Emulation of the cosmic density field

Mischa Knabenhans (UZH)

AMLD, Lausanne, January 2020

University of Zurich UZH

Image source: NASA/Apollo 17 crew; taken by either Harrison Schmitt or Ron Evans - https://web.archive.org/web/20160112123725/http://grin.hq.nasa.gov/ABSTRACTS/GPN-2000-001138.html (image link);

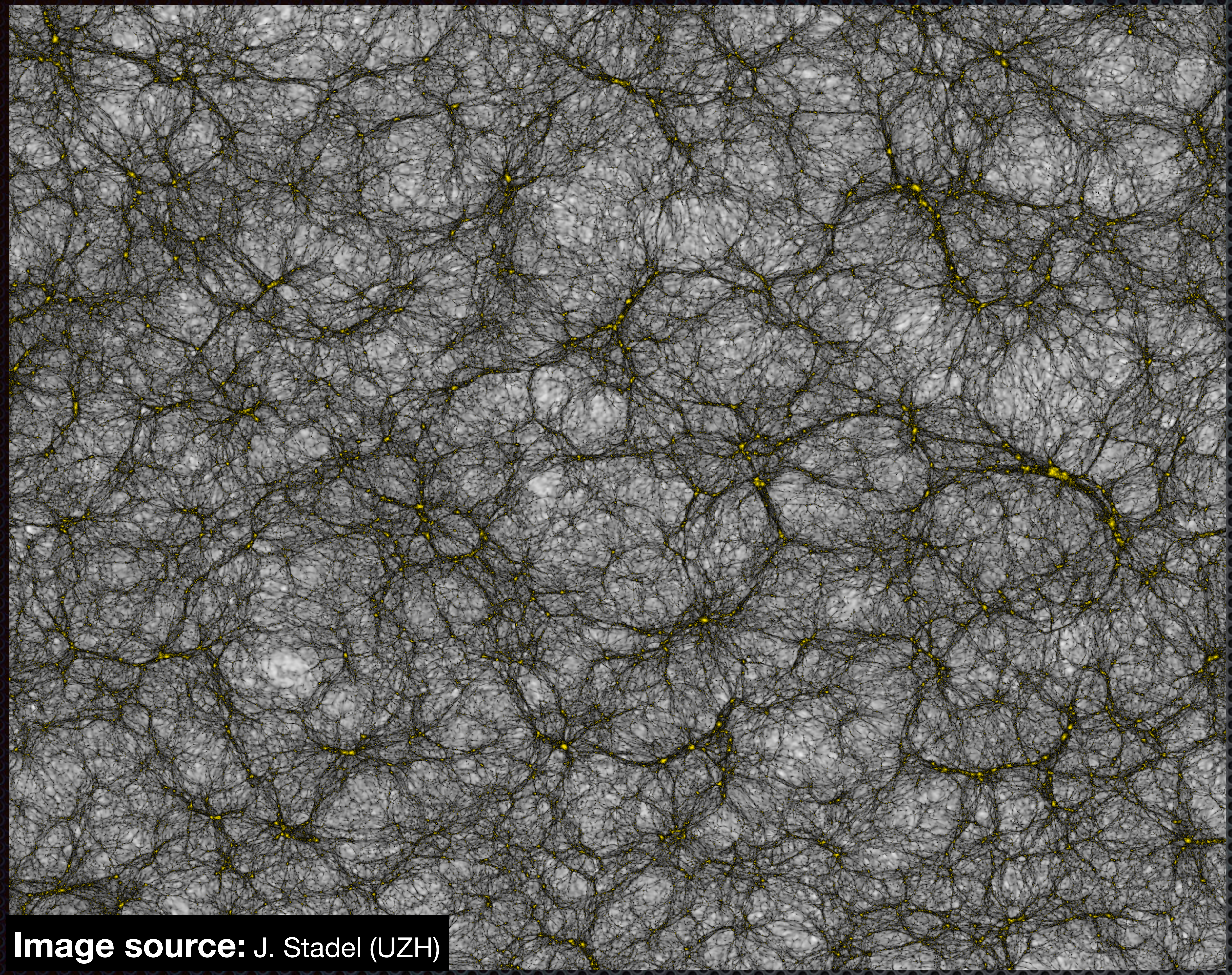
Image source: https://wtamu.edu/~cbaird/sq/2013/08/28/when-do-the-planets-in-our-solar-system-all-line-up/

Image source: https://www.starobserver.org/tag/andromedagalaxie/page/2/; **Credits:** Robert Gendler

Image source: Doug Potter (UZH)

Image source: J. Stadel (UZH)

The cosmic density field



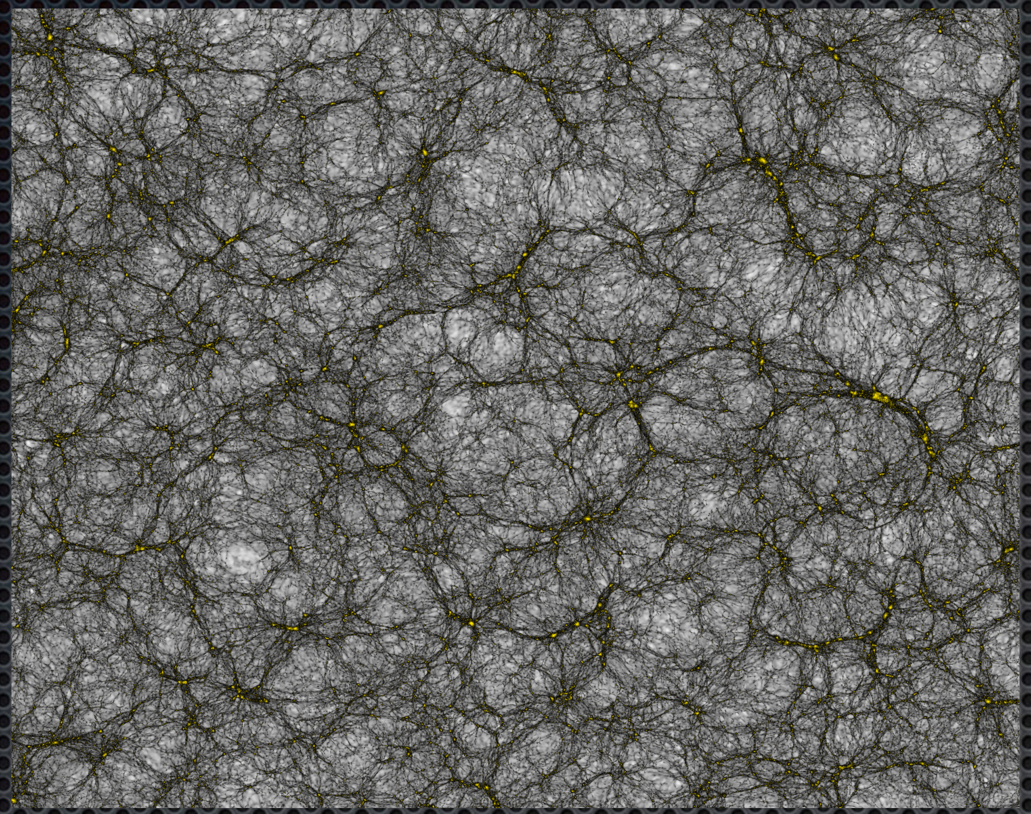
is described by fundamental properties of the Universe (called „cosmological parameters“) such as

- the amount of matter
- the amount of radiation
- amplitude of initial density fluctuations
- others

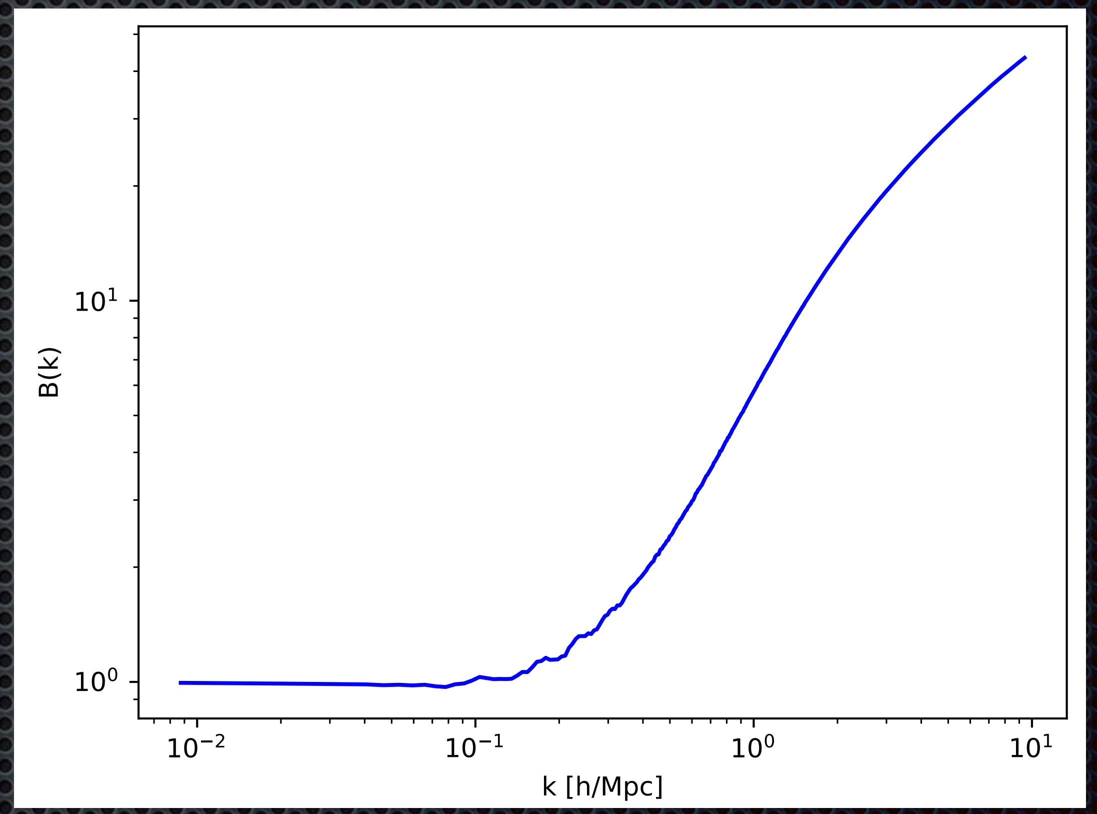
Cosmological parameters are the features

Statistics of the cosmic density field

$$\begin{pmatrix} \Omega_b \\ \Omega_m \\ \vdots \\ A_s \end{pmatrix} \xrightarrow{\text{Sim}}$$



post-processing → ⋯⋯ →



feature vector

statistical measure = target

What is the mapping [cosmo params → statistical measure]?

Two options:

Standard approach: Simulation (physics-driven)
Are very accurate but computationally demanding (HPC)

Alternative approach: Emulation (data-driven)
Still accurate but less demanding (HPC)

Two options:

Standard approach: Simulation (physics-driven)
Are very accurate but computationally demanding (HPC)

Alternative approach: Emulation (data-driven)
Still accurate but less demanding (HPC)

We use polynomial chaos expansion

Polynomial chaos expansion (PCE)

Supervised regression technique based on orthonormal polynomials:

- PCE performs well even when trained on small data sets (!!!)
- PCE is interpretable (coefficients have a statistical interpretation)
- …

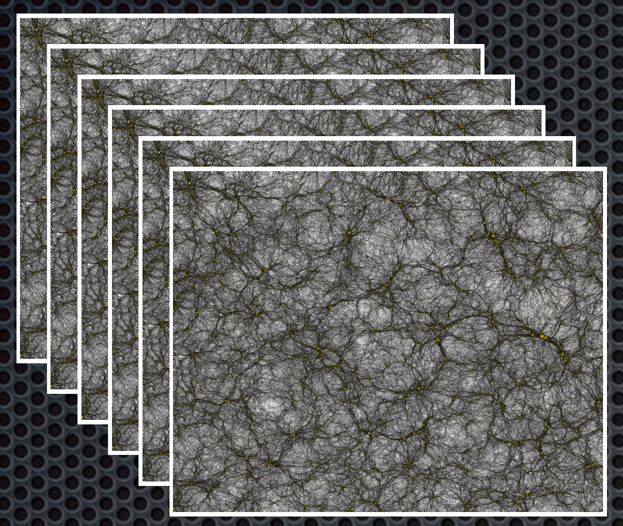
www.uqlab.com

The training pipeline

$$[\Omega_b^1, \Omega_m^1, \ldots, A_s^1]$$
$$[\Omega_b^2, \Omega_m^2, \ldots, A_s^2]$$
$$\vdots$$
$$[\Omega_b^n, \Omega_m^n, \ldots, A_s^n]$$

Sims

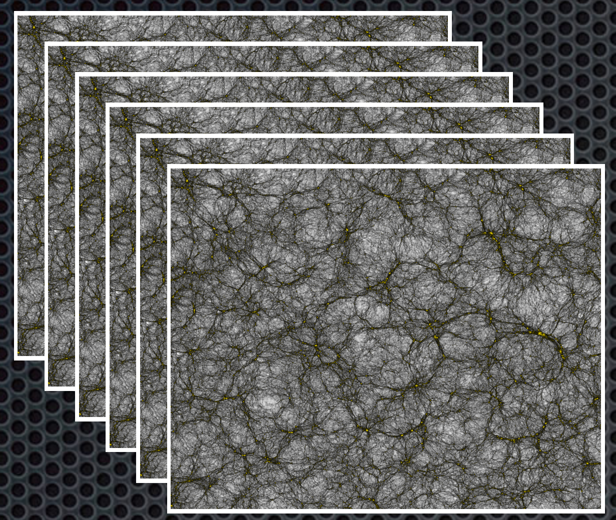
$\longrightarrow$



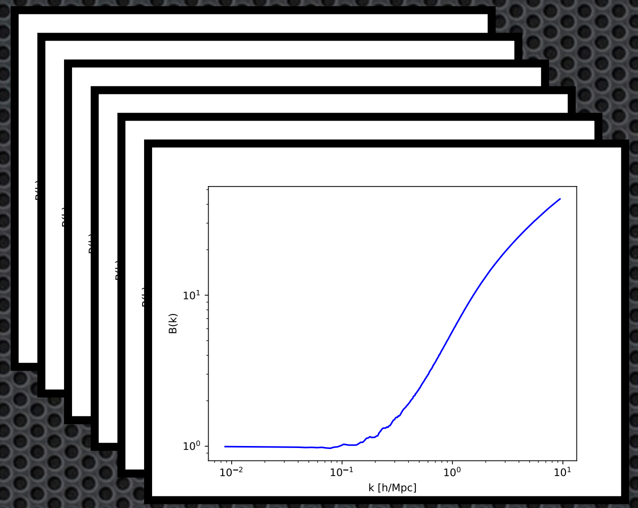
The training pipeline

$$[\Omega_b^1, \Omega_m^1, \ldots, A_s^1]$$
$$[\Omega_b^2, \Omega_m^2, \ldots, A_s^2]$$
$$\vdots$$
$$[\Omega_b^n, \Omega_m^n, \ldots, A_s^n]$$

Sims $\longrightarrow$



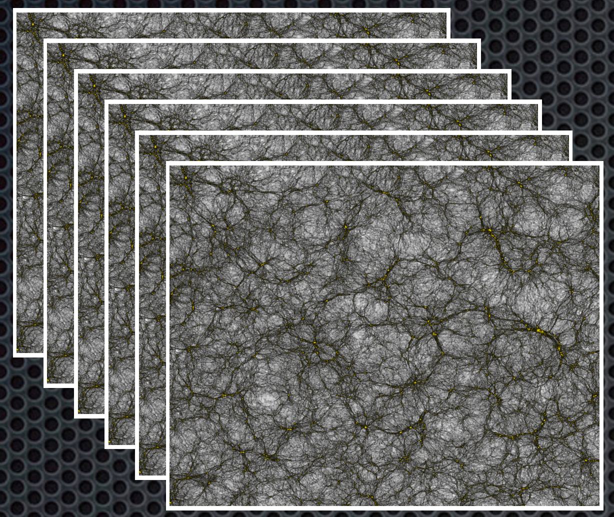
Stats $\longrightarrow$



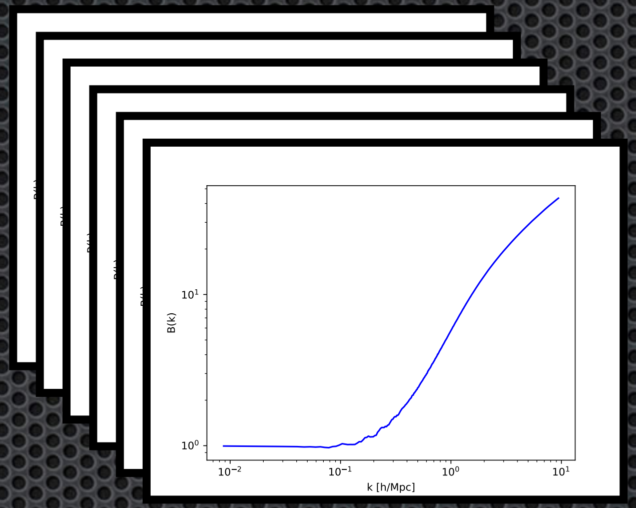
The training pipeline

$$[\Omega_b^1, \Omega_m^1, \ldots, A_s^1]$$
$$[\Omega_b^2, \Omega_m^2, \ldots, A_s^2]$$
$$\vdots$$
$$[\Omega_b^n, \Omega_m^n, \ldots, A_s^n]$$

Sims $\longrightarrow$

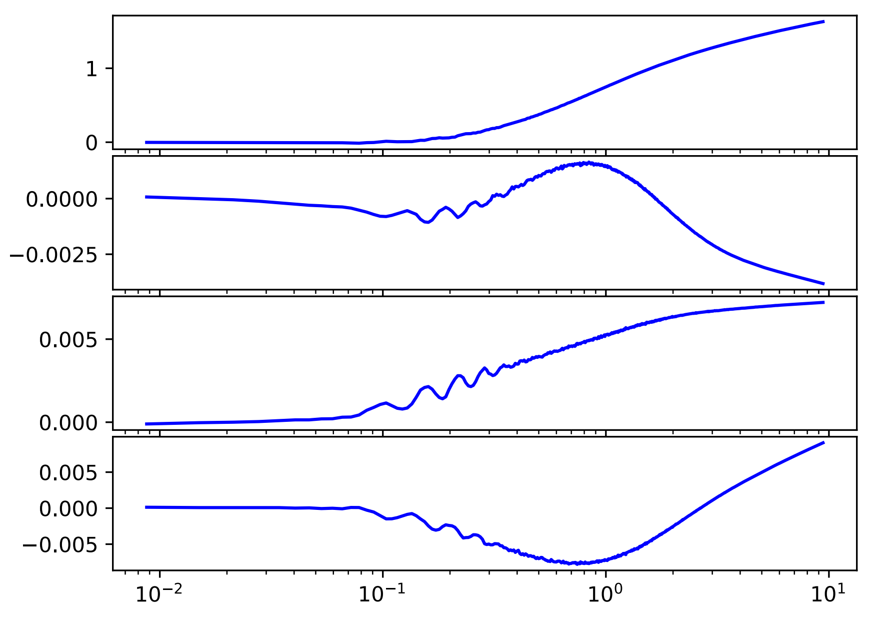


Stats $\longrightarrow$



PCA $\longrightarrow$

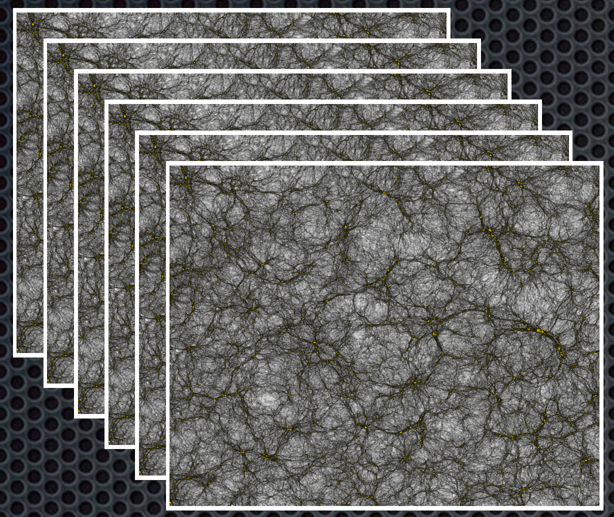
$$\begin{pmatrix} 1 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \end{pmatrix} \times$$



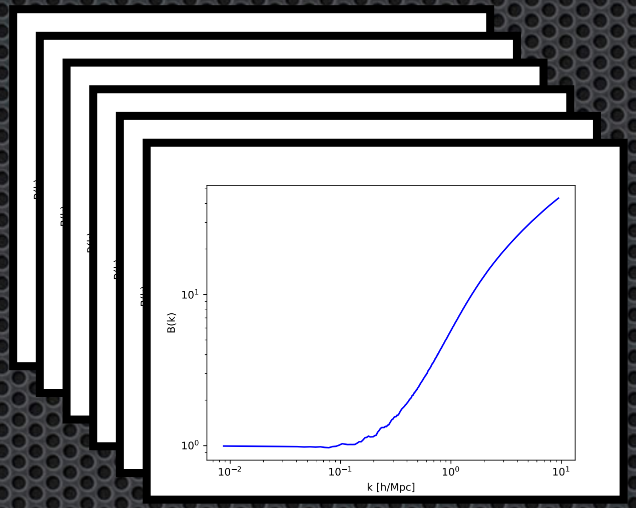
The training pipeline

$$[\Omega_b^1, \Omega_m^1, \dots, A_s^1]$$
$$[\Omega_b^2, \Omega_m^2, \dots, A_s^2]$$
$$\vdots$$
$$[\Omega_b^n, \Omega_m^n, \dots, A_s^n]$$

Sims $\longrightarrow$

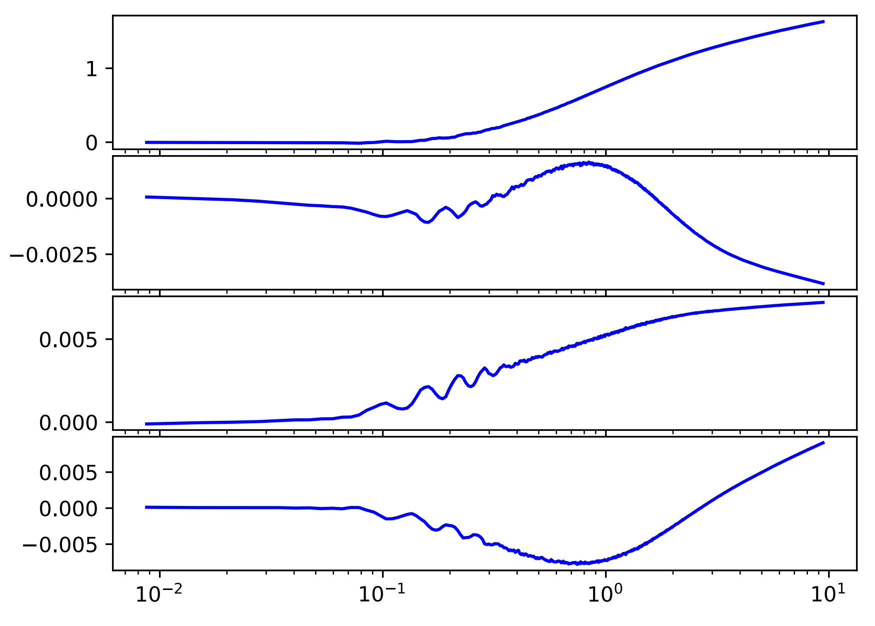


Stats $\longrightarrow$



PCA $\longrightarrow$

$$\begin{pmatrix} 1 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \end{pmatrix} \times$$



$$w_i(\Omega_b, \dots, A_s) = \sum_{\alpha_i \in \mathcal{A}^i} \eta_{\alpha_i}(\Omega_b, \dots, A_s)\Psi_{\alpha_i}$$

One PCE emulator per principal component (PC)!

EuclidEmulator specifications

EuclidEmulator1

(published)

- 6D feature space
- 11 PCs → 11 emulators
  (each with a 1D target)
- 100 training examples

EuclidEmulator2

(in prep.)

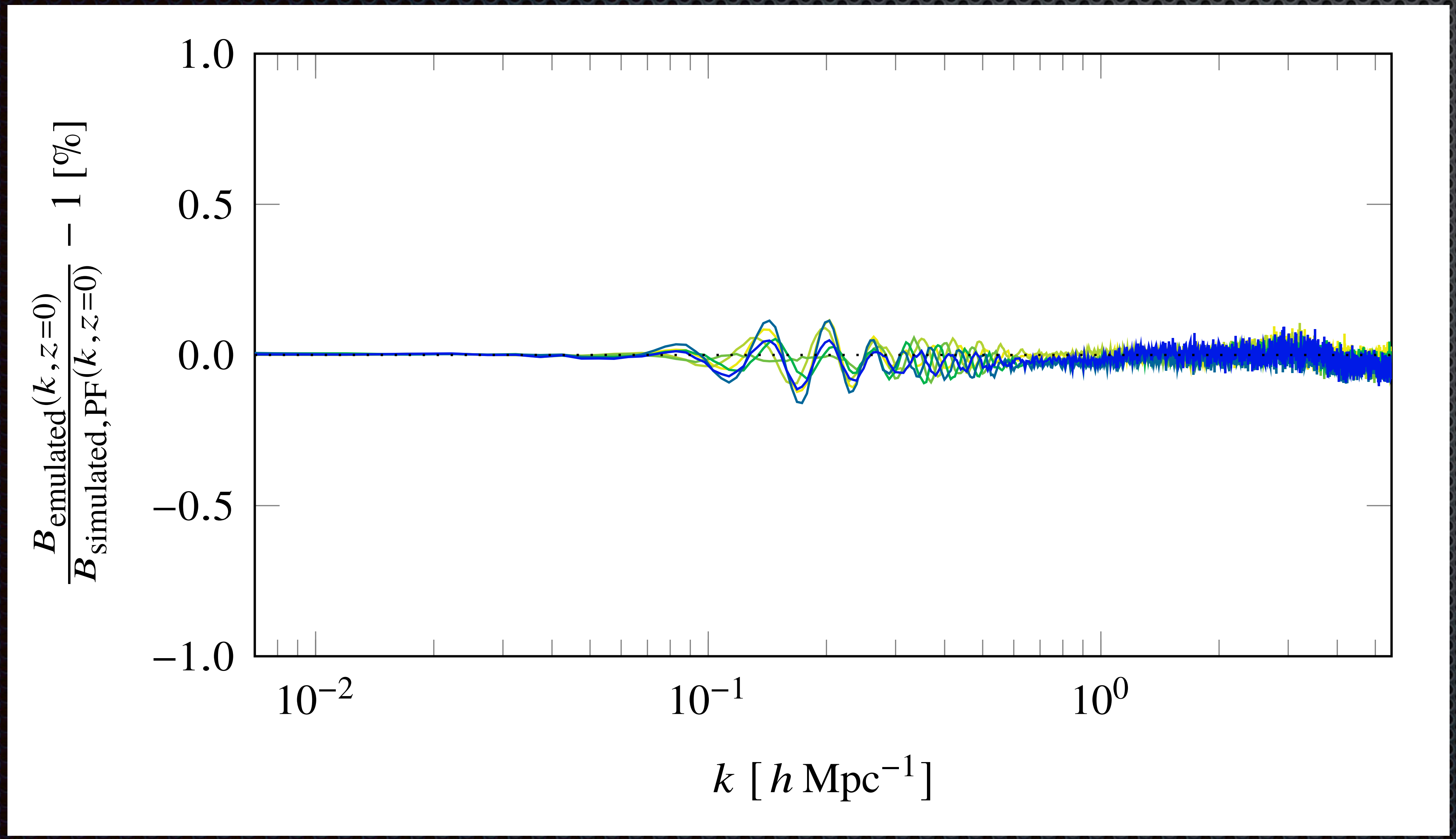
- 8D feature space
- 14 PCs → 14 emulators
  (each with a 1D target)
- 127 training examples

Performance (EuclidEmulator1)

Validation

36 validation examples

max validation error < 1%



Speed

$\mathcal{O}(10^{-5})$ CPU hours per evaluation

compared to

$\mathcal{O}(10^{4})$ CPU hours per simulation

Speed-up factor: 10^9

Exploiting the non-black-box nature

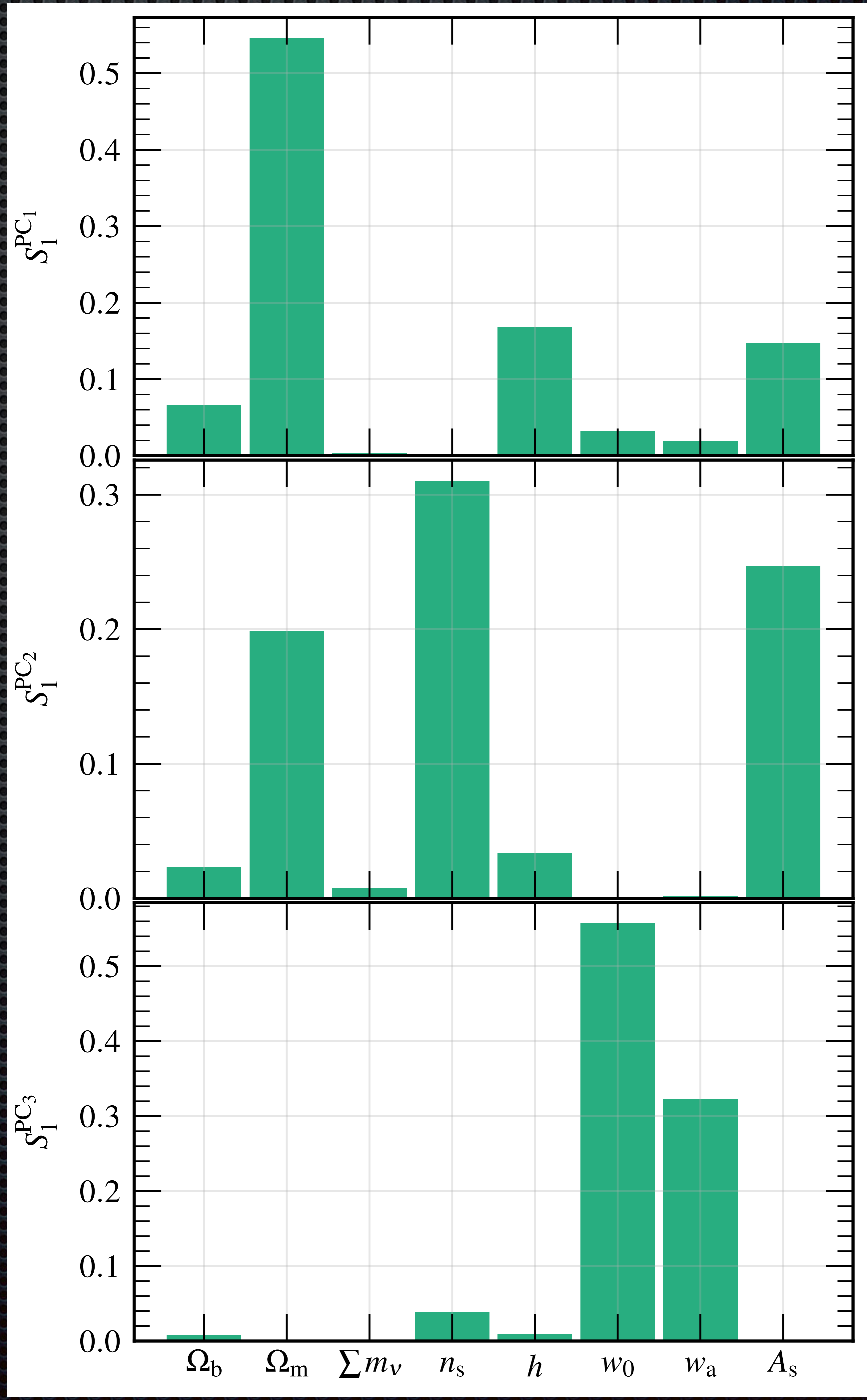
Sobol' analysis

Just by re-ordering expansion terms

we get a sensitivity analysis

(for free - no sampling required!)

→ **Very valuable e.g. for feature selection!**



Find me online…

https://github.com/miknab/EuclidEmulator

https://arxiv.org/abs/1809.04695

https://www.linkedin.com/in/mischa-knabenhans/

Thank you…

Further reading

M. Knabenhans et al. (2019), MNRAS, 484 (4), 5509-5529

M. Timpe et al. (2020, submitted)

S. Marelli and B. Sudret, ICVRAM 2014, University of Liverpool (UK), July 13–16, 2014, pp. 2554–2563

Torre et al. (2019), Journal of computational physics, 388, 601-623

D. Xiu and G. E. Karniadakis (2002), SIAM, 24, 619

B. Efron et al. (2004), The Annals of Statistics, 32 (2), 407-499