Goku: A 10-Parameter Simulation Suite for Cosmic Emulation (Project AST21005)

Frontera User Meeting 2024

Yanhui Yang August 5, 2024
Co-authors: Simeon Bird, Ming-Feng Ho

UCR NASA NSF TACC
TEXAS ADVANCED COMPUTING CENTER

Content

- **Background**
- **Methods**
- **Results (Preliminary)**
- **Summary**

Background

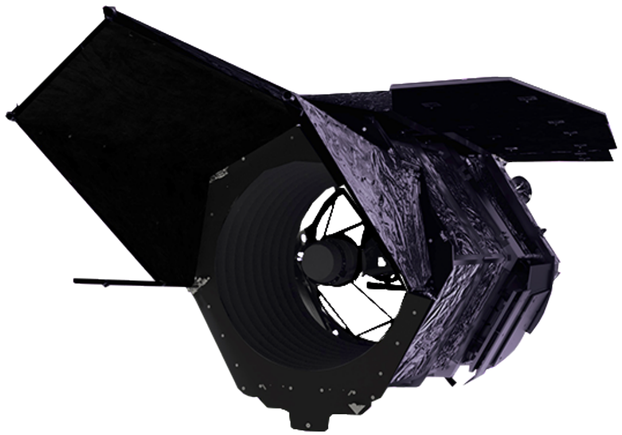
- Puzzles: dark matter, dark energy, the sum of the neutrino masses, the Hubble tension, etc. (Parameters: e.g., $\Omega_{\rm m}$, $\Omega_{\rm b}$, Σm_ν, h)

- Cosmological surveys: the Roman Space Telescope, Euclid, LSST, DESI, etc.

Background

- Puzzles: dark matter, dark energy, the sum of the neutrino masses, the Hubble tension, etc. (Parameters: e.g., $\Omega_{\rm m}$, $\Omega_{\rm b}$, Σm_ν, h)

- Cosmological surveys: the Roman Space Telescope, Euclid, LSST, DESI, etc.



Roman Space Telescope

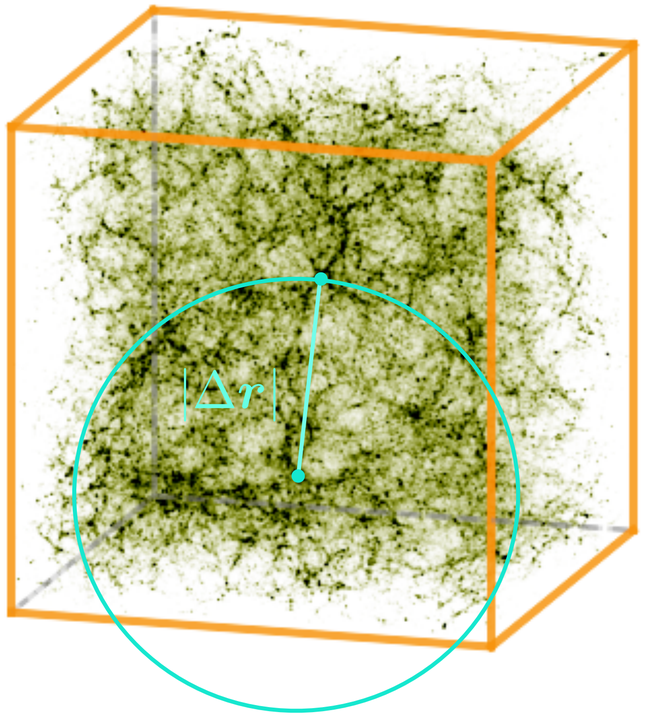
→ $\sim 10^9$ galaxies will be observed

↓

Summary statistics (high precision)

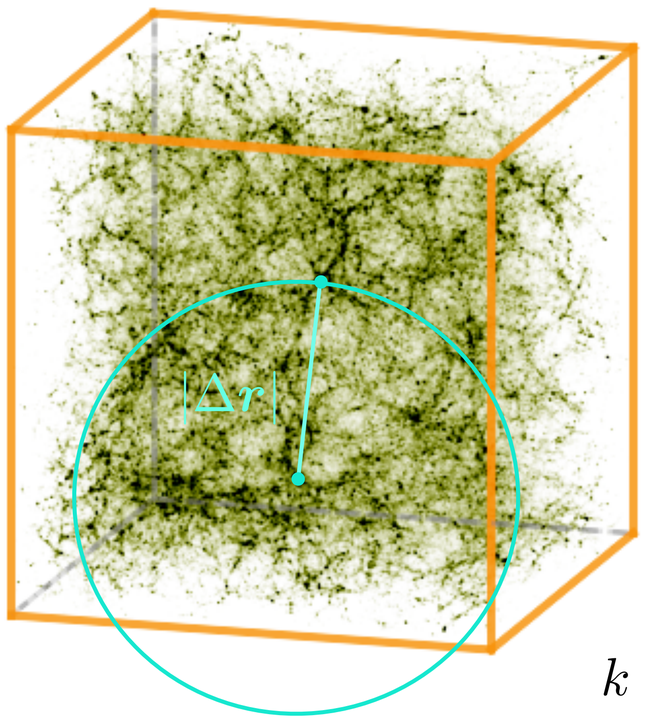
Background

- The matter power spectrum: how "clumpy" the Universe is at different scales (Fourier transform of the two-point correlation function of the overdensity field)



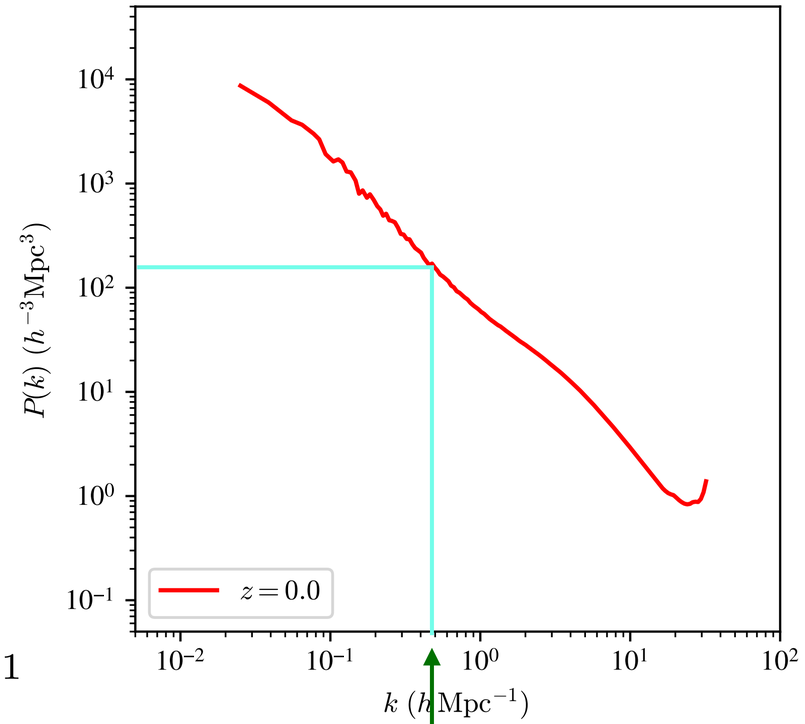
Background

- The matter power spectrum: how "clumpy" the Universe is at different scales (Fourier transform of the two-point correlation function of the overdensity field)



$$|\Delta \boldsymbol{r}| = 130\,\mathrm{Mpc}/h$$

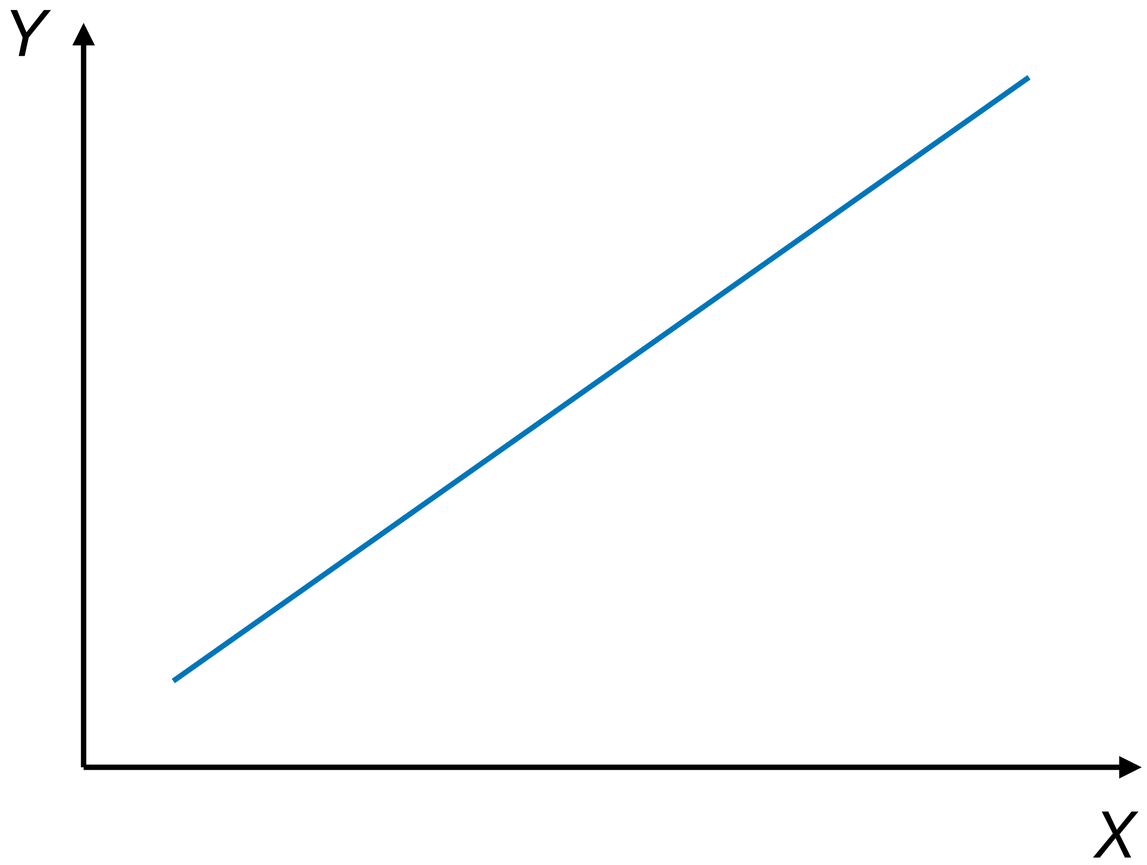
$$k = \frac{2\pi}{|\Delta \boldsymbol{r}|} \approx 0.048\,h\,\mathrm{Mpc}^{-1}$$



Background

- Cosmological inference: Bayesian methods $$P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model})P(\text{model})}{P(\text{data})}$$

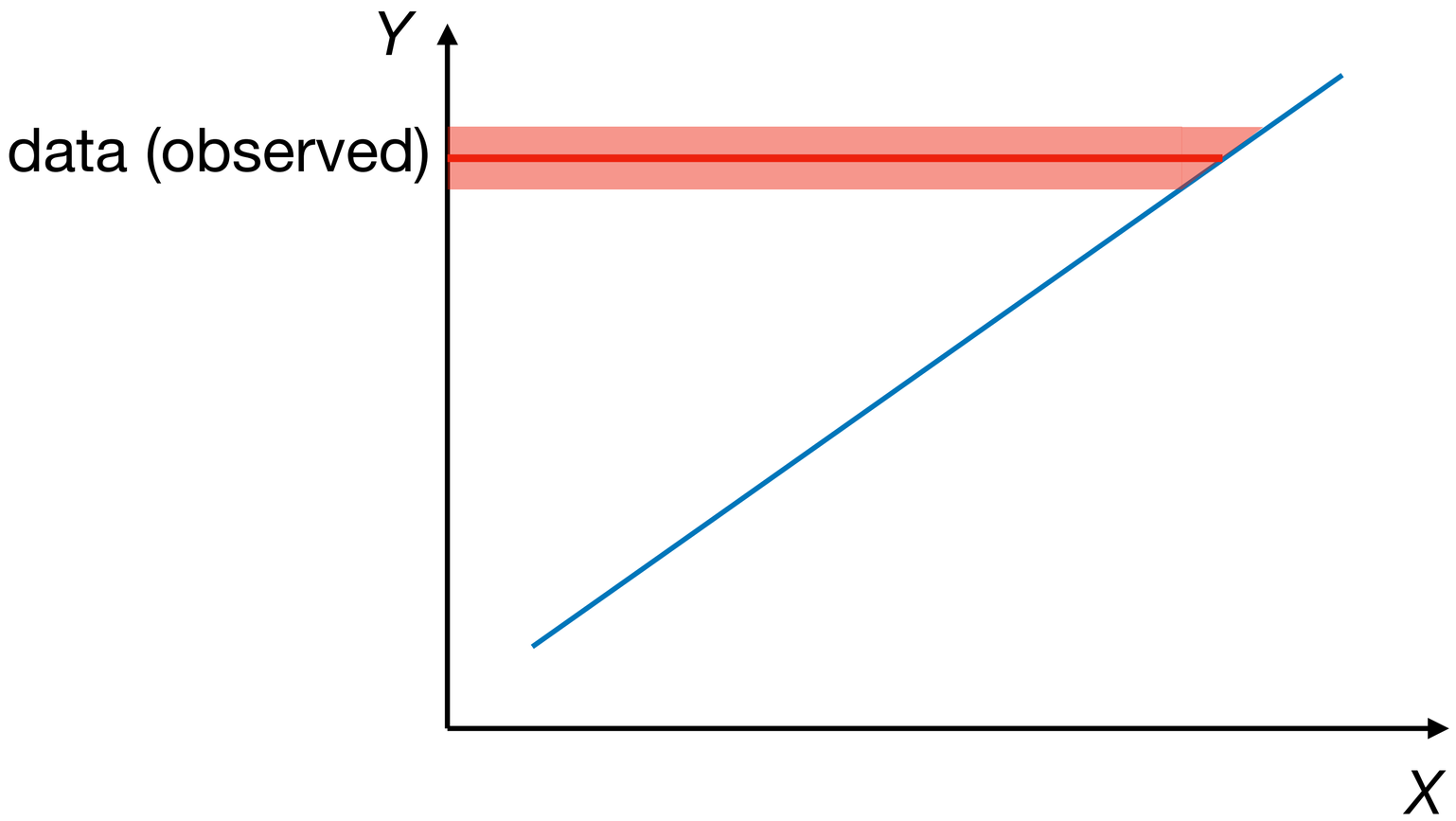
  Basic idea: an analogy, suppose we know *Y=f(X)*



Background

- Cosmological inference: Bayesian methods $P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model})P(\text{model})}{P(\text{data})}$

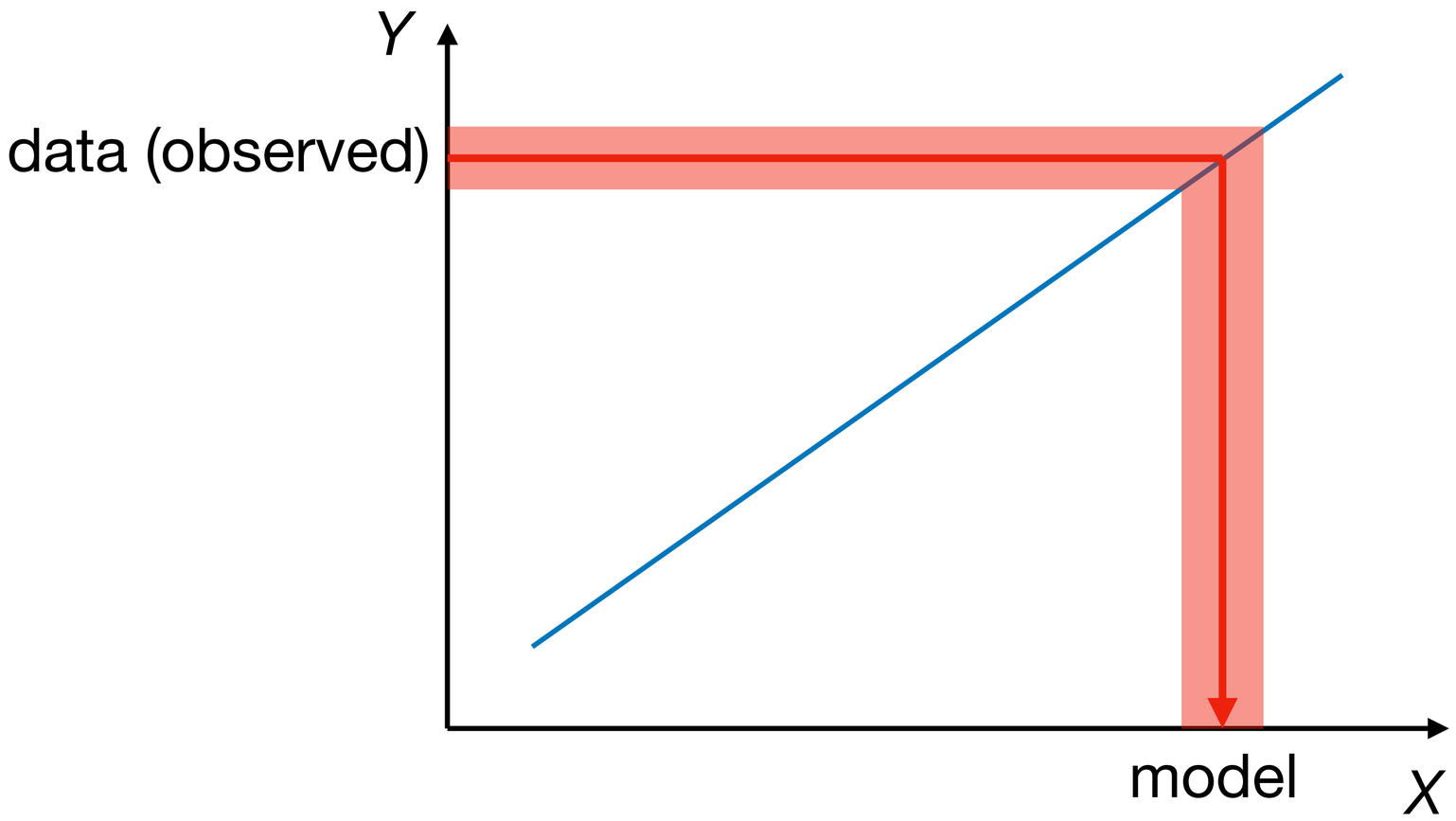
  Basic idea: an analogy, suppose we know *Y=f(X)*



Background

- Cosmological inference: Bayesian methods $P(\mathrm{model}|\mathrm{data}) = \dfrac{P(\mathrm{data}|\mathrm{model})P(\mathrm{model})}{P(\mathrm{data})}$

  Basic idea: an analogy, suppose we know *Y=f(X)*

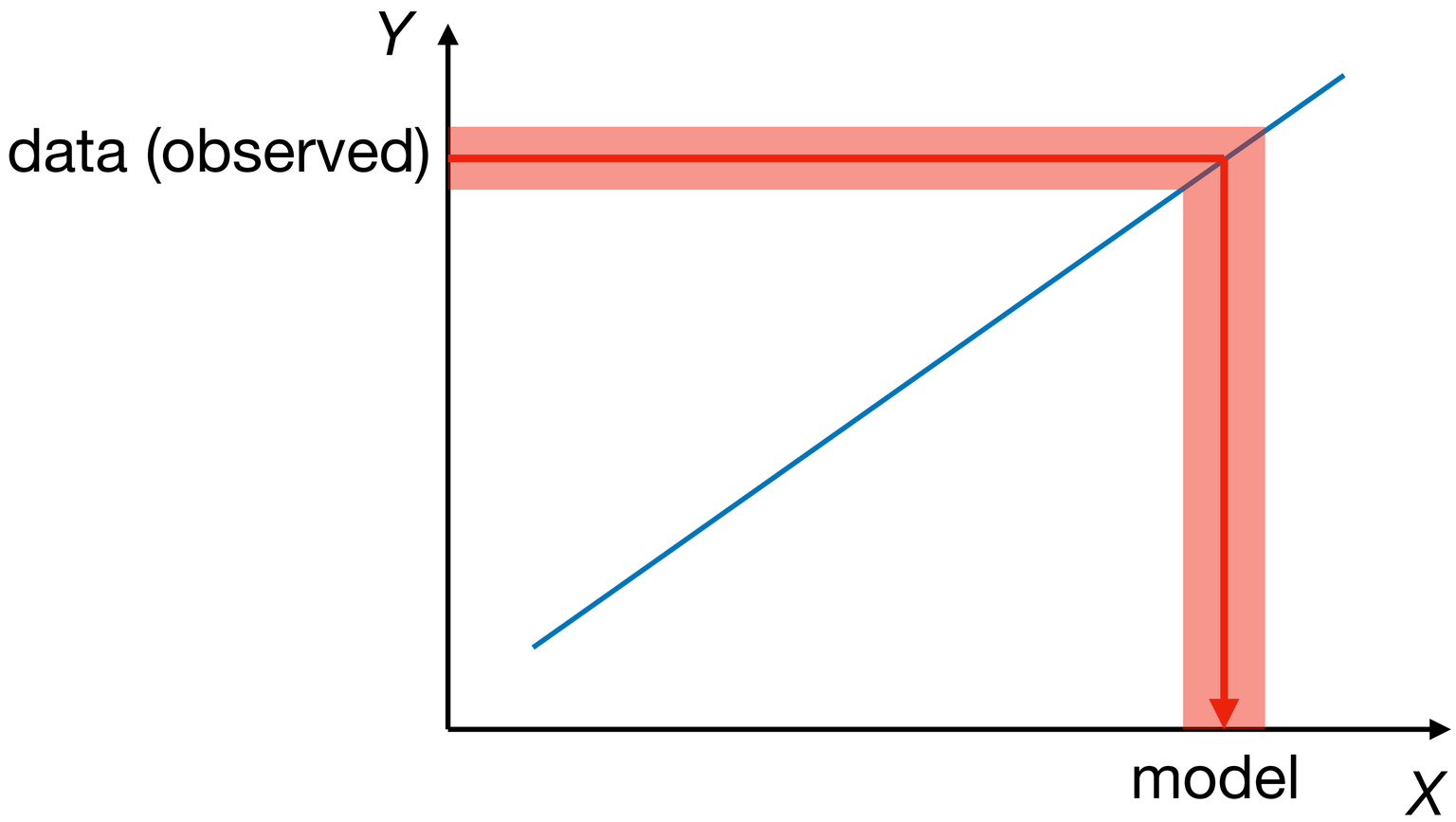


Background

- Cosmological inference: Bayesian methods

$$P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model})P(\text{model})}{P(\text{data})}$$

Basic idea: an analogy, suppose we know *Y=f(X)*



In practice, *X* and *Y* can be vectors!

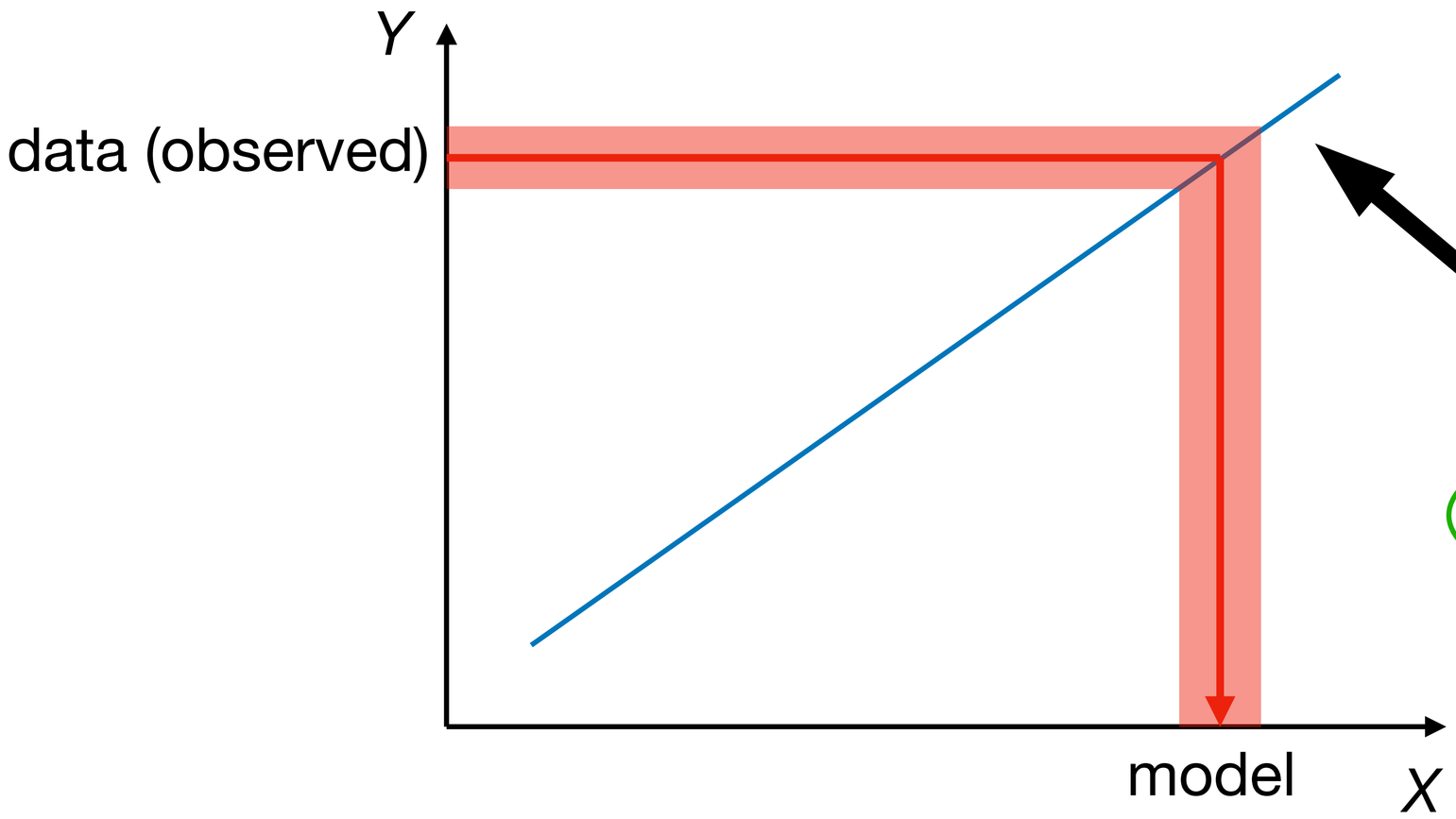
And *f* can be complicated!

Background

- Cosmological inference: Bayesian methods

$$P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model})P(\text{model})}{P(\text{data})}$$

Basic idea: an analogy, suppose we know *Y=f(X)*



In practice, *X* and *Y* can be vectors!

And *f* can be complicated!

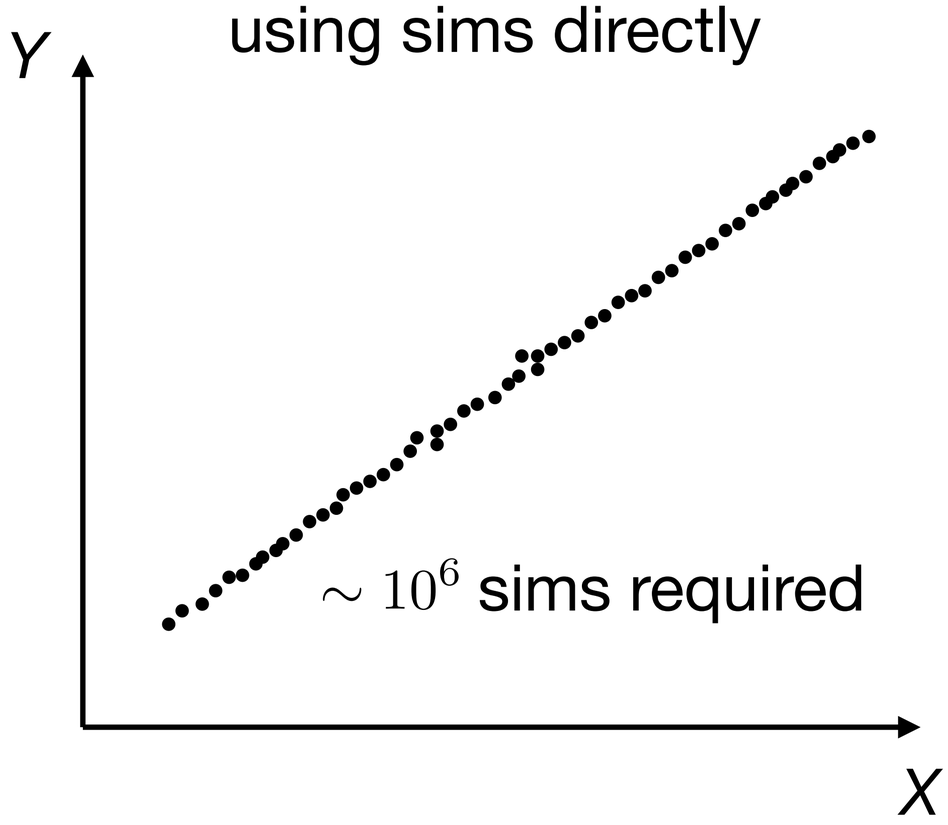
Focus

How do we get the dependence *f*?

Background

by N-body simulations

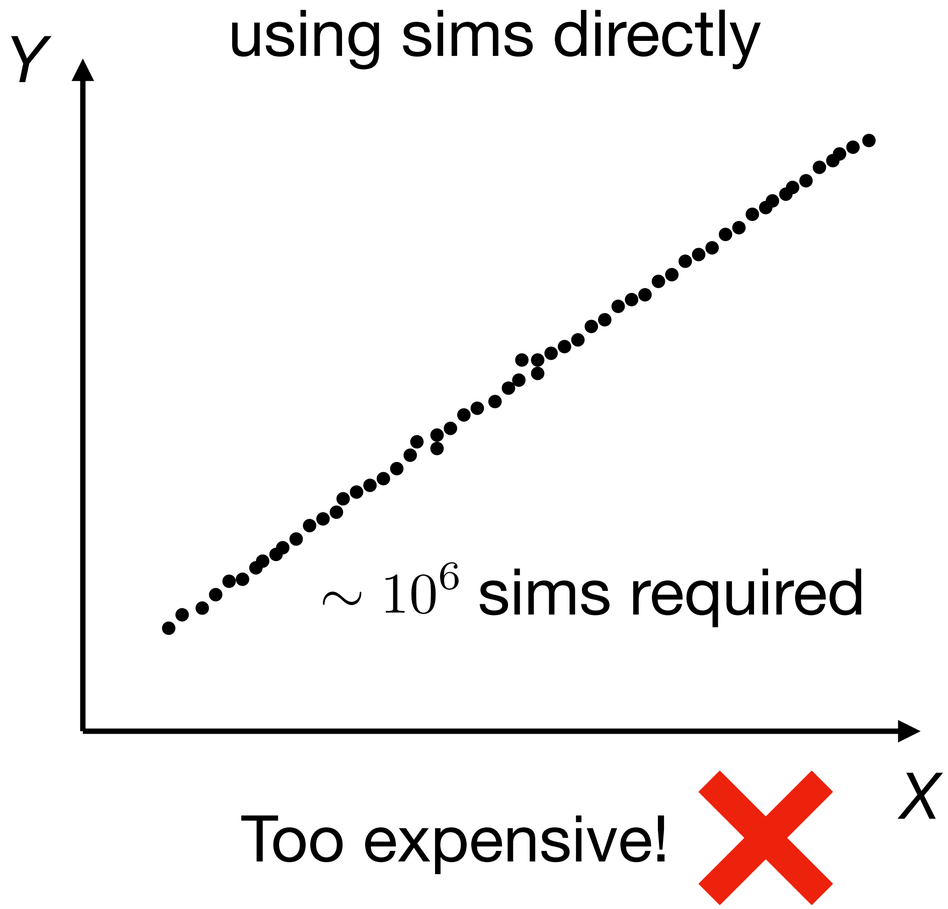
- Theoretical predictions: *X* (cosmological parameters)→ *Y* (matter power)



Background

by N-body simulations

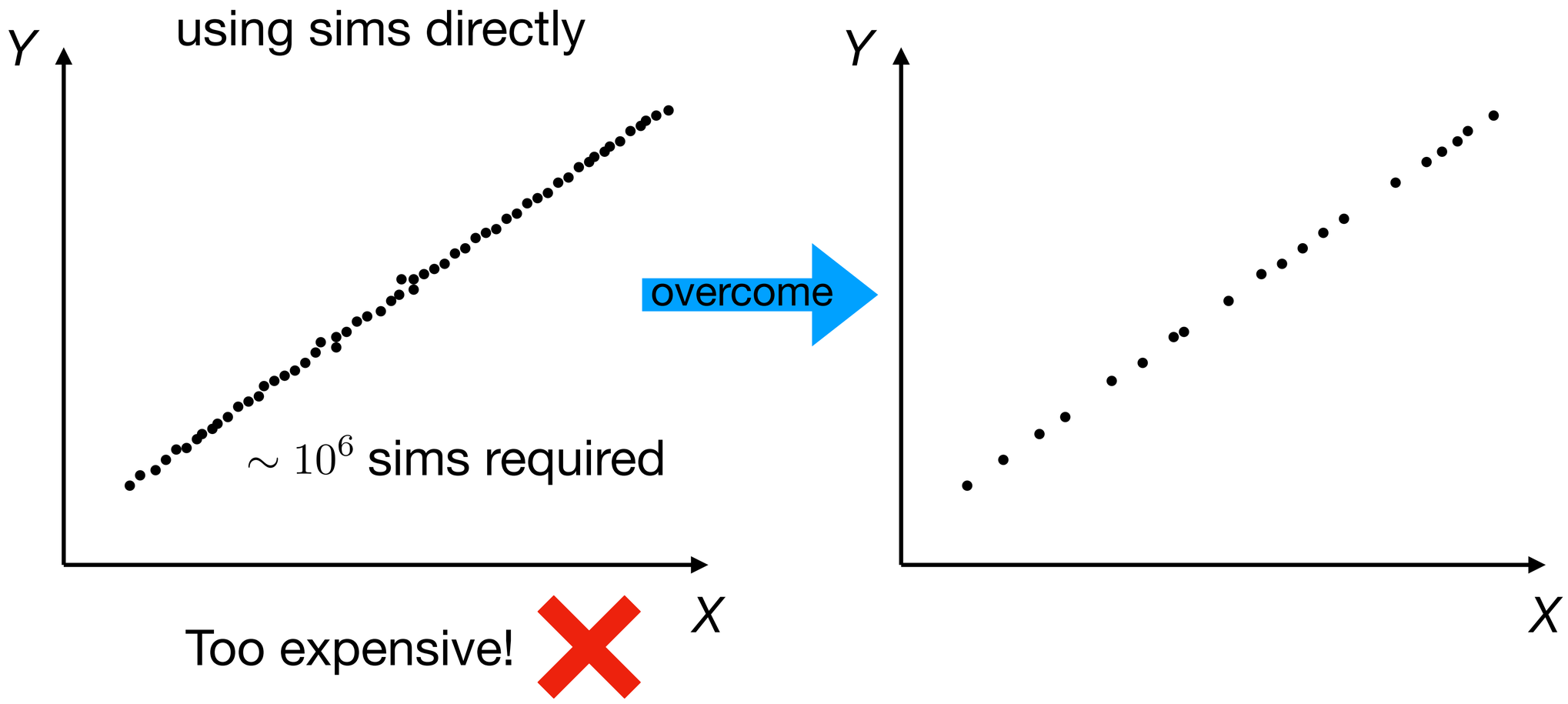
- Theoretical predictions: *X* (cosmological parameters)→ *Y* (matter power)



Background

by N-body simulations

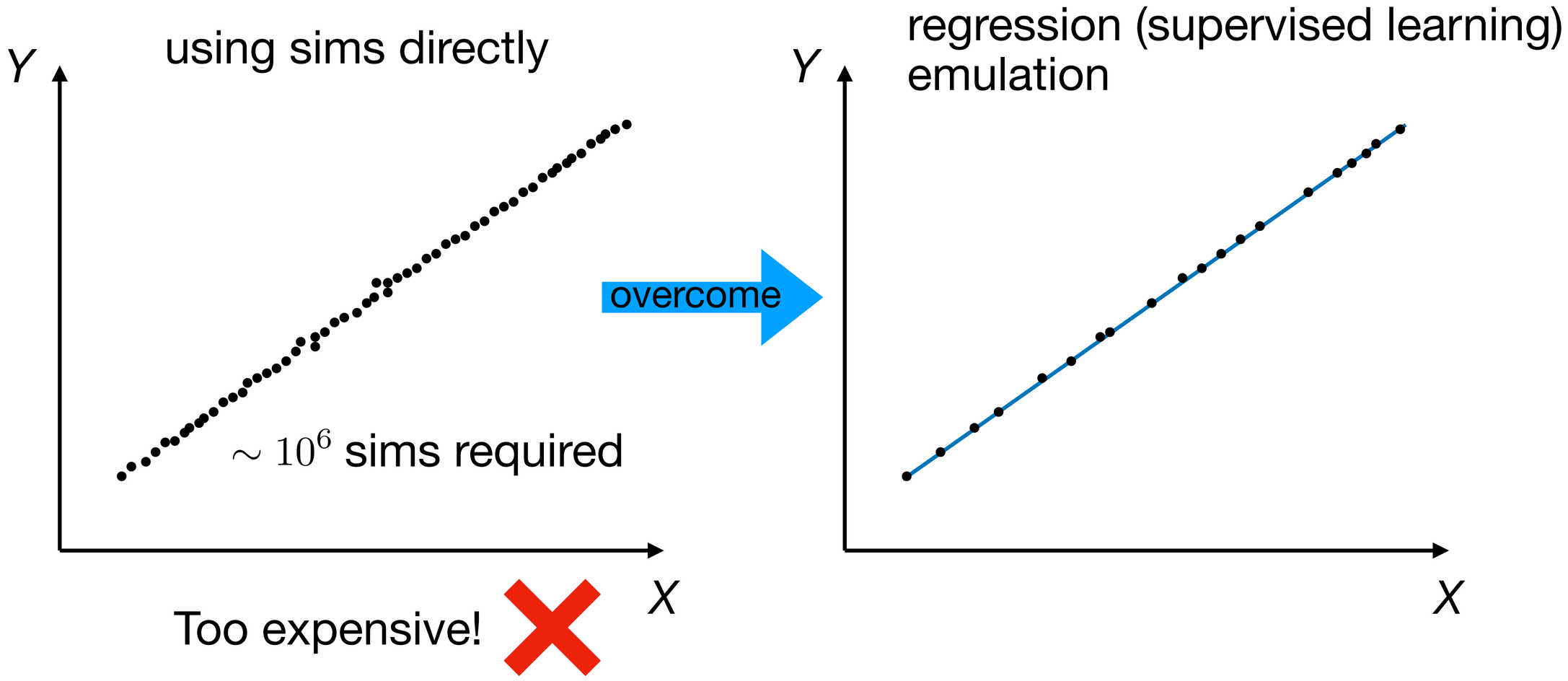
- Theoretical predictions: *X* (cosmological parameters)→ *Y* (matter power)



Background

by N-body simulations

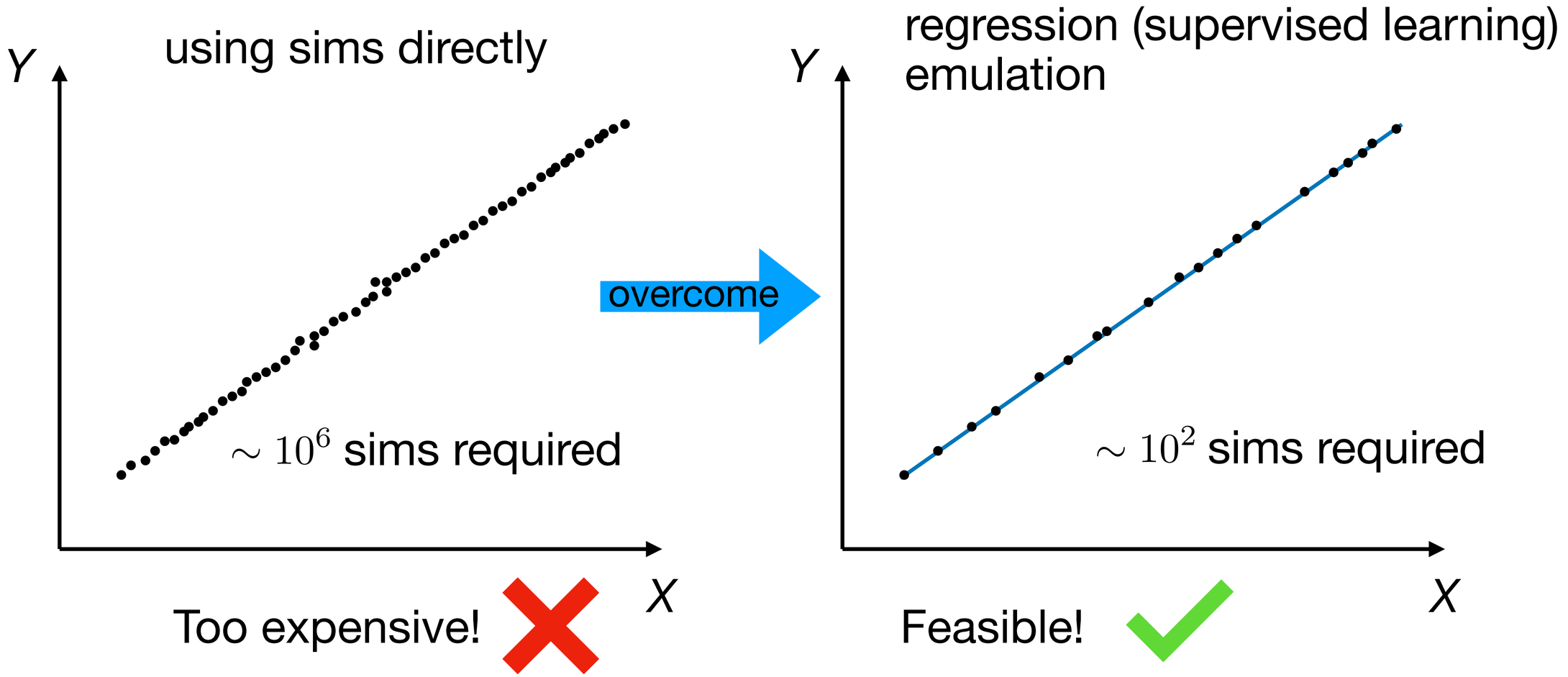
- Theoretical predictions: *X* (cosmological parameters)→ *Y* (matter power)



Background

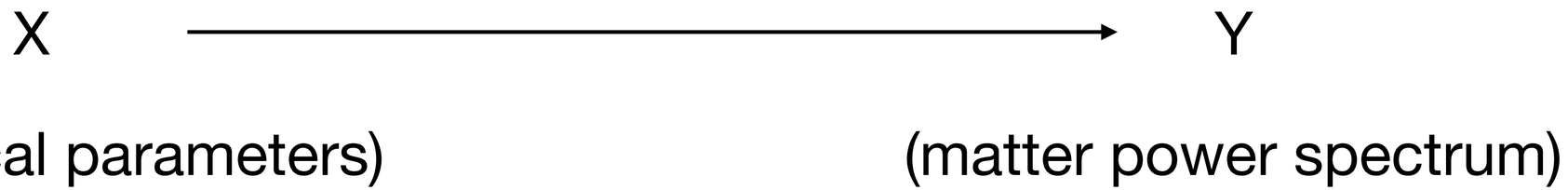
by N-body simulations

- Theoretical predictions: *X* (cosmological parameters)→ *Y* (matter power)



Background

- We are running a suite of N-body simulations
- and will build an emulator based on the simulations that can predict the matter power spectrum at the percent-level accuracy



Methods

- Cosmological parameters: 10-dimensional parameter space
  - Sampling technique: Sliced Latin Hypercube Design (SLHD)

	Parameter	Definition/Description	Lower bound	Upper bound
	Ω_{m}	the total matter density parameter (DM and baryons)	0.22	0.40
	Ω_{b}	the total baryon density parameter	0.040	0.055
	h	the Hubble parameter	0.60	0.76
	A_{s}	the primordial perturbation amplitude	1×10^{-9}	3×10^{-9}
	n_{s}	the primordial spectral index	0.8	1.1
extensions	w_0	the parameter of the time-independent part of the DE equation of state	-1.30	0.25
extensions	w_a	the parameter of the time-dependent part of the DE equation of state	-3.0	0.5
extensions	$\sum m_\nu$	the sum of the neutrino masses	0.0	0.6 eV
extensions	N_{eff}	the effective number of neutrinos	2.2	4.5
extensions	α_{s}	the running of the scalar spectral index	-0.05	0.05

Methods

- Cosmological parameters: 10-dimensional parameter space
  - Sampling technique: Sliced Latin Hypercube Design (SLHD)

	Parameter	Definition/Description	Lower bound	Upper bound
	$\Omega_{\rm m}$	the total matter density parameter (DM and baryons)	0.22	0.40
	$\Omega_{\rm b}$	the total baryon density parameter	0.040	0.055
	h	the Hubble parameter	0.60	0.76
	$A_{\rm s}$	the primordial perturbation amplitude	1×10^{-9}	3×10^{-9}
	$n_{\rm s}$	the primordial spectral index	0.8	1.1
extensions	w_0	the parameter of the time-independent part of the DE equation of state	-1.30	0.25
extensions	w_a	the parameter of the time-dependent part of the DE equation of state	-3.0	0.5
extensions	$\sum m_\nu$	the sum of the neutrino masses	0.0	0.6 eV
extensions	$N_{\rm eff}$	the effective number of neutrinos	2.2	4.5
extensions	$\alpha_{\rm s}$	the running of the scalar spectral index	-0.05	0.05

- N-body simulations: MP-Gadget (MPI + OpenMP threads)
  - goal: 3000^3 particles (such high resolution is required to make full use of data) in a box of $(1000\,{\rm Mpc}/h)^3$ (large box) ⟶ a big simulation!

Methods

- The role of Frontera:
  - essential for running simulations:
    - one aforementioned simulation takes 18 hours on 256 nodes (14336 cores) on Frontera
    - small systems: e.g., on UCR's HPCC: 256 cores accessible, a single run would take 42 days to finish!
    - this project: equivalent to ~30 runs, would take 3.5 years on HPCC, while Frontera makes it achievable within 1 month!

Methods

- The role of Frontera:
  - necessary for visualizing simulations:
    - the visualization requires ~1.5 TB memory (10 nodes are used in practice)
    - not possible on small systems: e.g., UCR's HPCC 1 TB accessible (per user), my laptop 16 GB

Frontera enables our science goals!

Methods

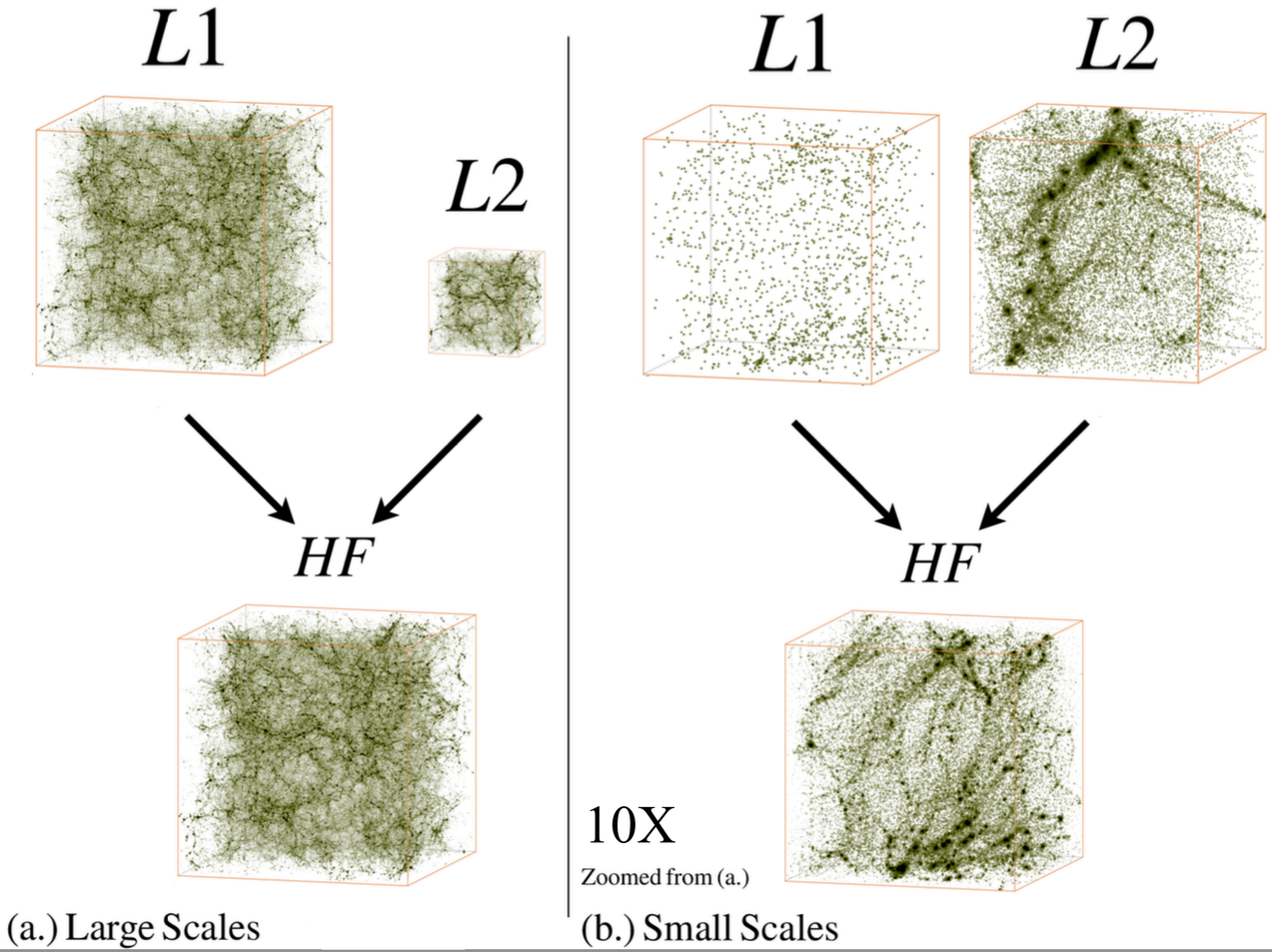
- Emulation based on simulations:
  - Traditional emulator: single-fidelity (# of particles, box size)
    - e.g., 8D emulator: EuclidEmulator2 (Euclid collaboration 2020)
    - efficient but still needs a large number (>100) of high-resolution sims and hard to expand the parameter space to higher dimensions

Methods

- Emulation based on simulations:
  - Traditional emulator: single-fidelity (# of particles, box size)
    - e.g., 8D emulator: EuclidEmulator2 (Euclid collaboration 2020)
    - efficient but still needs a large number (>100) of high-resolution sims and hard to expand the parameter space to higher dimensions
  - MF-Box, a multi-fidelity emulation framework based on Gaussian process regression; builds the training set on simulations of different fidelities (Ho, Bird et al. 2023)
    - further reduces the computational budget! (a low-fidelity simulation is much less expensive than a high-fidelity one, while can provide a fairly good prediction on certain scales)

Methods

- MF-Box emulation: tree structure (Ho, Bird et al. 2023), two low-fidelity (LF) nodes (information sources) and one high-fidelity (LF) node



In our work:

L1: 750^3 particles
$1000\,\mathrm{Mpc}/h$ box

L2: 750^3 particles
$250\,\mathrm{Mpc}/h$ box

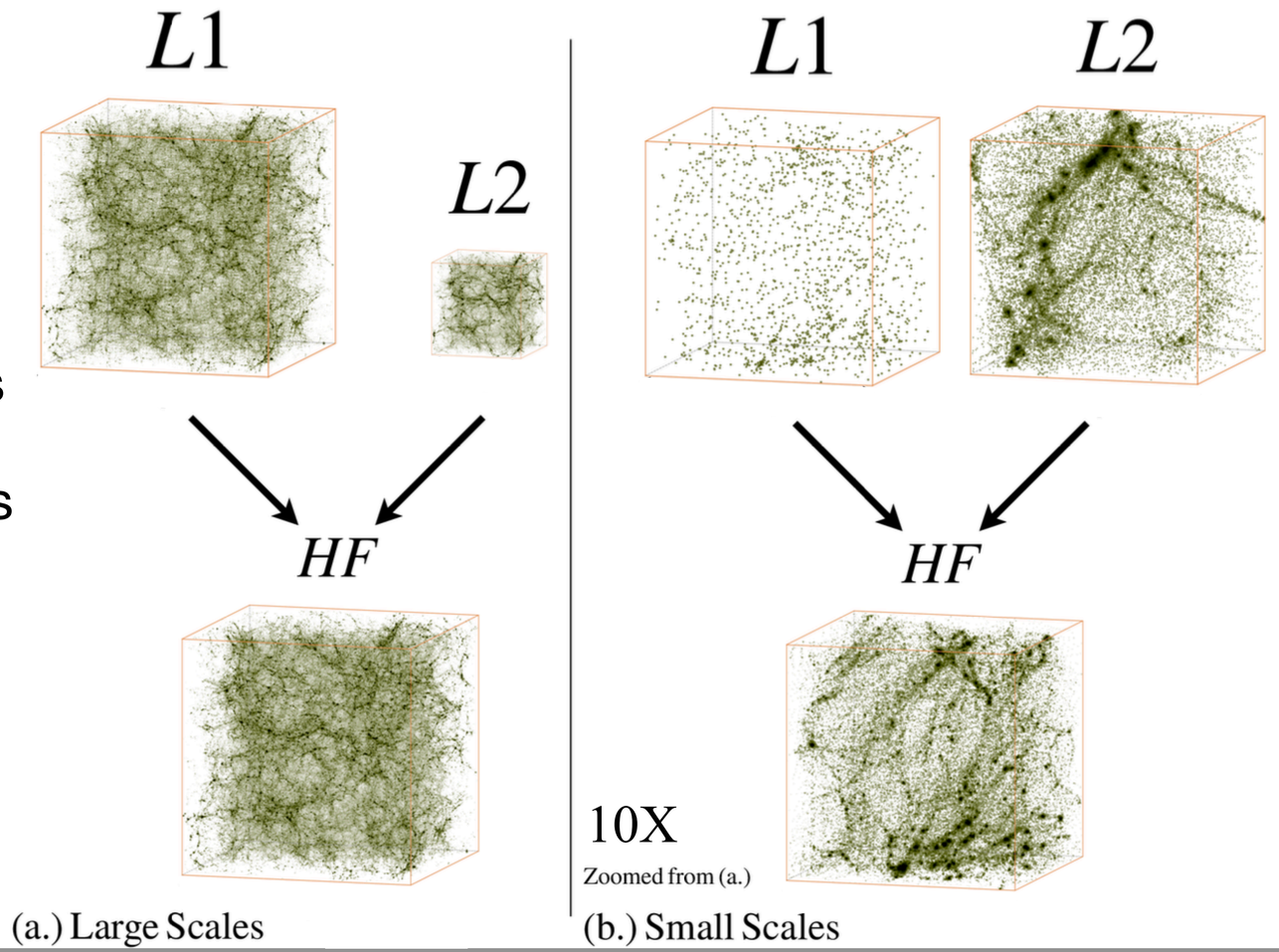
HF: 3000^3 particles
$1000\,\mathrm{Mpc}/h$ box

Methods

- MF-Box emulation: tree structure (Ho, Bird et al. 2023), two low-fidelity (LF) nodes (information sources) and one high-fidelity (LF) node

L1: large scales

L2: small scales



In our work:

L1: 750^3 particles
$1000\,\mathrm{Mpc}/h$ box

L2: 750^3 particles
$250\,\mathrm{Mpc}/h$ box

HF: 3000^3 particles
$1000\,\mathrm{Mpc}/h$ box

Methods

- Optimization of the computational budget
  - n_{L} pairs of LF sims, and n_{H} HF sims: total cost
    - $C(n_{\mathrm{L}}, n_{\mathrm{H}}) = C_{\mathrm{L}} n_{\mathrm{L}} + C_{\mathrm{H}} n_{\mathrm{H}}$

Methods

- Optimization of the computational budget
  - n_{L} pairs of LF sims, and n_{H} HF sims: total cost
    - $C(n_{\mathrm{L}}, n_{\mathrm{H}}) = C_{\mathrm{L}} n_{\mathrm{L}} + C_{\mathrm{H}} n_{\mathrm{H}}$
  - Target accuracy $\sim 1\%$
    - constraint: $\Phi(n_{\mathrm{L}}, n_{\mathrm{H}}) = \Phi_{\mathrm{target}}$ (error function will be detailed in our paper)

the Lagrange multiplier method

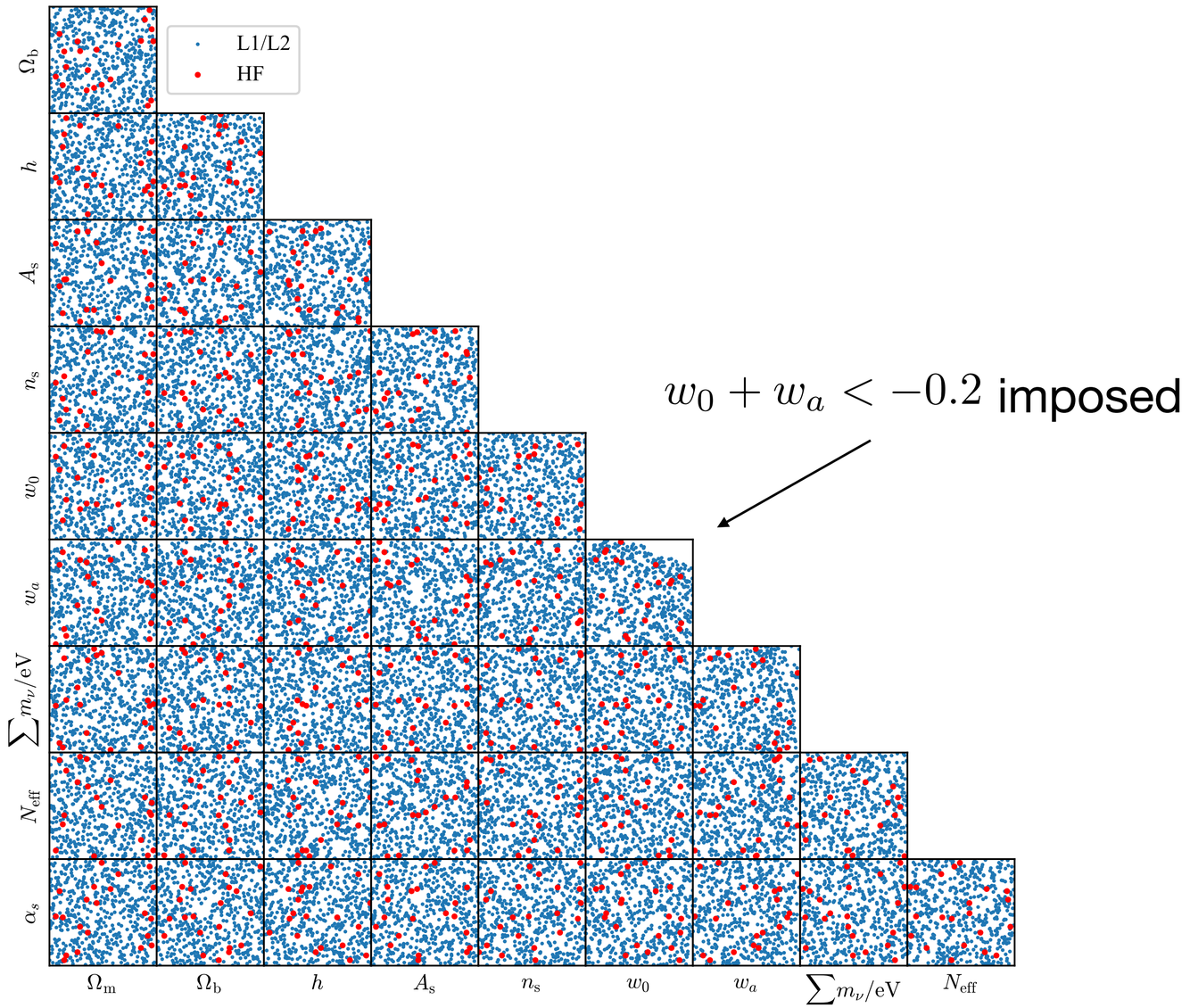
The optimal $(n_{\mathrm{L}}, n_{\mathrm{H}})$ that minimizes C

Results (Preliminary)

- # of sims:
  - 564 pairs of LF simulations and 21 HF simulations
- computational cost $\sim 1.1 \times 10^5$ node hours (Frontera)
  - In contrast, a single-fidelity emulator based on 564 HF simulations would consume $\sim 2.1 \times 10^6$ node hours (much more computationally expensive!)

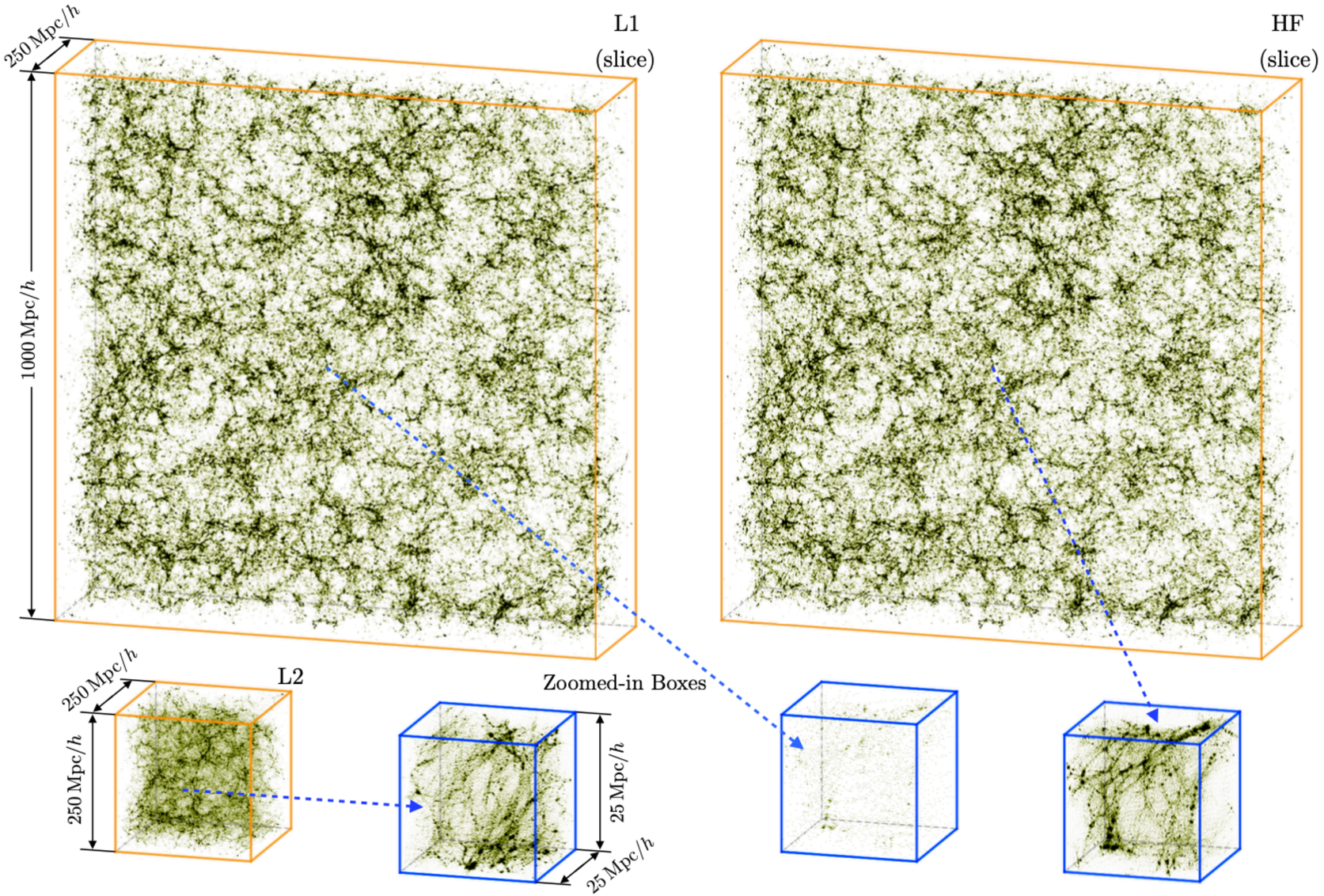
Results (Preliminary)

- Sampling of cosmologies:
  - a good space-filling design
  - HF points are selected from LF points



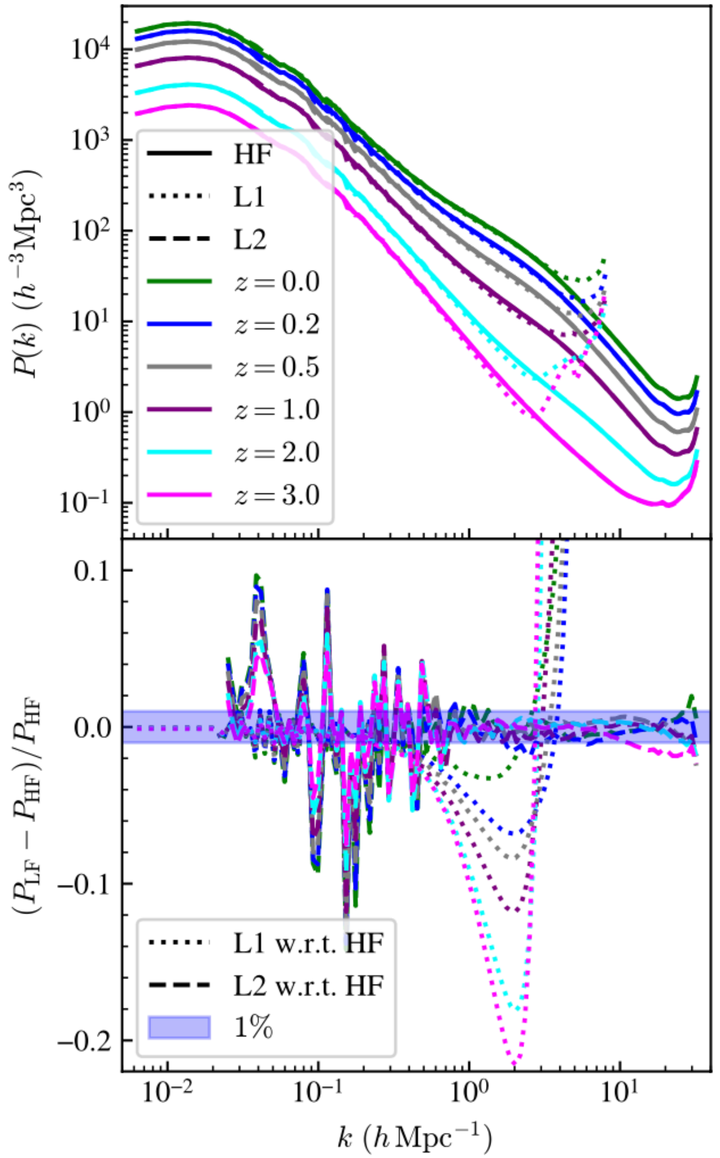
Results (Preliminary)

- Case study:
  - an HF sim and its LF counterparts
  - L1 almost the same as HF on large scales
  - L2 is accurate on small scales



Results (Preliminary)

- Case study:
  - LF and HF matter power spectra
  - L1 and L2 approximate the matter power spectrum very well at *large* and *small* scales respectively



Summary

- Goku: 564 pairs of LF sims and 21 HF sims
- 10D emulator for the matter power spectrum: GokuEmu (the highest-dimensional emulator at present)

constrain cosmological models in an unprecedentedly high-dimensional parameter space, using data from the Roman Space Telescope!

- Confirms: MF-Box significantly reduces the computational cost of building a cosmological emulator

Summary

- Goku: 564 pairs of LF sims and 21 HF sims
- 10D emulator for the matter power spectrum: GokuEmu (the highest-dimensional emulator at present)

constrain cosmological models in an unprecedentedly high-dimensional parameter space, using data from the Roman Space Telescope!

- Confirms: MF-Box significantly reduces the computational cost of building a cosmological emulator

Thank you !