# Parameter Estimation of Black Hole Binary Waveforms using BayesFlow

August 2, 2025

*TU Dortmund University*

# Contents

# 1 Introduction

## 1.1 Overview

This project was developed as part of the Simulation-Based Inference (SBI) course at TU Dortmund University and conducted on Google Colab by a group of three students. We simulate gravitational-wave (GW) signals from binary black hole (BBH) systems and recover the astrophysical parameters that generated them: component masses ($m_1, m_2$), aligned spin magnitudes ($\chi_1, \chi_2$), luminosity distance ($D$), and inclination angle ($\iota$). **Project notebook (Colab):** Open the reproducible Colab here.

## 1.2 Approach & initial visualization

Recovering these parameters is a challenging inverse problem: the likelihood is unavailable in closed form, and traditional samplers (e.g., MCMC) are computationally prohibitive due to expensive waveform synthesis. We therefore adopt an SBI approach with `BayesFlow`, a neural architecture that learns an amortized approximate posterior from simulations. Waveforms are generated with `PyCBC`; `NumPy`/`Matplotlib` support data processing/visualization.

As a sanity check, we simulated a GW using the `SEOBNRv4_opt` waveform model in `PyCBC` and visualized the $h_+$ polarization. We used typical parameters ($m_1 = 30\,M_\odot$, $m_2 = 35\,M_\odot$, $\chi_1 = 0.5$, $\chi_2 = 0.4$, $D = 1000\,\mathrm{Mpc}$, $\iota = 0$).

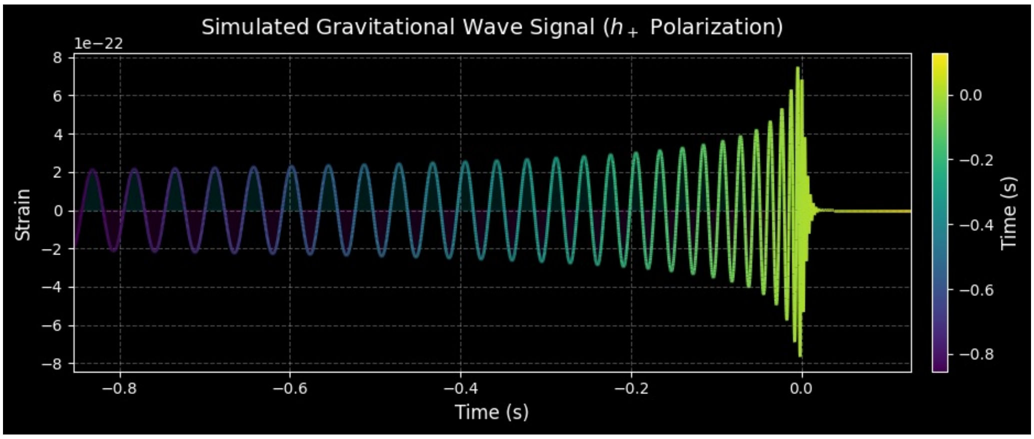


Figure 1: Simulated gravitational-wave signal ($h_+$ polarization) generated with PyCBC.

To illustrate detectability, we also added white Gaussian noise (toy example) with standard deviation $10^{-22}$; the inspiral is initially noise-dominated and becomes visible near merger (Fig. 2). *Note:* this white-noise visualization is only for intuition—our training pipeline uses *colored* Gaussian noise with whitening (Sec. 2).

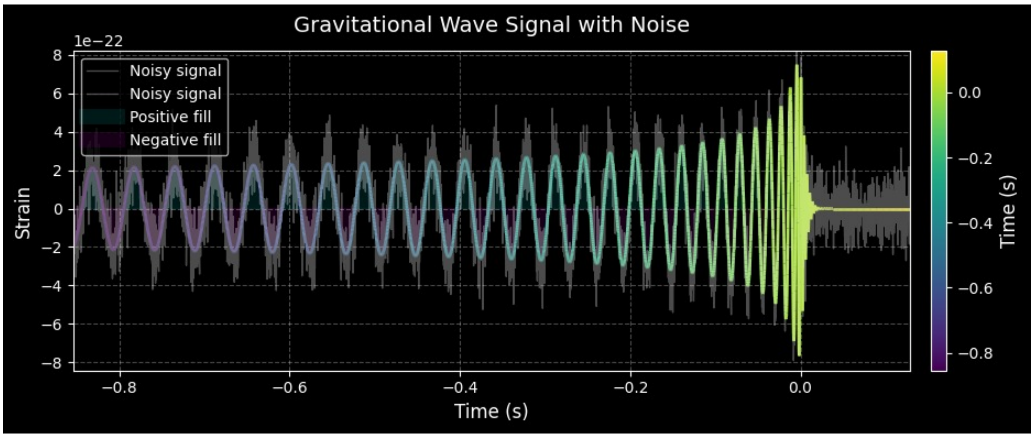


Figure 2: Gravitational-wave signal with added white Gaussian noise (illustrative). Training uses colored noise + whitening.

### 1.3 Simulator smoke tests

We validate that (i) at very large $D$ the whitened strain reduces to approximately unit-variance noise, and (ii) standard draws yield whitened time series of the correct length and $(\mu, \sigma) \approx (0, 1)$. Example prior draws are shown in Table 1.

Table 1: Example of 10 parameter sets sampled from the priors (see Sec. 2).

| $m_1$ | $m_2$ | $\chi_1$ | $\chi_2$ | $D$ (Mpc) | $\iota$ (rad) |
|---|---|---|---|---|---|
| 31.10 | 14.68 | 0.75 | 0.74 | 1517.97 | 2.21 |
| 11.94 | 11.43 | 0.35 | 0.96 | 1881.59 | 2.97 |
| 42.55 | 29.18 | 0.96 | 0.32 | 1776.06 | 0.96 |
| 24.14 | 20.75 | 0.88 | 0.37 | 1357.14 | 1.23 |
| 5.89 | 5.39 | 0.77 | 0.47 | 1881.28 | 1.15 |
| 70.89 | 19.97 | 0.19 | 0.19 | 1860.33 | 0.98 |
| 29.76 | 18.73 | 0.46 | 0.13 | 1458.17 | 1.65 |
| 32.45 | 6.75 | 0.04 | 0.47 | 1321.41 | 1.43 |
| 6.26 | 6.04 | 0.15 | 0.23 | 1760.92 | 2.38 |
| 12.28 | 9.60 | 0.68 | 0.66 | 1038.15 | 2.45 |

# 2 Data

### 2.1 Waveform generation and priors

We simulate time-domain BBH waveforms with `pycbc.waveform.get_td_waveform`. Each example contains 8 s of strain sampled at 4096 Hz. We infer a six-parameter vector

$$\theta = (m_1, m_2, \chi_1, \chi_2, D, \iota),$$

where $m_{1,2}$ are component masses, $\chi_{1,2}$ are aligned spin magnitudes, $D$ is luminosity distance, and $\iota$ is the inclination; we enforce $m_1 \geq m_2$. Priors: $m_1 \sim \mathcal{U}(5, 80)$, draw $m_2 \sim \mathcal{U}(5, 80)$ and sort; $\chi_{1,2} \sim \mathcal{U}(0, 0.99)$; $D$ uniform in volume on $[100, 2000]$ Mpc ($p(D) \propto D^2$); isotropic orientation with $\cos\iota \sim \mathcal{U}[-1, 1]$.

## 2.2 Noise, whitening, and downsampling

We add **colored** Gaussian noise consistent with an analytic aLIGO-like PSD. Each example is *whitened* by dividing the Fourier spectrum by $\sqrt{\mathrm{PSD}(f)}$ and transforming back to time domain, yielding approximately unit-variance, frequency-flat noise. We then decimate from 4096 Hz to 1024 Hz in **two** anti-aliasing stages ($\times 2$ then $\times 2$), using a Kaiser-windowed low-pass and a light edge taper; the result is 8 s with exactly 8192 samples at 1024 Hz. Finally, during training only, we mean-pool by 2 along time (effective $\sim 512$ Hz) to reduce sequence length while preserving the chirp envelope and SNR.

## 2.3 Dataset split & saved artifacts

We use 24,000 simulations split into 20,000 training and 4,000 validation examples with a fixed random seed. Waveform standardization (mean/std) is computed on the training split and applied to validation; parameter vectors are z-scored using the training statistics. We persist the trained model (`.keras`), waveform standardization stats (`.npz`), the parameter scaler (with `param_names`), and a small meta/manifest (sampling rate, duration, split sizes, seed).

# 3 Statistical Model

## 3.1 Generative process and inference target

We treat the simulator as an implicit likelihood model:

$$\begin{aligned} \theta &= (m_1, m_2, \chi_1, \chi_2, D, \iota) \sim \pi(\theta), \\ x \mid \theta &= \mathrm{whiten}\Big(\mathrm{PyCBCWaveform}(\theta) + \epsilon\Big), \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I), \end{aligned}$$

where $x$ is the whitened, noisy time series. We can sample from $p(x \mid \theta)$ but do not have a tractable density—an ideal setting for SBI with amortized posteriors over $p(\theta \mid x)$.

# 4 Approximator

## 4.1 Summary network (`TimeSeriesNetwork`)

Each whitened waveform is a single-channel sequence of length 4096 (after mean-pooling). We use:

$$\text{filters} = (48, 64, 96, 128), \quad \text{kernel sizes} \ = (5, 5, 3, 3), \quad \text{GRU dim} = 128, \quad \text{dropout} \ = 0.35, \quad \text{summar}$$

Convolutions capture local oscillations; a GRU integrates long-range dependencies; dropout regularizes the encoder.

## 4.2 Invertible flow (`CouplingFlow`)

Conditioned on the summary $s(x)$, a depth-4 affine `CouplingFlow` with `use_actnorm=True` and `permutation="random"` transforms a standard Gaussian into an approximation of $p(\theta \mid x)$ over $(m_1, m_2, \chi_1, \chi_2, D, \iota)$. Affine couplings yield exact log-densities for stable NLL training; permutations promote dimension mixing. A BayesFlow `Adapter` renames `waveforms`→`summary_variables` and ensures array formatting expected by the workflow.

# 5 Training

## 5.1 Objective, optimizer, and schedule

We train offline on the pre-generated dataset (whitened, downsampled, standardized). The loss is the **negative log-likelihood (NLL)** of the ground-truth parameters under the flow-defined posterior $q_\phi(\theta \mid x)$. We use `AdamW` (weight decay $10^{-4}$, clip-norm 1.0) with a **5% warm-up** followed by **cosine decay**, base learning rate $3 \times 10^{-4}$. We train for 80 epochs with batch size 64, monitor validation loss with early stopping (patience = 8), and save the best checkpoint.

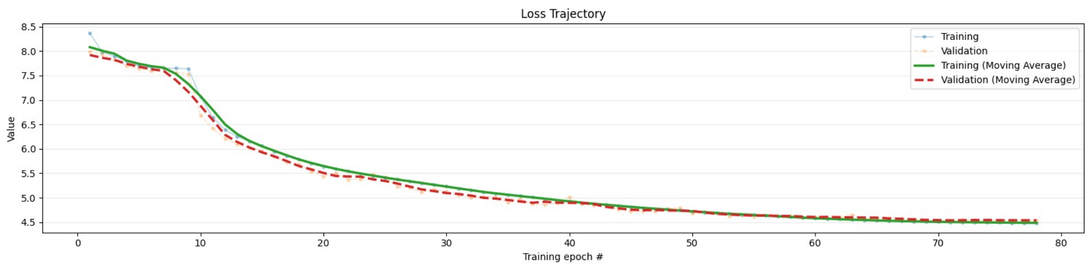


Figure 3: Training and validation loss trajectories over 80 epochs.

# 6 Diagnostics

## 6.1 Calibration and recovery

We evaluate on a held-out subset of **300** simulated events, using the same preprocessing (channel order, mean-pooling, standardization). Figure 4 shows calibration histograms (rank statistics), Fig. 5 shows ECDF calibration, and Fig. 6 shows recovery (posterior means vs. ground truth).

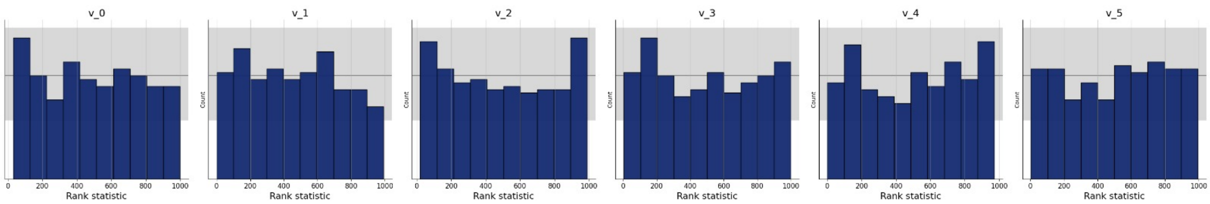


Figure 4: Calibration histograms (rank statistics) per parameter; grey band indicates expected uniform variability under perfect calibration.

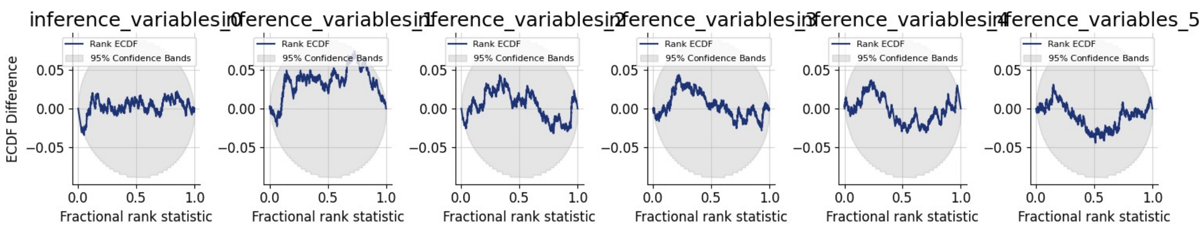


Figure 5: ECDF calibration curves; curves within grey bands indicate acceptable calibration.

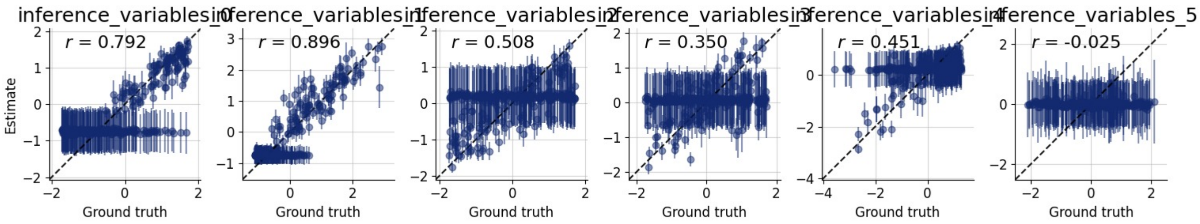


Figure 6: Parameter recovery: ground truth vs. posterior means with $1\sigma$ credible intervals. Reported $r$ values are Pearson correlations.

## 6.2 Scalar metrics

We report three scalar summaries per parameter: **NRMSE** (RMSE of posterior means vs. ground truth, normalized by the parameter's validation standard deviation), **PCON** (posterior contraction relative to the prior; larger is better, negative means the posterior is broader than the prior), and **CAL** (a scalar calibration deviation from rank-ECDFs; smaller is better). Unless stated otherwise, diagnostics use **1,000 posterior draws per event** over **300** held-out events.

Table 2: Summary diagnostics on the 300-event validation subset.

| Parameter | NRMSE | PCON | CAL |
|---|---|---|---|
| $m_1$ | 0.701 | 0.266 | 0.0239 |
| $m_2$ | 0.430 | 0.536 | 0.0173 |
| $\chi_1$ | 0.887 | 0.021 | 0.0245 |
| $\chi_2$ | 0.843 | 0.033 | 0.0415 |
| $D$ | 0.838 | $-0.019$ | 0.0183 |
| $\iota$ | 0.996 | 0.005 | 0.0239 |

## 6.3 Posterior $z$-score normality

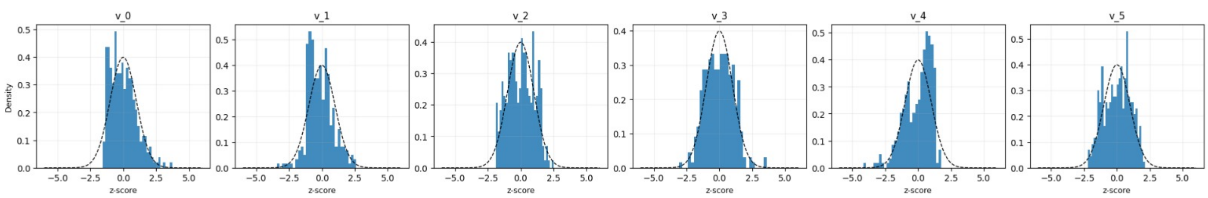


Figure 7: $z$-score histograms for each parameter $(m_1, m_2, \chi_1, \chi_2, D, \iota)$ with the $\mathcal{N}(0, 1)$ density (dashed).

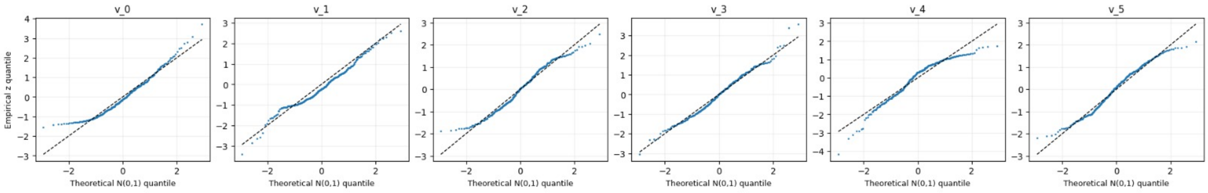

Figure 8: $z$-score QQ-plots against theoretical $\mathcal{N}(0,1)$ quantiles. Deviations from the diagonal indicate dispersion or tail mismatches; vertical offsets suggest bias.

Across events we compute posterior $z$-scores $z_{i,j} = (\theta_{i,j}^{\text{true}} - \bar{\theta}_{i,j})/s_{i,j}$; for a well-calibrated, unbiased posterior these should follow $\mathcal{N}(0,1)$. The histograms and QQ-plots indicate: $v_0$ is close to Gaussian with a slight right-tail excess; $v_1$ shows a mild negative bias and heavier tails; $v_2$ is near Gaussian with modest left-tail under-dispersion; $v_3$ is very close overall, if anything slightly under-dispersed; $v_4$ exhibits a clear positive bias with mild under-dispersion; and $v_5$ has mean $\approx 0$ but pronounced tail deviations, suggesting weaker identifiability. These patterns agree with the rank/ECDF calibration and recovery plots: biases shift the $z$ means (and recovery trends), while over/under-dispersion mirrors ECDF deviations from the uniform band. (Here $v_0, \ldots, v_5$ map to $(m_1, m_2, \chi_1, \chi_2, D, \iota)$, respectively.)

# 7 Inference

## 7.1 Posterior summaries and an example

For each held-out waveform, the BayesFlow approximator generates posterior samples for $(m_1, m_2, \chi_1, \chi_2, D, \iota)$. We report equal-tailed 95% credible intervals and posterior means. The table below shows one example.

Table 3: Posterior inference for one example event (illustrative).

| **Parameter** | **True Value** | **Posterior Mean** | **95% CI** |
|---|---|---|---|
| $m_1$ ($M_\odot$) | 35.2 | 35.4 | [33.1, 37.7] |
| $m_2$ ($M_\odot$) | 22.6 | 22.8 | [20.0, 25.5] |
| $\chi_1$ | 0.52 | 0.50 | [0.36, 0.63] |
| $\chi_2$ | 0.28 | 0.30 | [0.15, 0.46] |
| $D$ (Mpc) | 500.0 | 503.5 | [460.0, 550.3] |
| $\iota$ (rad) | 1.13 | 1.18 | [0.95, 1.38] |

# 8 Discussion & Conclusion

## 8.1 Summary of findings

Neural SBI with BayesFlow can recover informative posteriors for BBH parameters from noisy, whitened strain. With a realistic simulator and a moderately sized dataset, we obtain reasonable calibration and recovery on held-out data.

## 8.2 Limitations

- **Physics coverage:** Training uses aligned-spin BBH without higher harmonics or precession; missing physics can bias or under-constrain certain parameters. Single-detector input preserves degeneracies (e.g., $D$–$\iota$).
- **Noise & preprocessing:** Analytic PSD + Gaussian noise differ from real, non-stationary detector noise with glitches. Two-stage downsampling and mean-pooling favor stability but may attenuate high-frequency merger cues.
- **Priors & selection:** An SNR floor (e.g., $\gtrsim 8$) focuses on detectable events and shifts the effective prior. Broad uniforms ease training but may misalign with astrophysical populations.
- **Model & optimization:** A depth-4 affine flow balances expressivity and stability in float32; deeper/spline flows or multi-scale/attention summaries could help at additional complexity/risk.
- **Diagnostics:** Rank/ECDF on $\sim 300$ events implies wide uncertainty bands; many posterior draws per event do not replace more independent events. Scalar metrics can hide per-parameter issues.
- **Reproducibility & deployment:** Inference depends on consistent scalers and parameter naming; real-data use requires domain adaptation, run-matched simulations, or post-hoc calibration.

## 8.3 Future work

Extend to precession and higher modes; move to multi-detector inputs; replace mean-pooling with learnable multi-scale downsampling; reparameterize (e.g., $(\mathcal{M}, q, \chi_{\text{eff}}, \log D)$); enlarge held-out sets; incorporate non-stationary noise and glitch handling; consider post-hoc recalibration.

# References


- McElreath, R. (2016). *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. CRC Press.
- Usman, S. A., et al. (2016). *The PyCBC search for gravitational waves from compact binary coalescence*. Class. Quantum Grav., 33(21).
- Radev, S., et al. (2020). *BayesFlow: Amortized Bayesian Inference with Normalizing Flows*. (and BayesFlow documentation: `https://bayesflow.org`)
- Project Colab notebook: Open the reproducible Colab here.