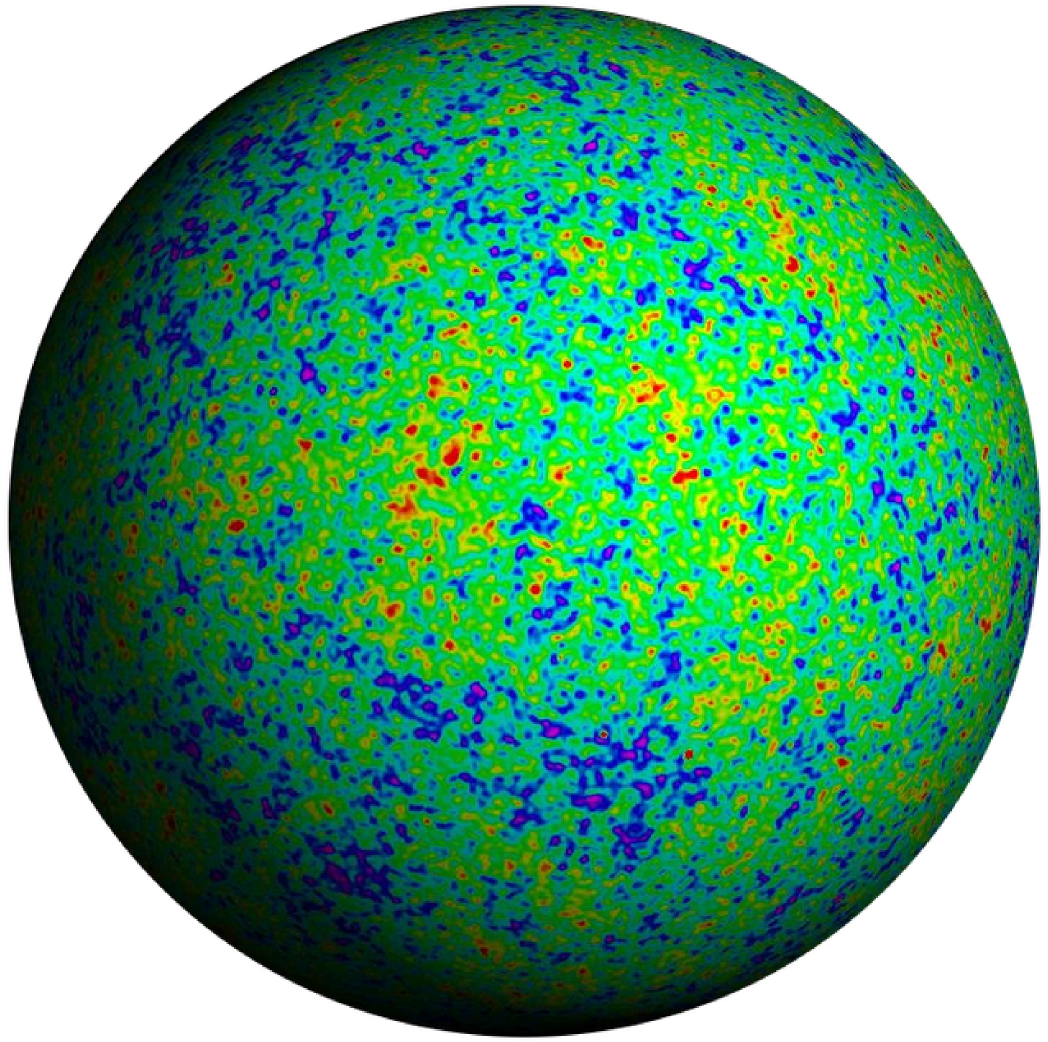
Geometric deep learning for atmospheric science

Jason D. McEwen
www.jasonmcewen.org

Mission Director, Fundamental Research, Alan Turing Institute
Scientific AI Team, Department Space & Climate Physics, UCL

Physics-Informed Machine Learning for Atmospheres, Exeter, July 2025

Cosmological and planetary data live on the sphere

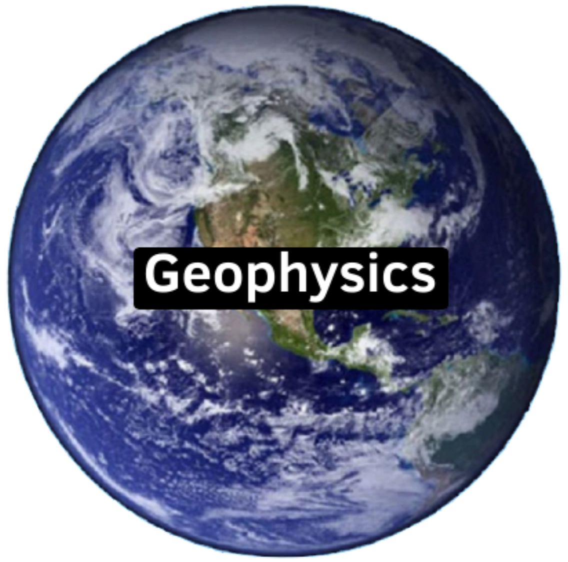


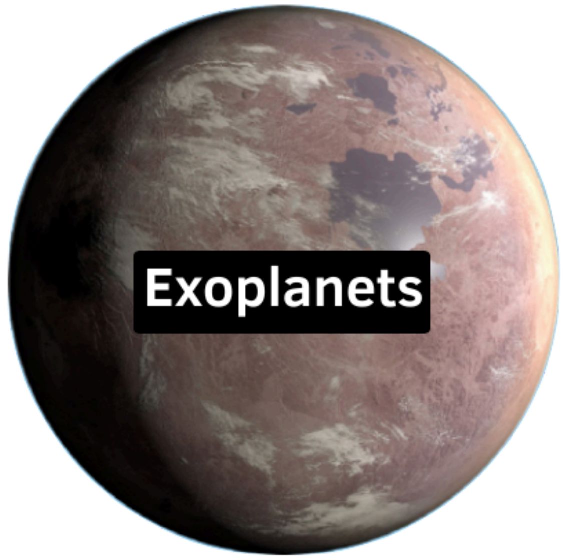
Cosmic microwave background

Earth observation

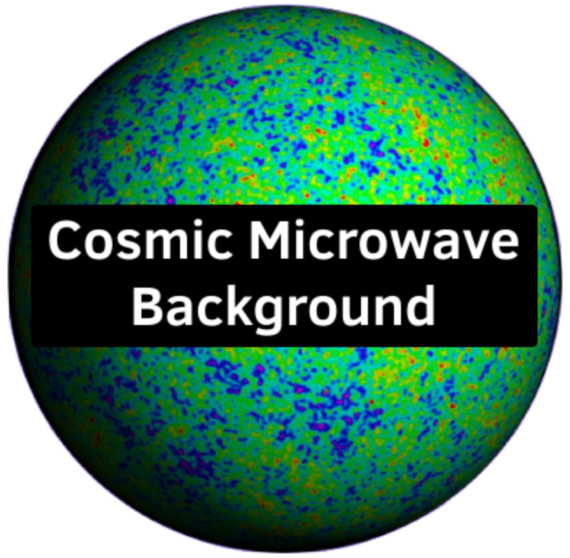
Data observed on the sphere are prevalent

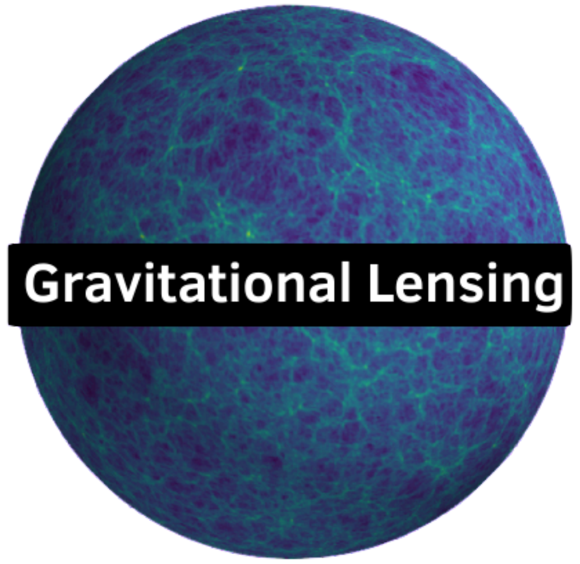
Planetary Science



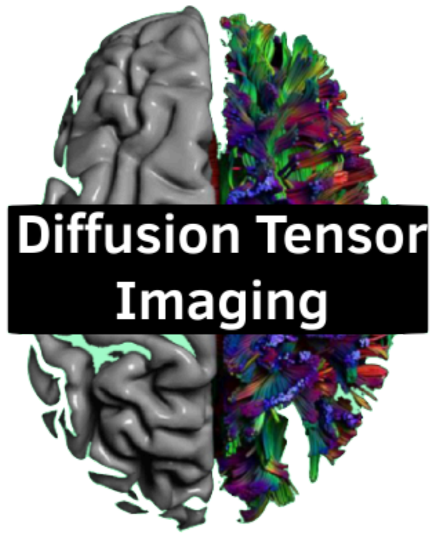


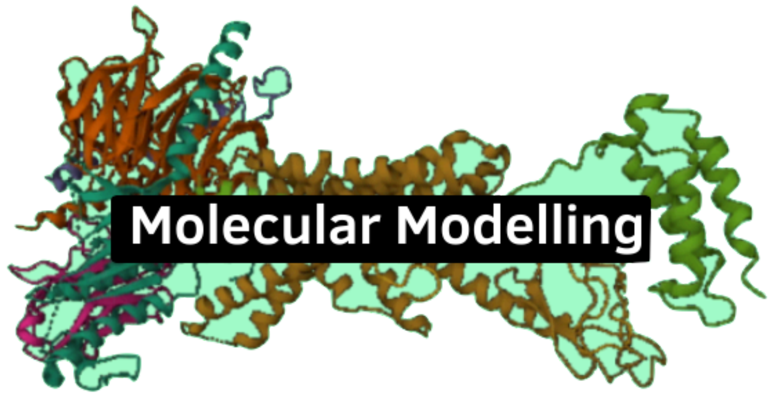
Cosmology





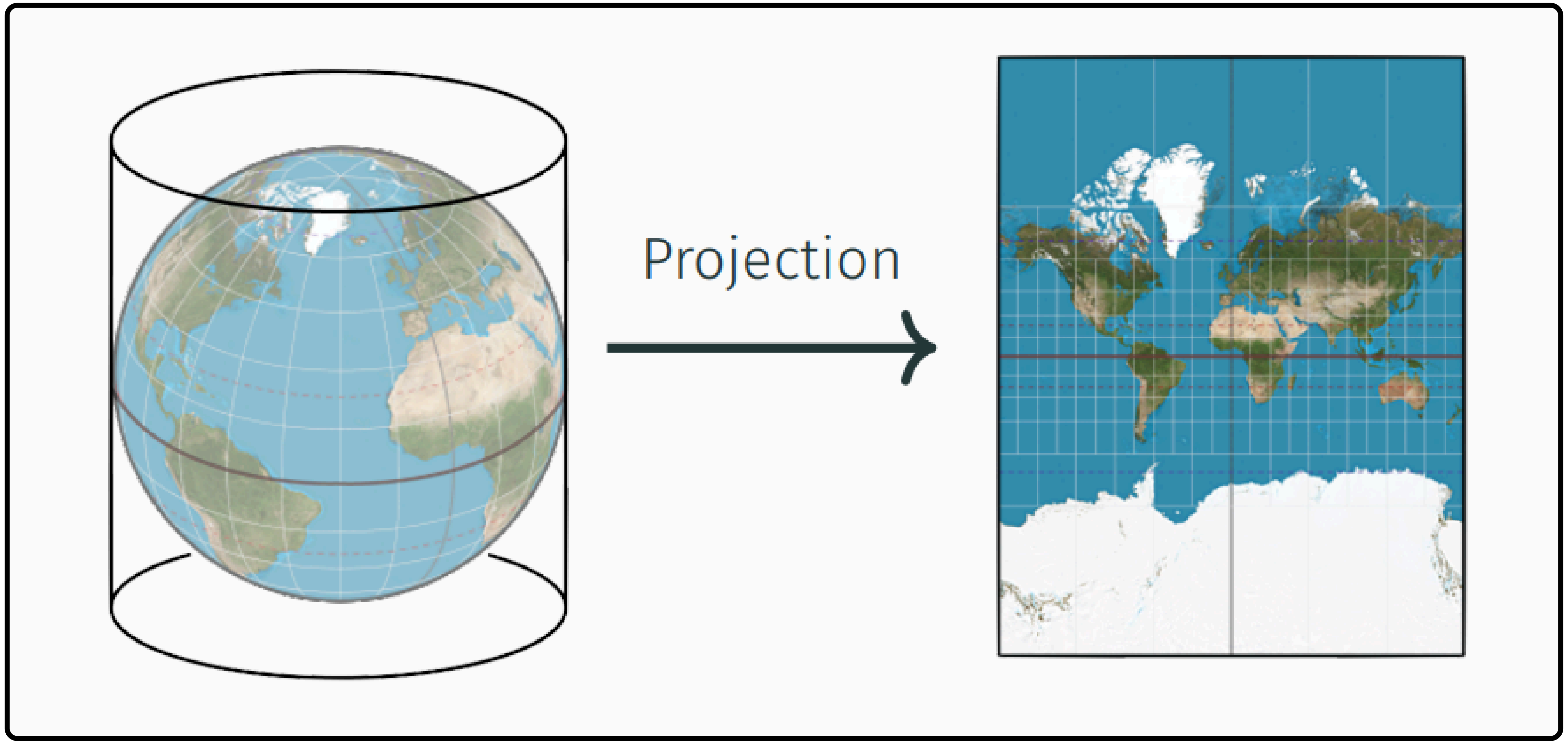
Bioinformatics



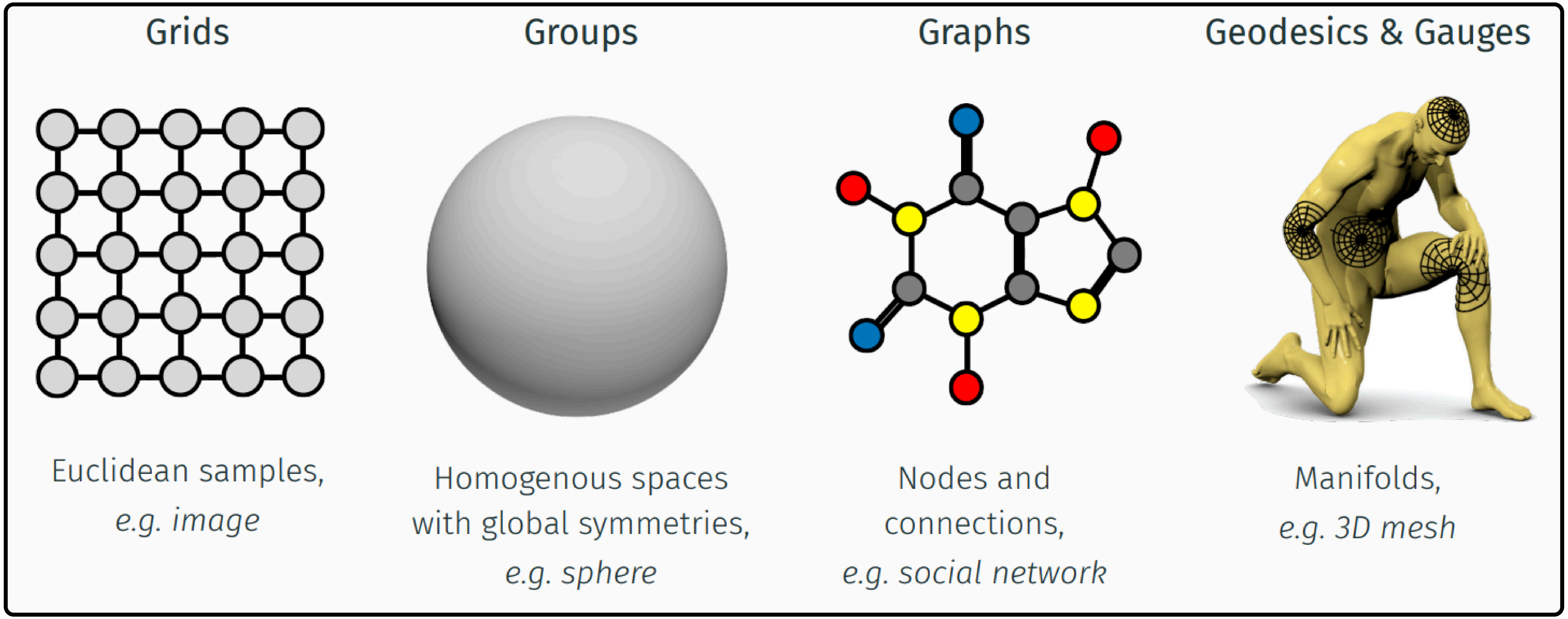


Others

Projection breaks symmetries and geometric properties



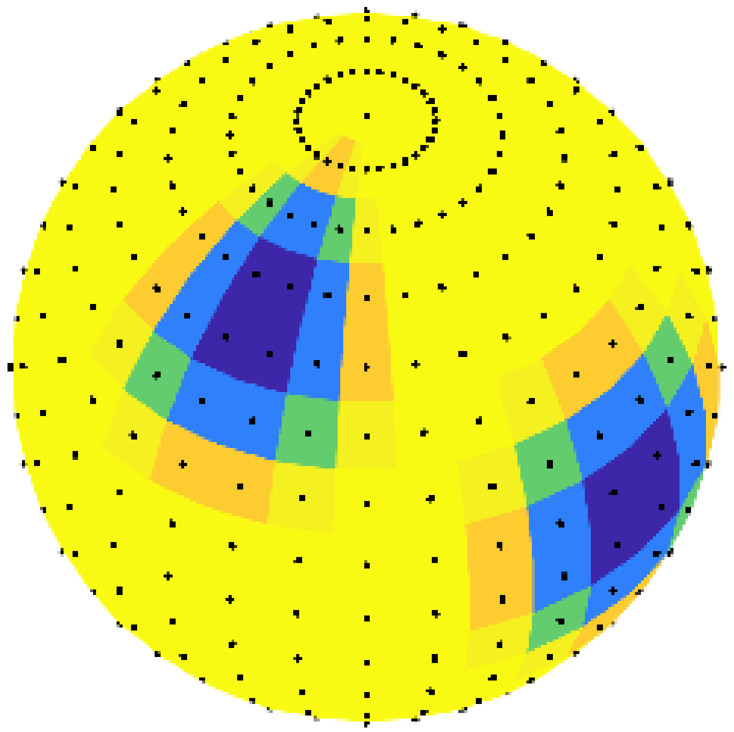
Geometric deep learing



(Bronstein et al. 2022)

Scalable and equivariant deep learning on the sphere

Discrete

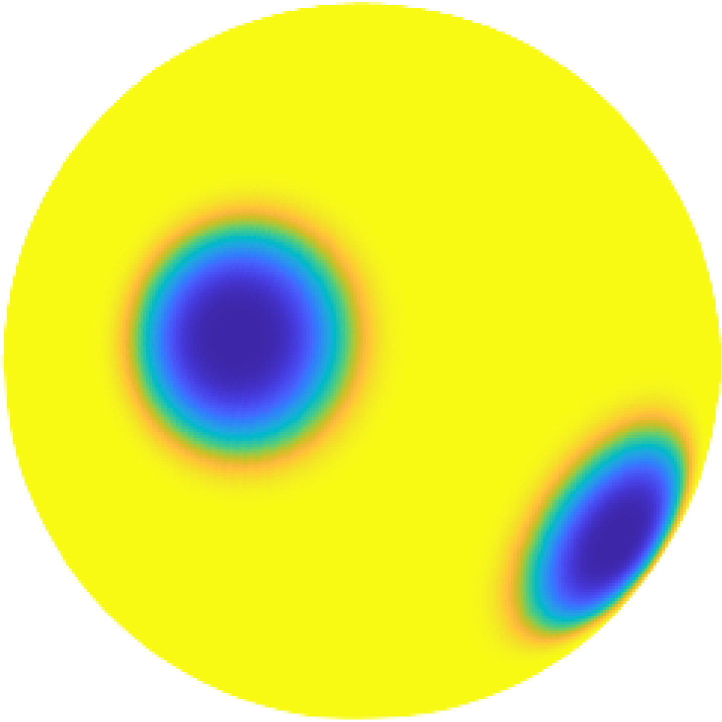


Not Equivariant

Scalable

(Jiang et al. 2019, Zhange et al. 2019, Perraudin et al. 2019, Cohen et al. 2019, ...)

Continuous

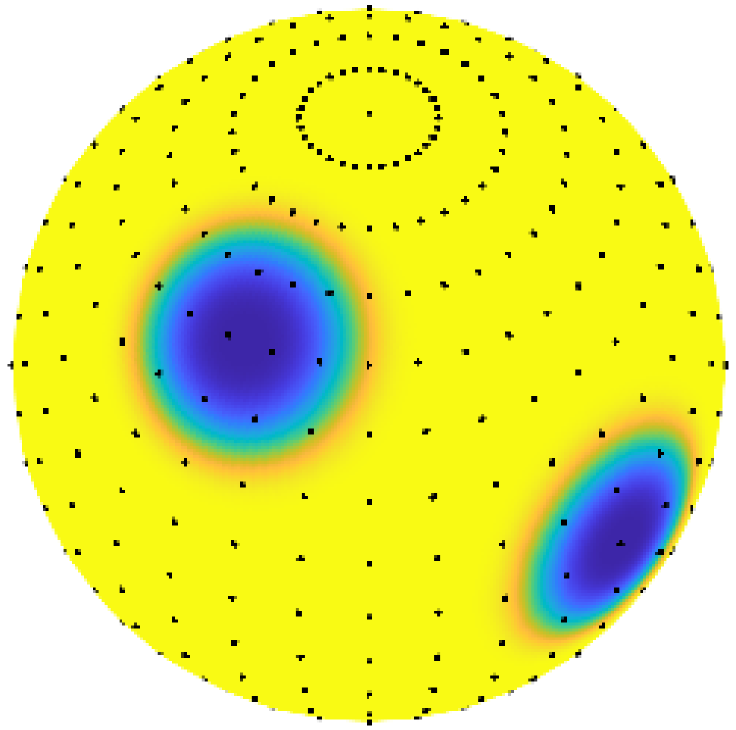


Equivariant

Not Scalable

(Cohen et al. 2018, Esteves et al. 2018, Kondor et al. 2018, Cobb et al. McEwen 2021, McEwen et al. 2022, ...)

Discrete-Continuous (DISCO)



Equivariant

Scalable

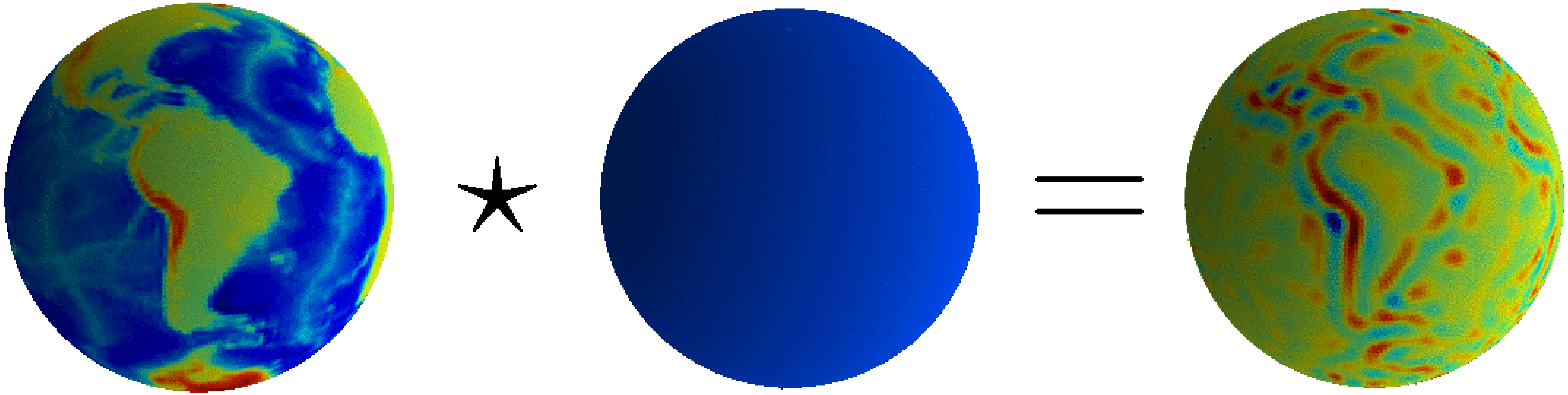
(Ocampo, Price & McEwen 2021)

Spherical convolution

Convolutions are the only linear equivariant layers (Kondor & Trivedi 2018).

Spherical convolution:

$$\underbrace{(f \star \psi)}_{\text{Convolution}}(\mathrm{R}) = \underbrace{\int_{\mathbb{S}^2} f(\theta, \phi)}_{\text{Projection}} \, \underbrace{(\mathrm{R}\,\psi)}_{\text{Rotation of kernel}}^{*}(\theta, \phi) \, \mathrm{d}\mu(\theta, \phi)$$



Scalable and equivariant spherical CNNs by DISCO convolutions

Scalable and Equivariant Spherical CNNs by DIScrete-COntinuous (DISCO) Convolutions
(Ocampo, Price & McEwen, ICLR, 2023)

Follows by a **careful hybrid representation** of the spherical convolution:

- some components left continuous to facilitate accurate rotational equivariance;
- while other components are discretized to yield scalable computation.

DISCO spherical convolution

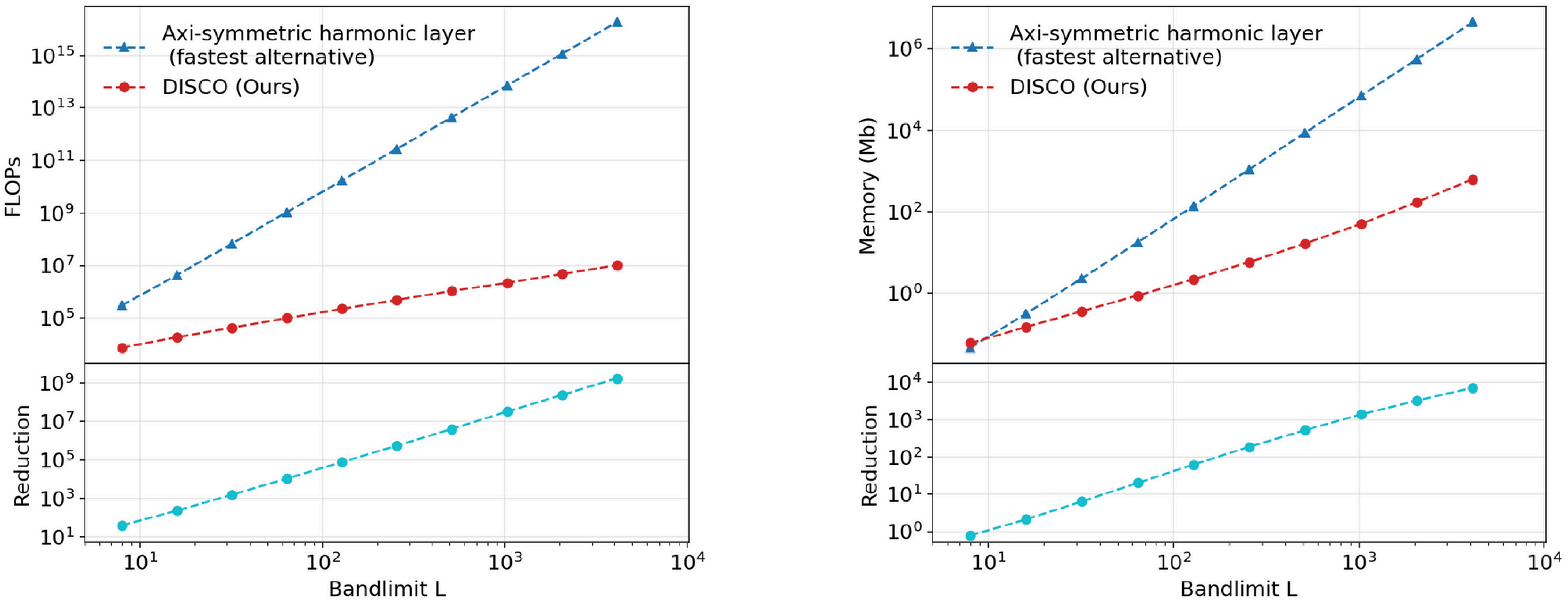
Spherical convolution can be carefully approximated by DISCO representation:

$$(f \star \psi)(\mathsf{R}) = \int_{\mathbb{S}^2} f(\theta, \phi)\, (\mathsf{R}\,\psi)^*(\theta, \phi)\, \mathrm{d}\mu(\theta, \phi) \approx \underbrace{\sum_i q(\theta_i, \phi_i)}_{\text{Exact quadrature}} \underbrace{f[\theta_i, \phi_i]}_{\text{Discrete}} \underbrace{\psi(\mathsf{R}^{-1}(\theta_i, \phi_i))}_{\text{Continuous}}$$

for spherical signal and filter kernel $f, \psi : \mathbb{S}^2 \rightarrow \mathbb{R}$, spherical coordinates $(\theta, \phi) \in \mathbb{S}^2$, and 3D rotations $\mathsf{R} \in \mathrm{SO}(3)$. Leverage sampling theory of McEwen & Wiaux (2011).

Scalable and equivariant spherical CNNs by DISCO convolutions

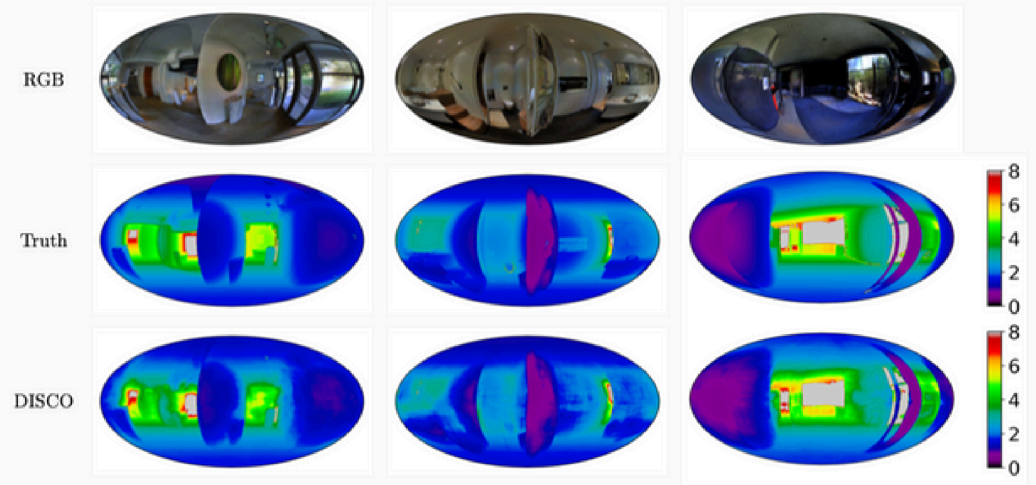
Dramatic **computational savings** in FLOPs and memory.



For 4k spherical image, **10^9 saving in computational cost** and **10^4 saving in memory usage**.

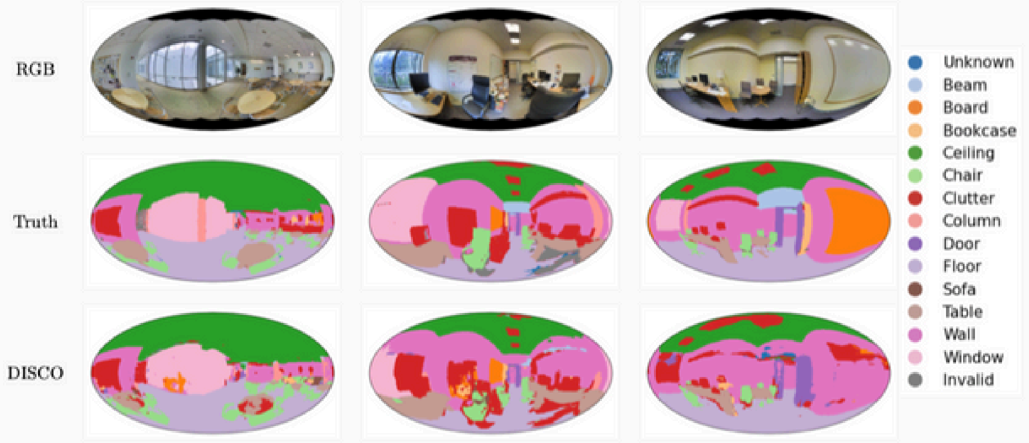
DISCO achieves SOTA performance

Depth estimation for Pano3D



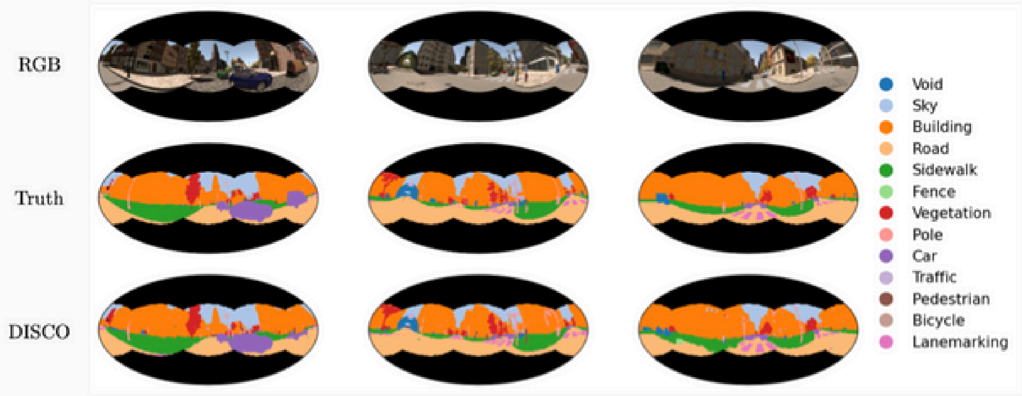
Example predictions for depth estimation of Pano3D data (depth plotted in meters).

Semantic segmentation for 2D3Ds dataset



Example predictions for semantic segmentation of 2D3DS data.

Semantic segmentation for Omni-SYNTHIA dataset



Example predictions for semantic segmentation of Omni-SYNTHIA data.

Depth estimation for Pano3D

Model	Parameters	Depth Error Metrics				Depth Accuracy Metrics			
		wRMSE	wRMSLE	wAbsRel	wSqRel	$\delta^{\text{ico}}_{1.05}$	$\delta^{\text{ico}}_{1.1}$	$\delta^{\text{ico}}_{1.25}$	$\delta^{\text{ico}}_{1.25^2}$
Planar UNet	27M	**0.4520**	**0.1300**	0.1147	**0.0811**	36.68%	60.59%	88.31%	96.96%
DISCO-Directional **(Ours)**	**658k**	0.5063	0.1695	**0.1109**	0.0852	**38.32%**	**62.12%**	**88.65%**	**97.29%**

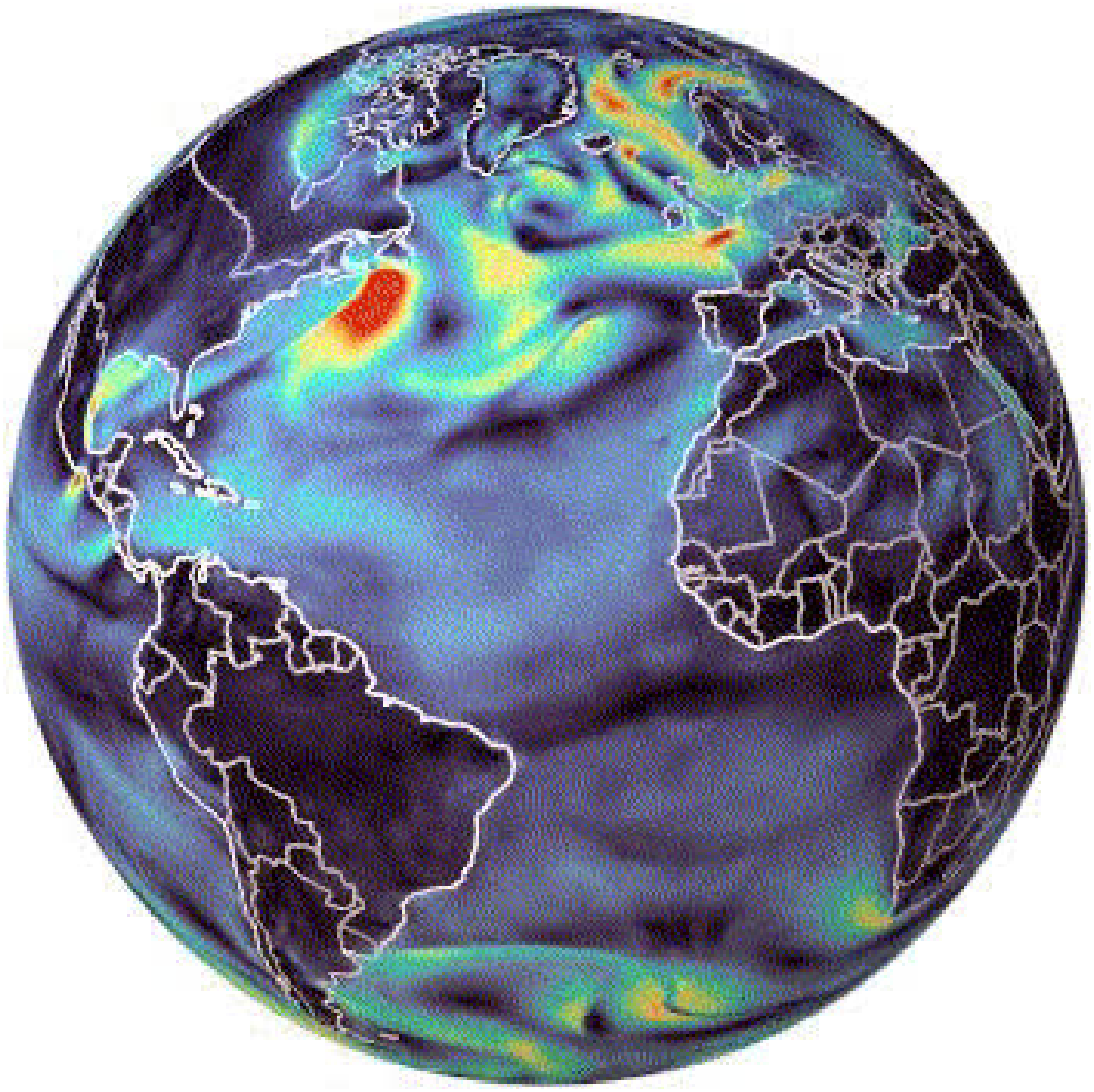
Semantic segmentation for 2D3Ds dataset

Model	mIoU	mAcc
Planar UNet	35.9	50.8
UGSCNN	38.3	54.7
GaugeNet	39.4	55.9
HexRUNet	43.3	58.6
SWSCNNs	43.4	58.7
CubeNet	45.0	62.5
MöbiusConv	43.3	60.9
DISCO-Axisymmetric (Ours)	39.7	54.1
DISCO-Directional-Separable (Ours)	43.9	60.9
DISCO-Directional (Ours)	45.2	61.5
DISCO-Directional-Aug (Ours)	**45.7**	**62.7**

Semantic segmentation for Omni-SYNTHIA dataset

Model	mIoU	mAcc
Planar UNet	44.6	52.6
UGSCNN	37.6	48.9
HexUNet	48.3	57.1
DISCO-Directional-Separable (Ours)	48.3	59.3
DISCO-Directional-Separable-Aug (Ours)	**49.2**	**63.7**

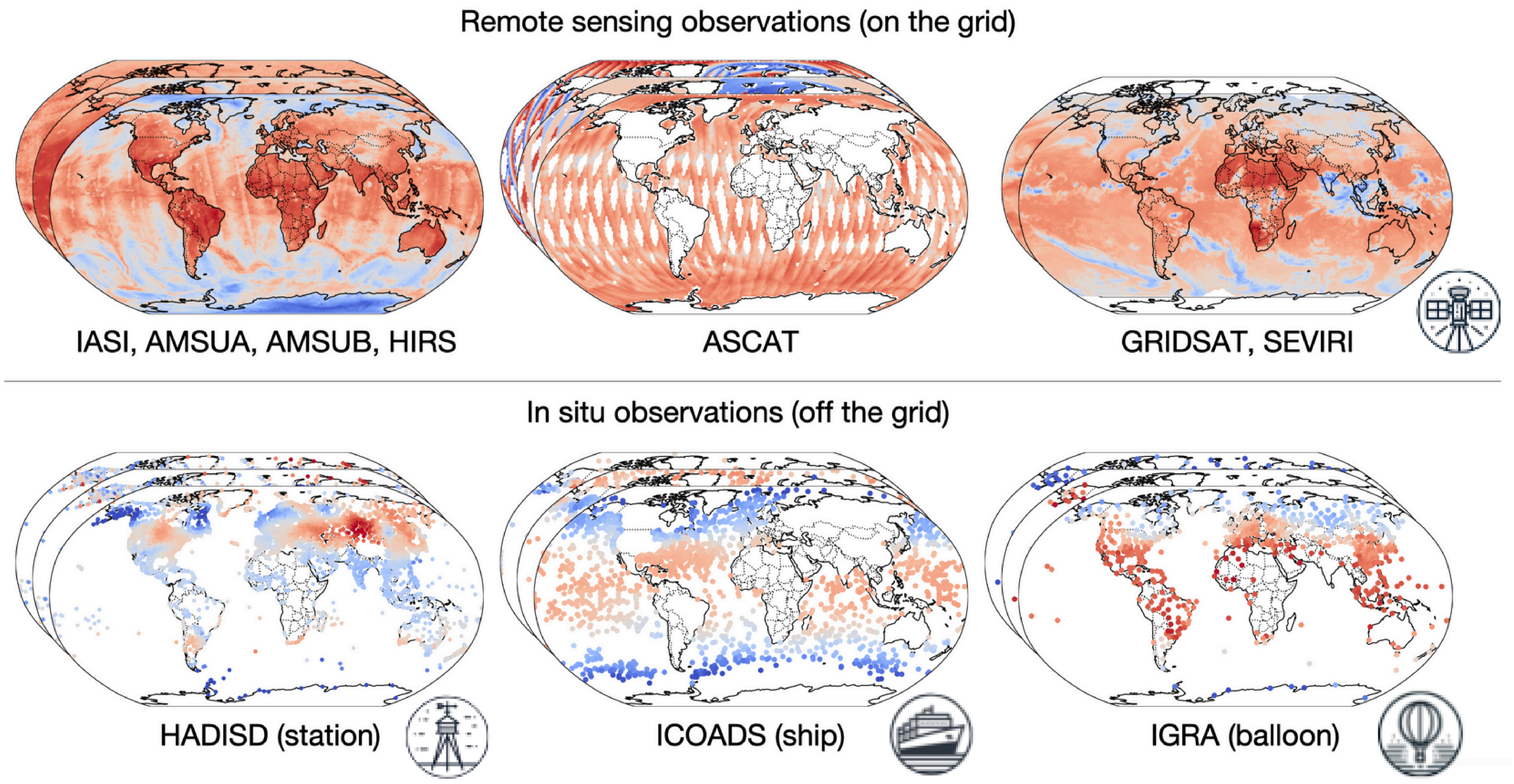
Global weather forecasting



FourCastNet 3 from NVIDIA

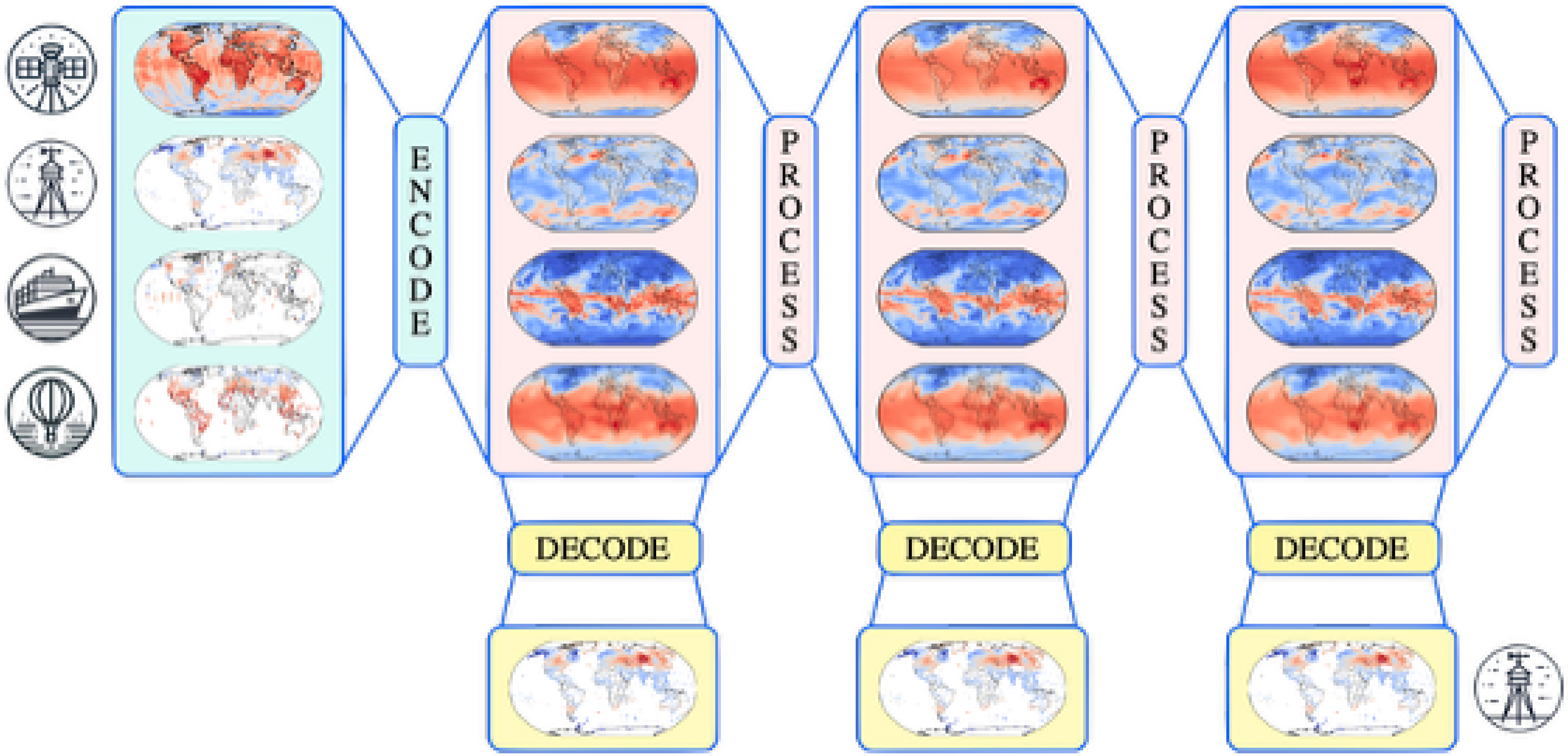
(Bonev et al. 2025)

Need for end-to-end and multi-modal learning



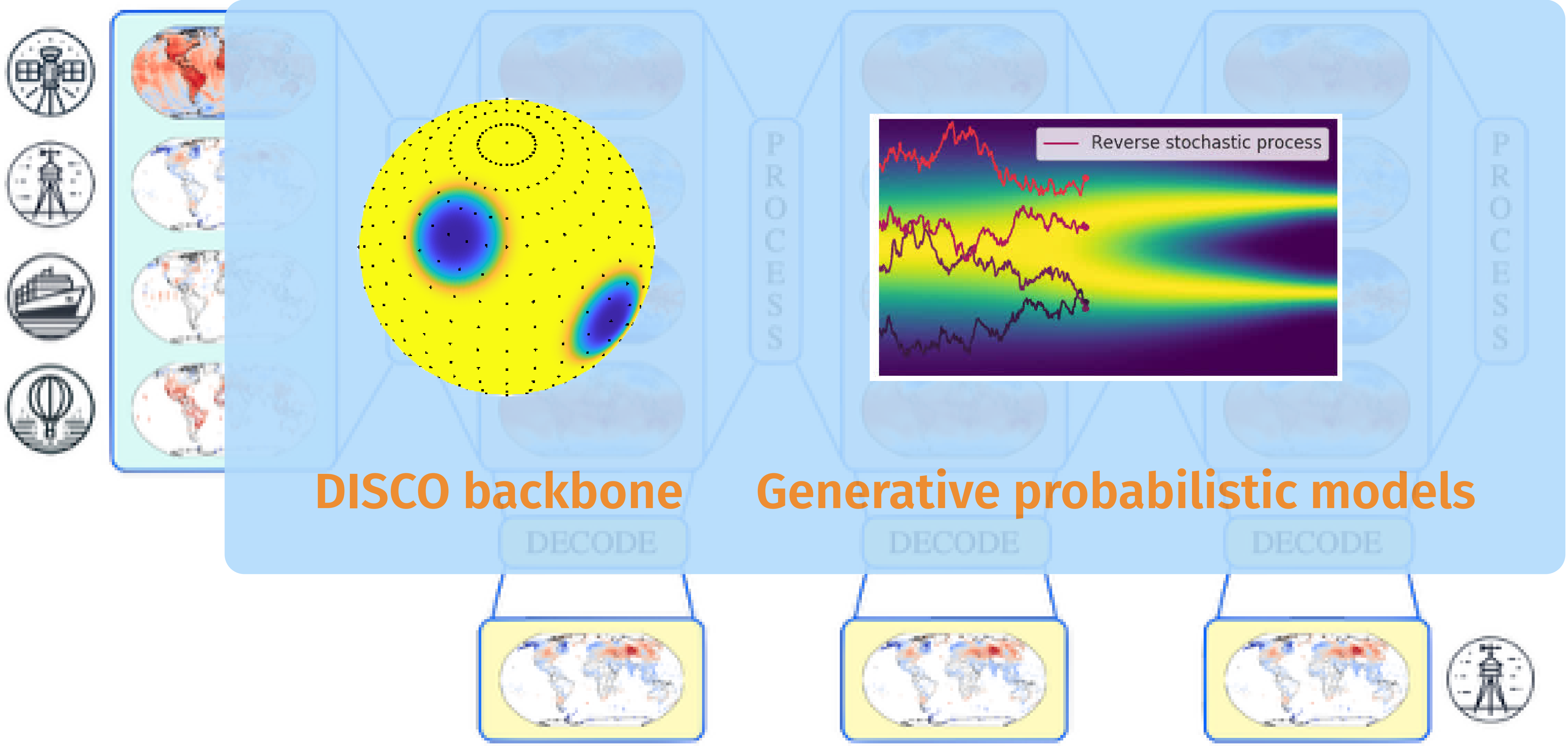
(Allen et al. 2025)

Aardvark: end-to-end, multi-modal



(Allen et al. 2025)

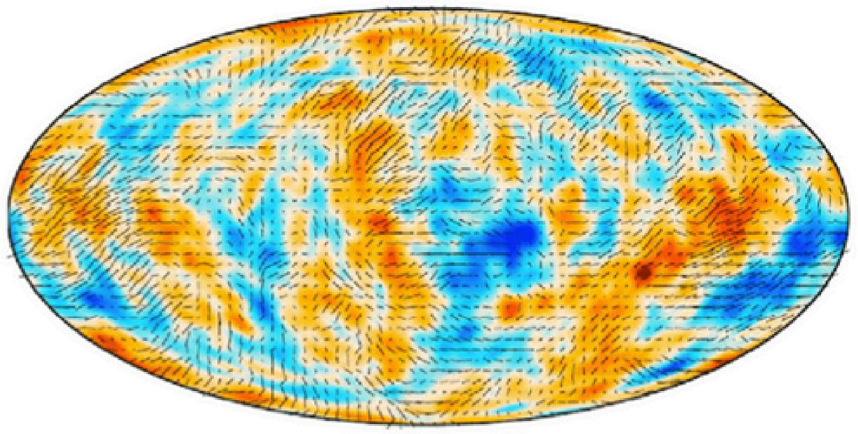
Future: end-to-end, multi-modal, geometric, probabilistic



(Allen et al. 2025)

Geometric deep learning for vector and spin fields

Wind $\rightsquigarrow$ vector (spin-1) field

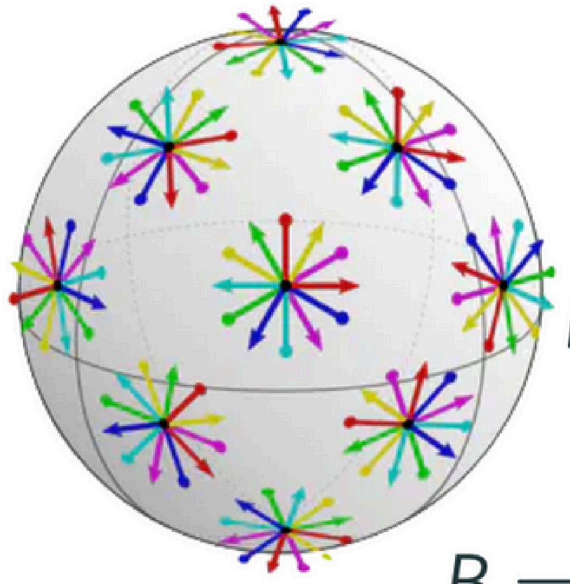


CMB polarization $\rightsquigarrow$ spin-2 field

Equivariant learning for spherical fields of different spin

Fibre bundle representation:

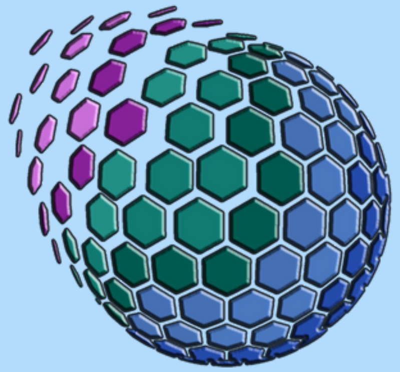
- Base space $B = \mathbb{S}^2 \simeq \mathrm{SO}(3)/\mathrm{SO}(2)$
- Fibre $H = \mathrm{SO}(2)$
- Fibre bundle $G = \mathrm{SO}(3)$



Spin equivariant approach:

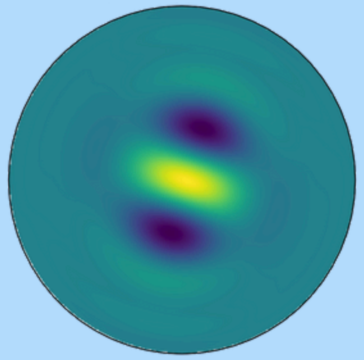
1. Equivariant lifting:
   $f \uparrow^{\mathrm{SO}(3)} (\rho) = \varrho(\mathrm{h}^{-1}(\rho))\, f(P(\rho))$, for projection $P : \mathrm{SO}(3) \to \mathbb{S}^2$, twist $\mathrm{h} : \mathrm{SO}(3) \to \mathrm{SO}(2)$ and representation $\varrho : \mathrm{SO}(2) \to \mathbb{R}$.
2. Group convolution on SO(3).
3. Equivariant projection:
   $f \downarrow_{\mathbb{S}^2} (\omega) = f \uparrow^{\mathrm{SO}(3)} (S(\omega))$, for section $S : \mathbb{S}^2 \to \mathrm{SO}(3)$.

s2x suite of codes



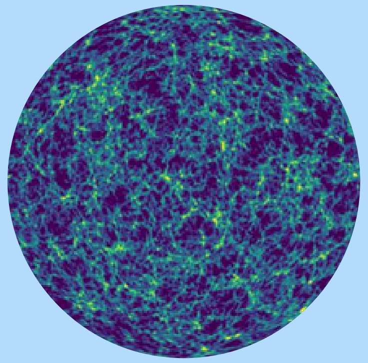
s2fft: Spherical harmonic transforms

https://github.com/astro-informatics/s2fft



s2wav: Spherical wavelet transforms

https://github.com/astro-informatics/s2wav



s2scat: Spherical wavelet scattering transforms

https://github.com/astro-informatics/s2scat

s2ai: Scalable and equivariant spherical AI

https://github.com/astro-informatics/s2ai

Emulation

The Alan Turing Institute

Auto Emulate

AutoEmulate is a Python library for automatically creating accurate and efficient emulators of complex simulations.

Run a complete machine learning pipeline to compare and optimise a wide range of models, with functions for downstream tasks like prediction, sensitivity analysis and calibration.

https://www.autoemulate.com

Open source & free to use

Low code
Data-processing, model comparison, cross-validation, hyperparameter search and more in few lines of code.

Domain agnostic
Can be applied to simulation models from any domain.

Easy Integration
All emulators are compatible with commonly used Python ML frameworks, making them easy to integrate into downstream applications.

State of the art emulators

Classical
Radial Basis Functions
Second Order Polynomials

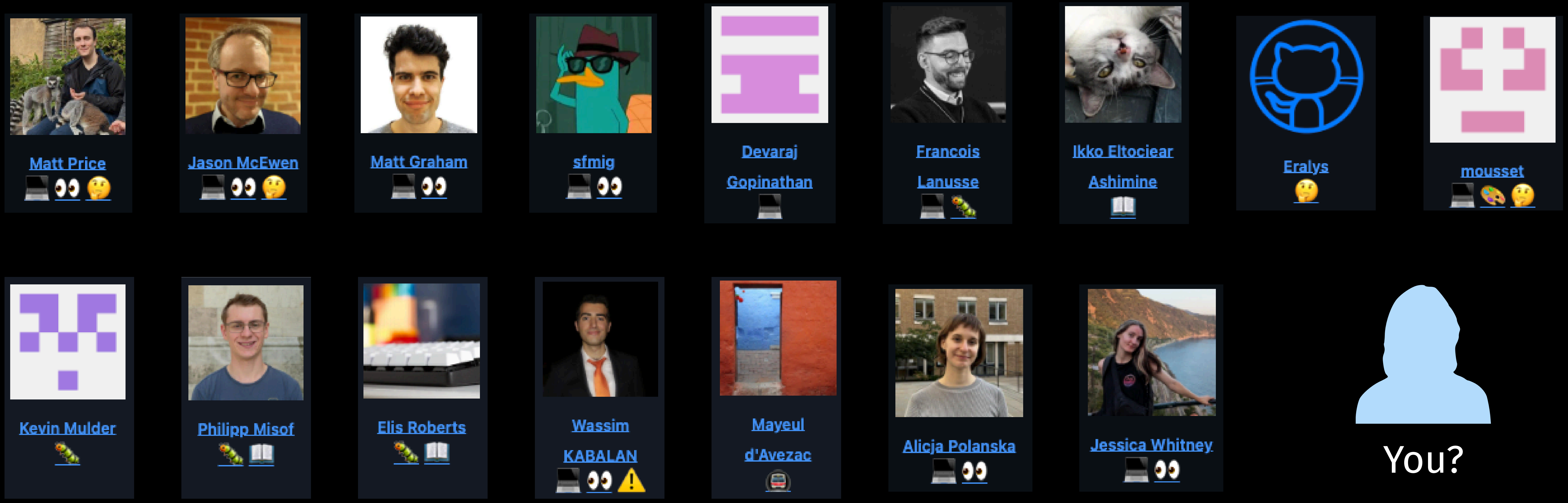
Machine Learning
Random Forests
Gradient Boosting
Support Vector Machines

Deep Learning
Gaussian Processes
Conditional Neural Processes

Questions?

Extra slides

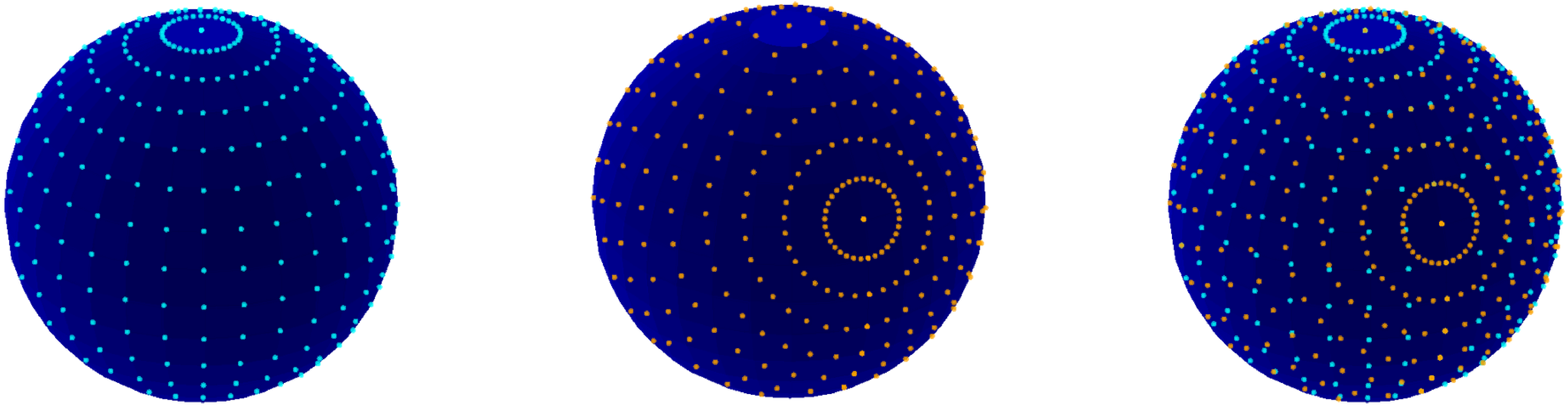
All contributors



Discretisation of the sphere and rotational equivariant

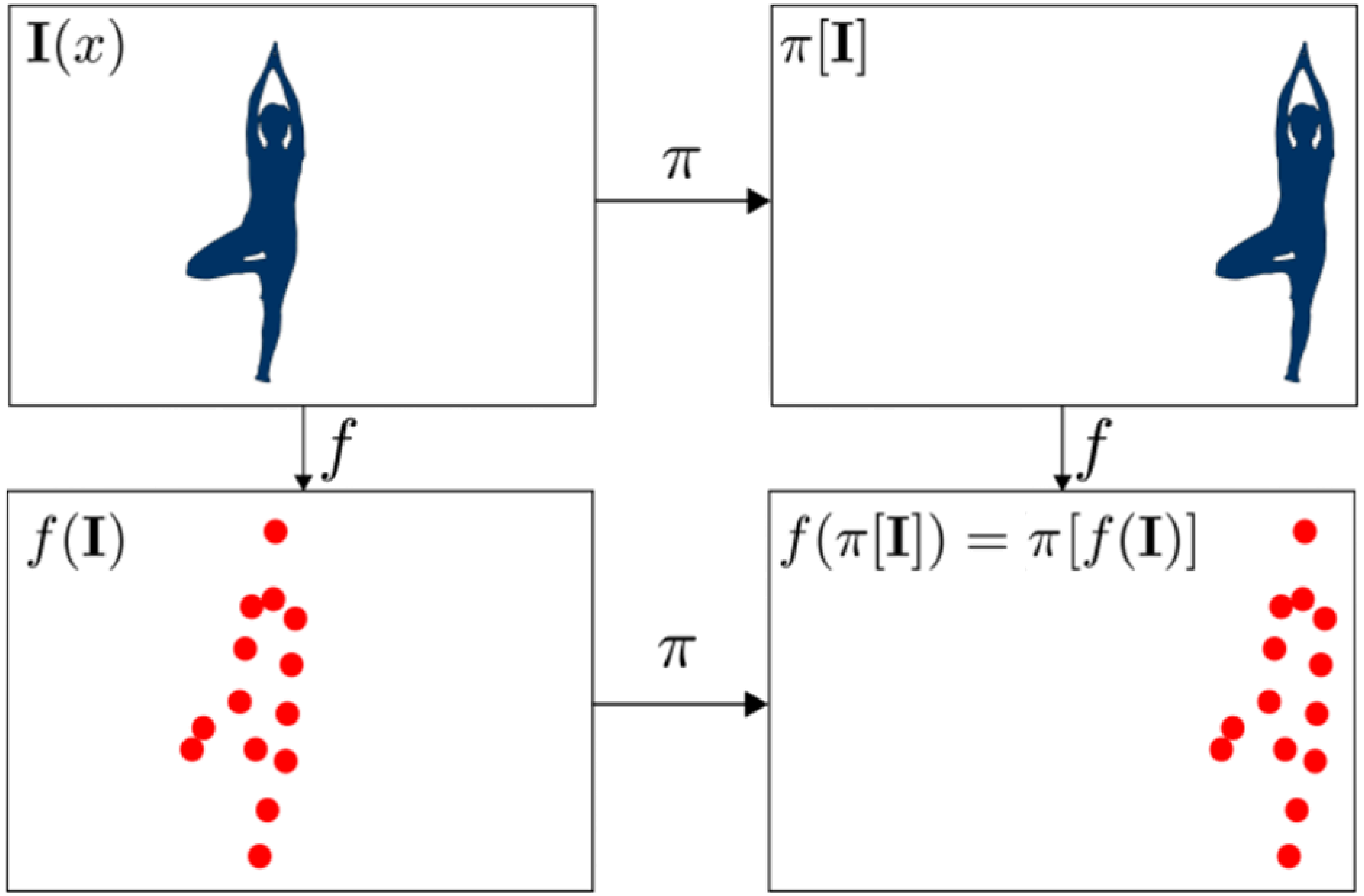
Well-know that regular discretisation of the sphere does not exist (e.g. Tegmark 1996).

$\Rightarrow$ Not possible to discretise sphere in a manner invariant to rotations.



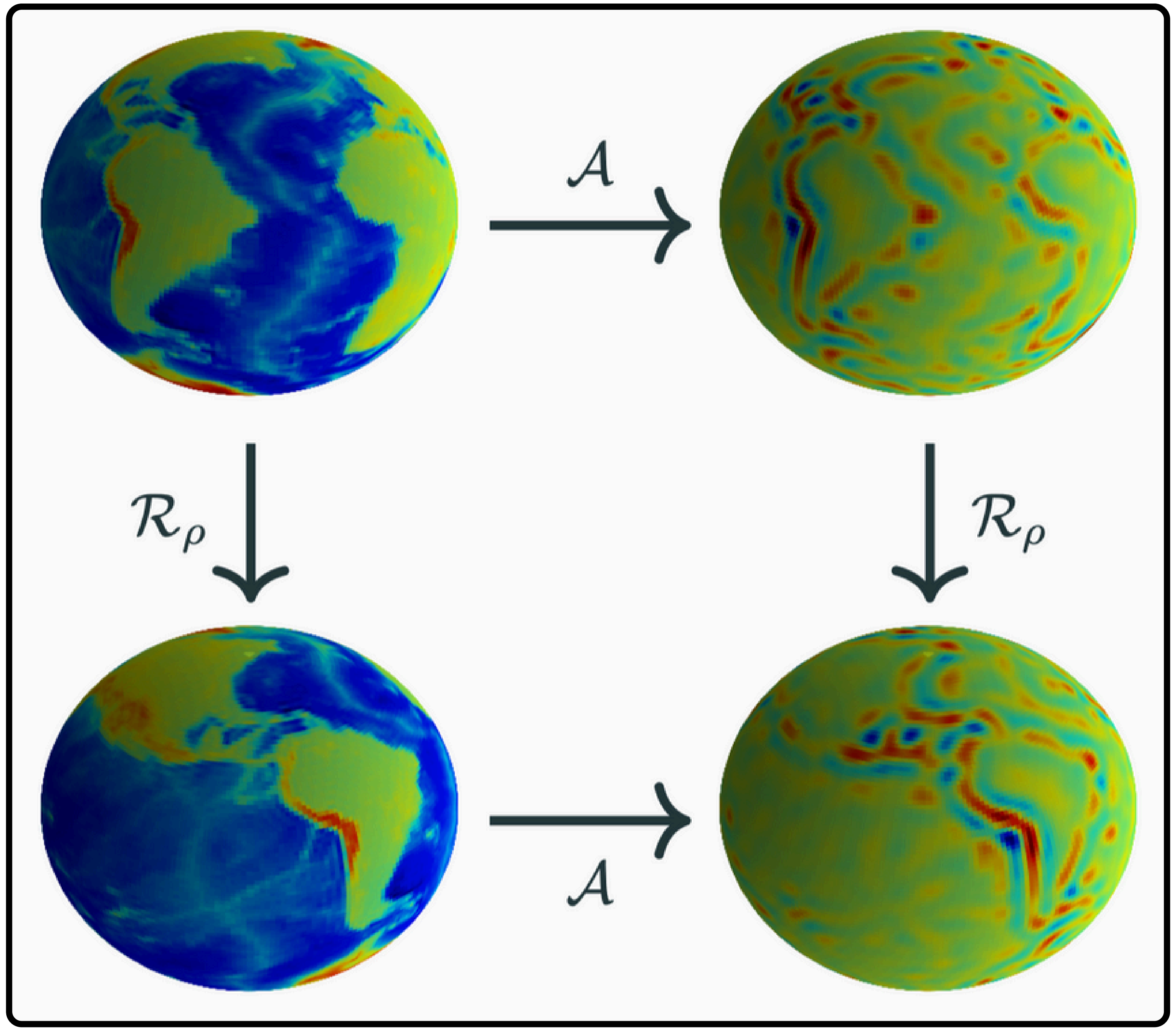
Capturing strict equivariance with operations defined directly in discretised (pixel) space **not possible** due to structure of the sphere.

Translational equivariance



Rotational equivariance

$$(\mathsf{R}\, f) \star \psi = \mathsf{R}\, (f \star \psi)$$



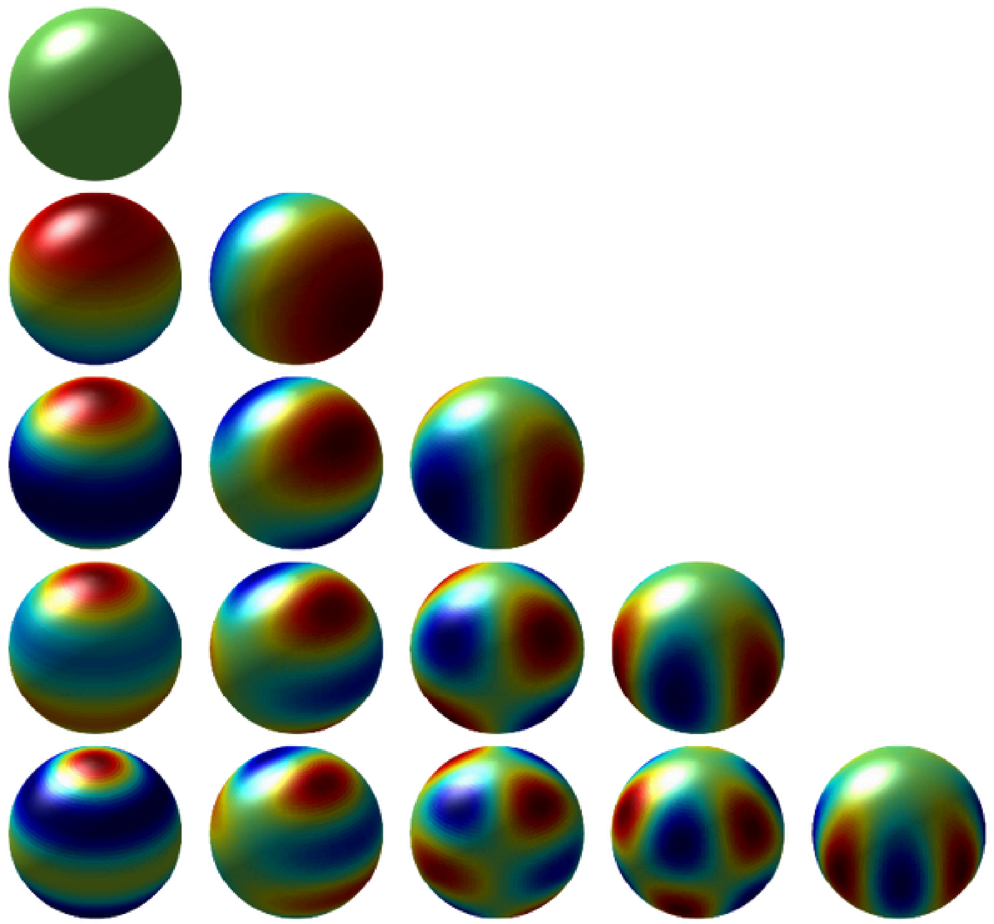
Spherical harmonic transform (Fourier transform on the sphere)

A field $f \in \mathrm{L}^2(\mathbb{S}^2)$ can be decomposed into its harmonic representation by

$$f(\theta, \phi) = \sum_{\ell,m} f_{\ell m} Y_{\ell m}(\theta, \phi),$$

where the spherical harmonic coefficients are given by the usual projection onto the basis functions:

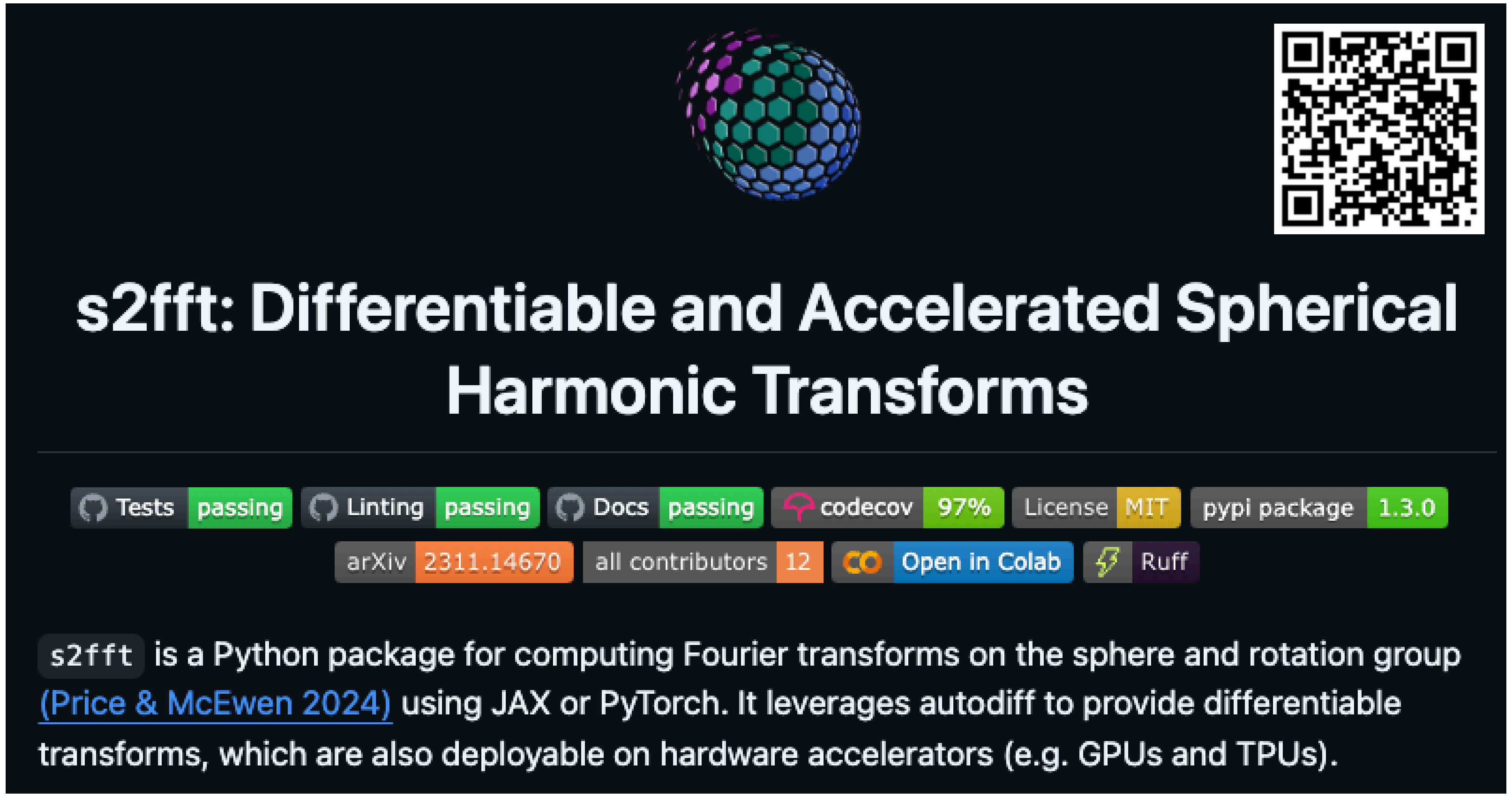
$$f_{\ell m} = \int_{\mathbb{S}^2} f(\theta, \phi) Y^*_{\ell m}(\theta, \phi) \sin\theta \mathrm{d}\theta \mathrm{d}\phi.$$



Spherical harmonics

Driscoll & Healy (1995), ..., McEwen & Wiaux (2011), Price & McEwen (2024)

`s2fft`: spherical harmonic and Wigner transforms



s2fft: Differentiable and Accelerated Spherical Harmonic Transforms

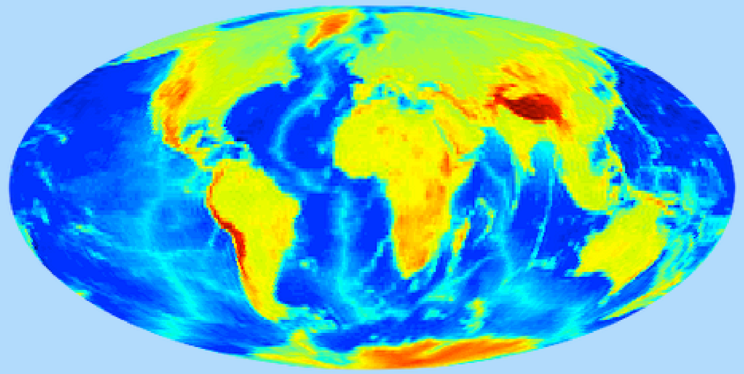
Tests passing | Linting passing | Docs passing | codecov 97% | License MIT | pypi package 1.3.0
arXiv 2311.14670 | all contributors 12 | Open in Colab | Ruff

`s2fft` is a Python package for computing Fourier transforms on the sphere and rotation group (Price & McEwen 2024) using JAX or PyTorch. It leverages autodiff to provide differentiable transforms, which are also deployable on hardware accelerators (e.g. GPUs and TPUs).

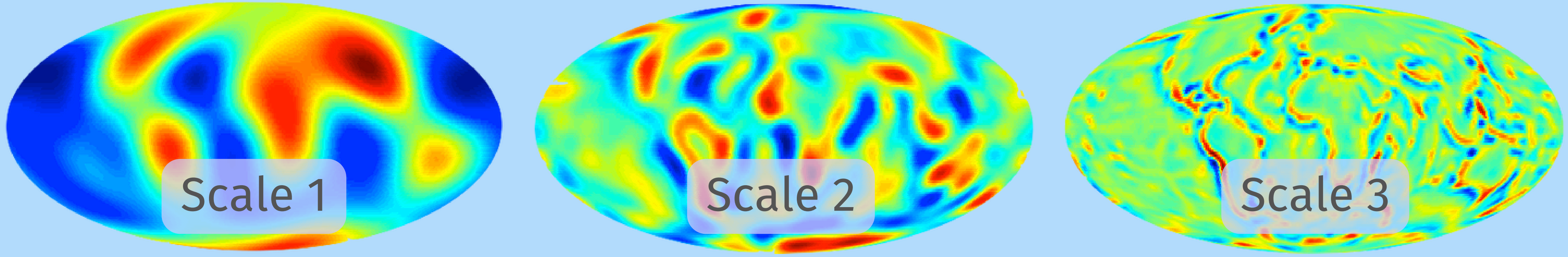
https://github.com/astro-informatics/s2fft

s2wav : wavelet transforms on the sphere

Wavelets capture signal content localised in both scale and space.



Original spherical data



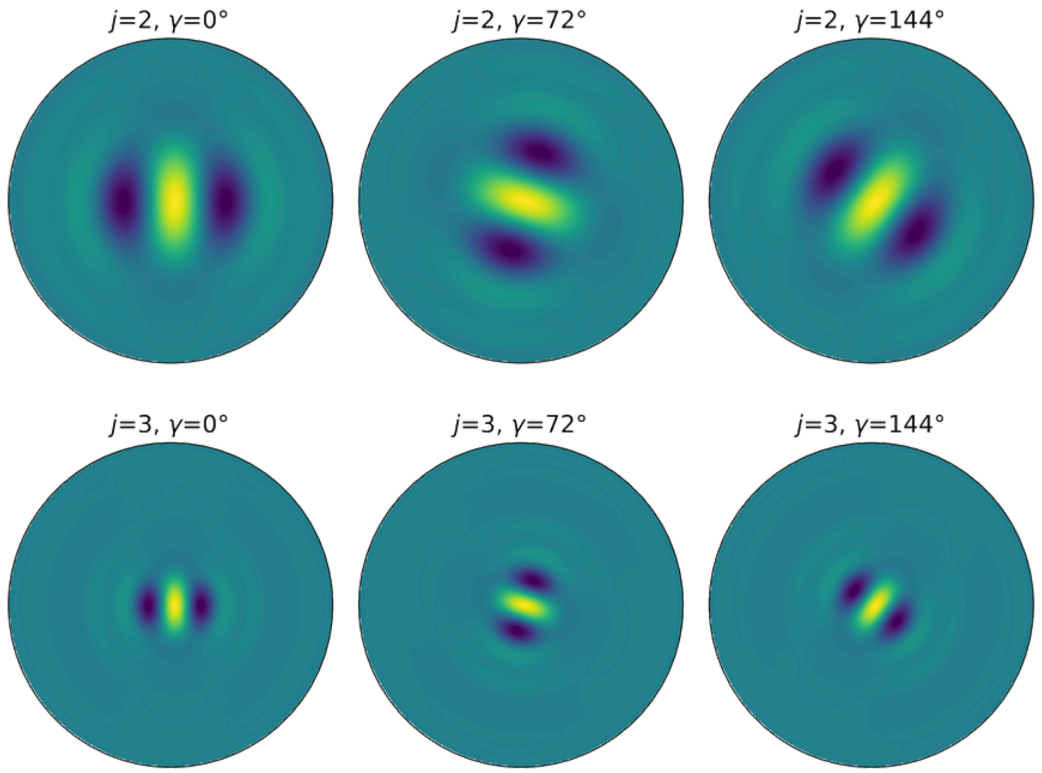
Wavelet coefficients

Spherical wavelet transform

Spherical wavelet transform, with wavelet ψ_j and scaling function φ, given by

$$W_j(\rho) = \underbrace{(f \star \psi_j)}_{\text{Spherical convolution}}(\rho) = \int_{\mathbb{S}^2} f(\theta, \phi)\, \underbrace{(R_\rho \psi_j)}_{\text{Rotated wavelet}}^*(\theta, \phi)\, \mathrm{d}\mu(\theta, \phi)$$

Wavelets carefully constructed to satisfy admissibility condition so that field can be reconstructed exactly from its wavelet coefficients.



Spherical wavelets (orthographic)

McEwen et al. (2007), Wiaux, McEwen et al. (2008), McEwen et al. (2013), McEwen et al. (2015), McEwen et al. (2018)

s2wav: wavelet transforms on the sphere

s2wav: Differentiable and Accelerated Wavelet Transforms on the Sphere

Tests passing | codecov 92% | License MIT | pypi package 1.0.4 | arXiv 2402.01282 | all contributors 4

Open in Colab

`s2wav` is a python package for computing wavelet transforms on the sphere and rotation group, both in JAX and PyTorch. It leverages autodiff to provide differentiable transforms, which are also deployable on modern hardware accelerators (e.g. GPUs and TPUs), and can be mapped across multiple accelerators.

https://github.com/astro-informatics/s2wav

`s2scat`: wavelet scattering transforms on the sphere

Spherical scattering network

Scattering coefficients for path p given by cascade of wavelet transforms with modulus activation function:

$$S[p]f = |||f \star \psi_{j_1}| \star \psi_{j_2}| \dots \star \psi_{j_d}| \star \varphi.$$

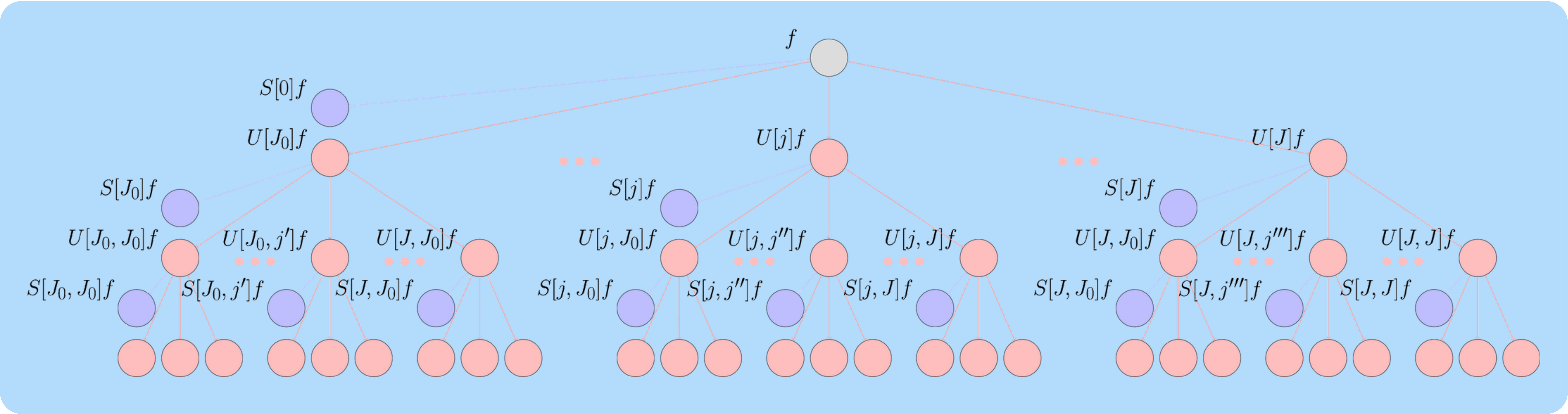
Spherical scatting networks is a collection of scattering transforms for a number of paths.

Mallat (2011), McEwen et al. (2022), Mousset et al. McEwen (2024)

Properties:

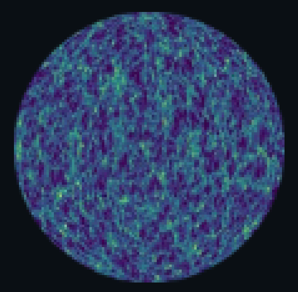
1. Rotational equivariance
2. Isometric invariance
3. Stability to diffeomorphisms

↪ Well-behaved representation space.



Spherical scattering network

s2scat: wavelet scattering transforms on the sphere



s2scat: Differentiable Scattering Covariances on the Sphere

Tests passing | codecov 91% | License MIT | pypi v0.0.3 | downloads 79/month | all contributors 4

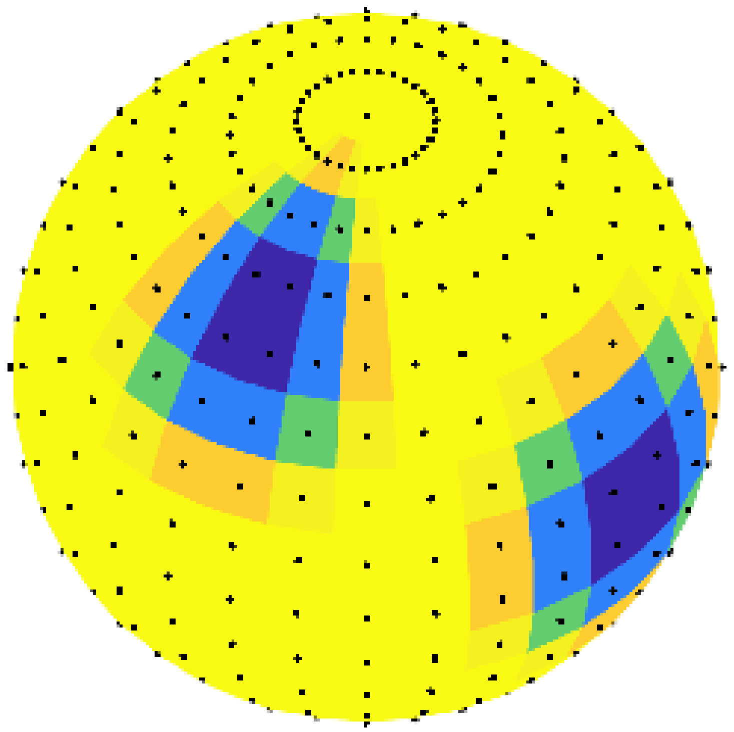
Open in Colab

`s2scat` is a Python package for computing scattering covariances on the sphere (Mousset et al. 2024) using JAX. It exploits autodiff to provide differentiable transforms, which are also deployable on hardware accelerators (e.g. GPUs and TPUs), leveraging the differentiable and accelerated spherical harmonic and wavelet transforms implemented in `s2fft` and `s2wav`, respectively. Scattering covariances are useful both for field-level generative modelling of complex non-Gaussian textures and for statistical compression of high dimensional field-level data, a key step of e.g. simulation based inference.

https://github.com/astro-informatics/s2scat

Equivariant and scalable deep learning on the sphere

Discrete

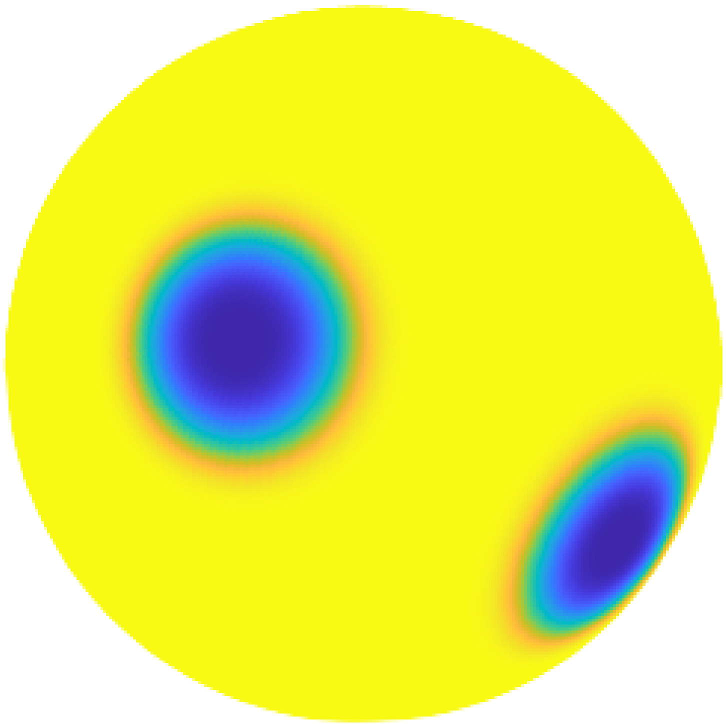


✗ Not Equivariant

✓ Scalable

(Jiang et al. 2019, Zhange et al. 2019, Perraudin et al. 2019, Cohen et al. 2019, ...)

Continuous

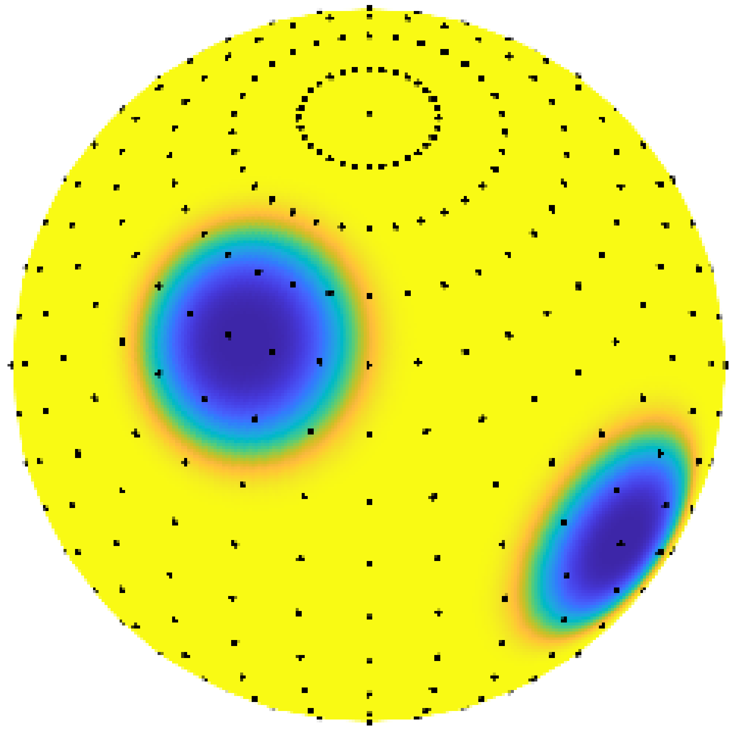


✓ Equivariant

✗ Not Scalable

(Cohen et al. 2018, Esteves et al. 2018, Kondor et al. 2018, Cobb et al. McEwen 2021, McEwen et al. 2022, ...)

Discrete-Continuous (DISCO)

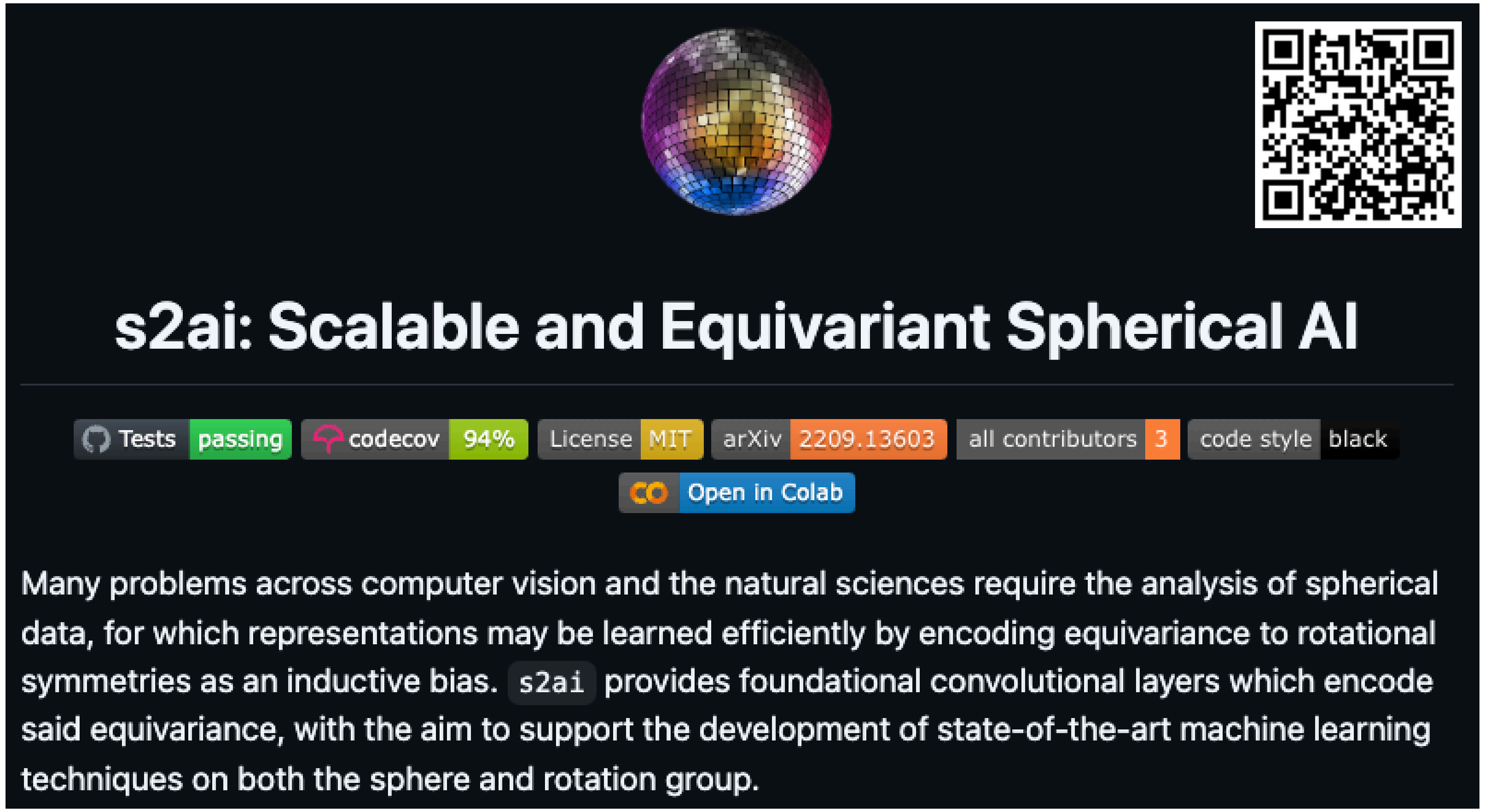


✓ Equivariant

✓ Scalable

(Ocampo, Price & McEwen 2021)

s2ai: spherical AI



s2ai: Scalable and Equivariant Spherical AI

Tests passing | codecov 94% | License MIT | arXiv 2209.13603 | all contributors 3 | code style black | Open in Colab

Many problems across computer vision and the natural sciences require the analysis of spherical data, for which representations may be learned efficiently by encoding equivariance to rotational symmetries as an inductive bias. `s2ai` provides foundational convolutional layers which encode said equivariance, with the aim to support the development of state-of-the-art machine learning techniques on both the sphere and rotation group.

https://github.com/astro-informatics/s2ai

Generative modelling of cosmological fields

Spherical scattering covariances for generative modelling

Scattering covariance statistics:

1. $S_1[\lambda]\, f = \mathbb{E}\big[\, |f \star \psi_\lambda|\, \big]$
2. $S_2[\lambda]\, f = \mathbb{E}\big[\, |f \star \psi_\lambda|^2\, \big]$
3. $S_3[\lambda_1, \lambda_2]\, f = \mathrm{Cov}\big[\, f \star \psi_{\lambda_2}, |f \star \psi_{\lambda_1}| \star \psi_{\lambda_2}\, \big]$
4. $S_4[\lambda_1, \lambda_2, \lambda_3]\, f = \mathrm{Cov}\big[\, |f \star \psi_{\lambda_1}| \star \psi_{\lambda_3}, |f \star \psi_{\lambda_2}| \star \psi_{\lambda_3}\, \big]$

Generative modelling by **matching set of scattering covariance statistics** with a (single) target simulation:

$$\min_f \left\| \mathcal{S}(f) - \mathcal{S}(f_{\text{target}}) \right\|^2$$

↪ Solve by gradient-based optimisation, leverging autodiff (requires s2fft, s2wav, s2scat)

Generative modelling of cosmic strings

Symmetry breaking **phase transitions** in the early Universe → topological defects.

Cosmic strings well-motivated phenomenon that arise when axial/cylindrical symmetry broken → **line-like discontinuities** in the fabric of the Universe.

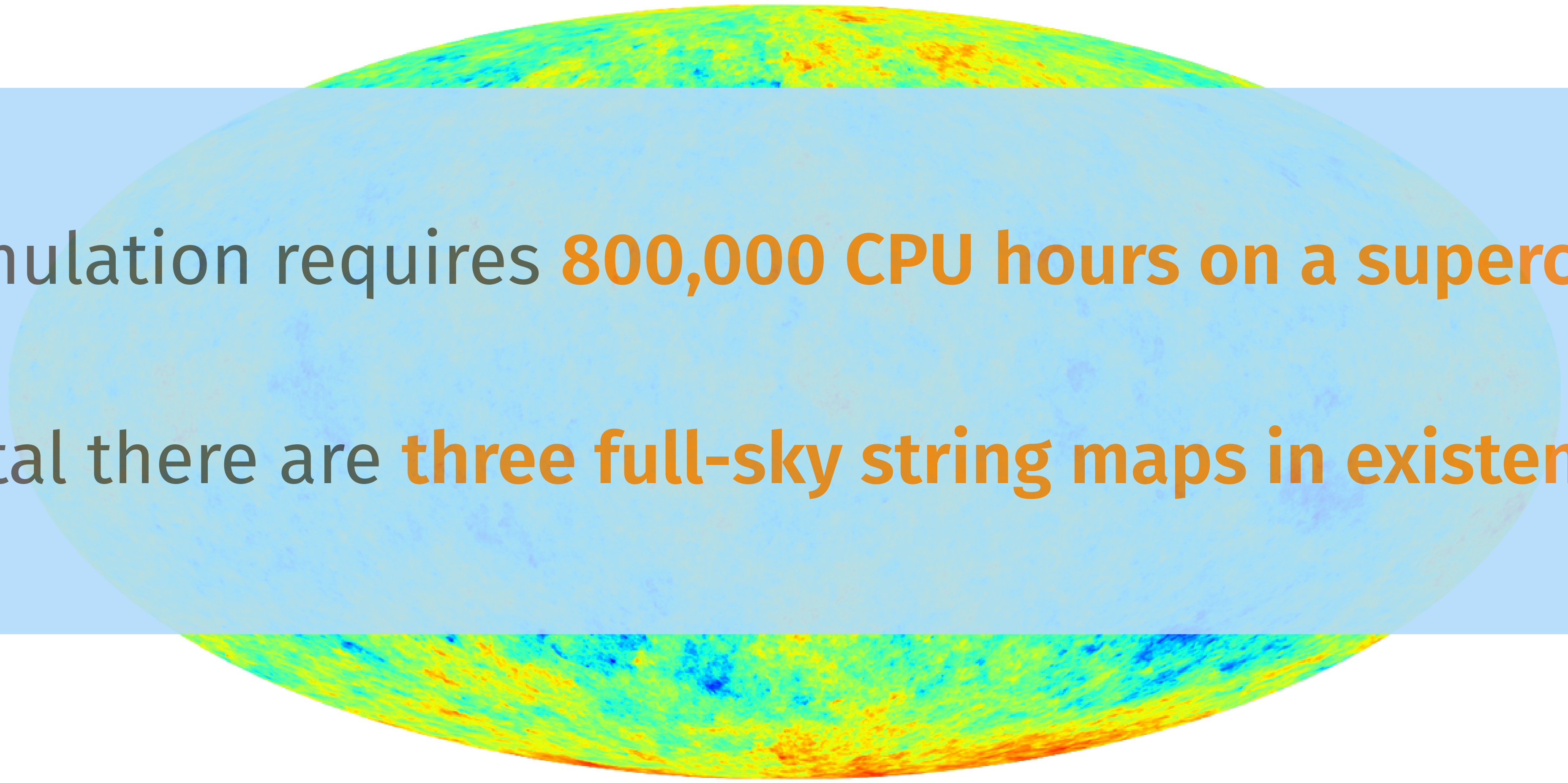
Observed transitions string-like topological defects in other media.

Detection of cosmic strings would open a **new window into the physics of the Universe.**

Optical microscope photograph of liquid crystal following temperature quench (Chuang et al. 1991)

Contact between theory and observation via high-resolution simulations (Ringeval et al. 2012).

Need to **simulate full physics**, evolving a network of string through cosmic time and then ray-tracing CMB photons through the string network.



A single simulation requires **800,000 CPU hours on a supercomputer**.

In total there are **three full-sky string maps in existence.**

Generative modelling of cosmic strings

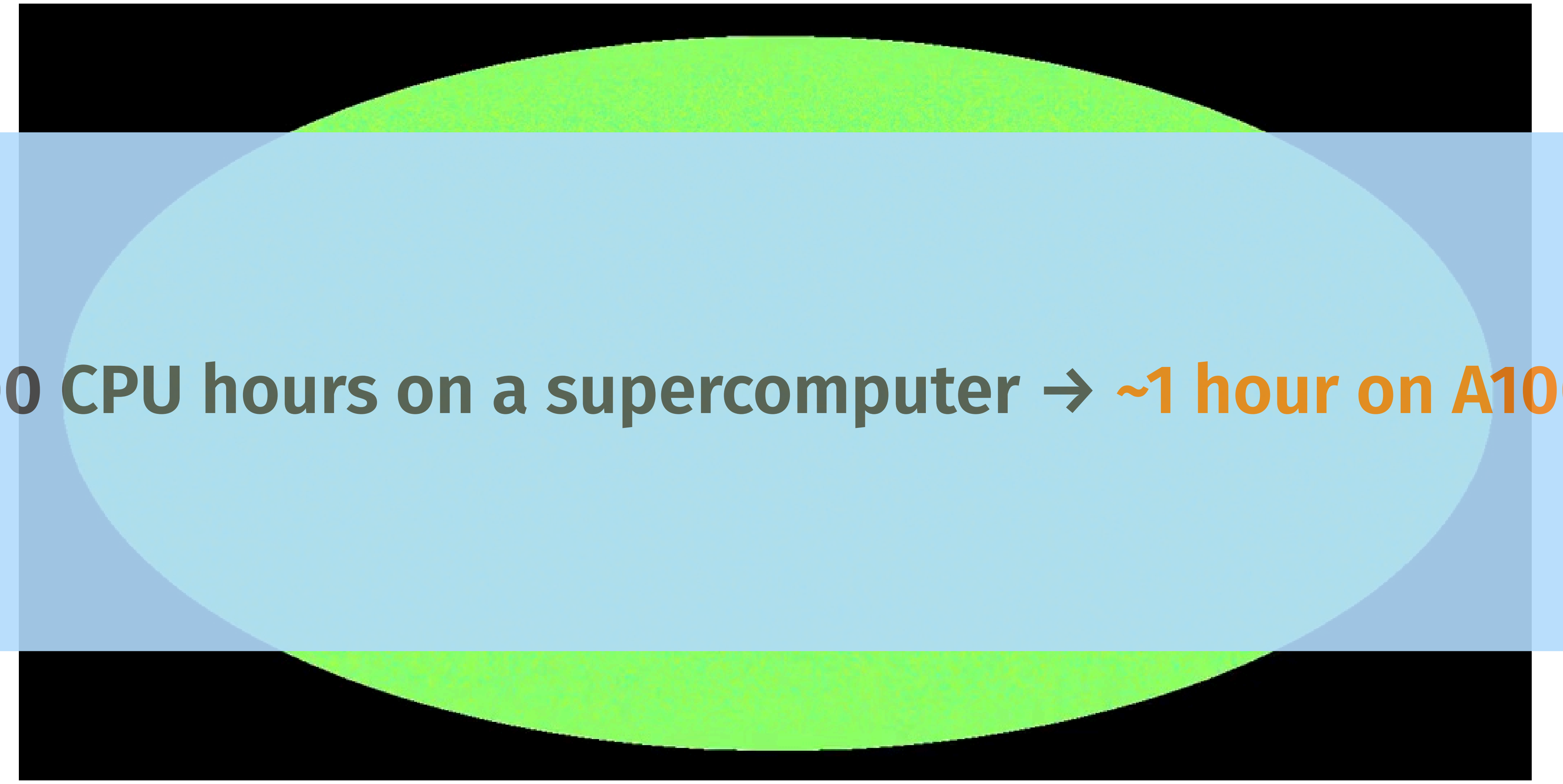
Instead of simulating full physics, **emulate with a scattering covariance generative model**.

Requires only **single target simulation**.

Generative modelling of cosmic strings

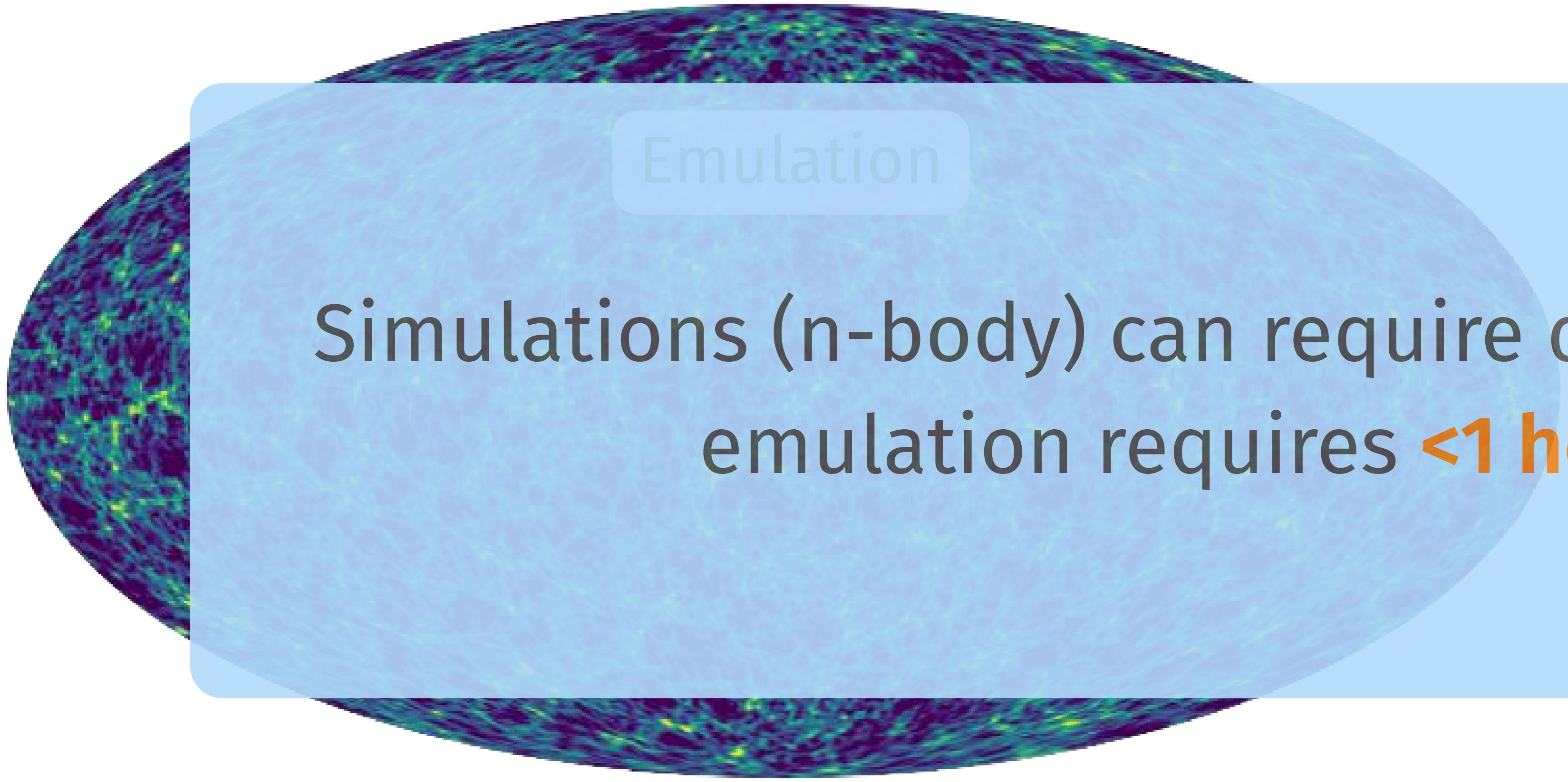
Instead of simulating full physics, **emulate with a scattering covariance generative model**.

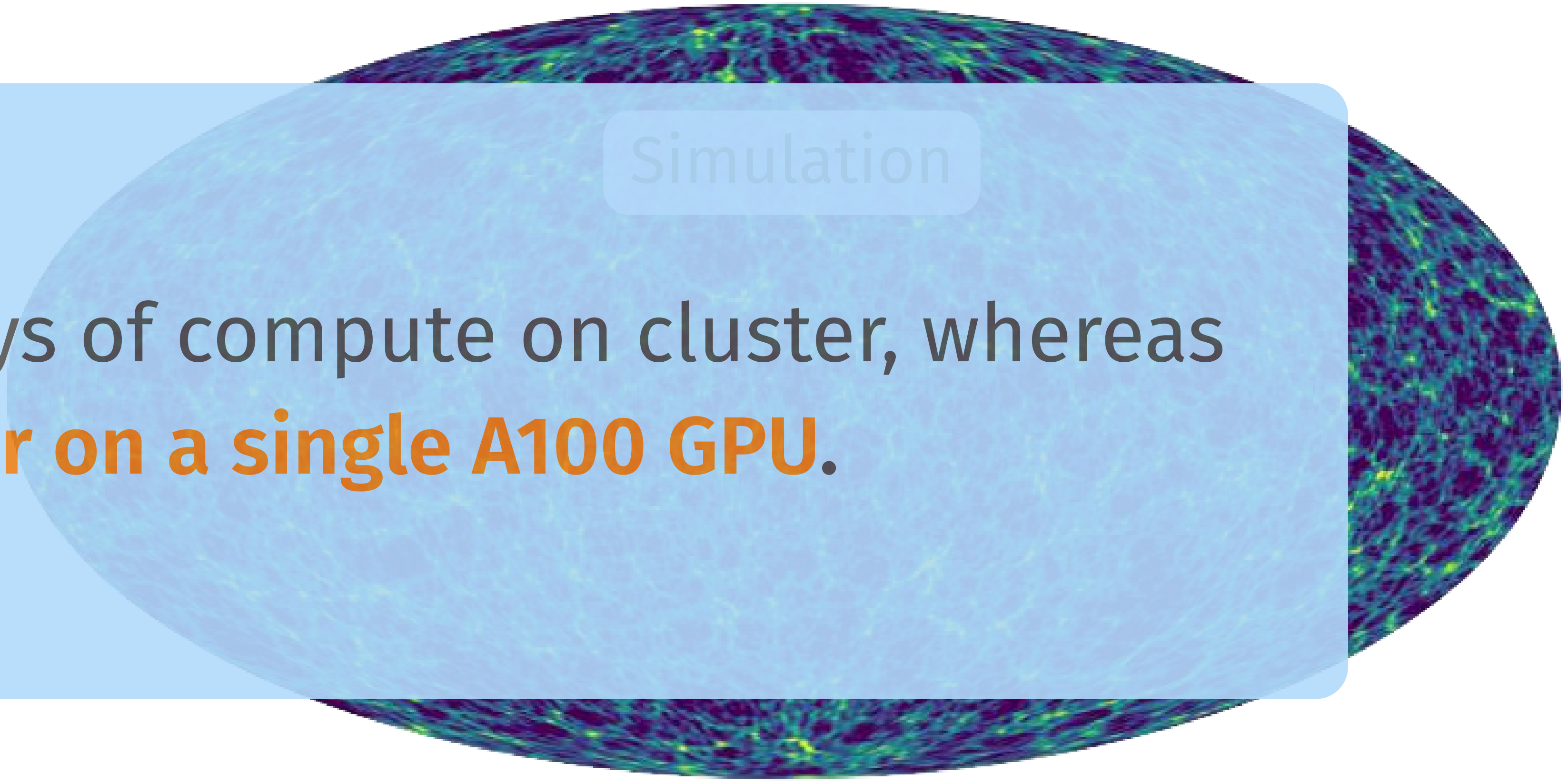
Requires only **single target simulation**.



800,000 CPU hours on a supercomputer → ~1 hour on A100 GPU.

Which field is simulated and which emulated?



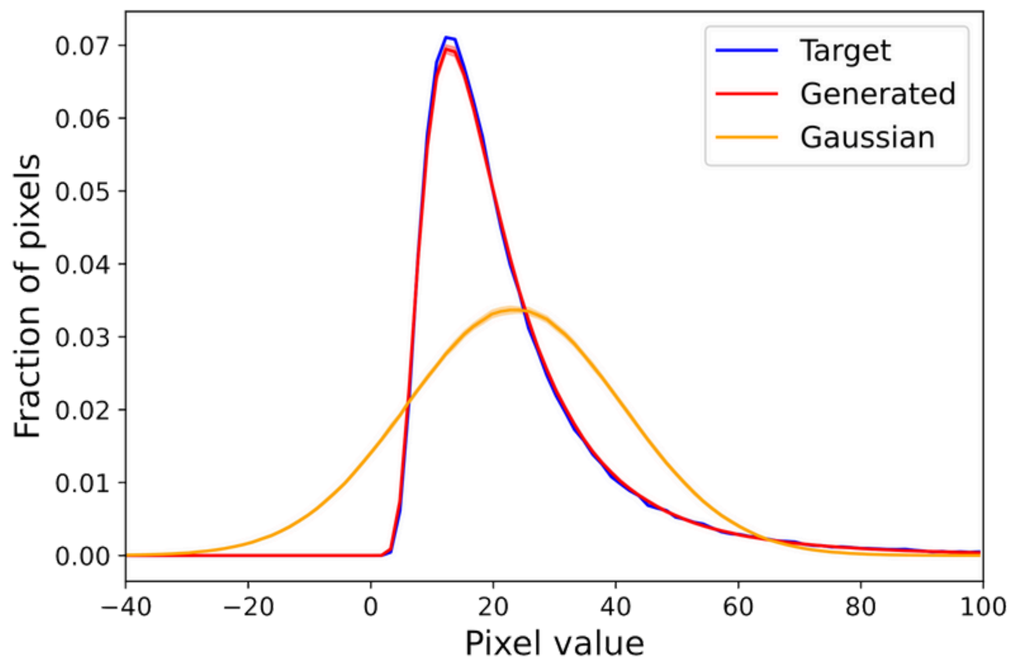


Simulations (n-body) can require days of compute on cluster, whereas emulation requires **<1 hour on a single A100 GPU**.

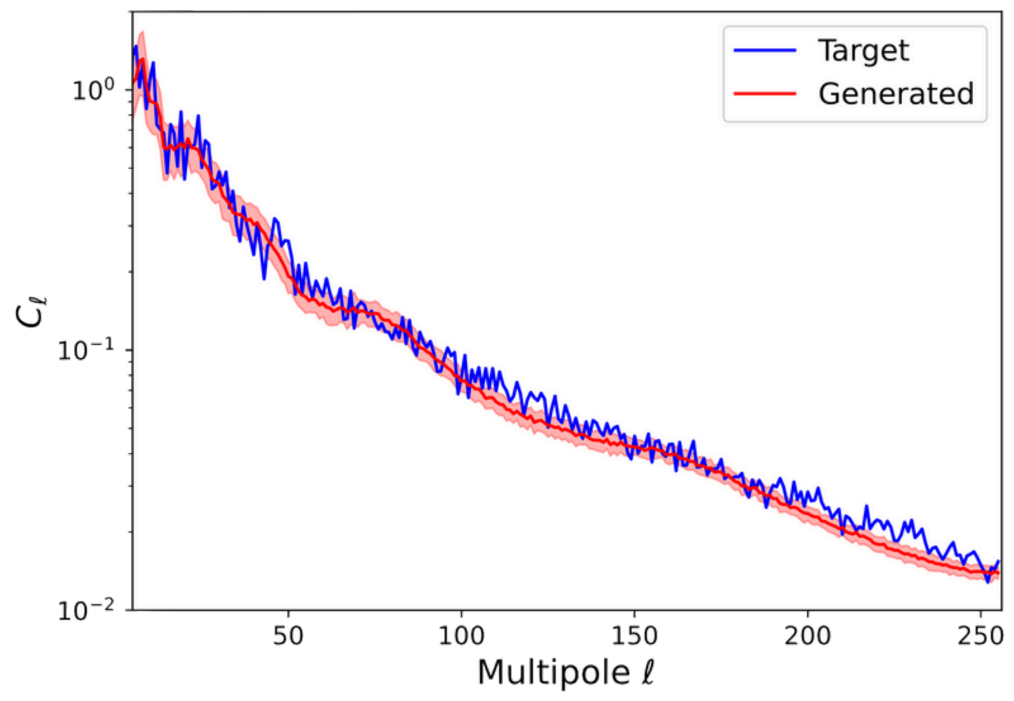
Logarithm (for visualisation) of weak lensing field.

Generative modelling of large-scale structure (LSS)

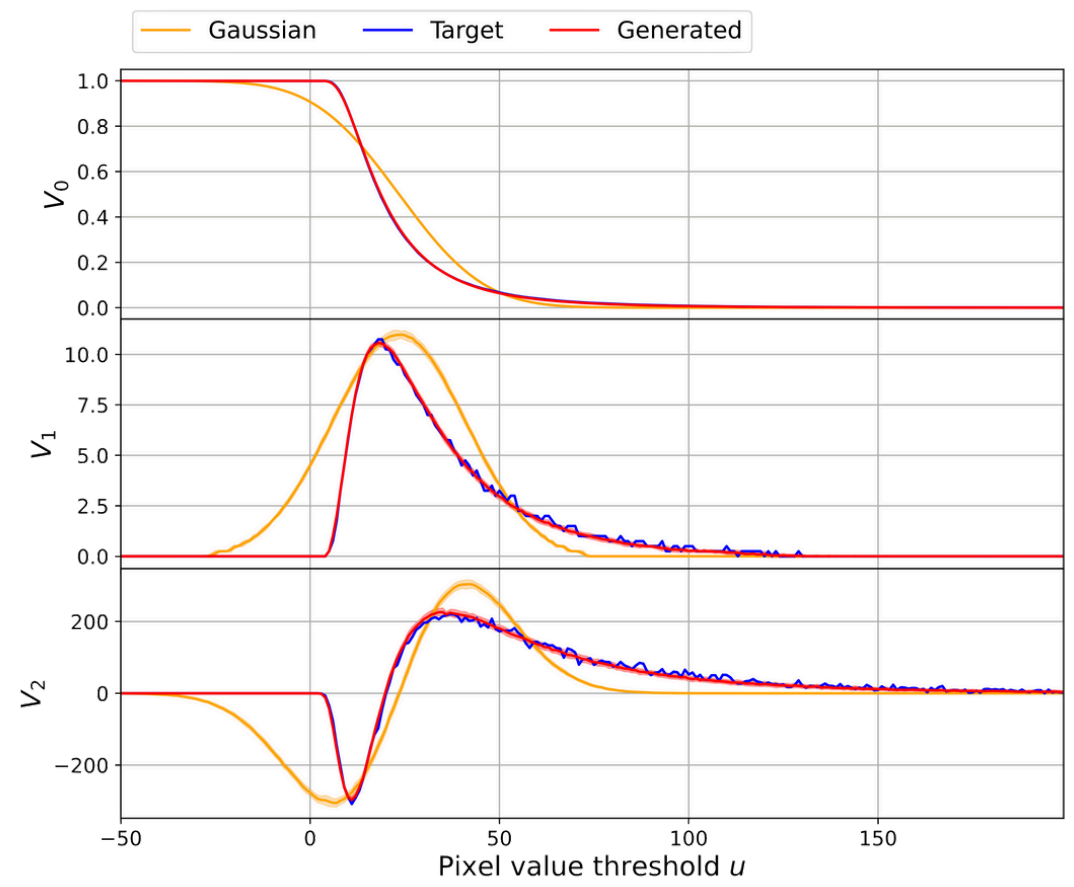
Validation of higher-order statistics.



Pixel distribution



Power spectrum



Minkowski functionals