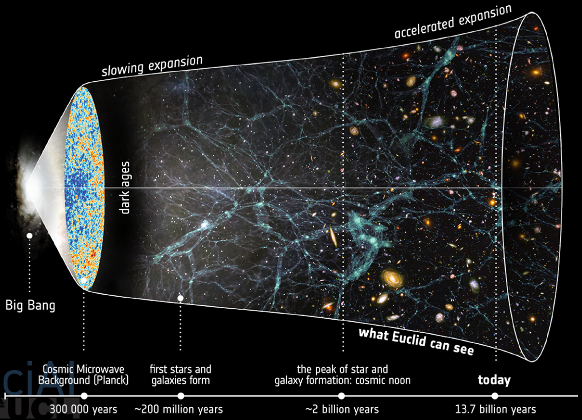
Scientific AI for Cosmology and Beyond

Jason D. McEwen
www.jasonmcewen.org

Scientific AI (SciAI) Group, Mullard Space Science Laboratory (MSSL)
University College London (UCL)

University of Southampton, April 2025

Towards a fundamental understanding of our Universe

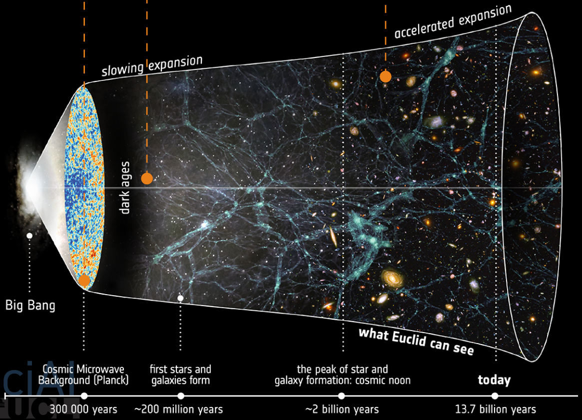


Towards a fundamental understanding of our Universe

What is the origin of structure?

How did luminous large-scale structure form?

What is the nature of dark energy and dark matter?

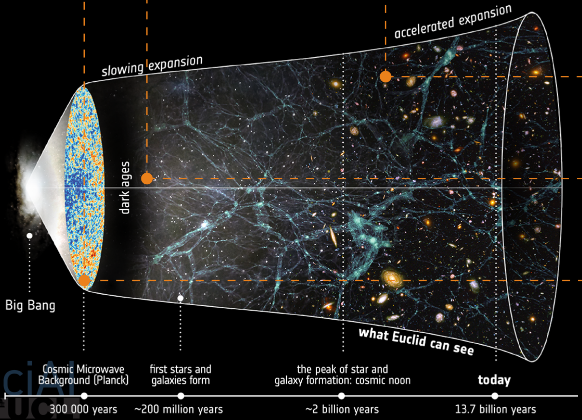


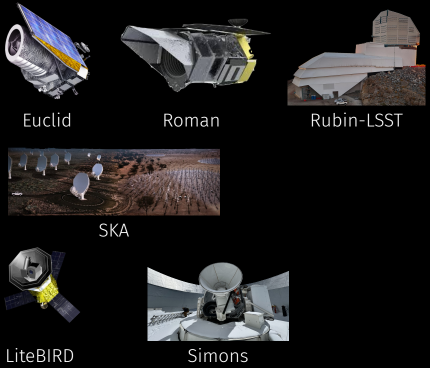
Towards a fundamental understanding of our Universe

What is the origin of structure?

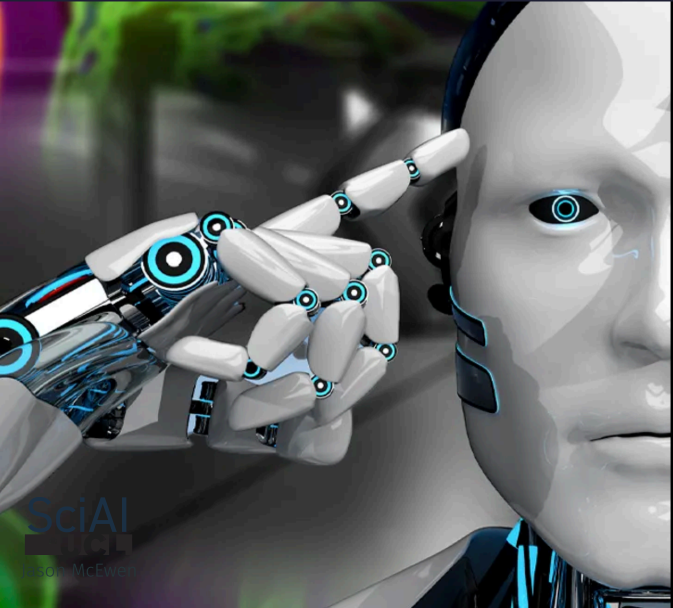
How did luminous large-scale structure form?

What is the nature of dark energy and dark matter?

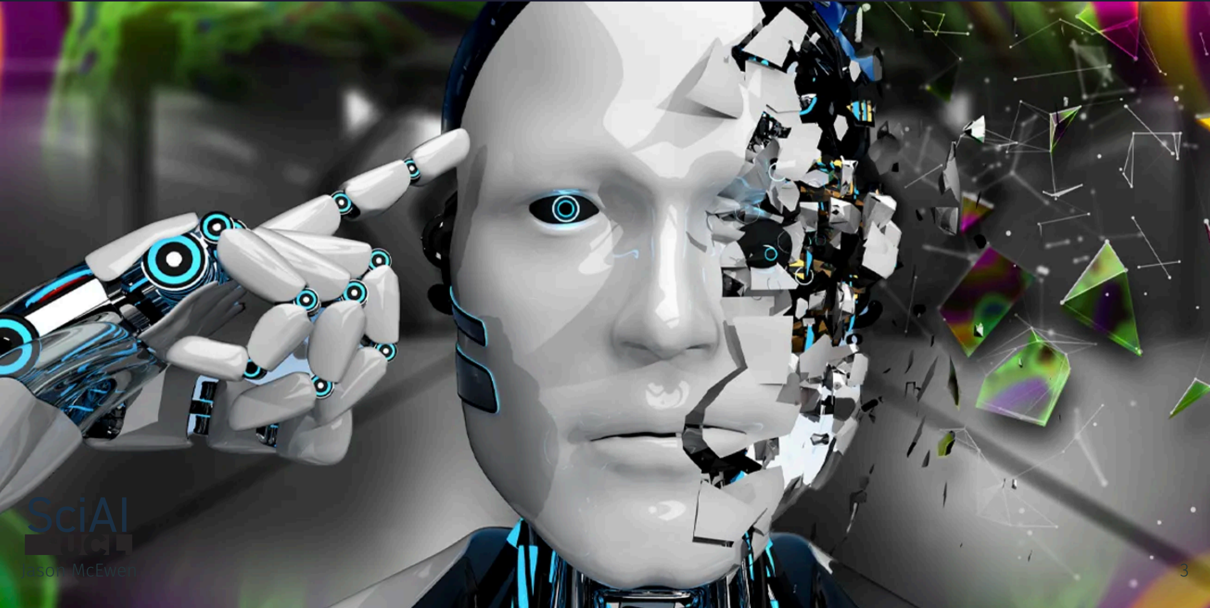




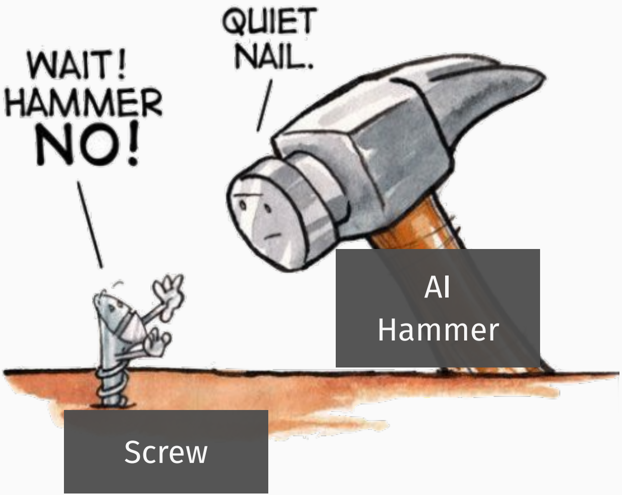
Harnessing AI for science...



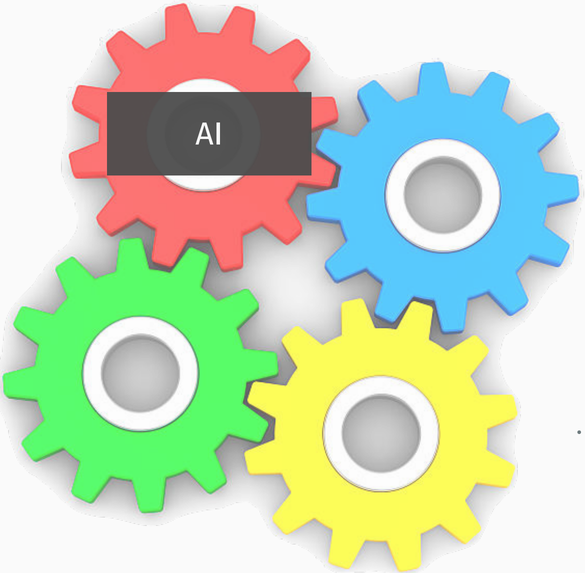
Harnessing AI for science... without hallucinations



The AI hammer

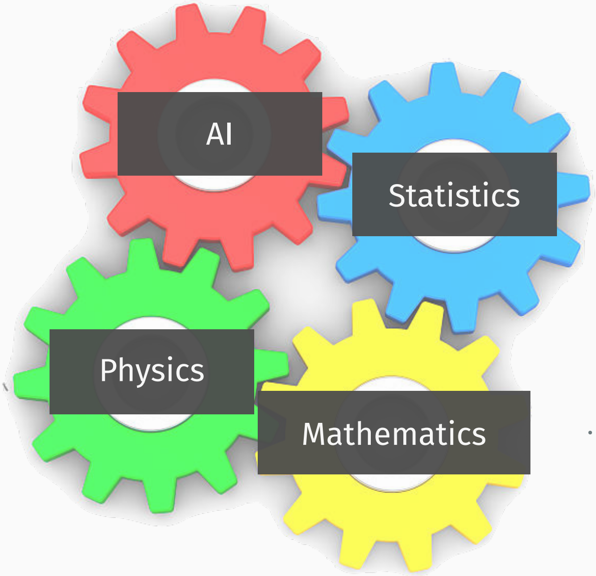


The AI cog



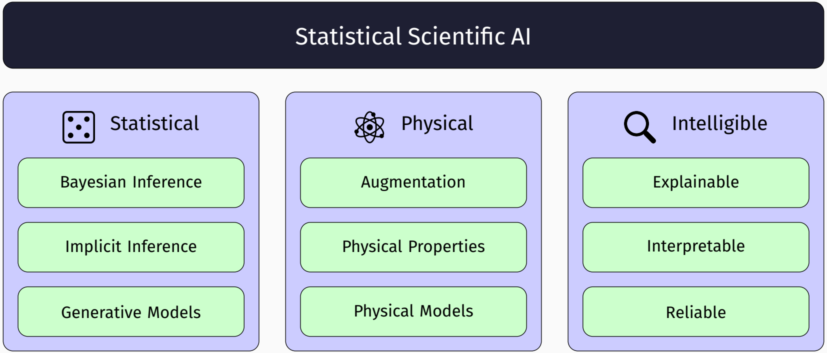
...

The AI cog

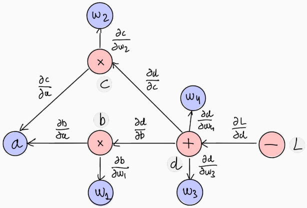


...

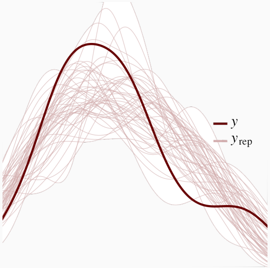
Statistics as the Key to Unlocking AI for Science



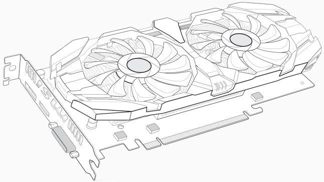
Harnessing modern computing paradigms



Automatic differentiation



Probabilistic programming



GPU acceleration

Outline

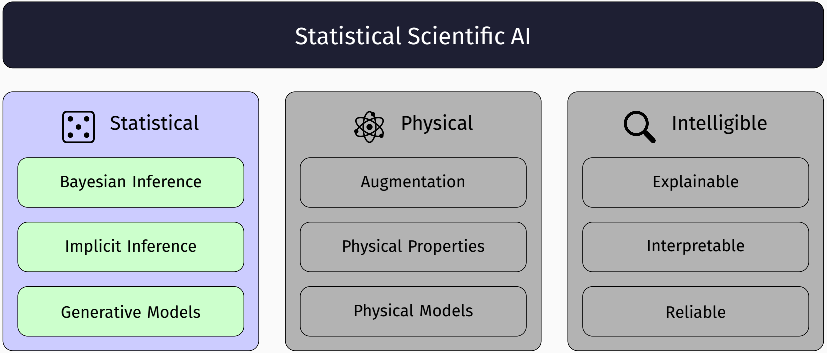
1. Statistical scientific AI
2. AI-enhanced Bayesian inference
3. Geometric AI on spherical manifolds
4. Scalable Bayesian inference with data-driven AI priors

Statistical scientific AI

Statistical AI

Embed a statistical representation of data, models and/or outputs.

(See Murray 2022.)

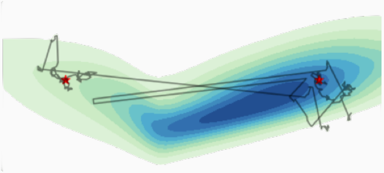


Bayesian inference

AI techniques can be integrated into Bayesian frameworks to **enhance accuracy and computational efficiency**, making some approaches accessible that were previously intractable.

AI techniques can be integrated into Bayesian frameworks to **enhance accuracy and computational efficiency**, making some approaches accessible that were previously intractable.

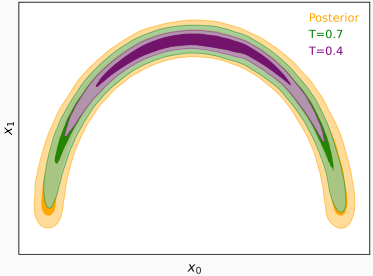
- Enhanced MCMC for parameter estimation (Grabrie *et al.* 2022, Karamanis *et al.* 2022).



Learned proposal distributions

AI techniques can be integrated into Bayesian frameworks to **enhance accuracy and computational efficiency**, making some approaches accessible that were previously intractable.

- Enhanced Bayesian model selection (**`harmonic`**; McEwen *et al.* 2021, Polanska *et al.* McEwen 2023, 2024, Piras *et al.* McEwen 2024, Spurio Mancini *et al.* McEwen 2023, 2024).



Learned harmonic mean estimator (**`harmonic`**)

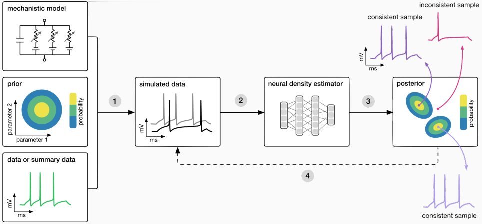
Implicit inference

AI techniques can be used to **learn surrogates of implicit distributions** when they are not tractable or are computationally infeasible.

Implicit inference

AI techniques can be used to **learn surrogates of implicit distributions** when they are not tractable or are computationally infeasible.

- Simulation-based inference (Cranmer *et al.* 2021).

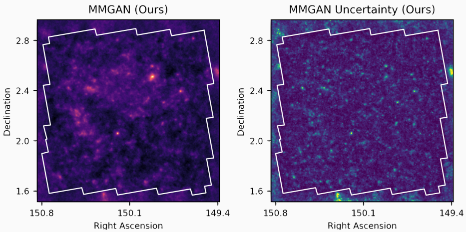


sbi

Implicit inference

AI techniques can be used to **learn surrogates of implicit distributions** when they are not tractable or are computationally infeasible.

- Variational inference
  (Whitney *et al.* McEwen 2024).



Mass mapping with uncertainties by variational inference (Whitney *et al.* McEwen 2024)

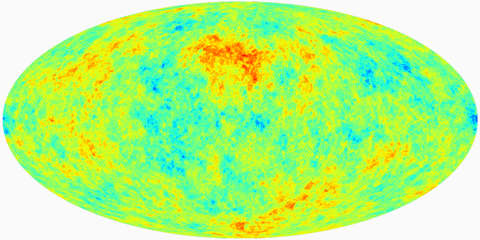
Generative models

Generative models **learn a prior distribution** from data for sampling and/or evaluating probability densities.

Generative models

Generative models **learn a prior distribution** from data for sampling and/or evaluating probability densities.

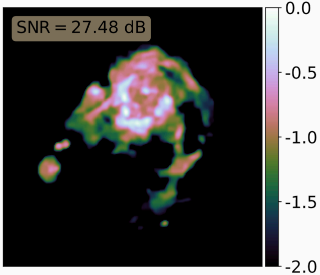
- Emulation: sample from learned prior (Price *et al.* McEwen 2023, Price *et al.* McEwen in prep., Mousset *et al.* McEwen 2024)



Emulated cosmic string maps (**`stringgen`**, Price *et al.* McEwen 2023, Price *et al.* McEwen in prep.)

Generative models **learn a prior distribution** from data for sampling and/or evaluating probability densities.

- Integrate learned priors into analysis
  (McEwen *et al.* 2023, Liaudat *et al.* McEwen 2024)

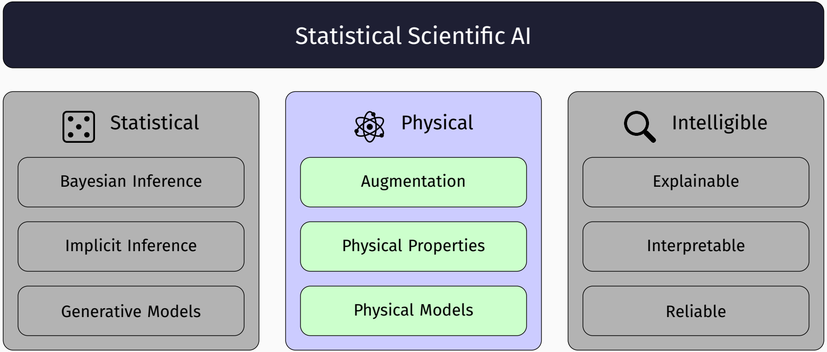


Learn radio galaxy prior
(Liaudat *et al.* McEwen 2024)

Physics Enhanced AI

Embed physical understanding of the world into AI models.

(See review by Karniadakis *et al.* 2021.)



Augmentation

Apply **physical transformations** that data known to satisfy to augment training data $\rightsquigarrow$ AI model **learns physics through training**.

Augmentation

Apply **physical transformations** that data known to satisfy to augment training data $\rightsquigarrow$ AI model **learns physics through training**.

- Common to augment image data-set with rotations, flips, shifts, scales, contrast, ...

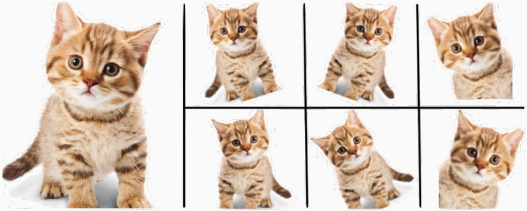
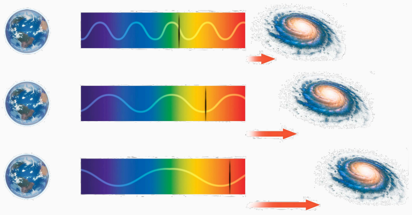


Image augmentation

Augmentation

Apply **physical transformations** that data known to satisfy to augment training data ⇝ AI model **learns physics through training**.

- Redshift augmentation of supernovae (Boone 2019, Alves *et al.* (inc. McEwen) 2022, 2023)



Redshift augmentation

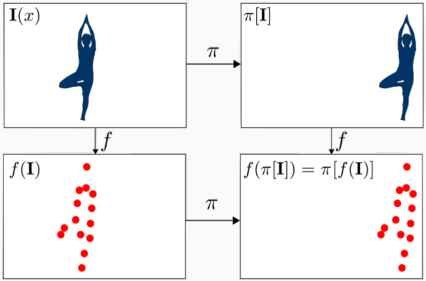
Physical properties: geometries, symmetries, conservation laws

Encode physical properties of the world into AI models (e.g. geometry, symmetries, conservation laws) $\rightsquigarrow$ **Physics embedded in architecture** of AI model.

Physical properties: geometries, symmetries, conservation laws

Encode physical properties of the world into AI models (e.g. geometry, symmetries, conservation laws) $\rightsquigarrow$ **Physics embedded in architecture** of AI model.

- Key factor CNNs so successful is due to encoding translational equivariance.

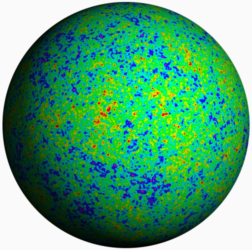


Translational equivariance

Physical properties: geometries, symmetries, conservation laws

Encode physical properties of the world into AI models (e.g. geometry, symmetries, conservation laws) ⇝ **Physics embedded in architecture** of AI model.

- Geometric deep learning on the sphere (Cobb et al. 2021, McEwen et al. 2022, Ocampo, Price & McEwen 2023)



CMB observed on the celestial sphere

Physical models: PINNS and differentiable physics

Encode physical models of world into AI models:

1. Encode dynamics (differential equations) via loss functions (PINNs).
2. Embed full (differentiable) physical models inside AI model.

⇝ Physics learned in training and embedded in model.

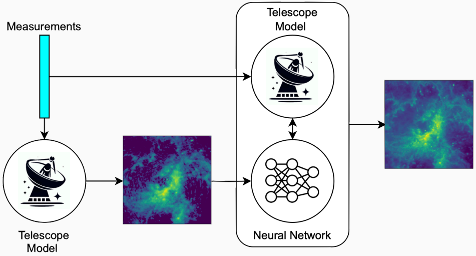
Physical models: PINNS and differentiable physics

Encode physical models of world into AI models:

1. Encode dynamics (differential equations) via loss functions (PINNs).
2. Embed full (differentiable) physical models inside AI model.

⇝ **Physics learned in training and embedded in model.**

- Differentiable physical models
  - Instrument models (Mars *et al.* McEwen 2023, 2024, Liaudat *et al.* McEwen 2024)
  - Physical models (Piras *et al.* McEwen 2024, Spurio Mancini *et al.* McEwen 2024, Whitney *et al.* McEwen in prep.)



Hybrid physics-enhanced AI model (Mars *et al.* McEwen 2023, 2024)

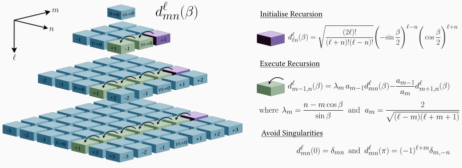
Physical models: PINNS and differentiable physics

Encode physical models of world into AI models:

1. Encode dynamics (differential equations) via loss functions (PINNs).
2. Embed full (differentiable) physical models inside AI model.

⇝ **Physics learned in training and embedded in model.**

- Differentiable mathematical methods
  - Fourier transforms
  - Spherical harmonic transforms (**s2fft**; Price & McEwen 2023)
  - Spherical wavelet transforms (**s2wav**; Price *et al.* McEwen 2024)
  - Spherical scattering transforms (**s2scat**; Mousset *et al.* McEwen 2024)

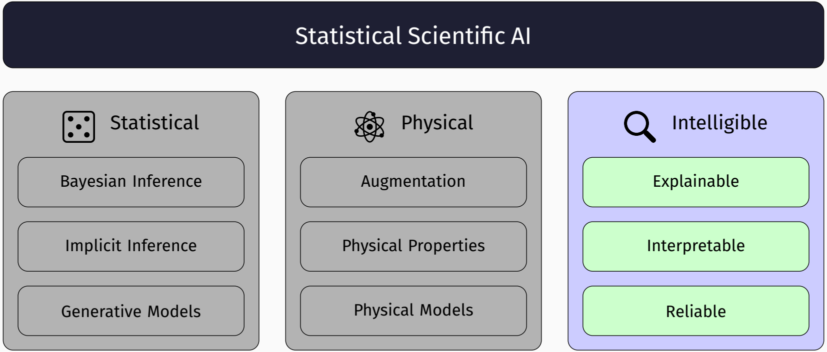


Differentiable and GPU-friendly recursions (Price & McEwen 2023)

Intelligible AI

AI methods that are able to be understood by humans and are reliable.

(See Weld & Bansal 2018, Ras *et al.* 2020.)



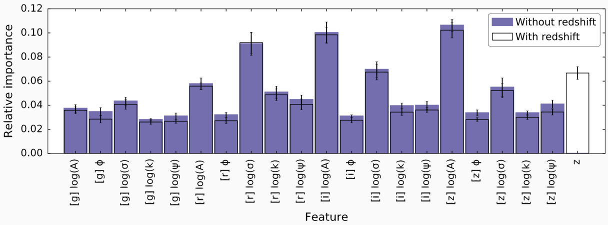
Explainability

Explainable AI techniques may or may not be interpretable themselves but their **outputs can be explained to humans.**

Explainability

Explainable AI techniques may or may not be interpretable themselves but their **outputs can be explained to humans.**

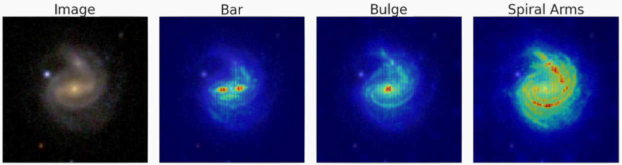
- Feature importances (Lochner *et al.* (inc. McEwen) 2016)



Supernova feature importances

Explainable AI techniques may or may not be interpretable themselves but their **outputs can be explained to humans.**

- Saliency maps
  (Bhambra *et al.* 2022)



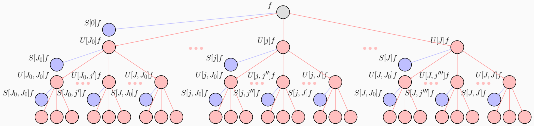
Galaxy saliency mapping

Interpretability

Interpretable AI models are **white boxes that can be understood by humans**.

Interpretable AI models are **white boxes that can be understood by humans**.

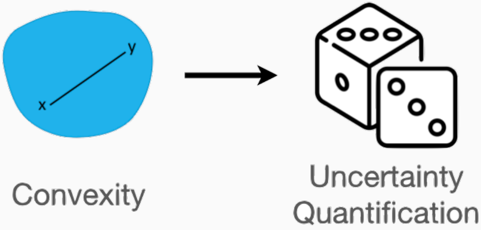
- Designed models such as wavelet scattering networks (McEwen *et al.* 2022, Mousset *et al.* McEwen 2024)



Scattering network (McEwen *et al.* 2022)

Interpretable AI models are **white boxes that can be understood by humans**.

- Interpretable constraints on AI models
  (Liaudat *et al.* McEwen 2024)

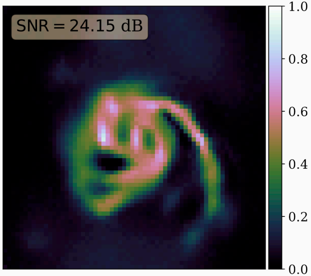


Impose convexity on learned model

Interpretability

Interpretable AI models are **white boxes that can be understood by humans.**

- Deep priors learned from training data (McEwen *et al.* 2023, Liaudat *et al.* McEwen 2024)



Compute Bayesian evidence for model selection (McEwen *et al.* 2023)

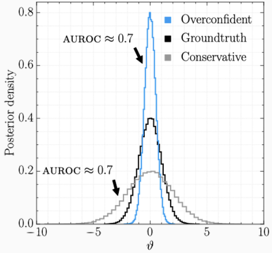
Reliability

Reliability and validity **critical for science** to have confidence in results of AI models. Closely coupled with a **meaningful statistical distribution** of outputs.

Reliability

Reliability and validity **critical for science** to have confidence in results of AI models. Closely coupled with a **meaningful statistical distribution** of outputs.

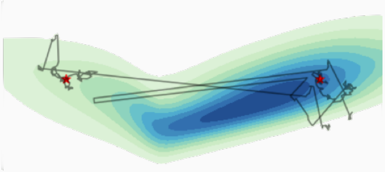
- Validity of statistical distributions (Lueckmann *et al.* 2021, Hermans *et al.* 2022, Cannon *et al.* 2023)



Validity of distribution (Hermans *et al.* 2022)

Reliability and validity **critical for science** to have confidence in results of AI models. Closely coupled with a **meaningful statistical distribution** of outputs.

- Integrate into statistical frameworks to inherit theoretical guarantees
  ↪ statistical component critical
  (McEwen *et al.* 2023, Liaudat *et al.* McEwen 2024, McEwen *et al.* 2021, Polanska *et al.* McEwen 2023, 2024, Piras *et al.* McEwen 2024)



Inherit guarantees from overarching statistical frameworks

Reliability

Reliability and validity **critical for science** to have confidence in results of AI models. Closely coupled with a **meaningful statistical distribution** of outputs.

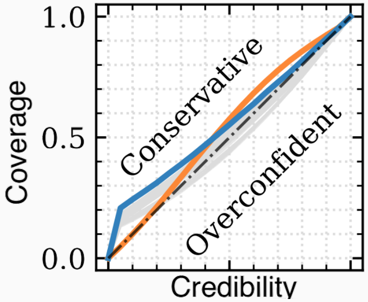
- Design to ensure conservative and avoid mode collapse (Delaunoy *et al.* 2022, Price *et al.* McEwen 2023, Whitney *et al.* McEwen 2024)

Recover probability distribution over full underlying data manifold (Price *et al.* McEwen 2023)

Reliability

Reliability and validity **critical for science** to have confidence in results of AI models. Closely coupled with a **meaningful statistical distribution** of outputs.

- Extensive validation checks:
  - Coverage testing (Lemos *et al.* 2023)
  - Simulation-based calibration checks (Talts *et al.* 2020)
  - Classifier two-sample tests (C2ST) (Lopez-Paz & Oquab 2017)
  - ...



Coverage analysis
(Cannon *et al.* 2023)

AI-enhanced Bayesian inference

First, let's set the notation (and colour code)...

Bayesian inference: parameter estimation

First, let's set the notation (and colour code)...

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:

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

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:

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

Bayesian inference: model comparison

By Bayes' theorem for model M_j:

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

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

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

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

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

Bayesian inference: model comparison

By Bayes' theorem for model M_j:

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

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

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

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

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

$\rightsquigarrow$ Challenging computational problem.

Challenge of Bayesian model selection

Naive Monte Carlo integration to compute marginal likelihood not effective.

Require **tailored computational techniques**.

Challenge of Bayesian model selection

Naive Monte Carlo integration to compute marginal likelihood not effective.

Require **tailored computational techniques**.

Challenges:

- Support **aribtrary sampling** strategies (*e.g.* accelerated).
- Support **implicit inference** (*e.g.* simulation-based inference and variational inference).
- Scale to **high-dimensions** (*e.g.* images).
- Support **data-driven AI priors** (*e.g.* priors captured by generative models).

The problem of nested sampling

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**...

The problem of nested sampling

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**.

SciAI UCL

The problem of nested sampling

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.

Learning model posterior odds ratio

Alternatively, learn model posterior odds ratio directly by leveraging the **likelihood ratio trick** (Goodfellow *et al.* 2014, Cranmer *et al.* 2020).

Train a classifier to distinguish models, *e.g.* with cross-entropy loss, which learns ratio

$$r(x) = \frac{p(M_1 \,|\, x)}{p(M_2 \,|\, x)}.$$

Numerous works considering this approach or variants (Radev *et al.* 2021, Elsemüller *et al.* 2024, Jeffrey *et al.* 2024, Karchev *et al.* 2023).

Learning model posterior odds ratio

Alternatively, learn model posterior odds ratio directly by leveraging the **likelihood ratio trick** (Goodfellow *et al.* 2014, Cranmer *et al.* 2020).

Train a classifier to distinguish models, *e.g.* with cross-entropy loss, which learns ratio

$$r(x) = \frac{p(M_1 \,|\, x)}{p(M_2 \,|\, x)}.$$

Numerous works considering this approach or variants (Radev *et al.* 2021, Elsemüller *et al.* 2024, Jeffrey *et al.* 2024, Karchev *et al.* 2023).

$\rightsquigarrow$ No consistency guarantees for $\mathcal{M}$-open scenario.

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

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

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

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

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

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

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

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)

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

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!**

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)

Learned harmonic mean estimator

Propose the ***learned* harmonic mean estimator**, leveraging AI to solve the catastrophic failure of the original harmonic mean (McEwen *et al.* 2021).

Chris Wallis

Matt Price

Alessio Spurio Mancini

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 of harmonic mean estimator

Alternative interpretation of harmonic mean relationship:

importance sampling

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

Importance sampling interpretation:

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

SciAI UCL
Jason McEwen

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**
$\rightsquigarrow$ must **not** have fatter tails than posterior.

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

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

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**
$\rightsquigarrow$ must **not** have fatter tails than posterior.

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

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

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

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

Variety of cases been considered:

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

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

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)

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

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

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:

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

Learned harmonic mean estimator

Learn an approximation of the optimal target distribution:

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

- 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:

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

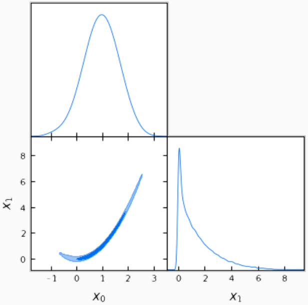
Solve by bespoke **mini-batch stochastic gradient descent**.

Cross-validation to select density estimation model and hyperparameters.

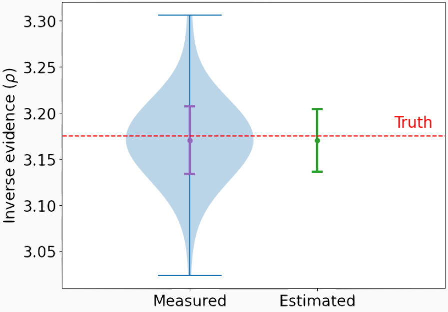
Rosenbrock example

Rosenbrock function is the classical example of a **pronounced thin curving degeneracy**, with likelihood defined by

$$f(\theta) = \sum_{i=1}^{n-1} \left[(a - \theta_i)^2 + b(\theta_{i+1} - \theta_i^2)^2 \right] , \qquad \log(\mathcal{L}(\theta)) = -f(\theta) \, .$$



Posterior by MCMC sampling

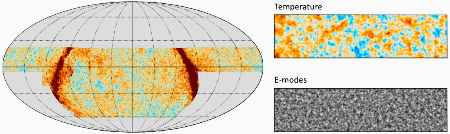


Reciprocal evidence

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× **acceleration**

SciAI UCL

Constraining tails of target approach 2: normalizing flows

Learned harmonic mean with normalizing flows (Polanska *et al.* 2023, 2024)

Elegant way to constrain tails of target distribution $\varphi(\theta)$.

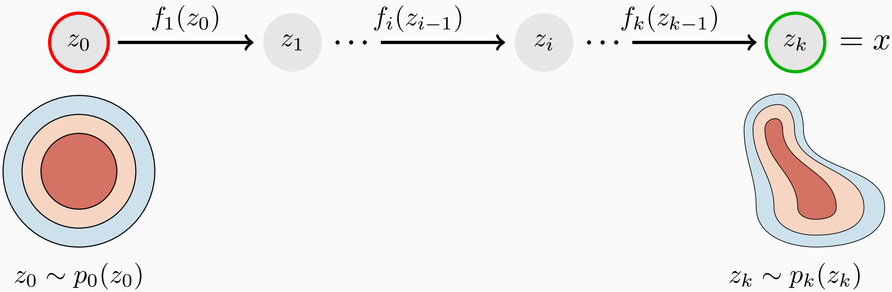
Alicja Polanska

Matt Price

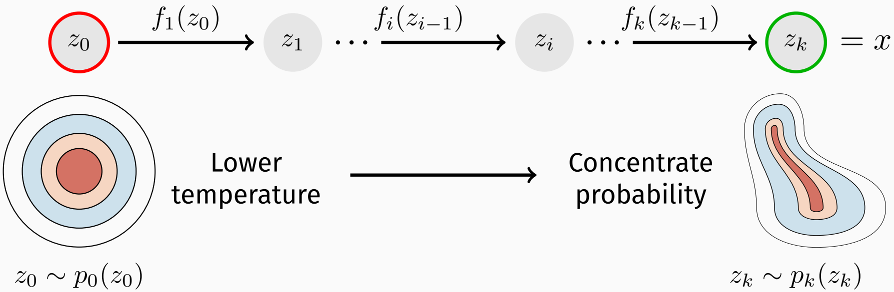
Davide Piras

Alessio Spurio Mancini

Constraining tails of target approach 2: normalizing flows

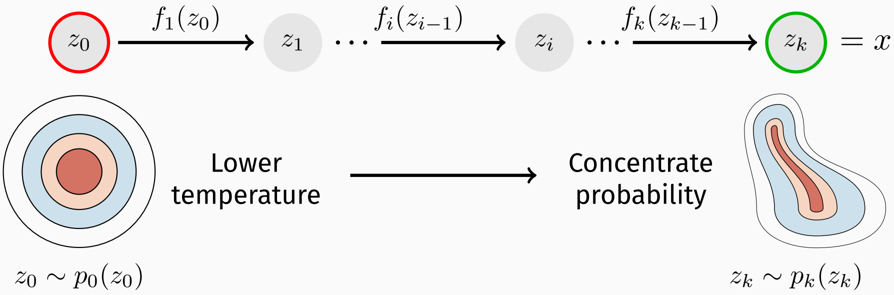


Constraining tails of target approach 2: normalizing flows



SciAI UCL
Jason McEwen

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.

Harmonic code

GitHub harmonic | Tests passing | docs passing | codecov 96% | pypi package 1.2.3 | License GPL | arXiv 2111.12720
arXiv 2207.04037 | arXiv 2307.00048 | arXiv 2405.05969

HARMONIC

`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

JAX: Automatic differentiation + GPU acceleration

Leveraging AI to accelerate Bayesian inference further

4 pillars of AI-accelerated Bayesian inference (Piras *et al.* McEwen 2024).

Leveraging AI to accelerate Bayesian inference further

4 pillars of AI-accelerated Bayesian inference (Piras *et al.* McEwen 2024).

1. **Emulation** to accelerate physical model encapsulated in likelihood,
   e.g. CosmoPower (Spurio Mancini *et al.* 2022, Piras & Spurio Mancini 2023)

Leveraging AI to accelerate Bayesian inference further

4 pillars of AI-accelerated Bayesian inference (Piras *et al.* McEwen 2024).

1. **Emulation** to accelerate physical model encapsulated in likelihood,
   e.g. CosmoPower (Spurio Mancini *et al.* 2022, Piras & Spurio Mancini 2023)
2. **Differentiable and probabilistic programming** to accelerate gradient calculations and development of statistical models, *e.g.* JAX, NumPyro

Leveraging AI to accelerate Bayesian inference further

4 pillars of AI-accelerated Bayesian inference (Piras *et al.* McEwen 2024).

1. **Emulation** to accelerate physical model encapsulated in likelihood,
   e.g. CosmoPower (Spurio Mancini *et al.* 2022, Piras & Spurio Mancini 2023)
2. **Differentiable and probabilistic programming** to accelerate gradient calculations and development of statistical models, *e.g.* JAX, NumPyro
3. **Scalable (gradient-based) MCMC sampling** to accelerate sampling and parameter estimation, *e.g.* NUTS

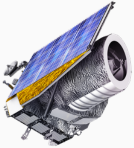
Leveraging AI to accelerate Bayesian inference further

4 pillars of AI-accelerated Bayesian inference (Piras *et al.* McEwen 2024).

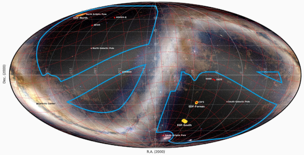
1. **Emulation** to accelerate physical model encapsulated in likelihood, *e.g.* CosmoPower (Spurio Mancini *et al.* 2022, Piras & Spurio Mancini 2023)
2. **Differentiable and probabilistic programming** to accelerate gradient calculations and development of statistical models, *e.g.* JAX, NumPyro
3. **Scalable (gradient-based) MCMC sampling** to accelerate sampling and parameter estimation, *e.g.* NUTS
4. **Scalable and decoupled marginal likelihood computation** to accelerate model selection, *e.g.* learned harmonic mean (McEwen *et al.*2021, Polanska *et al.* 2023, 2024)

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



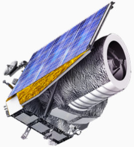
Observation field

37D vs 39D models:	$\log(z_{\Lambda\text{CDM}})$	$\log(z_{w_0 w_a\text{CDM}})$	$\log \text{BF}_{\Lambda\text{CDM-}w_0 w_a\text{CDM}}$	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)

$\rightsquigarrow$ 120$\times$ **acceleration**

Euclid-Rubin-Roman (3× 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:	$\log(z_{\Lambda\mathrm{CDM}})$	$\log(z_{w_0 w_a \mathrm{CDM}})$	$\log \mathrm{BF}$	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)

$\rightsquigarrow$ **Opens up new analyses (550× acceleration)**

Simulation-based inference (SBI)

Simulation-based inference (aka. likelihood-free inference) seeks to perform Bayesian inference by **estimating the posterior** $p(\theta \mid x_o, M)$ of **parameters** θ for **observed data** x_o using **simulations only**.

Key advantages:

- Forward modelling of complex physics, systematics, observational process.
- No assumptions on the form of the likelihood.

Neural simulation-based inference (SBI)

Neural posterior estimation

(Papamakarios & Murray 2016)

Learn **surrogate of posterior** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)p(\theta)}[\log q_\phi(\theta\,|\,x)].$$

Neural likelihood estimation

(Papamakarios *et al.* 2019)

Learn **surrogate of likelihood** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)\tilde{p}(\theta)}[\log q_\phi(x\,|\,\theta)].$$

Neural ratio estimation

(Hermans *et al.* 2020, Durkan *et al.* 2020)

Learn **surrogate of posterior-to-prior ratio** by training classifier with loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta,\theta'\sim p(x\,|\,\theta)p(\theta)p(\theta')}[\log(\sigma(\log r_\phi(\theta;x))) + \log(-\sigma(\log r_\phi(\theta';x)))],$$

where $\sigma(\cdot)$ is the sigmoid function.

Neural simulation-based inference (SBI)

Neural posterior estimation
(Papamakarios & Murray 2016)

Learn **surrogate of posterior** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)p(\theta)}[\log q_\phi(\theta\,|\,x)].$$

- ✖ Simulations from prior.

Neural likelihood estimation
(Papamakarios *et al.* 2019)

Learn **surrogate of likelihood** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)\tilde{p}(\theta)}[\log q_\phi(x\,|\,\theta)].$$

- ✔ Simulations from any proposal.

Neural ratio estimation
(Hermans *et al.* 2020, Durkan *et al.* 2020)

Learn **surrogate of posterior-to-prior ratio** by training classifier with loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta,\theta'\sim p(x\,|\,\theta)p(\theta)p(\theta')}[\log(\sigma(\log r_\phi(\theta;x))) + \log(-\sigma(\log r_\phi(\theta';x)))],$$

where $\sigma(\cdot)$ is the sigmoid function.

- ✖ Simulations from prior.

Neural simulation-based inference (SBI)

Neural posterior estimation
(Papamakarios & Murray 2016)

Learn **surrogate of posterior** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta \sim p(x \mid \theta)p(\theta)}[\log q_\phi(\theta \mid x)].$$

- ✖ Simulations from prior.
- ✖ Density estimator architecture.

Neural likelihood estimation
(Papamakarios *et al.* 2019)

Learn **surrogate of likelihood** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta \sim p(x \mid \theta)\tilde{p}(\theta)}[\log q_\phi(x \mid \theta)].$$

- ✔ Simulations from any proposal.
- ✖ Density estimator architecture.

Neural ratio estimation
(Hermans *et al.* 2020, Durkan *et al.* 2020)

Learn **surrogate of posterior-to-prior ratio** by training classifier with loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta,\theta' \sim p(x \mid \theta)p(\theta)p(\theta')}[\log(\sigma(\log r_\phi(\theta; x))) + \log(-\sigma(\log r_\phi(\theta'; x)))],$$

where $\sigma(\cdot)$ is the sigmoid function.

- ✖ Simulations from prior.
- ✔ Flexible architecture.

SciAI UCL

Jason McEwen

Neural simulation-based inference (SBI)

Neural posterior estimation
(Papamakarios & Murray 2016)

Learn **surrogate of posterior** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)p(\theta)}[\log q_\phi(\theta\,|\,x)].$$

- ✗ Simulations from prior.
- ✗ Density estimator architecture.
- ✓ Draw samples from surrogate.

Neural likelihood estimation
(Papamakarios *et al.* 2019)

Learn **surrogate of likelihood** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)\tilde{p}(\theta)}[\log q_\phi(x\,|\,\theta)].$$

- ✓ Simulations from any proposal.
- ✗ Density estimator architecture.
- ✗ MCMC sampling.

Neural ratio estimation
(Hermans *et al.* 2020, Durkan *et al.* 2020)

Learn **surrogate of posterior-to-prior ratio** by training classifier with loss

$$\mathscr{L} = -\,\mathbb{E}_{x,\theta,\theta'\sim p(x\,|\,\theta)p(\theta)p(\theta')}[\log(\sigma(\log r_\phi(\theta;x))) \\ + \log(-\sigma(\log r_\phi(\theta';x)))],$$

where $\sigma(\cdot)$ is the sigmoid function.

- ✗ Simulations from prior.
- ✓ Flexible architecture.
- ✗ MCMC sampling.

Neural simulation-based inference (SBI)

Neural posterior estimation
(Papamakarios & Murray 2016)

Learn **surrogate of posterior** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)p(\theta)}[\log q_\phi(\theta\,|\,x)].$$

- ✗ Simulations from prior.
- ✗ Density estimator architecture.
- ✓ Draw samples from surrogate.
- ✓ Theoretical accuracy guarantees.

Neural likelihood estimation
(Papamakarios *et al.* 2019)

Learn **surrogate of likelihood** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)\tilde{p}(\theta)}[\log q_\phi(x\,|\,\theta)].$$

- ✓ Simulations from any proposal.
- ✗ Density estimator architecture.
- ✗ MCMC sampling.
- ✓ **Theoretical accuracy guarantees.**

Neural ratio estimation
(Hermans *et al.* 2020, Durkan *et al.* 2020)

Learn **surrogate of posterior-to-prior ratio** by training classifier with loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta,\theta'\sim p(x\,|\,\theta)p(\theta)p(\theta')}[\log(\sigma(\log r_\phi(\theta;x))) + \log(-\sigma(\log r_\phi(\theta';x)))],$$

where $\sigma(\cdot)$ is the sigmoid function.

- ✗ Simulations from prior.
- ✓ Flexible architecture.
- ✗ MCMC sampling.
- ✗ Density chasm problem.

SciAI UCL

Neural simulation-based inference (SBI)

Neural posterior estimation
(Papamakarios & Murray 2016)

Learn **surrogate of posterior** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)p(\theta)}[\log q_\phi(\theta\,|\,x)].$$

- ✗ Simulations from prior.
- ✗ Density estimator architecture.
- ✓ Draw samples from surrogate.
- ✓ Theoretical accuracy guarantees.

Neural likelihood estimation
(Papamakarios *et al.* 2019)

Learn **surrogate of likelihood** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)\tilde{p}(\theta)}[\log q_\phi(x\,|\,\theta)].$$

- ✓ Simulations from any proposal.
- ✗ Density estimator architecture.
- ✗ MCMC sampling.
- ✓ Theoretical accuracy guarantees.
- ✓ Surrogate likelihood available.

Neural ratio estimation
(Hermans *et al.* 2020, Durkan *et al.* 2020)

Learn **surrogate of posterior-to-prior ratio** by training classifier with loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta,\theta'\sim p(x\,|\,\theta)p(\theta)p(\theta')}[\log(\sigma(\log r_\phi(\theta;x))) + \log(-\sigma(\log r_\phi(\theta';x)))],$$

where $\sigma(\cdot)$ is the sigmoid function.

- ✗ Simulations from prior.
- ✓ Flexible architecture.
- ✗ MCMC sampling.
- ✗ Density chasm problem.

SciAI UCL

Jason McEwen

Neural simulation-based inference (SBI)

Neural posterior estimation
(Papamakarios & Murray 2016)

Learn **surrogate of posterior** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)p(\theta)}[\log q_\phi(\theta\,|\,x)].$$

- ✗ Simulations from prior.
- ✗ Density estimator architecture.
- ✓ Draw samples from surrogate.
- ✓ Theoretical accuracy guarantees.

Neural likelihood estimation
(Papamakarios *et al.* 2019)

Learn **surrogate of likelihood** by minimising loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta\sim p(x\,|\,\theta)\tilde{p}(\theta)}[\log q_\phi(x\,|\,\theta)].$$

- ✓ Simulations from any proposal.
- ✗ Density estimator architecture.
- ✗ MCMC sampling.
- ✓ Theoretical accuracy guarantees.
- ✓ Surrogate likelihood available.

Neural ratio estimation
(Hermans *et al.* 2020, Durkan *et al.* 2020)

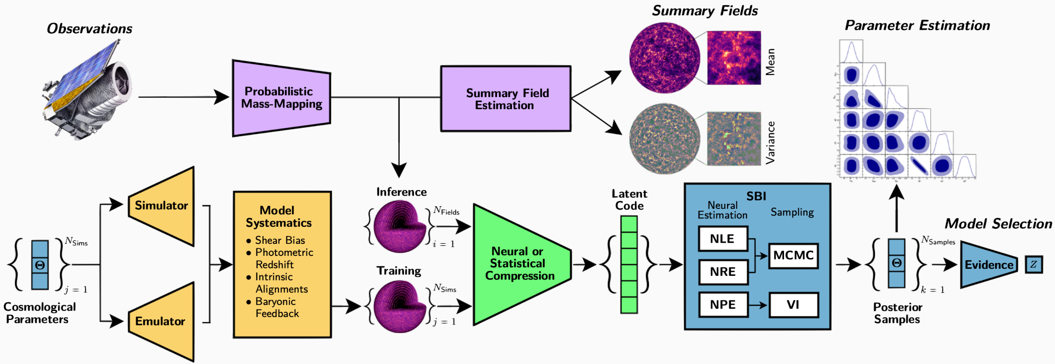
Learn **surrogate of posterior-to-prior ratio** by training classifier with loss

$$\mathscr{L} = -\mathbb{E}_{x,\theta,\theta'\sim p(x\,|\,\theta)p(\theta)p(\theta')}[\log(\sigma(\log r_\phi(\theta;x))) + \log(-\sigma(\log r_\phi(\theta';x)))],$$

where $\sigma(\cdot)$ is the sigmoid function.

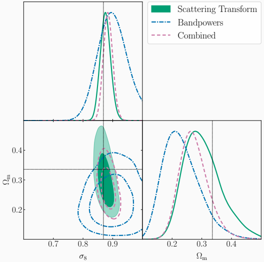
- ✗ Simulations from prior.
- ✓ Flexible architecture.
- ✗ MCMC sampling.
- ✗ Density chasm problem.

Field-level SBI pipeline for weak gravitational lensing

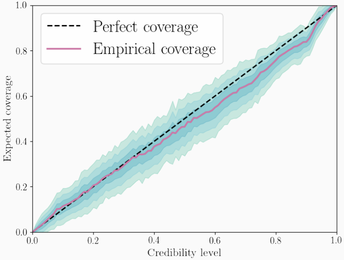


Effectiveness of field-level SBI for weak gravitational lensing

Effectiveness of field-level SBI demonstrated already in **small-field planar setting**.



Posterior



Coverage

Field-level SBI with scattering transforms (Lin, Joachimi & McEwen 2024)

⇝ **Tightest cosmic shear constraints to date** from SBI
(Gatti *et al.* 2023, Jeffrey *et al.* 2024, Cheng *et al.* 2024).

Could field-level SBI distinguish dynamical dark energy?

Recent results from DESI experiment provide tantalising hints of dynamical dark energy (Adame *et al.* 2024a, 2024b).

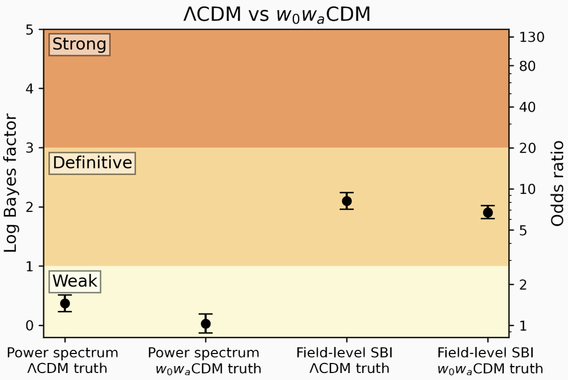
If these results reflected true underlying nature of the Universe, **could a field-level SBI analysis of a Stage IV survey distinguish dynamical dark energy definitively?** (Spurio Mancini *et al.* 2024)

Alessio Spurio Mancini

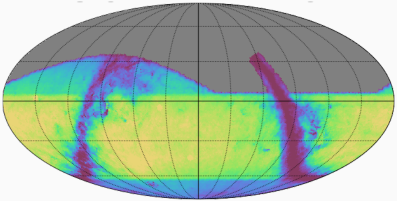
Kiyam Lin

Could field-level SBI distinguish dynamical dark energy?

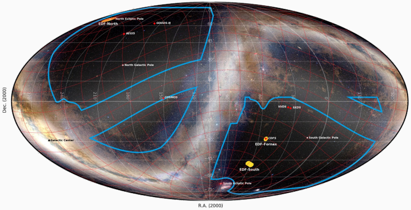
If these results reflected true underlying nature of the Universe, **could a field-level SBI analysis of a Stage IV survey distinguish dynamical dark energy definitively?**
(Spurio Mancini *et al.* 2024)



Wide-field surveys require analysis techniques on spherical manifolds



Rubin-LSST



Euclid

Sky coverage of imminent Stage IV galaxy surveys

⇝ Wide-field surveys require **spherical analysis methods defined on the curved sky**.

Geometric AI on spherical manifolds

Wavelet scattering network representations are an excellent representation space for statistical characterization and generative modelling of fields.

Inspired by CNNs but designed rather than learned filters (Mallat 2012).

⇝ Scattering networks on the sphere
(McEwen et al. 2022)

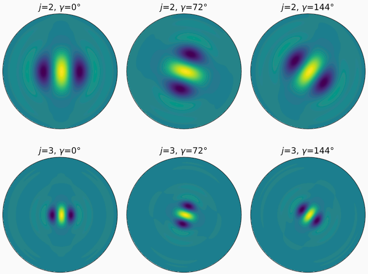
⇝ Generative models of astrophysical fields with scattering transforms on the sphere
(Mousset *et al.* McEwen 2024)

Wavelets on the sphere

Adopt scale-discretized wavelets on the sphere (e.g. McEwen et al. 2018, McEwen et al. 2015).

Wavelets $\psi_j \in \mathrm{L}^2(\mathbb{S}^2)$ capture spatially-localised, high-frequency signal content at scale j.

Scaling function $\phi \in \mathrm{L}^2(\mathbb{S}^2)$ captures spatially-localised, low-frequency content.



Orthographic plot of spherical wavelets.

Wavelets on the sphere

Adopt scale-discretized wavelets on the sphere (e.g. McEwen et al. 2018, McEwen et al. 2015).

Wavelets $\psi_j \in \mathrm{L}^2(\mathbb{S}^2)$ capture spatially-localised, high-frequency signal content at scale j.

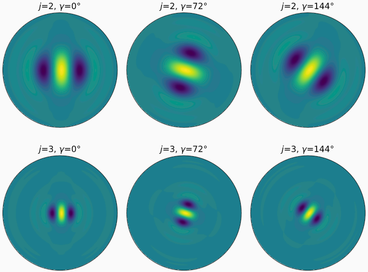
Scaling function $\phi \in \mathrm{L}^2(\mathbb{S}^2)$ captures spatially-localised, low-frequency content.

Spherical wavelet transform given by

$$W_j(\rho) = (f \star \psi_j)(\rho) = \int_{\mathbb{S}^2} \mathrm{d}\mu(\omega') f(\omega') (R_\rho \psi_j)^*(\omega').$$

Spherical convolution

Rotated wavelet



Orthographic plot of spherical wavelets.

Fast algorithms available

(e.g. McEwen et al. 2007, 2013, 2015).

Wavelet Localisation

(McEwen *et al.* 2016)

Directional scale-discretised wavelets $\psi_j \in \mathrm{L}^2(\mathbb{S}^2)$, defined on the sphere $\mathbb{S}^2$ and centred on the North pole, satisfy the localisation bound:

$$\left|\psi_j(\theta, \varphi)\right| \leq \frac{C_1^{(j)}}{\left(1 + C_2^{(j)}\,\theta\right)^{\xi}}$$

(there exist strictly positive constants $C_1^{(j)}, C_2^{(j)} \in \mathbb{R}_*^+$ for any $\xi \in \mathbb{R}_*^+$).

Wavelet Asymptotic Uncorrelation

(McEwen *et al.* 2016)

For Gaussian random fields on the sphere, directional scale-discretised wavelet coefficients are asymptotically uncorrelated. The directional wavelet correlation satisfies the bound:

$$\mathrm{corr}_{jj'}(\rho_1, \rho_2) \leq \frac{C_1^{(j)}}{\left(1 + C_2^{(xj)}\beta\right)^{\xi}} ,$$

where $\beta \in [0, \pi)$ is an angular separation between Euler angles ρ_1 and ρ_2 (there exist strictly positive constants $C_1^{(j)}, C_2^{(j)} \in \mathbb{R}_*^+$ for any $\xi \in \mathbb{R}_*^+$, $\xi \geq 2M$, where M is the azimuthal band-limit of the wavelet and $|j - j'| < 2$).

Spherical scattering propagator for scale j:

$$U[j]f = |f \star \psi_j|.$$

Modulus function is adopted for the activation function (since non-expansive and preserves stability of wavelet representation).

Scattering transform on the sphere

Spherical scattering propagator for scale j:

$$U[j]f = |f \star \psi_j|.$$

Modulus function is adopted for the activation function (since non-expansive and preserves stability of wavelet representation).

Spherical cascade of propagators:

$$U[p]f = |||f \star \psi_{j_1}| \star \psi_{j_2}| \dots \star \psi_{j_d}|,$$

for the path $p = (j_1, j_2, \dots, j_d)$ with depth d.

Scattering transform on the sphere

Spherical scattering propagator for scale j:

$$U[j]f = |f \star \psi_j|.$$

Modulus function is adopted for the activation function (since non-expansive and preserves stability of wavelet representation).

Spherical cascade of propagators:

$$U[p]f = |||f \star \psi_{j_1}| \star \psi_{j_2}| \dots \star \psi_{j_d}|,$$

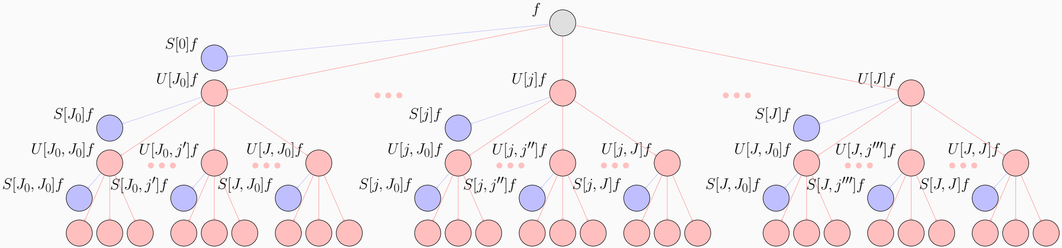
for the path $p = (j_1, j_2, \dots, j_d)$ with depth d.

Scattering coefficients:

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

Scattering networks on the sphere

Spherical scattering network is collection of scattering transforms for a number of paths: $\mathcal{S}_{\mathbb{P}} f = \{S[p]f : p \in \mathbb{P}\}$, where the general path set $\mathbb{P}$ denotes the infinite set of all possible paths $\mathbb{P} = \{p = (j_1, j_2, \ldots, j_d) : J_0 \leq j_i \leq J, 1 \leq i \leq d,\ d \in \mathbb{N}_0\}$.



Capture all information content at infinite depth and typically $> 99\%$ for depth $d = 3$.

SciAI UCL
Jason McEwen

Properties

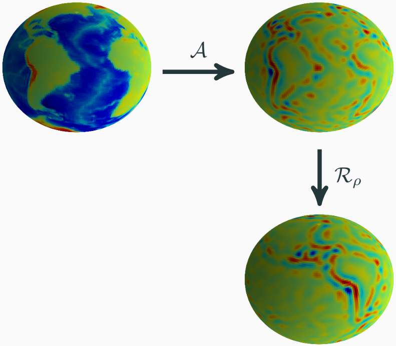
Latent representation is very well-behaved and satisfies a number of important properties:

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

Rotationally equivariance

Rotational Equivariance (McEwen *et al.* 2022)

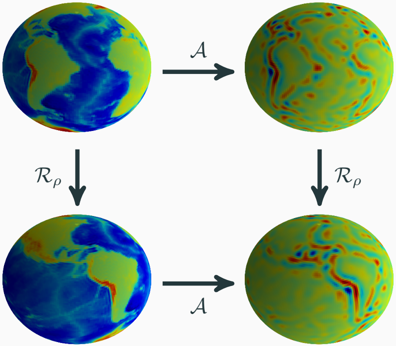
$$((\mathcal{R}_\rho f) \star \psi)(\rho') = (\mathcal{R}_\rho (f \star \psi))(\rho').$$



Rotationally equivariance

Rotational Equivariance (McEwen *et al.* 2022)

$$((\mathcal{R}_\rho f) \star \psi)(\rho') = (\mathcal{R}_\rho(f \star \psi))(\rho').$$



Isometric invariance

Isometric Invariance (McEwen *et al.* 2022)

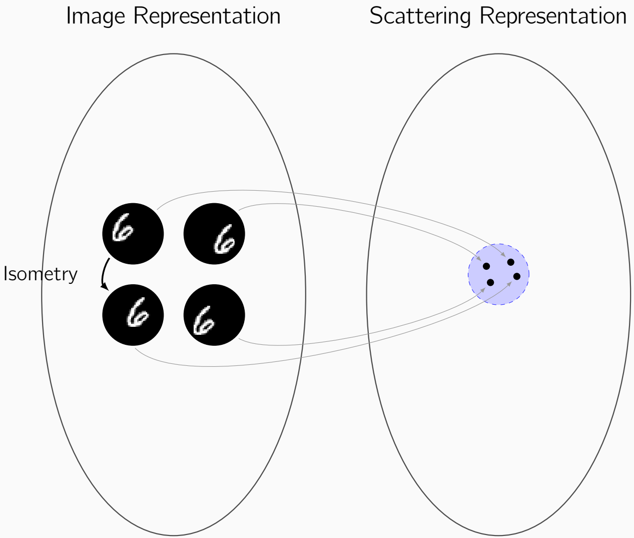
Let $\zeta \in \mathrm{Isom}(\mathbb{S}^2)$, then there exists a constant C such that for all $f \in \mathrm{L}^2(\mathbb{S}^2)$,

$$\|\mathcal{S}_{\mathbb{P}_D} f - \mathcal{S}_{\mathbb{P}_D} V_\zeta f\|_2 \leq C L^{5/2} (D+1)^{1/2} \lambda^{J_0} \|\zeta\|_\infty \|f\|_2 .$$

Difference in representation.

Scattering network representation is invariant to isometries up to a scale.

Isometric invariance



Stability to Diffeomorphisms (McEwen *et al.* 2022)

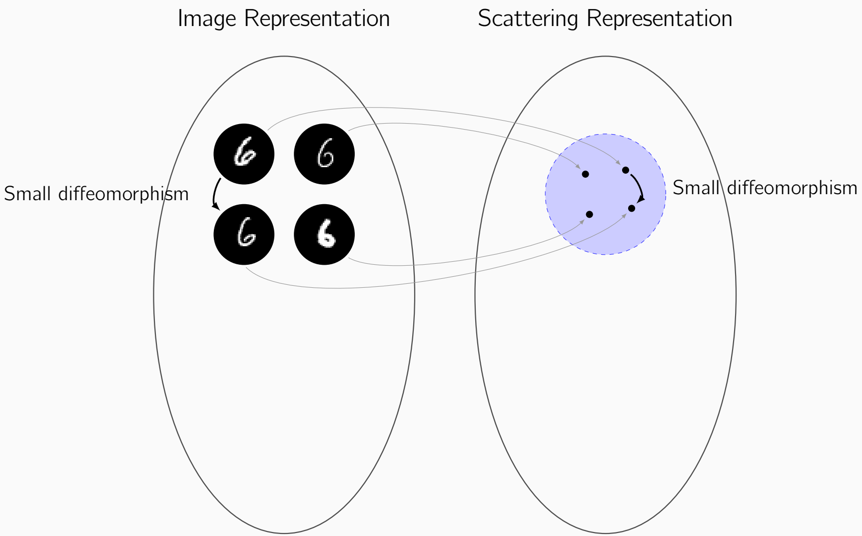
Let $\zeta \in \mathrm{Diff}(\mathbb{S}^2)$. If $\zeta = \zeta_1 \circ \zeta_2$ for some isometry $\zeta_1 \in \mathrm{Isom}(\mathbb{S}^2)$ and diffeomorphism $\zeta_2 \in \mathrm{Diff}(\mathbb{S}^2)$, then there exists a constant C such that for all $f \in \mathrm{L}^2(\mathbb{S}^2)$,

$$\|\mathcal{S}_{\mathbb{P}_D} f - \mathcal{S}_{\mathbb{P}_D} V_\zeta f\|_2 \leq C L^2 \left[L^2 \, \|\zeta_2\|_\infty + L^{1/2} (D+1)^{1/2} \lambda^{J_0} \, \|\zeta_1\|_\infty \right] \|f\|_2 .$$

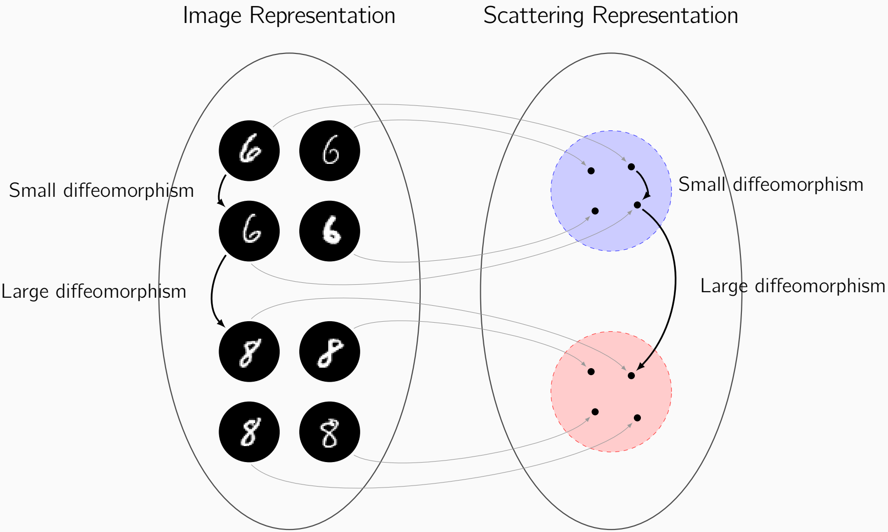
Difference in representation.

Scattering network representation is stable to small diffeomorphisms about isometry.

Stability to diffeomorphisms

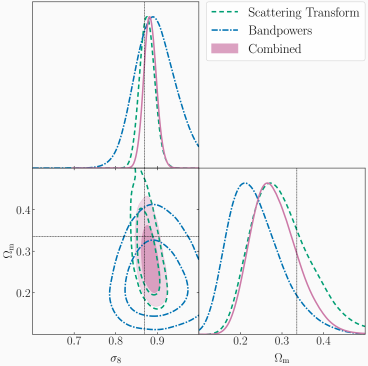


Stability to diffeomorphisms

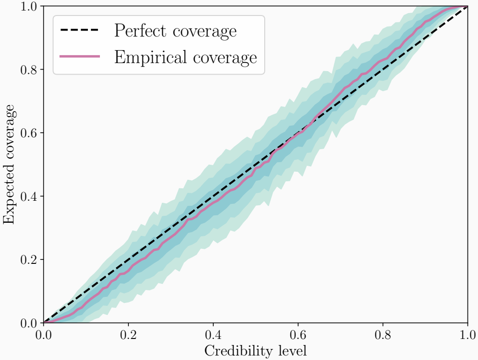


Scattering for simulation-based inference (SBI)

Wavelet scattering as a representation space for SBI (Lin, Joachimi & McEwen 2024).



Posterior



Coverage

SciAI UCL

Generative models of astrophysical fields with scattering transforms on the sphere

(Mousset *et al.* McEwen 2024; **`s2scat`** code)

Louise Mousset

Erwan Allys

Matt Price

Spherical scattering covariance for generative modelling

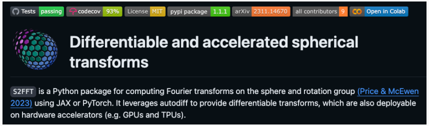
Scattering covariance statistics:

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. $S_3[\lambda_1, \lambda_2] f = \text{Cov}\big[\, f \star \psi_{\lambda_2}, |f \star \psi_{\lambda_1}| \star \psi_{\lambda_2} \,\big]$.
4. $S_4[\lambda_1, \lambda_2, \lambda_3] f = \text{Cov}\big[\, |f \star \psi_{\lambda_1}| \star \psi_{\lambda_3}, |f \star \psi_{\lambda_2}| \star \psi_{\lambda_3} \,\big]$.

Generative modelling by **matching set of scattering covariance statistics** $\mathcal{S}(f)$ with a (single) target simulation:

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

Differentiable and GPU-accelerated spherical transform codes (in JAX)



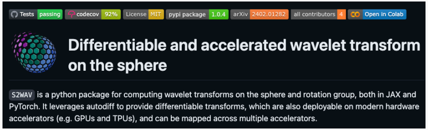
Tests passing | codecov 93% | License MIT | pypi package 1.1.1 | arXiv 2311.14670 | all contributors 9 | Open in Colab

Differentiable and accelerated spherical transforms

`S2FFT` is a Python package for computing Fourier transforms on the sphere and rotation group (Price & McEwen 2023) using JAX or PyTorch. It leverages autodiff to provide differentiable transforms, which are also deployable on hardware accelerators (e.g. GPUs and TPUs).

`s2fft`: Spherical harmonic transforms

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



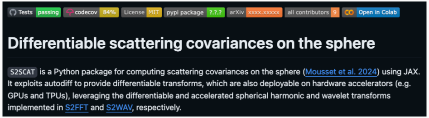
Tests passing | codecov 92% | License MIT | pypi package 1.0.4 | arXiv 2402.01282 | all contributors 4 | Open in Colab

Differentiable and accelerated wavelet transform on the sphere

`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.

`s2wav`: Spherical wavelet transforms

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



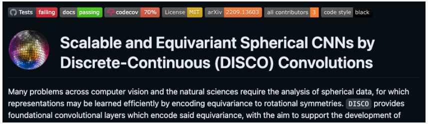
Tests passing | codecov 84% | License MIT | pypi package ?.?.? | arXiv xxxx.xxxxx | all contributors 9 | Open in Colab

Differentiable scattering covariances on the sphere

`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.

`s2scat`: Spherical scattering transforms

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



Tests failing | docs passing | codecov 70% | License MIT | arXiv 2209.13603 | all contributors 3 | code style black

Scalable and Equivariant Spherical CNNs by Discrete-Continuous (DISCO) Convolutions

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. `DISCO` provides foundational convolutional layers which encode said equivariance, with the aim to support the development of

`s2ai`: Spherical AI

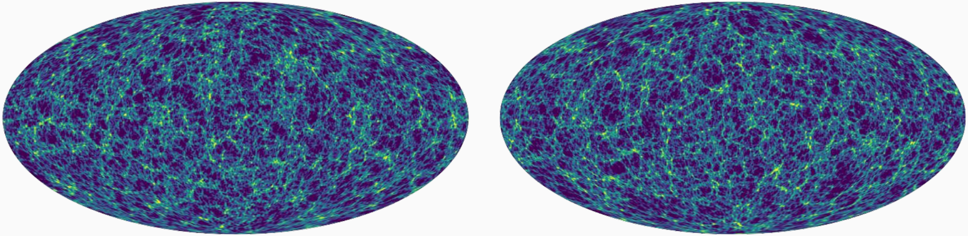
Coming very soon! Contact us for early access.

SciAI UCL

Jason McEwen

Generative modelling of large scale structure (LSS)

Which field is emulated and which simulated?

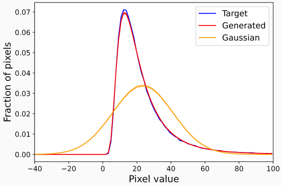


Logarithm (for visualization) of weak lensing field.

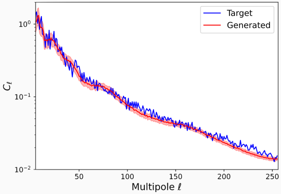
SciAI UCL

Jason McEwen

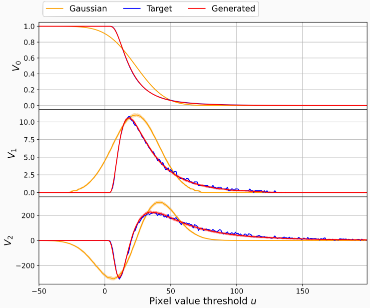
Generative modelling of large scale structure (LSS)



Pixel distribution



Power spectrum



Minkowski functionals

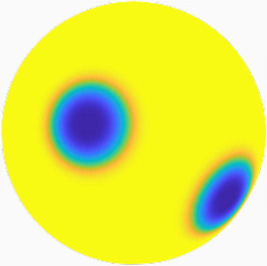
Statistical validation.

SciAI UCL

Jason McEwen

Equivariant learning for spherical fields

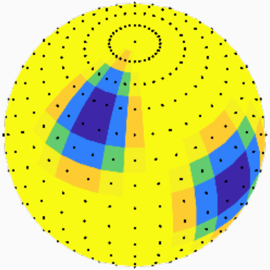
Continuous



✔ Equivariant
✖ Not Scalable

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

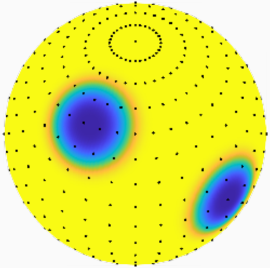
Discrete



✖ Not Equivariant
✔ Scalable

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

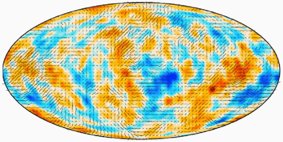
Discrete-Continuous (DISCO)



✔ Equivariant
✔ Scalable

(Ocampo, Price & McEwen 2023)

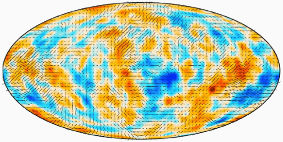
Wind ⇝ vector (spin-1) field



CMB polarization ⇝ spin-2 field

Equivariant learning for spherical fields of different spin

Wind ⇝ vector (spin-1) field

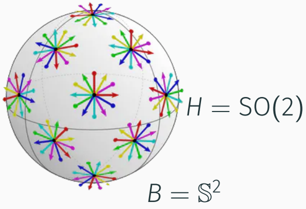


CMB polarization ⇝ spin-2 field

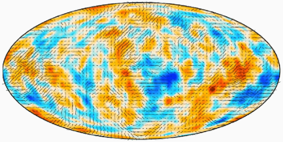
Equivariant learning for spherical fields of different spin

Fibre bundle representation:

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



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

Spin equivariant approach:

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

Scalable Bayesian inference with data-driven AI priors

Exascale imaging

Artist impression of the Square Kilometer Array (SKA)

Sampling vs optimisation

MCMC sampling

- ✖ Based on sampling so **computationally demanding**.
- ✔ Uncertatinties encoded in posterior.
- ✖ **Hand-crafted priors** (traditionally).

MAP estimation

- ✔ Based on optimisation so computationally efficient.
- ✖ **No uncertainties** (traditionally).
- ✖ **Hand-crafted priors** (traditionally).

Computational imaging strategy

Goals:

- Computationally efficient (optimisation + distribution).
- Quantifies uncertainties (for scientific inference).
- Data-driven AI priors (enhance reconstruction fidelity).

Computational imaging strategy

Goals:

- Computationally efficient (optimisation + distribution).
- Quantifies uncertainties (for scientific inference).
- Data-driven AI priors (enhance reconstruction fidelity).

Solution:

1. Statistical framework: Bayesian inference and MAP estimation.
2. Mathematical theory: probability concentration theorem for log-convex distributions.
3. Constrained AI model: convex AI model with explicit potential.

Scalable Bayesian uncertainty quantification with data-driven AI priors

Scalable Bayesian uncertainty quantification with data-driven priors for radio interferometric imaging
(Liaudat *et al.* McEwen 2024)

Tobias Liaudat

Matthijs Mars

Matt Price

Marcelo Pereyra

Marta Betcke

Solve optimisation problem

Solve optimisation problem (MAP estimation by variation regularisation):

$$\hat{x}_{\text{MAP}} = \arg\max_{x} \left[\log p(\boldsymbol{y} \,|\, \boldsymbol{x}) \right] = \arg\min_{x} \left[\, \|\boldsymbol{y} - \boldsymbol{\Phi x}\|_2^2 + \underbrace{\lambda R(\boldsymbol{x})}_{\text{regulariser}} \, \right]$$

SciAI UCL

Solve optimisation problem

Solve optimisation problem (MAP estimation by variation regularisation):

$$\hat{x}_{\text{MAP}} = \arg\max_{x} \left[\log p(\boldsymbol{y} \,|\, \boldsymbol{x}) \right] = \arg\min_{x} \left[\|\boldsymbol{y} - \boldsymbol{\Phi x}\|_2^2 + \lambda R(\boldsymbol{x}) \right]$$

regulariser

Traditionally, **hand-crafted regularisers** used
(*e.g.* $R(\boldsymbol{x}) = \|\boldsymbol{\Psi}^\dagger \boldsymbol{x}\|_1$ to promote sparsity in some (wavelet) dictionary $\boldsymbol{\Psi}$).

Instead, adopt data-driven AI prior for regulariser trained on simulations.

Solve optimisation problem

Solve optimisation problem (MAP estimation by variation regularisation):

$$\hat{x}_{\text{MAP}} = \arg\max_{x} \left[\log p(y \,|\, x) \right] = \arg\min_{x} \left[\|y - \mathbf{\Phi} x\|_2^2 + \lambda R(x) \right]$$

regulariser

Traditionally, **hand-crafted regularisers** used
(*e.g.* $R(x) = \|\mathbf{\Psi}^{\dagger} x\|_1$ to promote sparsity in some (wavelet) dictionary $\mathbf{\Psi}$).

Instead, adopt data-driven AI prior for regulariser trained on simulations.

Solve by highly distributed and parallelised optimisation algorithms, with **low communication** overhead (Pratley, McEwen *et al.* 2016, Pratley, Johnston-Hollitt & McEwen 2018, 2019, Pratley & McEwen 2019).

Block distribution

Solve resulting convex optimisation problem by **proximal splitting**.

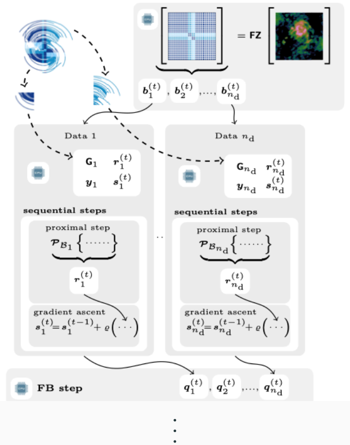
Block algorithm to distribute data and compute (telescope model):

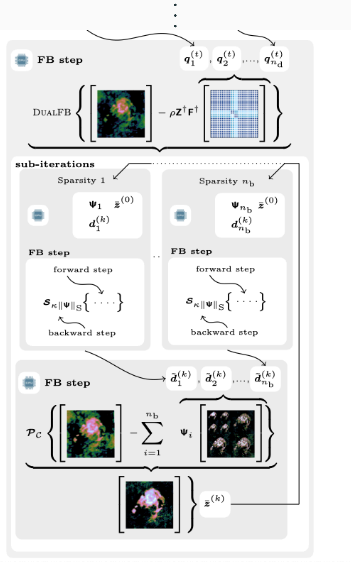
(Carrillo, McEwen & Wiaux 2014; Onose *et al.* (inc. McEwen) 2016; Pratley, Johnston-Hollitt & McEwen 2019; Pratley, McEwen *et al.* 2019; Pratley, Johnston-Hollitt & McEwen 2020)

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{n_\mathrm{d}} \end{bmatrix}, \quad \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_{n_\mathrm{d}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_1\boldsymbol{M}_1 \\ \vdots \\ \boldsymbol{G}_{n_\mathrm{d}}\boldsymbol{M}_{n_\mathrm{d}} \end{bmatrix} \boldsymbol{F}\boldsymbol{Z} .$$

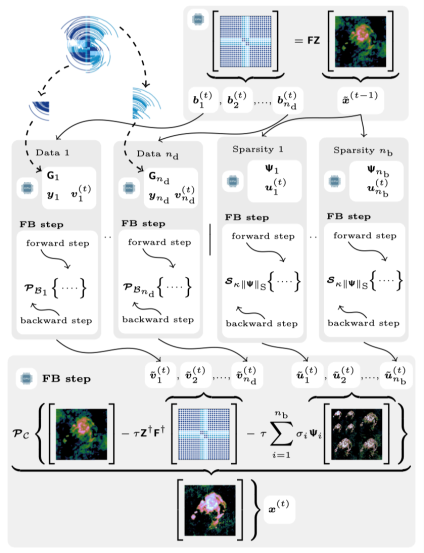
- Stochastic updates to support big-data.
- Two internal distribution strategies:
  1. Distribute image (*i.e.* distribute $\boldsymbol{\Phi}_i$)
  2. Distribute Fourier grid (*i.e.* distribute $\boldsymbol{G}_i\boldsymbol{M}_i$)

Block distributed alternating direction method of multipliers (ADMM) algorithm

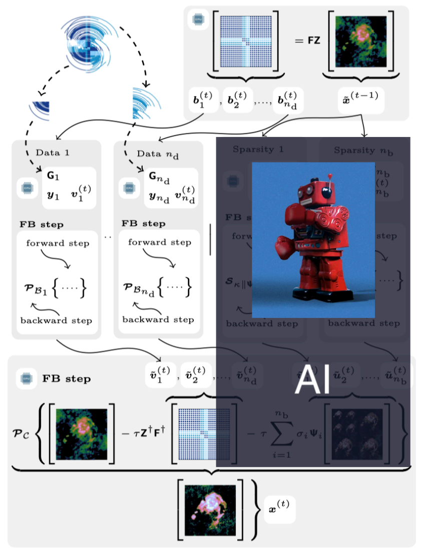




Block distributed primal dual algorithm



Block distributed primal dual algorithm with AI prior



SciAI UCL

Jason McEwen

Convex probability concentration for uncertainty quantification

Posterior credible region:

$$p(x \in C_\alpha | y) = \int_{x \in \mathbb{R}^N} p(x|y) \mathbb{1}_{C_\alpha} \mathrm{d}x = 1 - \alpha.$$

Consider the **highest posterior density (HPD) region**

$$C^*_\alpha = \left\{ x : -\log p(x) \leq \gamma_\alpha \right\}, \quad \text{with } \gamma_\alpha \in \mathbb{R}, \quad \text{and } p(x \in C^*_\alpha | y) = 1 - \alpha \text{ holds.}$$

Convex probability concentration for uncertainty quantification

Posterior credible region:

$$p(x \in C_\alpha | y) = \int_{x \in \mathbb{R}^N} p(x|y) \mathbb{1}_{C_\alpha} \mathrm{d}x = 1 - \alpha.$$

Consider the **highest posterior density (HPD) region**

$$C^*_\alpha = \{ x : -\log p(x) \leq \gamma_\alpha \}, \quad \text{with } \gamma_\alpha \in \mathbb{R}, \quad \text{and } p(x \in C^*_\alpha | y) = 1 - \alpha \text{ holds.}$$

Bound of HPD region for log-concave distributions (Pereyra 2017)

Suppose the posterior $\log p(x|y) \propto \log \mathcal{L}(x) + \log \pi(x)$ is log-concave on $\mathbb{R}^N$. Then, for any $\alpha \in (4\mathrm{e}^[(-N/3)], 1)$, the HPD region C^*_α is contained by

$$\hat{C}_\alpha = \left\{ x : \log \mathcal{L}(x) + \log \pi(x) \leq \hat{\gamma}_\alpha = \log \mathcal{L}(\hat{x}_{\text{MAP}}) + \log \pi(\hat{x}_{\text{MAP}}) + \sqrt{N}\tau_\alpha + N \right\},$$

with a positive constant $\tau_\alpha = \sqrt{16 \log(3/\alpha)}$ independent of $p(x|y)$.

Need **only evaluate** $\log \mathcal{L} + \log \pi$ **for the MAP estimate** $\hat{x}_{\text{MAP}}$!

Hypothesis testing of physical structure

(Pereyra 2017; Cai, Pereyra & McEwen 2018a)

1. Remove structure of interest from recovered image $x^\star$.
2. Inpaint background (noise) into region, yielding surrogate image x'.
3. Test whether $x' \in C_\alpha$:
   - If $x' \notin C_\alpha$ then reject hypothesis that structure is an artifact with confidence $(1-\alpha)\%$, *i.e.* **structure most likely physical**.
   - If $x' \in C_\alpha$ uncertainly too high to draw strong conclusions about the physical nature of the structure.

Local Bayesian credible intervals for sparse reconstruction

(Cai, Pereyra & McEwen 2018b)

Let Ω define the area (or pixel) over which to compute the credible interval $(\tilde{\xi}_{-}, \tilde{\xi}_{+})$ and ζ be an index vector describing Ω (*i.e.* $\zeta_i = 1$ if $i \in \Omega$ and 0 otherwise).

Consider the test image with the Ω region replaced by constant value ξ:

$$x' = x^{\star}(\mathcal{I} - \zeta) + \xi\zeta \,.$$

Given $\tilde{\gamma}_{\alpha}$ and $x^{\star}$, compute the credible interval by

$$\tilde{\xi}_{-} = \min_{\xi} \left\{ \xi \mid \log \mathcal{L}(x') + \log \pi(x') \leq \tilde{\gamma}_{\alpha}, \ \forall \xi \in [-\infty, +\infty) \right\} ,$$

$$\tilde{\xi}_{+} = \max_{\xi} \left\{ \xi \mid \log \mathcal{L}(x') + \log \pi(x') \leq \tilde{\gamma}_{\alpha}, \ \forall \xi \in [-\infty, +\infty) \right\} .$$

Convex data-driven AI prior

Adopt **neural-network-based convex regulariser** R (Goujon *et al.* 2022; Liaudat *et al.* McEwen 2024):

$$R(\boldsymbol{x}) = \sum_{n=1}^{N_C} \sum_{k} \psi_n \left((\boldsymbol{h}_n * \boldsymbol{x}) [k] \right) ,$$

- ψ_n are learned convex profile functions with Lipschitz continuous derivative;
- N_C learned convolutional filters $\boldsymbol{h}_n$.

Convex data-driven AI prior

Adopt **neural-network-based convex regulariser** R
(Goujon *et al.* 2022; Liaudat *et al.* McEwen 2024):

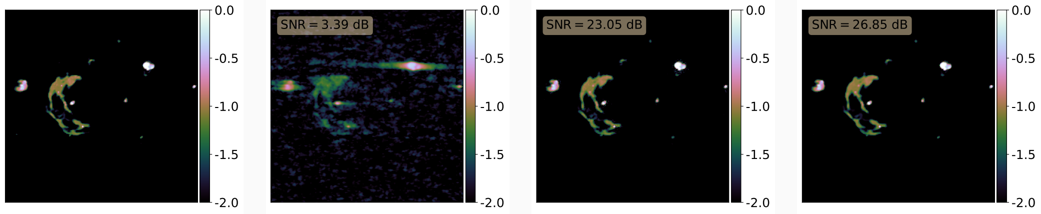
$$R(x) = \sum_{n=1}^{N_C} \sum_{k} \psi_n \left((\boldsymbol{h}_n * x) [k] \right),$$

- ψ_n are learned convex profile functions with Lipschitz continuous derivative;
- N_C learned convolutional filters $\boldsymbol{h}_n$.

Properties:

1. Convex + explicit potential $\Rightarrow$ leverage convex UQ theory.
2. Smooth regulariser with known Lipschitz constant $\Rightarrow$ convergence guarantees.

Reconstructed images



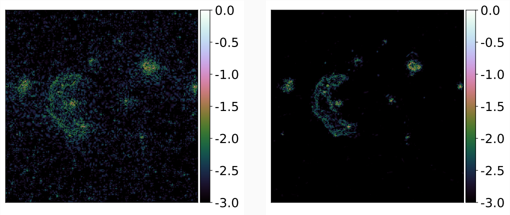
Ground truth

Dirty image
SNR=3.39 dB

Reconstruction (classical)
SNR=23.05 dB

Reconstruction (learned)
SNR= 26.85 dB

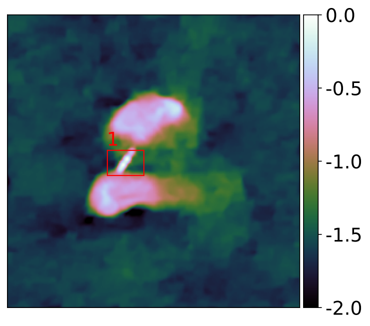
(Liaudat *et al.* McEwen 2024)



Error (classical)

Error (learned)

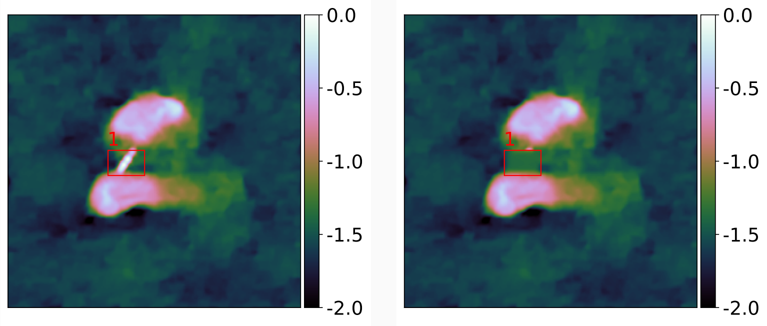
Hypothesis testing of structure



Reconstructed image

(Liaudat *et al.* McEwen 2024)

Hypothesis testing of structure



Reconstructed image

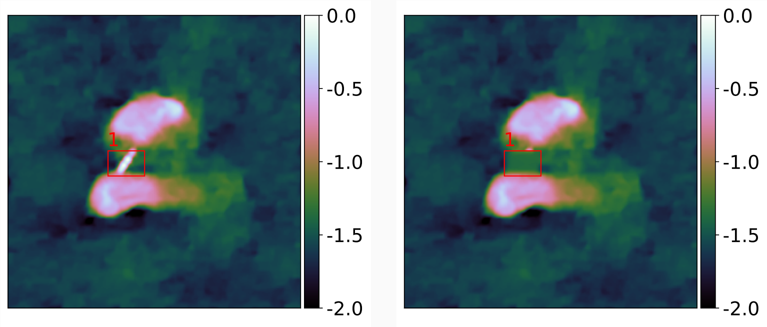
Surrogate test image (region removed)

(Liaudat *et al.* McEwen 2024)

SciAI UCL

Jason McEwen

Hypothesis testing of structure



Reconstructed image

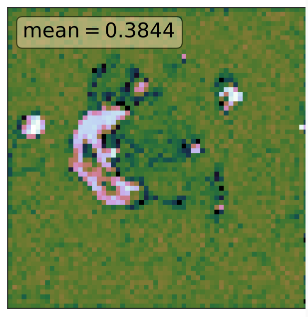
Surrogate test image (region removed)

Reject null hypothesis

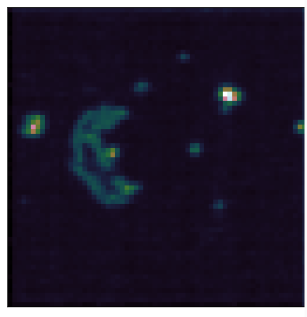
$\Rightarrow$ **structure physical**

(Liaudat *et al.* McEwen 2024)

Approximate local Bayesian credible intervals



LCI
(super-pixel size 4×4)



MCMC standard deviation
(super-pixel size 4×4)

(Liaudat *et al.* McEwen 2024)

QuantifAI code

GitHub quantifai | License GPL | arXiv 2312.00125

QuantifAI

`quantifai` is a PyTorch-based open-source radio interferometric imaging reconstruction package with scalable Bayesian uncertainty quantification relying on data-driven (learned) priors. This package was used to produce the results of Liaudat et al. 2023. The `quantifai` model relies on the data-driven convex regulariser from Goujon et al. 2022.

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

PyTorch: Automatic differentiation (including instrument model) + GPU acceleration

Exascale imaging codes

PURIFY

CI passing | codecov 86% | DOI 10.5281/zenodo.2555252

Description

PURIFY is an open-source collection of routines written in `C++` available under the license below. It implements different tools and high-level to perform radio interferometric imaging, *i.e.* to recover images from the Fourier measurements taken by radio interferometric telescopes.

GitHub: https://github.com/astro-informatics/purify

Sparse OPTimisation Library

CMake passing | codecov 96% | DOI 10.5281/zenodo.2584256

Description

SOPT is an open-source `C++` package available under the license below. It performs Sparse OPTimisation using state-of-the-art convex optimisation algorithms. It solves a variety of sparse regularisation problems, including the Sparsity Averaging Reweighted Analysis (SARA) algorithm.

GitHub: https://github.com/astro-informatics/sopt

Summary

Statistics as the Key to Unlocking AI for Science

Statistical Scientific AI

Statistical
- Bayesian Inference
- Implicit Inference
- Generative Models

Physical
- Augmentation
- Physical Properties
- Physical Models

Intelligible
- Explainable
- Interpretable
- Reliable