SURROGATE MODELS FOR GRAVITATIONAL WAVE ASTRONOMY

NIKOLAOS STERGIOULAS

DEPARTMENT OF PHYSICS

ARISTOTLE UNIVERSITY OF THESSALONIKI

4th G2Net Training School, Thessaloniki, March 29, 2023

GRAVITATIONAL WAVE TEMPLATE BANKS

- **SEOBNRv4** (Bohe´et al. 2017)
- Spin-aligned

$$\chi_i = \pm \frac{|S_i|}{m_i^2}$$

Intrinsic parameters are

$$\boldsymbol{\lambda} = (q, \chi_1, \chi_2)$$

where the mass ratio is

$$q = m_1/m_2$$

- Effective One Body: binary motion is replaced by the effective one-body problem of a test particle in a deformed Schwarzschild metric, with the symmetric mass ratio as the deformation parameter:

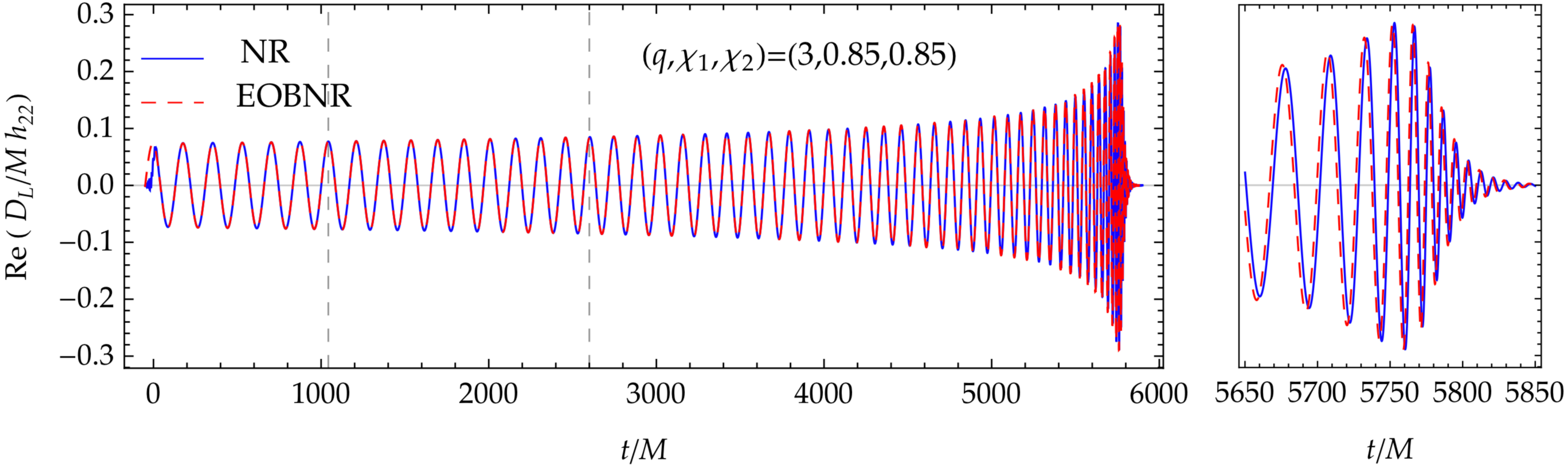
$$\eta = \frac{m_1 m_2}{M^2} = \frac{q}{(1+q)^2}$$

Includes all 4PN terms in the EOB radial potential + higher-order PN terms that are calibrated with numerical relativity simulations.

Describes dominant (2,2) mode including the merger-ringdown phase.

SEOBNRv4 TEMPLATE BANK

- Numerical-Relativity Calibrated to 141 simulations



- O(300,000) waveforms are used in detection and parameter estimation of BBH mergers using matched filtering -> very costly computations!

- Extension to random spins impractical -> construct fast surrogate models + use machine learning for rapid detection and parameter estimation!

SAMPLE SEOBNRv4 WAVEFORM

- Strain

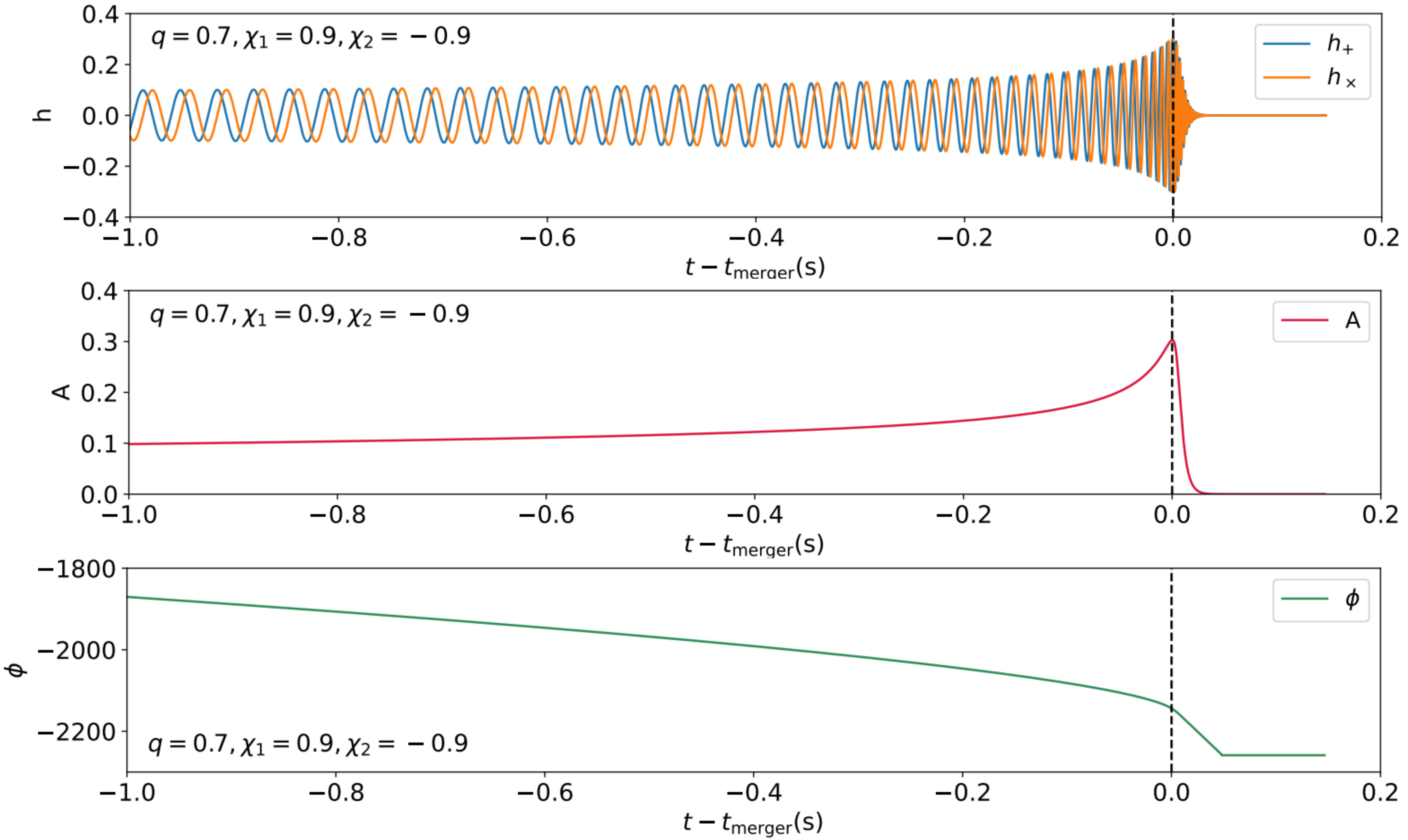
$$h = h_+ - ih_\times$$

- Amplitude

$$A = \sqrt{h_+^2 + h_\times^2}$$

- Phase

$$\phi = \tan^{-1}\left(-\frac{h_\times}{h_+}\right)$$



REDUCED BASIS

- Complex waveform: $$h(t;\boldsymbol{\lambda}) = h_+(t;\boldsymbol{\lambda}) - ih_\times(t;\boldsymbol{\lambda})$$

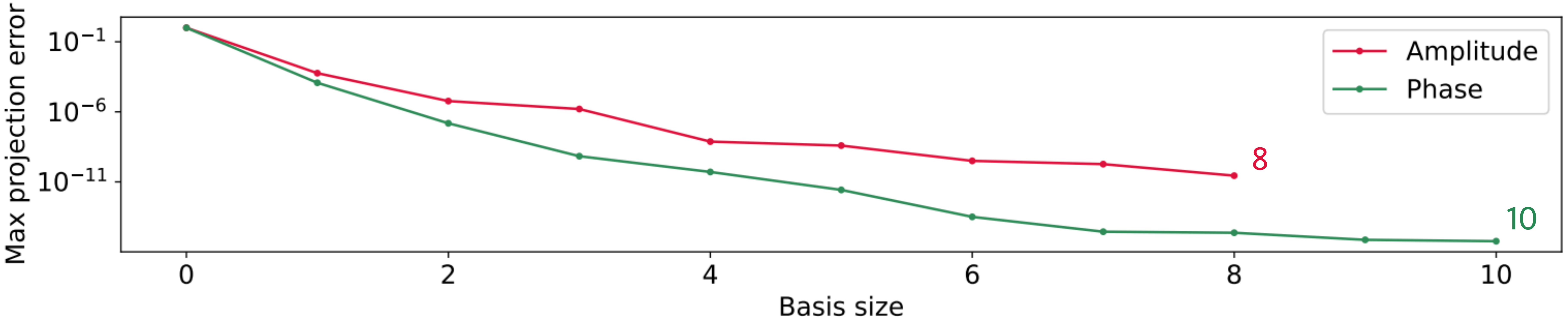
where $\boldsymbol{\lambda} = (q, \chi_1, \chi_2)$ = intrinsic parameters

- Use N waveforms, randomly chosen in range $0 \leq q \leq 8, \quad -1 \leq \chi_{1,2} \leq 1$ with mass $M = 60M_\odot$

- Construct reduced basis $e_j(t)$ of size $n < N$, using an iterative greedy algorithm and Gram-Schmidt orthonormalization

$$h(t;\boldsymbol{\lambda}_i) \approx \sum_{j=1}^{n} c_j(\boldsymbol{\lambda}_i)e_j(t) \qquad i = 1,\ldots,N$$

- Example: nonspinning N = 40 & tolerance = 10^{-12}, then n = 8 for amplitude and n= 10 for phase.



EMPIRICAL INTERPOLATION METHOD

- Construct Empirical Interpolation Method (EIM) basis $B_k(t)$ of size n
- The coefficients $\alpha_k(\boldsymbol{\lambda}_j)$ coincide with the waveform at the empirical time nodes $\{T_k\}_{k=1}^n$, i.e.

$$\alpha_k(\boldsymbol{\lambda}_j) = h(T_k; \boldsymbol{\lambda}_j)$$

- Then, for the greedy points:

$$h(t; \boldsymbol{\lambda}_j) = \sum_{k=1}^{n} \alpha_k(\boldsymbol{\lambda}_j) B_k(t) \qquad j = 1, \ldots, n$$

- and for any other waveform in the training set:

$$\alpha_k(\boldsymbol{\lambda}_i) = h(T_k; \boldsymbol{\lambda}_i)$$

(no costly projection is required!)

- Calculations with Chad Galley's `Rompy` package.

MISMATCH BETWEEN TWO WAVEFORMS

- Inner product

$$\langle h(\cdot;\boldsymbol{\lambda}_1), h(\cdot;\boldsymbol{\lambda}_2)\rangle = 4\Re \int_{f_{min}}^{f_{max}} \frac{\tilde{h}(f;\boldsymbol{\lambda}_1)\tilde{h}^*(f;\boldsymbol{\lambda}_2)}{S_n(f)} df$$

power spectral density (PSD)

- Scaled waveform spectrum

$$\hat{h}(f;\boldsymbol{\lambda}) = \frac{\tilde{h}(f;\boldsymbol{\lambda})}{\langle h(\cdot;\boldsymbol{\lambda}), h(\cdot;\boldsymbol{\lambda})\rangle}$$

- Overlap

$$\mathcal{O}(\hat{h}(\cdot;\boldsymbol{\lambda}_1), \hat{h}(\cdot;\boldsymbol{\lambda}_2)) = \max_{t_0,\phi_0} \langle h(\cdot;\boldsymbol{\lambda}_1), h(\cdot;\boldsymbol{\lambda}_2)\rangle$$

- Mismatch

$$\boxed{\mathcal{M}(\hat{h}(\cdot;\boldsymbol{\lambda}_1), \hat{h}(\cdot;\boldsymbol{\lambda}_2)) = 1 - \mathcal{O}(\hat{h}(\cdot;\boldsymbol{\lambda}_1), \hat{h}(\cdot;\boldsymbol{\lambda}_2))}$$

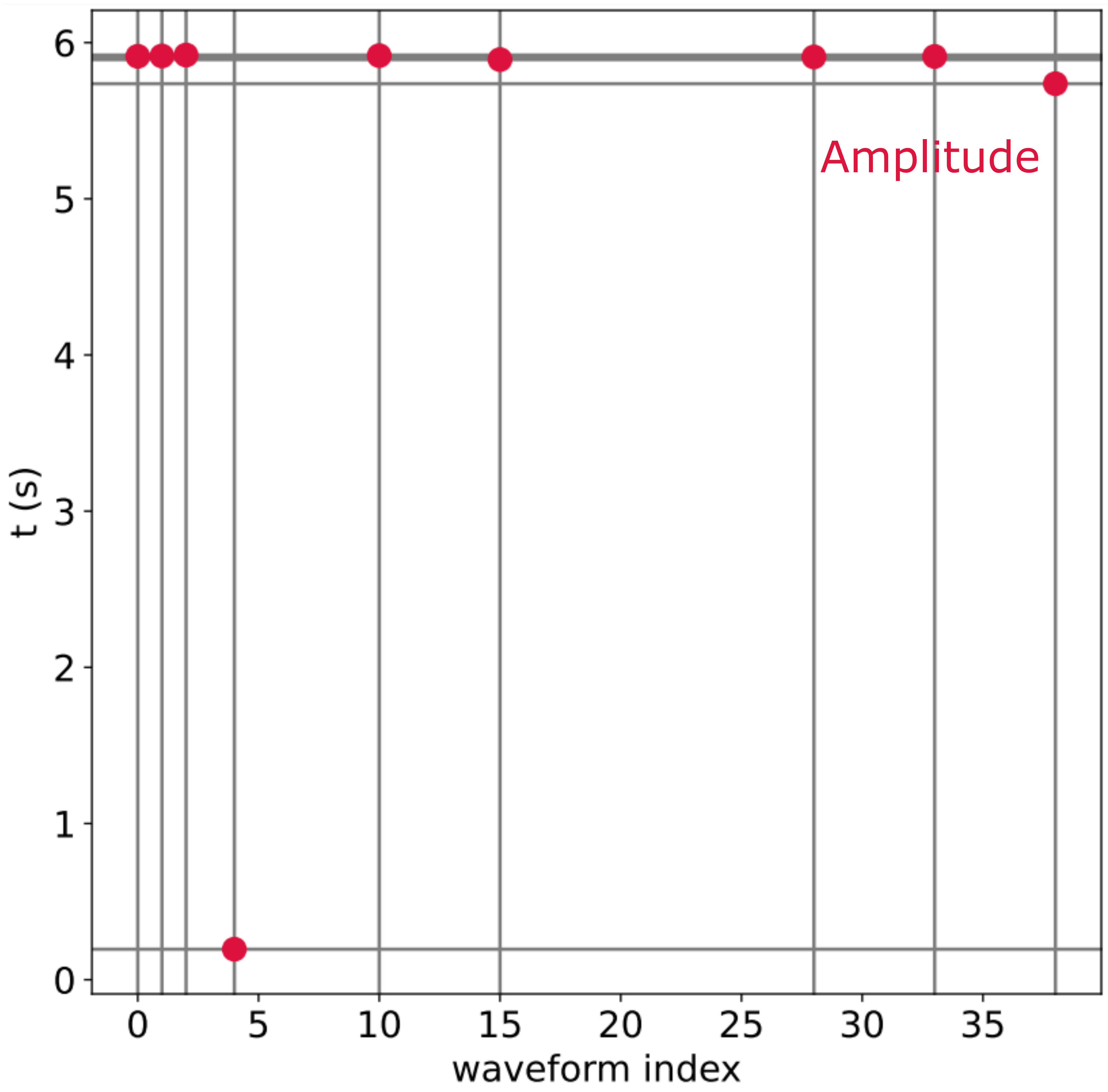
- Use N = 2 x 10^5 waveforms in range $0 \leq q \leq 8, \quad -1 \leq \chi_{1,2} \leq 1$

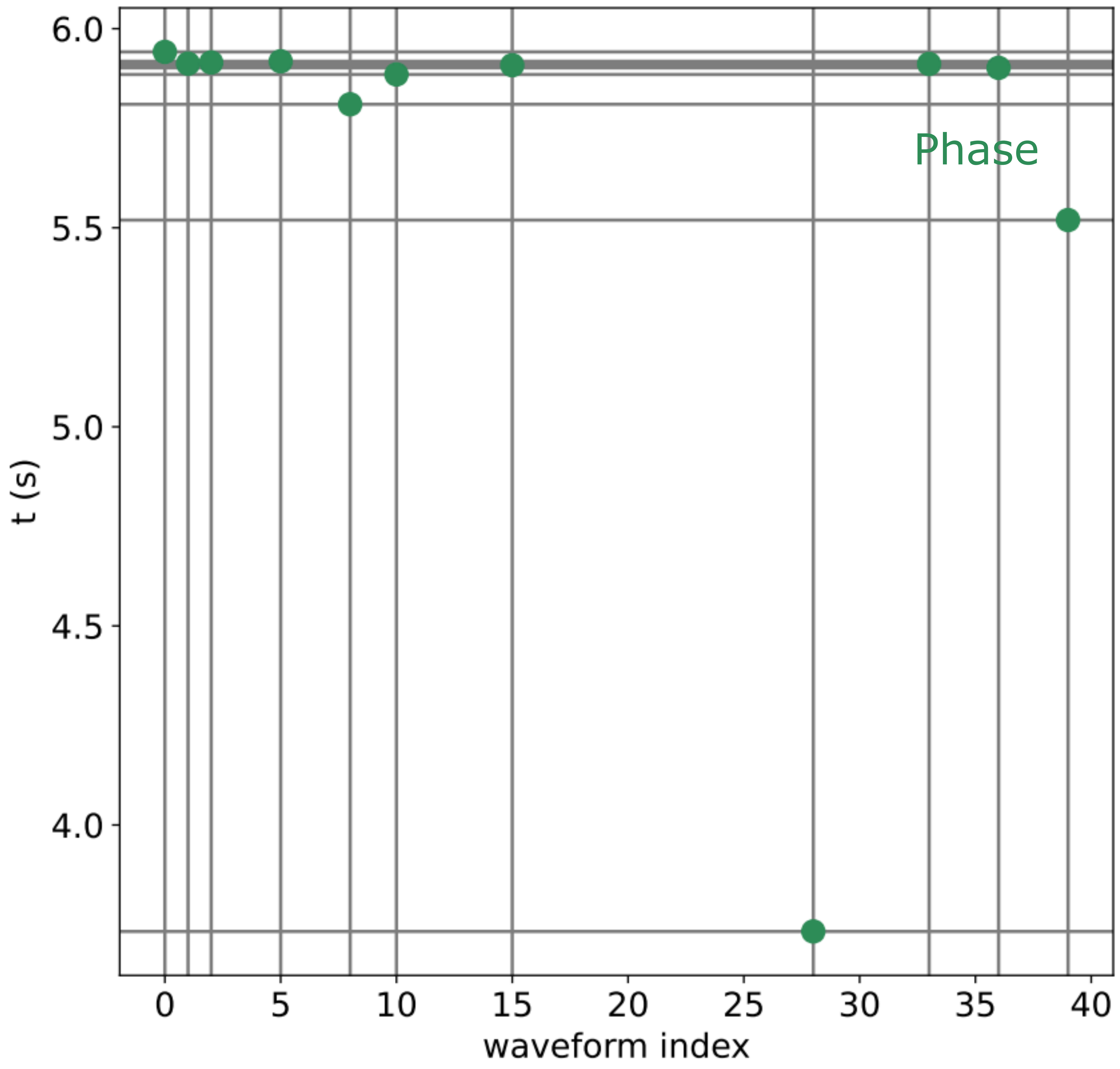
Greedy Tolerance	n (amplitude)	n (phase)	mismatch $\mathcal{M}$ (max)	(median)	(95^{th} percentile)
10^{-6}	8	4	8.44×10^{-3}	5.47×10^{-4}	1.81×10^{-3}
10^{-8}	13	4	8.44×10^{-3}	5.45×10^{-4}	1.80×10^{-3}
10^{-10}	18	8	4.95×10^{-4}	1.30×10^{-5}	8.22×10^{-5}
10^{-12}	41	12	2.07×10^{-6}	7.45×10^{-8}	2.83×10^{-7}
10^{-14}	84	32	1.34×10^{-8}	5.64×10^{-10}	3.95×10^{-9}
10^{-16}	93	48	6.60×10^{-9}	4.59×10^{-10}	3.02×10^{-9}

baseline for comparisons

WAVEFORMS AND EMPIRICAL TIMES OF THE EIM BASIS

- Example: nonspinning $N = 40$ & tolerance = 10^{-12}, then $n = 8$ for amplitude and $n = 10$ for phase.



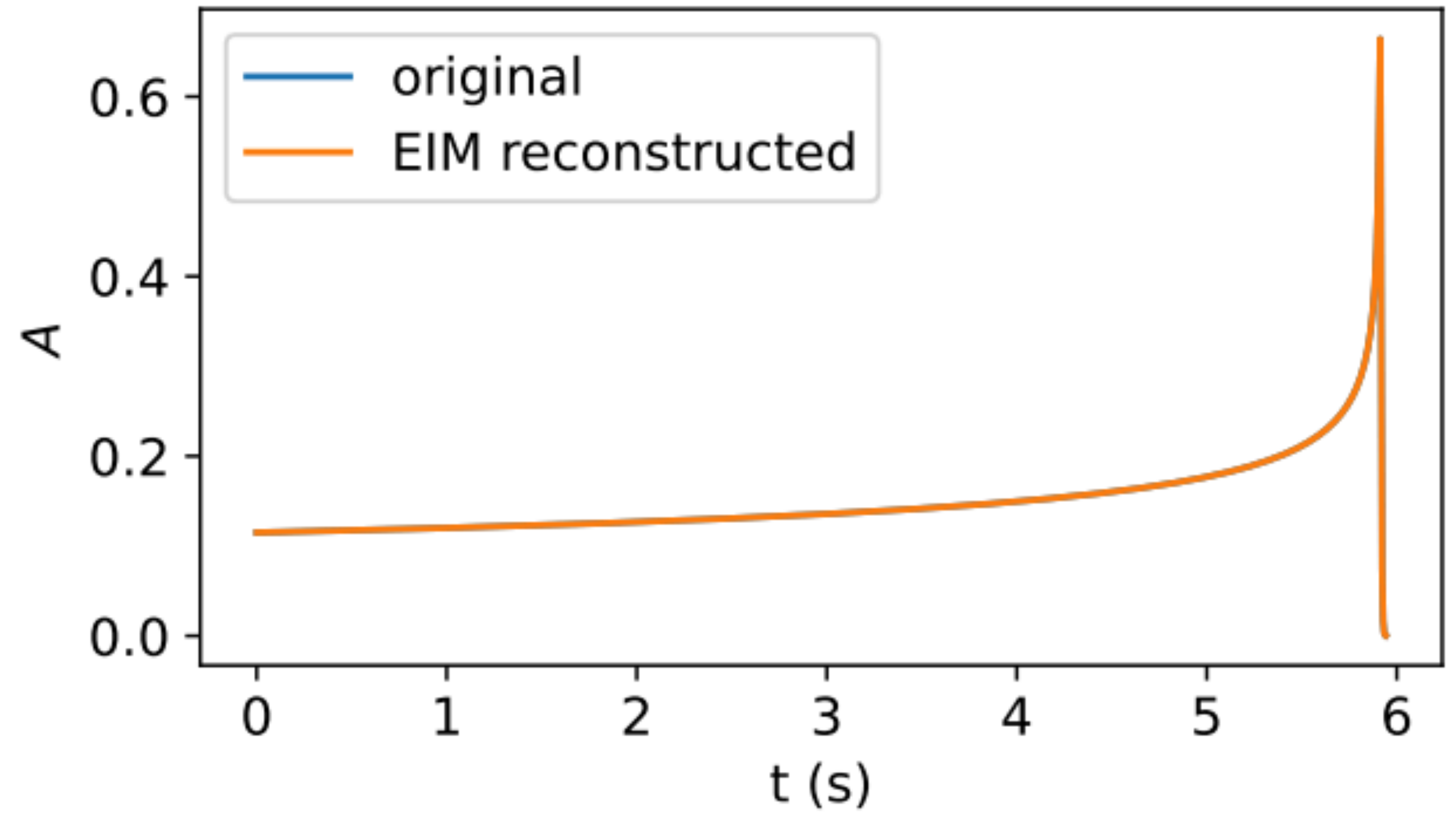


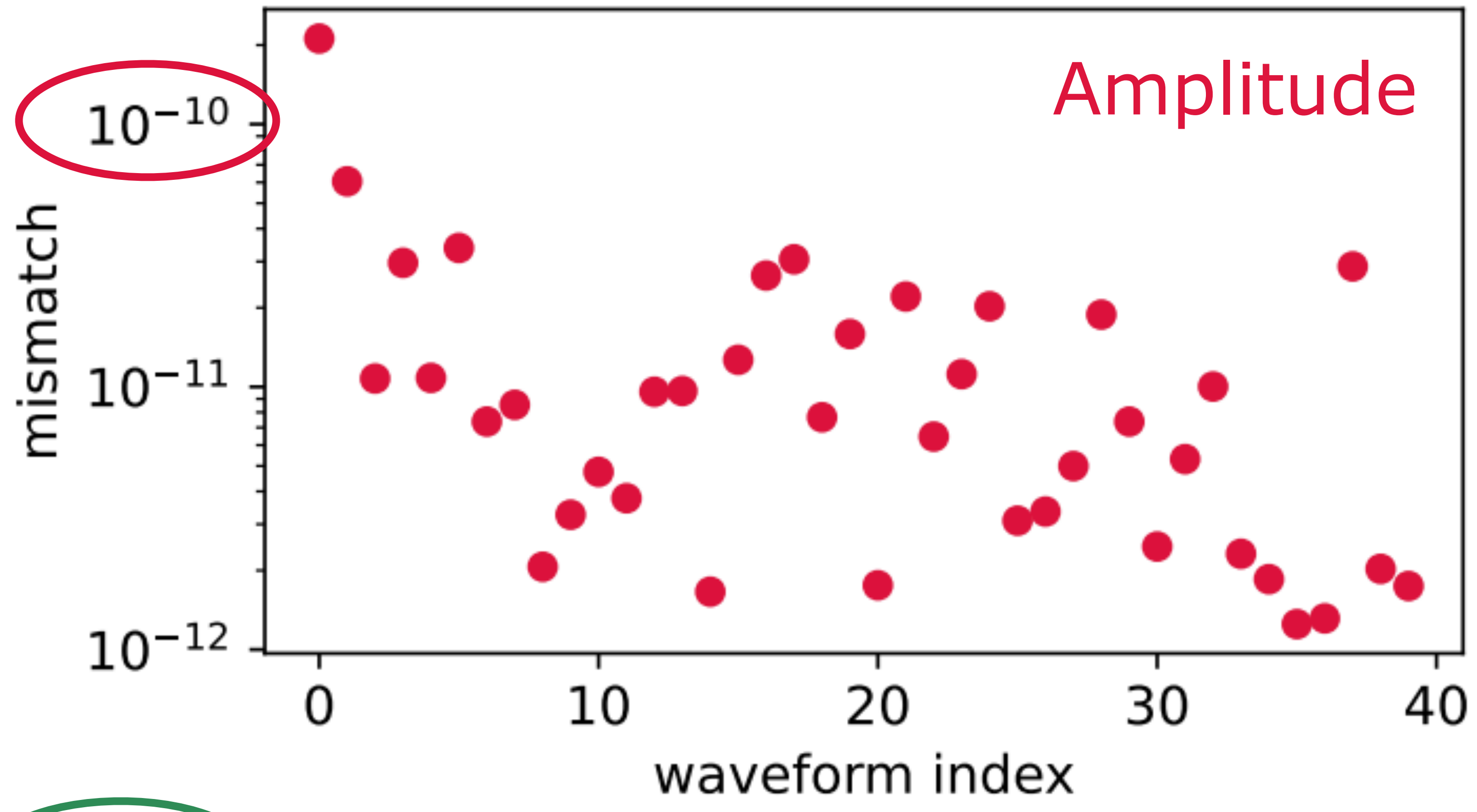
SURROGATE MODEL

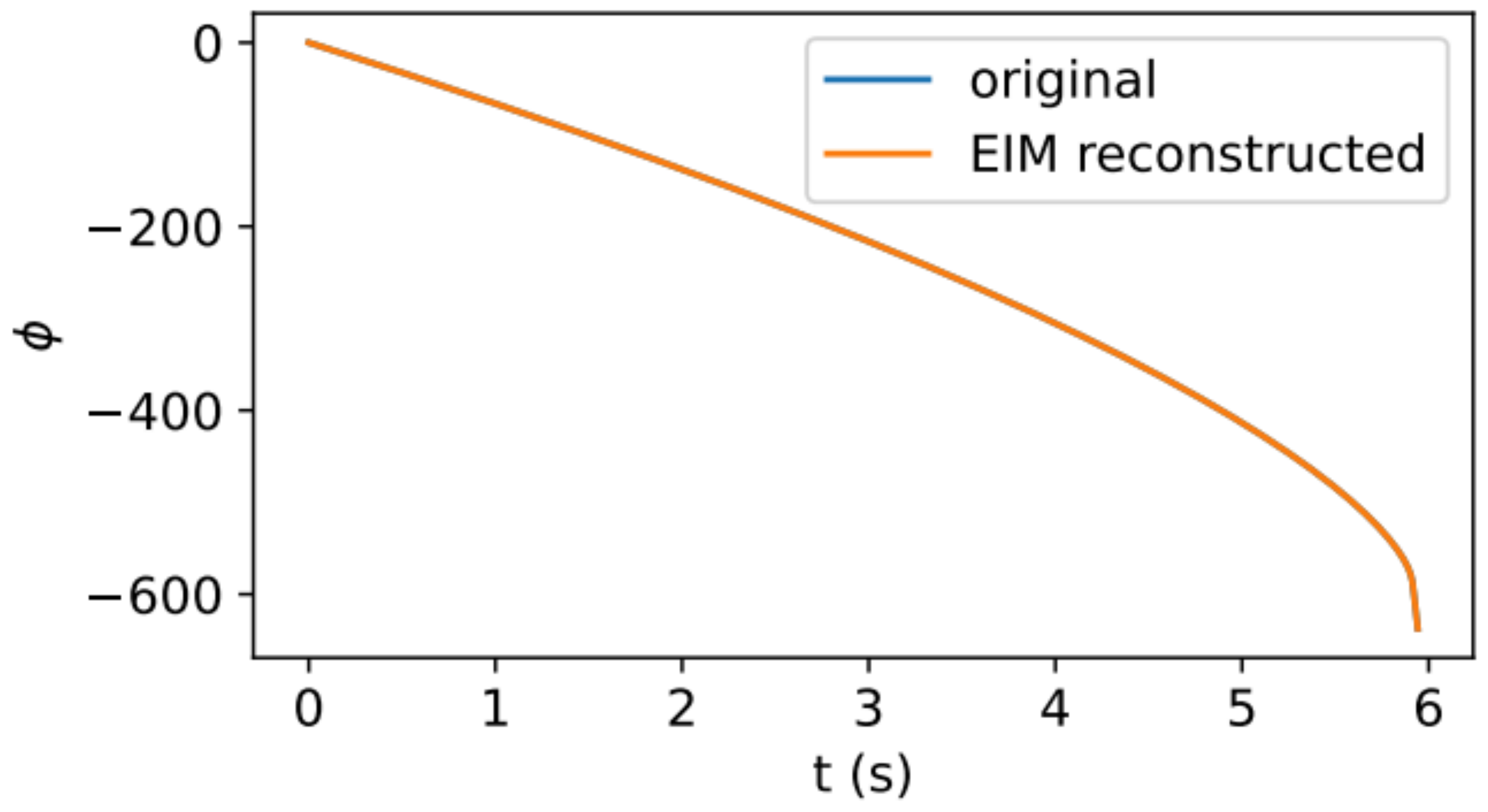
- Interpolating the EIM coefficients $\alpha_k(\boldsymbol{\lambda}_i)$ of the training set for arbitrary values $\boldsymbol{\lambda}$ of the intrinsic parameters, we obtain the interpolated EIM coefficients $\hat{\alpha}_k(\boldsymbol{\lambda})$
- A surrogate model of the waveform model is then

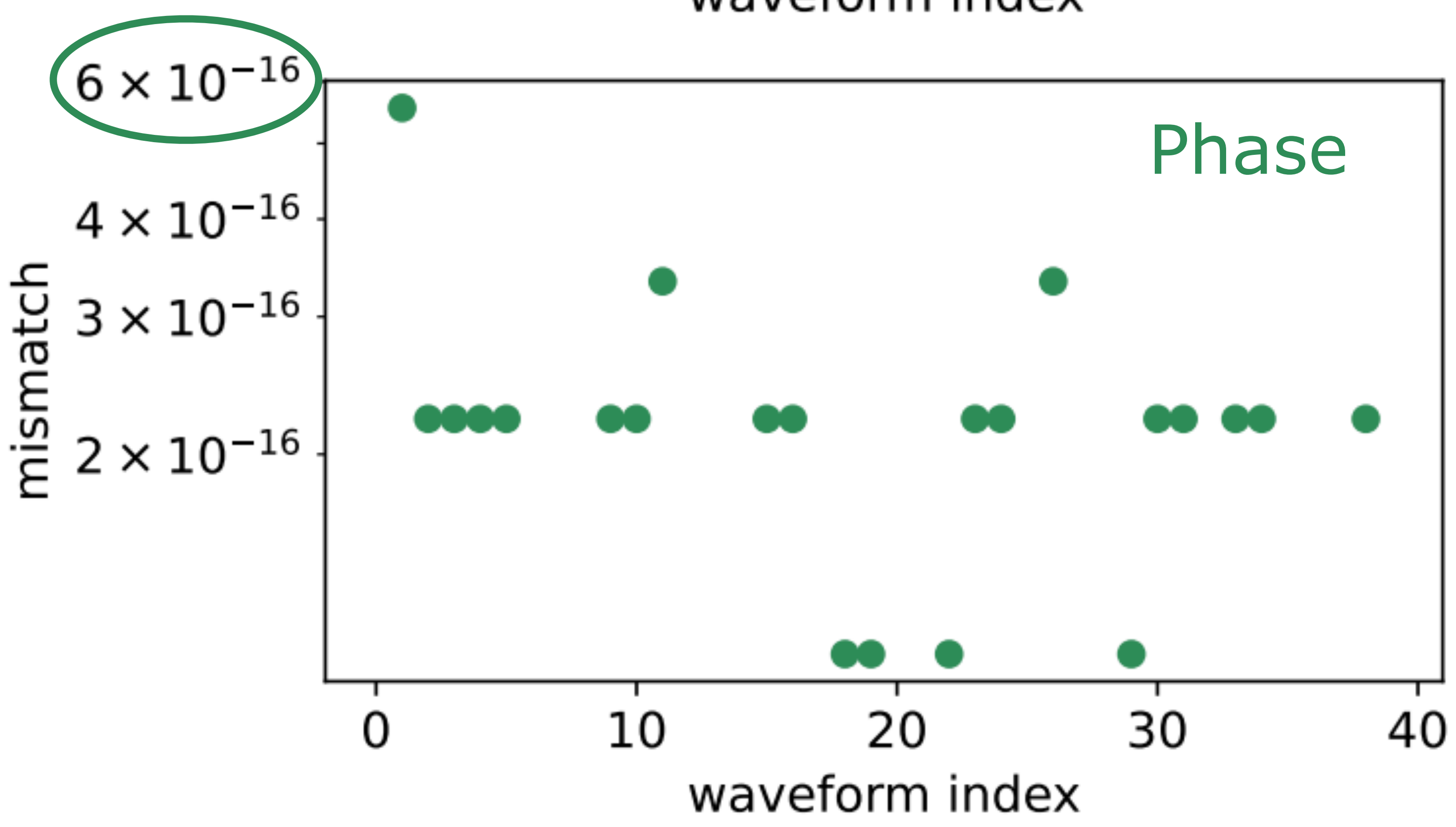
$$\boxed{h(t;\boldsymbol{\lambda}) \approx \sum_{k=1}^{n} \hat{\alpha}_k(\boldsymbol{\lambda}) B_k(t)}$$

- Example reconstruction:









Mismatch for random validation set of N=40 waveforms.

SURROGATE MODEL

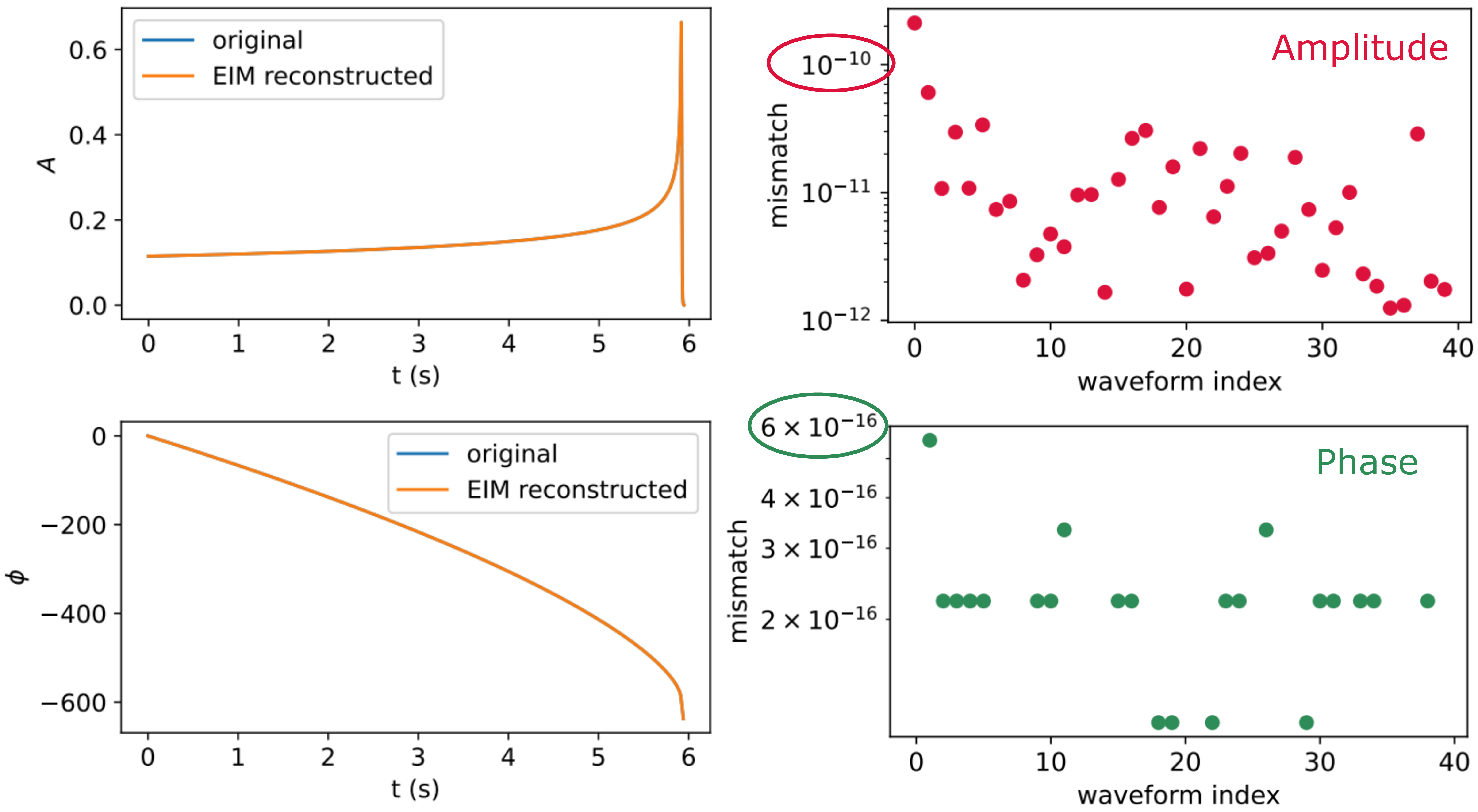
- Interpolating the EIM coefficients $\alpha_k(\boldsymbol{\lambda}_i)$ of the training set for arbitrary values $\boldsymbol{\lambda}$ of the intrinsic parameters, we obtain the interpolated EIM coefficients $\hat{\alpha}_k(\boldsymbol{\lambda})$

Artificial Neural Network

- A surrogate model of the waveform model is then

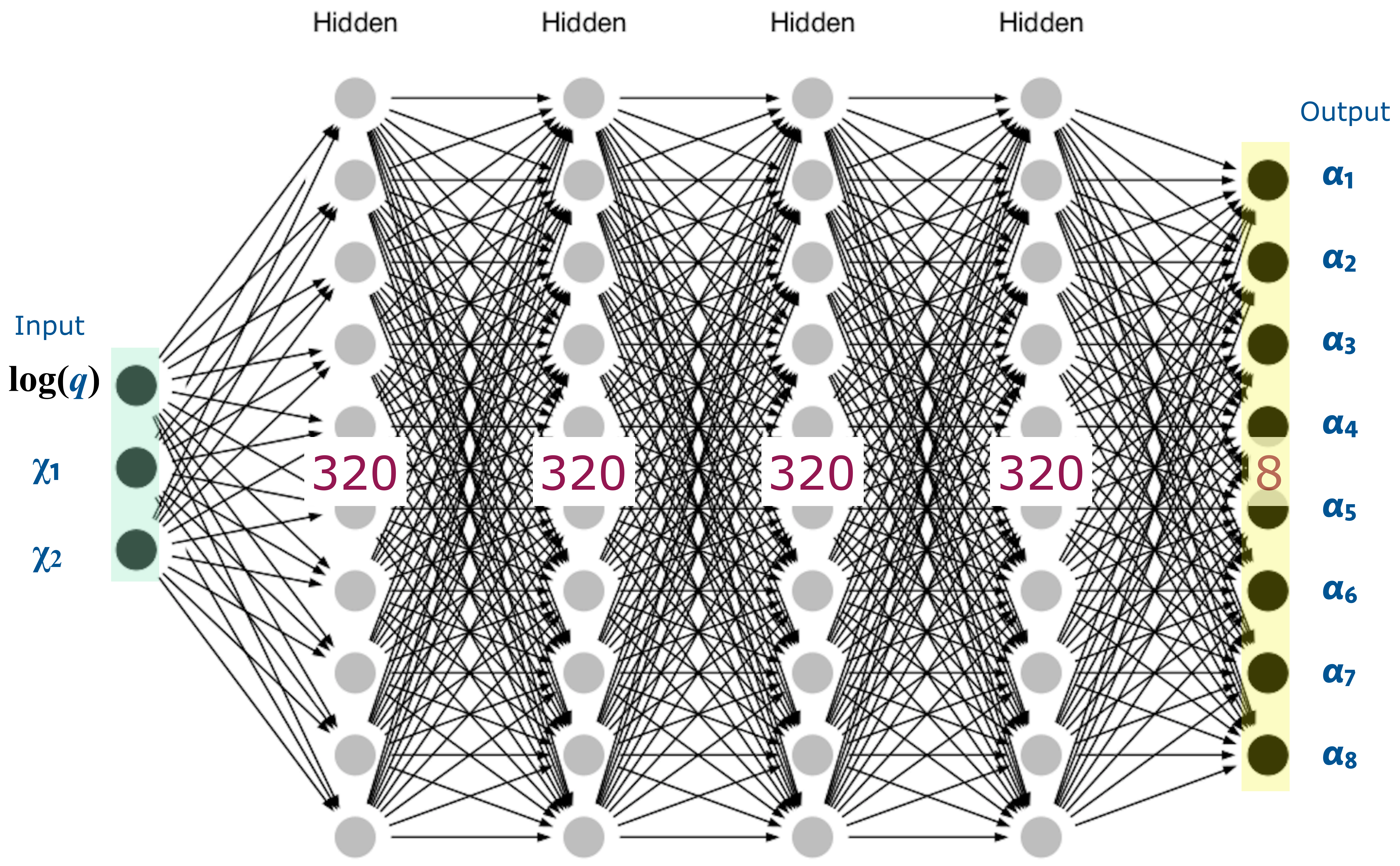
$$h(t;\boldsymbol{\lambda}) \approx \sum_{k=1}^{n} \hat{\alpha}_k(\boldsymbol{\lambda}) B_k(t)$$

- Example reconstruction:



Mismatch for random validation set of N=40 waveforms.

ARTIFICIAL NEURAL NETWORK



ARTIFICIAL NEURAL NETWORK

- Mini-batch size = 1000
- Epochs = 1000
- For amplitude:

  Input: standard scaler
  Output: raw
  Adam optimizer, initial learning rate=10^{-3}
  Activation function = RELU

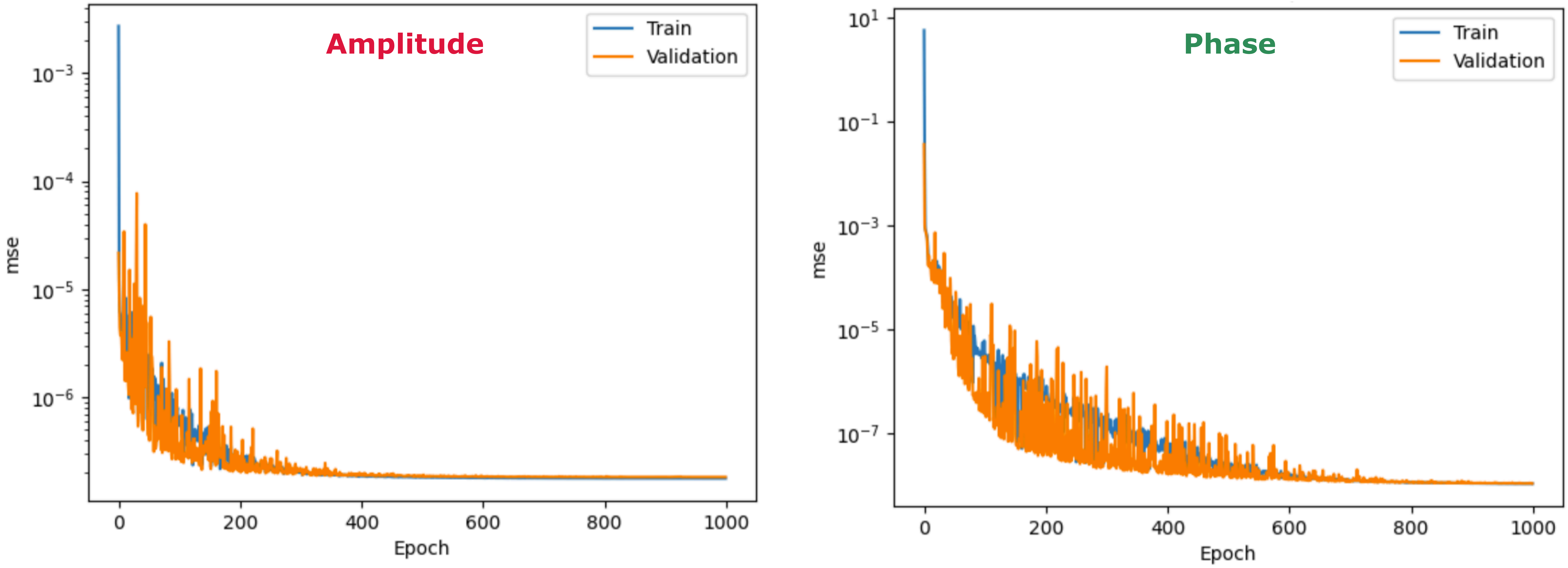
- For Phase:

  Input: standard scaler
  Output: minmax scaler
  Adamax optimizer, initial learning rate = 10^{-2}
  Activation function = softplus

Surrogate with ANN **~5000 faster** than original SEOBNRv4 waveform generation.

EOBNRv2 WAVEFORM MODEL

- Training: minimization of Mean Square Error $$MSE = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i\|_2^2$$
- Validation set: 3×10^4 waveforms used to confirm training of model



RESIDUAL NETWORK

- Define the residual (for each waveform) of the first network as

$$\boldsymbol{e}_i \equiv \boldsymbol{y}(\boldsymbol{\lambda}_i) - \hat{\boldsymbol{y}}(\boldsymbol{\lambda}_i)$$

- A second network, with *same architecture* as the first one, is trained on these residuals.

  The output scaling was MinMaxScaler

- The MSE is

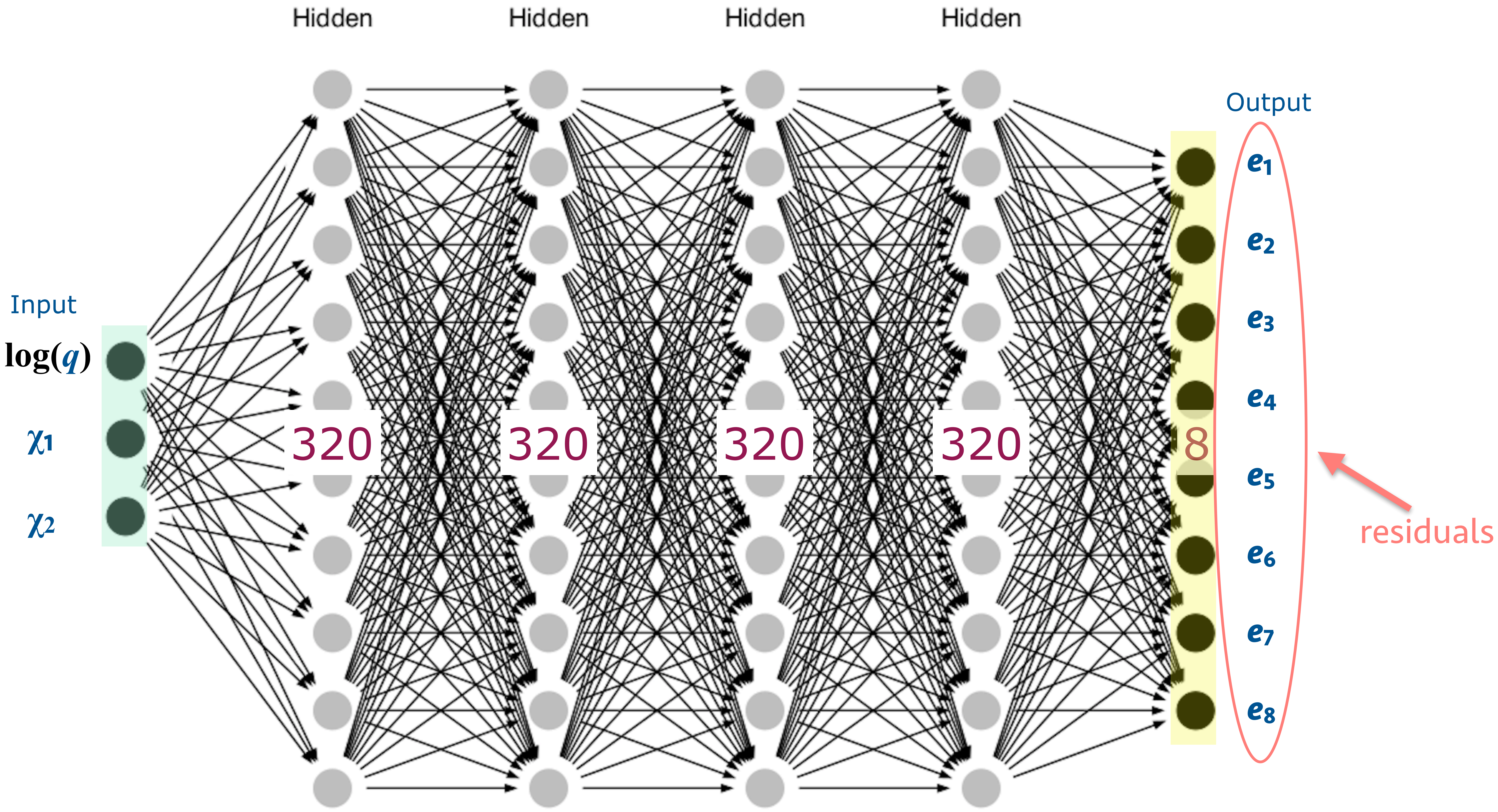
$$MSE = \frac{1}{N}\sum_{i=1}^{N} \|\hat{e}_i - \boldsymbol{e}_i\|_2^2$$

- The prediction $\hat{\boldsymbol{e}}$ of the second network is then added to the prediction $\hat{\boldsymbol{y}}$ of the first network

- Improved prediction:

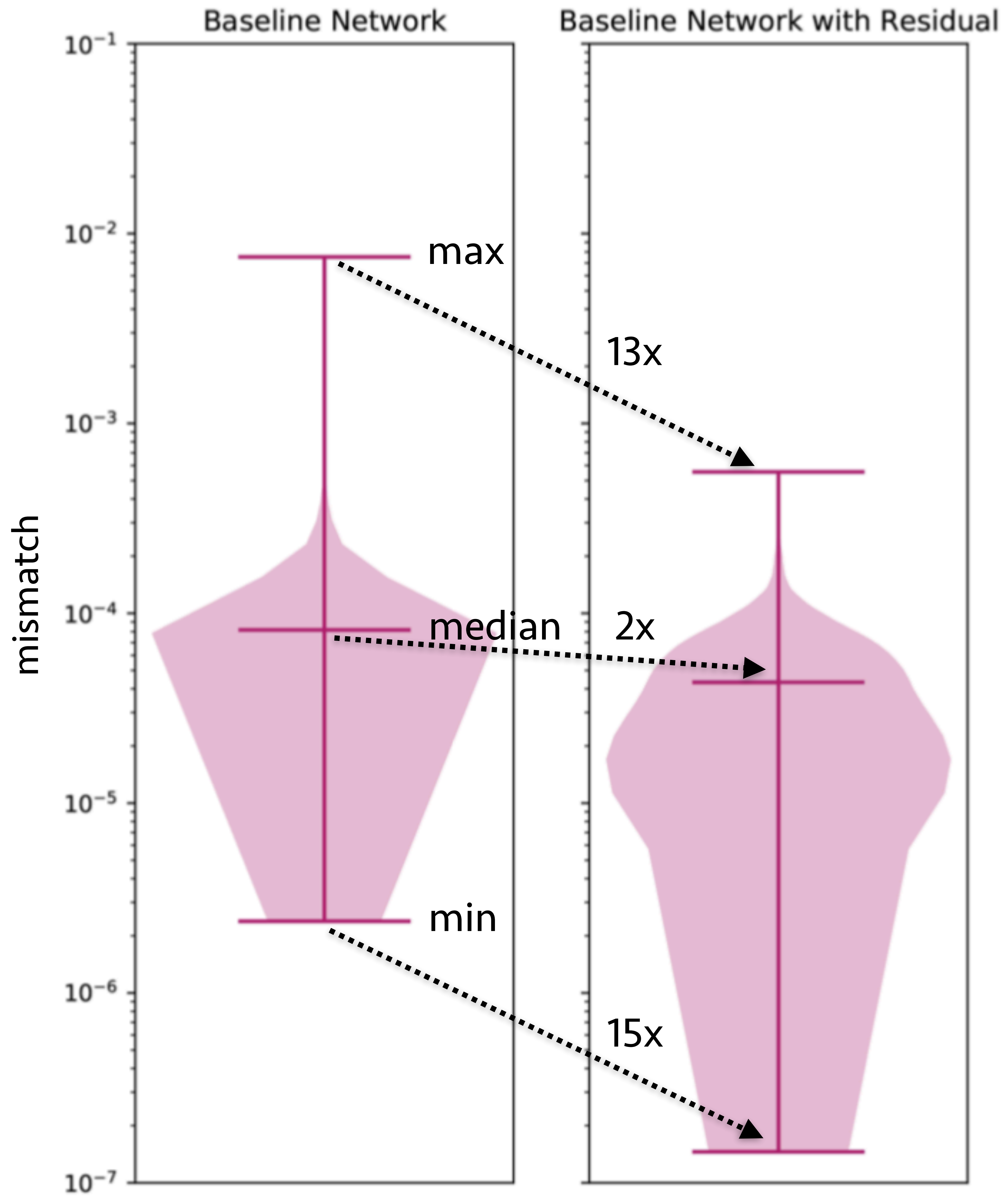
$$\boxed{\tilde{\boldsymbol{y}} \equiv \hat{\boldsymbol{y}} + \hat{\boldsymbol{e}}}$$

ARTIFICIAL NEURAL NETWORK



IMPACT OF RESIDUAL NETWORK

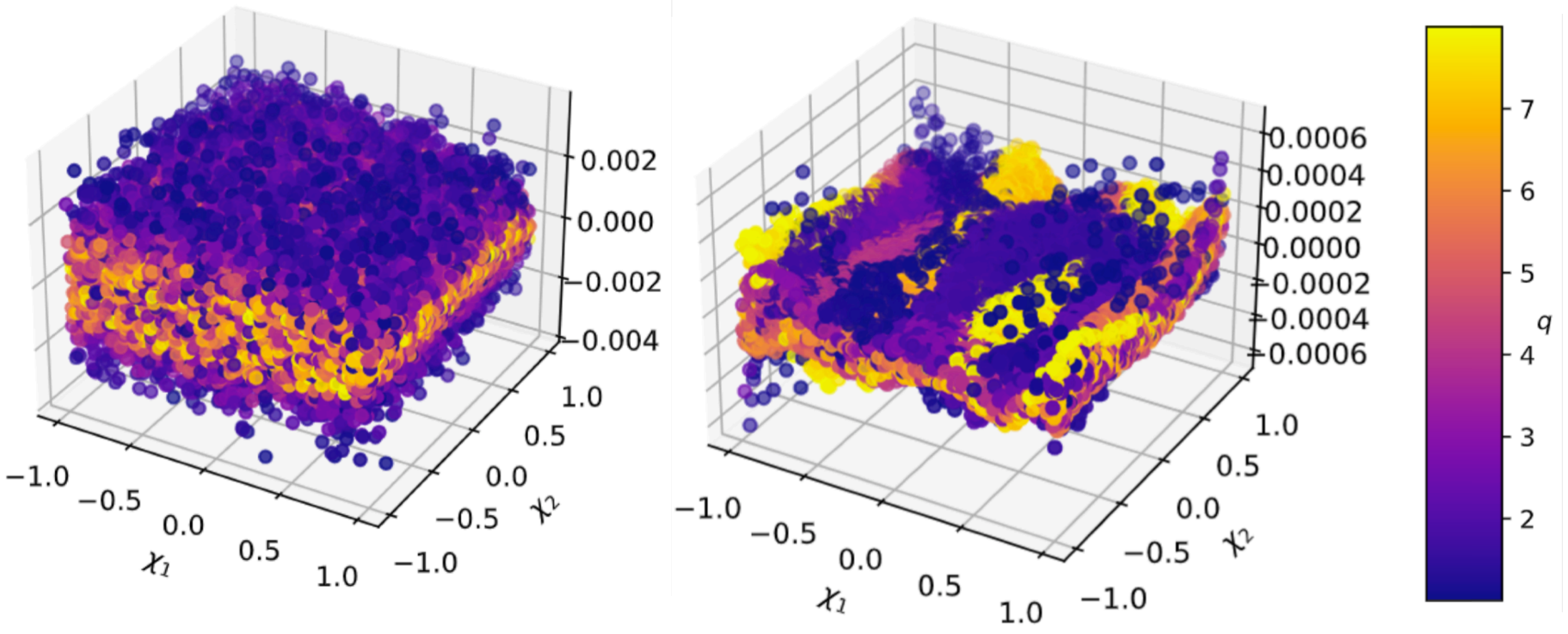
- Adding the second network trained on the residuals of the first, improves the maximum, median and minimum mismatches by factors of 13, 2 and 15, correspondingly.

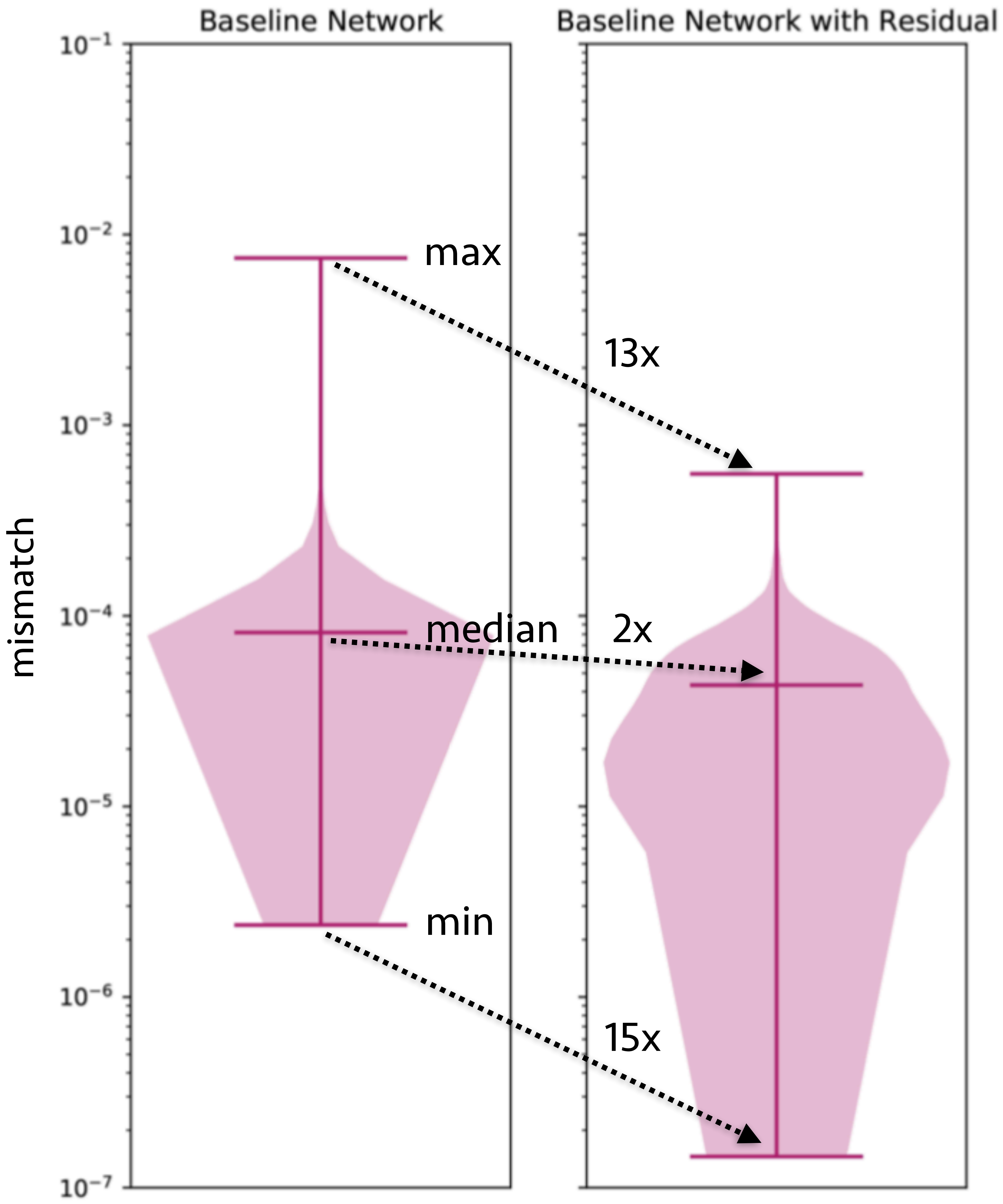


IMPACT OF RESIDUAL NETWORK

- Adding the second network trained on the residuals of the first, improves the maximum, median and minimum mismatches by factors of 13, 2 and 15, correspondingly.

- Explanation: residual errors of some coefficients $\hat{\alpha}_k$ obtained by the first network have structure. Examples:

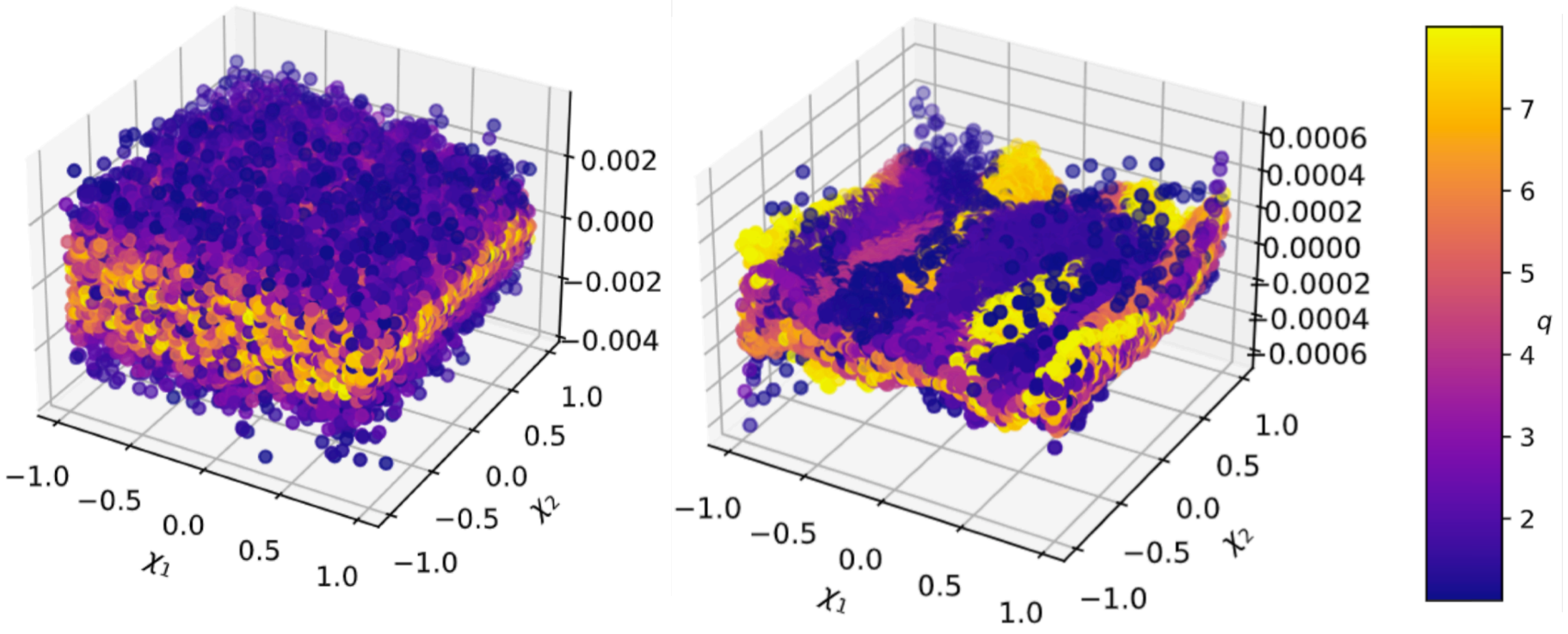




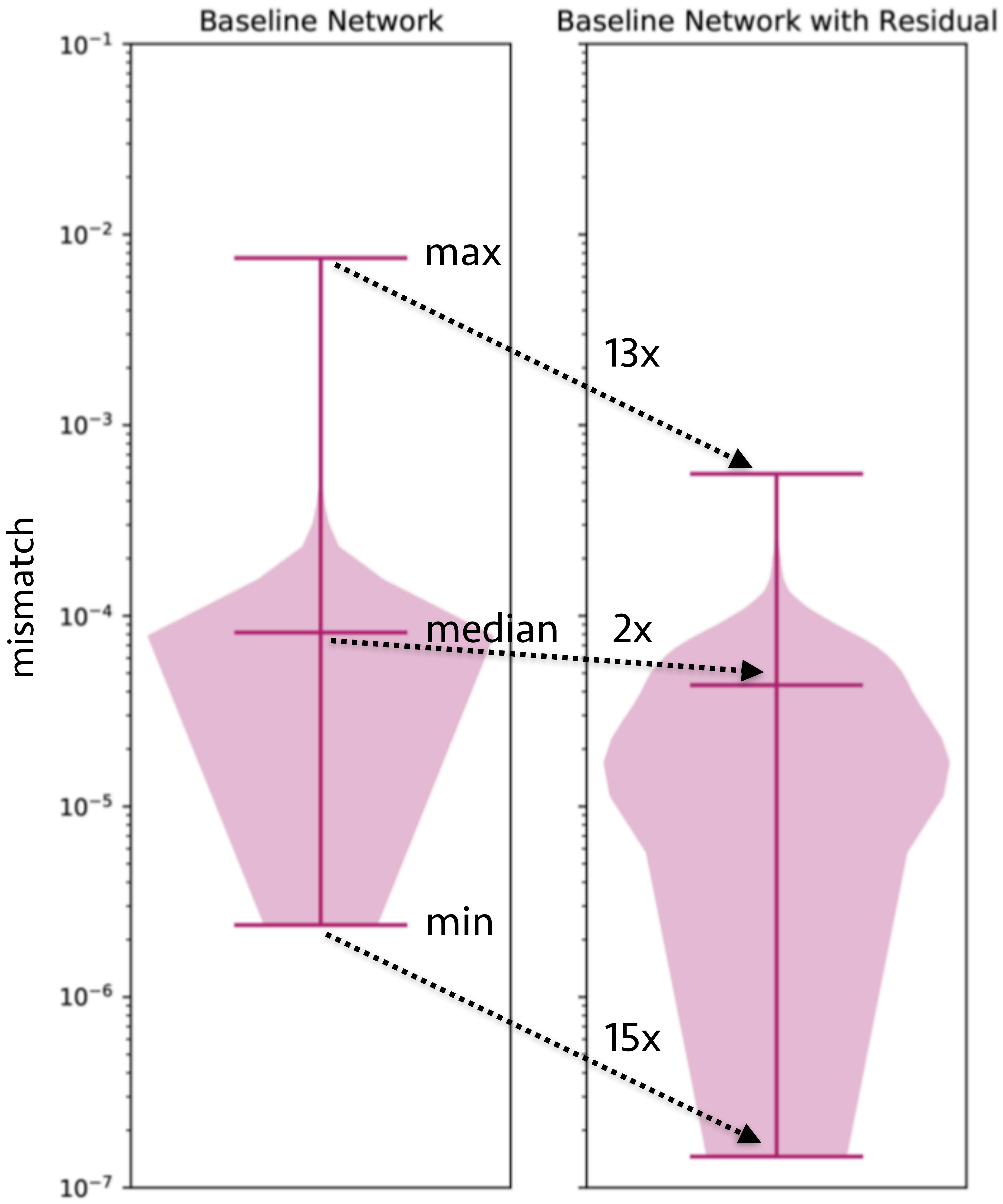
IMPROVEMENTS THROUGH RESIDUAL NETWORK

- Adding the second network trained on the residuals of the first, improves the maximum, median and minimum mismatches by factors of 13, 2 and 15, correspondingly.

- Explanation: residual errors of some coefficients $\hat{\alpha}_k$ obtained by the first network have structure. Examples:



- **Outlook**: generalize to precessing BBH and BNS waveform models.



- Nonspinning case

- Construct a surrogate model for the two polarizations $h_+, h_\times$ of nonspinning EOBNRv2 waveforms.

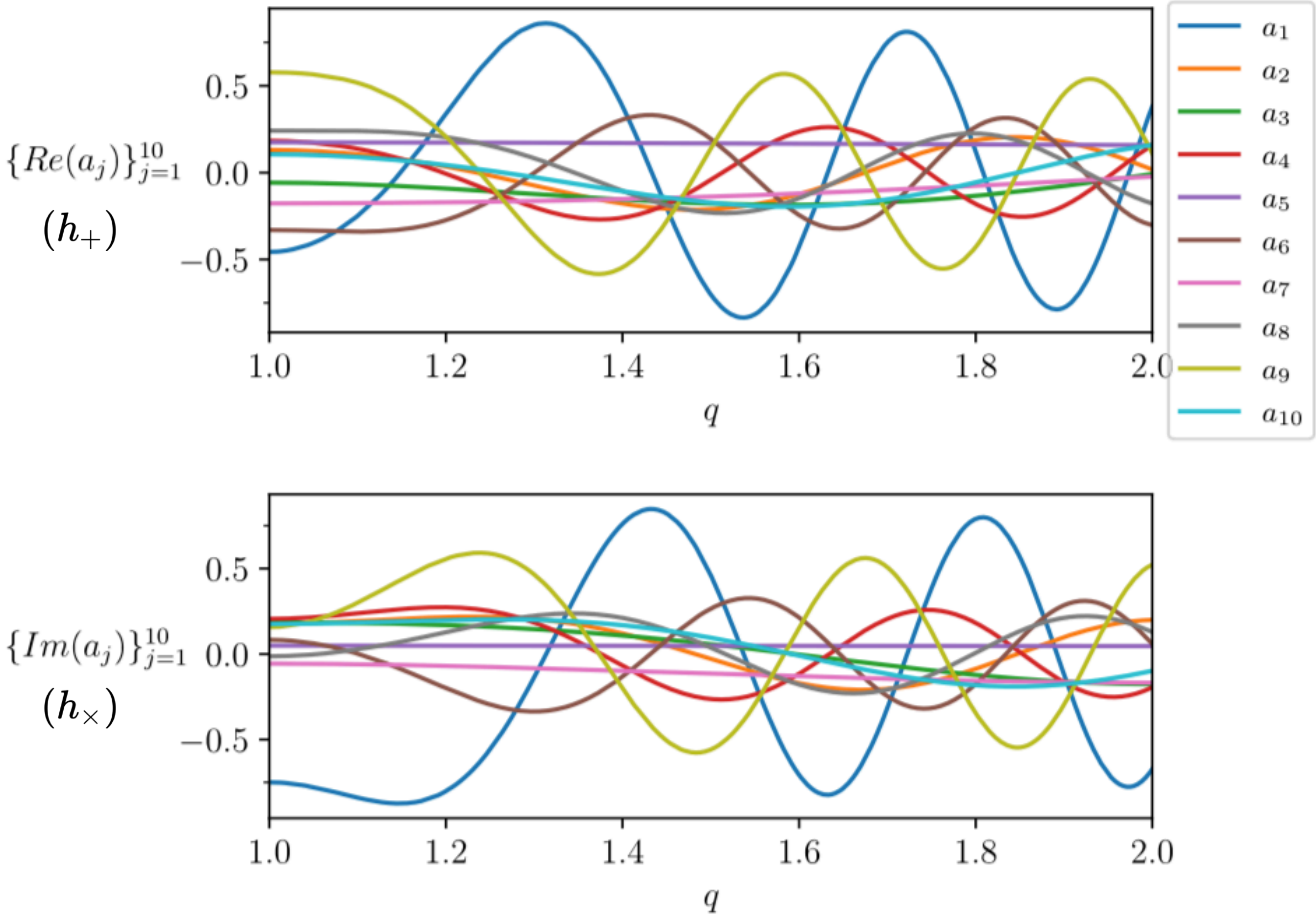
$$h(t;q) = h_+(t;q) - i h_\times(t;q)$$

- Initial range of mass ratio $0 \leq q \leq 2$

- N = 1000 waveforms

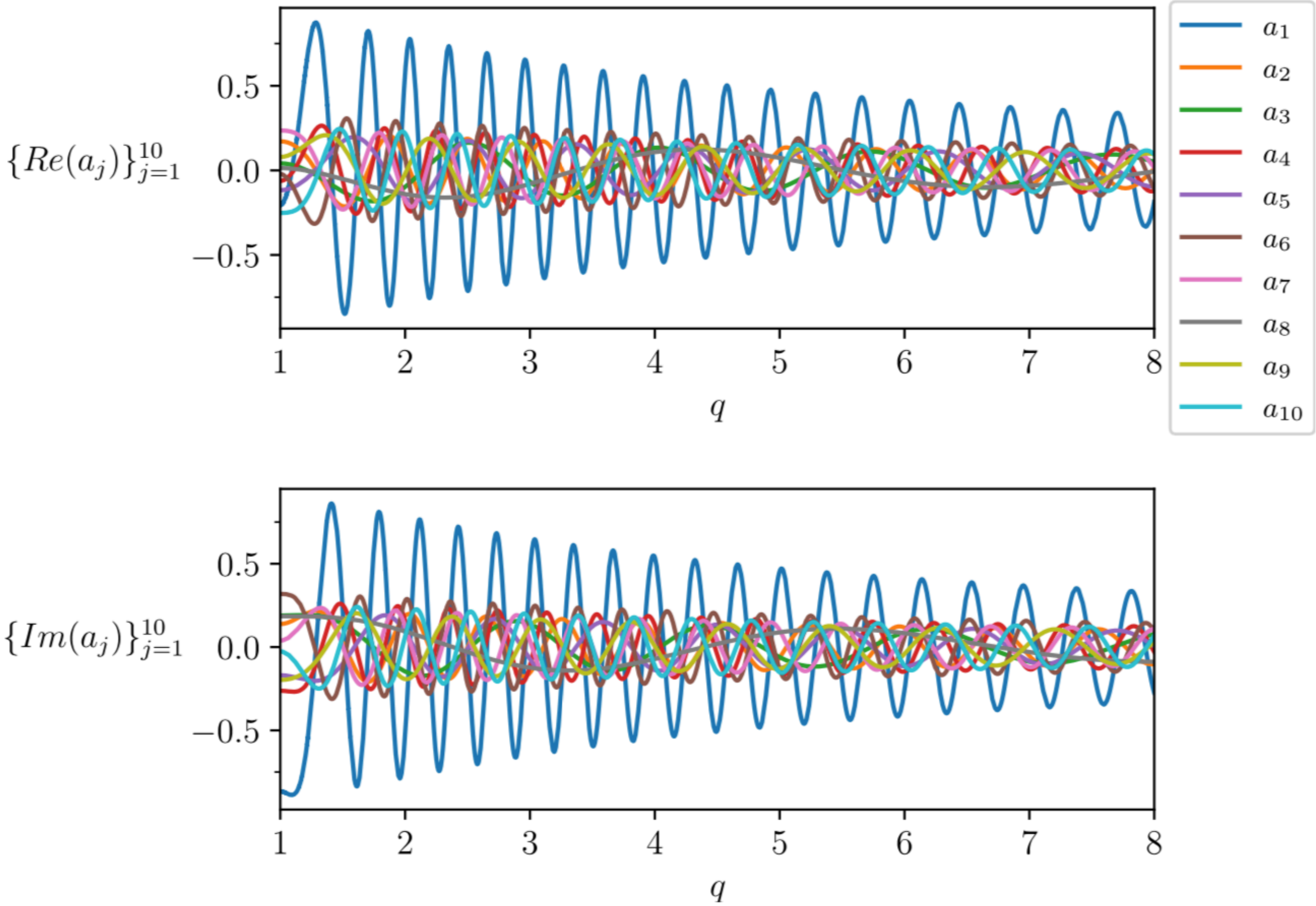
- With tolerance = 10^{-10} the reduced basis has a size of n = 11.

- The first ten coefficients a_j are shown as a function of q.



EXTENSION TO LARGER MASS RATIOS

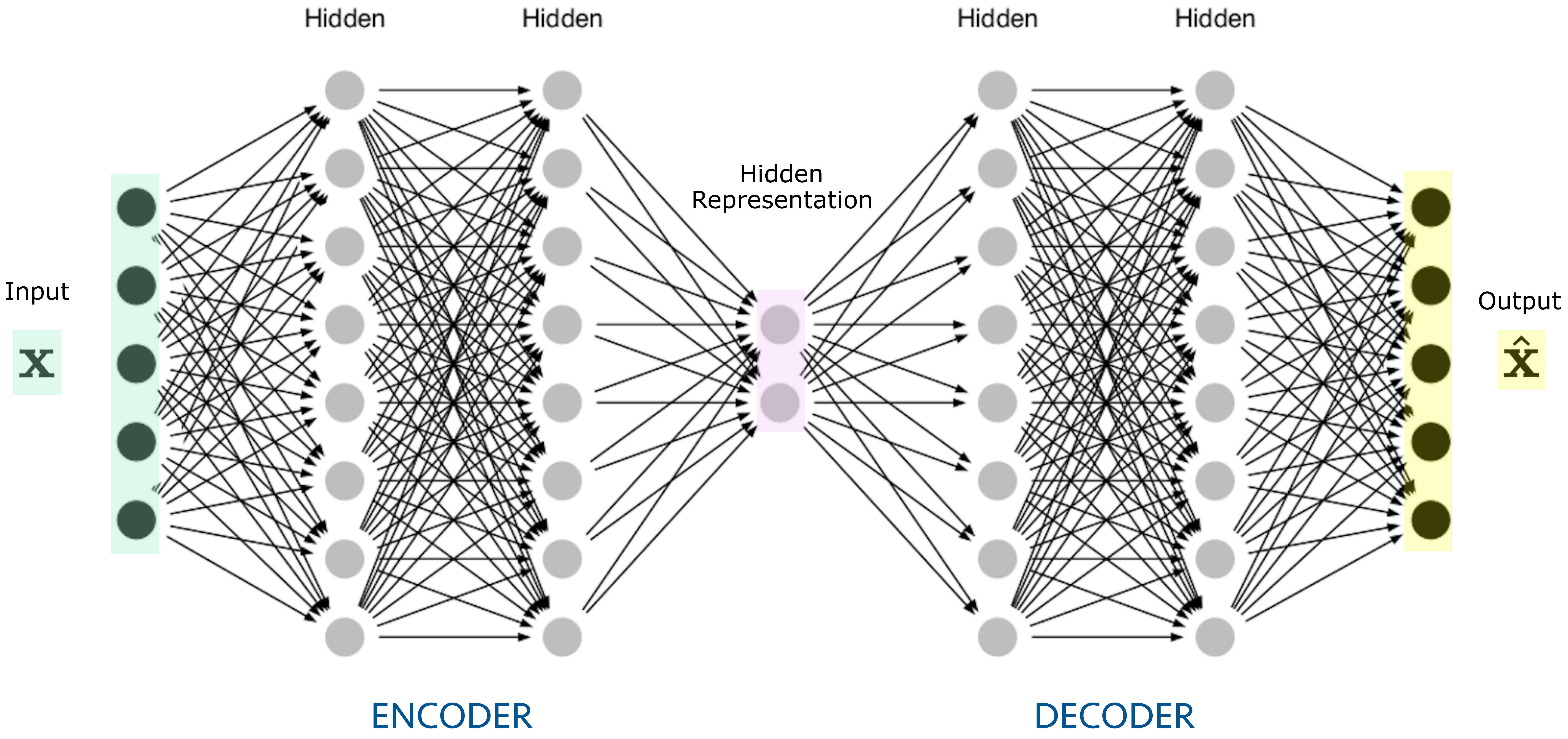
Reconstructed EIM coefficients for $0 \leq q \leq 8$



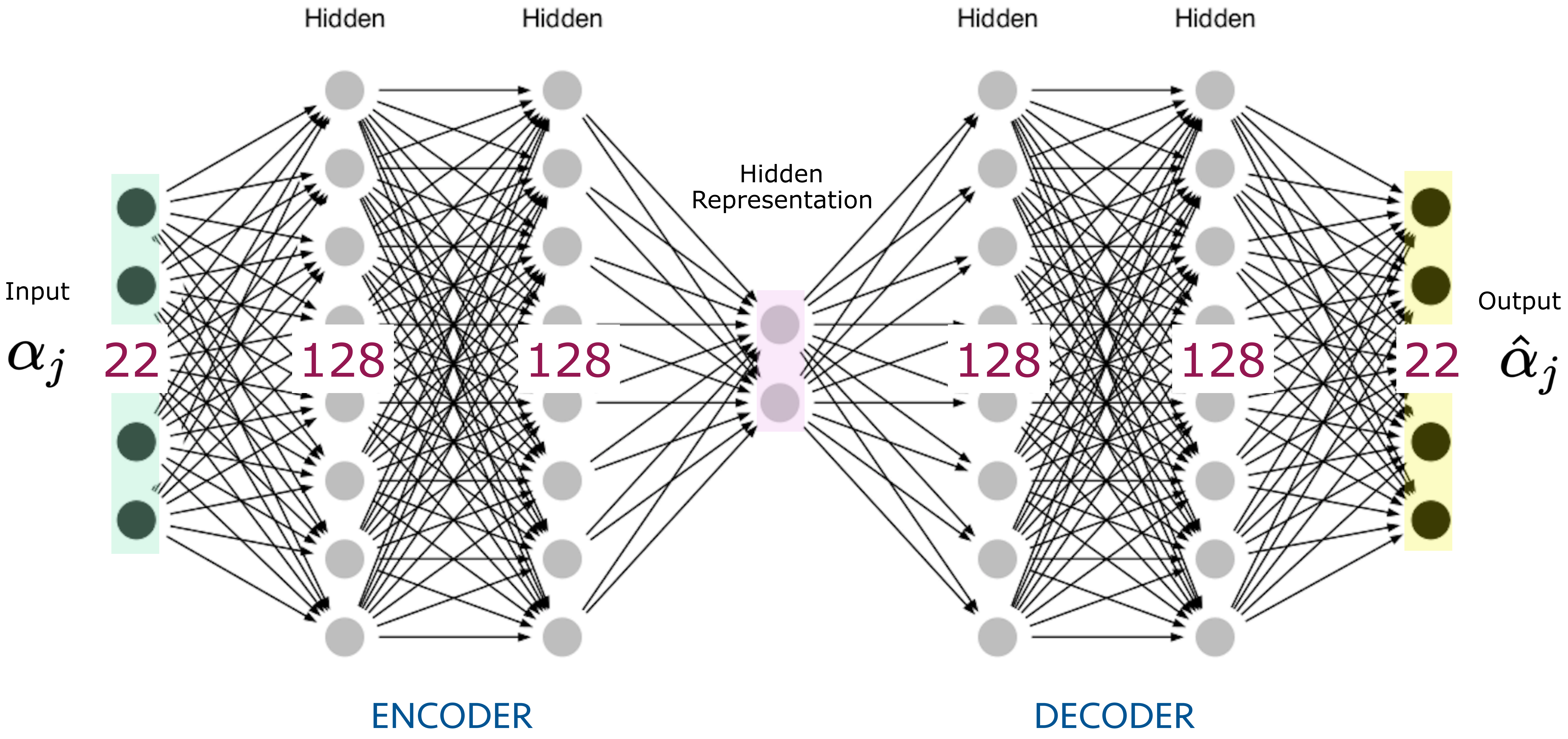
AUTOENCODER

Unsupervised - encodes the input $\mathbf{x}$ into a lower-dimensional representation $g(\mathbf{x})$ and then decodes it at the output as $\hat{\mathbf{x}} = f(g(\mathbf{x}))$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2$$

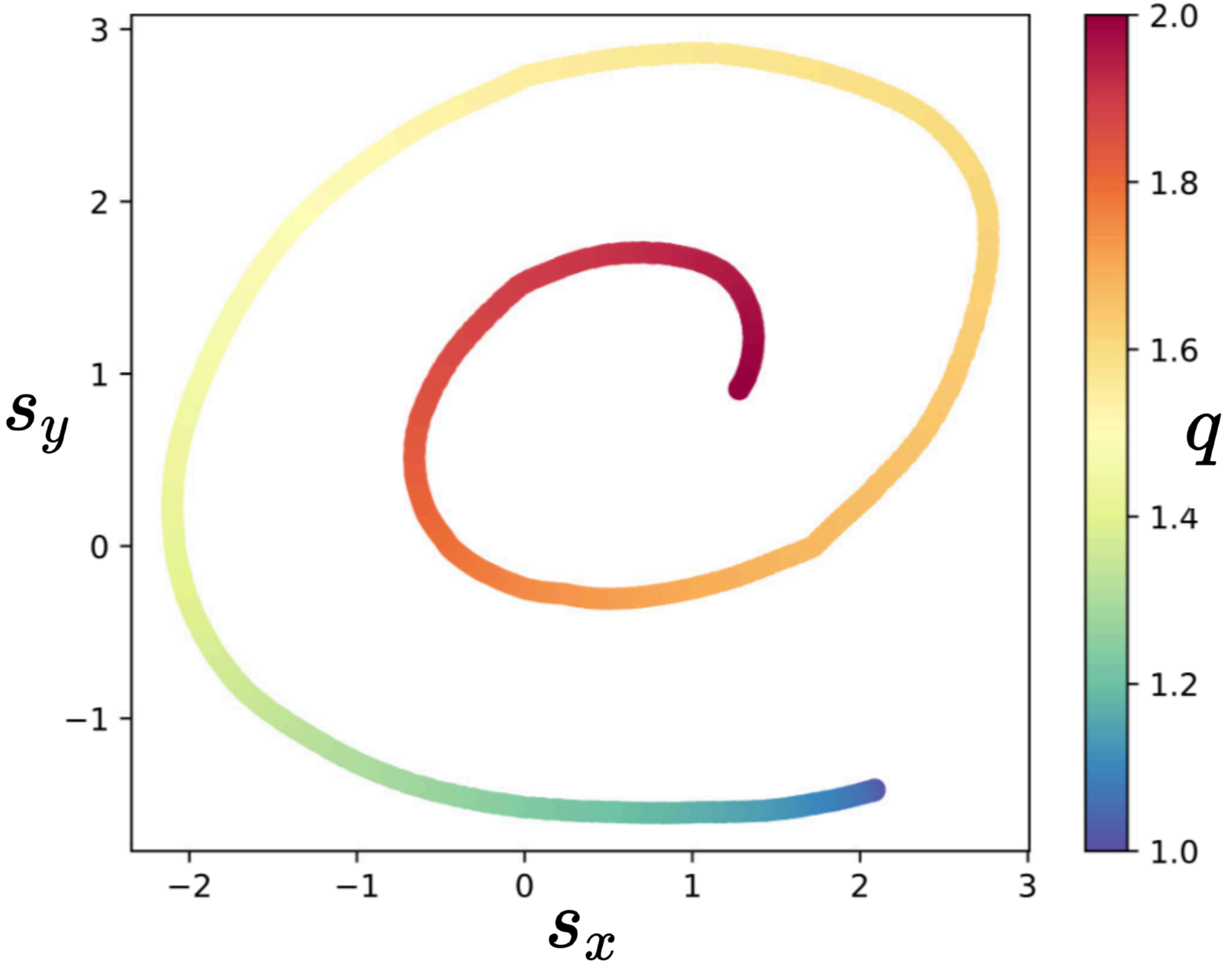


Training: 100 epochs. Batch size: 32. Initial learning rate = 1e-3. Learning schedule x0.9 every 15 epochs. Activation function PreLU (parametrized RELU), Gradient descent. Achieved MSE = 6.8e-5.



SPIRAL STRUCTURE OF HIDDEN REPRESENTATION

Values s_x, s_y of hidden representation form a spiral structure!



We use the knowledge of q from the training set, to write down a simple relation:

$$s_x = (\alpha + \beta \cdot \theta) \cdot \cos\theta$$
$$s_y = (\alpha + \beta \cdot \theta) \cdot \sin\theta$$

where

$$\theta = w \cdot q + b$$

depends on the mass ratio q.

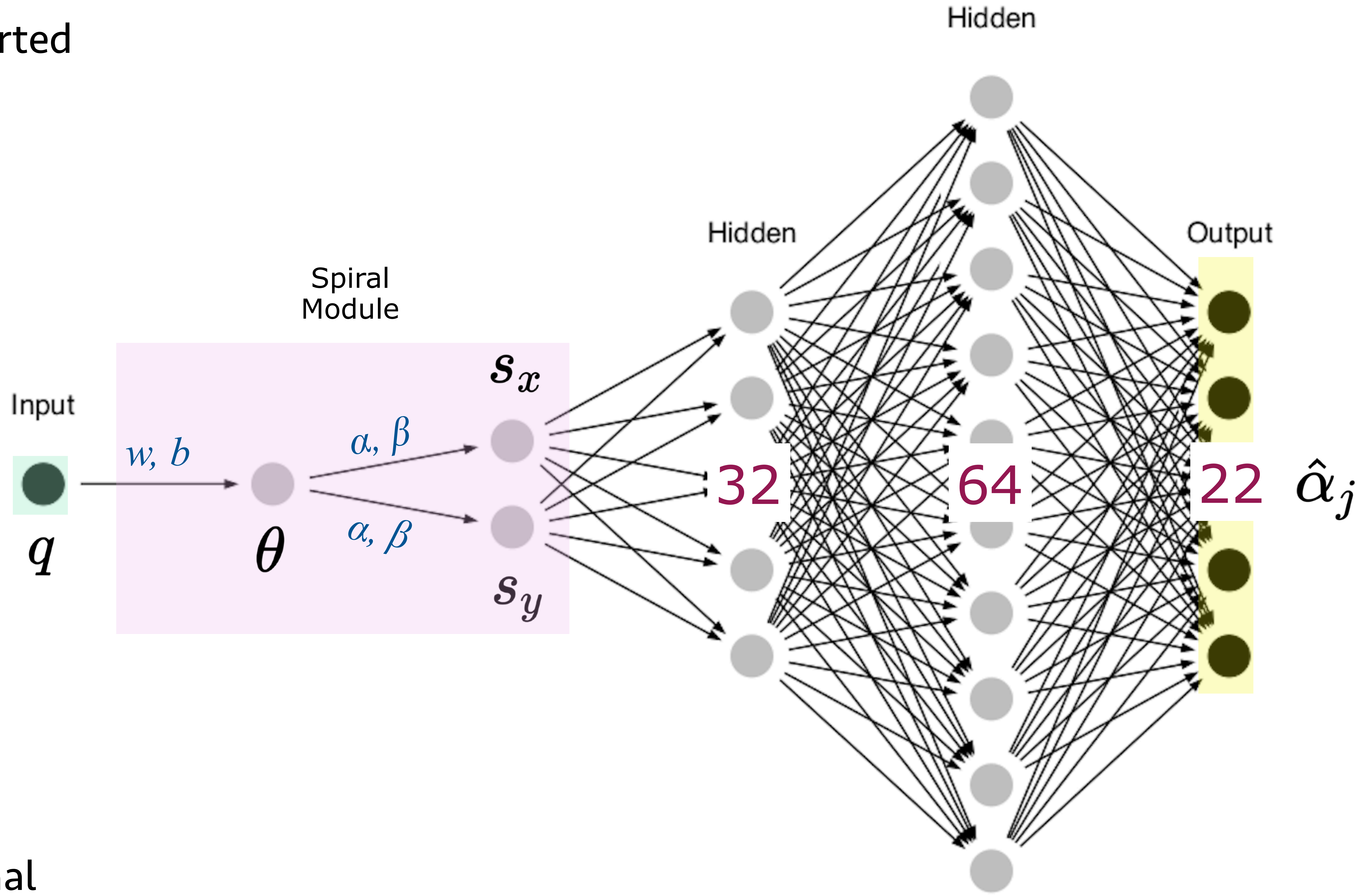
The coefficients α, β, w, b of the spiral are learnable.

We thus insert a spiral module that maps

$$\boxed{q \to \theta \to s_x, s_y}$$

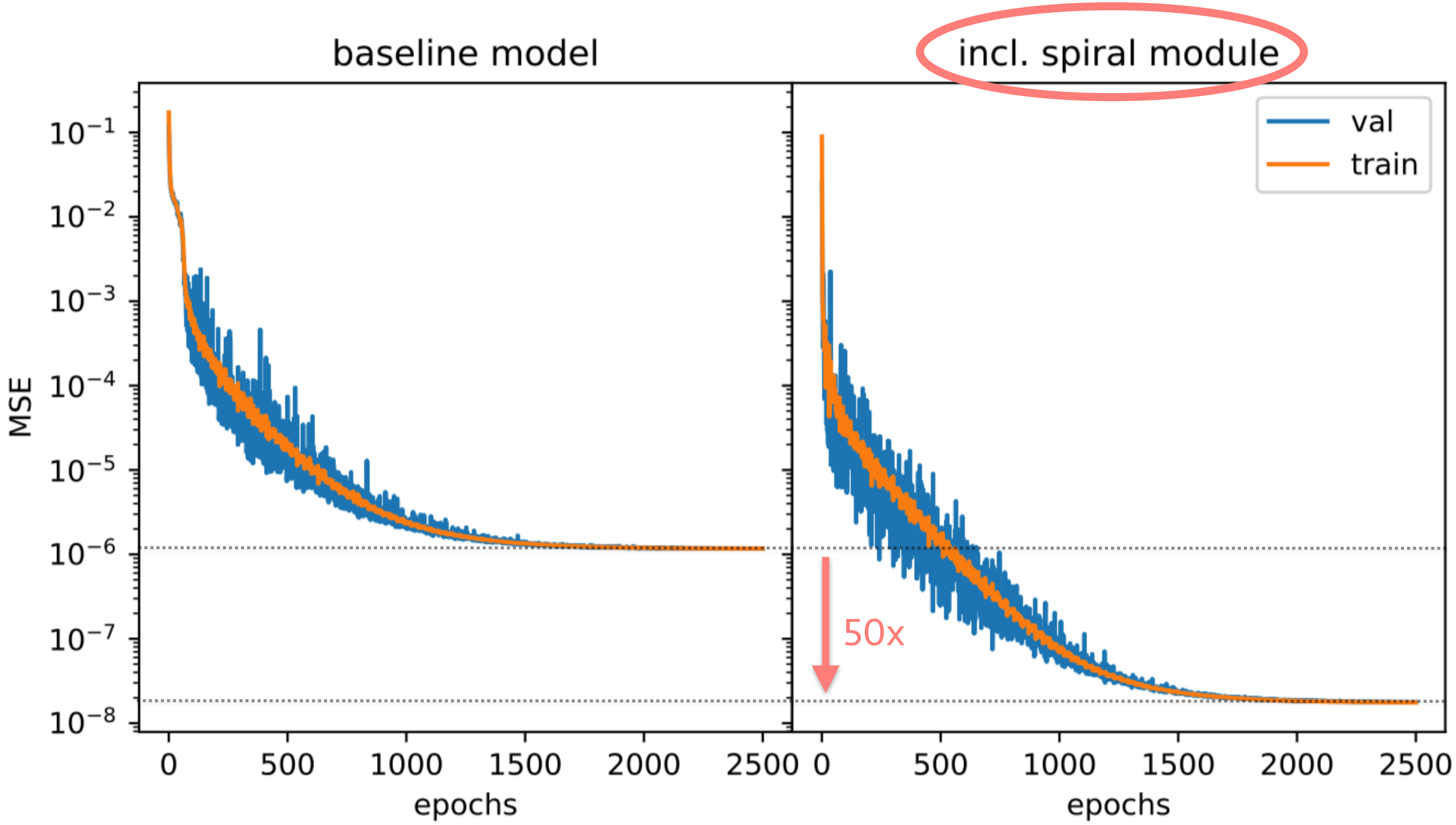
NEURAL NETWORK WTH SPIRAL MODULE

- Fully connected ANN with inserted spiral module.
- Training set: N = 10,000
  Validation set: N = 2,000
- Training for 2500 epochs
- Batch size = 16
- Adam optimizer
- Initial learning rate = 10^{-3}
- Learning schedule:
  x0.9 every 150 epochs
- 400 times faster than traditional spline interpolation



LOSS PER EPOCH

Training and validation loss per epoch for a 32–64-128–64 network:



COMPARISON OF ERRORS WITH AND WITHOUT SPIRAL MODULE

Same MSE is achieved with simpler networks -> faster execution time

network	max $\mathscr{M}$	median $\mathscr{M}$	$p = 95$ $\mathscr{M}$	network simplicity (max batch size)
16–64	4.36×10^{-1}	1.44×10^{-1}	3.47×10^{-1}	8 m
$\mathscr{S}$-16–64	1.33×10^{-3}	9.14×10^{-5}	2.25×10^{-4}	8 m
32–64	4.43×10^{-1}	5.85×10^{-3}	2.84×10^{-1}	7.3 m
$\mathscr{S}$-32–64	4.39×10^{-3}	1.12×10^{-5}	2.56×10^{-5}	7.3 m
32-32–64	1.48×10^{-3}	4.97×10^{-5}	1.87×10^{-3}	6.1 m
$\mathscr{S}$-32-32–64	2.99×10^{-5}	9.04×10^{-7}	1.99×10^{-6}	6.1 m
32–64-128	9.34×10^{-5}	8.00×10^{-6}	5.83×10^{-5}	4.2 m
$\mathscr{S}$-32–64-128	1.80×10^{-6}	1.66×10^{-7}	3.63×10^{-7}	4.2 m
32–64-128–64	7.79×10^{-5}	1.13×10^{-6}	4.62×10^{-6}	3.4 m
$\mathscr{S}$-32–64-128–64	2.55×10^{-6}	4.16×10^{-8}	1.07×10^{-7}	3.4 m
64–128-256–128	2.46×10^{-5}	1.50×10^{-7}	2.23×10^{-6}	1.9 m
$\mathscr{S}$-64–128-256–128	5.25×10^{-7}	1.69×10^{-8}	3.93×10^{-8}	1.9 m

COMPUTATIONAL GAINS

2x the batch size in a single forward pass when the spiral module is included (for same MSE)!