Title: Statistical Gravitational Waveform Models: What to Simulate Next?

Date: Jun 08, 2017 01:00 PM

URL: http://pirsa.org/17060016

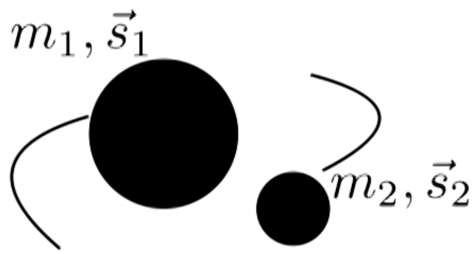
Abstract: <p>Models of gravitational waveforms play a critical role in detecting and characterizing the gravitational waves (GWs) from compact binary coalescences. Waveforms from numerical relativity (NR), while highly accurate, are too computationally expensive to produce to be directly used in parameter estimation. We propose a Gaussian process regression (GPR) method to generate accurate reduced-order-model waveforms based only on existing accurate (e.g. NR) simulations. Using a training set of simulated waveforms, our GPR approach produces interpolated waveforms along with uncertainties across the parameter space. Beyond interpolation of waveforms, we also present the ``Minimization of the Maximum Estimated Error Placement" (MMEEP) method, which utilizes the errors provided by our GPR model to optimize the placement of future simulations. </p>

Statistical Gravitational Waveform Models: What to Simulate Next?

Zoheyr Doctor, Ben Farr, Daniel Holz, Michael Pürrer

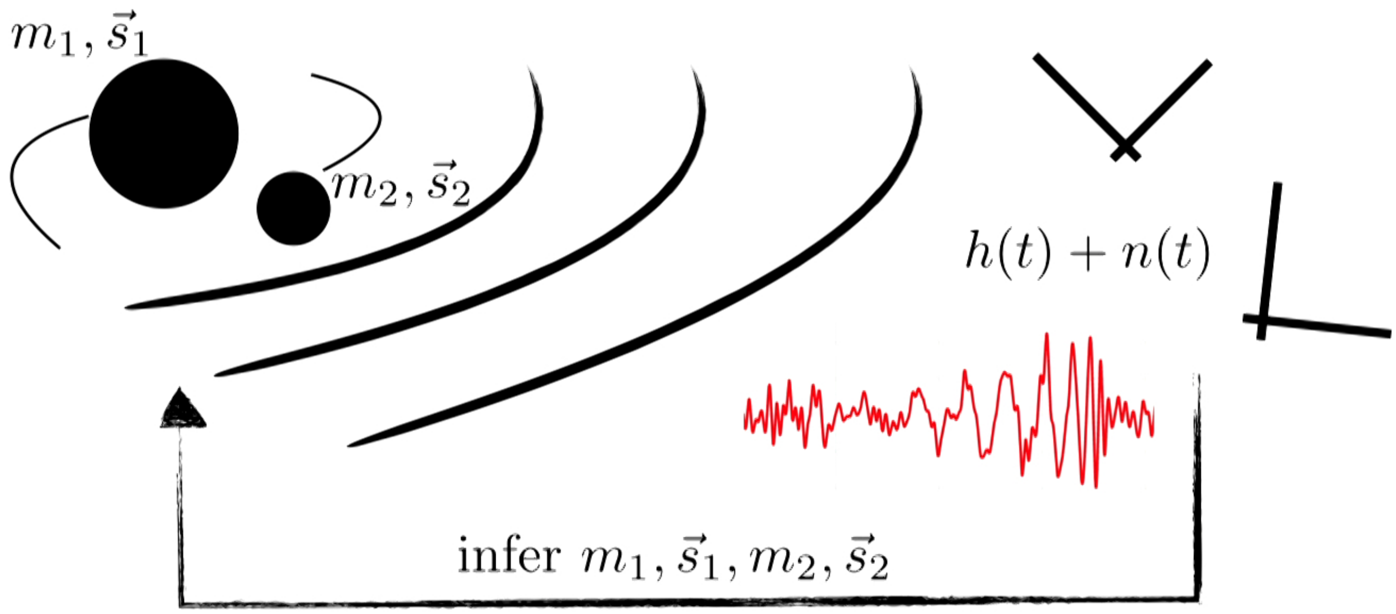
8 June 2017

(paper on arXiv soon!)

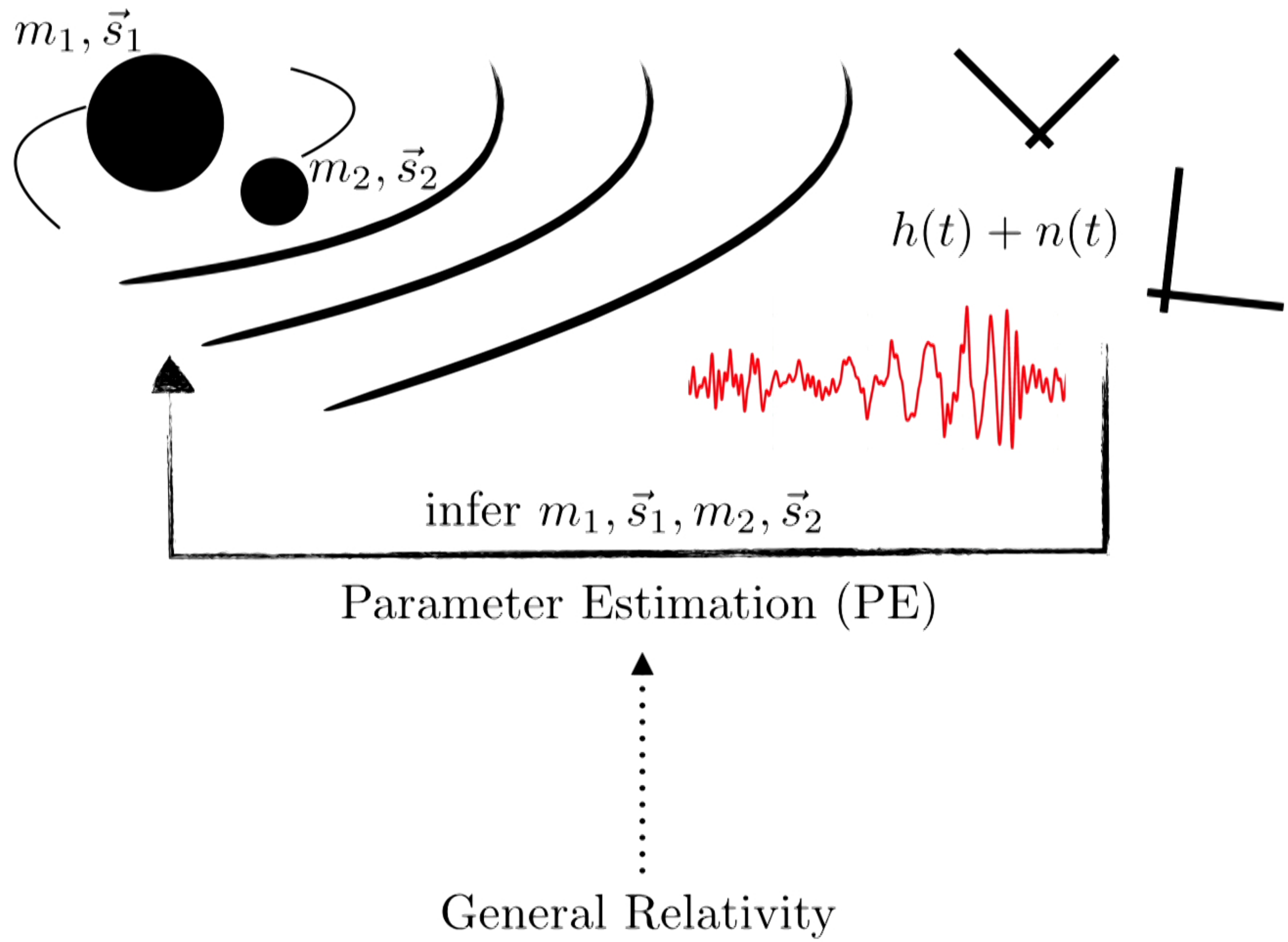


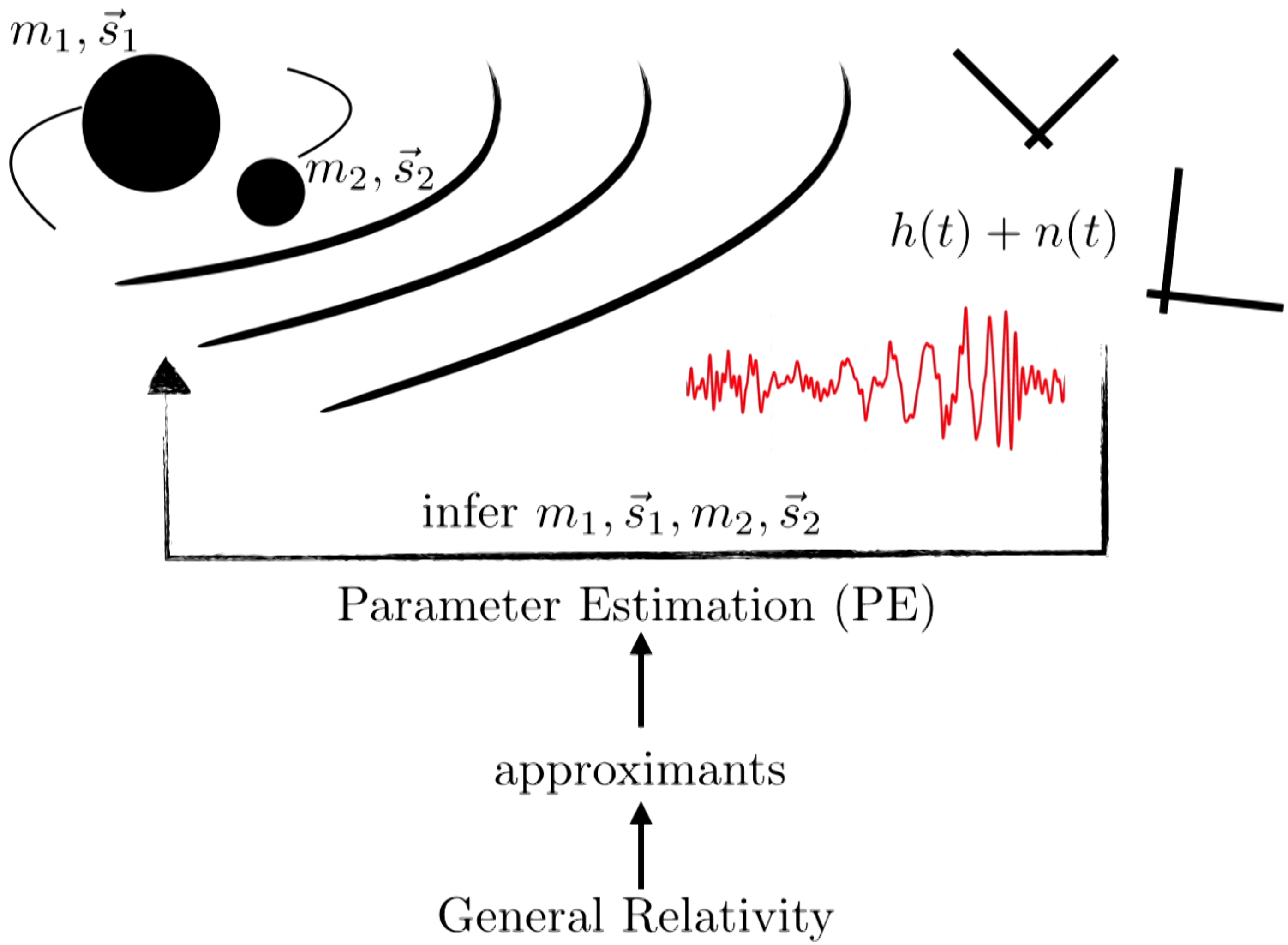
2

$m_1, \vec{s}_1$
$m_2, \vec{s}_2$
$h(t) + n(t)$



Parameter Estimation (PE)


$m_1, \vec{s}_1$
$m_2, \vec{s}_2$
$h(t) + n(t)$
infer $m_1, \vec{s}_1, m_2, \vec{s}_2$
Parameter Estimation (PE)
General Relativity


$m_1, \vec{s}_1$
$m_2, \vec{s}_2$
$h(t) + n(t)$
infer $m_1, \vec{s}_1, m_2, \vec{s}_2$
Parameter Estimation (PE)
approximants
General Relativity

2

Basics of PE

$$p(\vec{\lambda}|d) = \frac{p(d|\vec{\lambda})p(\vec{\lambda})}{p(d)}$$

Basics of PE

$$p(\vec{\lambda}|d) = \frac{p(d|\vec{\lambda})p(\vec{\lambda})}{p(d)}$$

probability of source parameters given data

probability of data given source parameters

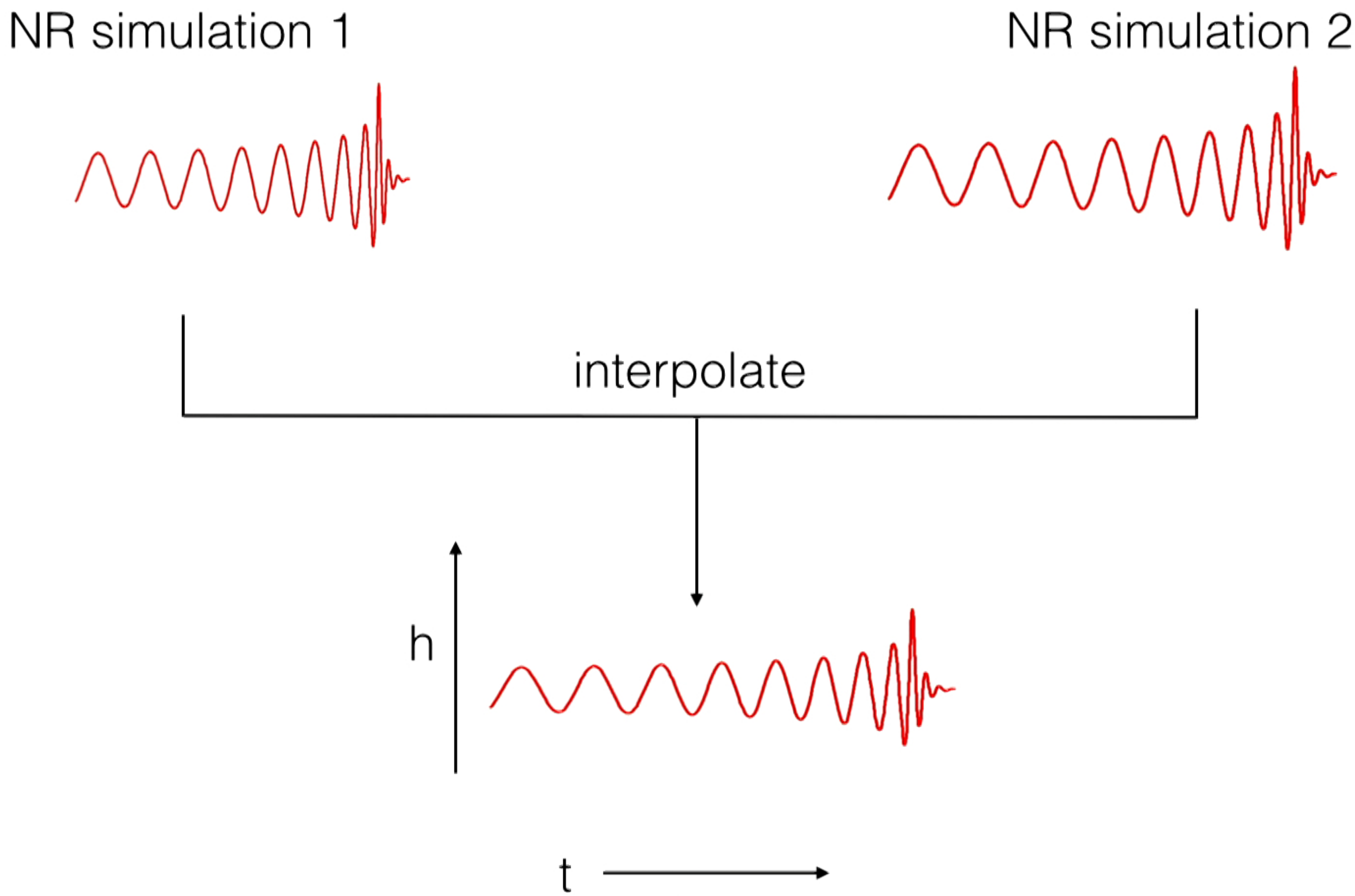
To evaluate the likelihood $p(d|\vec{\lambda})$ we need to predict what a GW looks like for each set of parameters

Challenges in PE

- Different approximants produce different waveforms and are applicable in different domains.
- Systematic errors in approximants can bias PE results.
- Approximants may be slow to evaluate.

How to Incorporate NR?

- Numerical relativity (NR) simulations each take weeks to months to run.
- One idea: Interpolate between NR simulations (a.k.a. *NR surrogates*)
- What is the error level of the surrogate?
- Generate new NR simulations at which parameter values?


NR simulation 1
NR simulation 2
interpolate
h
t

Proposal

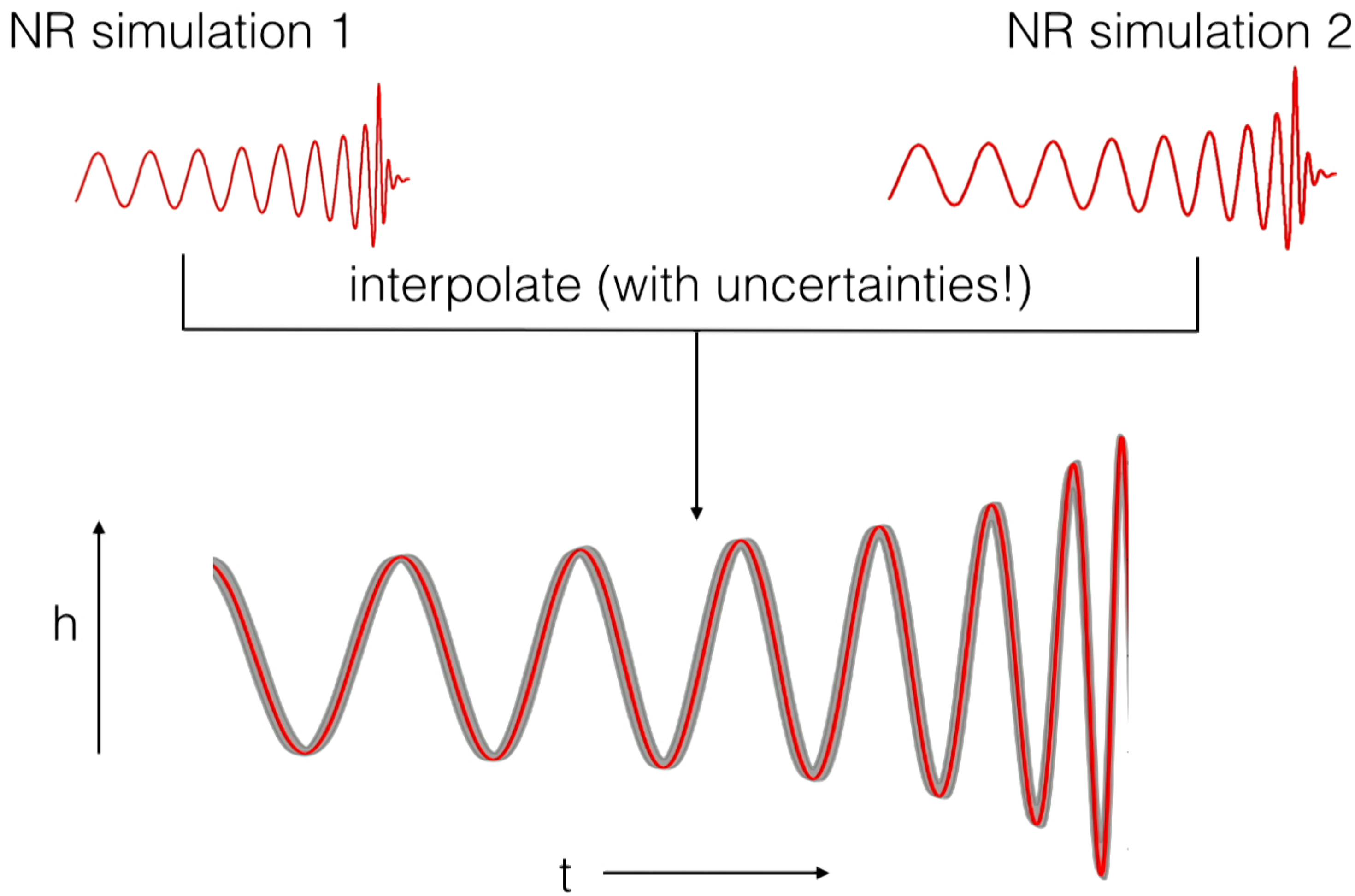
- Use Gaussian process regression (GPR) to *emulate* waveforms based on a training set of accurate (e.g. NR) waveforms.

- Why?

  1. waveforms come with uncertainties

  2. waveforms + uncertainties can be quickly evaluated

  3. errors on waveforms are minimized through strategic choices of accurate (e.g. NR) waveform parameters


NR simulation 1
NR simulation 2
interpolate (with uncertainties!)
h
t

8

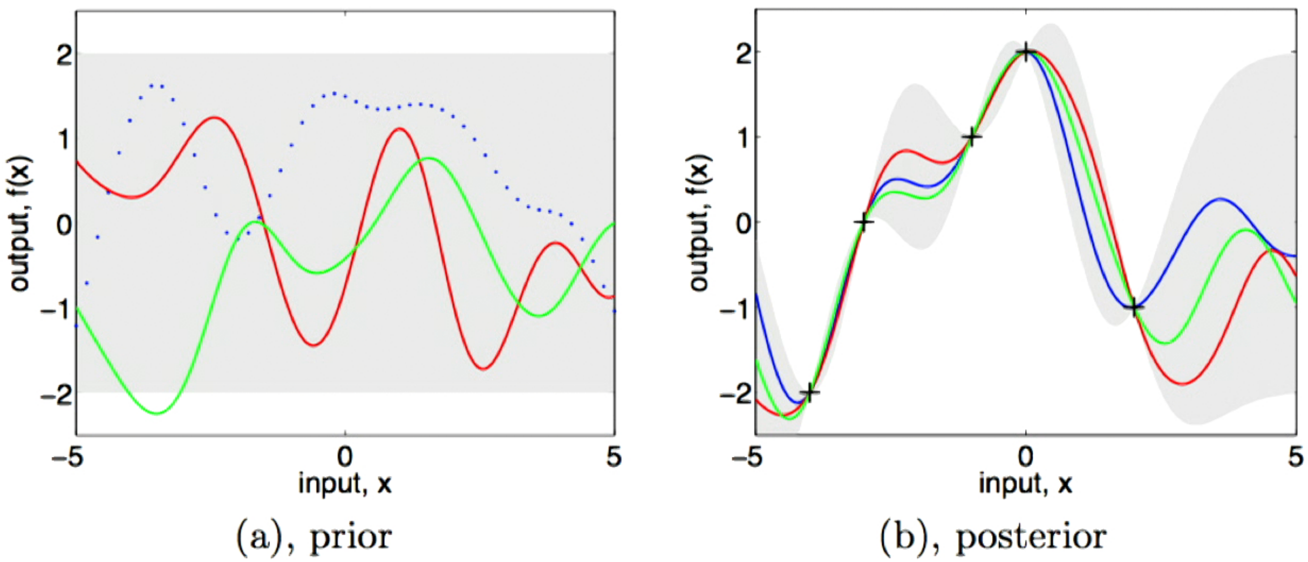
Gaussian Processes

- Model waveforms as *distributions* over possible functions.
- A Gaussian process assumes any subset of variables follows a multivariate Gaussian joint distribution.

$$f(\vec{x}) \sim \mathcal{GP}(\vec{\mu}, \mathbf{k}(\vec{x}, \vec{x}'))$$

Mean f(x)

Covariance between f(x) and f(x')



Rasmussen and Williams (2006)

Gaussian Process Regression

- Test values $f(X_*)$ at points X_* are related to known values f(X) at "training" points X by a normal dist.:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

- The elements of the covariance sub-matrices K are calculated using the covariance function.

- Find probability of $f(X_*)$ conditioned on f(X):

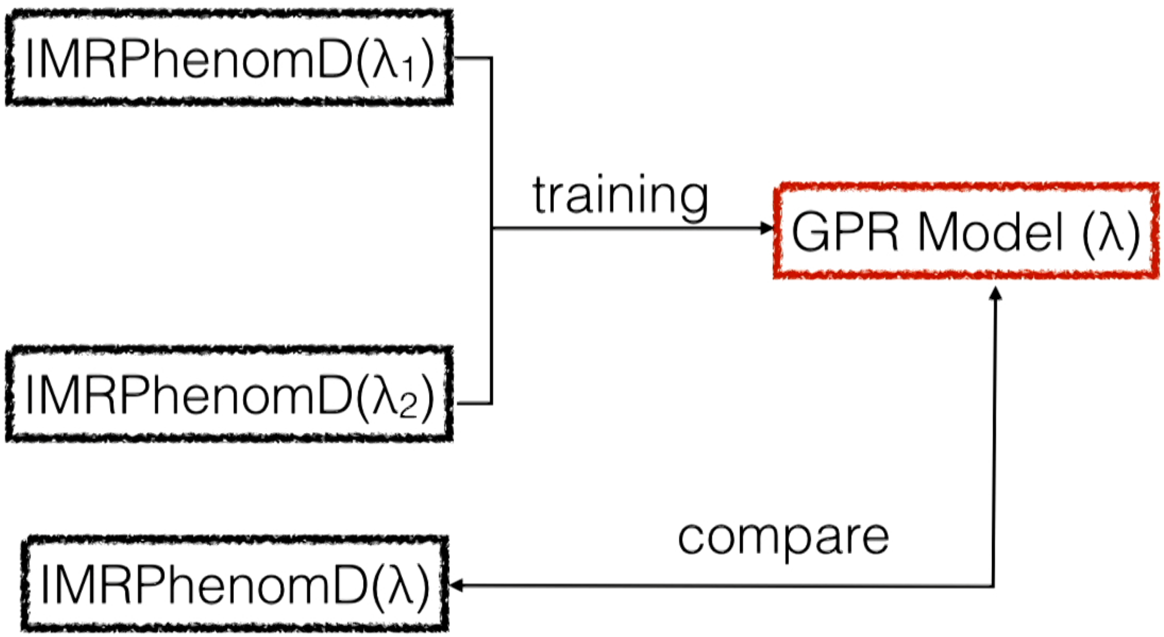
$$p(\mathbf{f}_*|\mathbf{f}) = \mathcal{N}\Big(K(X_*,X)K(X,X)^{-1}\mathbf{f},$$

$$K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)\Big)$$

Disclaimer!

As a proof of concept, IMRPhenomD waveforms are used as the "simulations" for training the GPR.

- IMRPhenomD approximant produces waveforms quickly for specified parameters.
- We can easily check if the GPR model, trained on a few IMRPhenomD waveforms, can reproduce IMRPhenomD across the parameter space.

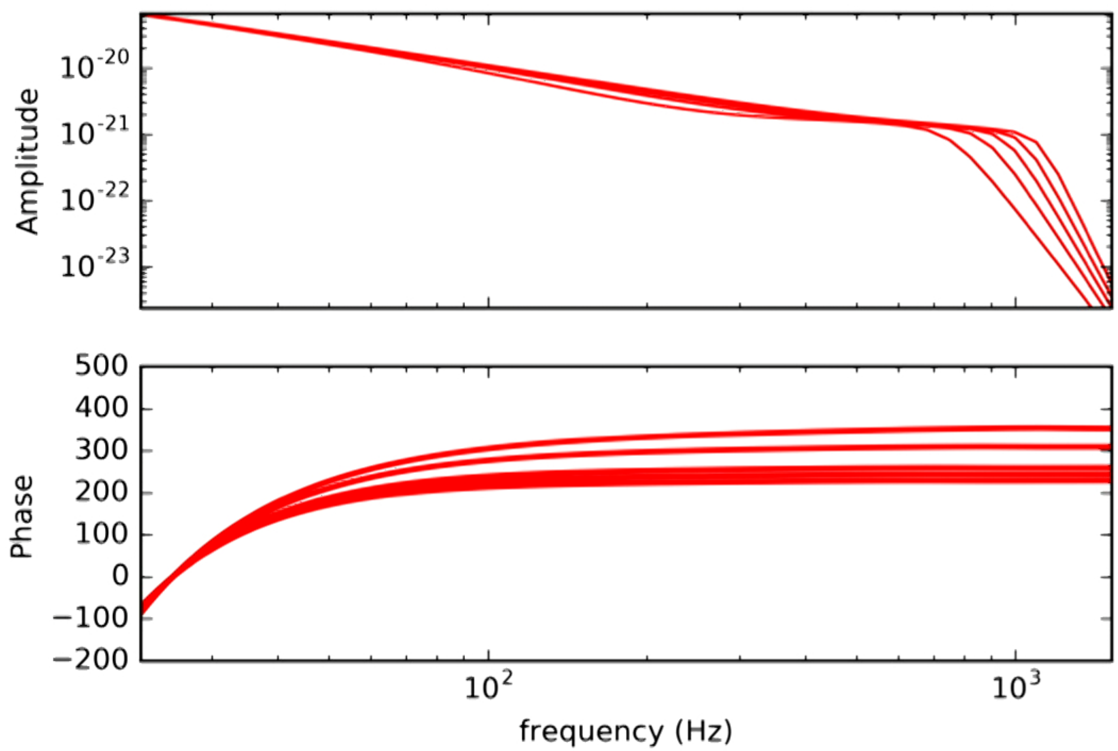


λ's are parameter values that describe the source, e.g. masses, spins

Implementation

- Each simulated waveform will be a long time series — need to reduce dimension of waveforms!
- We regress reduced-order-model (ROM) coefficients of waveforms based on a training set of simulations.
- We use the ROM from Pürrer (2014), an SVD-based model which operates in the frequency domain.

Frequency-Domain Waveforms



The ROM

- Split frequency-domain waveforms into amplitude and phase functions and interpolate these onto a sparse grid in frequency.

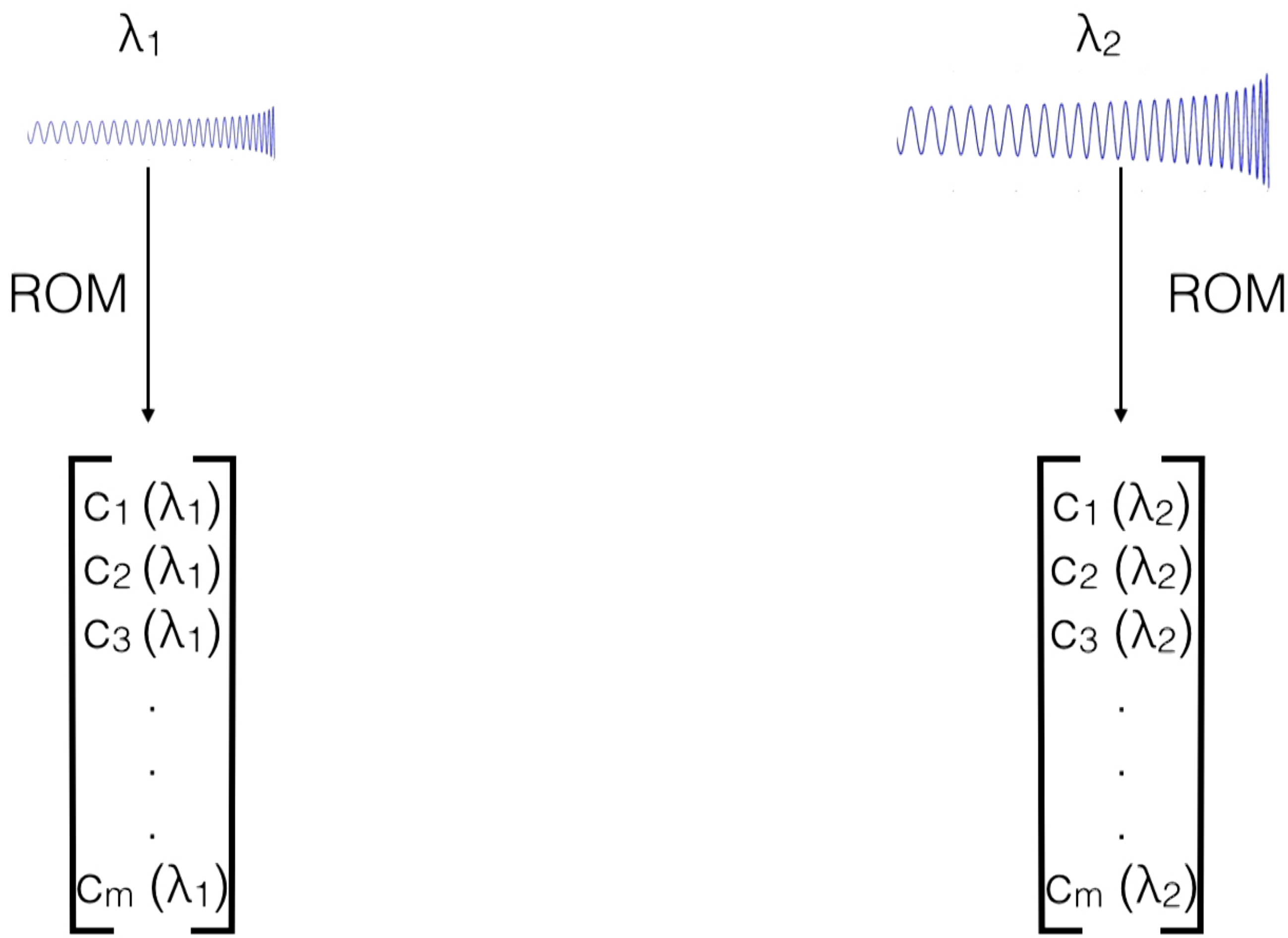
- Pack the interpolated amplitudes and phases for all waveforms into columns of matrices and perform the singular value decomposition:

$$\mathcal{T}_A = V_A \Sigma_A U_A^\top$$

$$\mathcal{T}_\Phi = V_\Phi \Sigma_\Phi U_\Phi^\top$$

- Transform interpolated amplitudes or phases to SVD basis:

$$c(\tau) = V^\top \tau$$


λ1
ROM
c1 (λ1)
c2 (λ1)
c3 (λ1)
.
.
.
cm (λ1)
λ2
ROM
c1 (λ2)
c2 (λ2)
c3 (λ2)
.
.
.
cm (λ2)

The Regression Problem

- In the amplitude/phase SVD bases, each waveform is described by a list of coefficients c_i.
- We want to infer each c_i as a function of the source parameters $\boldsymbol{\lambda}$ given training coefficients.
- Before regression, we de-trend and normalize the training c_i values.
- The regressed, de-trended, normalized coefficients can then be used to reconstruct waveforms.

Covariance Function

- We use the Matern 5/2 kernel:

$$\mathbf{k}_{\nu=5/2}(r) = \sigma^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right)$$

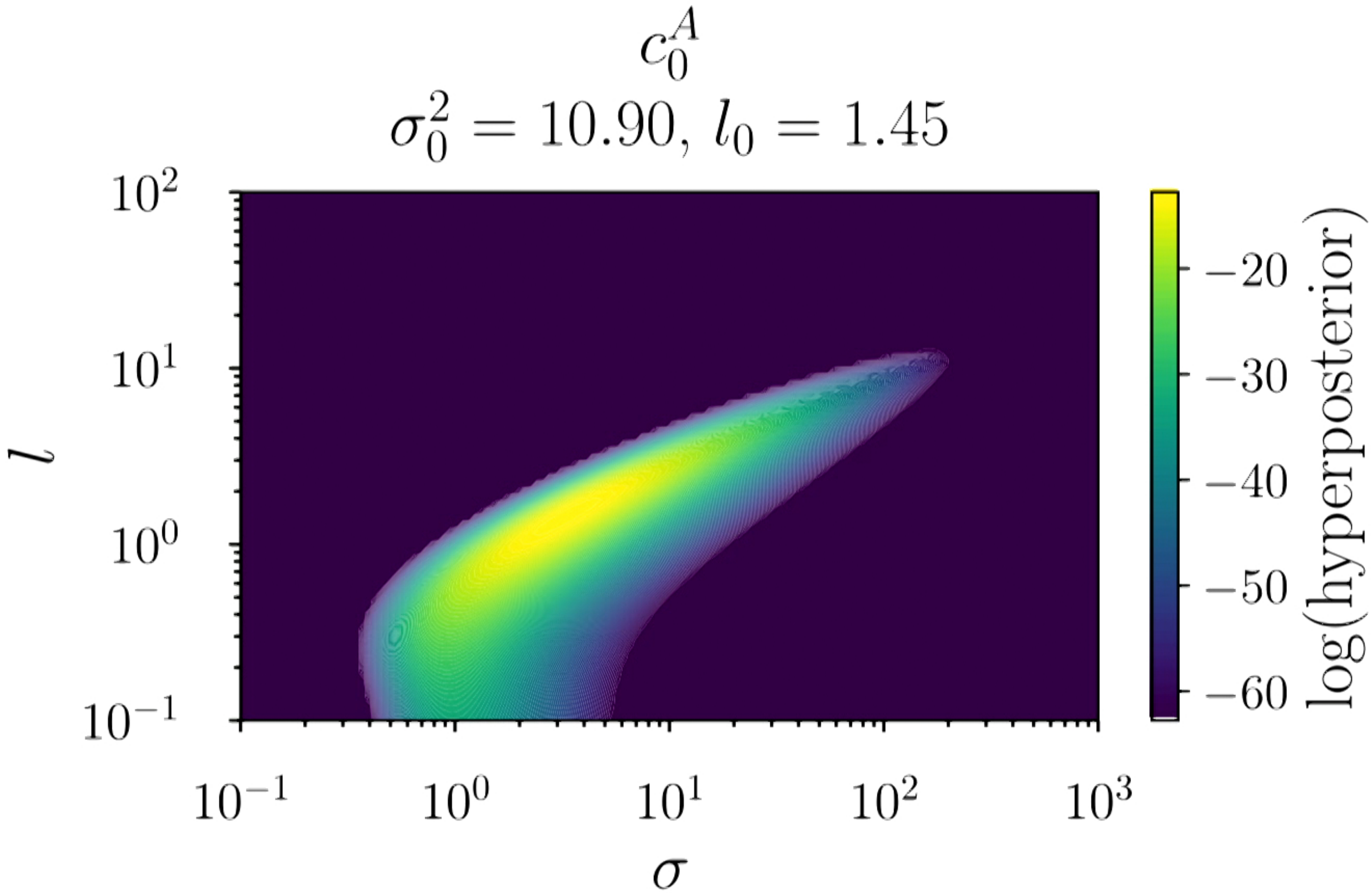
covariance scale (σ^2)

length scale (l)

- For each coefficient, the covariance and length scales (called hyperparameters) are chosen to maximize the *hyperposterior*:

$$\text{hyperposterior} \propto \text{hyperlikelihood} \times \text{hyperprior}$$

- The *hyperlikelihood* is the probability of the training points under the GP prior, and the *hyperprior* is a log-normal prior on each hyperparameter.



c_0^A
$\sigma_0^2 = 10.90,\ l_0 = 1.45$
10^2
10^1
10^0
10^{-1}
l
10^{-1}
10^0
10^1
10^2
10^3
σ
-20
-30
-40
-50
-60
log(hyperposterior)

Gaussian Process Regression

- Test values $f(X_*)$ at points X_* are related to known values f(X) at "training" points X by a normal dist.:

$$\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_*\end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}K(X,X) & K(X,X_*)\\ K(X_*,X) & K(X_*,X_*)\end{bmatrix}\right)$$

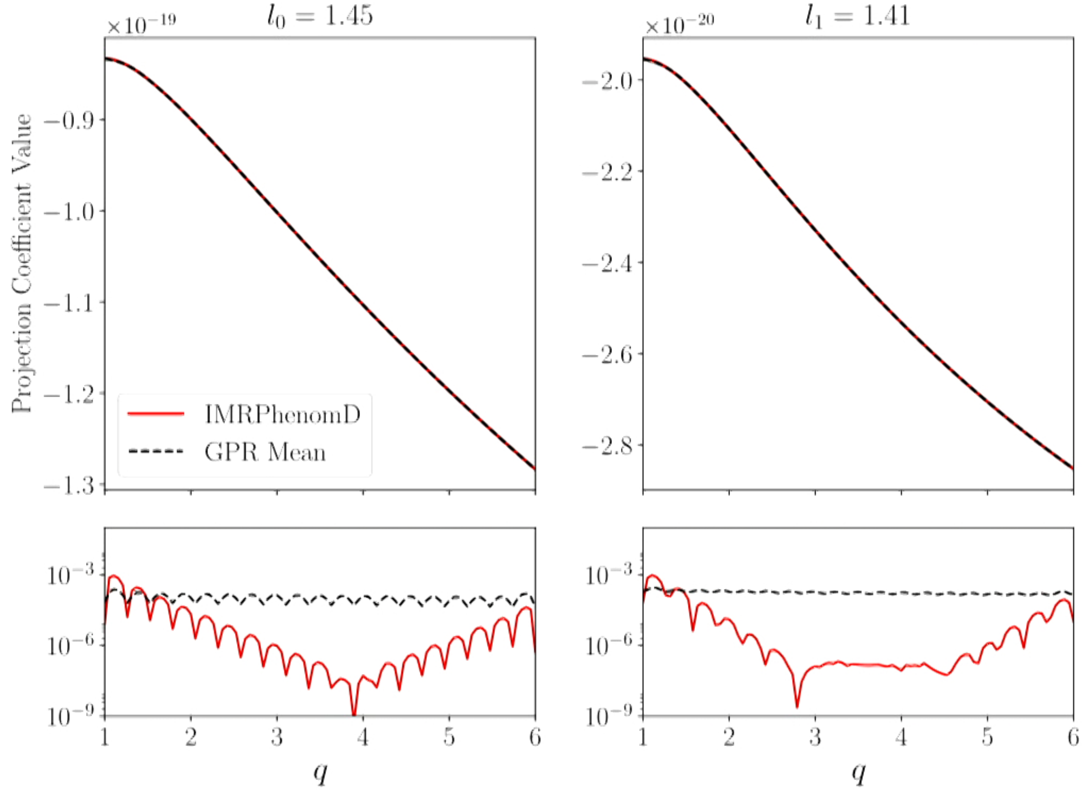
- The elements of the covariance sub-matrices K are calculated using the covariance function.

- Find probability of $f(X_*)$ conditioned on f(X):

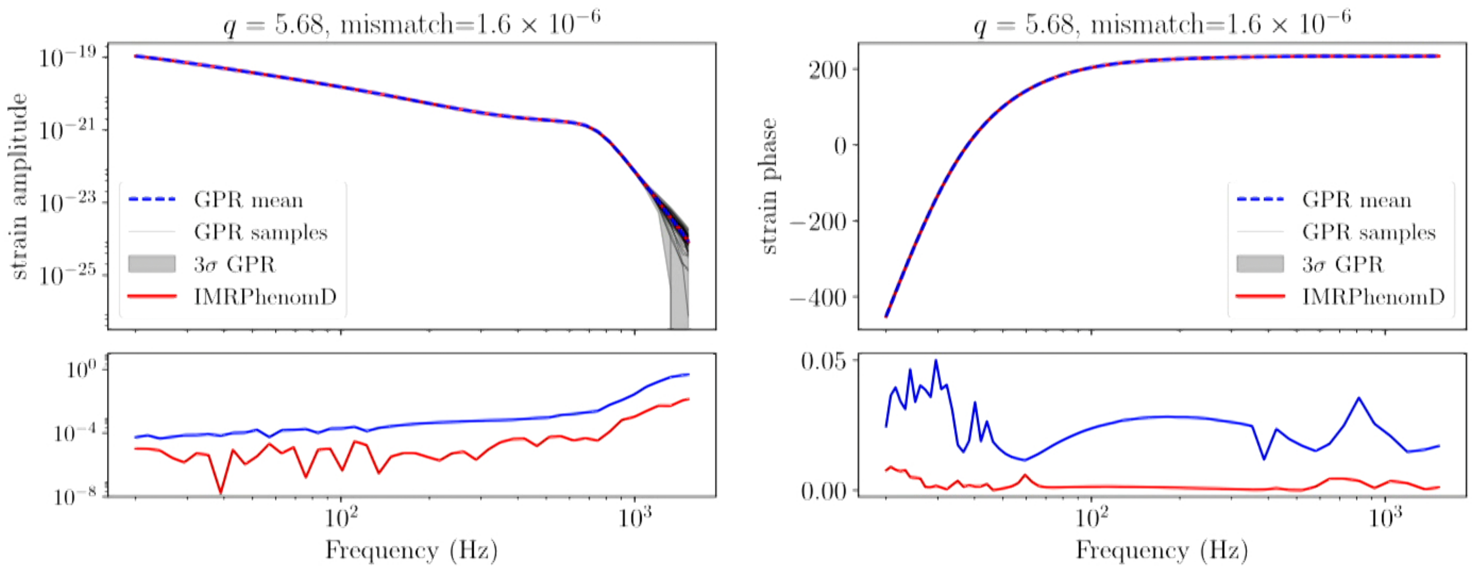
$$p(\mathbf{f}_*|\mathbf{f}) = \mathcal{N}\Big(K(X_*,X)K(X,X)^{-1}\mathbf{f},$$
$$K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)\Big)$$

1D Regression: Mass Ratio

1st two amplitude coefficients



Regressed Waveform Example



How accurate are the regressions?

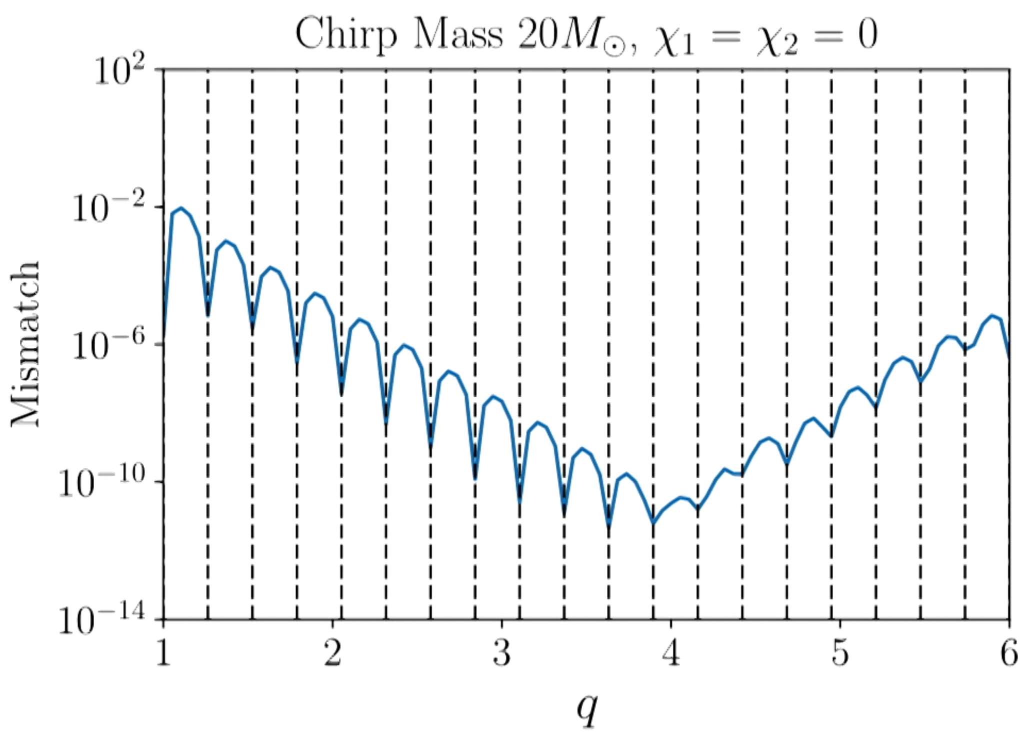
- Want to know how well our GPR model trained on a subset of IMRPhenomD waveforms can reproduce IMRPhenomD waveforms across the space.

- We calculate the *mismatch* between an IMRPhenomD waveform and the GPR mean waveform to quantify the GPR error.

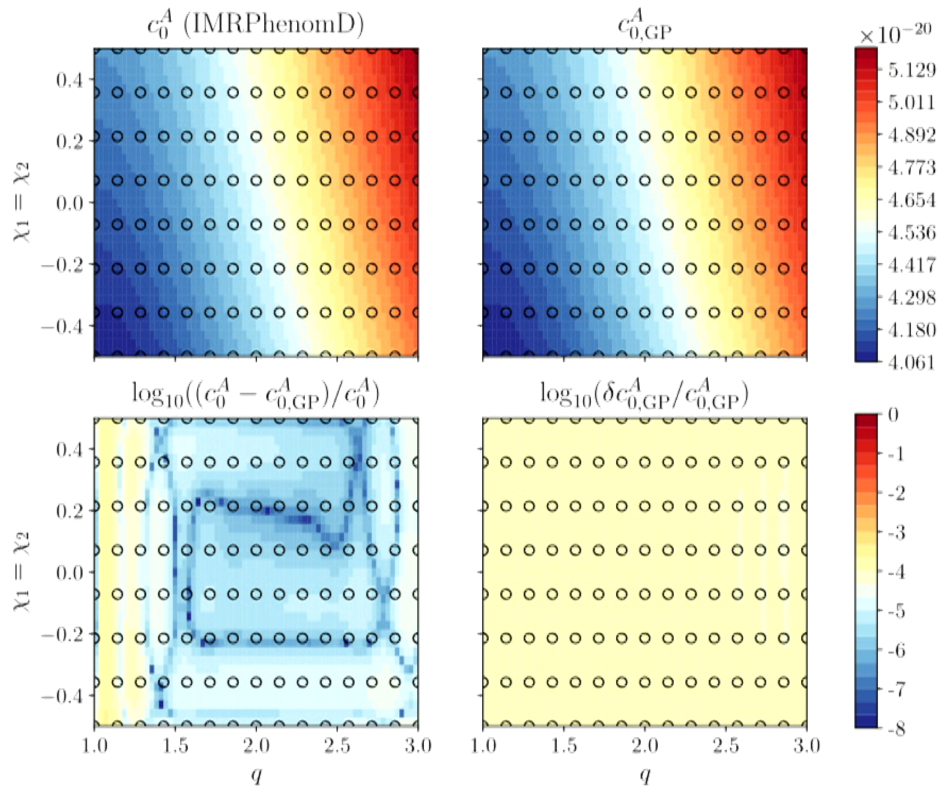
$$\mathrm{mismatch}(h_1, h_2) = 1 - \frac{4}{||h_1||||h_2||} \times \Re\left(\int_{f_{\min}}^{f_{\max}} \frac{h_1(f)h_2^*(f)}{S_f}\mathrm{d}f\right)$$

$$||h|| = 4\Re\left(\int_{f_{\min}}^{f_{\max}} \frac{h(f)h(f)^*}{S_f}\mathrm{d}f\right)$$

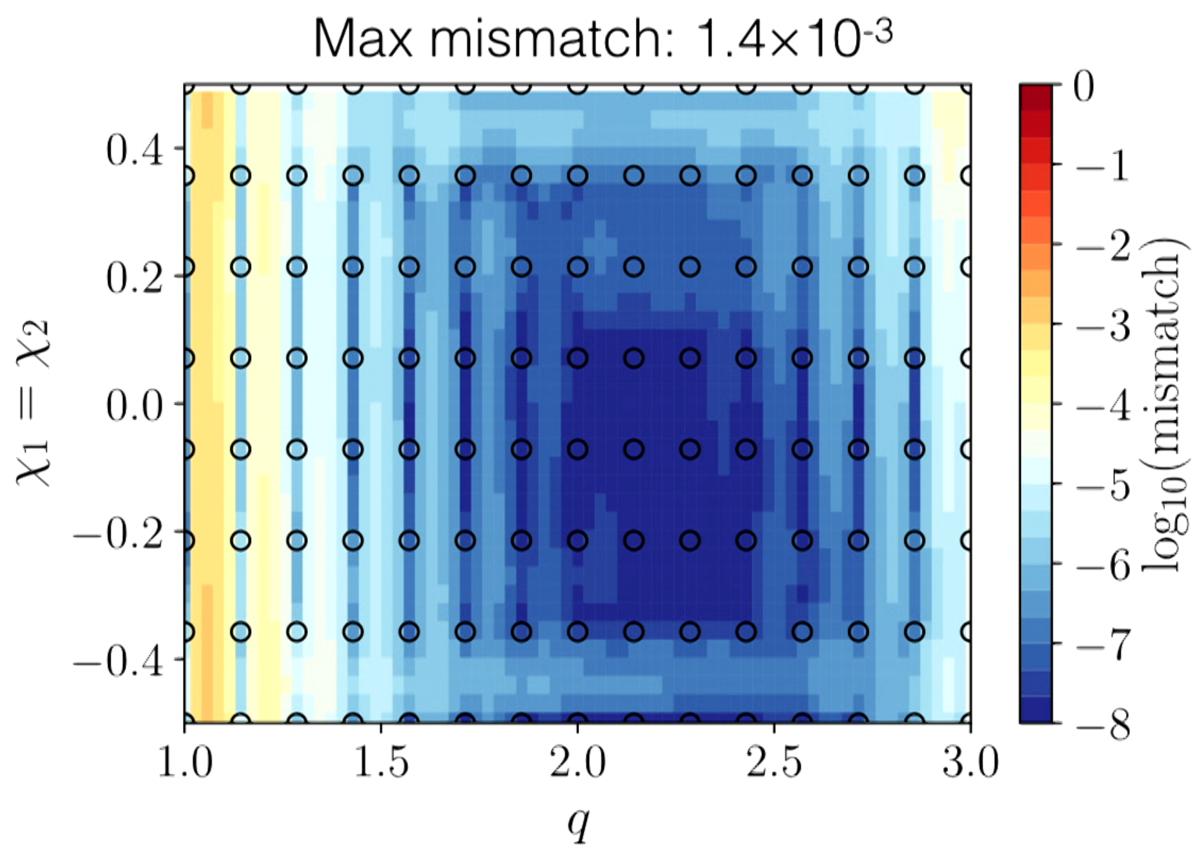
Mismatch between Training Model and GPR



2D Regression: Mass Ratio and Equal, Aligned Spins



2D Regression: Mismatches



27

Estimating the Error

- Use GPR to determine where the error is highest: in practice the true mismatch between a surrogate and NR is not known throughout the parameter space.

- Compare GPR mean to a handful of GPR samples to estimate error level:

$$\texttt{mismatch}_{\mathrm{est}} = \texttt{max}_i \left\{ \texttt{mismatch}(h^{\mathrm{mean}}_{\mathrm{GPR}}(\vec{\lambda}), h^{\mathrm{sample}}_{\mathrm{GPR}}(\vec{\lambda})) \right\}_{i=1}^{n_{\mathrm{samples}}}$$

What to simulate next?

- Our proposal: Choose simulation parameters at parameters where estimated error is maximum.
- In other words, we add a new training waveform to minimize error where estimated error is maximum.
- We call this simulation placement strategy "Minimization of maximum estimated error placement".

Minimization of maximum estimated error placement (MMEEP)

Seed Training Waveforms

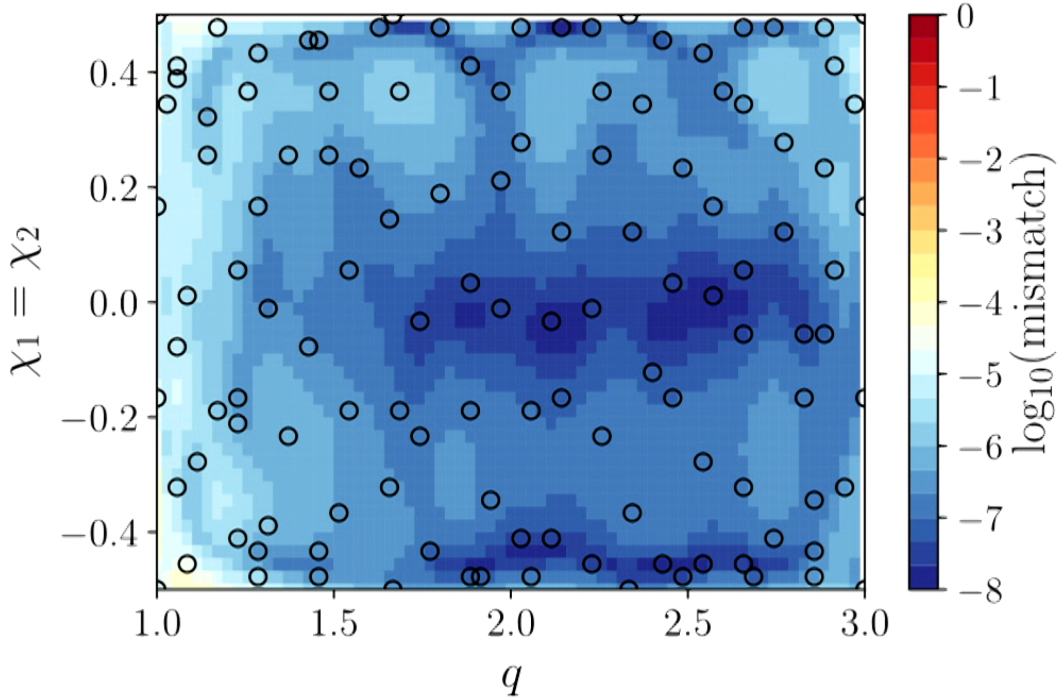
Build GPR ROM from existing simulations

Calculate estimated error across the space

Generate new simulation with parameters argmax(estimated error)

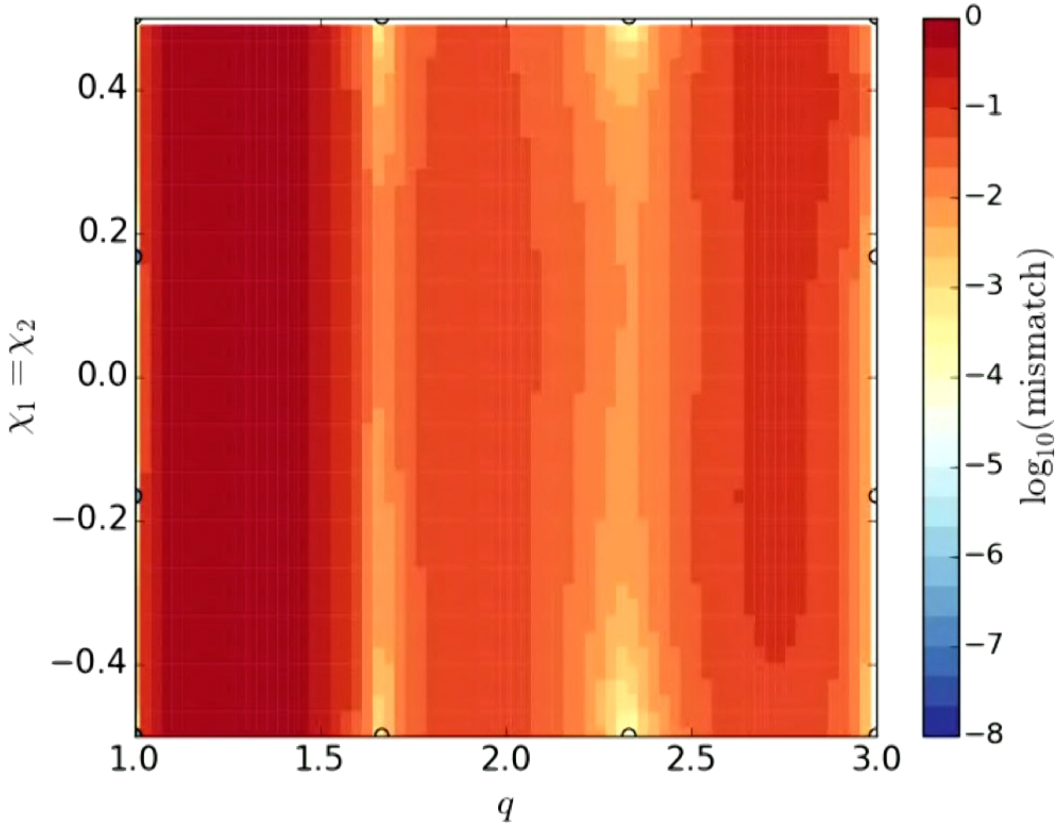
2D Regression using MMEEP

Max mismatch: 9.3×10^{-5}



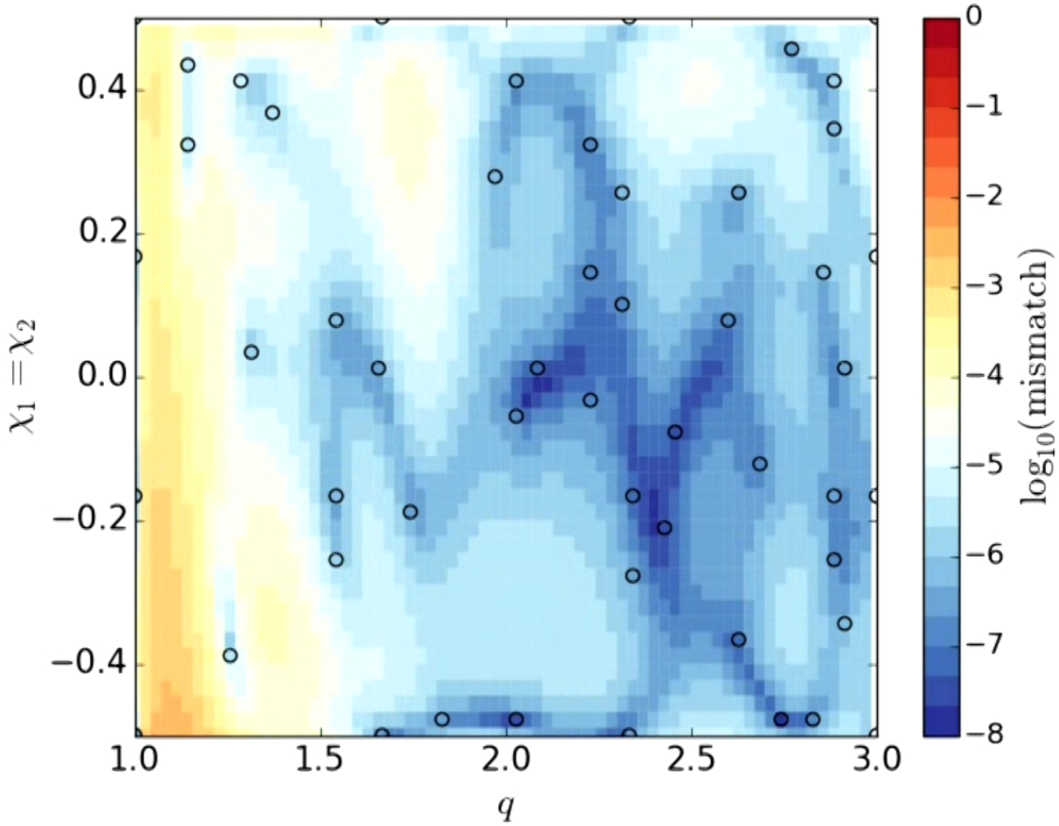
32

MMEEP: The Movie

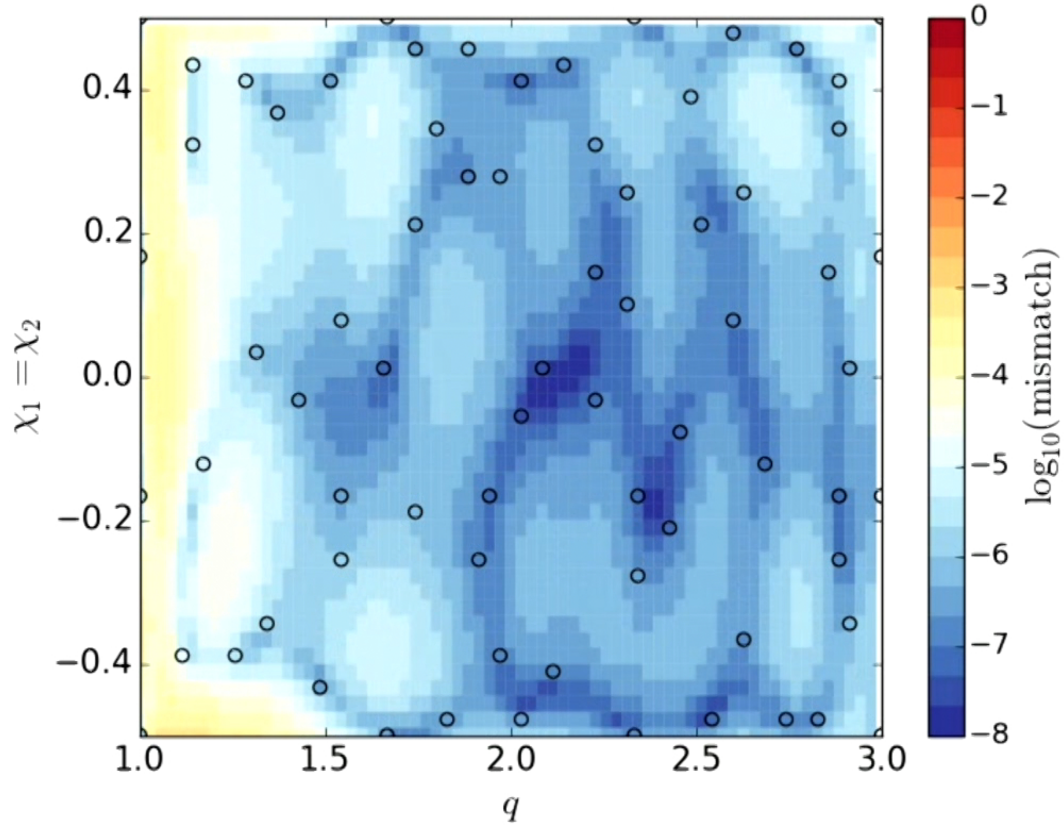


34

MMEEP: The Movie

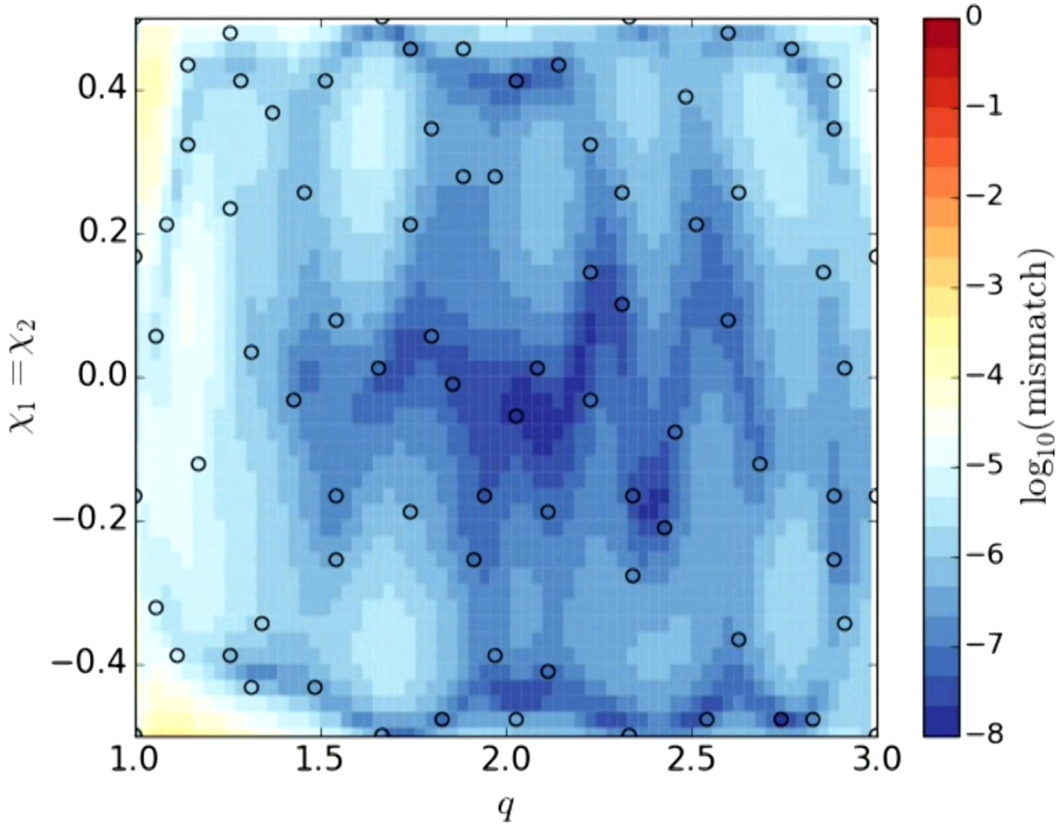


MMEEP: The Movie



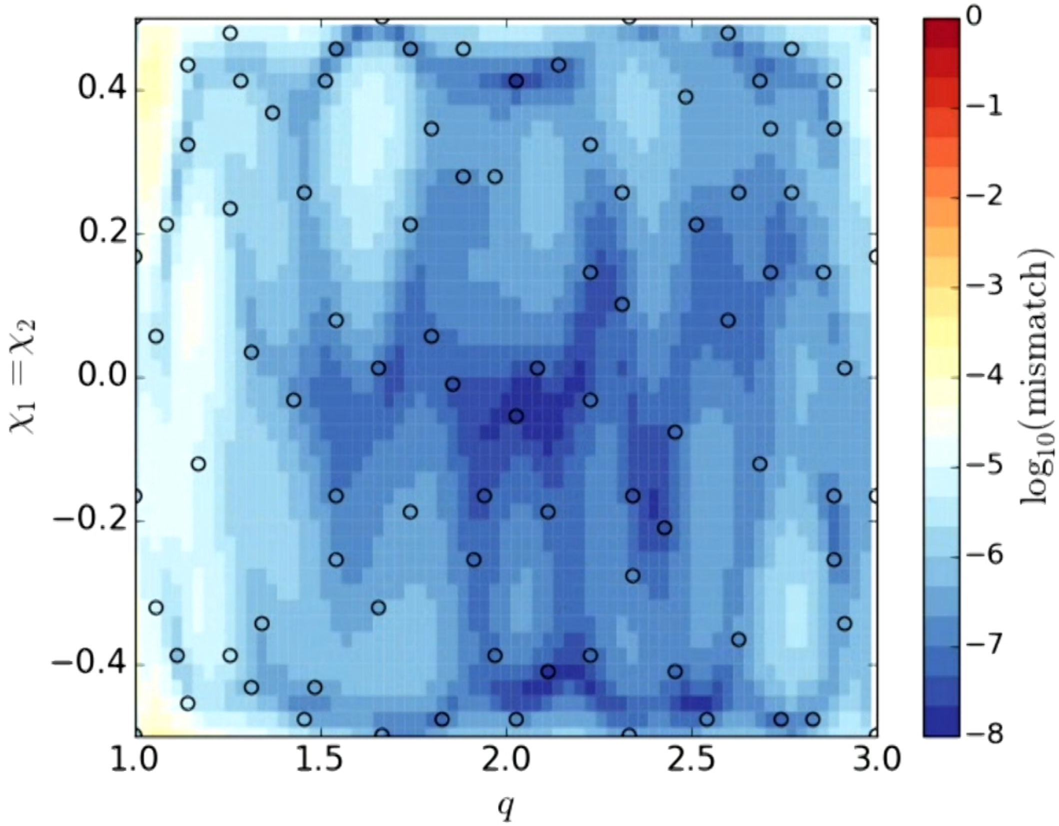
34

MMEEP: The Movie



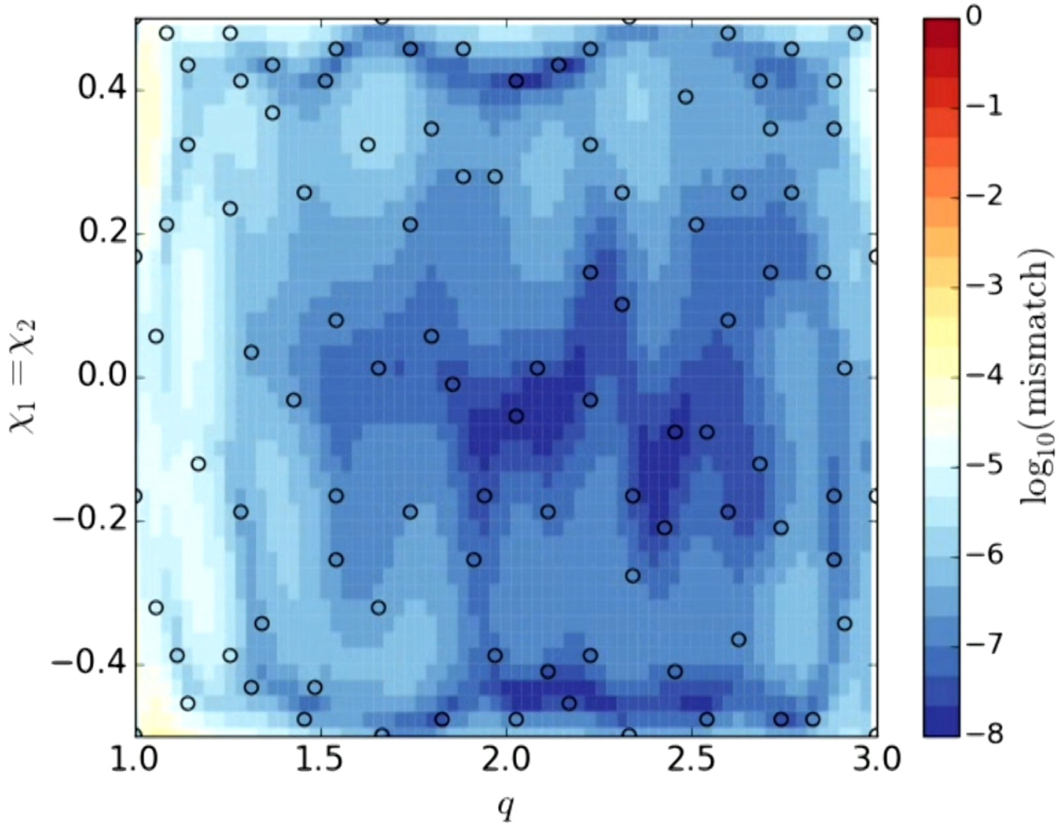
34

MMEEP: The Movie



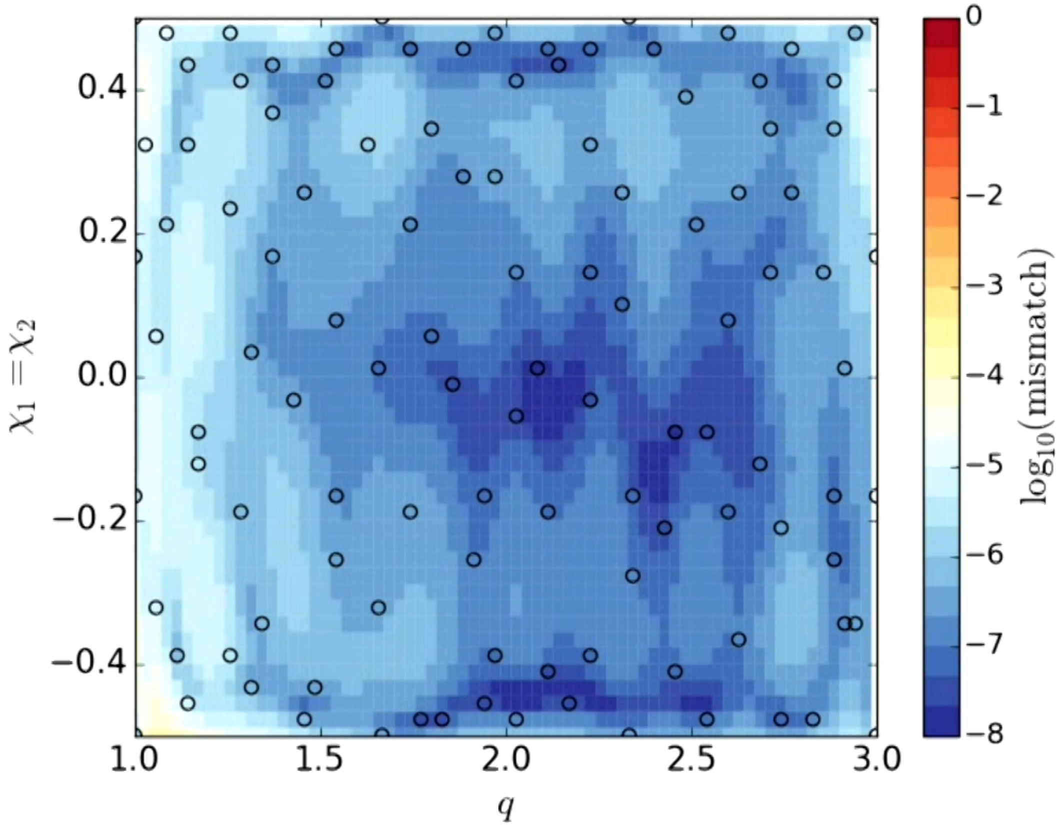
34

MMEEP: The Movie

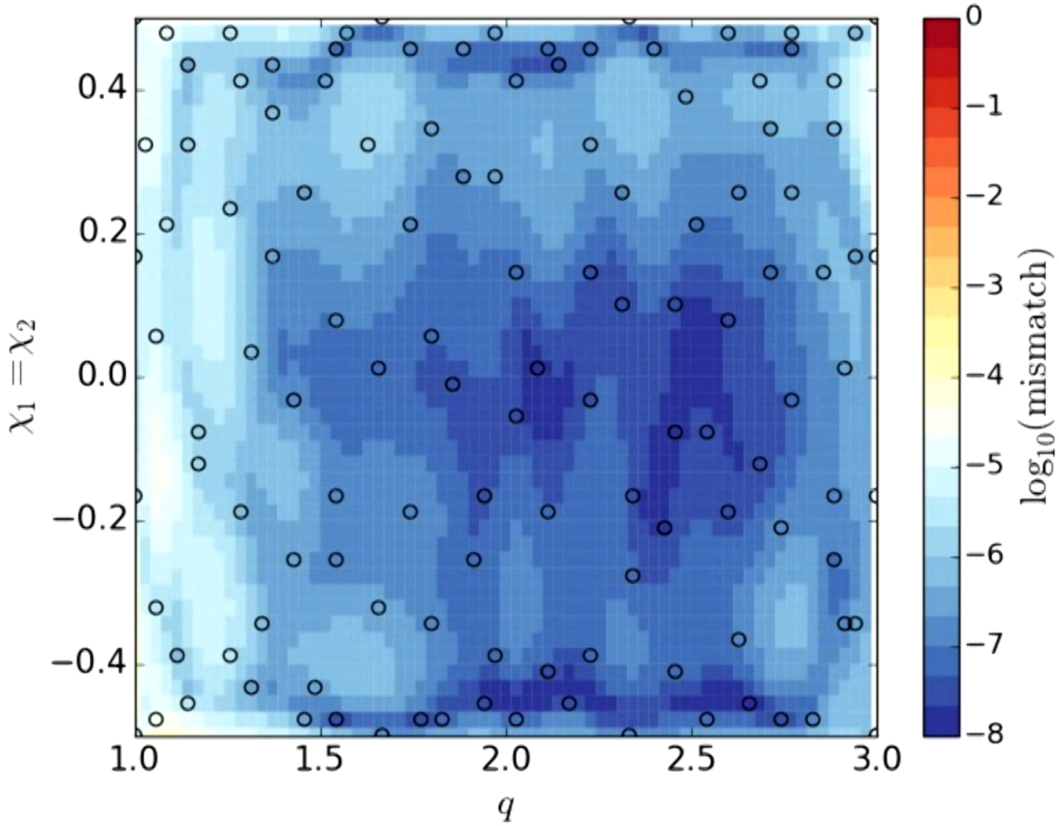


34

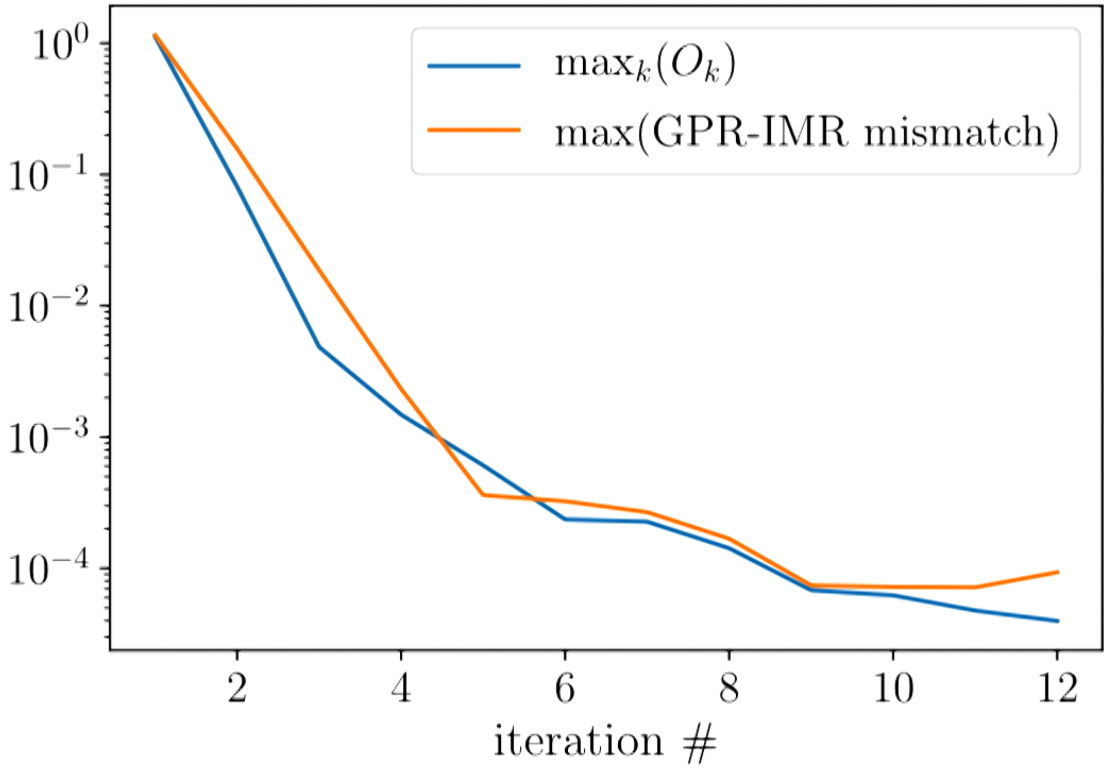
MMEEP: The Movie



MMEEP: The Movie



Can we estimate the maximum error level?



Evaluation Speed

- Recall what is being calculated:

$$p(\mathbf{f}_*|\mathbf{f}) = \mathcal{N}\Big(K(X_*, X)K(X, X)^{-1}\mathbf{f},$$

$$K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)\Big)$$

2: $L := \text{cholesky}(K + \sigma_n^2 I)$
$\boldsymbol{\alpha} := L^\top \backslash (L \backslash \mathbf{y})$
4: $\bar{f}_* := \mathbf{k}_*^\top \boldsymbol{\alpha}$
$\mathbf{v} := L \backslash \mathbf{k}_*$
6: $\mathbb{V}[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}$

Practical Implementation
(calculating A**b** takes $n^2/2$,
where **x**=A**b** solves A**x**=**b**)

Timing

- Evaluation time depends on # of training points.
- There is significant overhead in our current computations.
- With 1000 training points, and no parallelization:
  - ~ 100 ms/waveform with no overhead
  - ~ 5 s/waveform with overhead

The method is general…

- A few different choices could be made within the GPR framework presented here. E.g.:
  - employ another ROM
  - use different kernels
  - modify MMEEP

Conclusion

- We have shown that this method
  - produces accurate waveforms with Gaussian uncertainties,
  - allows fast waveform evaluation,
  - naturally suggests parameter values for new simulations.