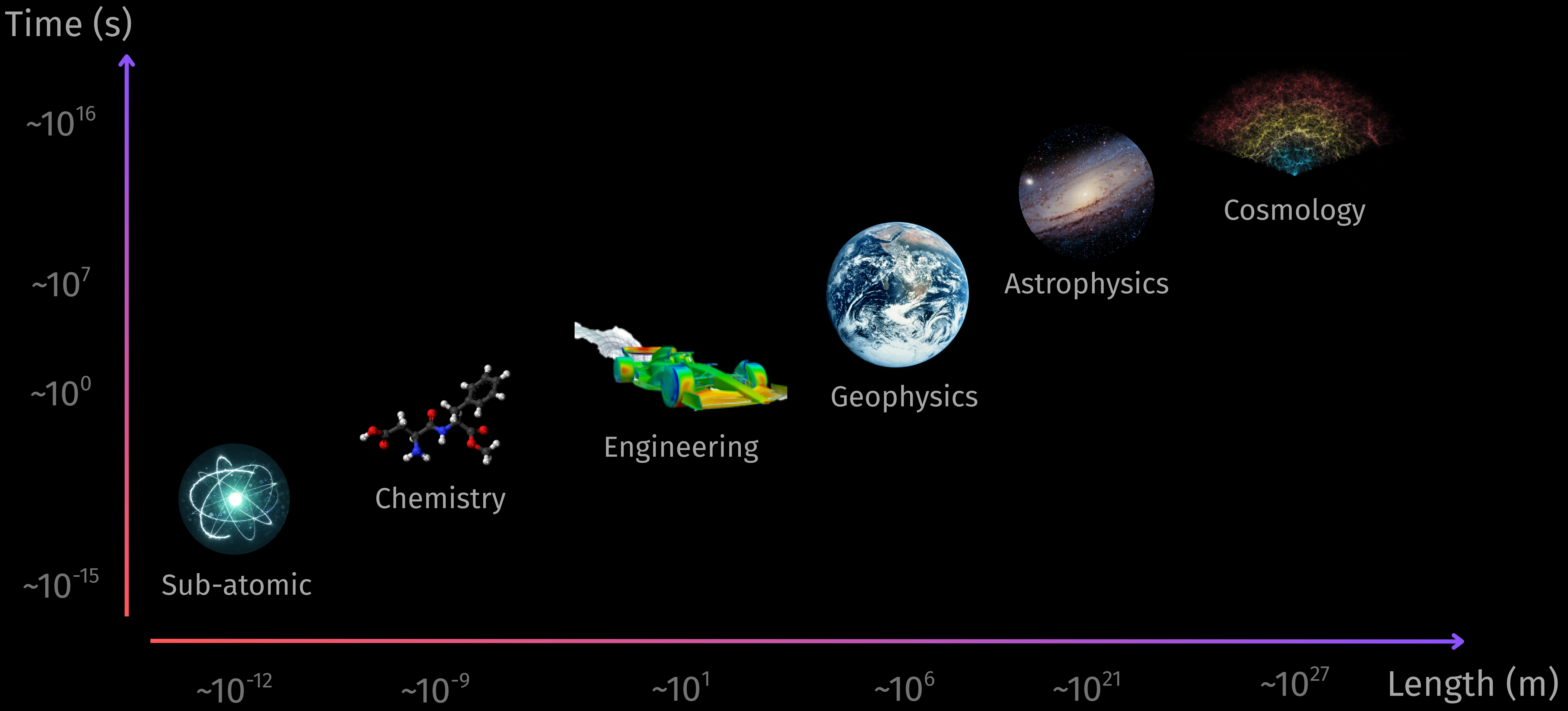
The Alan Turing Institute
SciAI UCL

Greener Science with AI:
Sustainable Inference of Physical Systems

Jason McEwen
Mission Director for Fundamental Research

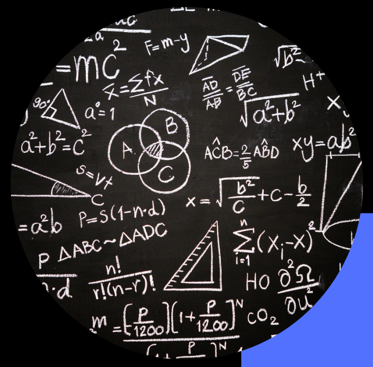
Centre for Intelligent Sustainable Computing Symposium
Queen's University Belfast, August 2025

Spatial and temporal scales of physical systems



Pillars of science

1st Pillar:
Experimental



2nd Pillar:
Theoretical

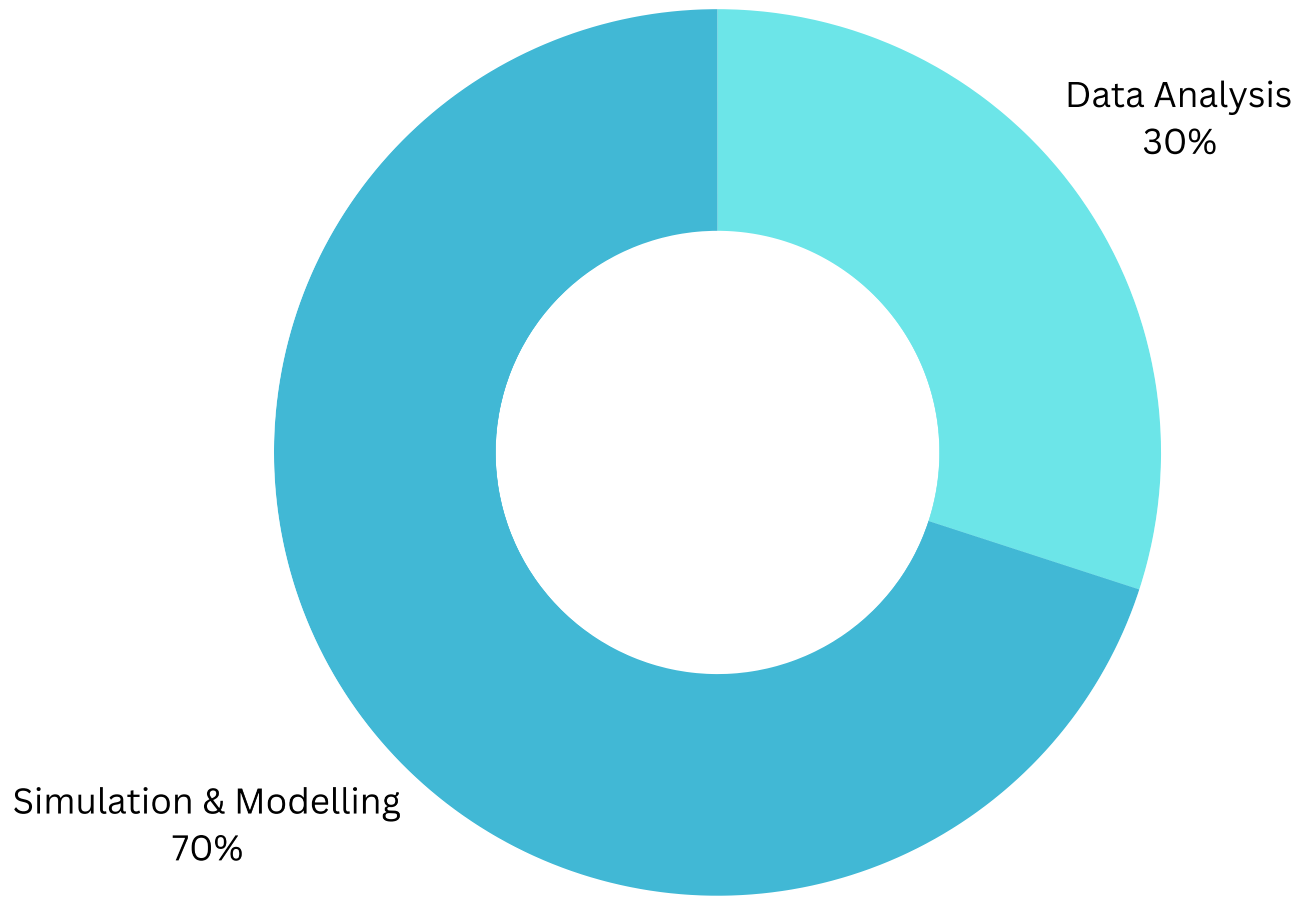
3rd Pillar:
Simulation

4th Pillar:
Data-Driven

~1500 ~1700 ~1950 ~2010 t

Computational cost of simulation

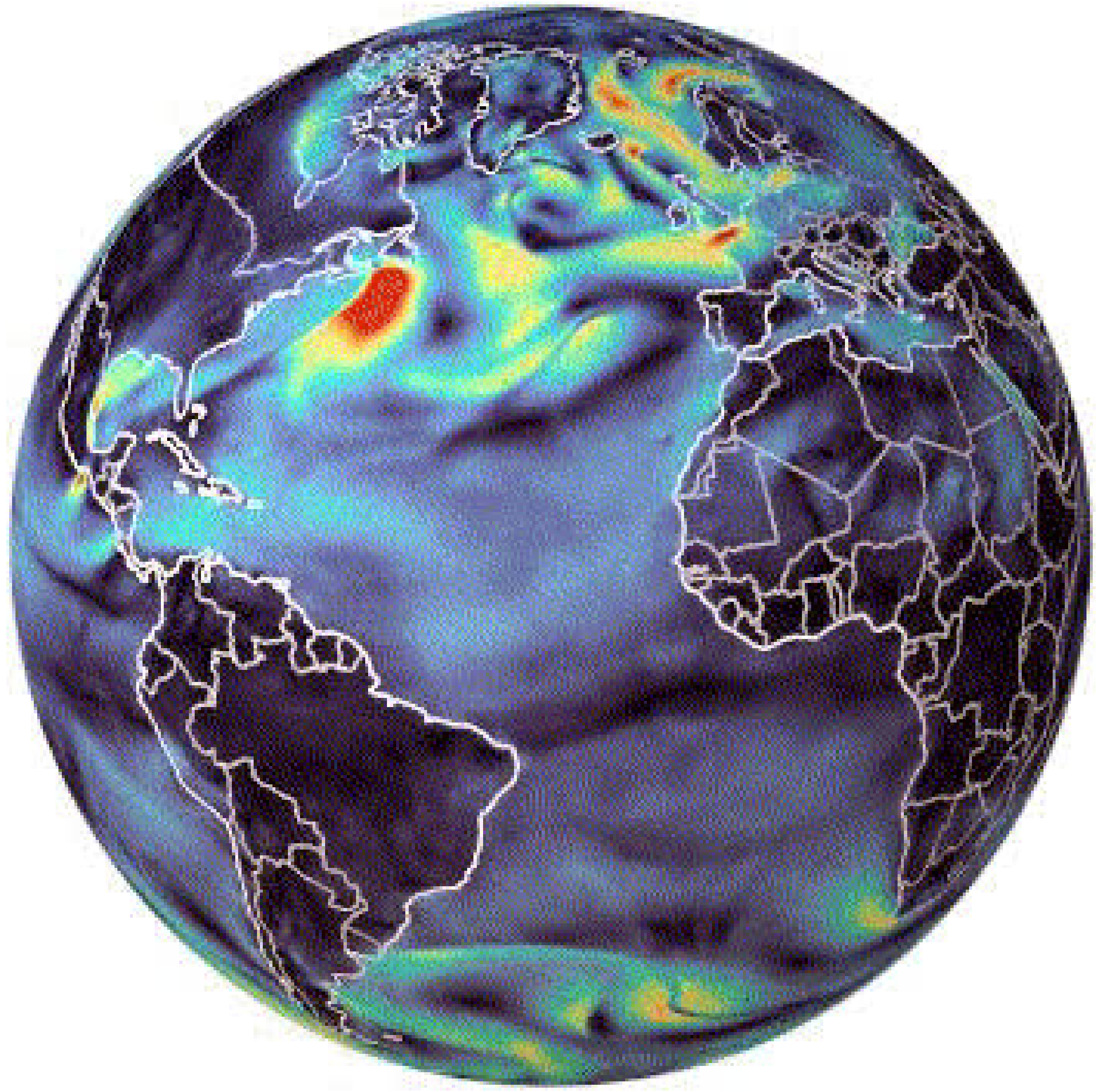
- **Modelling & simulation account for ~70% of high-performance computing (HPC)** usage
- Data analysis accounts for ~30%



Computational cost of simulation: climate simulation example

CMIP6: Coupled Model Intercomparison Project 6

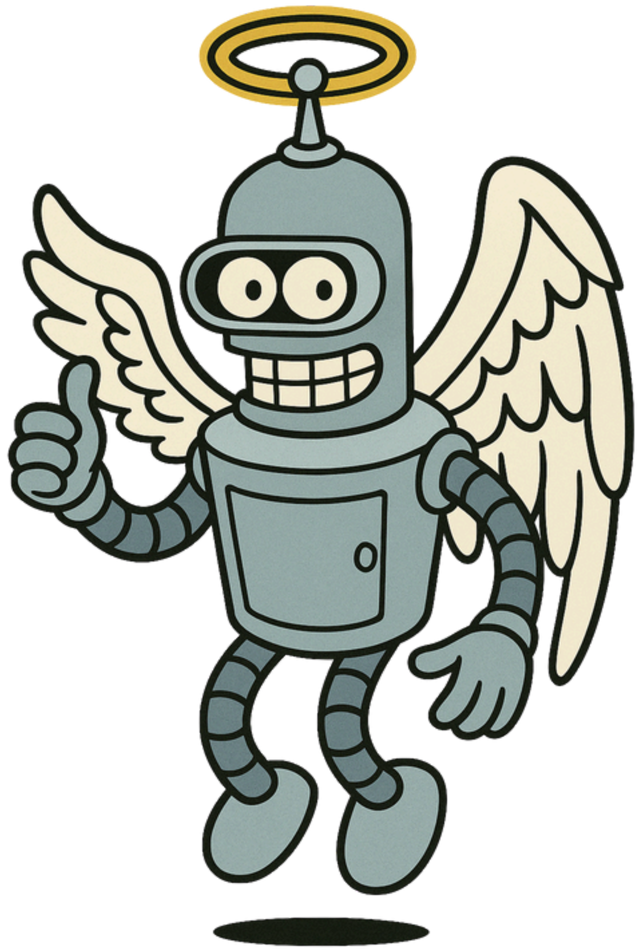
- 40,000 years of climate simulations
- ≥1 billion core hours
- 40PB of data
- Carbon footprint of 1692t CO_2 equivalent (Acosta et al. 2024)
  ≈ 6.2 million car miles ≈ driving around Earth 250x



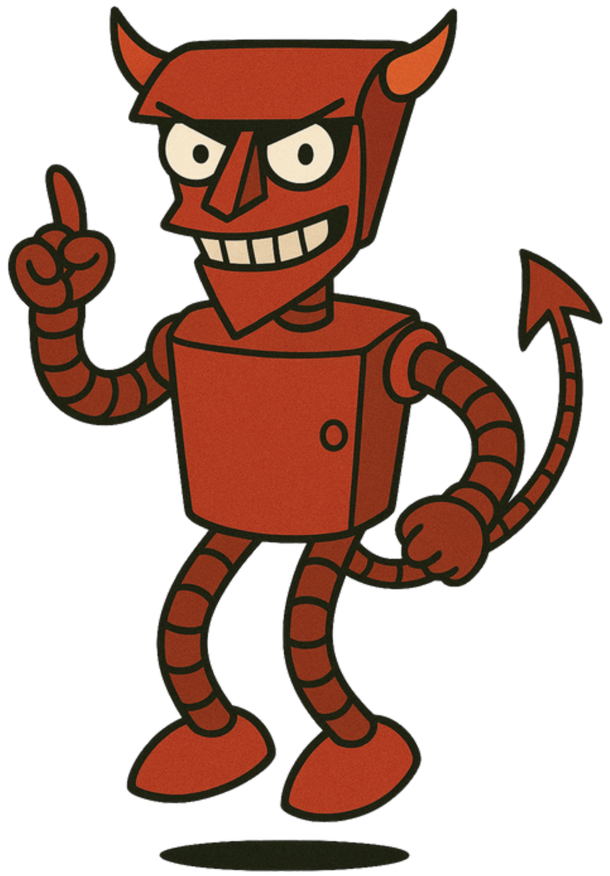
Still not enought!

- Need higher resolution
  (currently ~1°≈100 km but require 0.01°≈1 km; Palmer 2014)
- Too few emsembles for robust scenario analysis (uncertainties)
- Too few models (combinatorial explosion of parameters/forcings)

AI to the rescue?

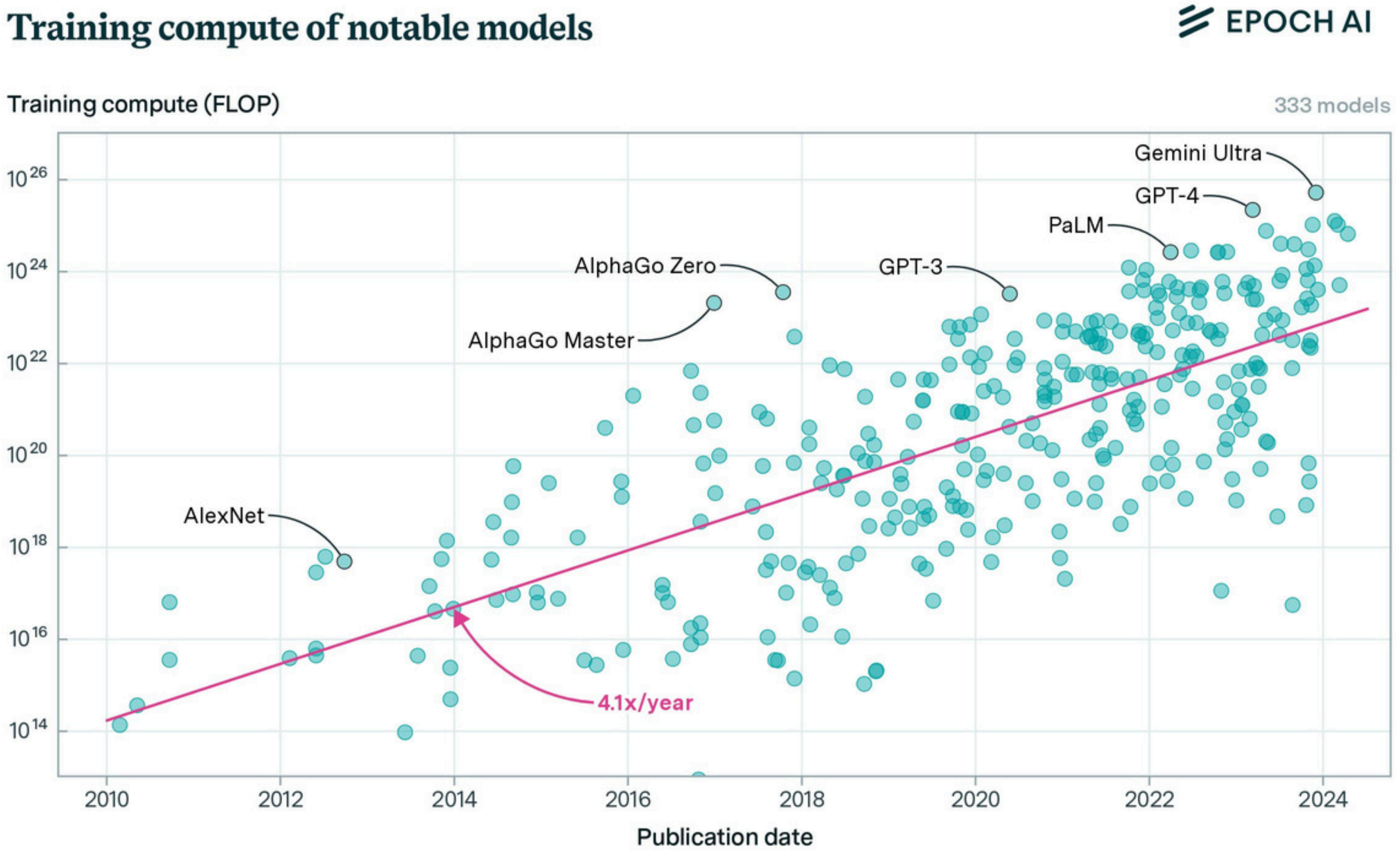


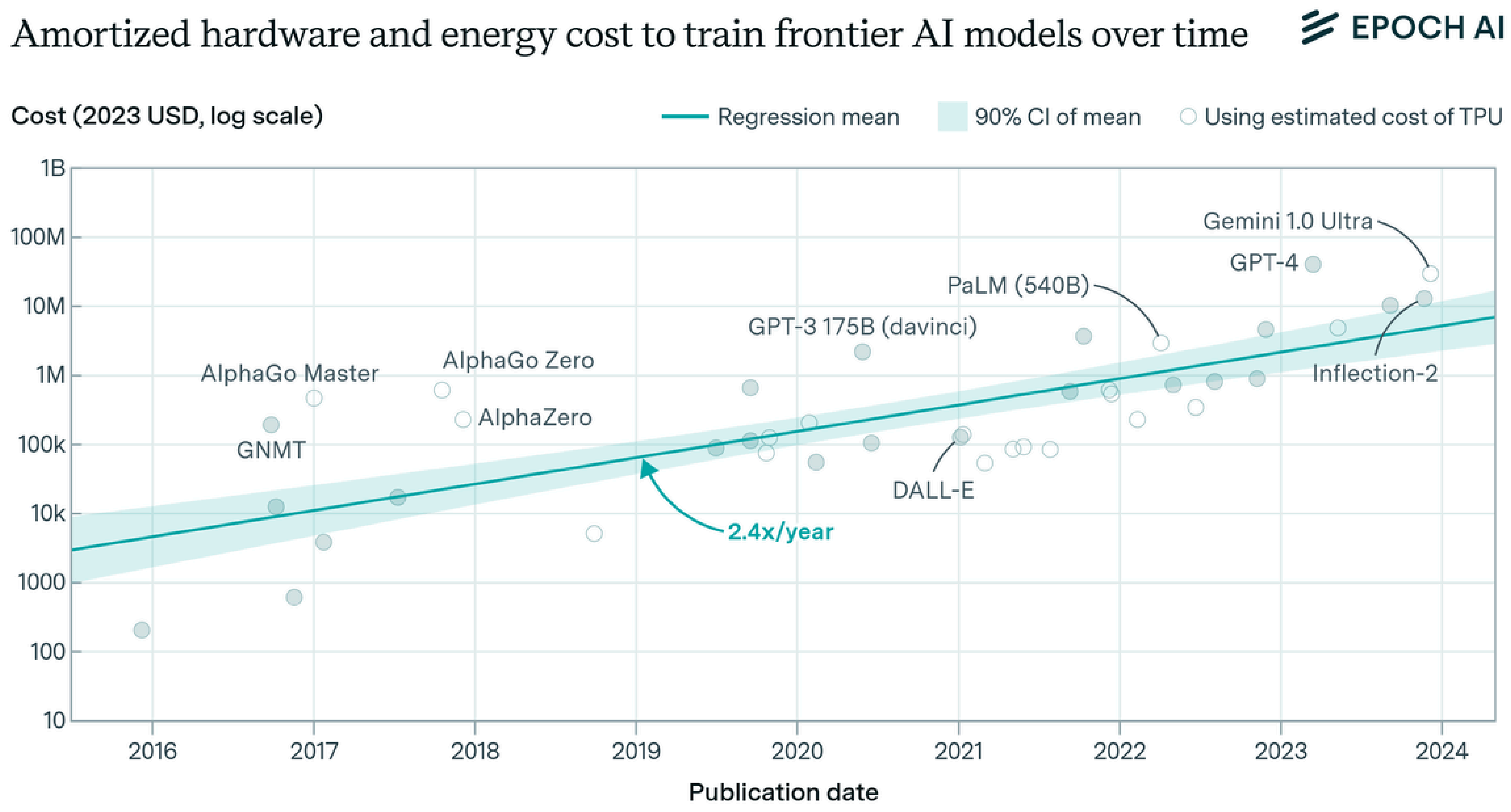
Can we use AI to alleviate the computation cost of simulating physical systems?



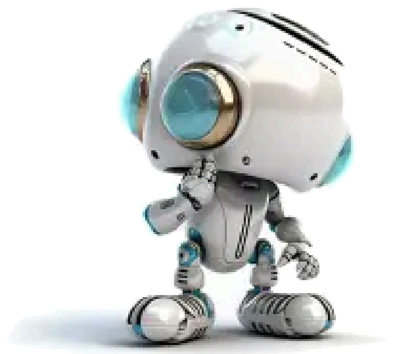
But doesn’t AI itself require huge computational costs?

Computational costs of training large AI models





Large vs small AI models



Large AI models

- ✗ Costly (compute, energy, carbon footprint)
- ✓ General purpose

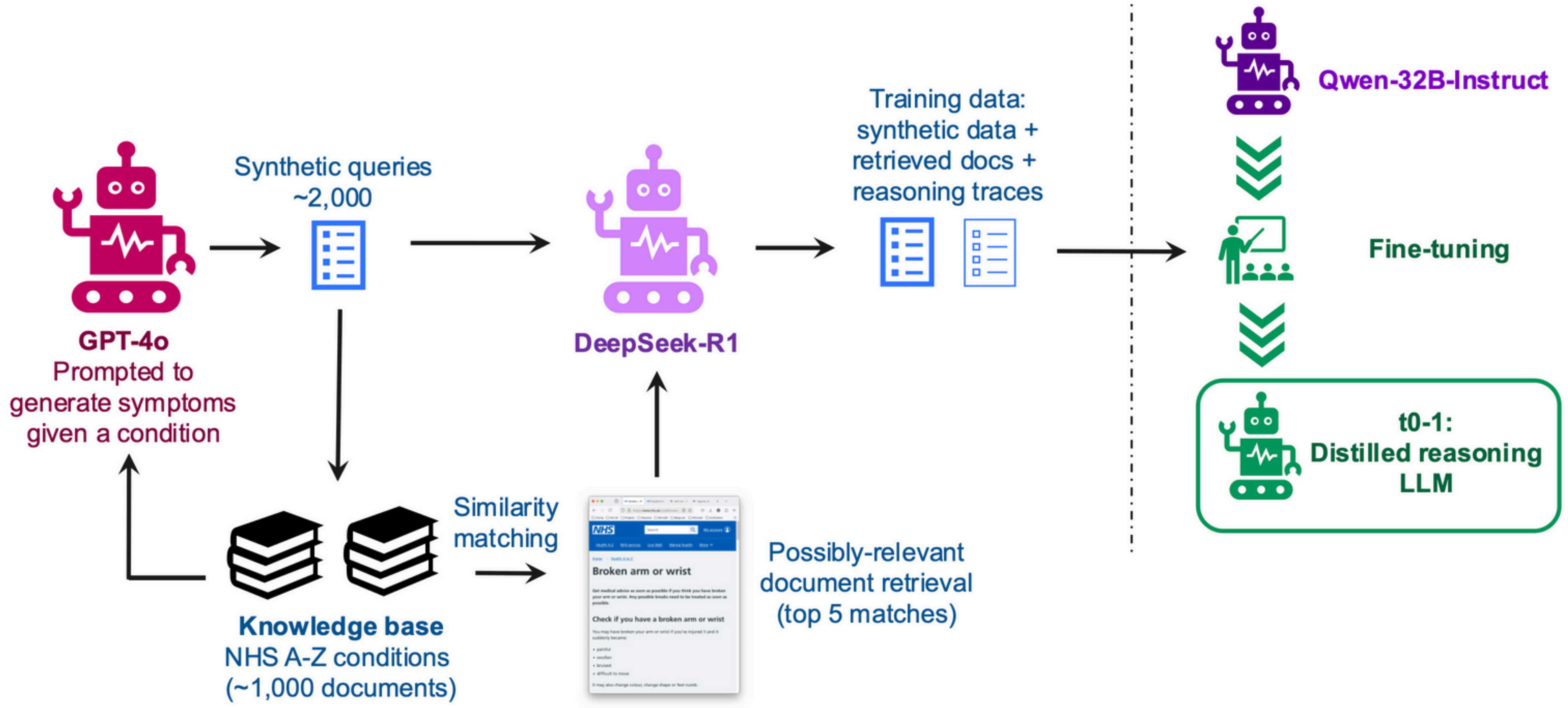
Small AI models

- ✓ Cheap (compute, energy, carbon footprint)
- ✓ Specialised

Project t0: small language models

RAG-augmented reasoning with lean language models (Chan et al. 2025)

- Distillation
- Fine-tuning
- Reasoning with budget forcing
- RAG (retrieval-augmented generation)



- ✔ Frontier performance, without frontier compute
- ✔ Compute-constrained environments
- ✔ Privacy-sensitive environments

Blog: Why we still need small language models

Outline

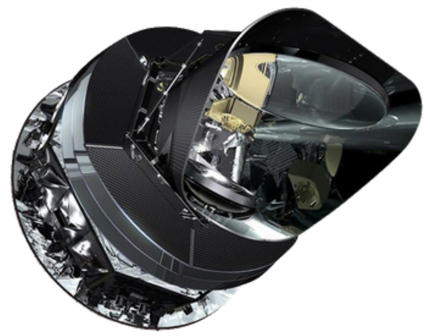
1. **Traditional scientific inference** for physical systems

2. **Accelerated scientific inference** for physical systems
    - Emulation
    - Programming frameworks
    - Gradient-based MCMC sampling
    - Decoupled Bayesian model comparison

3. Cosmological **case studies**

Traditional scientific inference
for physical systems

Traditional Bayesian inference for physical systems



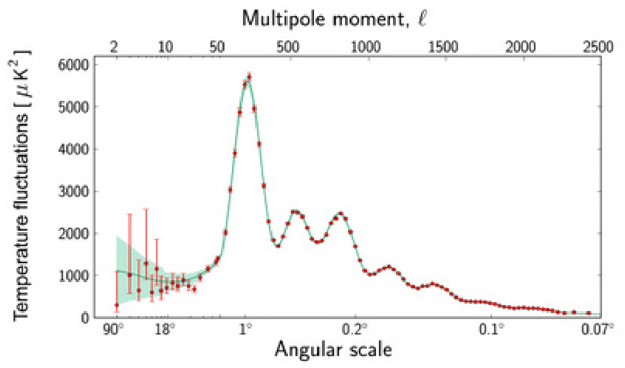
Instrument

Observational Data

Parameters

$\theta = (A_s, n_s, \tau, \Omega_{\rm b}h^2, \Omega_{\rm c}h^2, \Omega_\Lambda)$

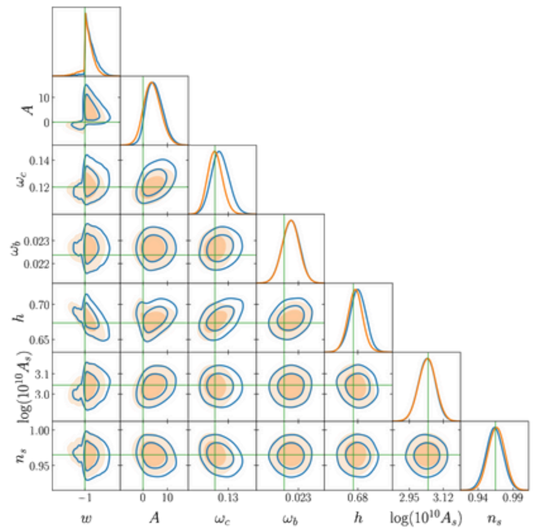
Physical Simulator



Bayesian Inference
(coupled parameter estimation
& model comparison)

Evidence $\mathcal{Z}$

MCMC samples $\{\theta\}_i$



Bayesian inference: parameter estimation

Bayes' theorem

$$\underbrace{p(\theta \mid x, M)}_{\text{posterior}} = \frac{\overbrace{p(x \mid \theta, M)}^{\text{likelihood}} \; \overbrace{p(\theta \mid M)}^{\text{prior}}}{\underbrace{p(x \mid M)}_{\text{marginal likelihood}}} = \frac{\overbrace{\mathcal{L}(\theta)}^{\text{likelihood}} \; \overbrace{\pi(\theta)}^{\text{prior}}}{\underbrace{z}_{\text{marginal likelihood}}},$$

for parameters θ, model M and observed data x.

For **parameter estimation**, typically draw samples from the posterior by *Markov chain Monte Carlo (MCMC)* sampling.

Bayesian inference: model comparison

By Bayes' theorem for model M_j:

$$p(M_j \,|\, x) = \frac{p(x \,|\, M_j) p(M_j)}{\sum_j p(x \,|\, M_j) p(M_j)} \, .$$

For **model comparison**, consider posterior model odds:

$$\underbrace{\frac{p(M_1 \,|\, x)}{p(M_2 \,|\, x)}}_{\text{posterior odds}} = \underbrace{\frac{p(x \,|\, M_1)}{p(x \,|\, M_2)}}_{\text{Bayes factor}} \times \underbrace{\frac{p(M_1)}{p(M_2)}}_{\text{prior odds}} \, .$$

Must compute the **marginal likelihood** (aka. **Bayesian model evidence**) given by the normalising constant

$$z = p(x \,|\, M) = \int \mathrm{d}\theta \, \mathcal{L}(\theta) \, \pi(\theta) \, .$$

$\rightsquigarrow$ Challenging computational problem.

Nested sampling (Skilling 2006)

Group the parameter space Ω into a series of **nested subspaces**: $\Omega_{L^*} = \{x \,|\, \mathcal{L}(x) \geq L^*\}$. Define the prior volume ξ within Ω_{L^*} by

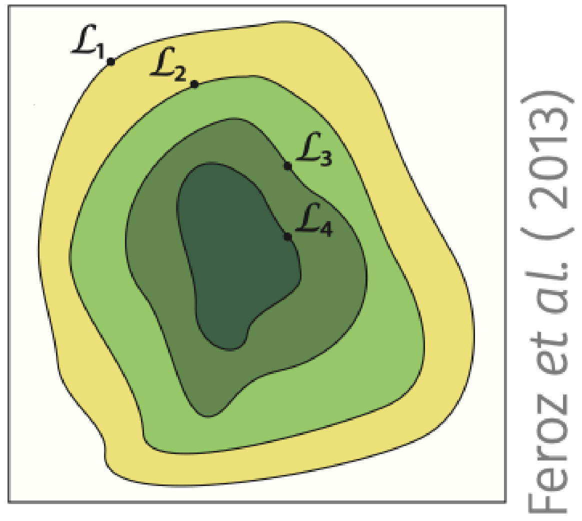
$$\xi(L^*) = \int_{\Omega_{L^*}} \pi(x)\mathrm{d}x.$$

Marginal likelihood can then be rewritten as

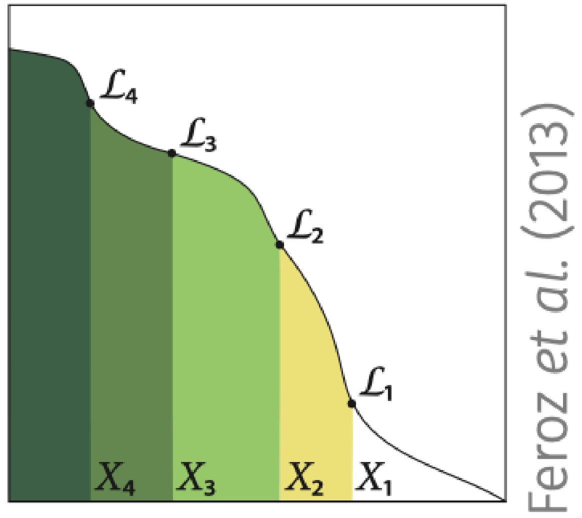
$$z = \int_0^1 \mathcal{L}(\xi)\mathrm{d}\xi.$$

Require computational strategy to compute likelihood level-sets (iso-contours) L_i and corresponding prior volumes $0 < \xi_i \leq 1$.

Crux: **sample from the prior, subject to the likelihood level-set constraint**, *i.e.* sample from the prior $\pi(x)$, such that $\mathcal{L}(x) > L^*$.



Nested subspaces



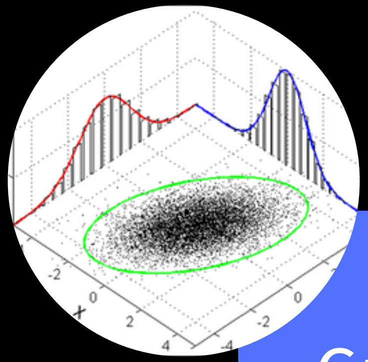
Reparameterised likelihood

Accelerated scientific inference for physical systems

Pillars of accelerated scientific inference

1st Pillar:
AI
Emulation

2nd Pillar:
Programming
Frameworks



3rd Pillar:
Gradient-Based
MCMC Sampling

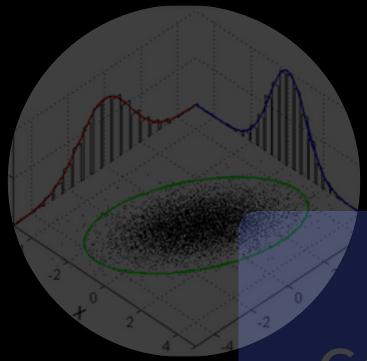
4th Pillar:
Decoupled Model
Comparison

Pillars of accelerated scientific inference

Auto Emulate

1st Pillar:
AI
Emulation

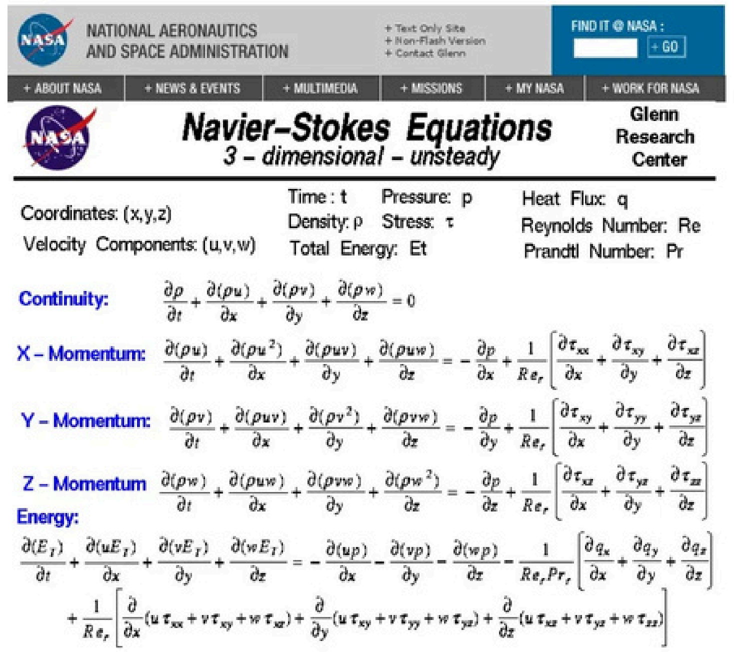
2nd Pillar:
Programming
Frameworks



3rd Pillar:
Gradient-Based
MCMC Sampling

4th Pillar:
Decoupled Model
Comparison

Simulation vs emulation



Simulate physical laws

Emulate by training an AI model to micmic physical laws

- ✓ Accurate representation of physical model
- ✗ Highly computationally costly

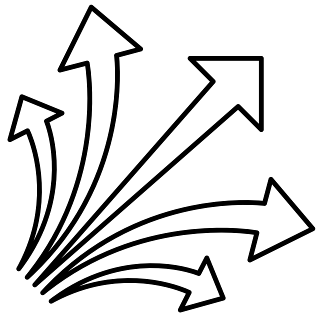
- ✗ Approximate representation of physical model
- ✓ Computationally efficient (once trained)
- ✓ Learning data-driven model has potential to be more accurate than physical model

AutoEmulate: general purpose emulation package

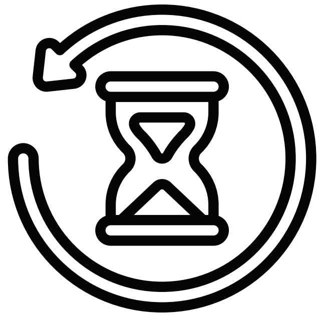
- Users with Domain expertise
- No Machine learning expertise in Emulation is necessary
- Democratising the use of AI for accelerating simulations in various industries

https://www.autoemulate.com

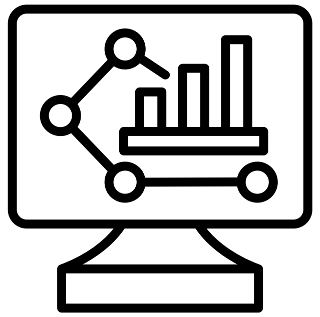
AutoEmulate: more than a learned emulator



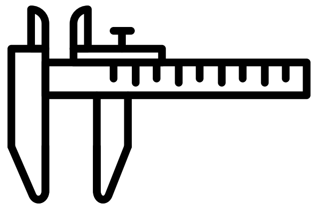
Sensitivity analysis



History matching

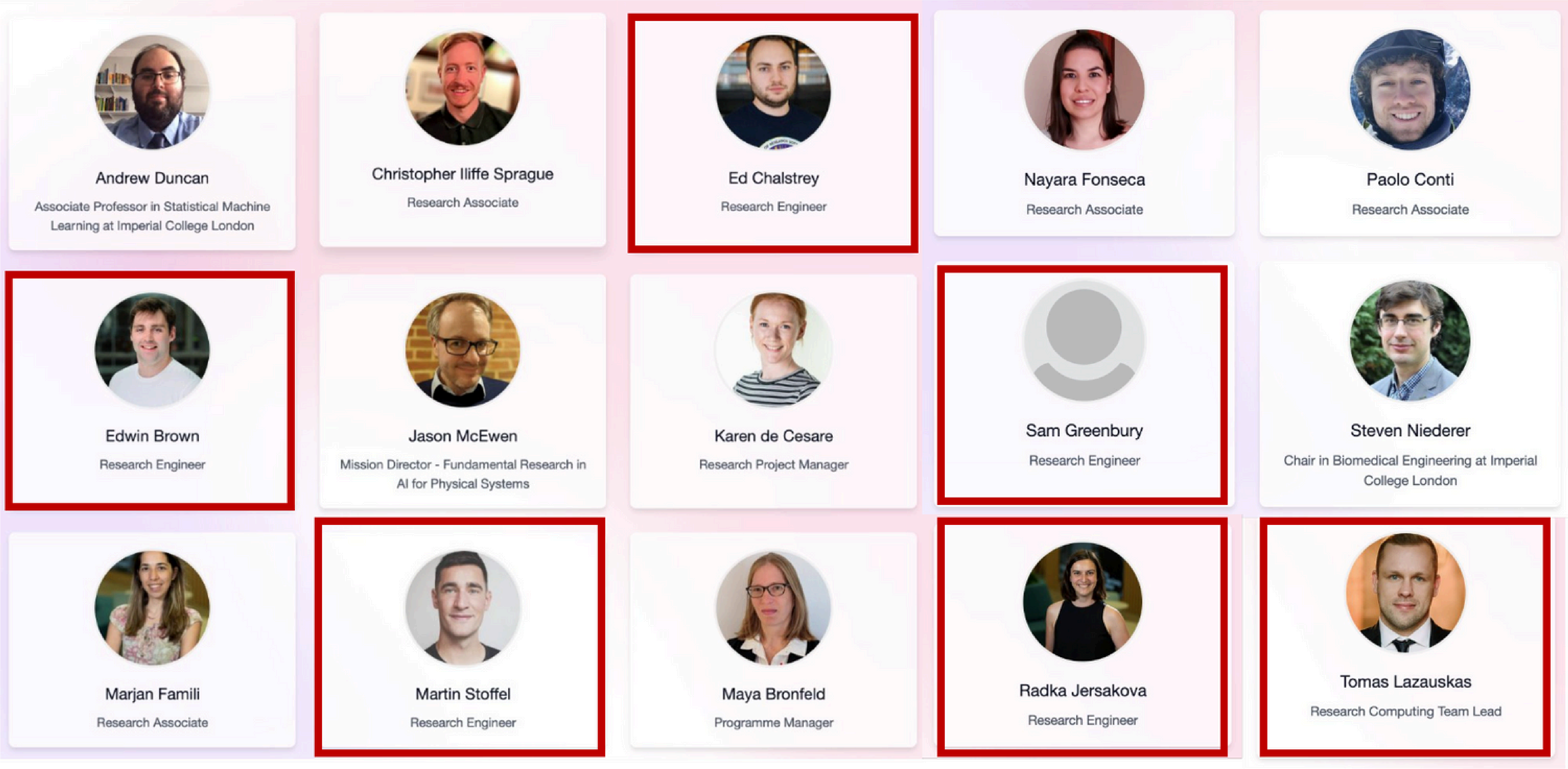


Simulator in the loop



Bayesian calibration

AutoEmulate team



Emulation for cosmology

COSMOPOWER

https://github.com/alessiospuriomancini/cosmopower

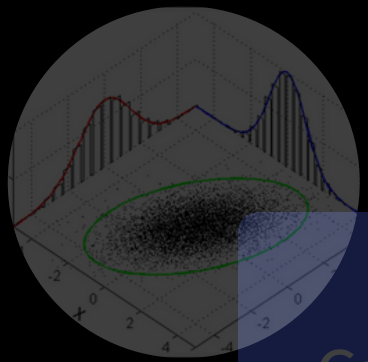
https://github.com/dpiras/cosmopower-jax

Pillars of accelerated scientific inference

Auto Emulate

1st Pillar:
AI
Emulation

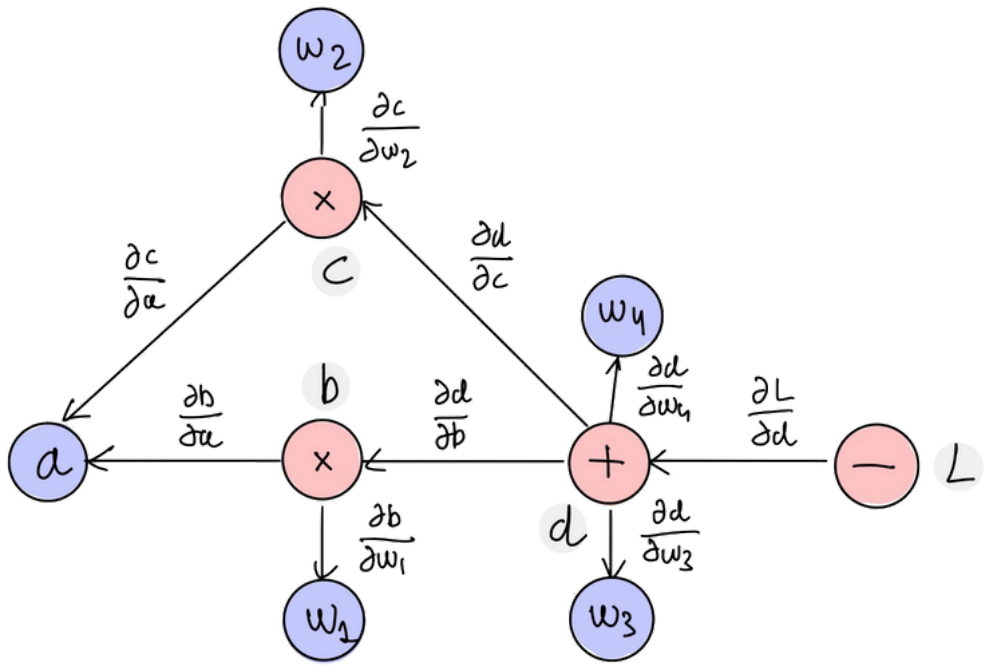
2nd Pillar:
Programming
Frameworks



3rd Pillar:
Gradient-Based
MCMC Sampling

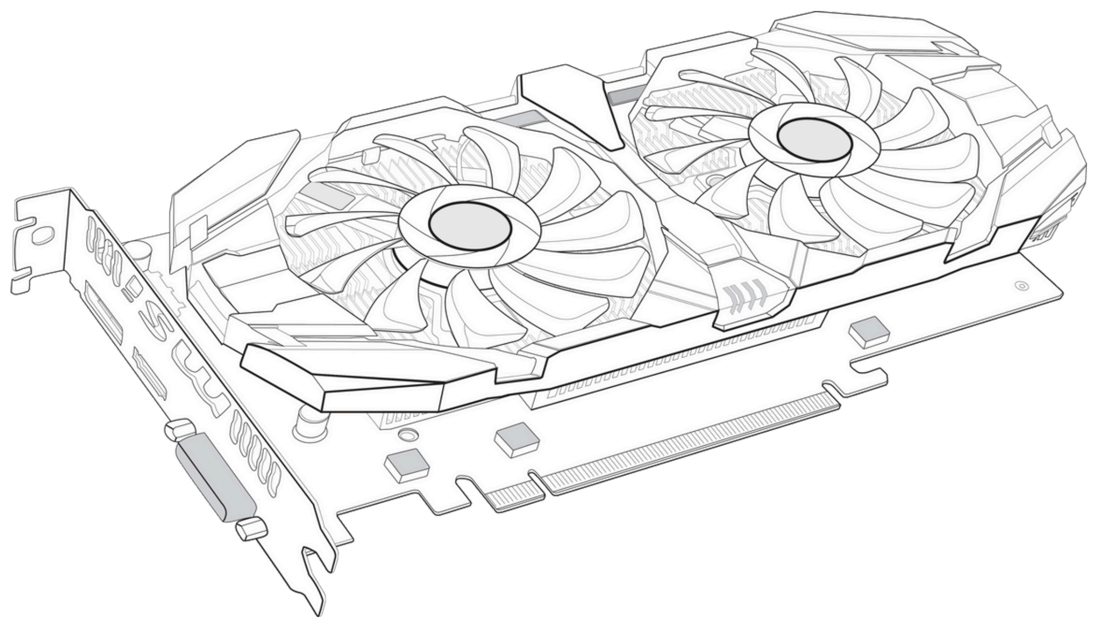
4th Pillar:
Decoupled Model
Comparison

Programming frameworks



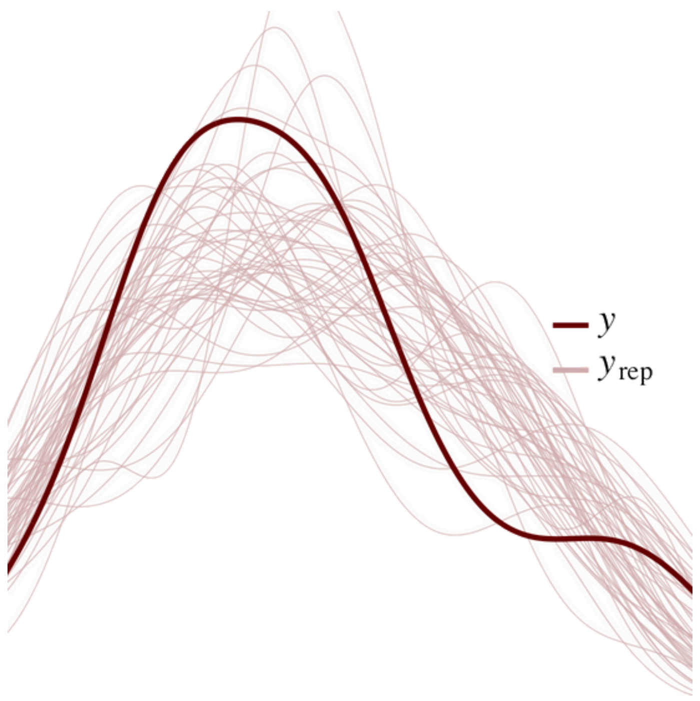
Automatic differentiation

JAX



GPU acceleration

XLA



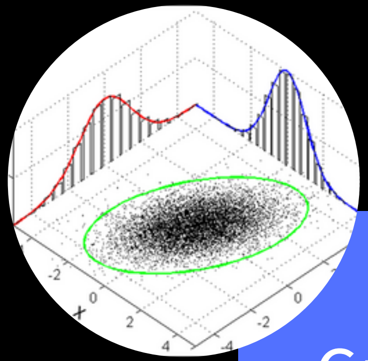
Probabilistic programming

NumPyro

Pillars of accelerated scientific inference

1st Pillar:
AI
Emulation

2nd Pillar:
Programming
Frameworks

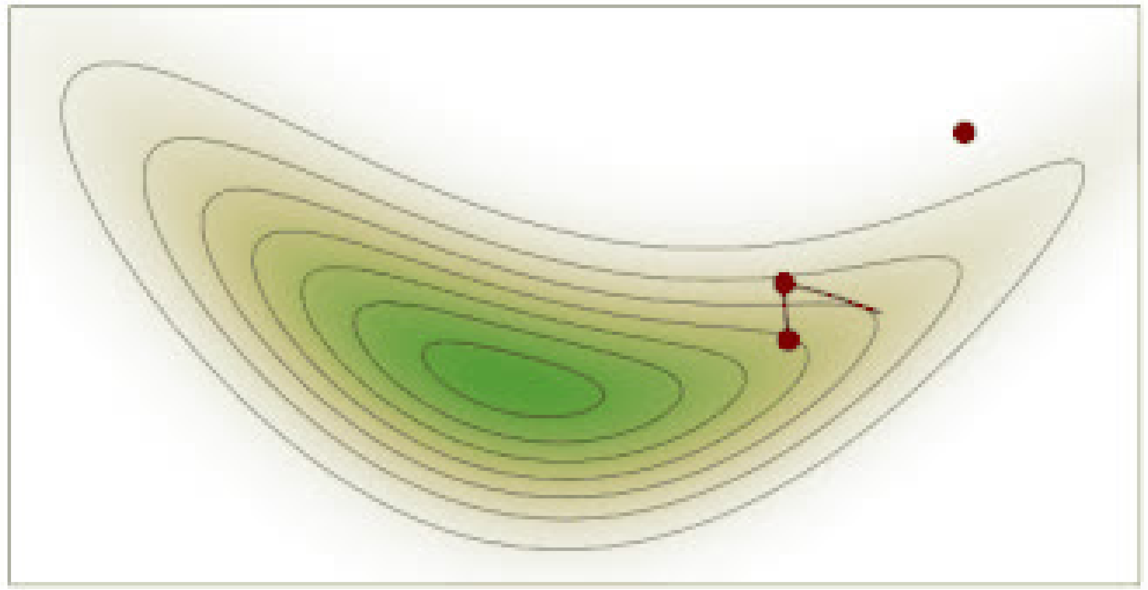


3rd Pillar:
Gradient-Based
MCMC Sampling

4th Pillar:
Decoupled Model
Comparison

Gradient-accelerated MCMC sampling

Exploit gradient information to scale MCMC efficiently to higher dimensional settings (e.g. Hamiltonian or Langevin dynamics).



Consider **Hamiltonian Monte Carlo (HMC)**, where samples θ augmented with momentum p. Hamiltonian given by

$$H(\theta, \mathrm{p}) = -\log p(\theta|x) + \frac{1}{2}\mathrm{p}^{\mathrm{T}} M^{-1} \mathrm{p} \, ,$$

where M is the mass matrix. Evolution given by dynamics

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{\partial H}{\partial \mathrm{p}} \, , \quad \frac{\mathrm{dp}}{\mathrm{d}t} = -\frac{\partial H}{\partial \theta} \, .$$

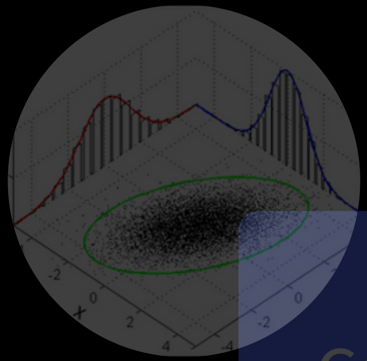
Consider **No U-Turn (NUTS)** algorithm.

Compute gradients efficiently by automatic differentiation.

Pillars of accelerated scientific inference

1st Pillar:
AI
Emulation

2nd Pillar:
Programming
Frameworks



3rd Pillar:
Gradient-Based
MCMC Sampling

4th Pillar:
Decoupled Model
Comparison

The problem of nested sampling for Bayesian model comparison

Nested sampling (Skilling 2006) has been the method of choice for almost two decades!

Many highly effective nested sampling algorithms (for a review see Ashton *et al.* 2022).

However, nested sampling has a **fundamental problem**...

Nested sampling tightly couples sampling strategy to marginal likelihood calculation.

As the name suggests, **one must sample in a nested manner**.

- **Precludes** many alternative **accelerated sampling** strategies that scale to high-dimensions.
- **Precludes** use in many **simulation-based inference (SBI)** and **variational inference (VI)** settings, where one draws posterior samples directly.

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

$$\rho = \mathbb{E}_{p(\theta \,|\, x)} \left[\frac{1}{\mathcal{L}(\theta)} \right] = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} p(\theta \,|\, x) = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \frac{\mathcal{L}(\theta)\pi(\theta)}{z} = \frac{1}{z}$$

Original harmonic mean estimator (Newton & Raftery 1994)

$$\hat{\rho} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\mathcal{L}(\theta_i)} \,, \quad \theta_i \sim p(\theta \,|\, x)$$

✔ **Only requires posterior samples!**

✖ But **can fail catastrophically!** (Neal 1994)

Importance sampling interpretation of harmonic mean estimator

Alternative interpretation of harmonic mean relationship:

$$\rho = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} p(\theta \,|\, x) = \frac{1}{z} \underbrace{\int \mathrm{d}\theta \frac{\pi(\theta)}{p(\theta \,|\, x)} p(\theta \,|\, x)}_{\text{importance sampling}} .$$

Importance sampling interpretation:

- Importance **sampling target distribution is prior** $\pi(\theta)$.
- Importance **sampling density is posterior** $p(\theta \,|\, x)$.

For importance sampling, want sampling density to have fatter tails than target.

Importance sampling failure mode when sampling density is posterior and target is prior.

Re-targeted harmonic mean estimator

***Re-targeted* harmonic mean relationship** (Gelfand & Dey 1994)

$$\rho = \mathbb{E}_{p(\theta \,|\, x)} \left[\frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \right] = \frac{1}{z}$$

Normalised distribution $\varphi(\theta)$ **now plays the role of the importance sampling target**
⇝ must **not** have fatter tails than posterior.

***Re-targeted* harmonic mean estimator** (Gelfand & Dey 1994)

$$\hat{\rho} = \frac{1}{N} \sum_{i=1}^{N} \frac{\varphi(\theta_i)}{\mathcal{L}(\theta_i)\pi(\theta_i)} , \quad \theta_i \sim p(\theta \,|\, x)$$

⇝ **How set importance sampling target distribution $\varphi(\theta)$?**

How set importance sampling target distribution?

Variety of cases been considered:

- ▷ Multi-variate Gaussian (*e.g.* Chib 1995)
- ▷ Indicator functions (*e.g.* Robert & Wraith 2009, van Haasteren 2009)

Optimal target: (McEwen *et al.* 2021)

$$\varphi^{\text{optimal}}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{z} .$$

But clearly **not feasible** since requires knowledge of the evidence z (recall the target must be normalised) $\rightsquigarrow$ **requires problem to have been solved already!**

Learned harmonic mean estimator

Learn an approximation of the optimal target distribution:

$$\varphi(\theta) \stackrel{\text{AI}}{\simeq} \varphi^{\text{optimal}}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{z} .$$

- Approximation not required to be highly accurate.
- Critically, **must not have fatter tails than posterior**.

Constraining tails of target approach 1: bespoke optimisation problem

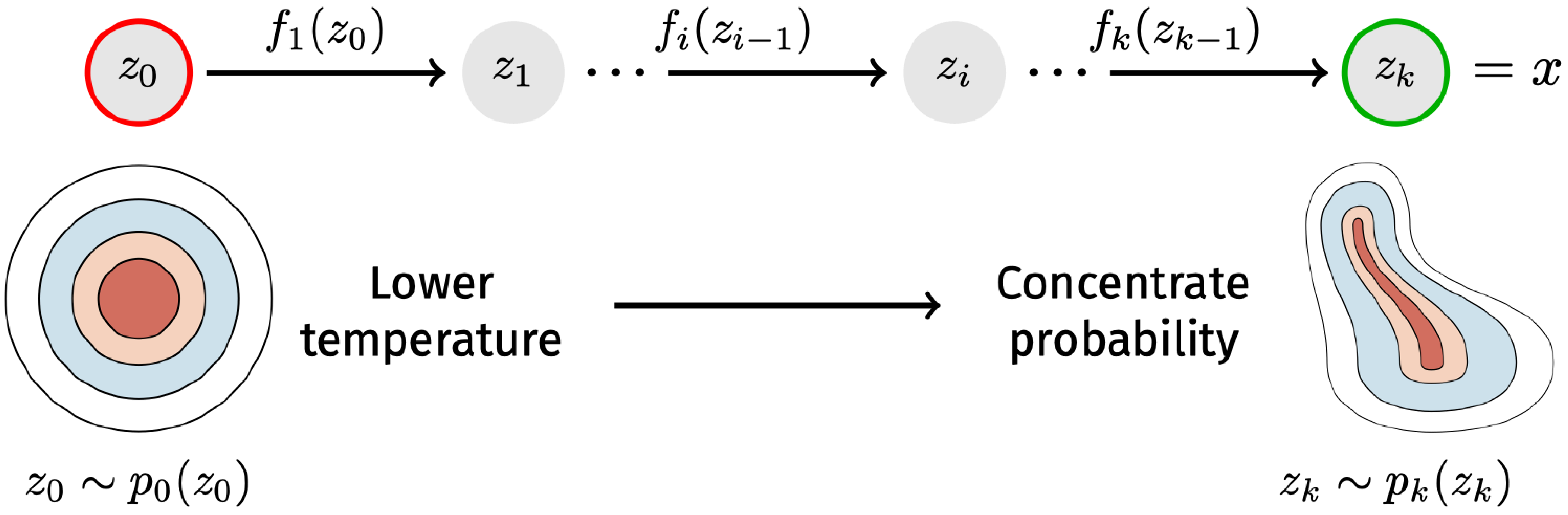
Fit density estimator by **minimising variance of resulting estimator**, with possible regularisation:

$$\min \ \hat{\sigma}^2 + \lambda R \quad \text{subject to} \quad \hat{\rho} = \hat{\mu}_1.$$

Solve by bespoke **mini-batch stochastic gradient descent**.

Cross-validation to select density estimation model and hyperparameters.

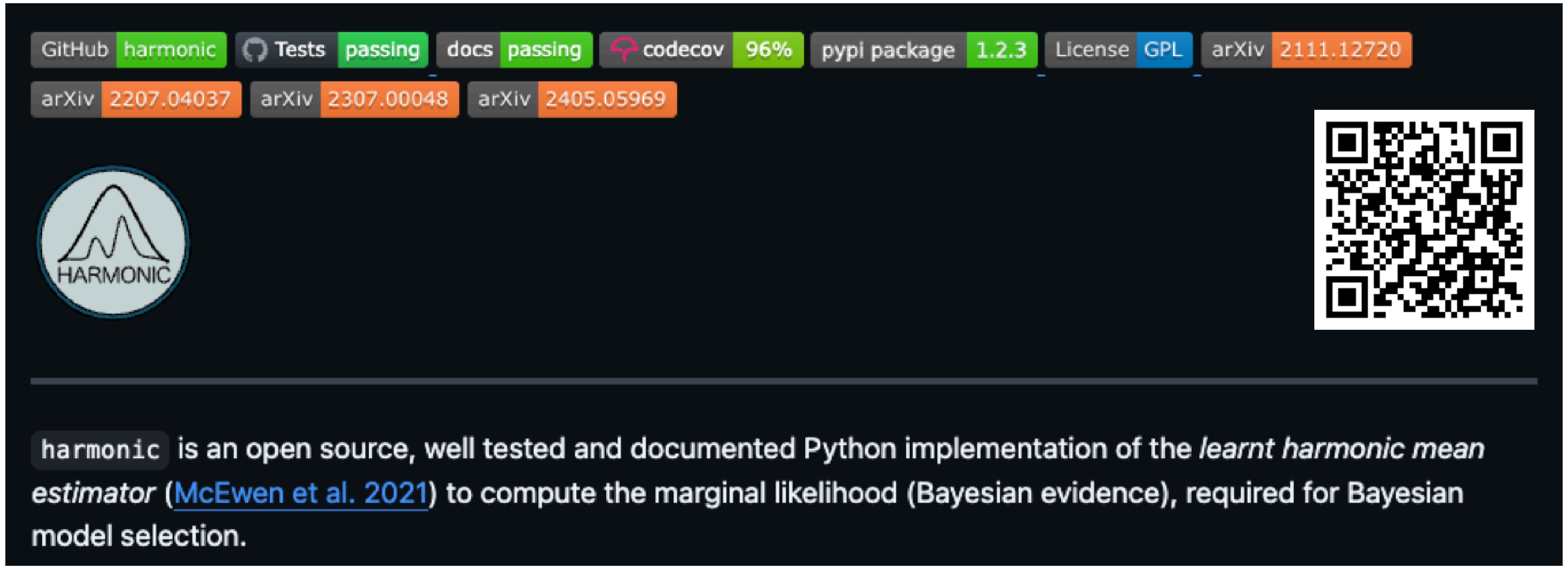
Constraining tails of target approach 2: normalizing flows



- **Flexible**: no bespoke training; can vary *T* after training.
- **Robust**: only one hyperparameter *T* that does not require fine tuning.
- **Scalable**: flows scale to higher dimensions than classical density estimators.

(Polanska et al. McEwen 2024)

Harmonic code

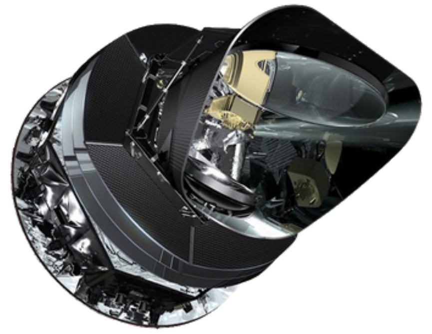


`harmonic` is an open source, well tested and documented Python implementation of the *learnt harmonic mean estimator* (McEwen et al. 2021) to compute the marginal likelihood (Bayesian evidence), required for Bayesian model selection.

Github: https://github.com/astro-informatics/harmonic

Docs: https://astro-informatics.github.io/harmonic

Traditional Bayesian inference for physical systems



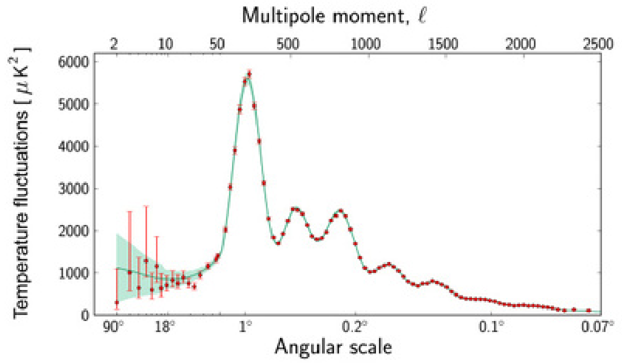
Instrument

Observational Data

Parameters

$\theta = (A_s, n_s, \tau, \Omega_b h^2, \Omega_c h^2, \Omega_\Lambda)$

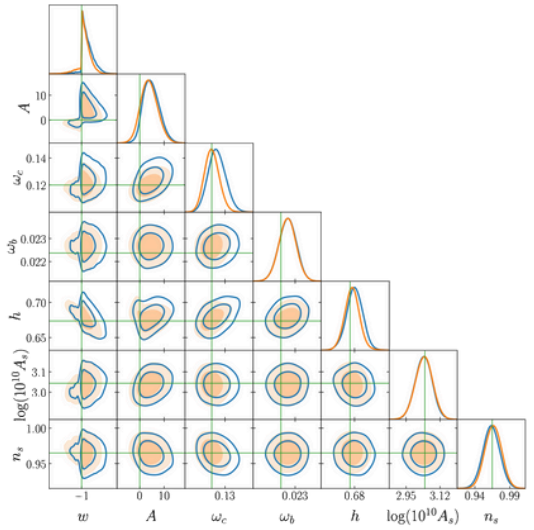
Physical Simulator



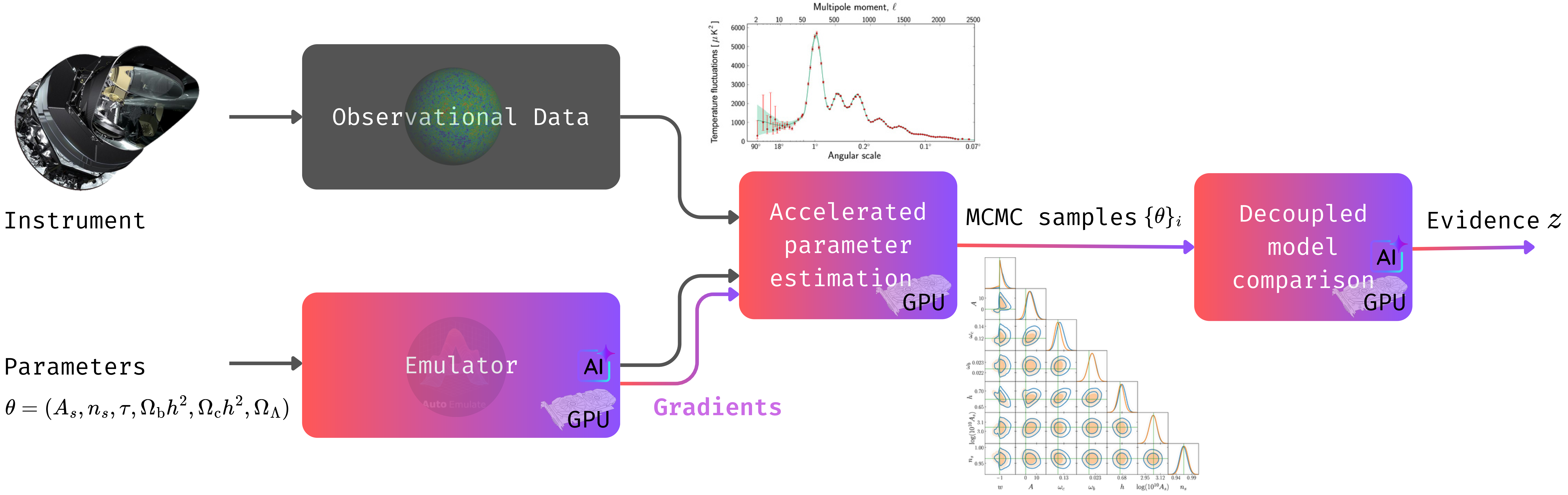
Bayesian Inference
(coupled parameter estimation
& model comparison)

Evidence z

MCMC samples $\{\theta\}_i$



Accelerated Bayesian inference for physical systems



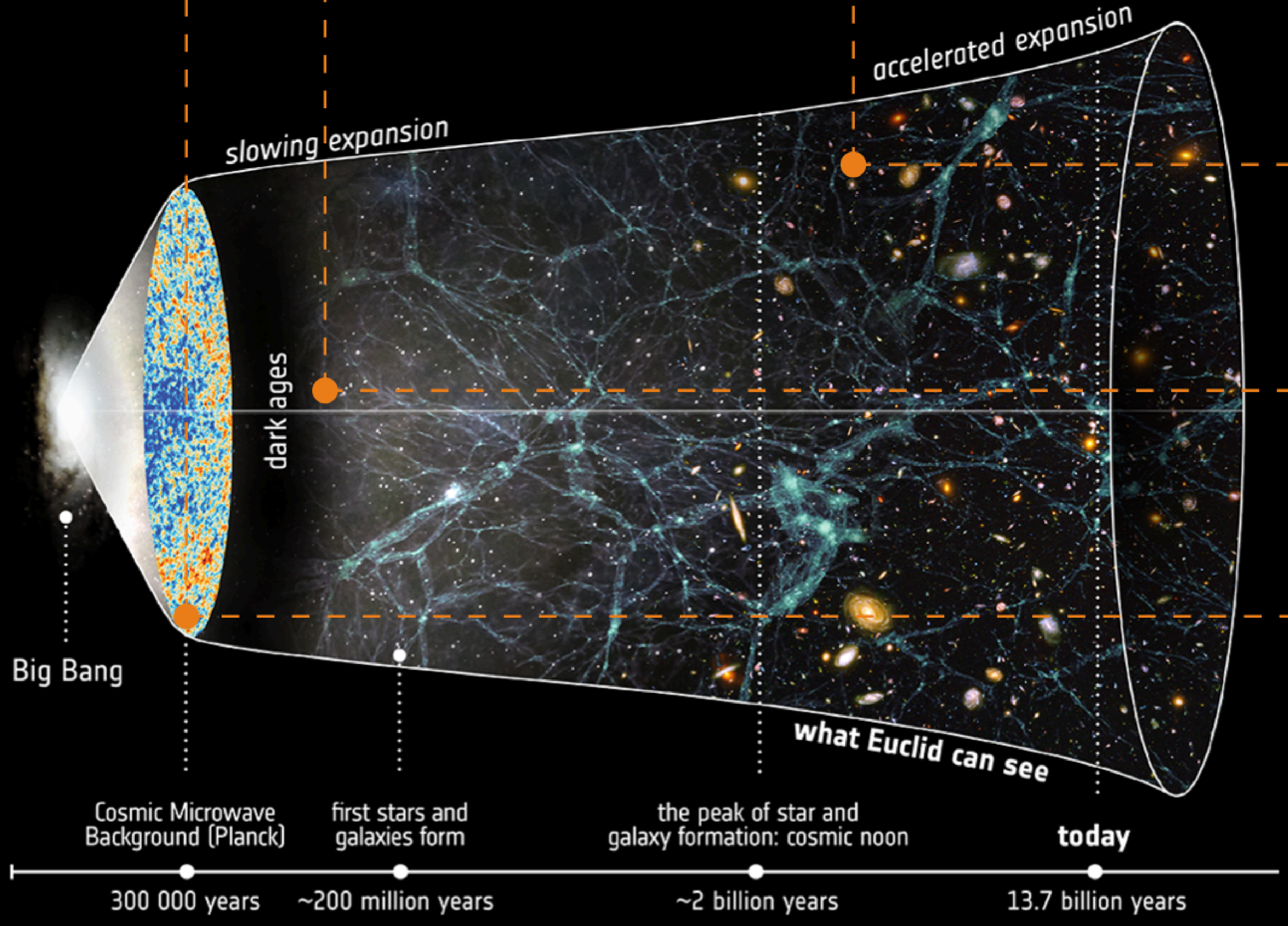
Cosmological **case studies**

Towards a fundamental understanding of our Universe

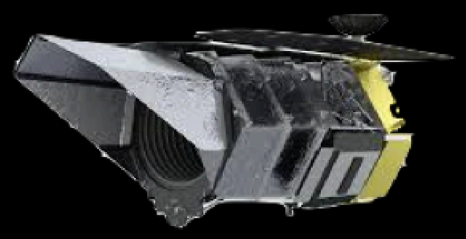
What is the origin of structure?

How did luminous large-scale structure form?

What is the nature of dark energy and dark matter?



Euclid



Roman

Rubin-LSST

SKA

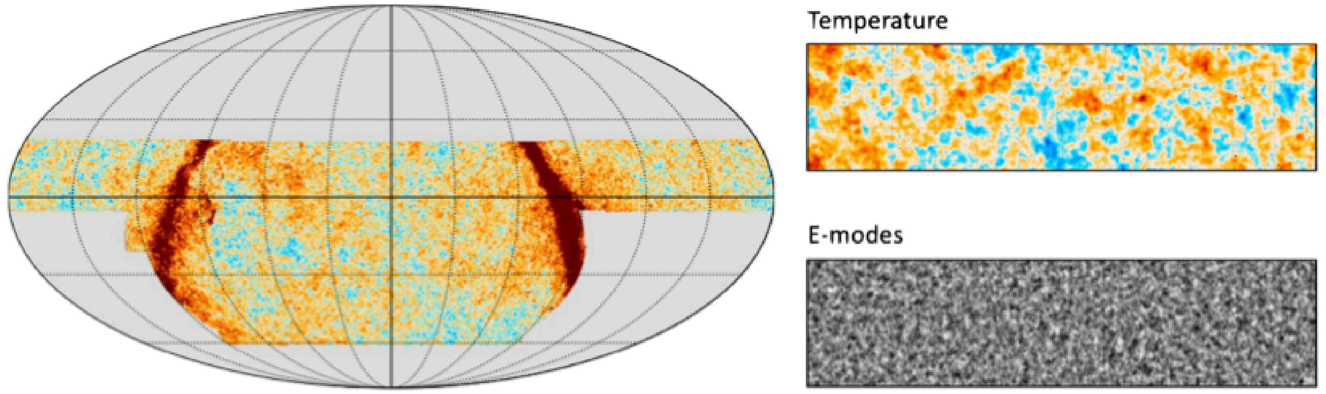
LiteBIRD

Simons

Atacama Cosmology Telescope (ACT) analysis

Compare ΛCDM (**Einstein's cosmological constant**) vs $w_0 w_a$CDM (**dynamical dark energy**) using learned harmonic mean (McEwen *et al.*2021) with ACT data (Aiola *et al.* 2020).

Atacama Cosmology Telescope (ACT)



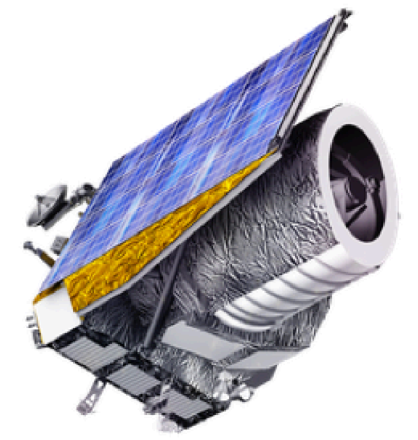
CMB observations

7D vs 9D models:	ΛCDM	$w_0 w_a$CDM	$\log \mathrm{BF}_{\Lambda\mathrm{CDM}\text{-}w_0 w_a\mathrm{CDM}}$
Nested sampling	-168.92 ± 0.35	-169.38 ± 0.24	0.46 ± 0.42
Learned harmonic mean	-168.87 ± 0.29	-169.32 ± 0.25	0.45 ± 0.38

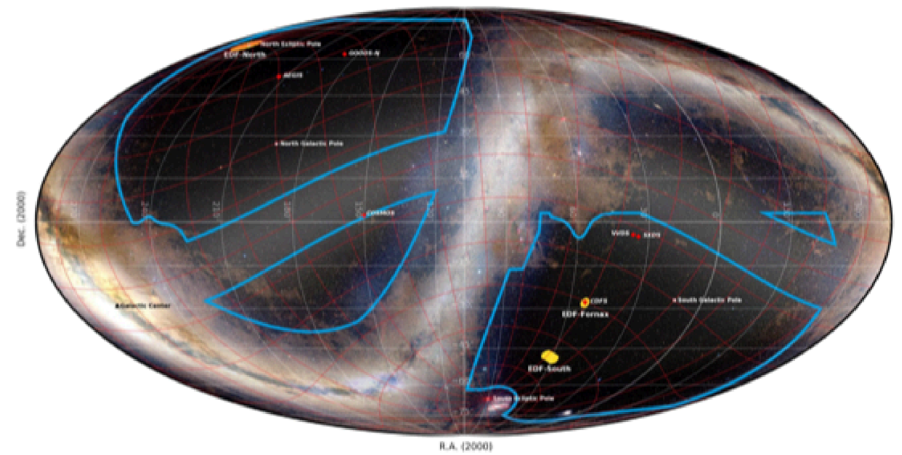
⇝ ΛCDM mildly favoured ⇝ $3\times$ **acceleration** (Only Pillar 4)

Euclid (Stage IV survey)-like analysis

Compare ΛCDM vs $w_0 w_a$CDM leveraging **4 pillars of AI-acceleration** with Euclid-like lensing and clustering simulations (Piras *et al.* 2024).



Euclid satellite



Observation field

37D vs 39D models:	$\log(z_{\Lambda\mathrm{CDM}})$	$\log(z_{w_0 w_a \mathrm{CDM}})$	$\log \mathrm{BF}_{\Lambda\mathrm{CDM}\text{-}w_0 w_a\mathrm{CDM}}$	Total computation time
Classical	-107.03 ± 0.27	-107.81 ± 0.74	0.78 ± 0.79	8 months (48 CPUs)
AI-accelerated (ours)	40956.55 ± 0.06	40955.03 ± 0.04	1.53 ± 0.07	2 days (12 GPUs)

Simulating training data = 1 CPU day | Training = 1 GPU hour | Amortized over all analyses

Euclid (Stage IV survey)-like analysis

Traditional approach

Energy ≈ 4,000 kWh ≈ 19 household electricity months

Carbon ≈ 0.8t CO_2 ≈ 3,000 car miles ≈ 22 car weeks

Accelerated approach (ours)

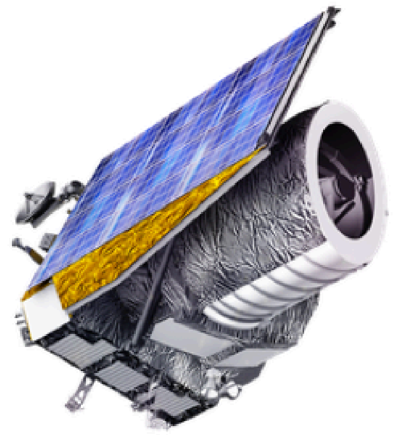
Energy ≈ 187 kWh ≈ 1 household electricity month

Carbon ≈ 0.04t CO_2 ≈ 143 car miles ≈ 1 car week

Legend: monthly UK household electricity; weekly UK car usage

Euclid-Rubin-Roman (3x Stage IV survey)-like analysis

Extend to combined 3× Stage IV Survey-like lensing and clustering simulations (Piras *et al.* 2024).



Euclid satellite

Rubin observatory

Roman satellite

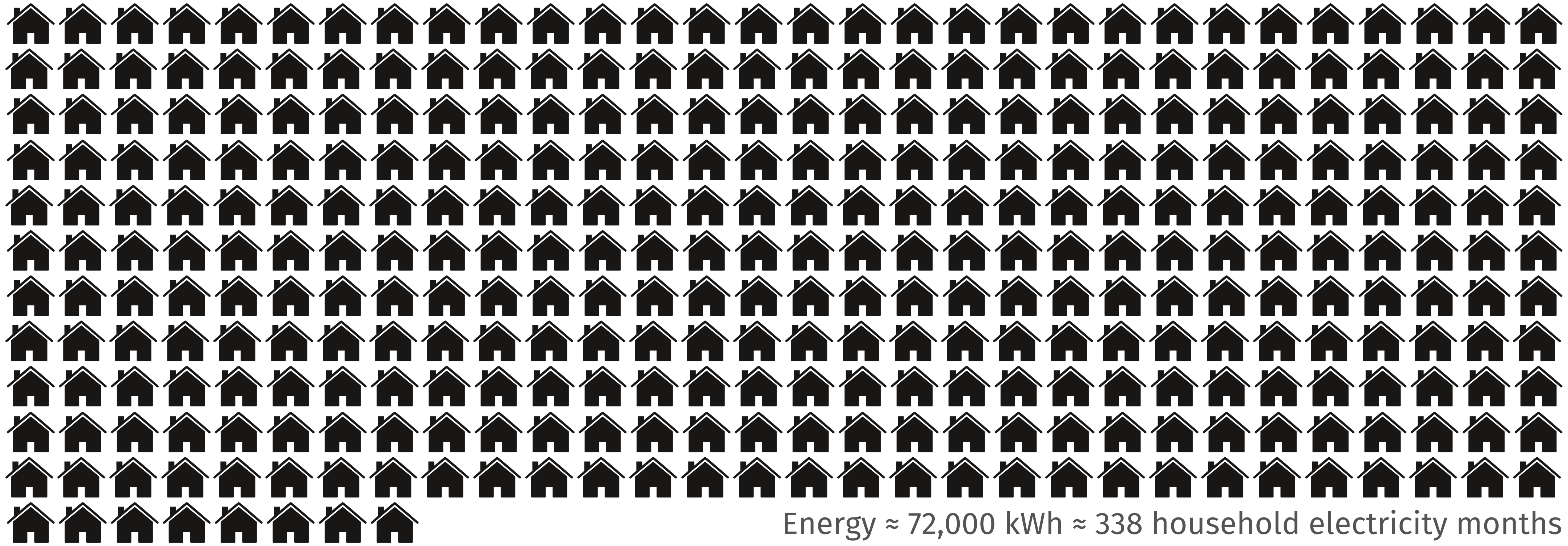
157D vs 159D models:	$\log(z_{\Lambda\mathrm{CDM}})$	$\log(z_{w_0 w_a \mathrm{CDM}})$	$\log$ BF	Total computation time
Classical	Unfeasible	Unfeasible	Unfeasible	12 years projected (48 CPUs)
AI-accelerated (ours)	$406689.6^{+0.5}_{-0.3}$	$406687.7^{+0.5}_{-0.3}$	$1.9^{+0.7}_{-0.5}$	8 days (24 GPUs)

Same trained emulator as used previously

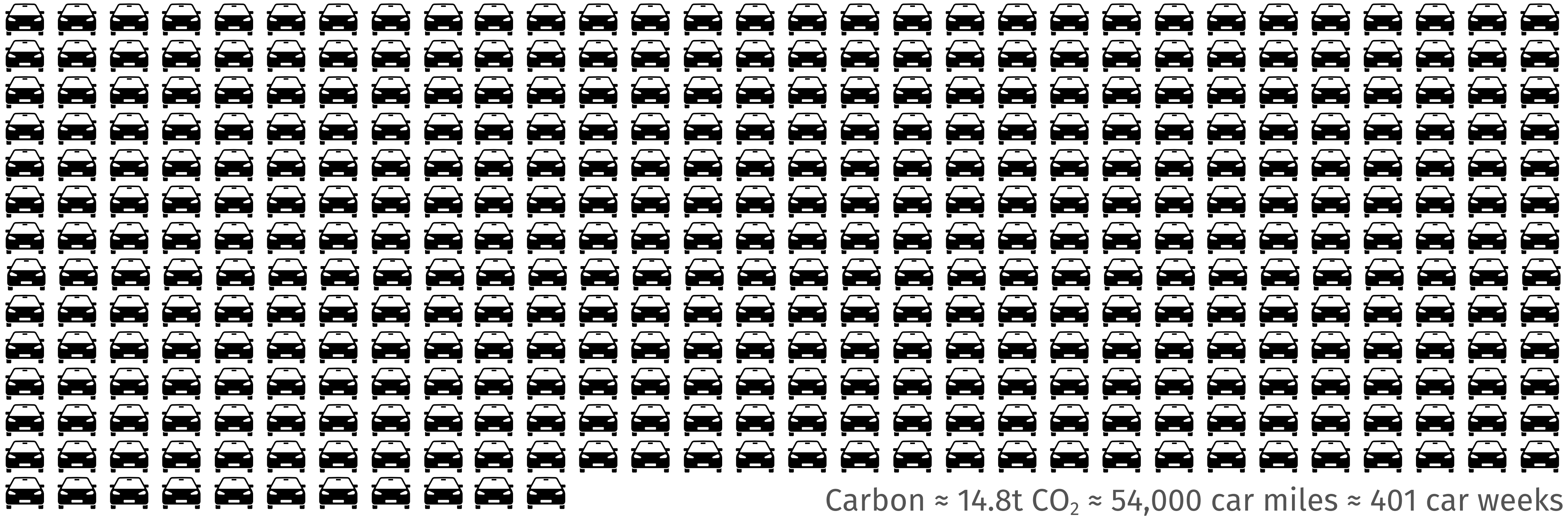
(Simulating training data = 1 CPU day | Training = 1 GPU hour | Amortized over all analyses)

Euclid-Rubin-Roman (3x Stage IV survey)-like analysis

Traditional approach



Energy ≈ 72,000 kWh ≈ 338 household electricity months



Carbon ≈ 14.8t CO_2 ≈ 54,000 car miles ≈ 401 car weeks

Accelerated approach (ours)

Energy ≈ 1,500 kWh ≈ 7 household electricity months

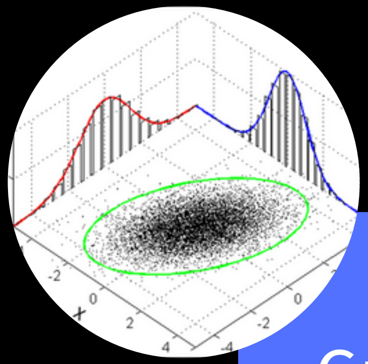
Carbon ≈ 0.3t CO_2 ≈ 1,140 car miles ≈ 9 car weeks

Legend: monthly UK household electricity weekly UK car usage

Accelerated scientific inference for physical systems

1st Pillar:
AI
Emulation

2nd Pillar:
Programming
Frameworks



3rd Pillar:
Gradient-Based
MCMC Sampling

4th Pillar:
Decoupled Model
Comparison

https://www.autoemulate.com

https://github.com/astro-informatics/harmonic

Dramatic reductions in compute cost, energy usage and carbon emissions... for every analysis.