The terms Eisenstein & Hu Missed

A precise symbolic emulator of the linear matter power spectrum

Debating the potential of machine learning for astronomical surveys (#2)

27 November 2023

Deaglan Bartlett

Institut d'Astrophysique de Paris
CNRS & Sorbonne Université

deaglan.bartlett@iap.fr

Lukas Kammerer, Gabriel Kronberger, Bogdan Burlacu

Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria

Benjamin D. Wandelt

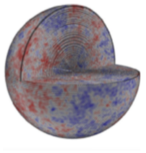
Institut d'Astrophysique de Paris & Center for Computational Astrophysics

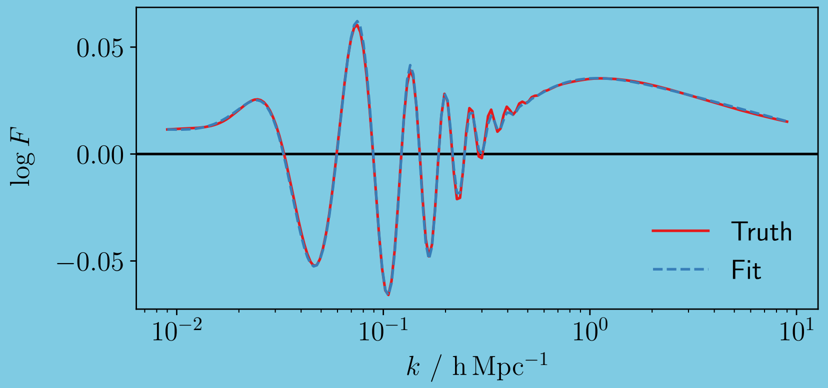
Harry Desmond

Institute of Cosmology & Gravitation, University of Portsmouth

Pedro G. Ferreira, David Alonso, Matteo Zennaro

Astrophysics, University of Oxford





We don’t always want to use a Boltzmann solver

Cosmological Parameters

A_s

Ω_m

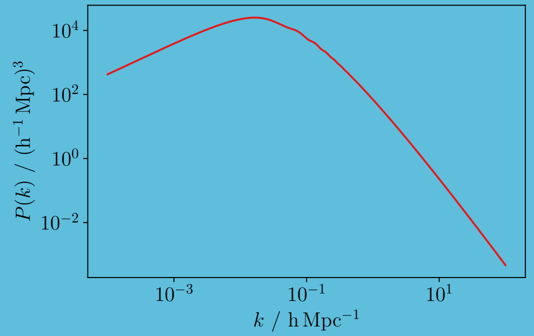
Ω_b

h

n_s

→ **Boltzmann Solver**

Power Spectra



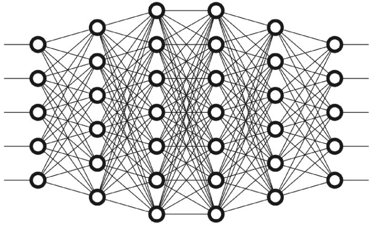
Derived Parameters

σ_8

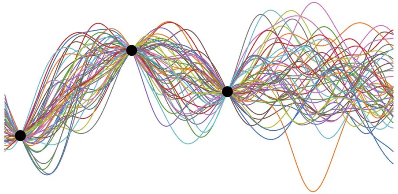
Many options exist instead of using Boltzmann solver

Numerical

Aricò et al. 2021;
Spurio Mancini et al. 2022
Euclid Collaboration et al. 2021



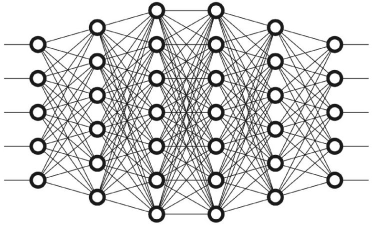
Mootoovaloo et al. 2022



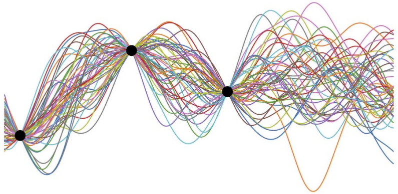
Many options exist instead of using Boltzmann solver

Numerical

Aricò et al. 2021;
Spurio Mancini et al. 2022
Euclid Collaboration et al. 2021



Mootoovaloo et al. 2022



Symbolic

Eisenstein & Hu 1998, 1998
BBKS 1986

The transfer function is written as a sum of the baryon and cold dark matter contributions at the drag epoch

$$T(k) = \frac{\Omega_b}{\Omega_0} T_b(k) + \frac{\Omega_c}{\Omega_0} T_c(k) . \tag{8}$$

The CDM transfer function can be solved exactly in terms of hypergeometric functions that are more conveniently approximated by the following form:

$$T_c \to \alpha_c \frac{\ln 1.8\beta_c q}{14.2 q^2} , \tag{9}$$

$$q = \left(\frac{k}{\mathrm{Mpc}^{-1}}\right)\Theta_{2.7}^2 (\Omega_0 h^2)^{-1} = \frac{k}{13.41 k_{\mathrm{eq}}} , \tag{10}$$

where α_c and β_c are fitted by

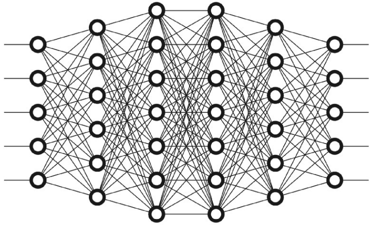
$$\alpha_c = a_1^{-\Omega_b/\Omega_0} a_2^{-(\Omega_b/\Omega_0)^3} ,$$
$$a_1 = (46.9\Omega_0 h^2)^{0.670}[1 + (32.1\Omega_0 h^2)^{-0.532}] ,$$
$$a_2 = (12.0\Omega_0 h^2)^{0.424}[1 + (45.0\Omega_0 h^2)^{-0.582}] , \tag{11}$$
$$\beta_c^{-1} = 1 + b_1[(\Omega_c/\Omega_0)^{b_2} - 1] ,$$
$$b_1 = 0.944[1 + (458\Omega_0 h^2)^{-0.708}]^{-1} ,$$
$$b_2 = (0.395\Omega_0 h^2)^{-0.0266} . \tag{12}$$

As $\Omega_b/\Omega_0 \to 0$, $\alpha_c, \beta_c \to 1$. Equation (9) shows the familiar ln $(k)/k^2$ dependence of the small-scale CDM transfer function.

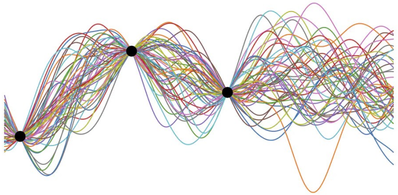
Many options exist instead of using Boltzmann solver

Numerical

Aricò et al. 2021;
Spurio Mancini et al. 2022
Euclid Collaboration et al. 2021



Mootoovaloo et al. 2022



Symbolic

Eisenstein & Hu 1998, 1998
BBKS 1986

The transfer function is written as a sum of the baryon and cold dark matter contributions at the drag epoch

$$T(k) = \frac{\Omega_b}{\Omega_0} T_b(k) + \frac{\Omega_c}{\Omega_0} T_c(k) . \quad (8)$$

The CDM transfer function can be solved exactly in terms of hypergeometric functions that are more conveniently approximated by the following form:

$$T_c \to \alpha_c \frac{\ln 1.8\beta_c q}{14.2q^2} , \quad (9)$$

$$q = \left(\frac{k}{\mathrm{Mpc}^{-1}}\right)\Theta_{2.7}^2(\Omega_0 h^2)^{-1} = \frac{k}{13.41 k_{\rm eq}} , \quad (10)$$

where α_c and β_c are fitted by

$$\alpha_c = a_1^{-\Omega_b/\Omega_0} a_2^{-(\Omega_b/\Omega_0)^3} ,$$
$$a_1 = (46.9\Omega_0 h^2)^{0.670}[1 + (32.1\Omega_0 h^2)^{-0.532}] ,$$
$$a_2 = (12.0\Omega_0 h^2)^{0.424}[1 + (45.0\Omega_0 h^2)^{-0.582}] , \quad (11)$$
$$\beta_c^{-1} = 1 + b_1[(\Omega_c/\Omega_0)^{b_2} - 1] ,$$
$$b_1 = 0.944[1 + (458\Omega_0 h^2)^{-0.708}]^{-1} ,$$
$$b_2 = (0.395\Omega_0 h^2)^{-0.0266} . \quad (12)$$

As $\Omega_b/\Omega_0 \to 0$, $\alpha_c, \beta_c \to 1$. Equation (9) shows the familiar ln $(k)/k^2$ dependence of the small-scale CDM transfer function.

- **Interpretable**
- **Easily incorporate Physics**
- **Portable**
- **Longevity**
- **Fewer parameters**

but …

Not accurate enough

Find symbolic emulators with Symbolic Regression

OPERON

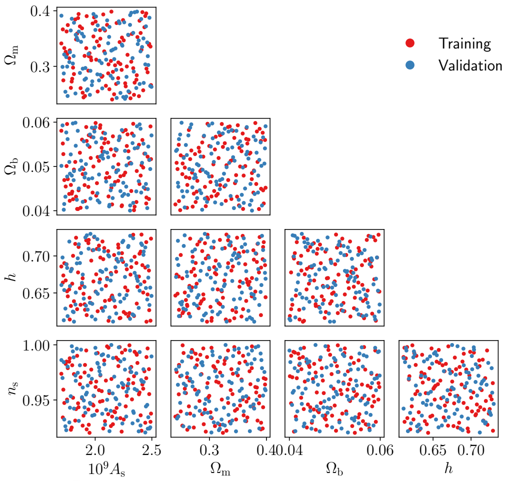
- Multi-Objective Genetic Programming
- Fast
- Memory efficient
- High performance on benchmarks
- C++ with python wrapper

Find symbolic emulators with Symbolic Regression

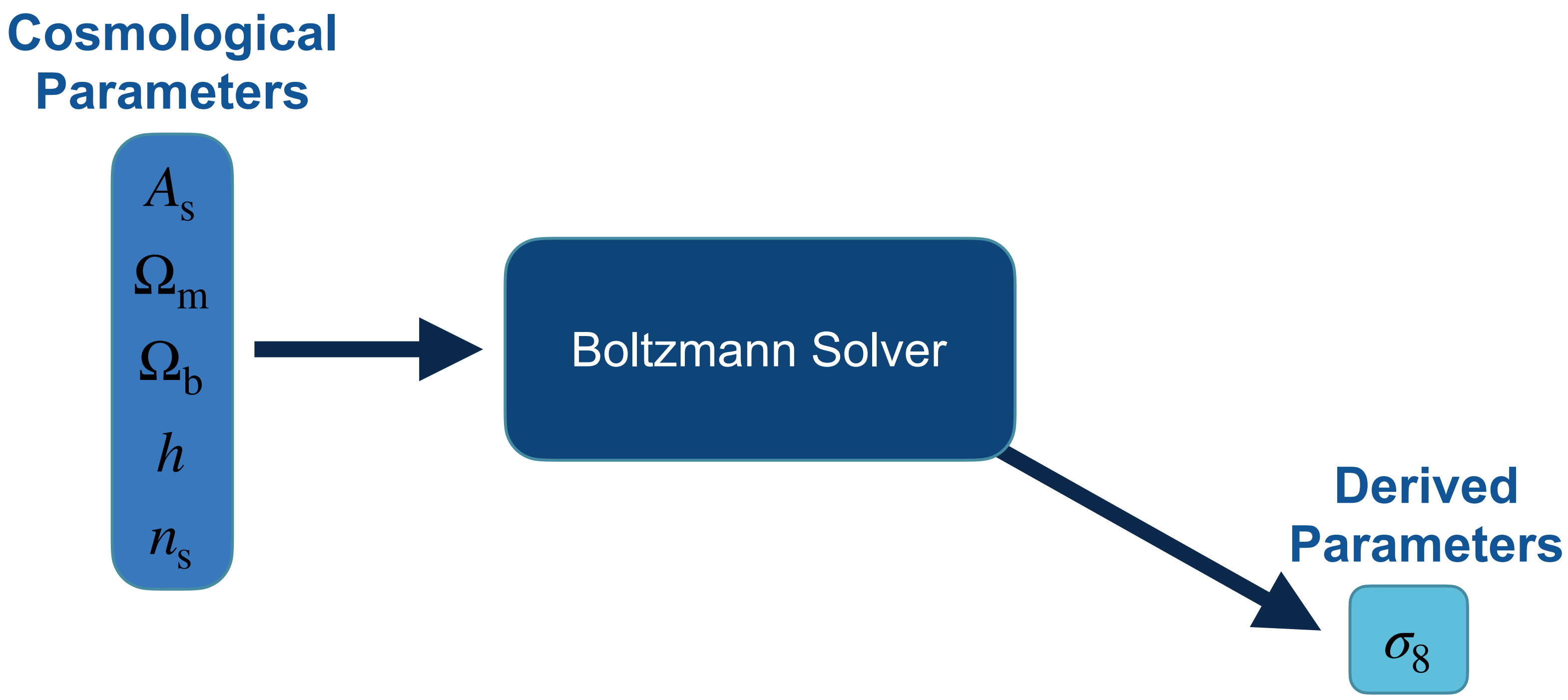
OPERON

- Multi-Objective Genetic Programming
- Fast
- Memory efficient
- High performance on benchmarks
- C++ with python wrapper

- 100 training, 100 validation cosmologies on latin hypercube (small number compared to e.g. neural networks)
- Truth computed with CAMB



We don’t always want to use a Boltzmann solver



Task: Map $\sigma_8 \leftrightarrow A_s$ given other cosmological parameters

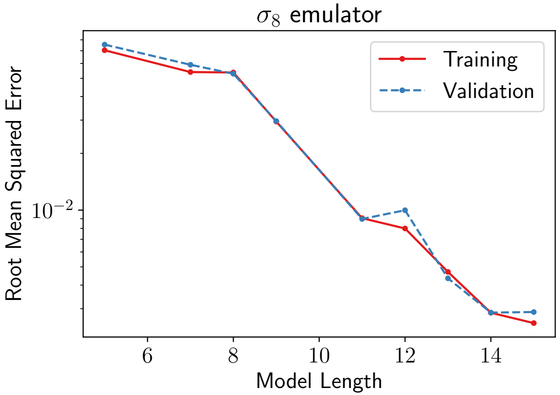
Normally: Run a Boltzmann solver with a guess of A_s and measure σ_8

Task: Map $\sigma_8 \leftrightarrow A_s$ given other cosmological parameters

Normally: Run a Boltzmann solver with a guess of A_s and measure σ_8

Now:

$$\sigma_8 \approx \left(a_0 A_s + a_1 n_s\right)\left(a_2 \Omega_b + \log\left(a_3 \Omega_m\right)\right) \log\left(a_4 h\right) + a_5$$



- One simple line
- RMSE of 0.3% across range of cosmologies

We don’t always want to use a Boltzmann solver

Cosmological Parameters

A_s

Ω_m

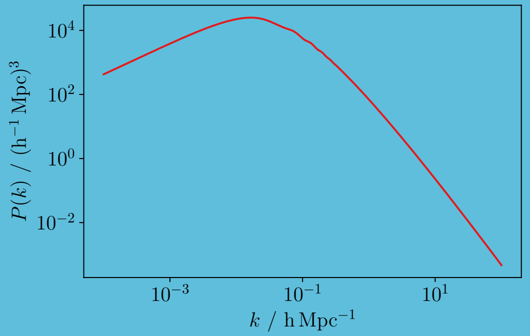
Ω_b

h

n_s

Boltzmann Solver

Power Spectra



Eisenstein & Hu did the hard work! Fit the residuals

$$P(k, \theta) = P_{\mathrm{EH}}(k, \theta) F(k, \theta)$$

Eisenstein & Hu did the hard work! Fit the residuals

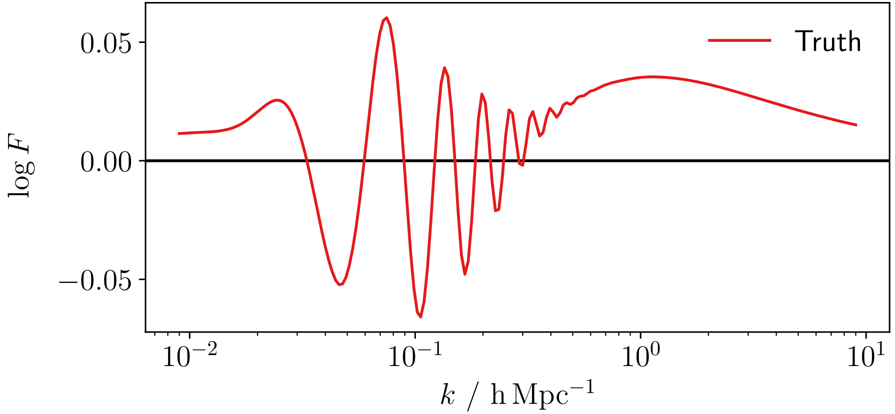
$$P(k, \theta) = P_{\mathrm{EH}}(k, \theta) F(k, \theta)$$

Physics!

- ~~Baryonic acoustic oscillations~~
- Compton drag
- Velocity overshoot
- Baryon infall
- Adiabatic damping
- Silk damping
- Cold dark matter growth suppression

Eisenstein & Hu did the hard work! Fit the residuals

$$P(k, \theta) = P_{\mathrm{EH}}(k, \theta) F(k, \theta)$$



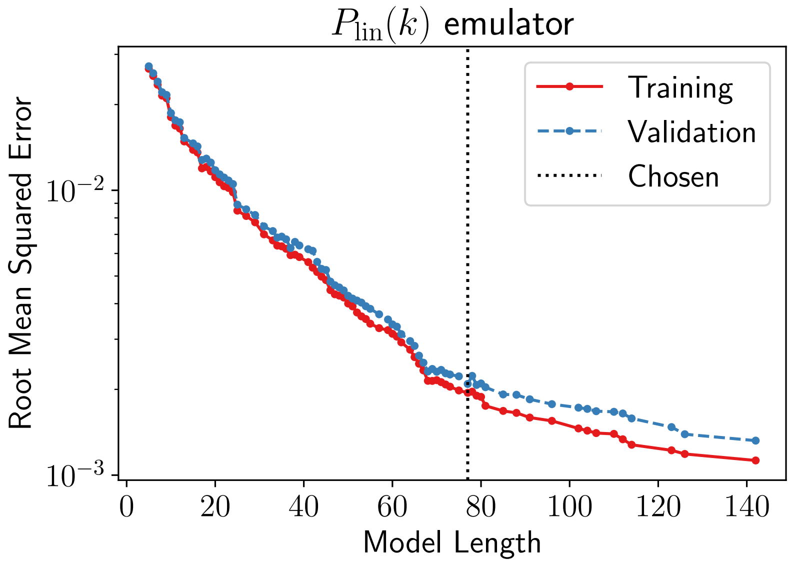
Physics!

- ~~Baryonic acoustic oscillations~~
- Compton drag
- Velocity overshoot
- Baryon infall
- Adiabatic damping
- Silk damping
- Cold dark matter growth suppression

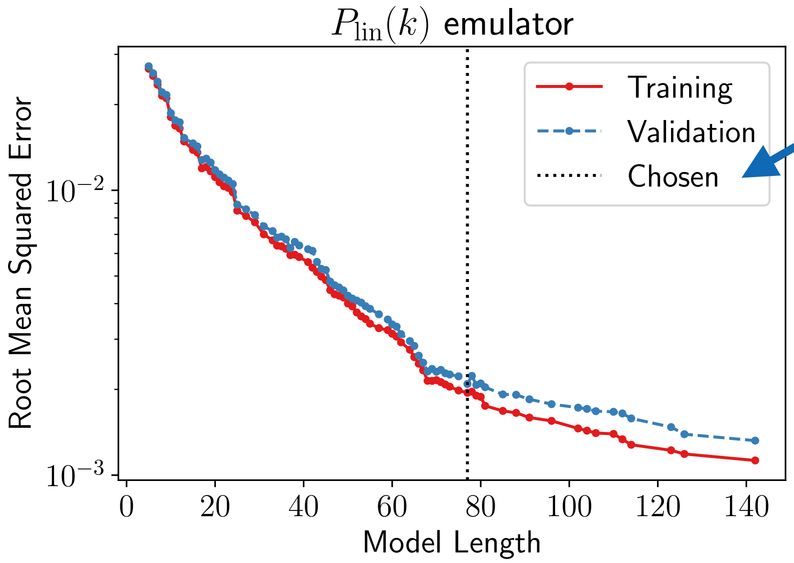
The rest

We will learn $\log F$

Percent-level fit achieved with (relatively) short expressions

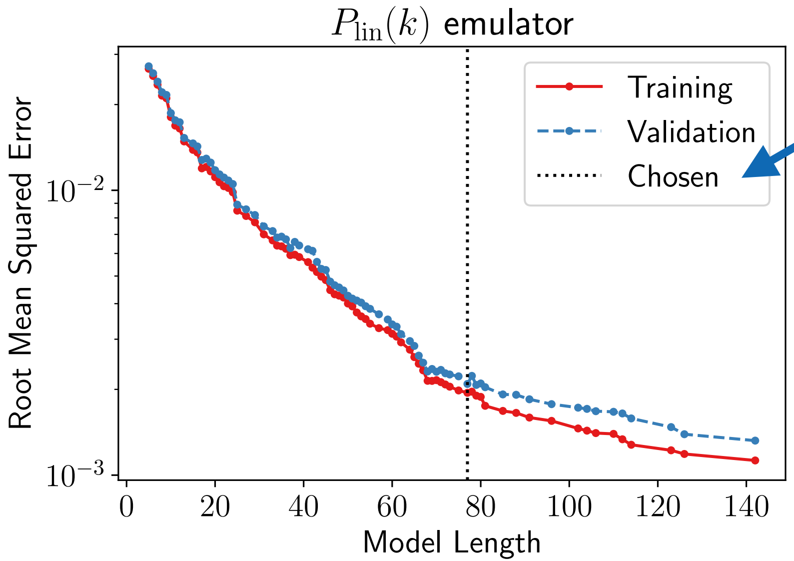


Percent-level fit achieved with (relatively) short expressions



Sufficient for % level accuracy

Percent-level fit achieved with (relatively) short expressions

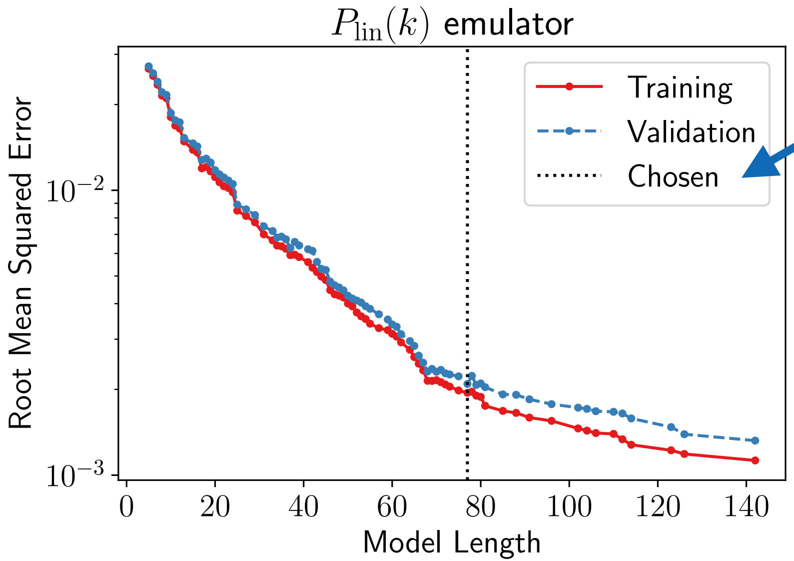


Sufficient for % level accuracy

Physically interpretable:

$$\sim \cos\left(b_{10}\Omega_{\rm m} - \frac{b_{11}k}{\sqrt{b_{12}+\Omega_{\rm b}^2}}\right)$$

Percent-level fit achieved with (relatively) short expressions



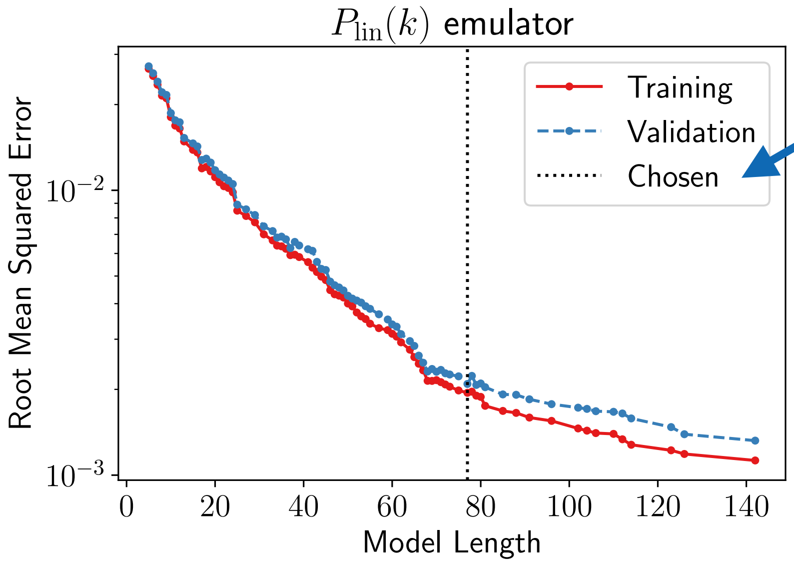
Sufficient for % level accuracy

Physically interpretable:

$$\sim \cos\left(b_{10}\Omega_{\rm m} - \frac{b_{11}k}{\sqrt{b_{12}+\Omega_{\rm b}^2}}\right)$$

BAO phase shift depends on $\Omega_{\rm m}$

Percent-level fit achieved with (relatively) short expressions



Sufficient for % level accuracy

Physically interpretable:

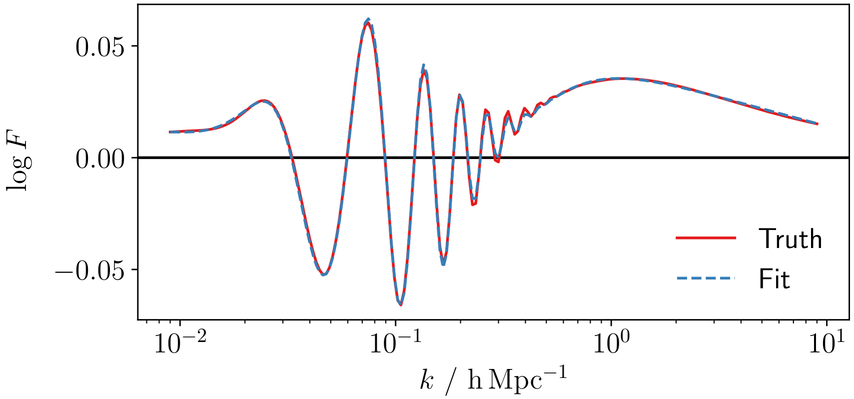
$$\sim \cos\left(b_{10}\Omega_{\rm m} - \frac{b_{11}k}{\sqrt{b_{12}+\Omega_{\rm b}^2}}\right)$$

BAO phase shift depends on $\Omega_{\rm m}$

More BAO cycles in given k range for larger $\Omega_{\rm b}$

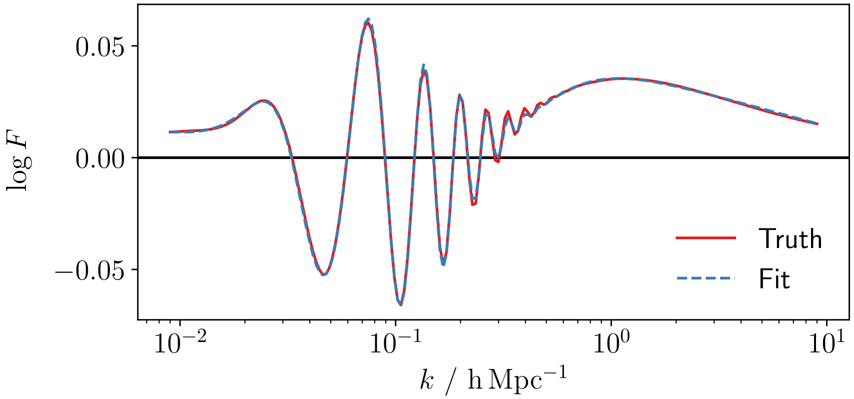
Symbolic emulator with sub-percent accuracy (RMSE 0.2%)

Fit almost perfect to the eye:

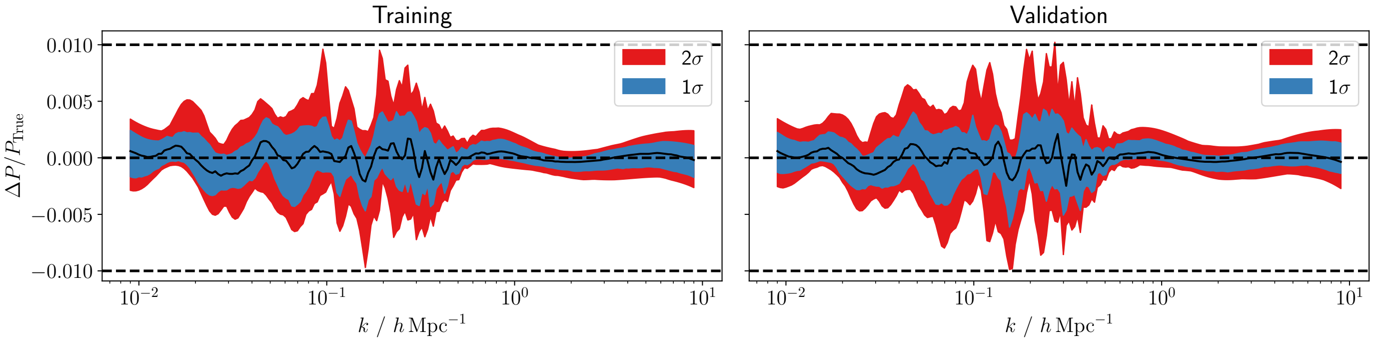


Symbolic emulator with sub-percent accuracy (RMSE 0.2%)

Fit almost perfect to the eye:

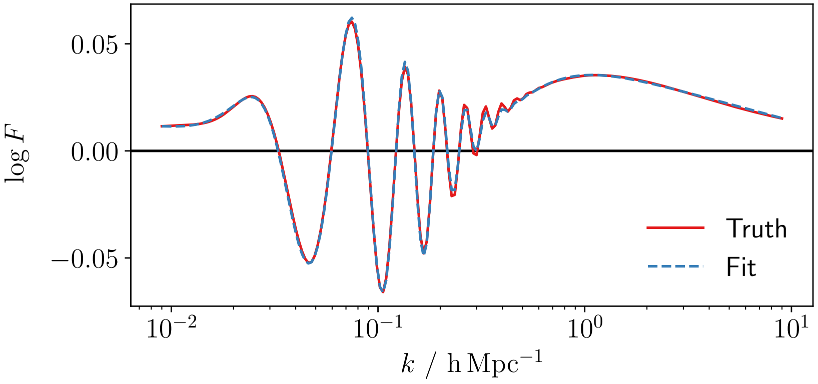


Sub-percent accuracy across training and validation:



Conclusions

- Symbolic emulator for linear $P(k)$ with sub-percent accuracy
- One line approximation for $\sigma_8 \leftrightarrow A_s$ with RMSE of 0.3%
- Do I *really* need a neural network?
    - Portability
    - Interpretability
    - Longevity
    - Fewer parameters



Extra Slides

Symbolic emulator for non-linear power spectrum

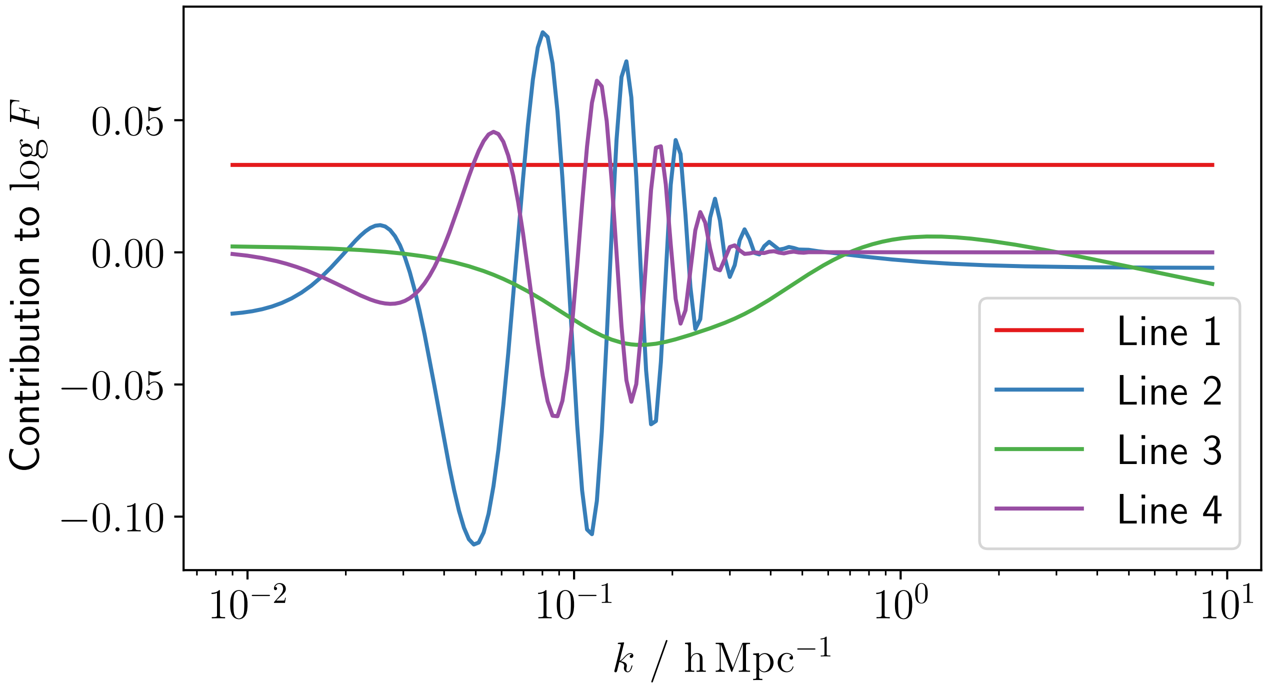
$$\log F \approx b_0 h - b_{26}$$

$$+\left(\frac{b_1\Omega_\mathrm{b}}{\sqrt{h^2+b_2}}\right)^{b_3\Omega_\mathrm{m}}\left[\frac{b_4 k-\Omega_\mathrm{b}}{\sqrt{b_5+(\Omega_\mathrm{b}-b_6 k)^2}} b_7 (b_8 k)^{-b_9 k}\cos\left(b_{10}\Omega_\mathrm{m}-\frac{b_{11}k}{\sqrt{b_{12}+\Omega_\mathrm{b}^2}}\right)-b_{13}\left(\frac{b_{14}k}{\sqrt{1+b_{15}k^2}}-\Omega_\mathrm{m}\right)\cos\left(\frac{b_{16}h}{\sqrt{1+b_{17}k^2}}\right)\right]$$

$$+b_{18}(b_{19}\Omega_\mathrm{m}+b_{20}h-\log(b_{21}k)+(b_{22}k)^{-b_{23}k})\cos\left(\frac{b_{24}}{\sqrt{1+b_{25}k^2}}\right)$$

$$+(b_{27}k)^{-b_{28}*k}\left(b_{29}k-\frac{b_{30}\log(b_{31}k)}{\sqrt{b_{32}+(\Omega_\mathrm{m}-b_{33}h)^2}}\right)\cos\left(b_{34}\Omega_\mathrm{m}-\frac{b_{35}k}{\sqrt{b_{36}+\Omega_\mathrm{b}^2}}\right)$$

Contributions to $P(k)$ emulator



Residuals for Planck 2018 Cosmology