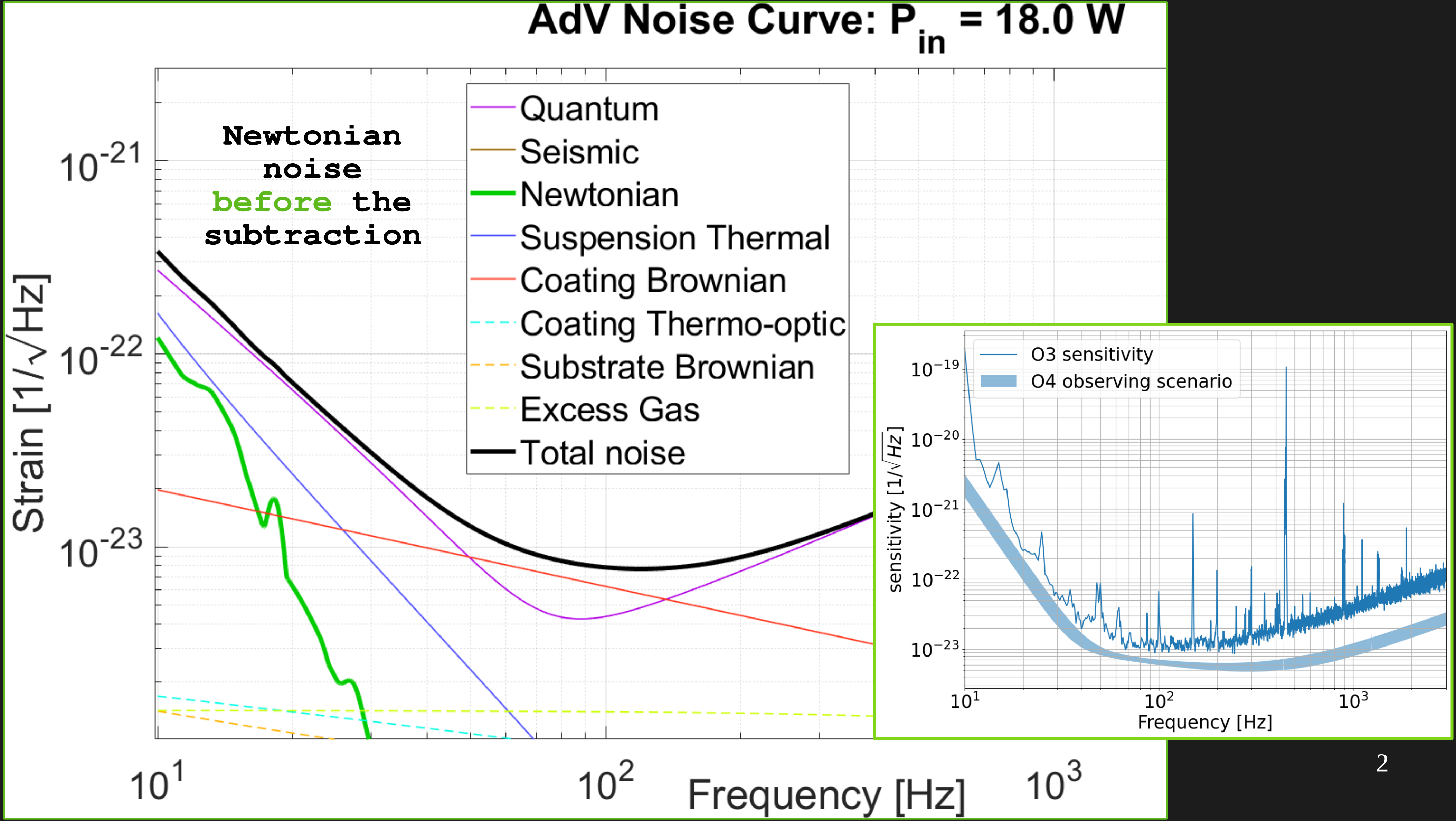
Bayesian ML algorithm for array optimization

ET-Site Studies and Characterization, November 2021

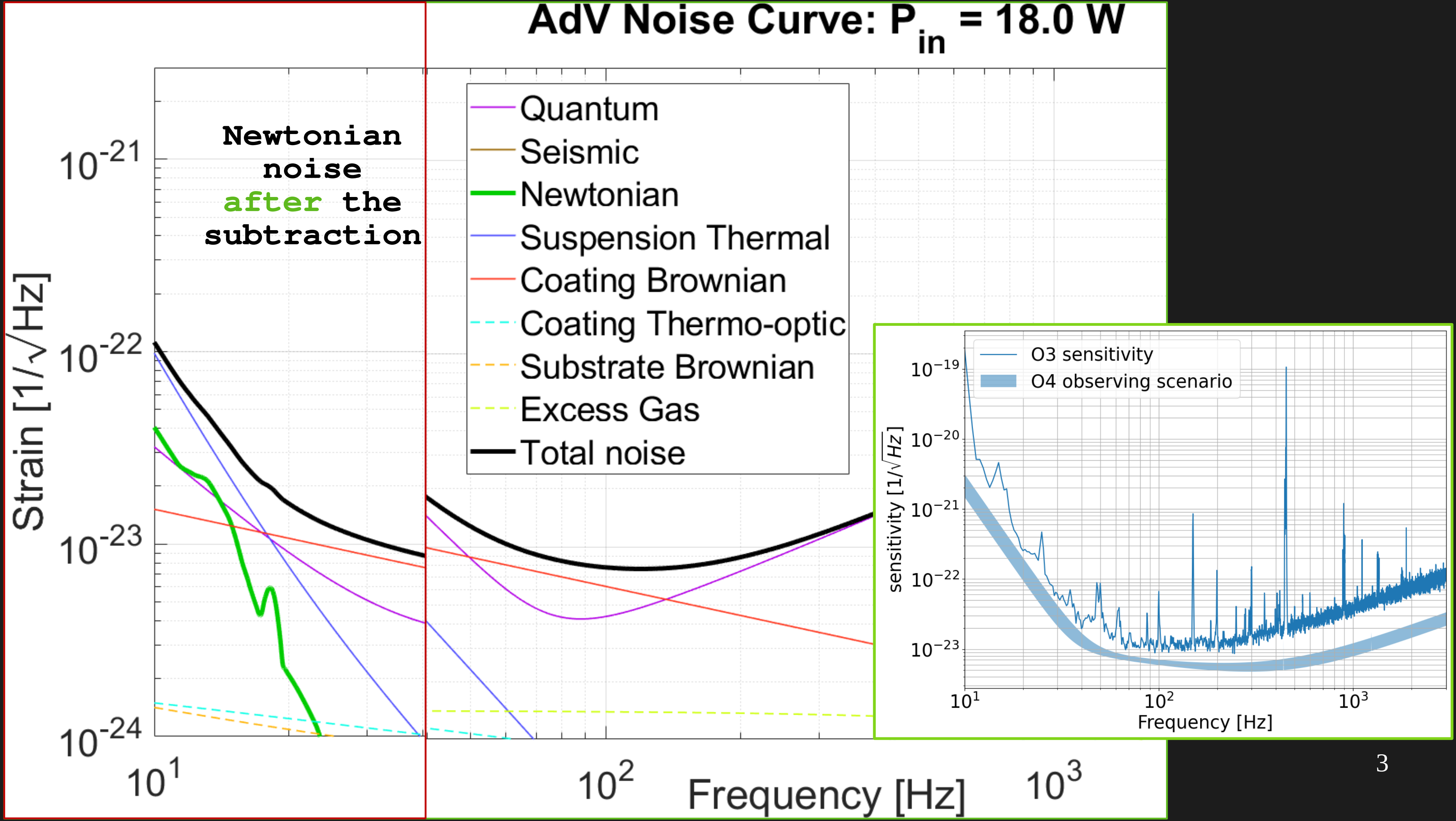
Authors: **Francesca Badaracco**
Jan Harms

UCLouvain

GSSI



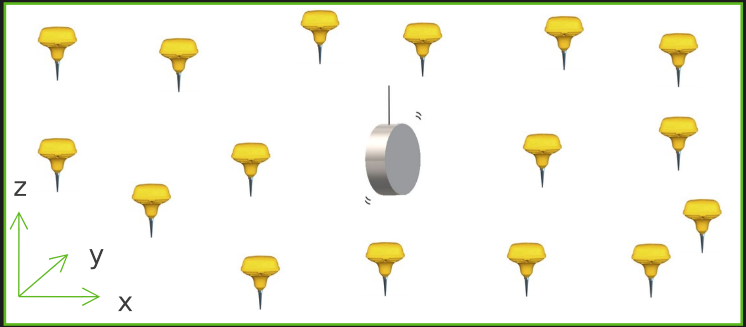
AdV Noise Curve: P in = 18.0 W
Newtonian noise before the subtraction
Quantum
Seismic
Newtonian
Suspension Thermal
Coating Brownian
Coating Thermo-optic
Substrate Brownian
Excess Gas
Total noise
Strain [1/√Hz]
10-21
10-22
10-23
10 1
10 2
10 3
Frequency [Hz]
O3 sensitivity
O4 observing scenario
sensitivity [1/√Hz]
10−19
10−20
10−21
10−22
10−23
10 1
10 2
10 3
Frequency [Hz]


AdV Noise Curve: P
in
= 18.0 W
Newtonian
noise
after the
subtraction
Quantum
Seismic
Newtonian
Suspension Thermal
Coating Brownian
Coating Thermo-optic
Substrate Brownian
Excess Gas
Total noise
Strain [1/√Hz]
10-21
10-22
10-23
10-24
10 1
10 2
10 3
Frequency [Hz]
O3 sensitivity
O4 observing scenario
sensitivity [1/√Hz]
10−19
10−20
10−21
10−22
10−23
10 1
10 2
10 3
Frequency [Hz]

Active noise cancellation

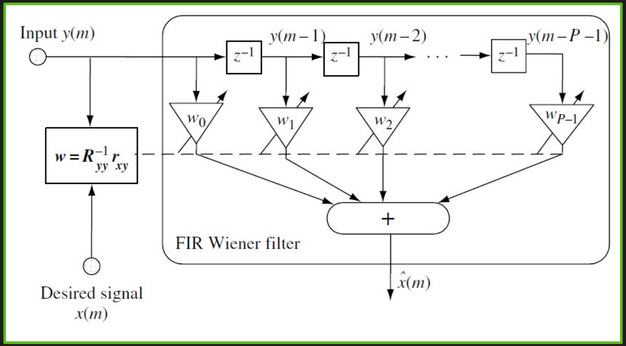
- NN: it cannot be physically **shielded**
- We can perform an **active** noise cancellation
- Linear filter: **Wiener filter** (optimal filter)

Sensor array



+

Wiener Filter

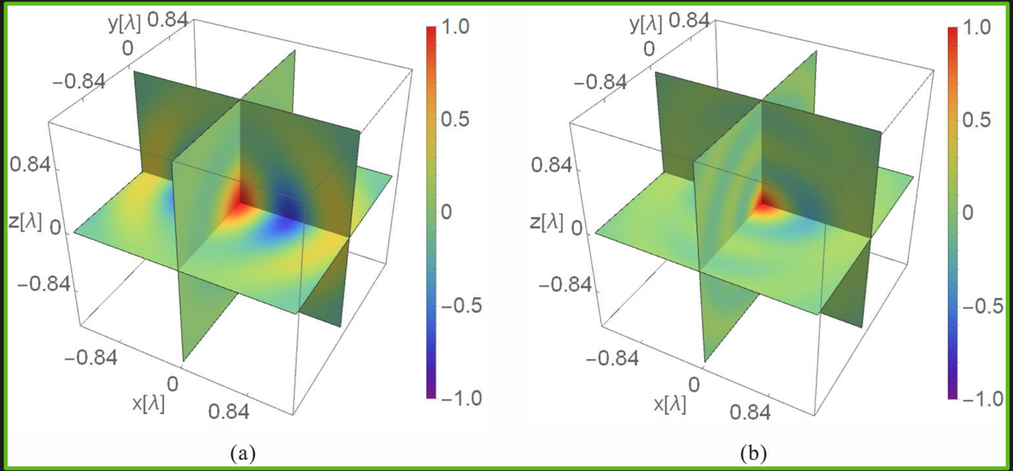


=

Newtonian **N**oise (NN) **cancellation**

Limited by P and S waves mixing:

Only P waves   Mixed: P and S



Correlation of the seismometer in the origin with all the other points in a homogeneous and isotropic field.

Remember:
P = compressional waves (always generate NN)
S = shear waves (**usually** don't generate NN)

Because of their **different propagation velocity** in the ground, P and S waves produce two-point correlations that are out of phase, thus affecting the configuration of the optimal array.

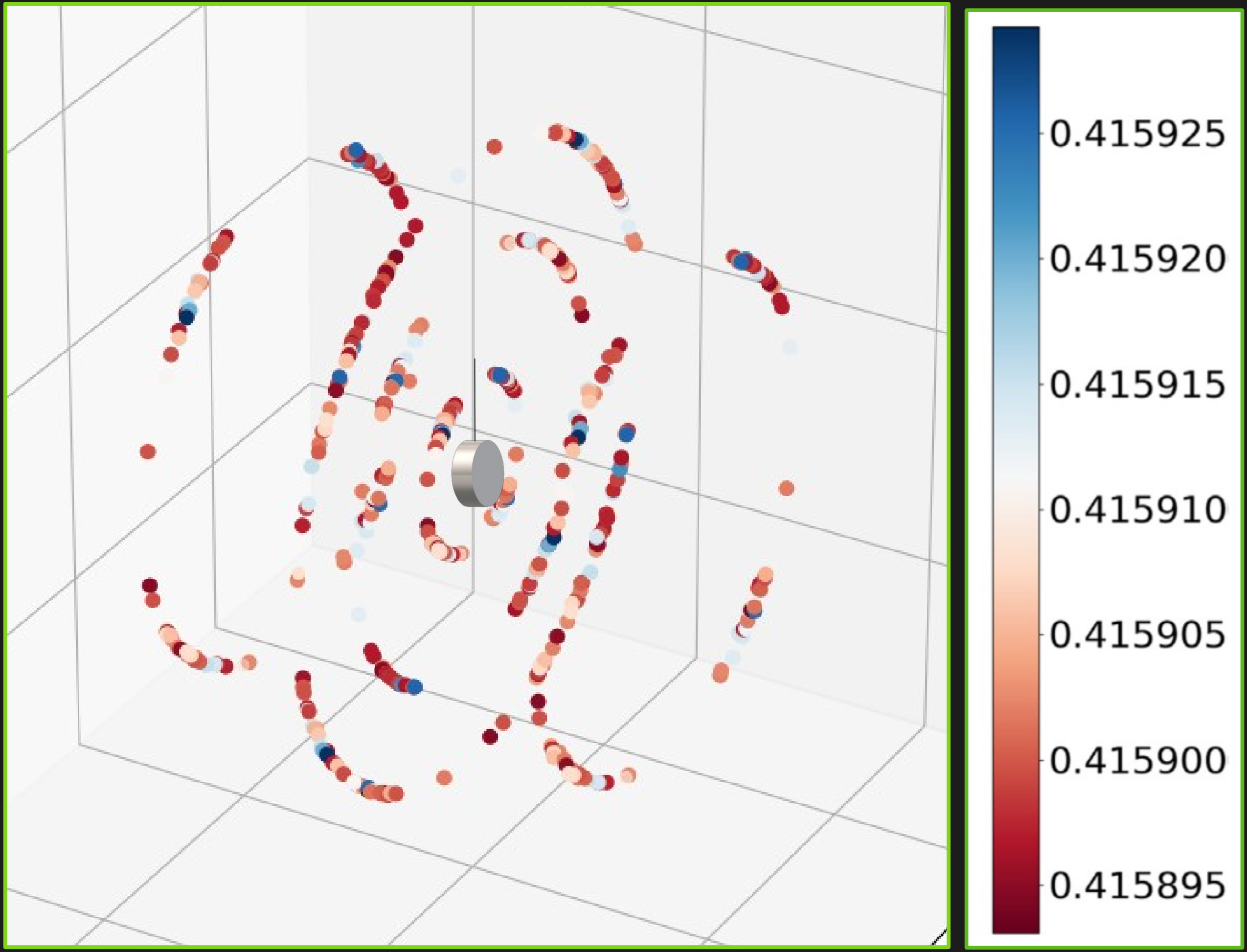
Let's go back to the optimization for the Newtonian noise:

$$R(\omega) = 1 - \frac{\vec{C}_{sn}^{\dagger} \mathbf{C}_{ss}^{-1} \vec{C}_{sn}}{C_{nn}}$$

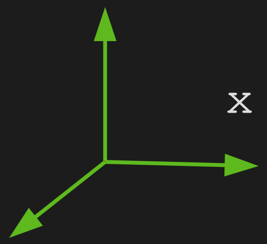
Isotropic and homogeneous seismic field for underground detectors.

All the **100 optimizations**

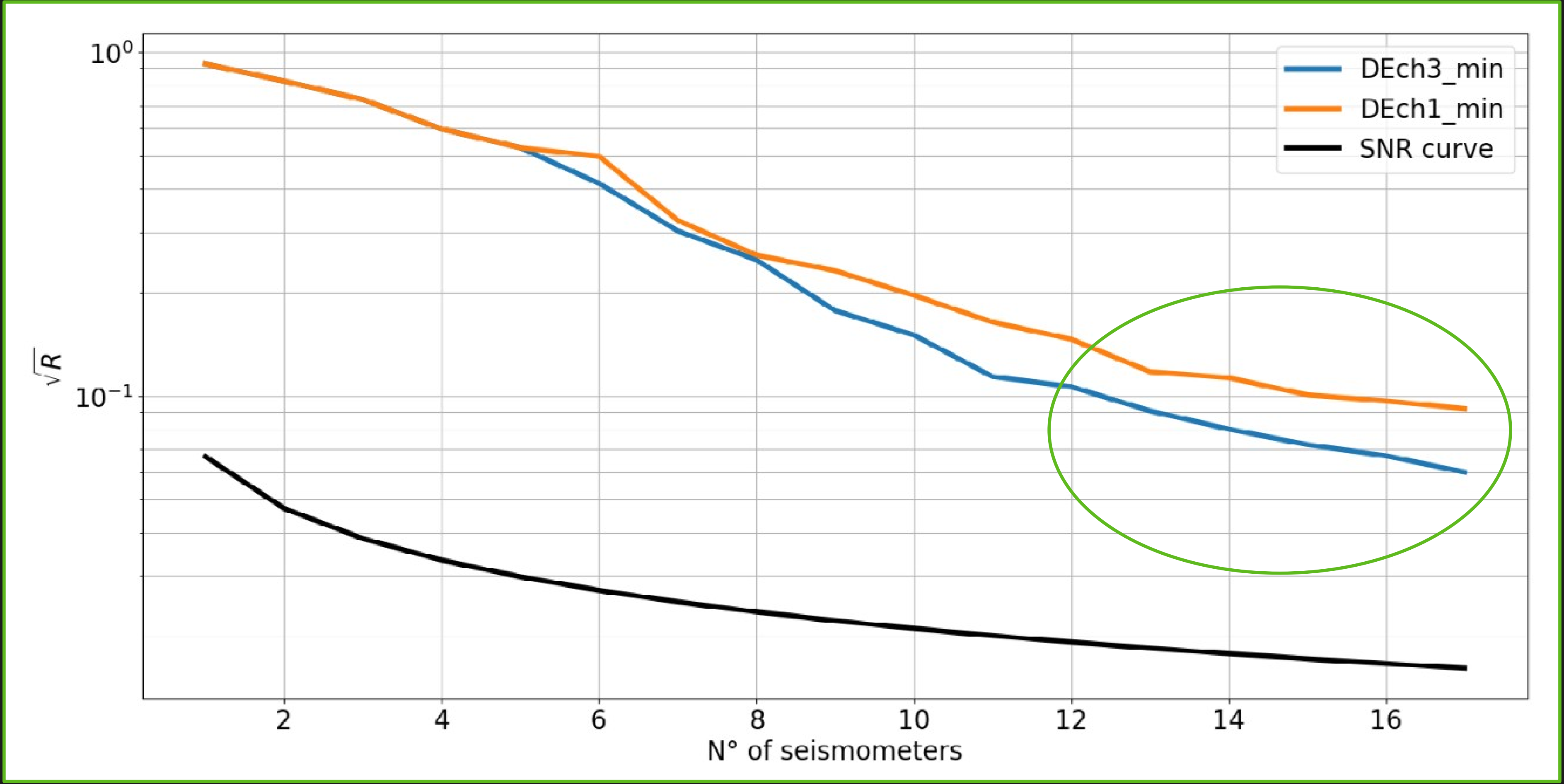
For arrays with **N = 6** seismometers each.

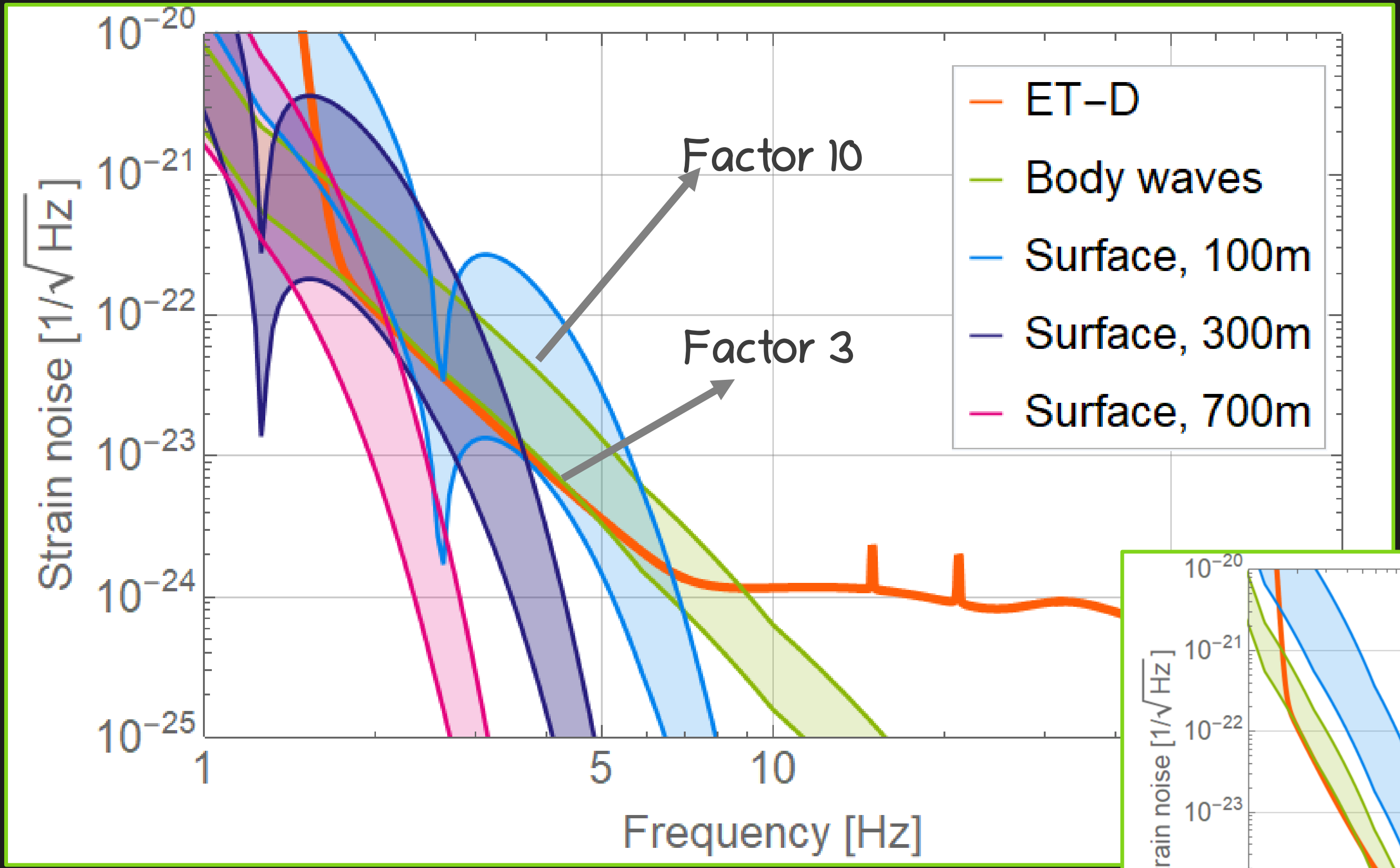


Underground case: we need to consider all the 3 directions of the seismic displacement:

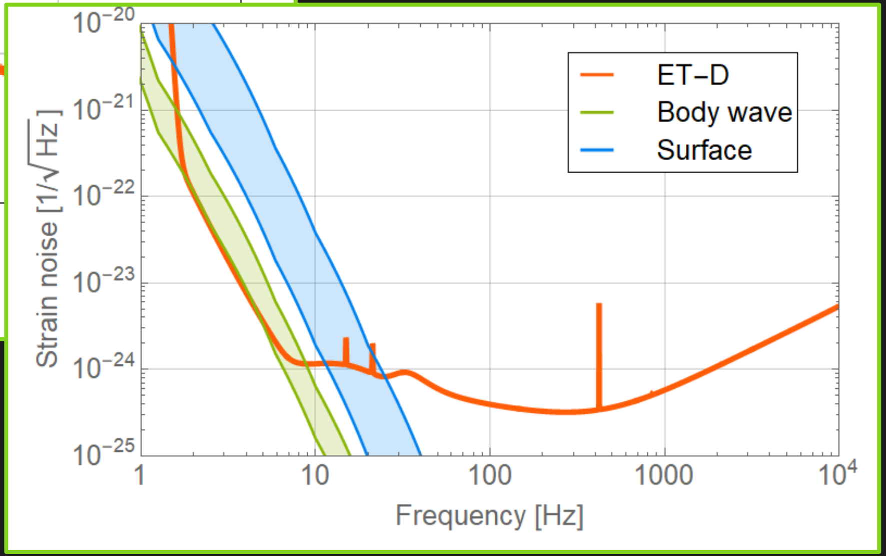


The more, the better:



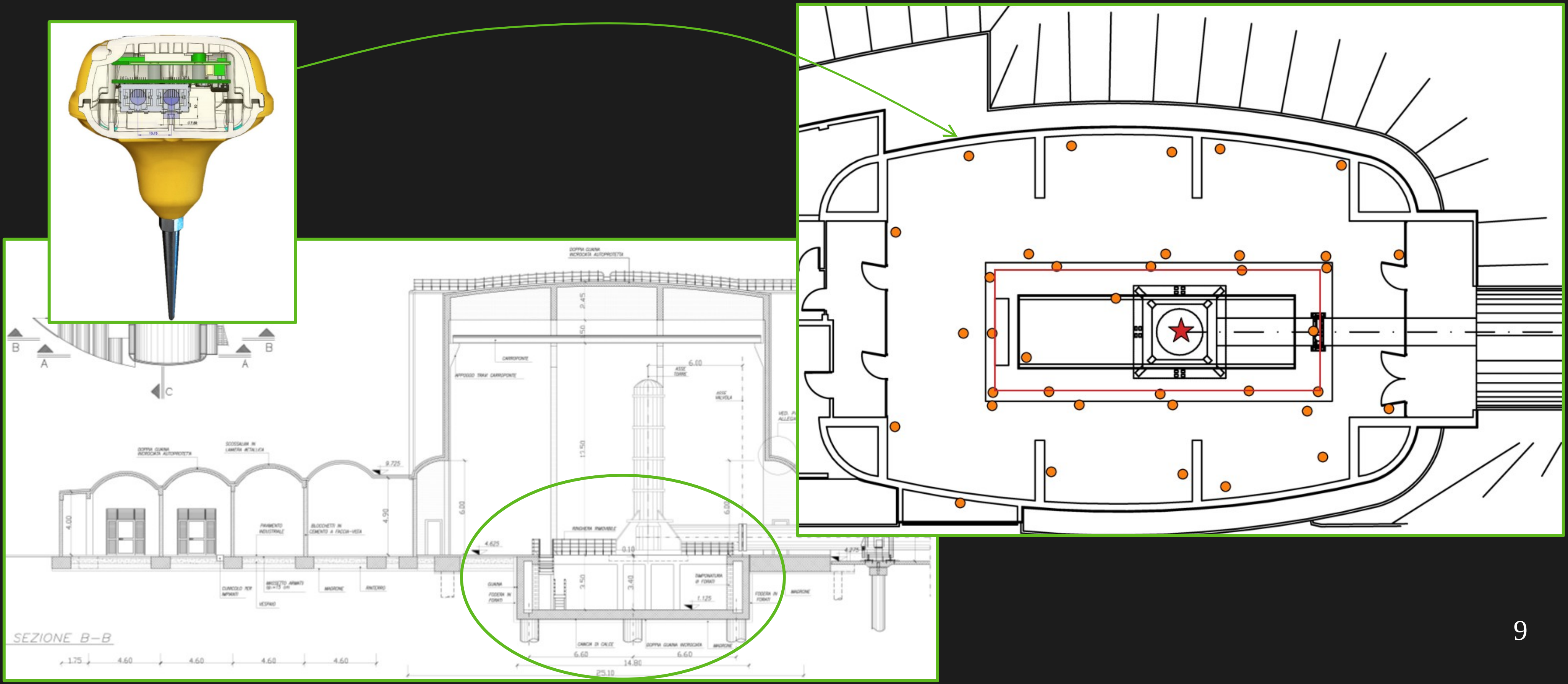


Strain noise [1/√Hz]
Frequency [Hz]
Factor 10
Factor 3
ET-D
Body waves
Surface, 100m
Surface, 300m
Surface, 700m



Strain noise [1/√Hz]
Frequency [Hz]
ET-D
Body wave
Surface

What if the seismic field is not homogeneous and isotropic?

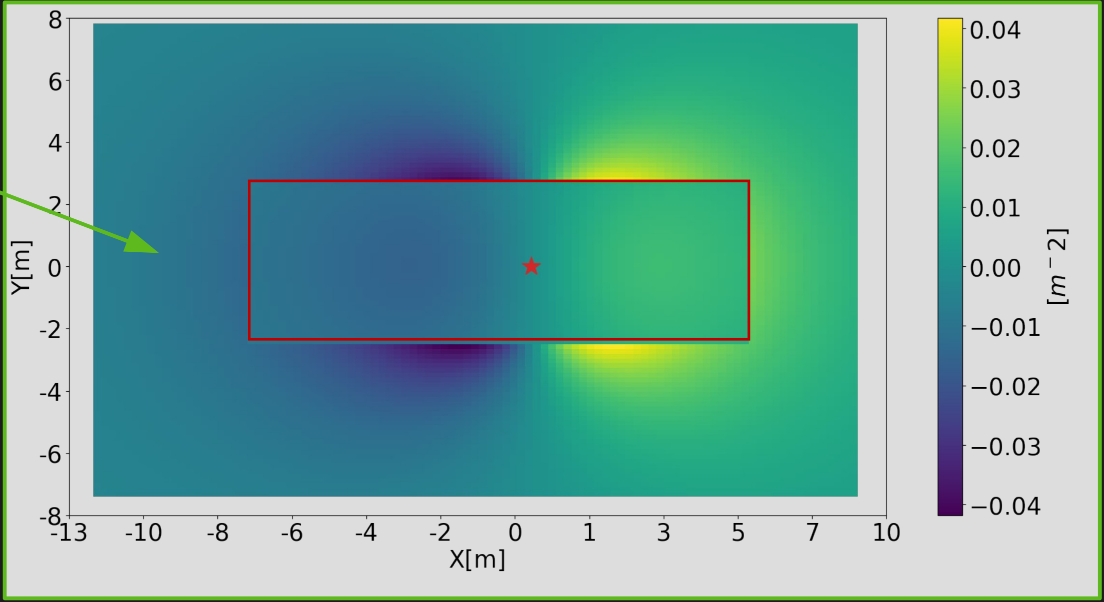


What if the seismic field is not homogeneous and isotropic?

$$C_{sn}(\mathbf{r},\mathbf{r}_0) = \mathcal{C}\int C_{ss}(\mathbf{r},\mathbf{r}_1)\frac{x_0 - x}{(h(\mathbf{r}_1)^2 + |\mathbf{r}_1 - \mathbf{r}_0|^2)^{3/2}}d\mathbf{r}_1$$

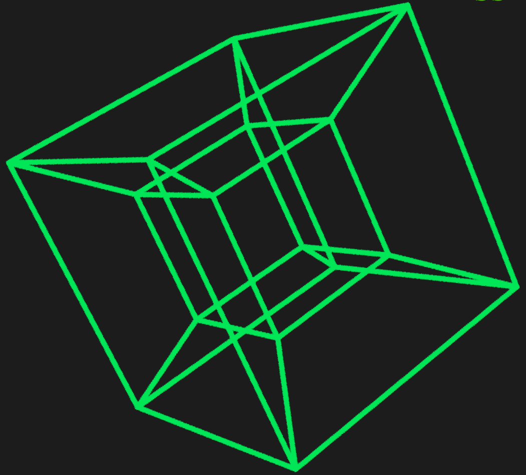
$$R(\omega) = 1 - \frac{\vec{C}_{sn}^{\dagger}\mathbf{C}_{ss}^{-1}\vec{C}_{sn}}{C_{nn}}$$

$$C_{sn}(\mathbf{r},\mathbf{r}_0) = \mathcal{C}\int C_{ss}(\mathbf{r},\mathbf{r}_1)\mathcal{K}(\mathbf{r}_1,\mathbf{r}_0)d\mathbf{r}_1$$

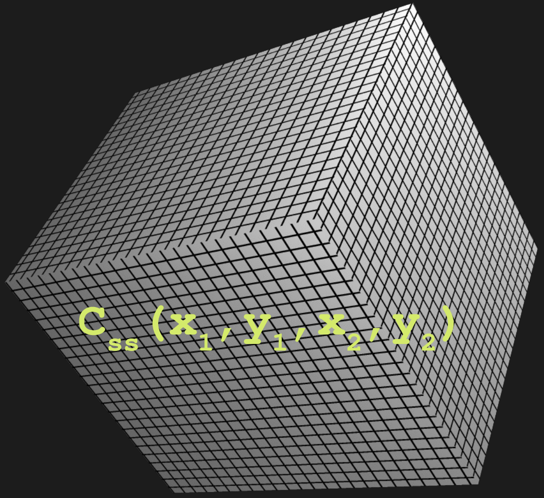


In the end, we only need to know this (and we can have it from data)

Every point of $C_{ss}(x_1,y_1,x_2,y_2)$ in the 4D space is calculated as before → We can virtually sample as many values of $C_{ss}(x_1,y_1,x_2,y_2)$ as we want, **wherever** we want.



Virtual Sampling + Linear interpolation: we created a **surrogate model** of $C_{ss}(x_1,y_1,x_2,y_2)$



$$C_{ss}(x_1,y_1,x_2,y_2) = \langle (FFT^*\{s(x_1,y_1)(\omega)\}\ FFT\{s(x_2,y_2)(\omega)\}) \rangle$$

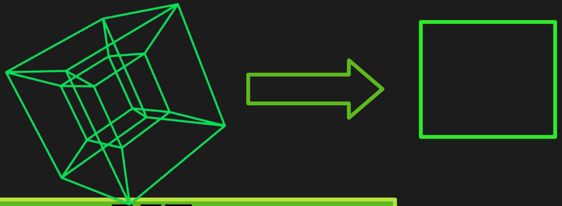
ith seismometer's data stream (1 hour, for example)

FFT_1

FFT_2

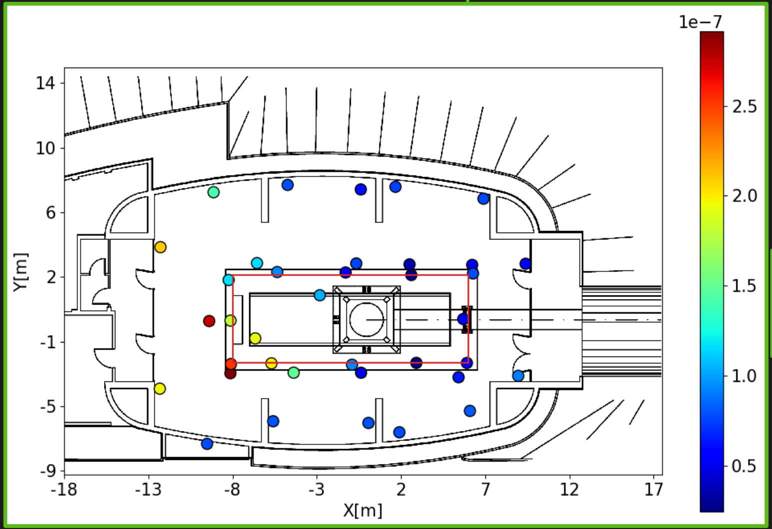
N segments with
50% overlapping

...

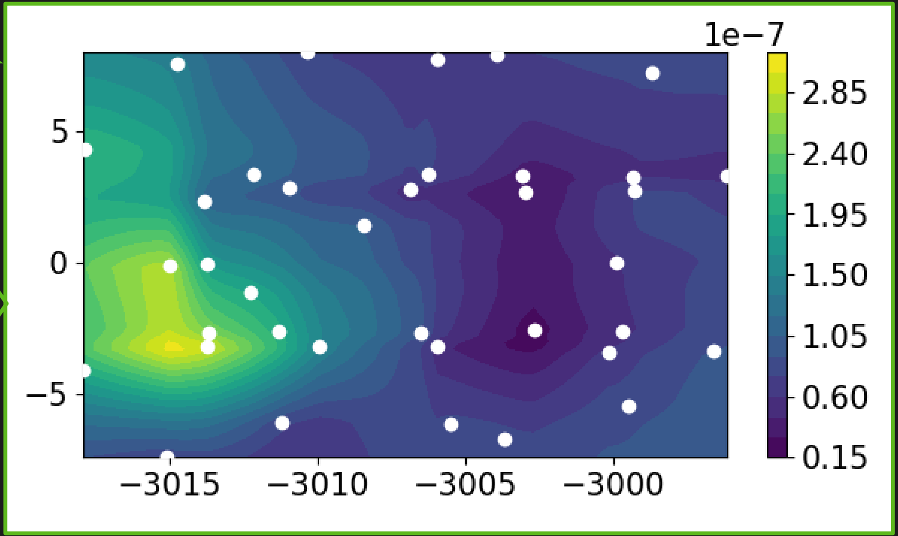


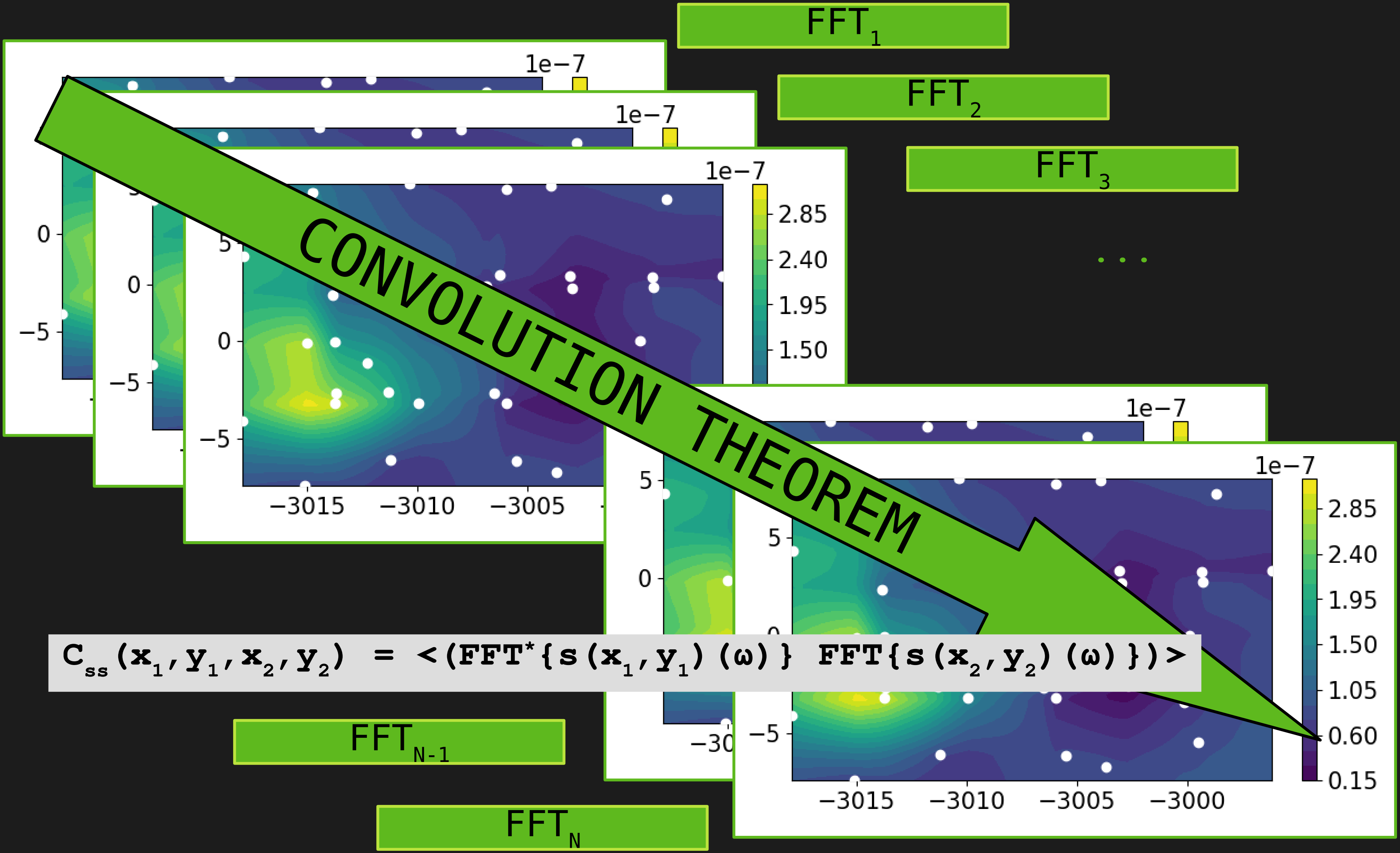
FFT_{N-1}

FFT_N

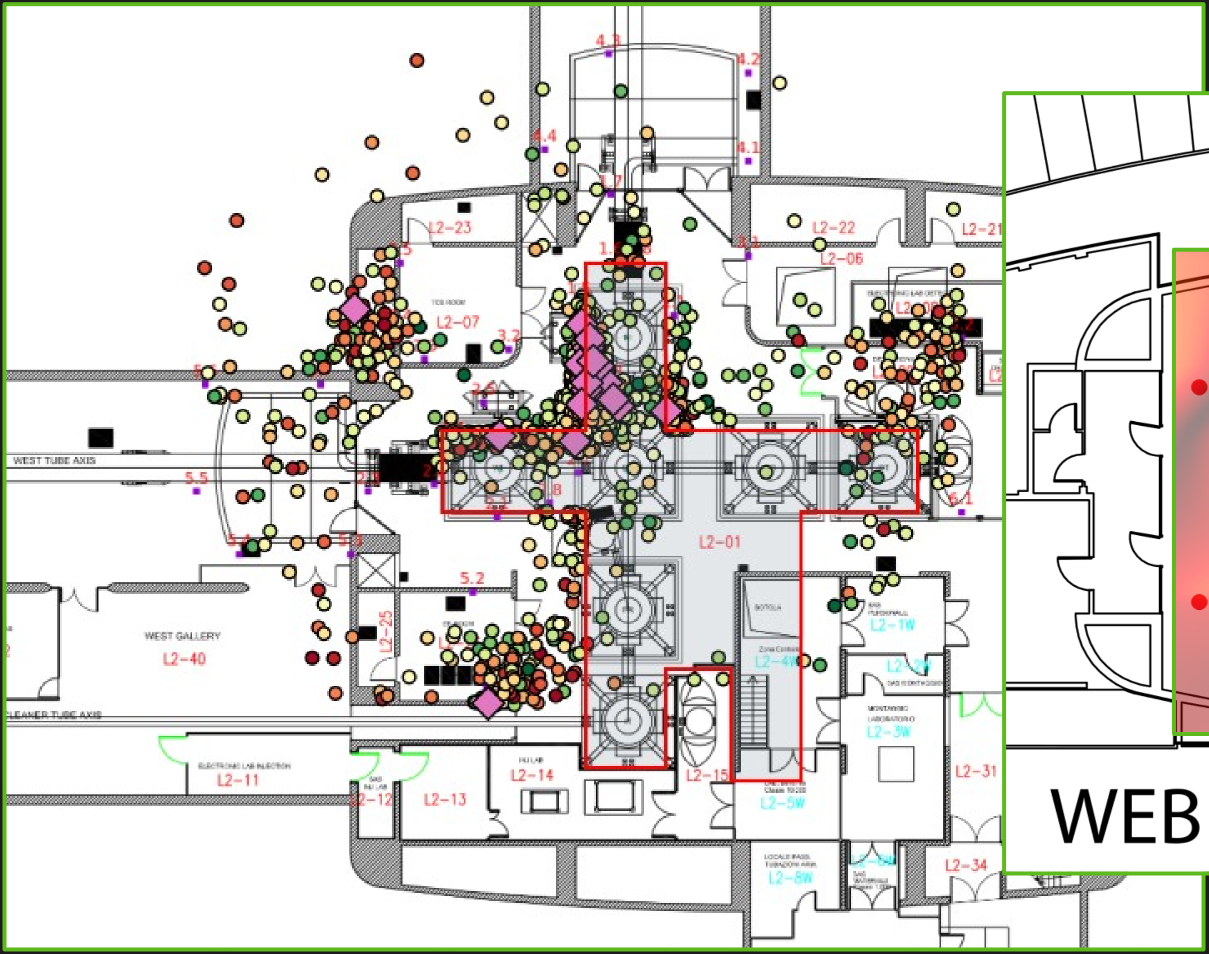


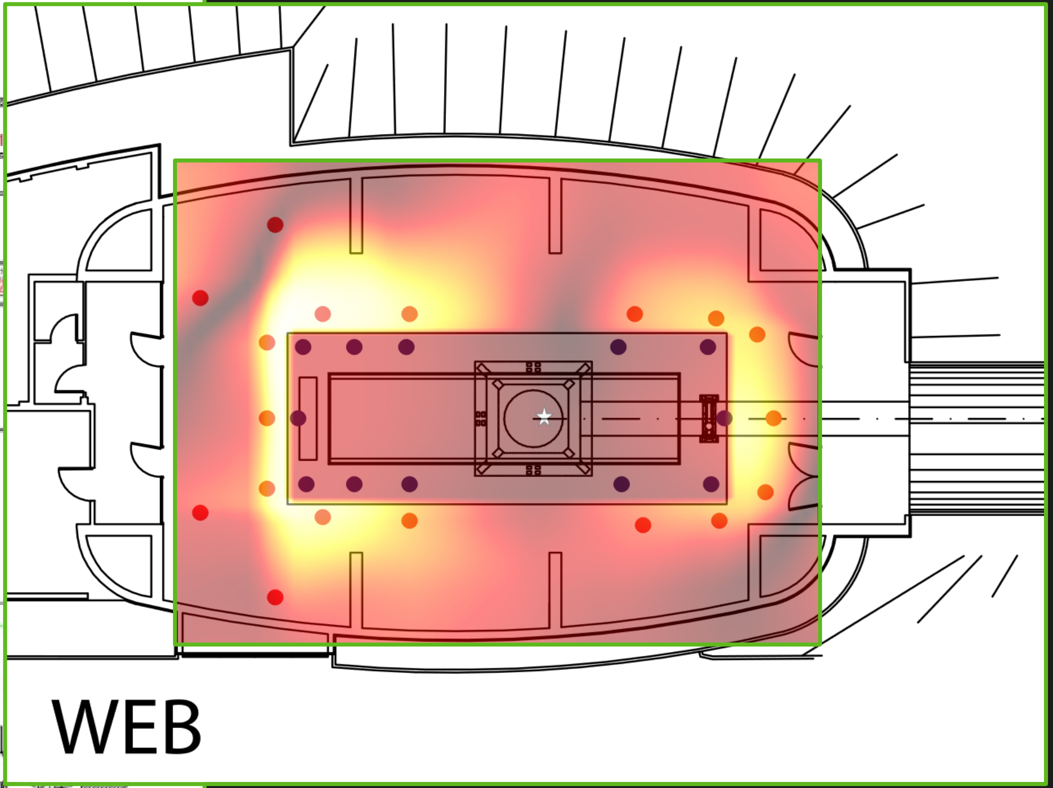
Gaussian Process





$$C_{ss}(x_1, y_1, x_2, y_2) = \langle (FFT^*\{s(x_1, y_1)(\omega)\}\ FFT\{s(x_2, y_2)(\omega)\}) \rangle$$


WEST TUBE AXIS
WEST GALLERY
L2-40
CLEANER TUBE AXIS
L2-23
L2-07
L2-22
L2-06
L2-21
L2-01
L2-11
L2-12
L2-13
L2-14
L2-15
L2-25
L2-1W
L2-3W
L2-4W
L2-5W
L2-8W
L2-31
L2-34


WEB

Which are the best hyper-parameters?

Hyper-parameters: they are **external to the model** and cannot be estimated from the data (like the learning rate for neural networks). However, they can be **optimized** in 2 ways:

Fully Bayesian framework:

→ non-gaussian likelihood → rely on **Monte Carlo methods** (computationally expensive)

or

Maximizing the log-likelihood:

Optimization + matrix inversion

Gaussian Processes are non-parametric models.

Which are the best hyper-parameters?

Hyper-parameters: they are **external to the model** and cannot be estimated from the data (like the learning rate for neural networks). However, they can be **optimized** in 2 ways:

Fully Bayesian framework:

→ non-gaussian likelihood
→ rely on **Monte Carlo methods** (computationally expensive)

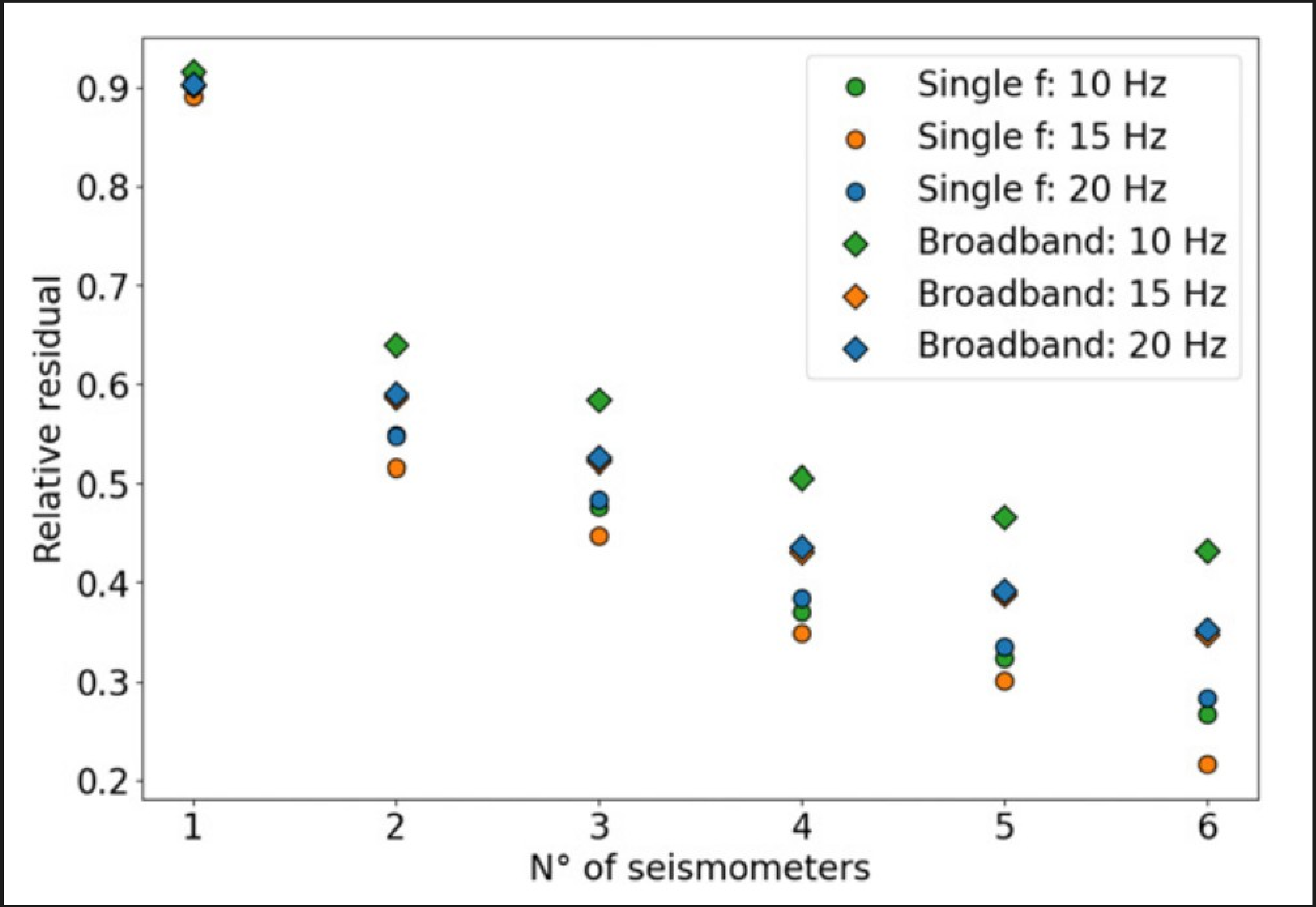
We need seismic simulations for this!!!

See **Tomislav** presentation for more details

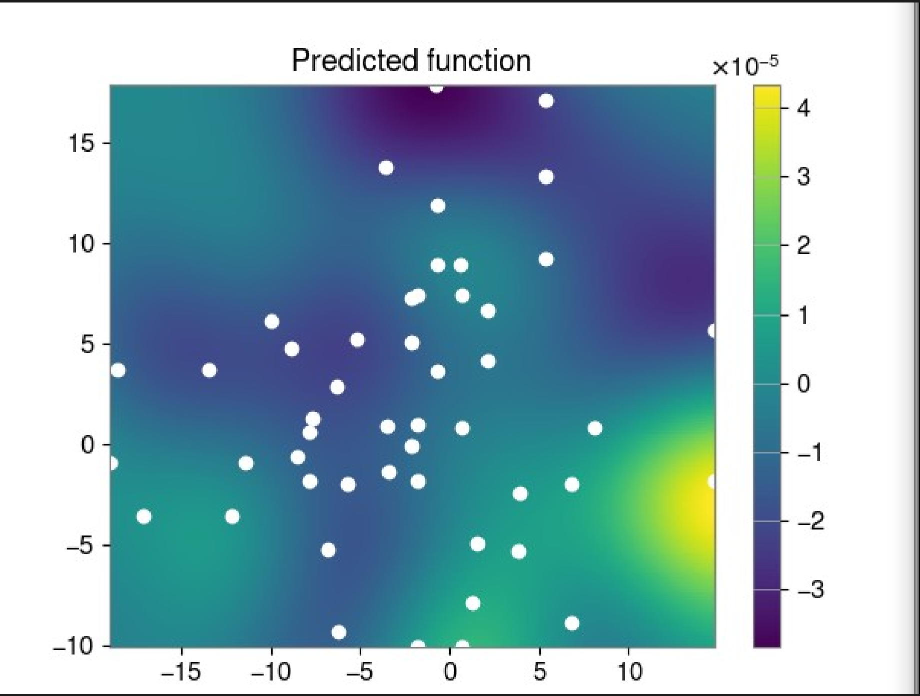
Summary

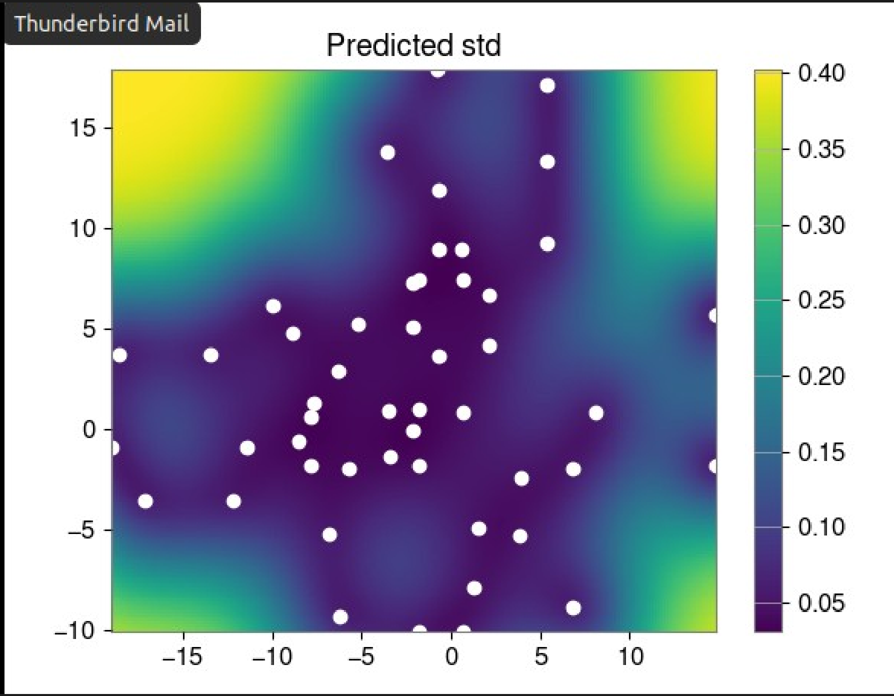
- **Newtonian noise** (NN) affects the **low frequency band** of GW detectors
- We can reduce it with an **active noise cancellation**
- If the **site is noisy** we will need up to a **factor 10** of NN reduction
- For a **factor 10** we will need **up to 15 sensors** for each TM → **€€€ €€€€€!!!!!**
- It will be difficult/expensive collecting seismic data for the array optimization in ET → we have to rely also on **seismic simulations as prior knowledge** for Gaussian Process!

Thank you for your attention!!!



0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
Relative residual
1
2
3
4
5
6
N° of seismometers
Single f: 10 Hz
Single f: 15 Hz
Single f: 20 Hz
Broadband: 10 Hz
Broadband: 15 Hz
Broadband: 20 Hz

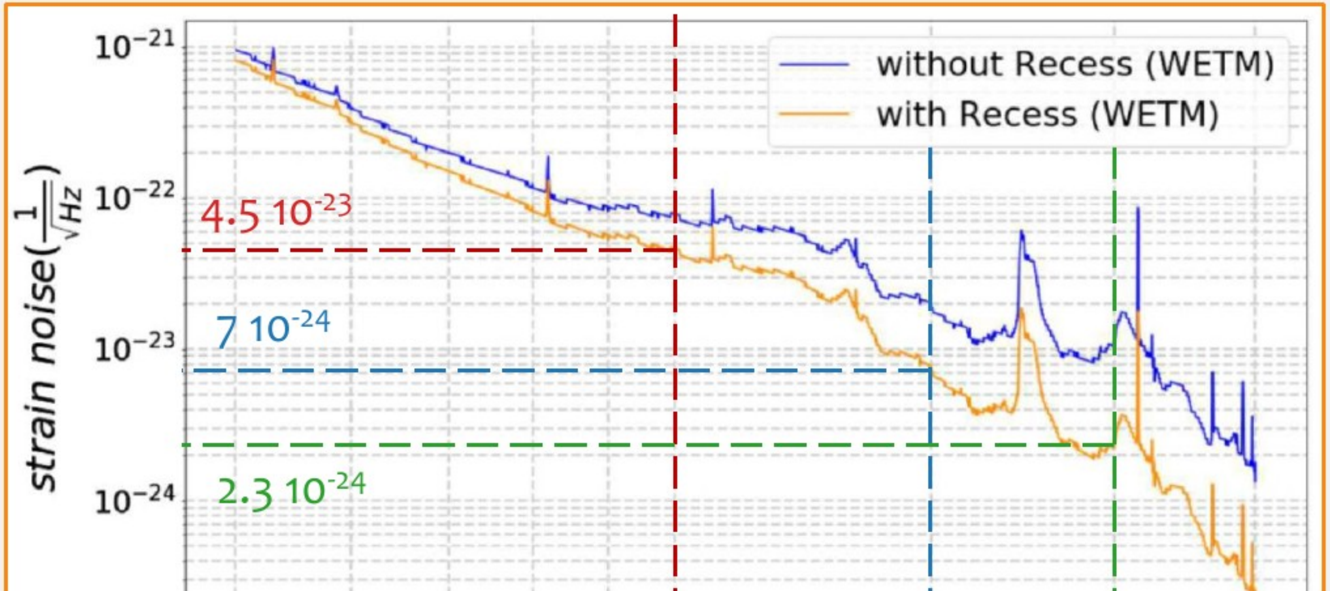

Predicted function
×10⁻⁵
15
10
5
0
−5
−10
−15
−10
−5
0
5
10
4
3
2
1
0
−1
−2
−3


Thunderbird Mail
Predicted std
15
10
5
0
−5
−10
−15
−10
−5
0
5
10
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05

Frequency	$\sqrt{C_{\mathrm{NN}}^{\mathrm{iso}}}$	$\sqrt{a}$
10 Hz	$1.33 \cdot 10^{-23}\ 1/\sqrt{\mathrm{Hz}}$	$4.04 \cdot 10^{-23}\ 1/\sqrt{\mathrm{Hz}}$
15 Hz	$7.21 \cdot 10^{-24}\ 1/\sqrt{\mathrm{Hz}}$	$1.04 \cdot 10^{-23}\ 1/\sqrt{\mathrm{Hz}}$
20 Hz	$4.34 \cdot 10^{-24}\ 1/\sqrt{\mathrm{Hz}}$	$4.44 \cdot 10^{-24}\ 1/\sqrt{\mathrm{Hz}}$

Homogeneous and isotropic model

Estimated Newtonian Noise



finite-element simulation

Ayatri Singha, Stefan Hild, and Jan Harms. \Newtonian-noise reassessment for the Virgo gravitational-wave observatory including local recess structures". In: Classical and Quantum Gravity 37.10 (Apr. 2020), p. 105007.
doi: 10.1088/1361-6382/ab81cb. url: https://doi.org/10.1088/1361-6382/ab81cb.

Array optimization

Wiener filter to perform a NN cancellation (**time domain**):

$$\hat{x}(m) = \sum_{k=0}^{P-1} w_k \, y(m-k)$$

$$R(\omega) = 1 - \frac{\vec{C}_{sn}^{\dagger} \mathbf{C}_{ss}^{-1} \vec{C}_{sn}}{C_{nn}}$$

Wiener filter performances (**frequency domain**):

REMEMBER!!!

What if the seismic field is not homogeneous and isotropic?

Residual in frequency domain

$$R(\omega) = 1 - \frac{\vec{C}_{sn}^{\dagger} \mathbf{C}_{ss}^{-1} \vec{C}_{sn}}{C_{nn}}$$

$$C_{sn_i}(\omega) = E[s_i^*(\omega)n(\omega)]$$

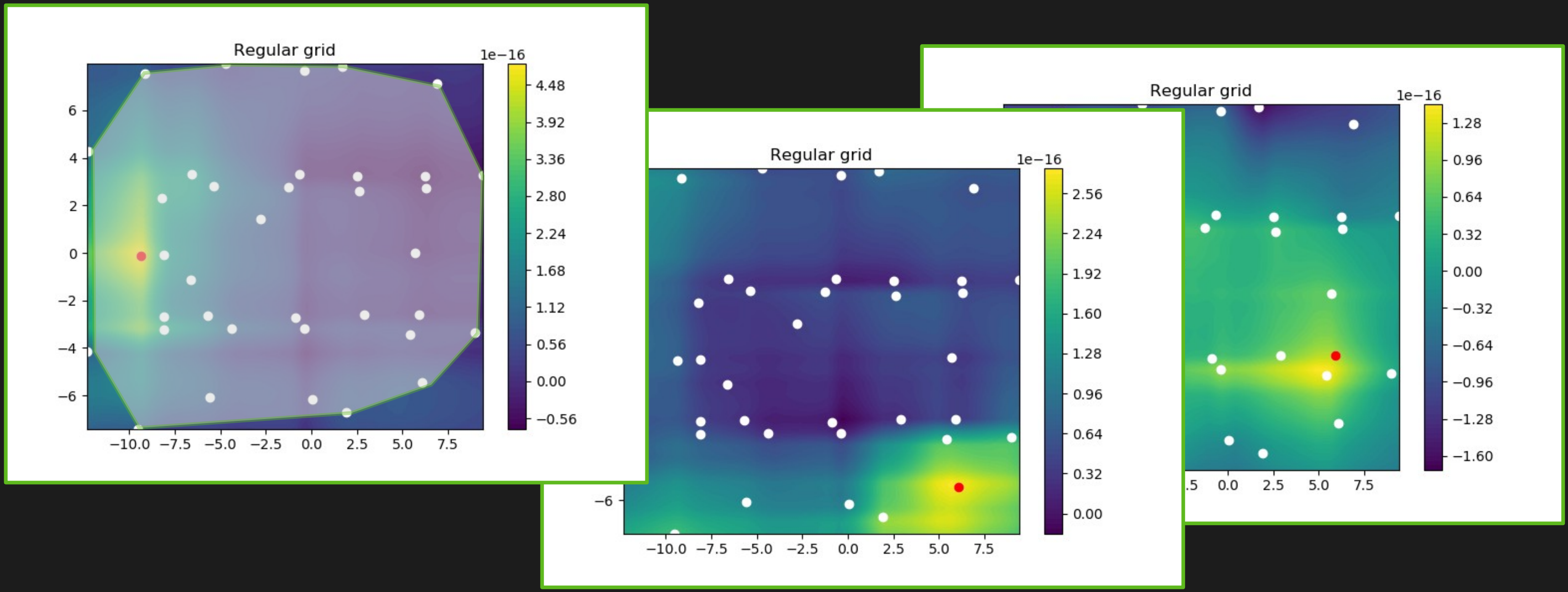
We can use a model (next slide)

$$C_{nn}(\omega) = E[n^*(\omega)n(\omega)]$$

We treat it just as a **unknown** constant

$$C_{ss_{ij}}(\omega) = E[s_i^*(\omega)s_j(\omega)]$$

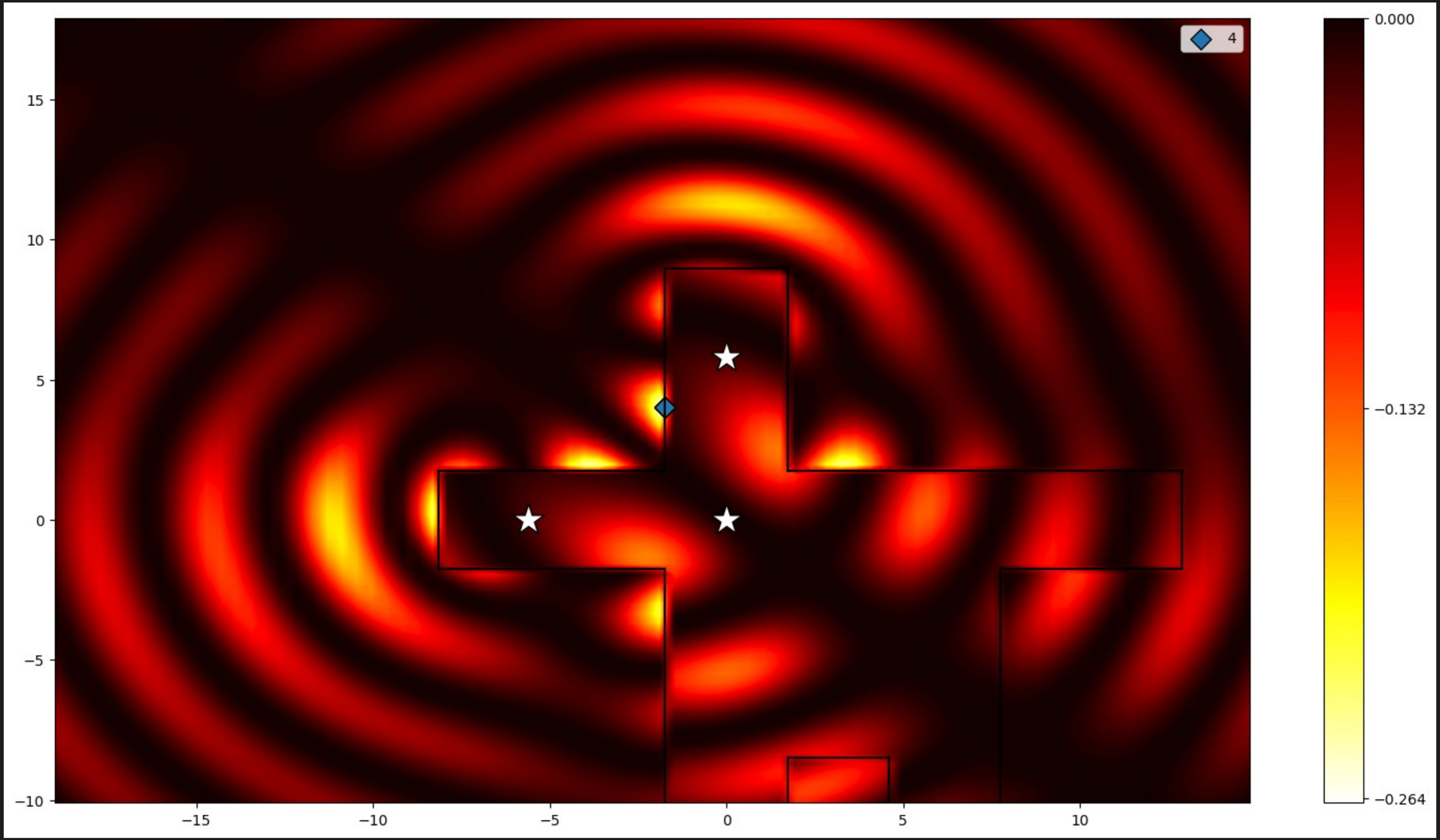
This is easy: we just need to **collect data**!



1) FFT of 37 seismometers' data (seismic displacement) → **2D gaussian process** at a frequency f_0: **Convolution theorem** → **surrogate model** of Css:

$$C_{ss}(x_1,y_1,x_2,y_2) = \langle(FFT^*\{s(x_1,y_1)(\omega)\}\ FFT\{s(x_2,y_2)(\omega)\})\rangle$$

2) Css Sampling → **4D Linear Interpolation on a Regular grid** (faster) → **Css** & **Csn** (integrated with Simpson method)


4
15
10
5
0
−5
−10
−15
−10
−5
0
5
10
0.000
−0.132
−0.264

Which are the best hyper-parameters?

Parameters: they **define the model** and can be learned from the data (e.g. coefficients of a linear model or the weights in a neural network).

Hyper-parameters: they are **external to the model** and cannot be estimated from the data (like the learning rate for neural networks). However, they can be **optimized** in 2 ways:

Fully Bayesian framework:

→ non-gaussian likelihood → rely on **Monte Carlo methods** (computationally expensive)

or

Maximizing the log-likelihood:

Optimization + matrix inversion

Gaussian Processes are non-parametric models.

Wiener Filter is the way:

Assumptions:

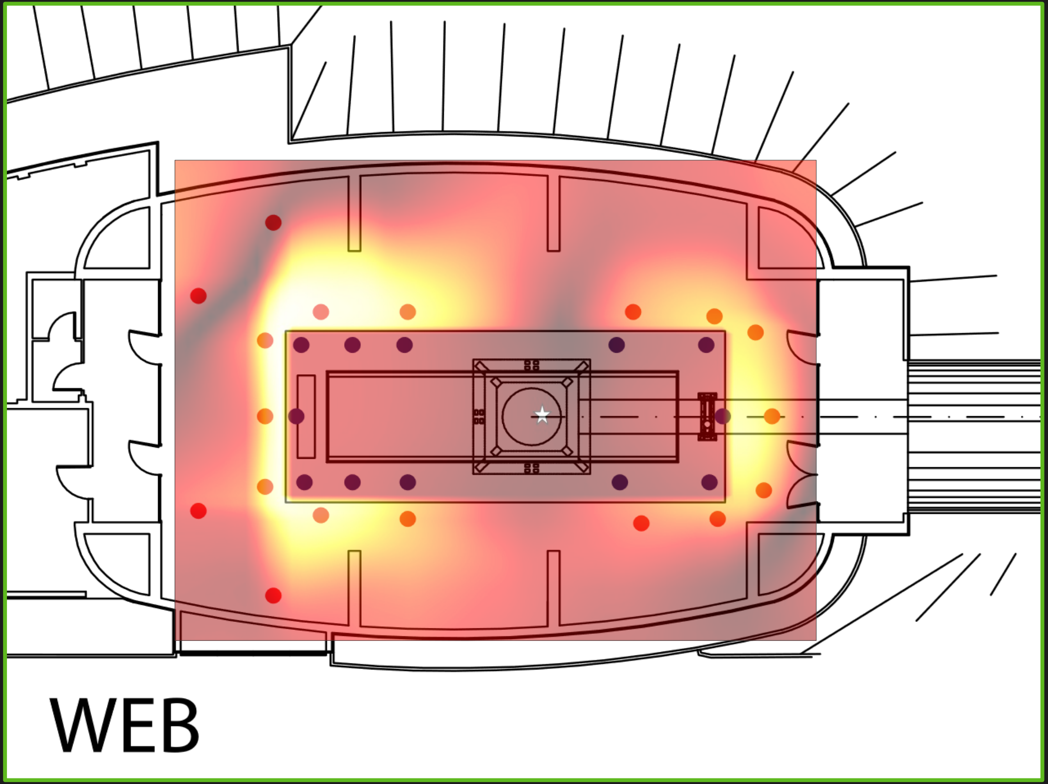
- Stationary signal
- **Linear** relationship

Filter order

$$\hat{x}(m) = \sum_{k=0}^{P-1} w_k \, y(m-k)$$

- Estimated value of the Newtonian noise
- Discrete time
- Wiener filter coefficients
- Measured signal (seismic displacement)

The residual will depend on the frequency, the number of sensors and on their positions:
In 2D we have **2N coordinates**, where N is the number of the sensors.



Update Wiener filter every hours: LINK

Array optimization

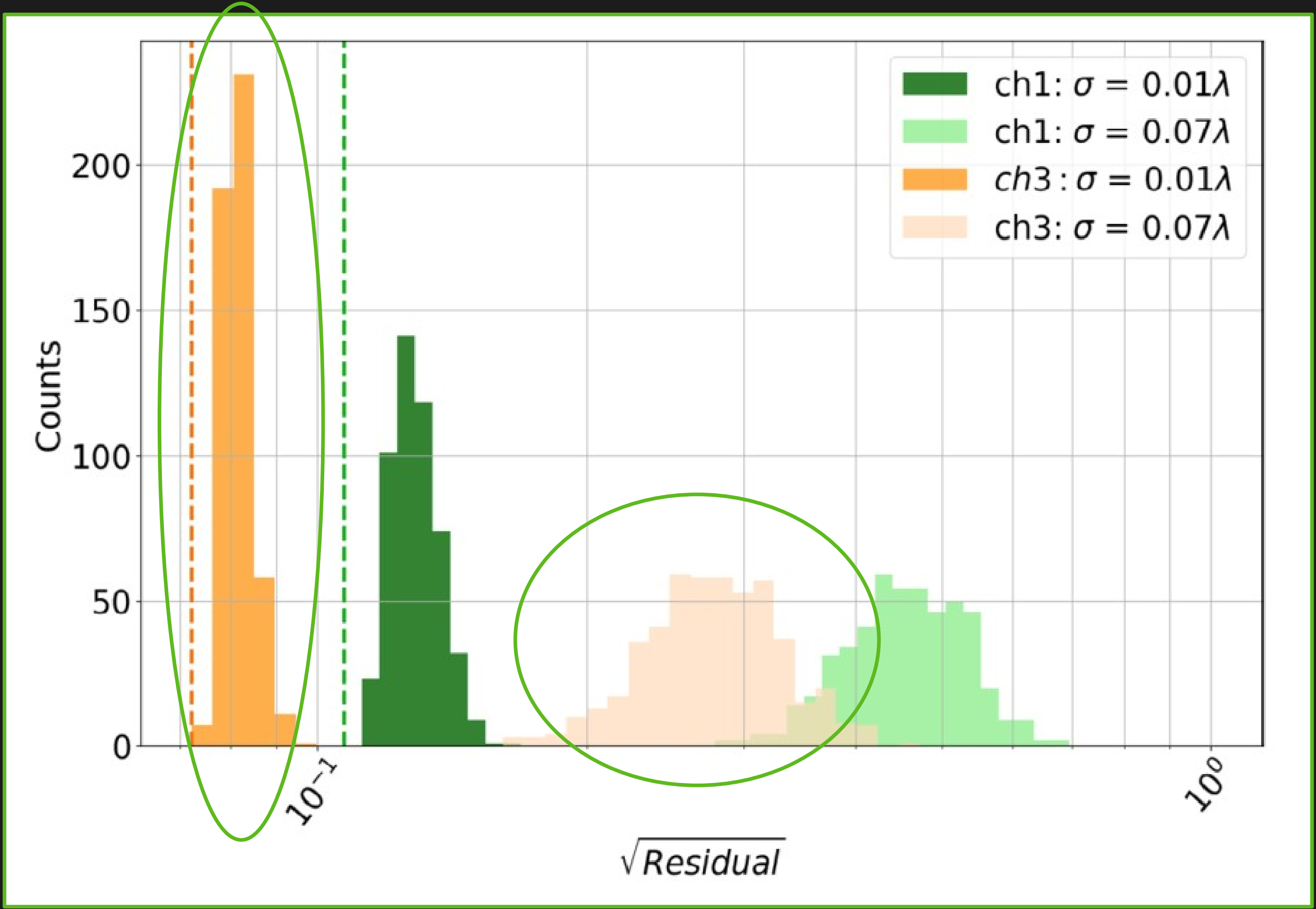
↓

Array deployment

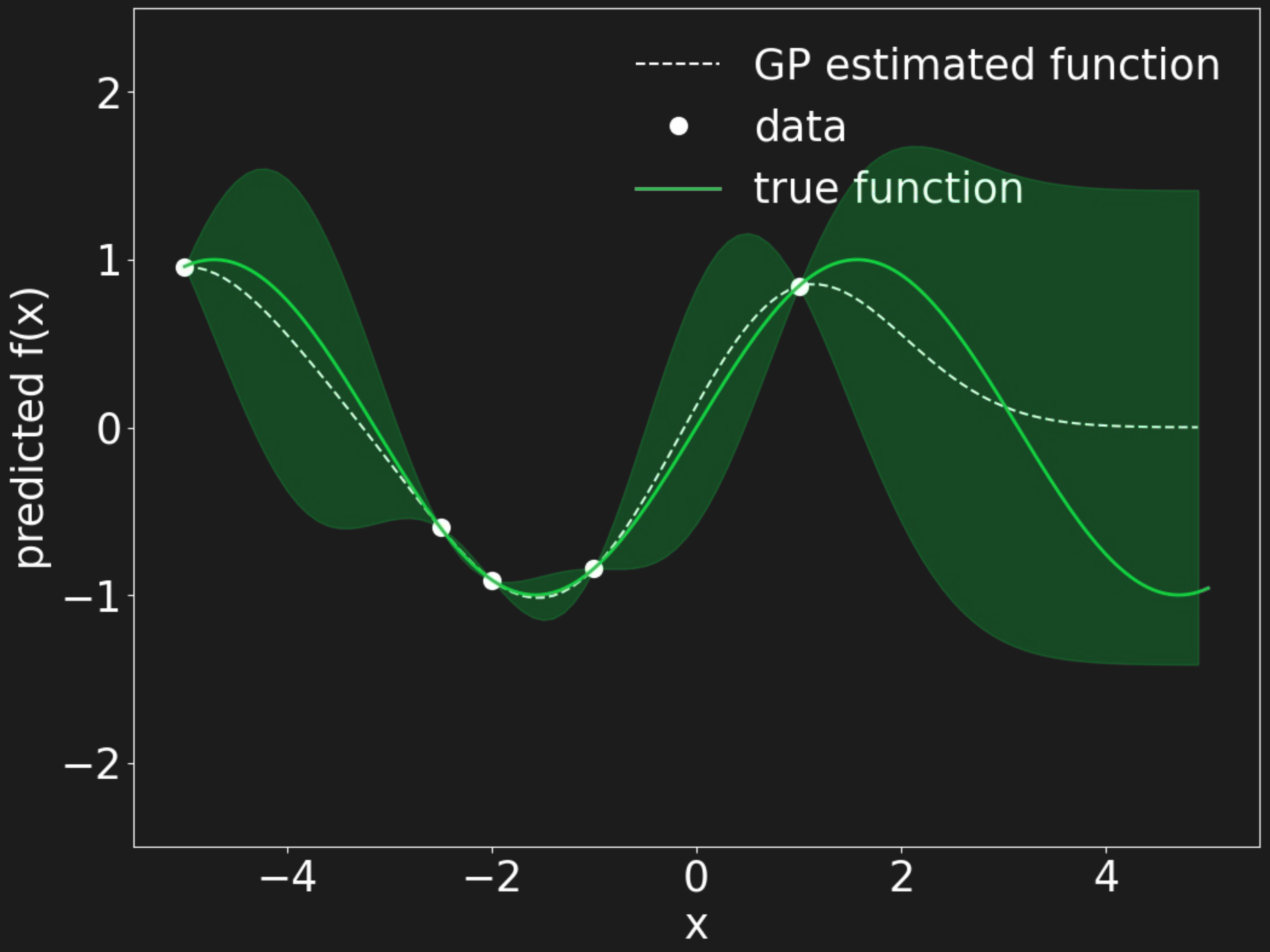
↓

Subtraction pipeline
(applying wiener filter)

We might misplace the sensors, then what…?



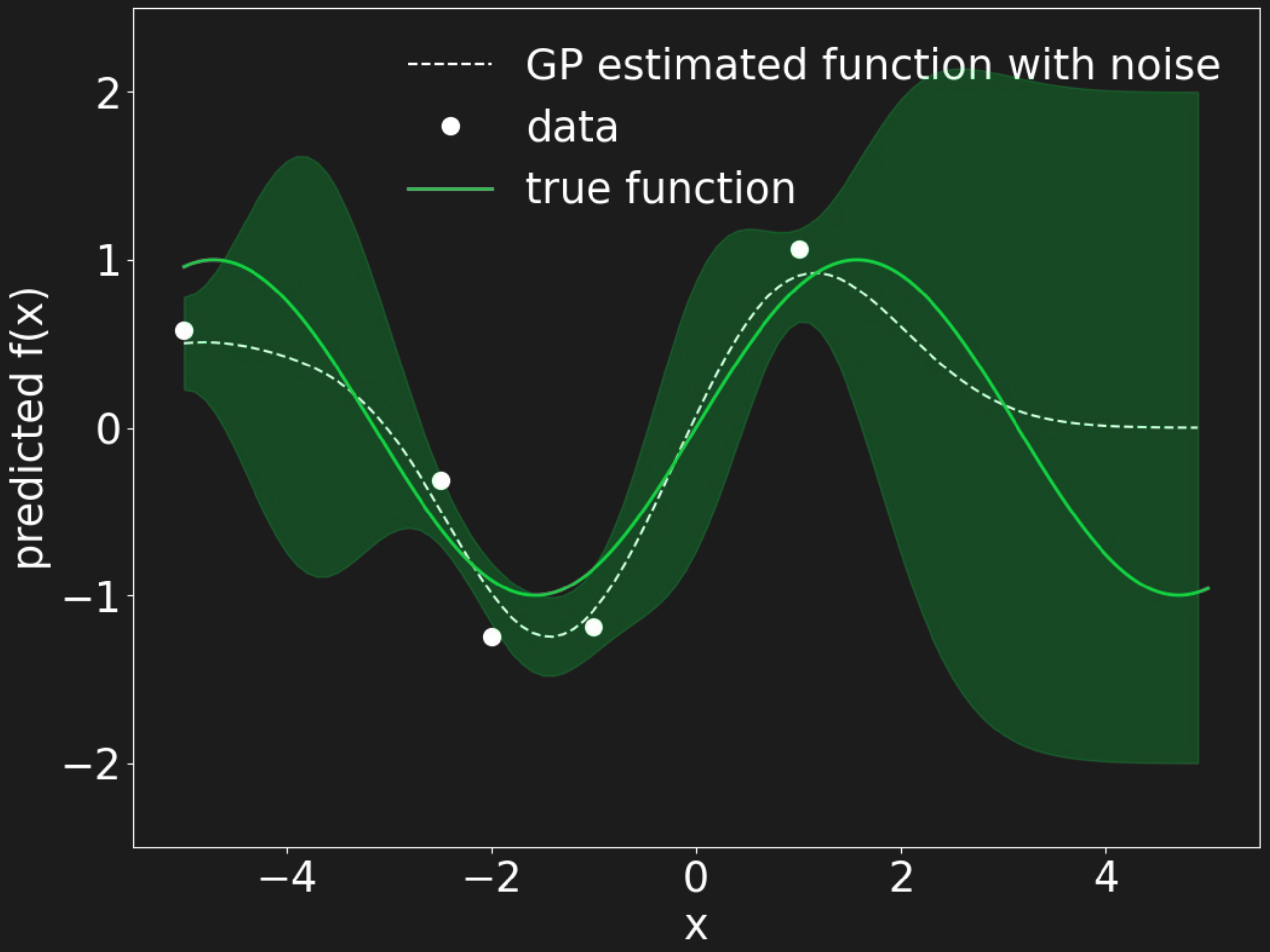
Free noise signal



Posterior obtained from the conditioning of the prior over the white data point. The white dashed curve represents $\mu*(x_*)$ and the shaded area $\pm 2\sigma_*(x_*)$.

The hyper-parameters were fixed: $\sigma_f=1$, $l=0$, $\sigma_\varepsilon=0$.

Noisy signal



Posterior obtained from the conditioning of the prior over the white data point. The white dashed curve represents $\mu_*(x_*)$ and the shaded area $\pm 2\sigma_*(x_*)$.

The hyper-parameters were fixed: σ_f=1, l=0 and σ_ε=0.4

Likelihood: given some parameters, the higer it is, the more likely it will be that we sample that observed data.

$$\log p(\mathbf{y}|\mathbf{x}_0) = -\frac{1}{2}\mathbf{y}^T(k(\mathbf{x}_0,\mathbf{x}_0) + \sigma_\varepsilon^2\bar{\mathcal{I}})^{-1}\mathbf{y} - \frac{1}{2}\log|k(\mathbf{x}_0,\mathbf{x}_0) + \sigma_\varepsilon^2\bar{\mathcal{I}}| - \frac{N}{2}\log 2\pi$$

Data fit:
It decreases monotonically with the length scale (l) → less flexible model → worse fit

Minus complexity penalty:
The simpler the model (big l scale) the bigger it becomes

N=number of training points

Likelihood: try to favour the least complex model able to explain the data (automatic Occam Razor).