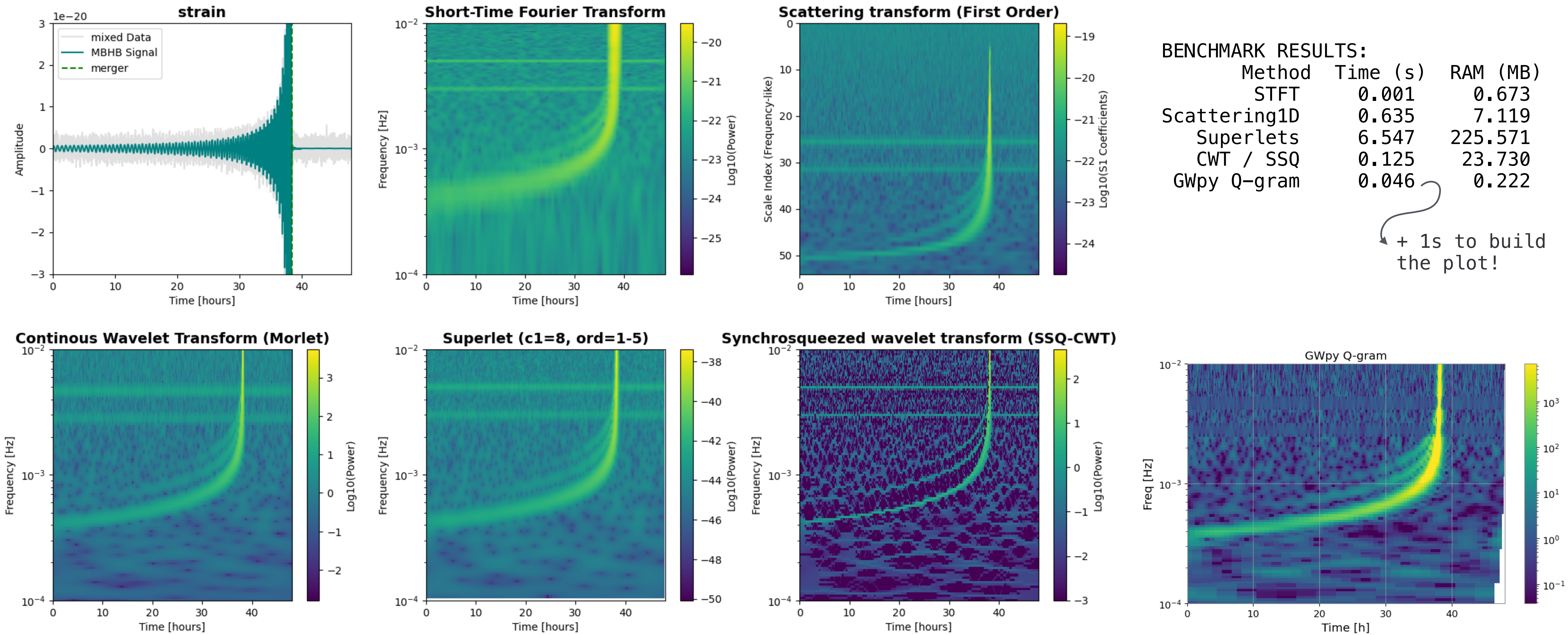
time-frequency

Might be easier for UNet 2D pattern recognition

Exploring different time-frequency representations

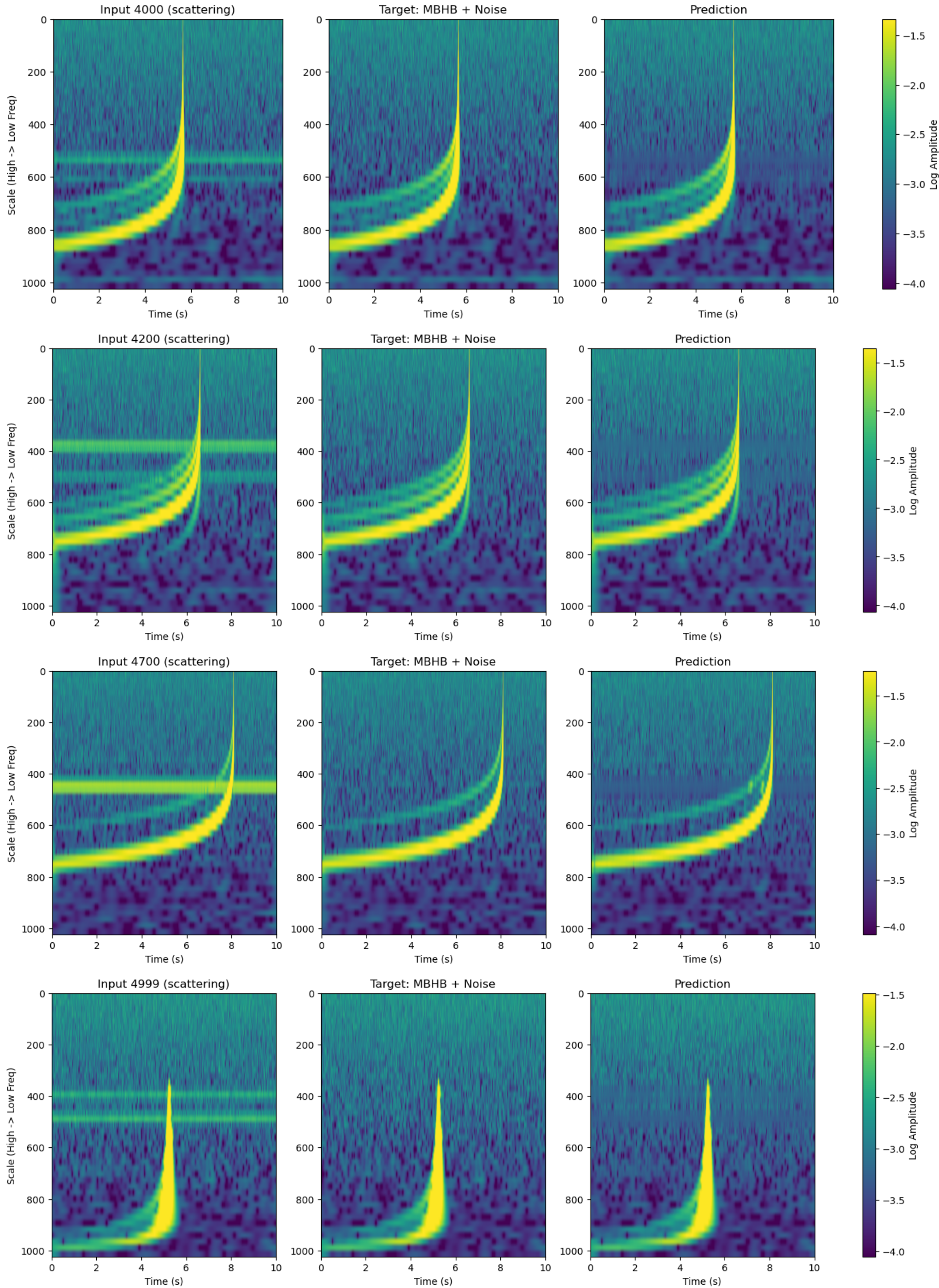
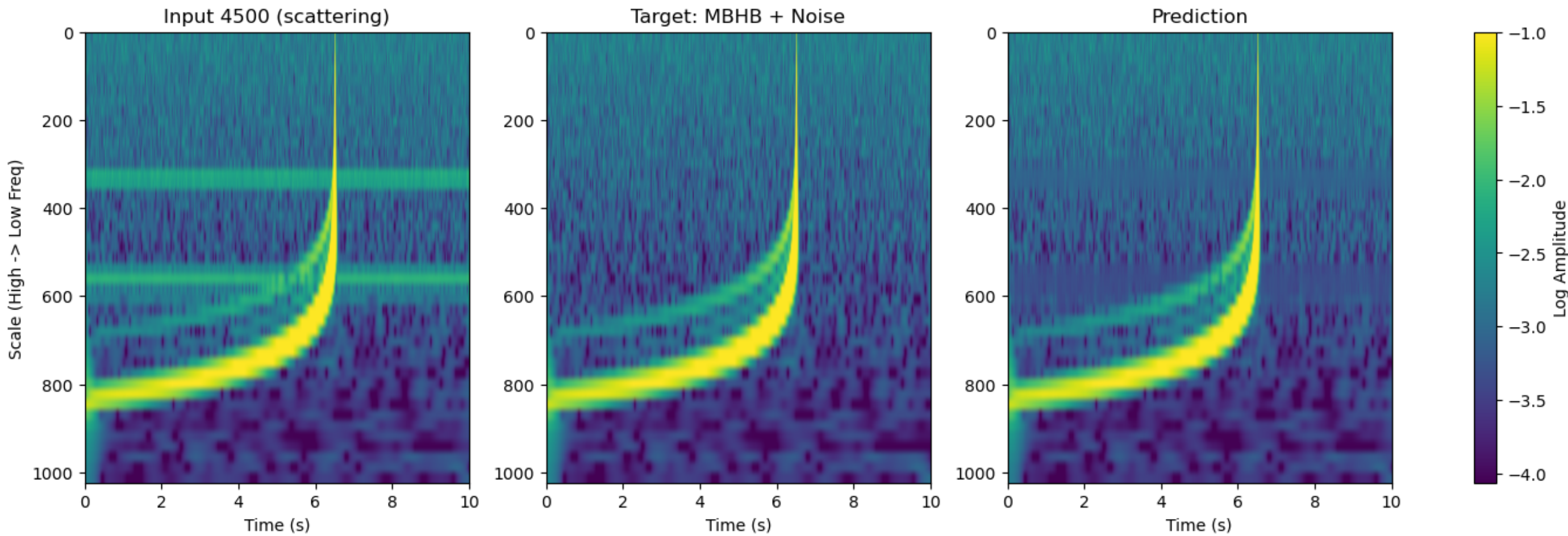
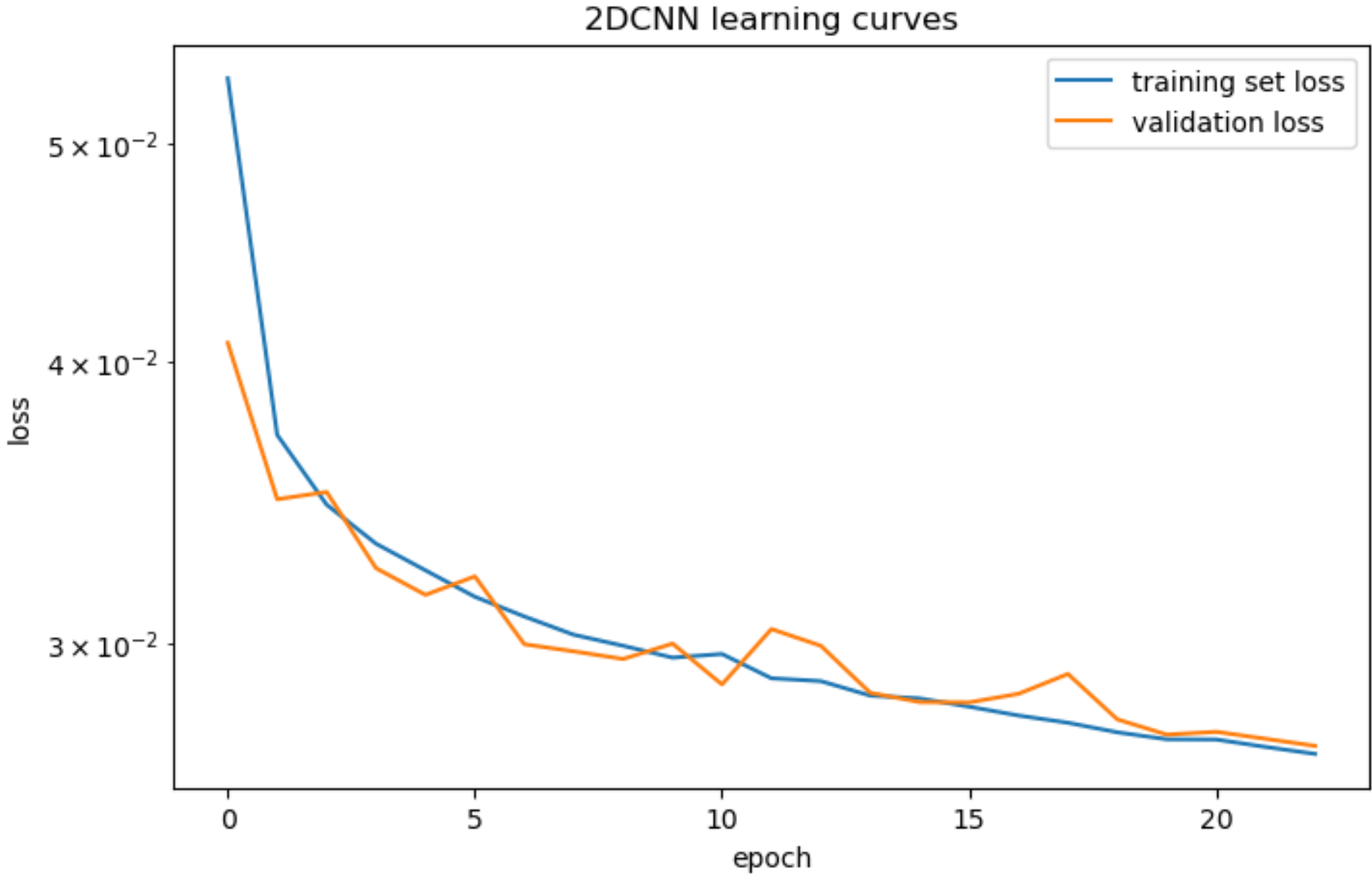


Choosing the optimal time-frequency representation

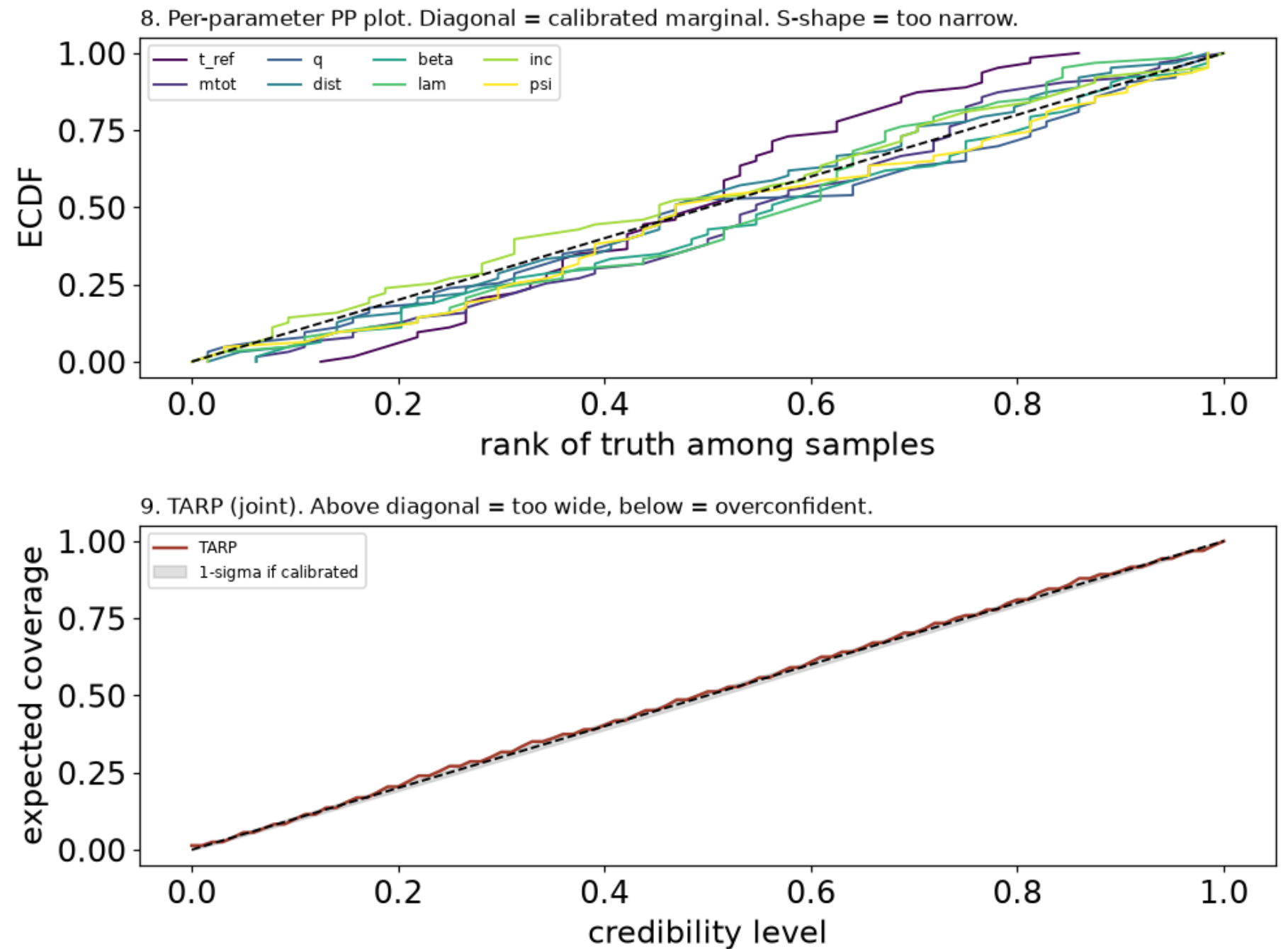
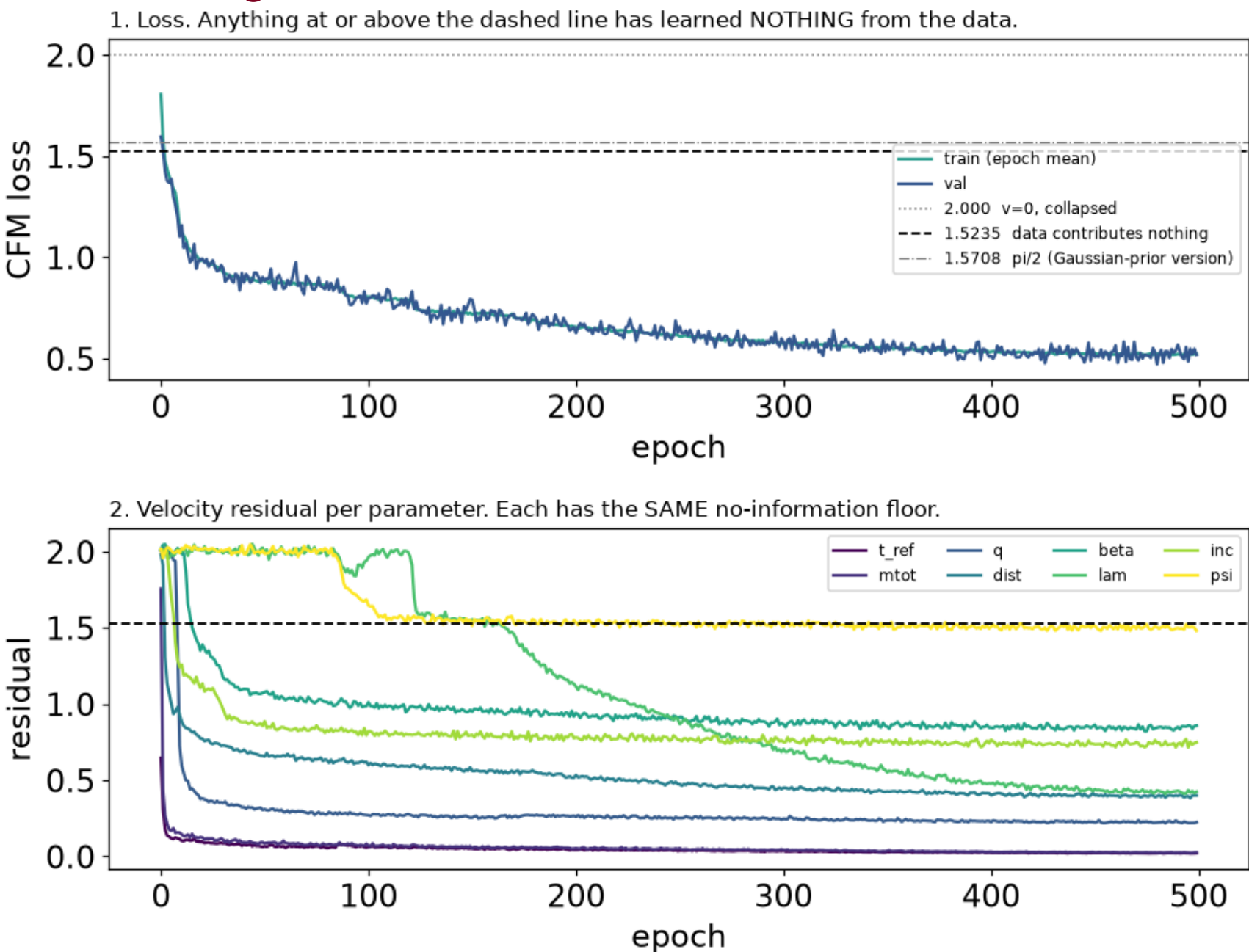
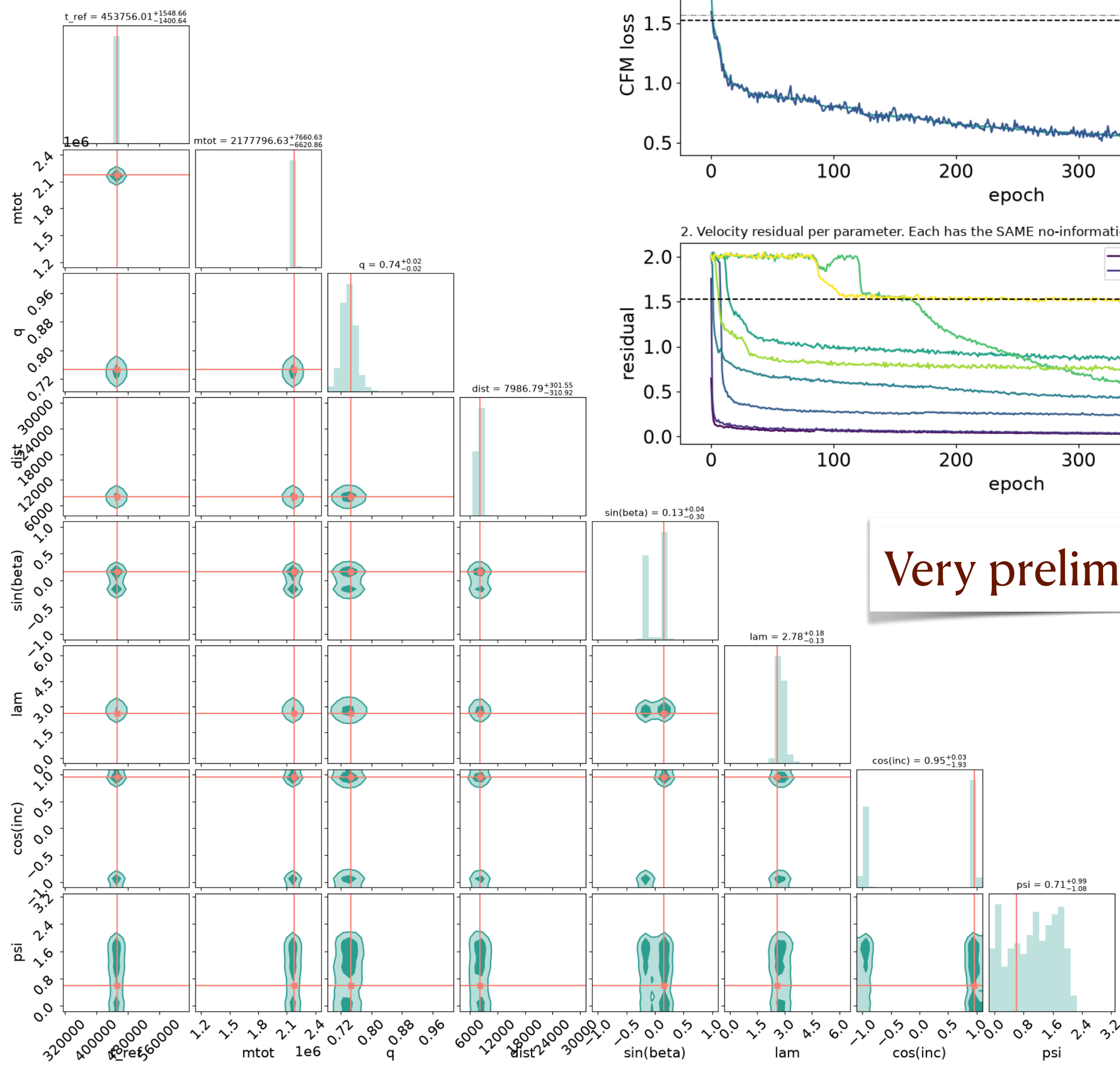


Marginalising out unwanted signals with autoencoders

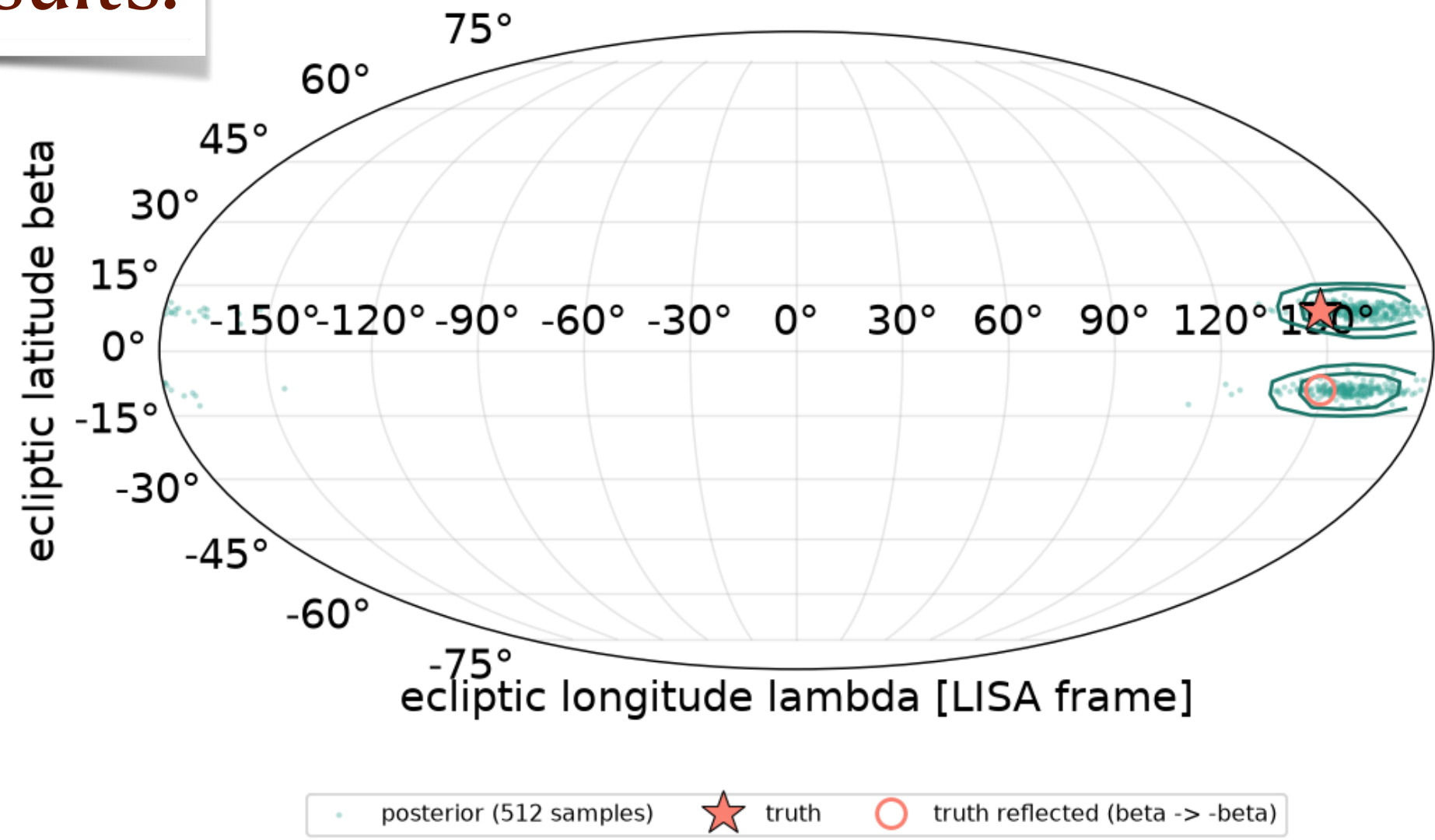
2D CNN
scattering
transform



PE on time-frequency



Very preliminary results!



To conclude

Point parameter estimation

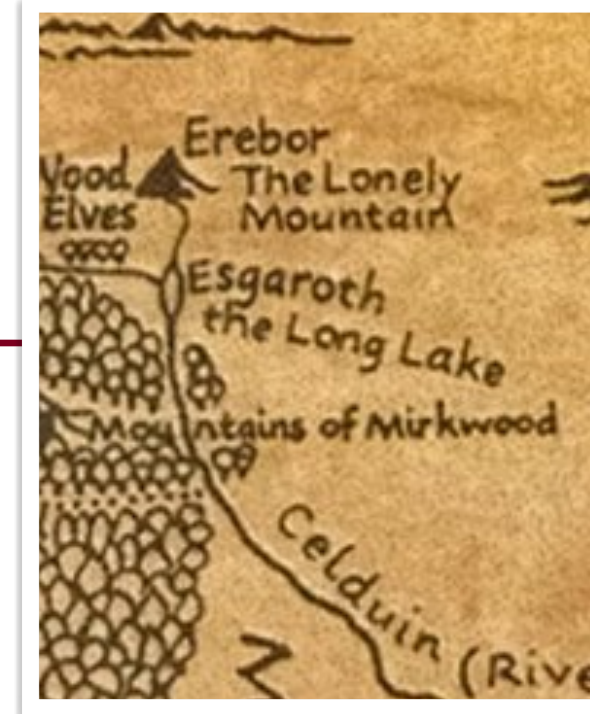
with Aryamann Rao & Jess McIver (UBC)

- **Detection:** excess power in the Q-transform, then a (2,2) **chirp fit** for the chirp mass and merger time
- A bank of **CNNs, one per mass range**, trained on the fly on GPU
- Initial guesses for the Global Fit
- MBHBs detected in Mojito Light; the CNNs are now being tested on these detections
- Point estimates only: the uncertainties come from spread

Conditional flow matching - **Celduin (flowing river)**

full posteriors

- SBI using Conditional Flow Matching and Normalizing Flows allows amortised, **likelihood-free** PE, **useful for low latency!**
- It can naturally **marginalise** over other **signals** and **noise artefacts**
- we use both time domain and **time-frequency** domain → time-frequency representations might better align with our architecture
- Exploring autoencoders to remove the galactic foreground



Next steps (for both!)

- Add other sources
- Analyse Mojito light (before December)
- Pre-merger alerts: signals truncated before merger
- Add realistic noise artefacts: glitches and gaps
- Update the estimates as new data arrive

Two routes to MBHB alerts: fast detection and point estimates that seed the Global Fit, and amortised posteriors

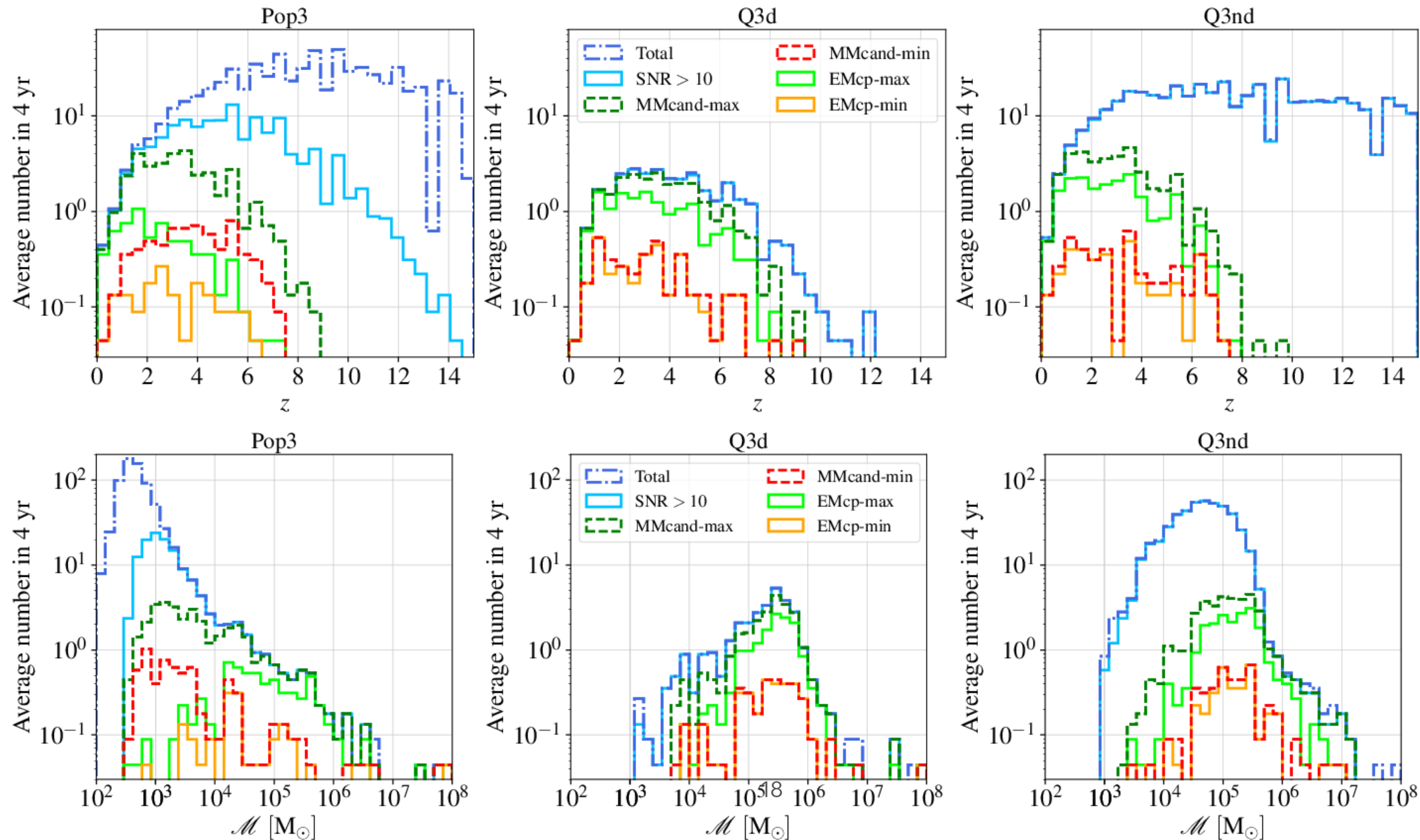
Thank you!

Backup

Population of MBHB we are focusing on

We focus on a sub-population of MBHBs, astrophysically motivated by the possibility of observing an EM signature for such systems

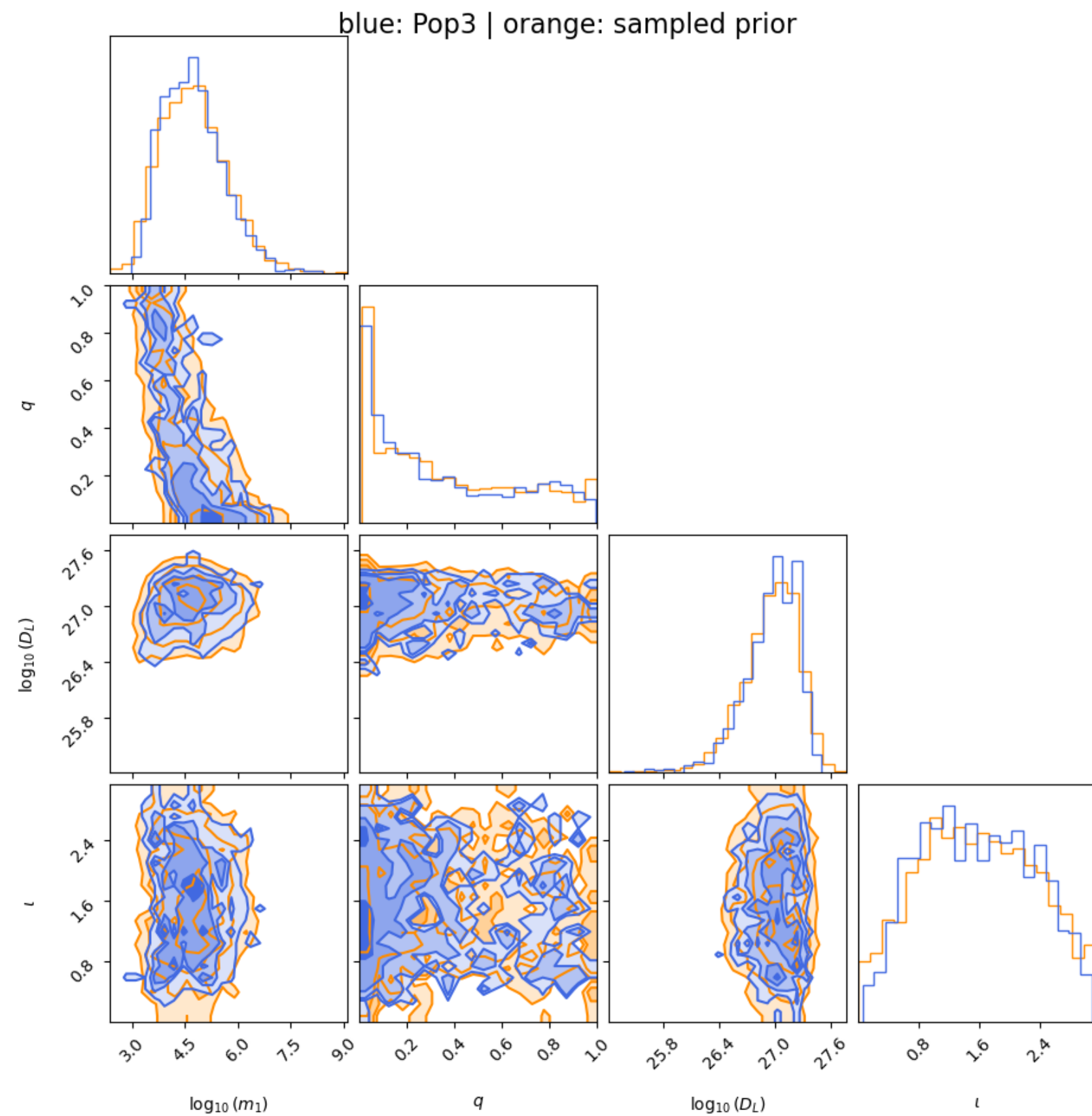
Mangiagli et al +22



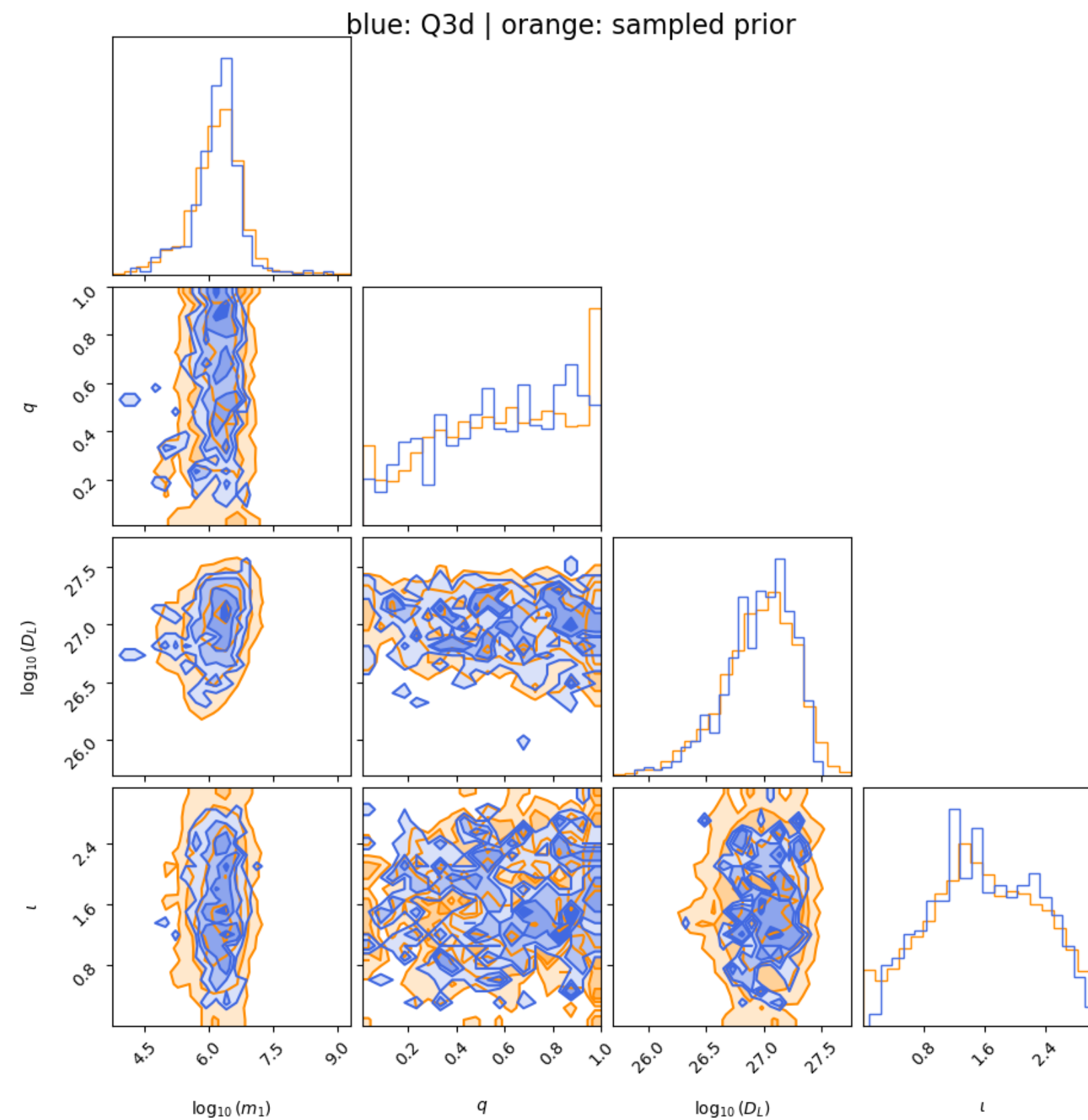
Population of MBHB we are focusing on

three different populations:

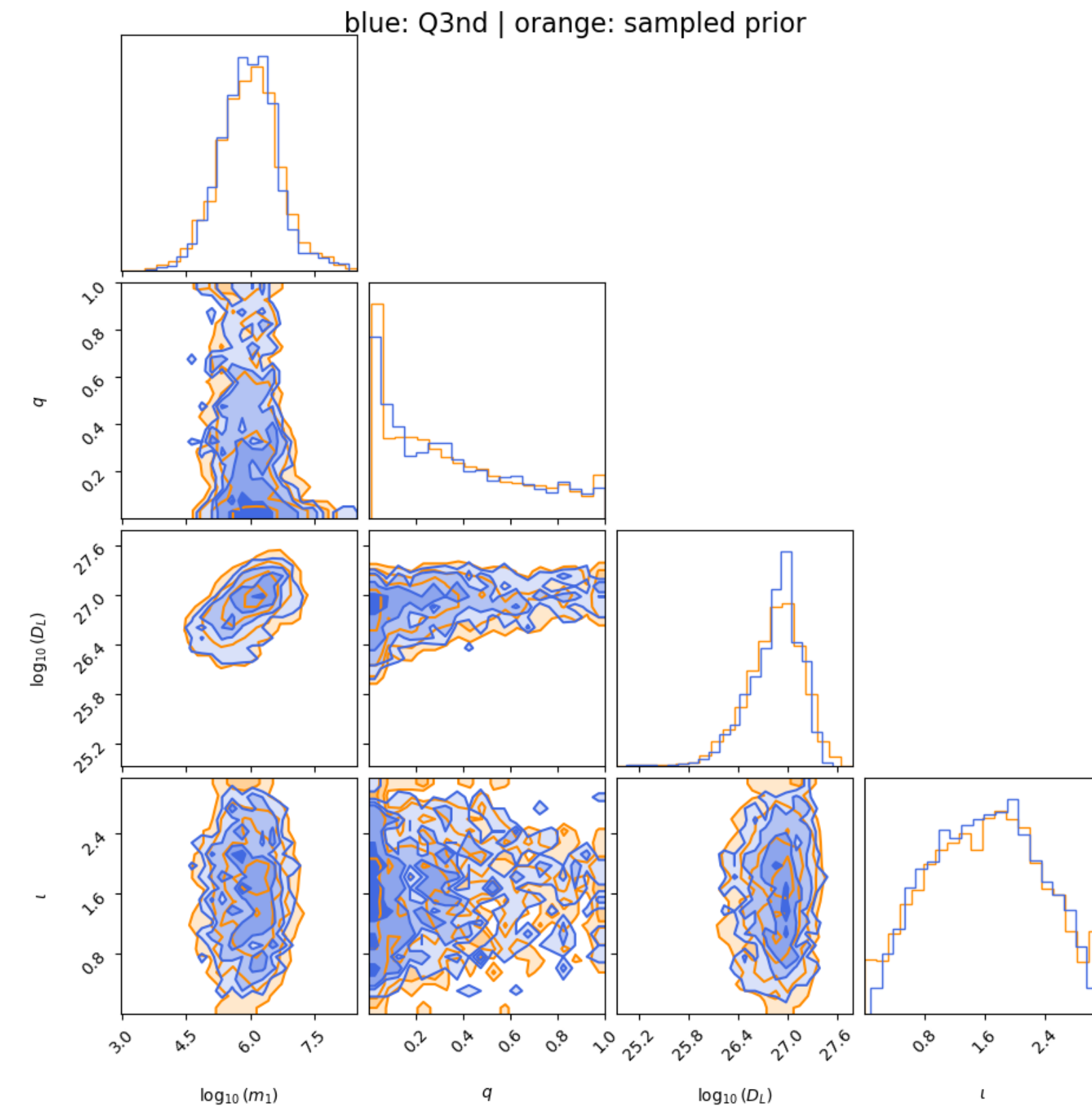
light-seed model



heavy-seed model, delay between the galaxy and BH merger



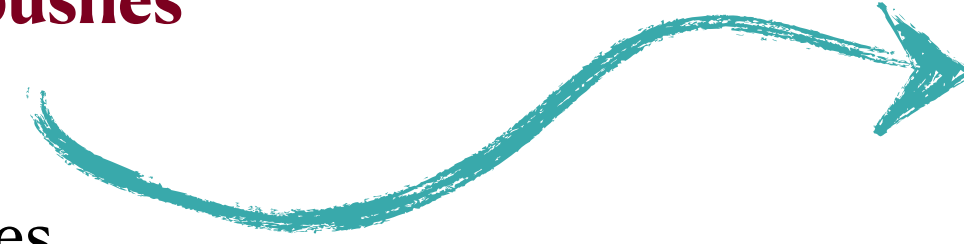
heavy-seed model, no delay



Conditional flow matching in our case

TRAINING: train a U-Net to learn a "velocity" vector field that pushes random noise toward true physical parameters.

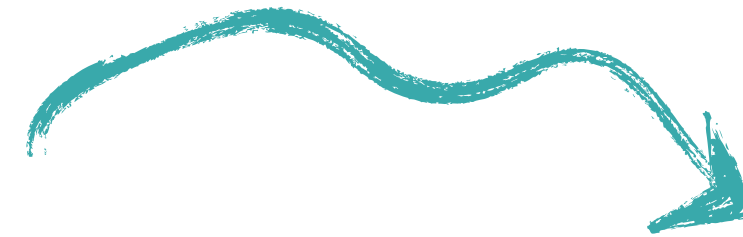
- Instead of chaotic diffusion steps, $\rightarrow$ straight-line trajectories connecting pure noise to the true parameter values.
- Condition on data (time or tf domain) $\rightarrow$ guide the network on exactly where the parameters should flow.
- Pick a random moment in time along the straight path and penalize the network if it doesn't predict the correct straight-line velocity.



1. Dataset: $(\theta_1, x) \sim p_{\text{data}}$ (true parameters and data)
2. Base Noise: $\theta_0 \sim \mathcal{N}(0, I)$ (noise at $t = 0$)
3. Optimal Transport Path: $\theta_t = (1 - t)\theta_0 + t\theta_1$ for $t \in [0, 1]$
4. Target Velocity: $v_t^* = \frac{d\theta_t}{dt} = \theta_1 - \theta_0$
5. Loss Function: $\mathcal{L}(\phi) = \mathbb{E}_{t, \theta_0, \theta_1} \left[\left\| v_\phi(\theta_t, t, x) - (\theta_1 - \theta_0) \right\|^2 \right]$

INFERENCE: Recover the full Bayesian posterior for a completely new, unknown gravitational wave signal.

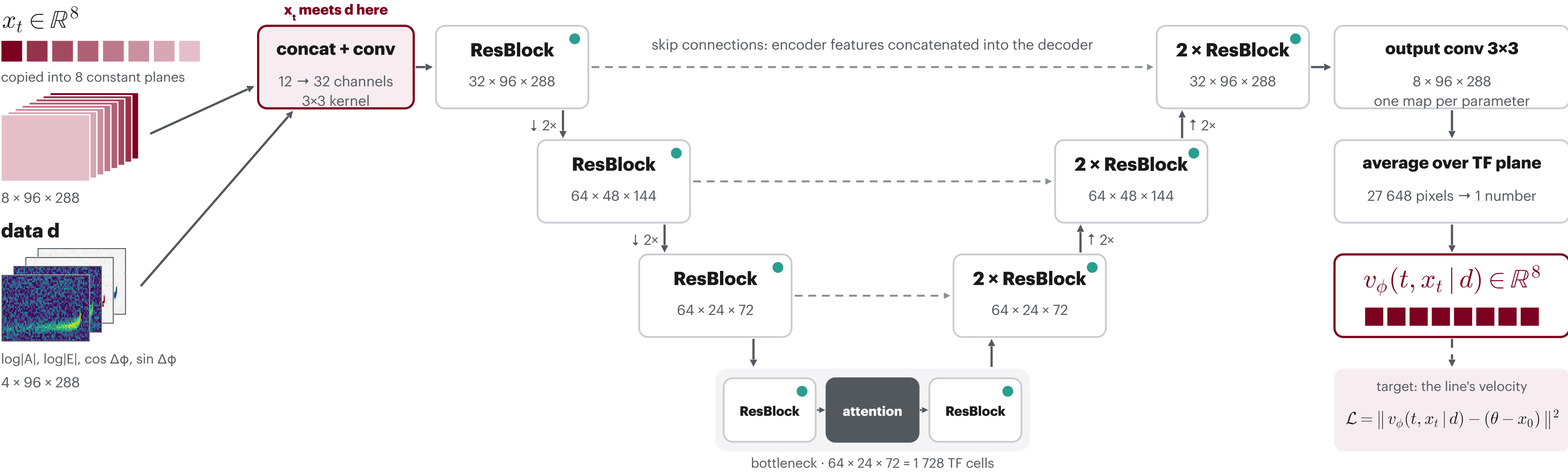
- Generate thousands of random points in the parameter space, representing pure Gaussian noise.
- Use an ODE solver to push these random points along the learned velocity field.
- At every step, the network looks at the observed data and guides the flow. By the end of the integration, the random points cluster into the posterior.



1. Observation: x_{obs} (GW event)
2. Initial State: $\theta_0^{(i)} \sim \mathcal{N}(0, I)$ for $i \in [1, 1000]$
3. ODE Solver: $\theta_1^{(i)} = \theta_0^{(i)} + \int_0^1 v_\phi(\theta_t^{(i)}, t, x_{\text{obs}}) dt$
4. Result: $\theta_1^{(i)} \sim p(\theta | x_{\text{obs}})$ (Bayesian Posterior)

Inside the network: a conditional UNet

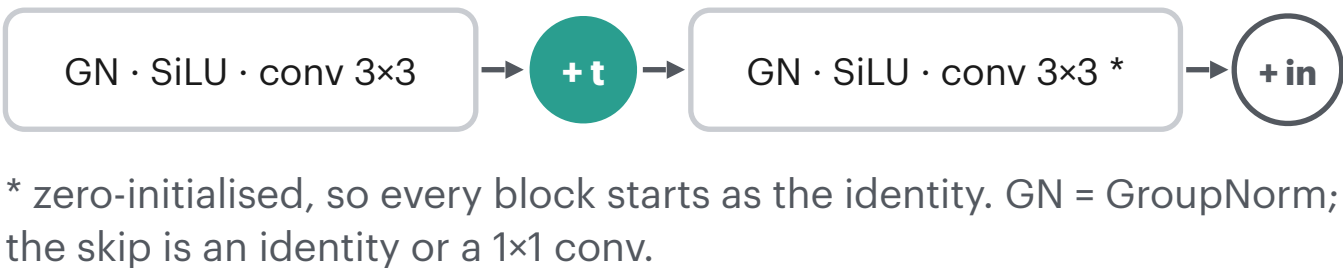
$v_\phi(t, x_t | d)$: the torchcfm UNet, modified to take the flow state as extra image channels · unet_concat_malvi.py, STFT input



Where parameters meet the data

x_t is copied into 8 constant planes and stacked with the 4 data channels. The first 3×3 convolution is the only place they meet; after that, it is one tensor.

Inside every ResBlock

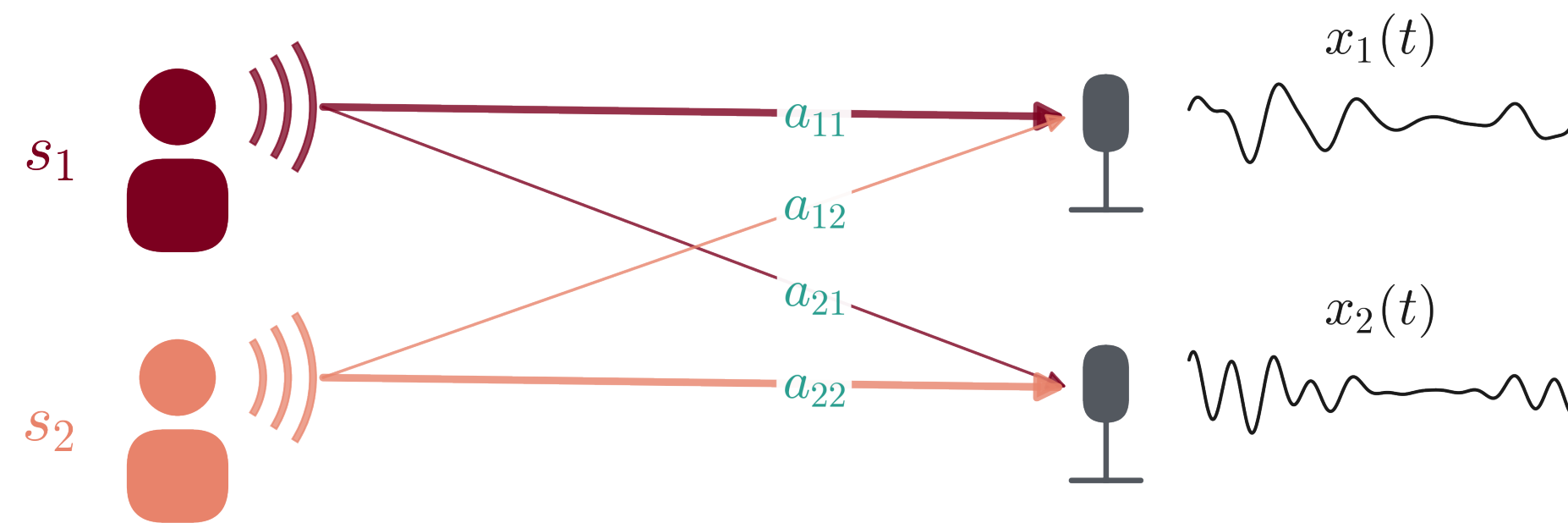


Skips and attention

Skips hand full-resolution time–frequency detail to the decoder. The single attention layer, at the bottleneck, lets every TF cell exchange information with every other.

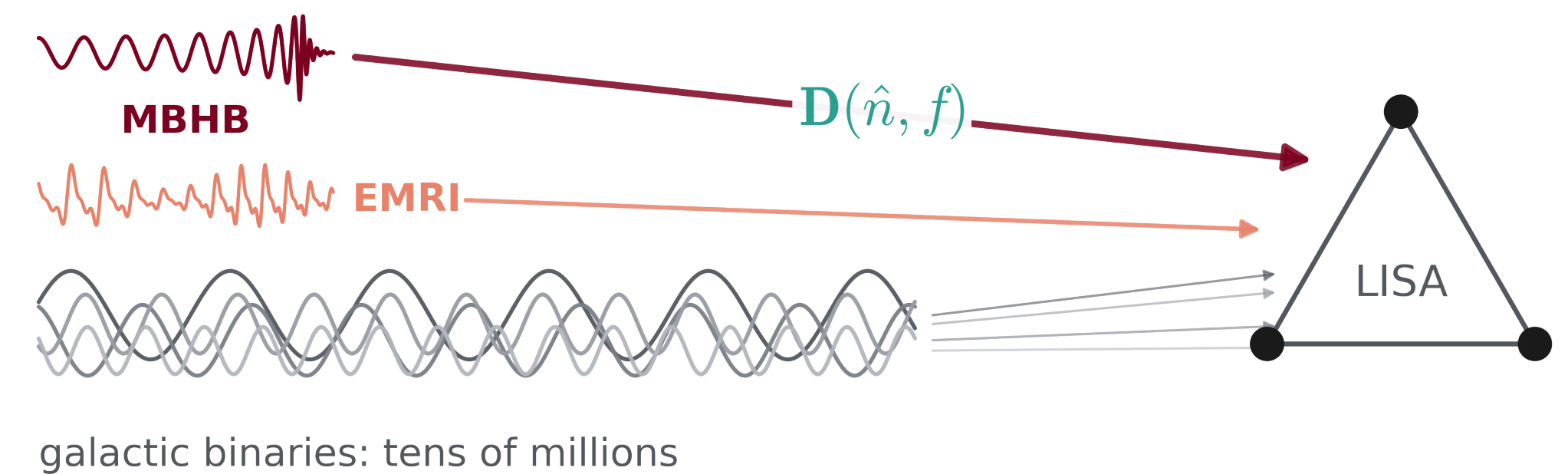
The cocktail party problem

At a party: 2 microphones, 2 voices



$$\underbrace{x_i(t)}_{\text{microphone } i} = \sum_j \underbrace{a_{ij}}_{\substack{\text{how loud voice } j \text{ is} \\ \text{at microphone } i}} \underbrace{s_j(t)}_{\text{voice } j}$$

In LISA: 3 channels, **tens of millions of voices**



$$\underbrace{\tilde{d}_I(f)}_{\text{channel } I \text{ (X, Y, Z)}} = \sum_k^{\substack{\text{every source} \\ \text{in the sky}}} \underbrace{\mathbf{D}_I(\hat{n}_k, f)}_{\substack{\text{detector response:} \\ \text{sky position, frequency}}} : \underbrace{\tilde{\mathbf{h}}_k(f; \xi_k)}_{\substack{\text{source } k, \\ \text{parameters } \xi_k}} + \underbrace{\tilde{n}_I(f)}_{\text{noise}}$$

Far more voices than microphones

3 channels for tens of millions of sources that overlap in time and frequency

The mixing depends on sky position

$\mathbf{D}$ changes with sky position and frequency, and with time as LISA orbits.
Let's localise the source.

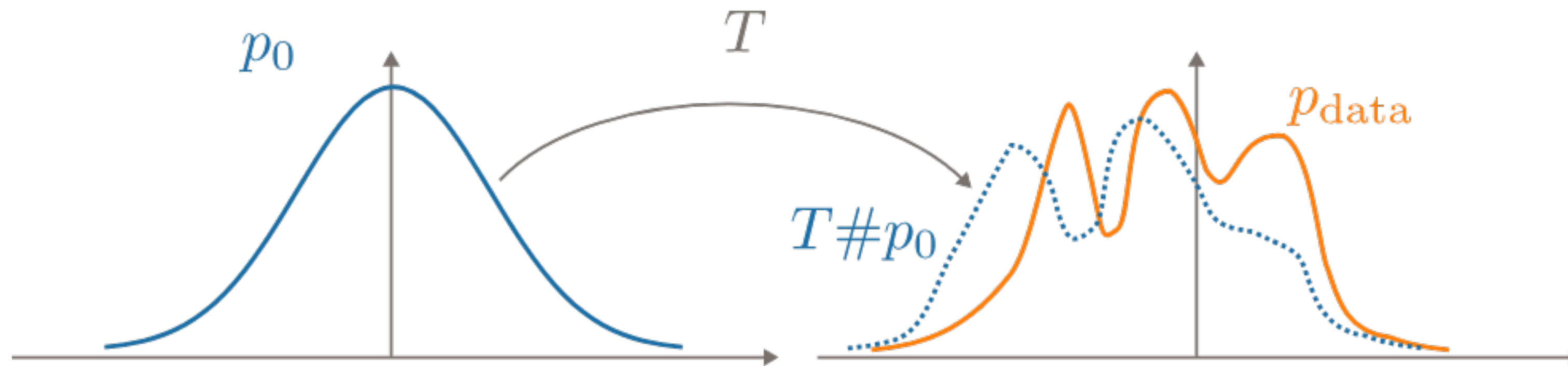
Two ways to listen

Global Fit: analyze every voice at once.
Accurate, but **slow**.

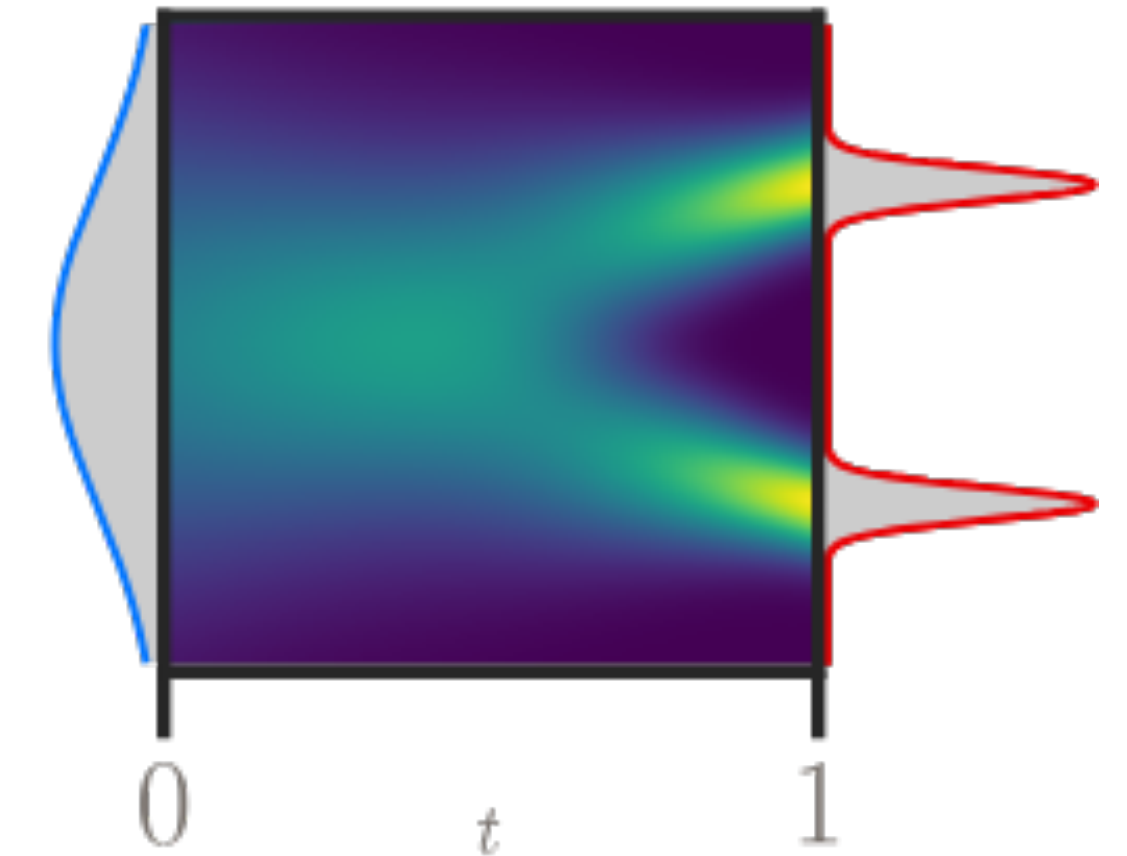
Our networks: estimate MBHB, and treat the rest as background.

Our SBI framework: conditional flow matching

Flow Matching trains a neural network to predict the **velocity vector field** describing a the path from an easy distribution into a complex target distribution (parameters' posteriors)



<https://dl.heeere.com/conditional-flow-matching/blog/conditional-flow-matching/>



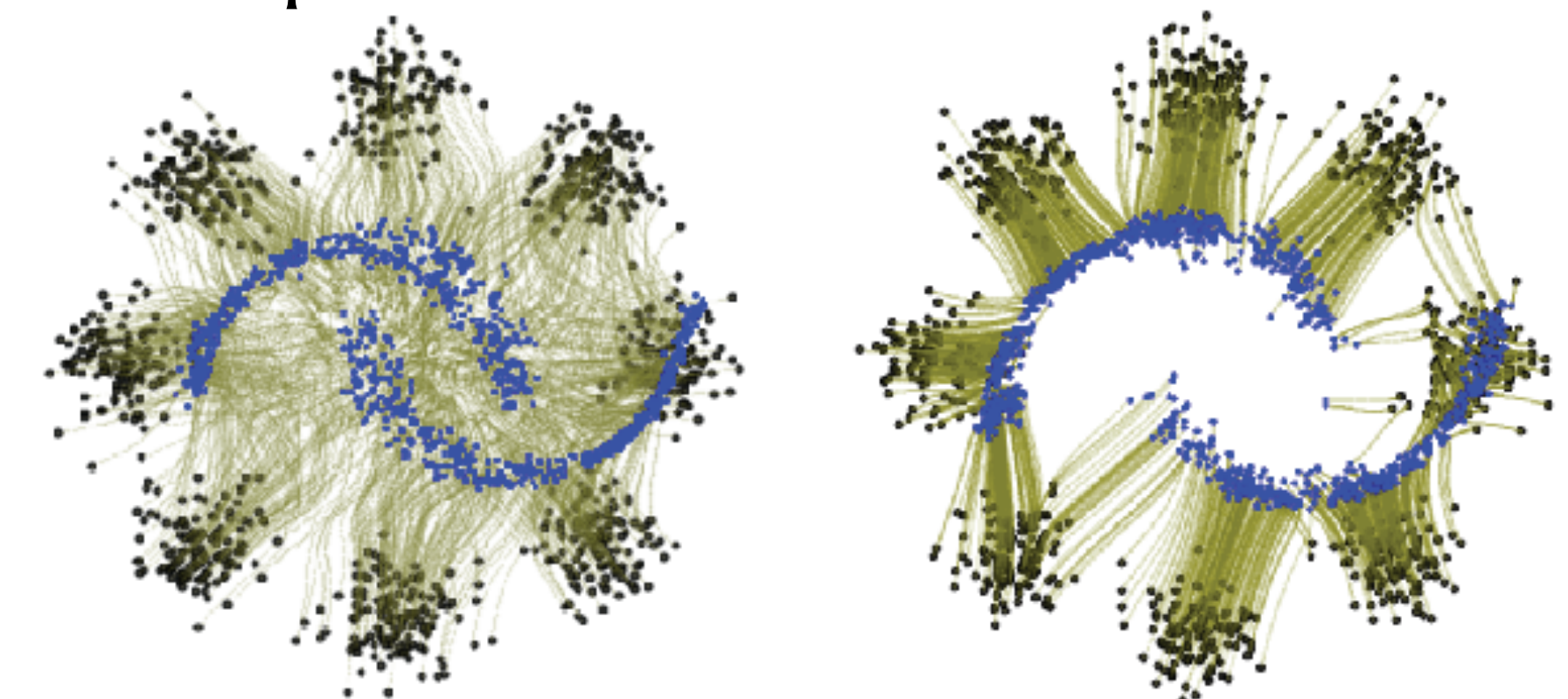
Optimal transport:

forces the paths to be straight lines with constant speed.

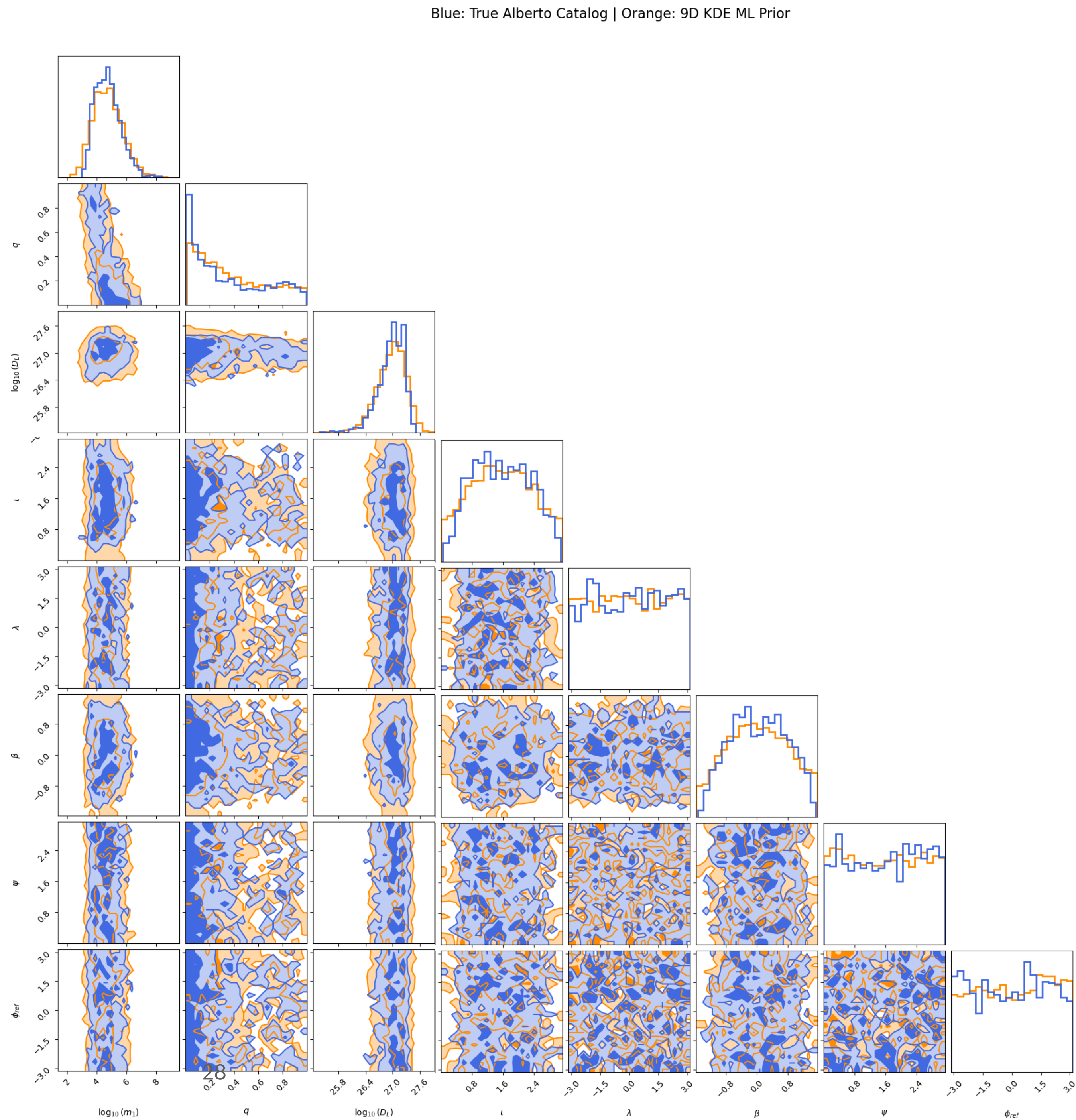
$$\mathcal{L}(\theta) = \mathbb{E}_{t, x_0, x_1} \left[\left\| v_{\theta}(t, x_t | \mathbf{y}) - u_t \right\|^2 \right]$$

Loss: Mean Squared Error regression

prediction target velocity



Population of MBHB we are focusing on



Parameter estimation

Time series

Benchmarking against **standard Bayesian inference methods** (MCMC or Nested Sampling) is **necessary** to establish the posteriors that we expect.

Very preliminary results!

