Differentiable Cosmological Simulation with Adjoint Method

Yin Li (Peng Cheng Laboratory)

with Yucheng Zhang (PCL/NYU), Libin Lu (Flatiron), Chirag Modi (Flatiron), Drew Jamieson (MPA), Yu Feng (Google), Francois Lanusse (CEA Saclay), and Leslie Greengard (Flatiron/NYU)

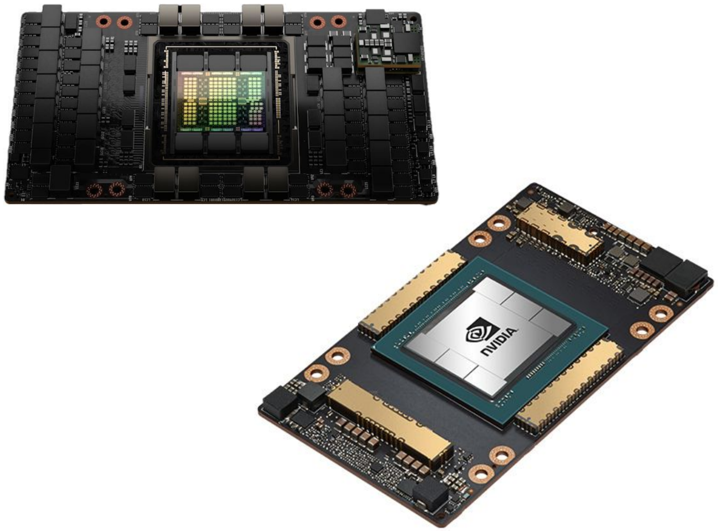
2211.09815, 2211.09958

Suzhou Bay 2023-06-22


Credit: Kaze Wong (Flatiron)

Thanks to the Deep Learning Revolution

Hardware



Software

PyTorch

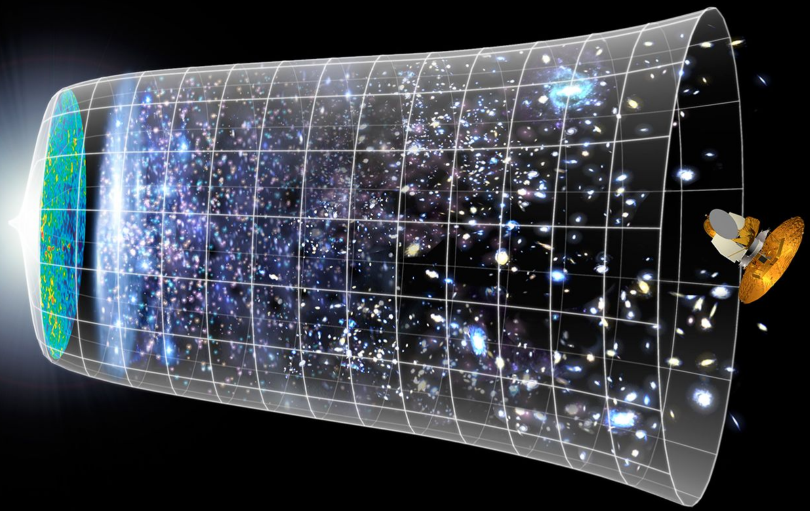
TensorFlow

julia

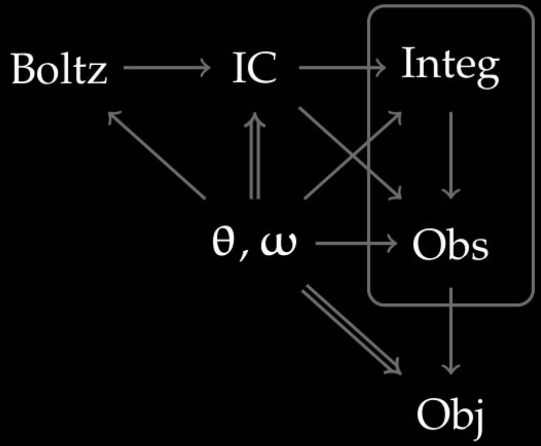
Differentiability: what & why

- Specifically, the gradient of the objective/posterior to the input parameters
  - cosmological + ICs + astrophysical + foregrounds + systematics
  - forward model at field level, great for joint SBI/LFI of multiple probes while avoiding summary statistics and covariances
- Optimization
  - Backpropagation (automatic differentiation, a special case of the Adjoint Method)
- Inference
  - Sampling (MCMC): Hamiltonian Monte Carlo & Langevin Monte Carlo
  - Variational Inference (VI): inference by optimization
  - NN+Sampling, NN+VI (Modi, YL, & Blei 2023)
- Future applications
  - Can be combined with other differentiable models: semi-analytic, empirical, NNs
  - Baseline for new physics (PNG and dark matter) and galaxy formation

Cosmological Forward Model

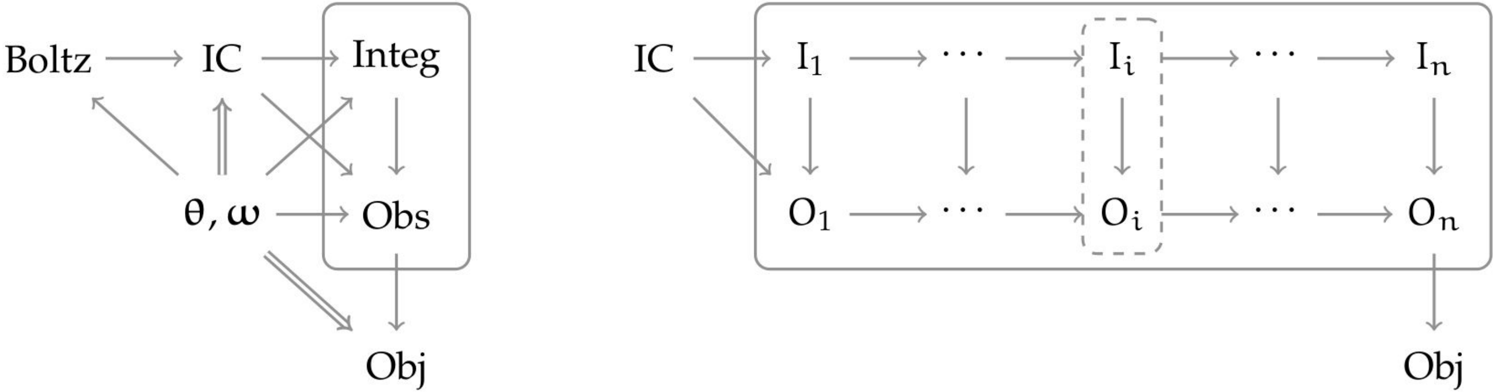


NASA/WMAP Science Team



(a) model structure

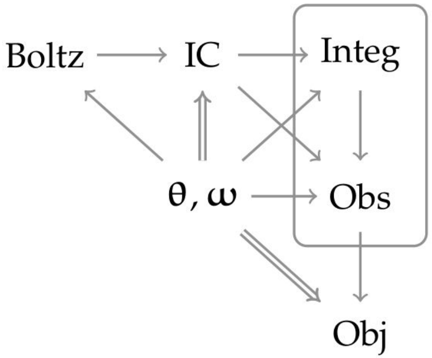
Forward to Model



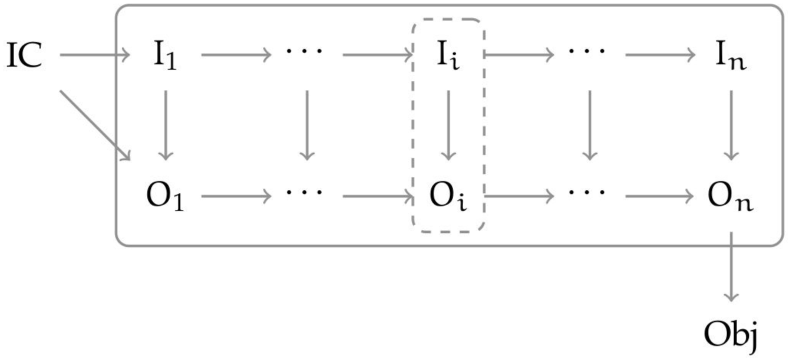
(a) model structure

(b) integration-observation loop

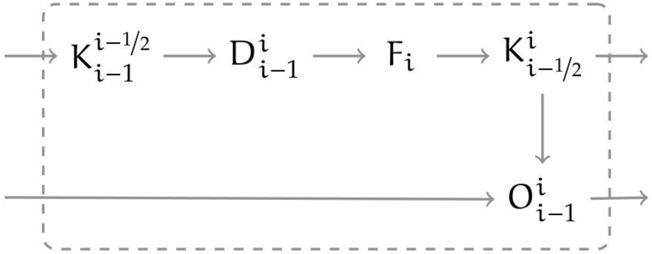
Forward to Model



(a) model structure

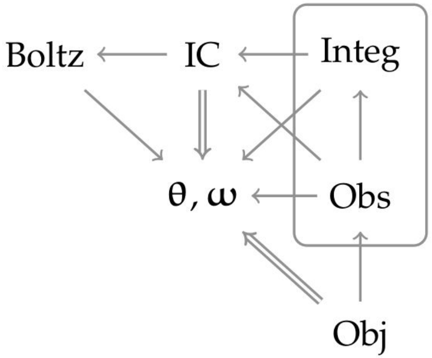


(b) integration-observation loop

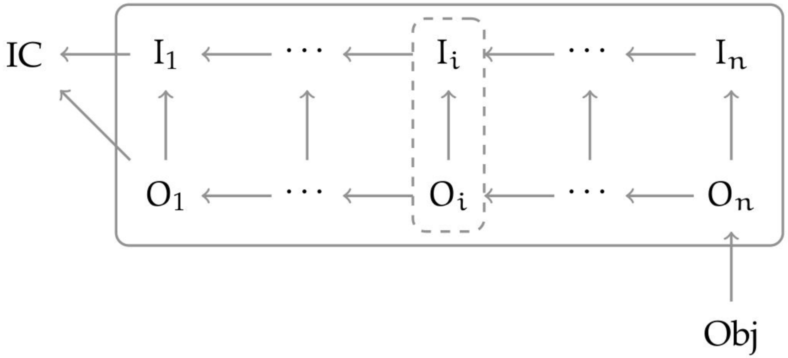


(c) i-th (leapfrog) integration-observation step

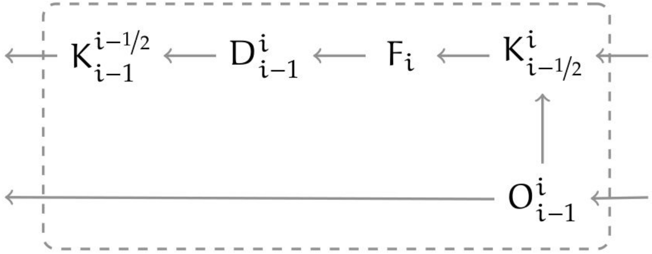
Backward for Inference or Optimization



(a) model structure



(b) integration-observation loop



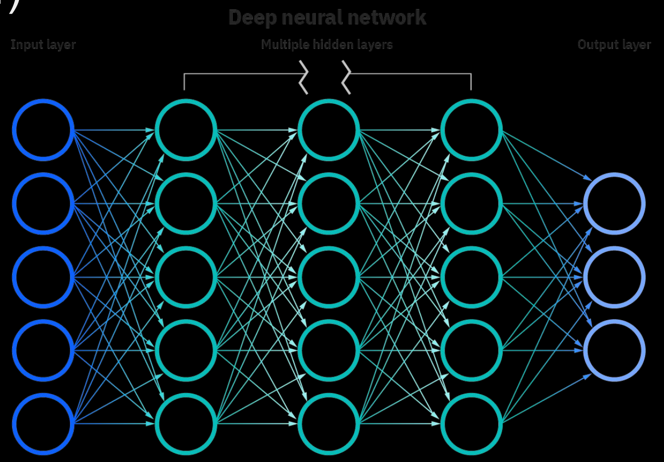
(c) i-th (leapfrog) integration-observation step

Existing differentiable cosmological simulation codes

- `BORG` (Jasche & Wandelt 2013)
- `ELUCID` (Wang et al. 2014, …)
- `FastPM+vmad` (Feng et al. 2016 & Seljak et al. 2017, …)
- `BORG-PM` (Jasche & Lavaux 2019, …)
- `FlowPM` (Modi et al. 2020)
- ...

Existing differentiable cosmological simulation codes

- `BORG` (Jasche & Wandelt 2013)
- `ELUCID` (Wang et al. 2014, …)
- `FastPM+vmad` (Feng et al. 2016 & Seljak et al. 2017, …)
- `BORG-PM` (Jasche & Lavaux 2019, …)
- `FlowPM` (Modi et al. 2020)
- …



Trade-off between spatial and time complexity

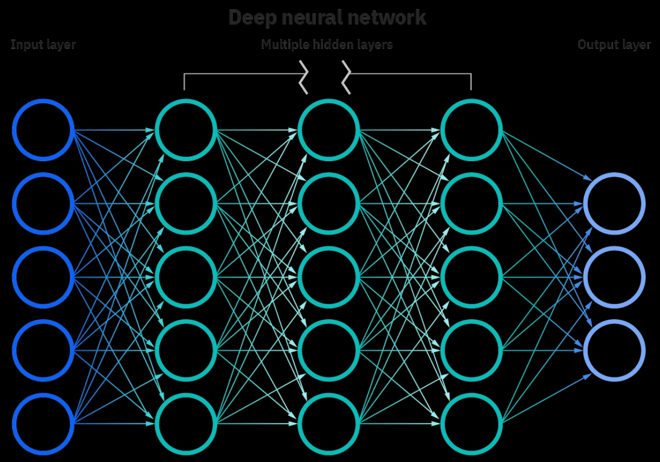
Existing differentiable cosmological simulation codes

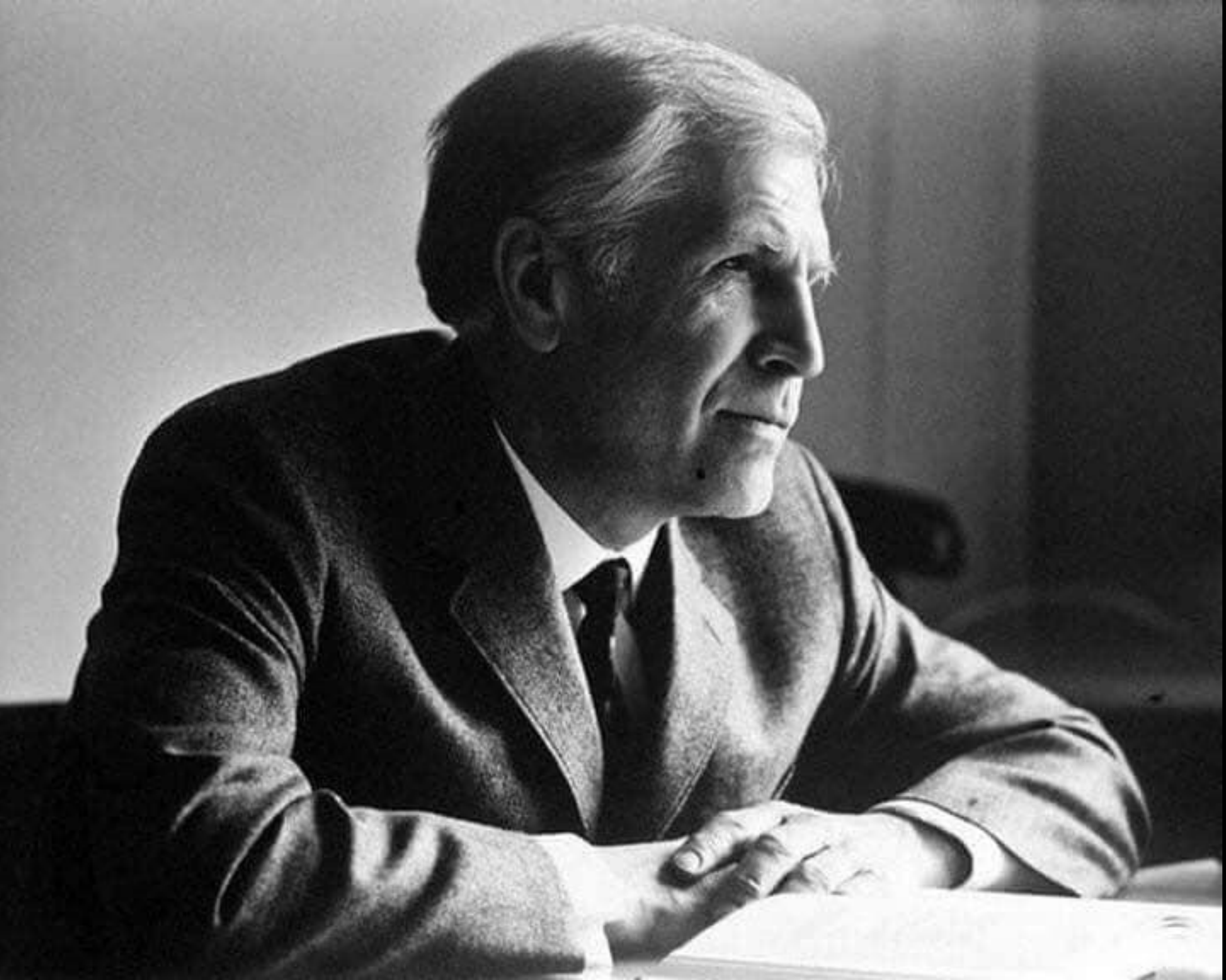
- `BORG` (Jasche & Wandelt 2013)
- `ELUCID` (Wang et al. 2014, …)
- `FastPM+vmad` (Feng et al. 2016 & Seljak et al. 2017, …)
- `BORG-PM` (Jasche & Lavaux 2019, …)
- `FlowPM` (Modi et al. 2020)
- …

~~Trade-off between spatial and time complexity~~

Adjoint method: optimization under constraint

Lev Pontryagin 1956, also cf. Neural ODE





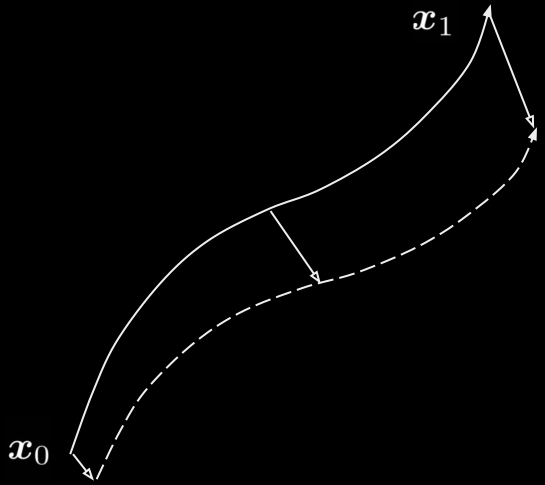
Adjoint method 1950s

Automatic differentiation 1960s~1970s

Backprop popularized for training NNs 1980s

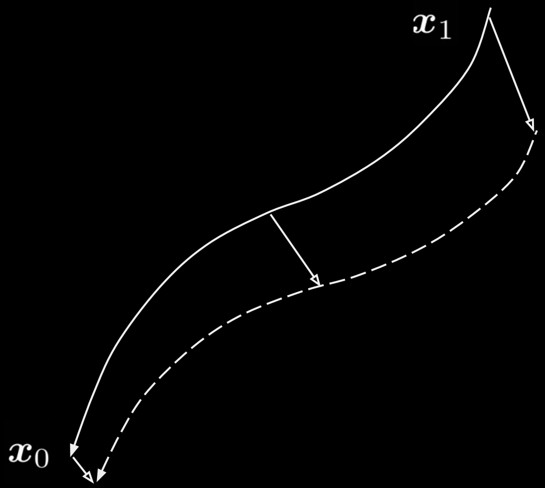
Deep learning revolution 2010s

Algorithm Summary: Forward



$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{F}_k(\boldsymbol{x}_k, \boldsymbol{\theta}), \qquad k = 0, \cdots, n-1,$$

Algorithm Summary: Backward



$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{F}_k(\boldsymbol{x}_k, \boldsymbol{\theta}), \qquad k = 0, \cdots, n-1,$$

$$\boldsymbol{\lambda}_{k-1} = \boldsymbol{\lambda}_k + \boldsymbol{\lambda}_k \cdot \frac{\partial \boldsymbol{F}_{k-1}}{\partial \boldsymbol{x}_{k-1}}, \qquad \boldsymbol{\lambda}_n = -\frac{\partial \mathcal{J}}{\partial \boldsymbol{x}_n},$$

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\boldsymbol{\theta}} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}} - \boldsymbol{\lambda}_0 \cdot \frac{\partial \boldsymbol{x}_0}{\partial \boldsymbol{\theta}} - \sum_{k=1}^{n} \boldsymbol{\lambda}_k \cdot \frac{\partial \boldsymbol{F}_{k-1}}{\partial \boldsymbol{\theta}}.$$

pmwd: Differentiable Particle-Mesh Library

- A new code based on `JAX`
  - ✓ fast sim on GPU
  - ✓ 2x memory & 3~4x runtime with derivatives
  - ✓ open sourced at github.com/eelregit/pmwd

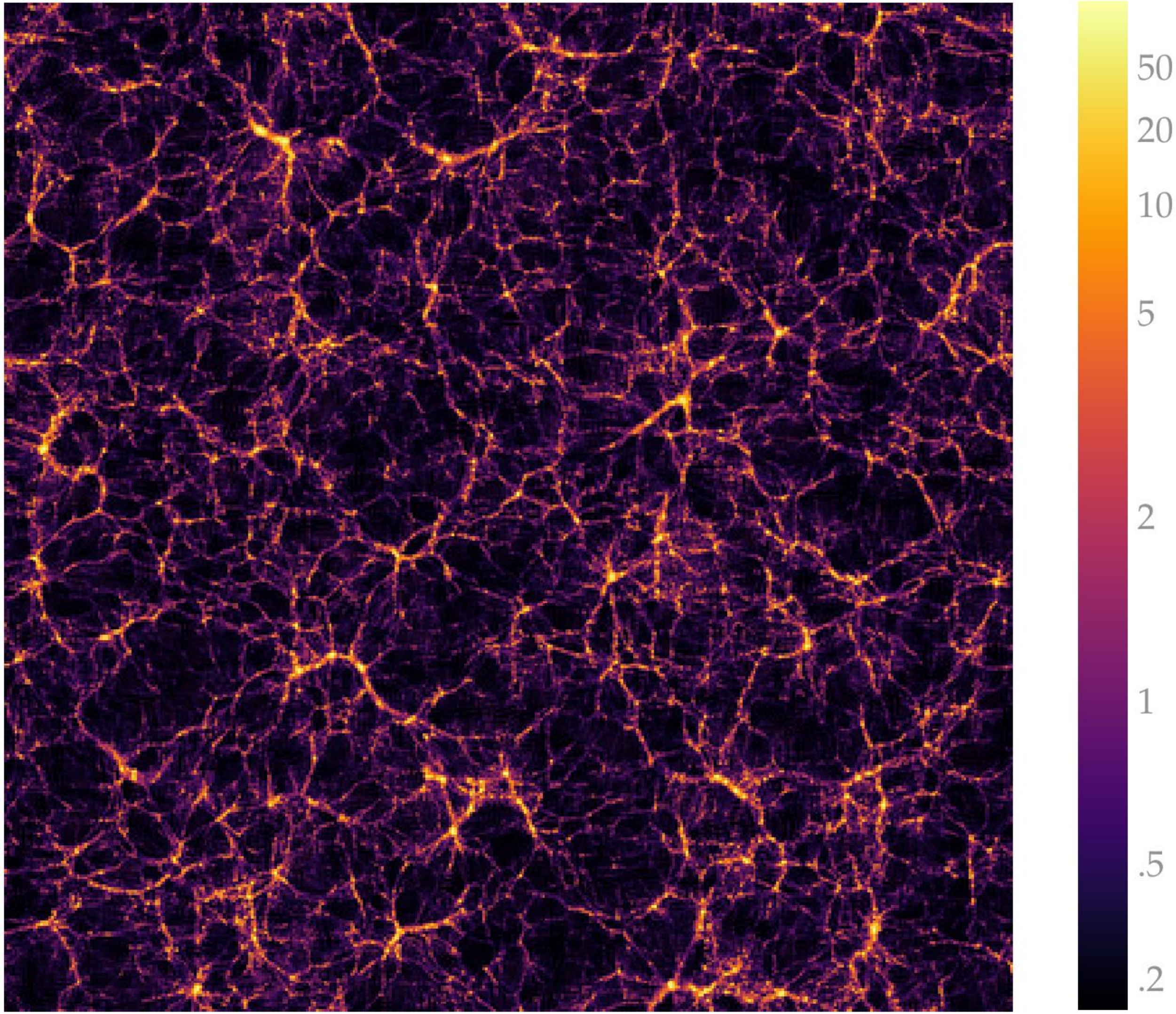
20s to run this:

512^3 particles

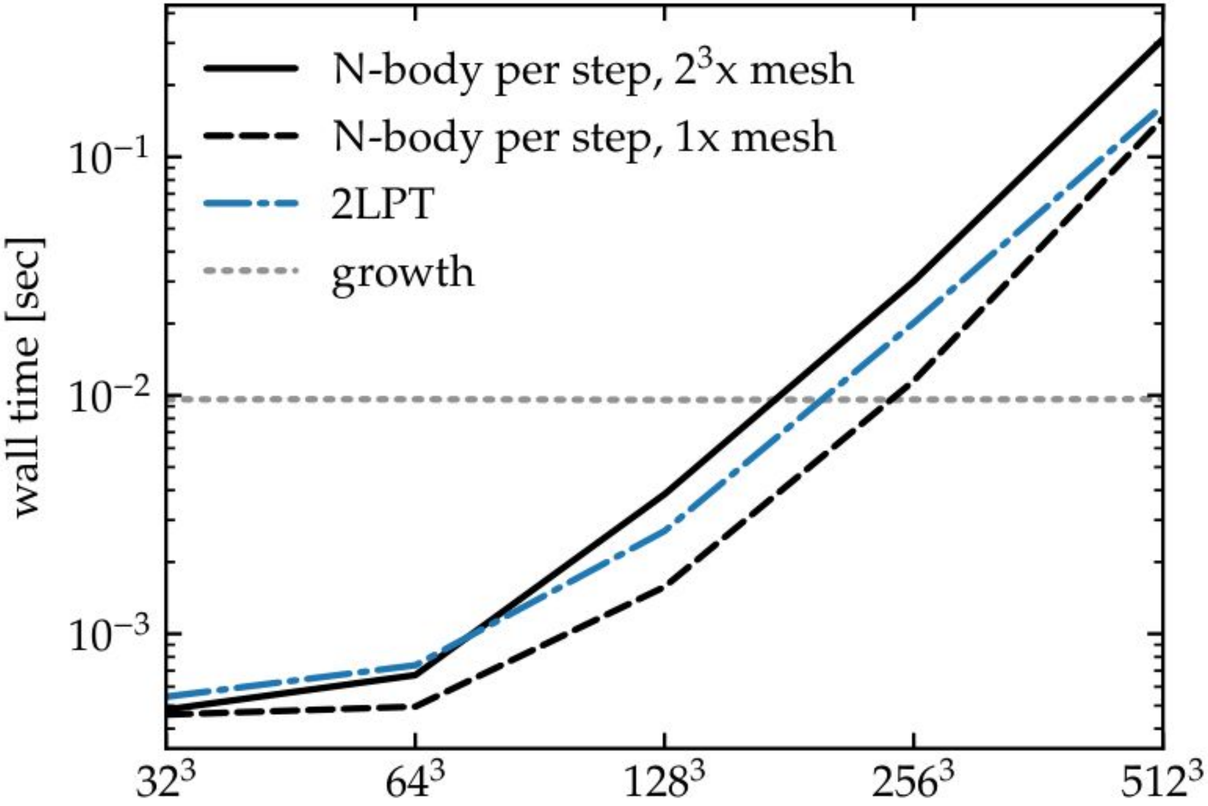
1024^3 mesh

63 time steps

1 A100 80GB SXM4



Performance



Alternative to Established Workflows in Cosmology

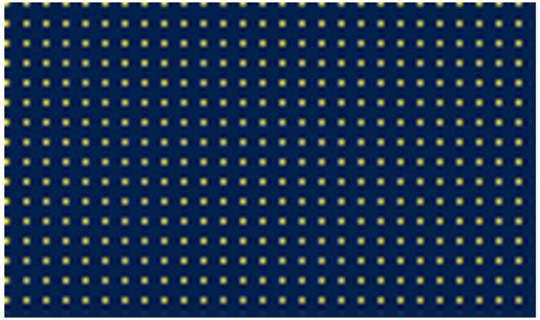
- Data compression into summary statistics:
  - Power spectra or 2-point correlation functions
  - Polyspectra or N-point correlation functions
  - Histograms/PDFs, skewness, kurtosis, …
  - Abundances of halos, galaxies, …

  Risk of information loss

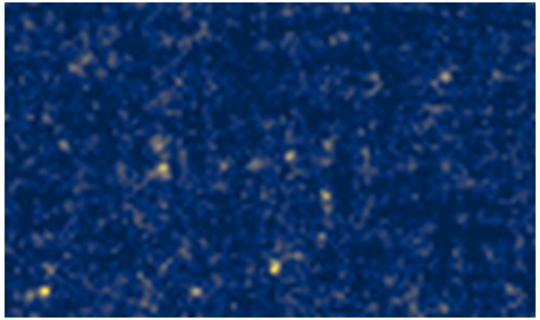
- Modeling and Inference
  - Perturbation theories (EFTs), Emulators calibrated with simulations
  - MCMC assuming Gaussian likelihood

  Difficulties on nonlinearity, cross-correlation, cross-covariance, Gaussianity of likelihood

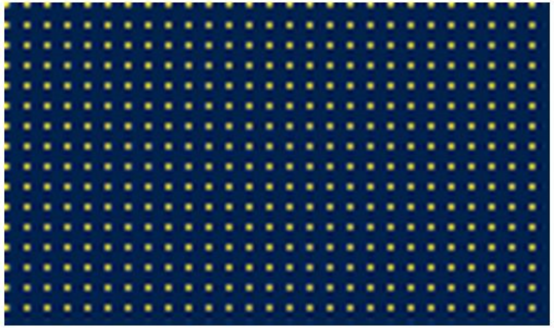
A Toy Optimization Problem



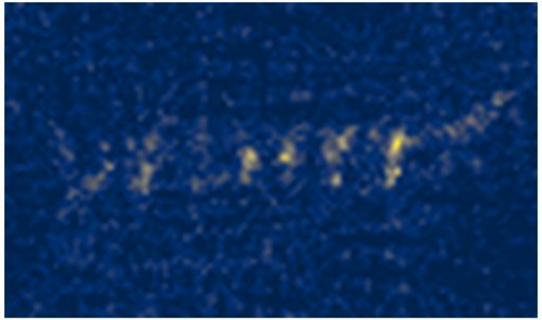
(a) initial conditions at $a = 1/64$



(b) final snapshot at $a = 1$



(c) reverse evolution back to $a = 1/64$

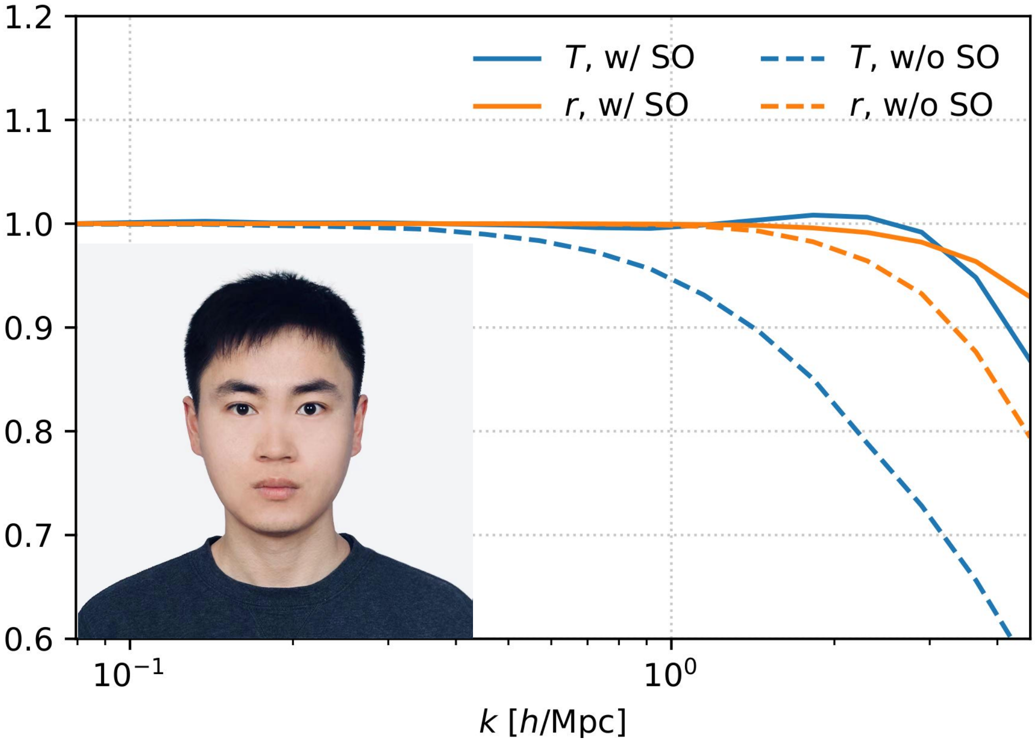


(d) optimization for 10 iterations at $a = 1$

(e) optimization for 100 iterations at $a = 1$

(f) optimization for 1000 iterations at $a = 1$

Spatial optimization (SO) to match Gadget-4 under high accuracy settings



"sharpen" the PM force:

$$\frac{ik_i}{k^2} \to \frac{ik_i}{k^2} f(k_i;\vartheta) g(k_1, k_2, k_3;\vartheta),$$

quantify accuracy by:

$$T(\mathbf{k}) \triangleq \sqrt{\frac{|\hat{\delta}(\mathbf{k})|^2}{|\delta(\mathbf{k})|^2}}, \quad r(\mathbf{k}) \triangleq \frac{\Re\left[\hat{\delta}(\mathbf{k})\delta^*(\mathbf{k})\right]}{\sqrt{|\hat{\delta}(\mathbf{k})|^2|\delta(\mathbf{k})|^2}}.$$

2k CPU-hours vs O(100s) of GPU-ms

Applications

- Fast mocks
  - Covariance for established workflows with summary statistics
  - Simulation-based inference (Likelihood free inference, implicit inference)
- Field-level inference (explicit inference) for cosmology
  - marginalizing over initial conditions, astrophysics, etc.
- Inferring initial conditions
  - fixing/marginalizing over cosmology, astrophysics, etc.
  - explicit inference
  - implicit inference?

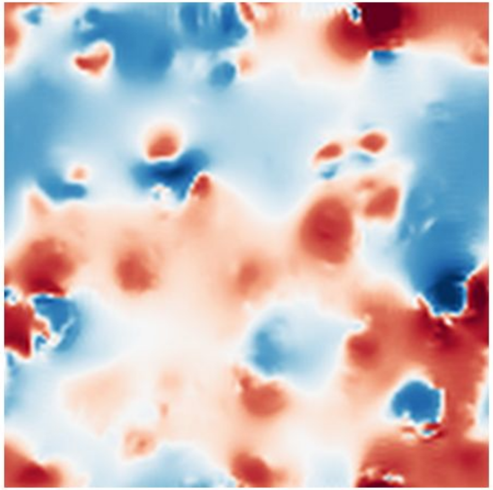
Summary

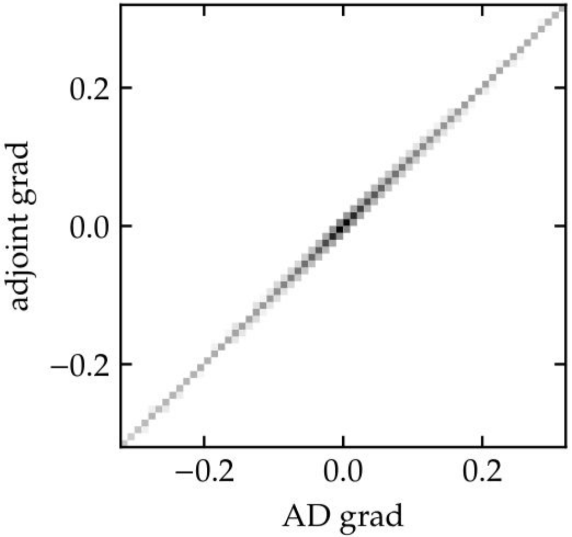
- Efficient differentiation in cosmological simulations, only $O(1)$x increase in space & time complexities
- pmwd: fast differentiable particle-mesh simulation code on GPUs
- Many future development and potential applications

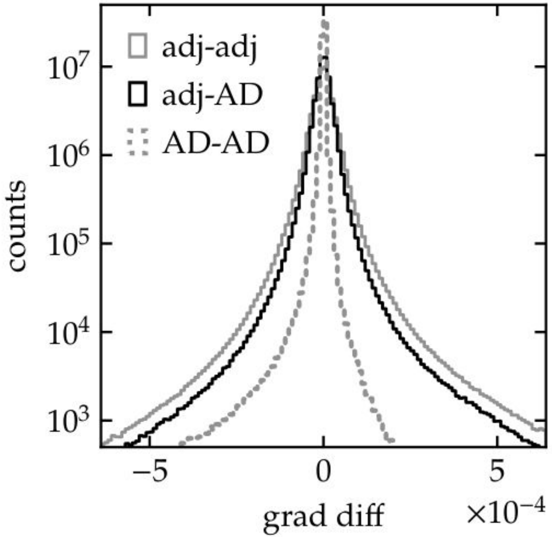
Future development and applications

- Works in progress/planning:
  - short-range PP force
  - distributed parallelization on GPUs: 2 ways, sCOLA & direct
  - various cosmological probes: weak lensing, galaxies, Lya skewers, intensity mapping, etc.
  - inference algorithms in high-dimensional space
- Differentiable simulations as base for cosmo. & astrophys. models
  - differentiable dark matter-observable connection à la biasing/HOD/SAM/etc.
  - differentiable foregrounds and systematics
  - from parameterized models to neural networks
- New physics
  - Add massive neutrinos
  - Baseline for inflation models, dark matters, etc
- Galaxy formation simulations?

Accuracy of the Gradients: Comparison with Automatic Differentiation







Reversibility

precision	cell/ptcl	ptcl mass [$10^{10} M_\odot$]	time steps	disp rel diff	vel rel diff
single	8	1	63	5.2×10^{-2}	7.1×10^{-2}
single	8	1	126	2.1×10^{-2}	3.6×10^{-2}
single	8	8	63	3.3×10^{-3}	7.6×10^{-3}
single	8	8	126	3.7×10^{-3}	7.0×10^{-3}
single	1	1	63	1.4×10^{-3}	2.2×10^{-3}
single	1	1	126	1.3×10^{-3}	1.7×10^{-3}
single	1	8	63	4.3×10^{-4}	7.3×10^{-4}
single	1	8	126	4.4×10^{-4}	6.3×10^{-4}
double	8	1	63	5.4×10^{-11}	1.3×10^{-10}
double	8	1	126	3.5×10^{-11}	7.0×10^{-11}
double	8	8	63	5.8×10^{-12}	1.4×10^{-11}
double	8	8	126	6.4×10^{-12}	1.2×10^{-11}
double	1	1	63	2.2×10^{-12}	3.4×10^{-12}
double	1	1	126	2.1×10^{-12}	2.9×10^{-12}
double	1	8	63	7.2×10^{-13}	1.2×10^{-12}
double	1	8	126	6.9×10^{-13}	9.4×10^{-13}