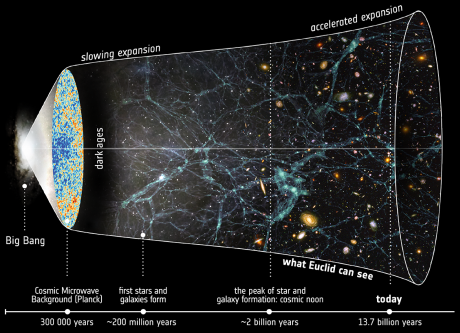
Scientific AI for the Physical Sciences

Jason D. McEwen
www.jasonmcewen.org

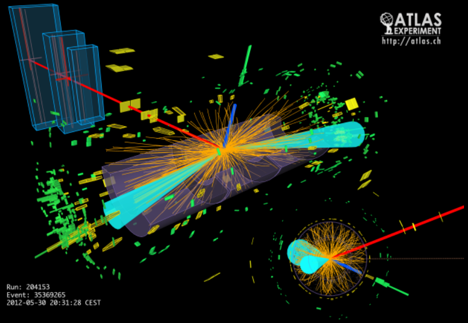
SciAI Group, Mullard Space Science Laboratory (MSSL)
Centre for Data Intensive Science & Industry (DISI)
University College London (UCL)

ICML @ London 2024

Towards a fundamental understanding of our Universe

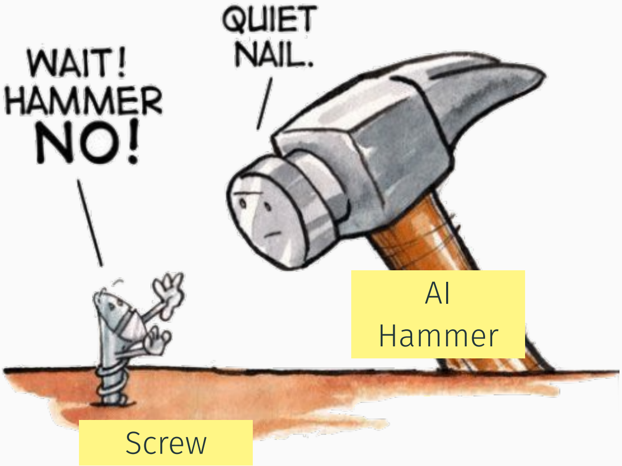


Astrophysics & Cosmology

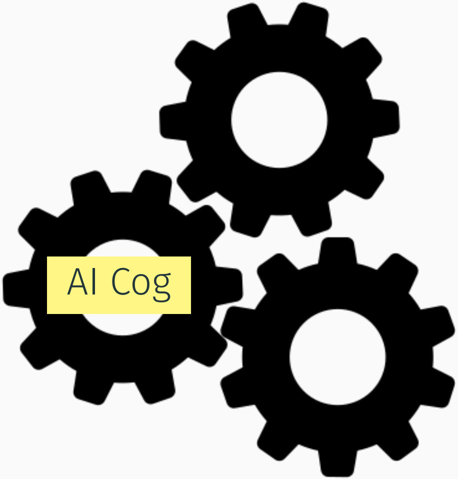


High Energy Physics

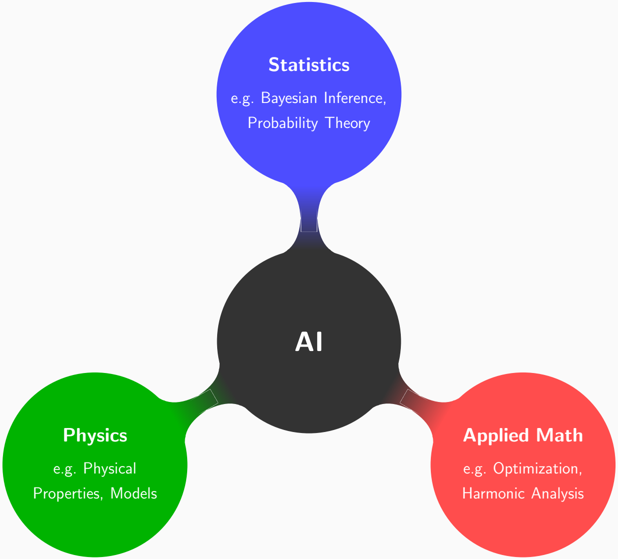
The AI hammer



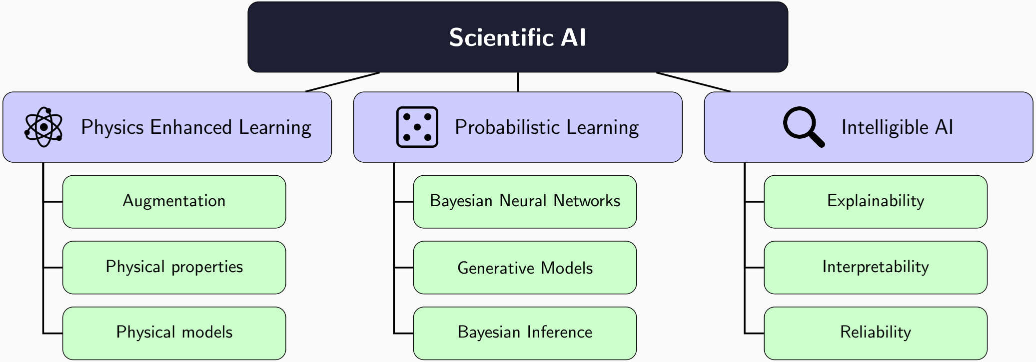
The AI cog



Merging paradigms



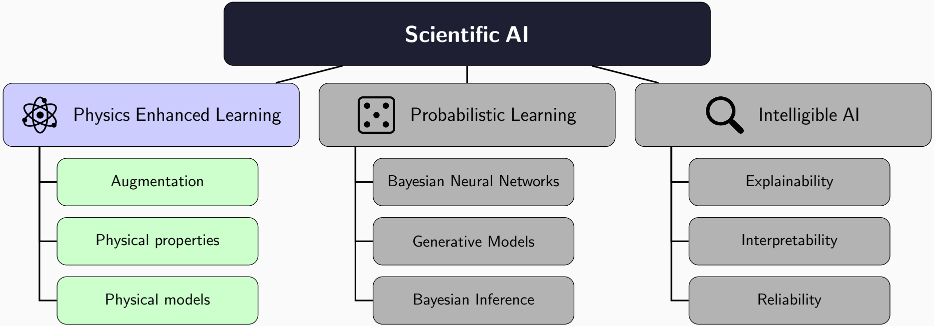
Scientific AI for the physical sciences



Physics Enhanced Learning

Embed physical understanding of the world into machine learning models.

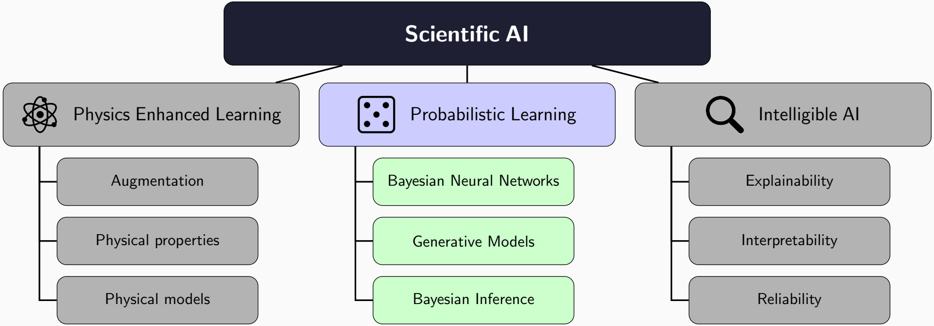
(See review by Karniadakis *et al.* 2021.)



Probabilistic Learning

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

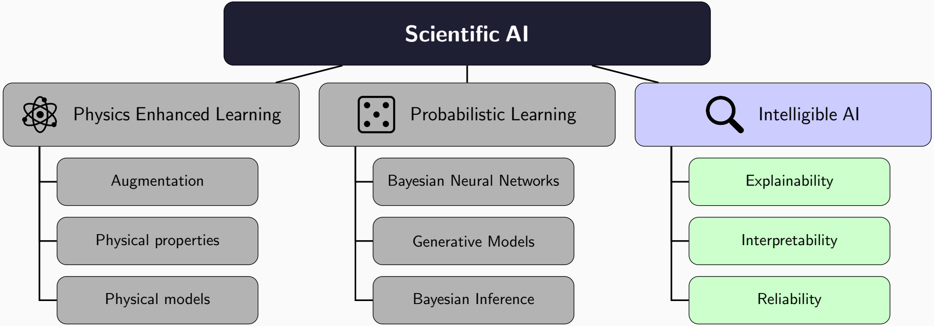
(See Murray 2022.)



Intelligible AI

Machine learning methods that are able to be understood by humans.

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



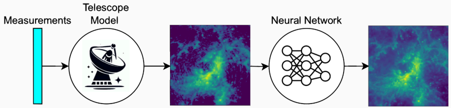
Some (very!) brief case studies

1. Differentiable Physics
2. Geometric & Equivariant Deep Learning
3. Generative Models for Textures
4. Accelerated Bayesian Inference
5. Denoising Diffusion MCMC for Imaging

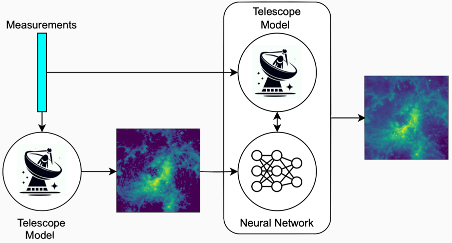
Differentiable Physics

Differentiable physics

- Differentiable physical models
  - Radio interferometric telescope (Mars *et al.* 2023, 2024)
    - ⇝ Reconstruction quality ↑ (∼20dB)
    - ⇝ Computation time ↓ (∼600×)
  - Weak gravitational lensing (Whitney *et al.* in prep.)
  - JAX-Cosmo, CosmoPower-JAX (Campagne *et al.* 2023, Spurio Mancini *et al.* 2021, Piras *et al.* 2023)



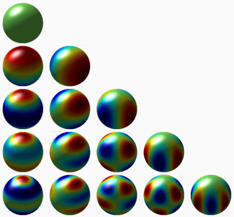
Classical AI model



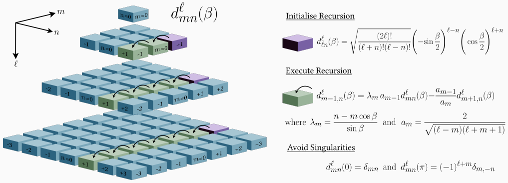
Hybrid physics-enhanced AI model

Differentiable physics allows hybrid physics-enhanced AI models.

- ▷ Differentiable mathematical methods
  - ▶ Spherical harmonic transforms
    (Price & McEwen 2024; **s2fft** code)
    ⇝ Computation time ↓ (∼400×)
  - ▶ Spherical wavelet transforms
    (Price *et al.* 2024; **s2wav** code)
    ⇝ Computation time ↓ (∼300×)



Spherical harmonics



Differentiable and GPU-friendly recursions

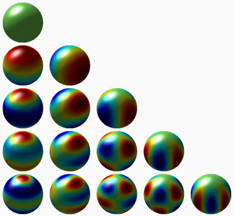
- Differentiable mathematical methods
  - Spherical harmonic transforms (Price & McEwen 2024; **`s2fft`** code) ⇝ Computation time ↓ ($\sim$400$\times$)
  - Spherical wavelet transforms (Price *et al.* 2024; **`s2wav`** code) ⇝ Computation time ↓ ($\sim$300$\times$)

See poster

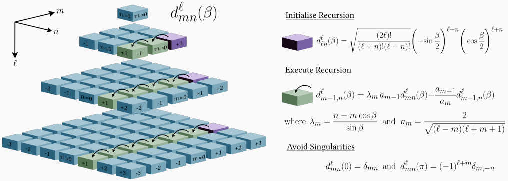
Matt Price

Alicja Polanska

Jess Whitney



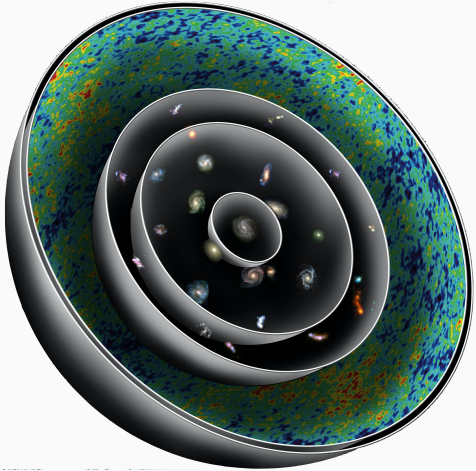
Spherical harmonics



Differentiable and GPU-friendly recursions

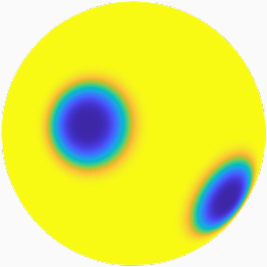
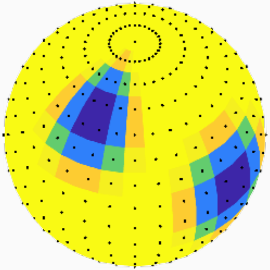
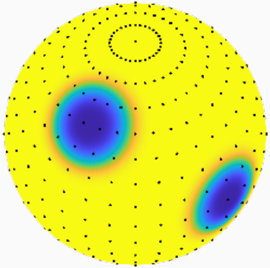
Geometric & Equivariant Deep Learning

Geometric & equivariant deep learning

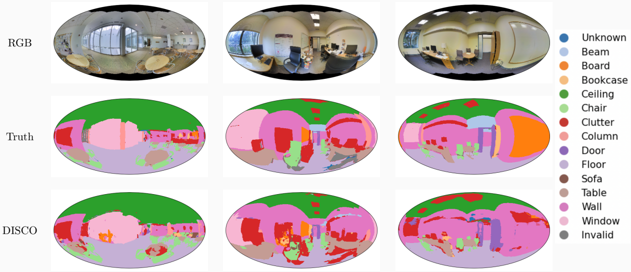


Cosmological observations made on the celestial sphere.

Categorization of spherical CNN frameworks

Continuous	Discrete	Discrete-Continuous (DISCO)
		
✔ Equivariant ✖ Not Scalable	✖ Not Equivariant ✔ Scalable	✔ Equivariant ✔ Scalable
(Cohen et al. 2018, Esteves et al. 2018, Kondor et al. 2018, Cobb et al. 2021, McEwen et al. 2022, ...)	(Jiang et al. 2019, Zhang et al. 2019, Perraudin et al. 2019, Cohen et al. 2019, ...)	(Ocampo, Price & McEwen 2023; `s2ai` code)

- 10^9 saving in compute and 10^4 saving in memory (for 4k spherical image).
- SOTA performance on variety of benchmark problems (classification, depth estimation, semantic segmentation).

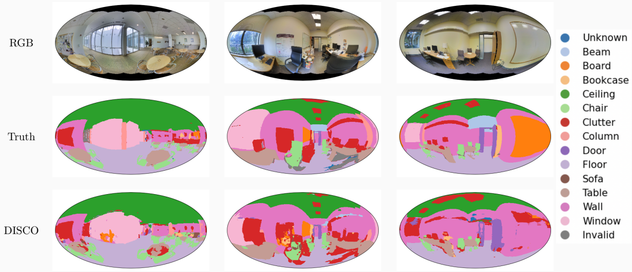


Semantic segmentation for 2D3DS data-set.

Geometric & equivariant deep learning

- 10^9 saving in compute and
  10^4 saving in memory
  (for 4k spherical image).
- SOTA performance on variety of benchmark problems
  (classification, depth estimation, semantic segmentation).

See poster



Semantic segmentation for 2D3DS data-set.

Kevin Mulder

Matt Price

Generative Models for Textures

Why not use standard AI generative models?

Standard machine learning techniques:

- **require substantial training data** (which we often do not have);
- **suffers covariate shift** (*i.e.* change in physical model);
- **fails to capture symmetries** of data (unless encode in model architecture).

Why not use standard AI generative models?

Standard machine learning techniques:

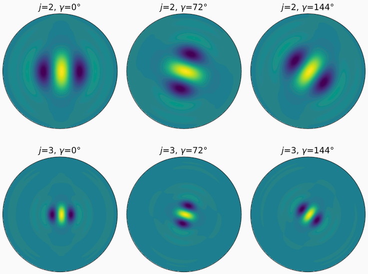
- ▷ **require substantial training data** (which we often do not have);
- ▷ **suffers covariate shift** (*i.e.* change in physical model);
- ▷ **fails to capture symmetries** of data (unless encode in model architecture).

⇒ Statistical characterization and generative modelling.

- ▷ **Wavelet scattering networks** inspired by CNNs but designed rather than learned filters (Mallat 2012).
- ▷ Extend to **spherical scattering networks** (McEwen et al. 2022).

Spherical scattering propagator for scale j:

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



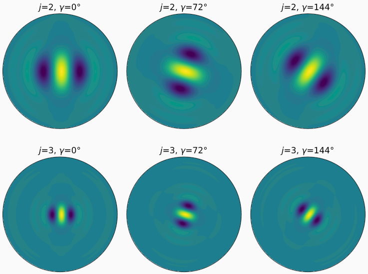
Orthographic plot of spherical wavelets.

Spherical scattering propagator for scale j:

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

Spherical cascade of propagators:

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



Orthographic plot of spherical wavelets.

Scattering transform on the sphere

Spherical scattering propagator for scale j:

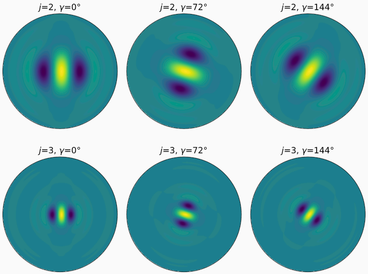
$$U[j]f = |f \star \psi_j|.$$

Spherical cascade of propagators:

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

Scattering coefficients:

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



Orthographic plot of spherical wavelets.

Isometric Invariance

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.

Stability to Diffeomorphisms

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.

Generative models of astrophysical fields with scattering transforms on the sphere

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

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 models of astrophysical fields with scattering transforms on the sphere

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

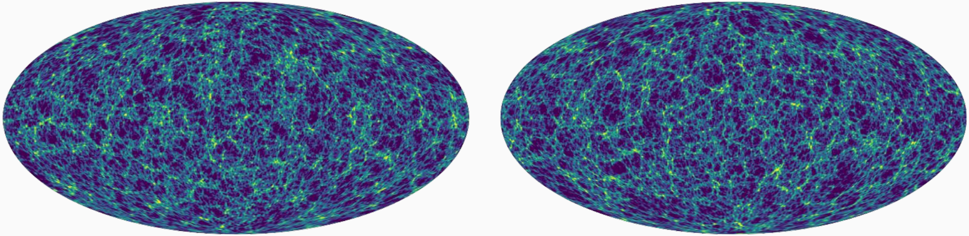
Scattering covariance statistics:

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** $\mathcal{S}(f)$ with a (single) target simulation:

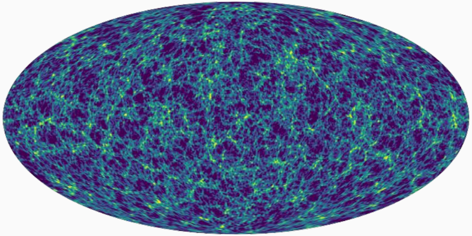
$$\min_f \|\mathcal{S}(f) - \mathcal{S}(f_{\mathrm{target}})\|^2 .$$

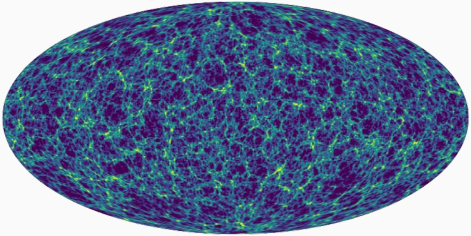
Which field is emulated and which simulated?



Logarithm (for visualization) of cosmological weak lensing field.

Which field is emulated and which simulated?





Logarithm (for visualization) of cosmological weak lensing field.

See poster

Matt Price

Accelerated Bayesian Inference

Bayes' theorem

$$\underbrace{p(\theta \mid \mathbf{y}, M)}_{\text{posterior}} = \frac{\overbrace{p(\mathbf{y} \mid \theta, M)}^{\text{likelihood}} \; \overbrace{p(\theta \mid M)}^{\text{prior}}}{\underbrace{p(\mathbf{y} \mid M)}_{\text{evidence}}}$$

for parameters θ, model M and observed data $\mathbf{y}$.

Bayes' theorem

$$\underbrace{p(\theta \,|\, \mathbf{y}, M)}_{\text{posterior}} = \frac{\overbrace{p(\mathbf{y} \,|\, \theta, M)}^{\text{likelihood}} \; \overbrace{p(\theta \,|\, M)}^{\text{prior}}}{\underbrace{p(\mathbf{y} \,|\, M)}_{\text{evidence}}} = \frac{\overbrace{\mathcal{L}(\theta)}^{\text{likelihood}} \; \overbrace{\pi(\theta)}^{\text{prior}}}{\underbrace{Z}_{\text{evidence}}} ,$$

for parameters θ, model M and observed data $\mathbf{y}$.

Bayes' theorem

$$\underbrace{p(\theta \mid \mathbf{y}, M)}_{\text{posterior}} = \frac{\overbrace{p(\mathbf{y} \mid \theta, M)}^{\text{likelihood}} \; \overbrace{p(\theta \mid M)}^{\text{prior}}}{\underbrace{p(\mathbf{y} \mid M)}_{\text{evidence}}} = \frac{\overbrace{\mathcal{L}(\theta)}^{\text{likelihood}} \; \overbrace{\pi(\theta)}^{\text{prior}}}{\underbrace{Z}_{\text{evidence}}} ,$$

for parameters θ, model M and observed data $\mathbf{y}$.

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

Bayes' theorem

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

for parameters θ, model M and observed data $\boldsymbol{y}$.

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

For **model selection**, must compute **Bayesian evidence** (marginal likelihood):

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

Leverage recent machine learning developments and underlying technology.

Four pillars of a new paradigm (Piras *et al.* 2024):

New paradigm for accelerated Bayesian inference

Leverage recent machine learning developments and underlying technology.

Four pillars of a new paradigm (Piras *et al.* 2024):

1. **Emulation**, *e.g.* CosmoPower-JAX (Spurio Mancini *et al.* 2021, Piras *et al.* 2023).
2. **Differentiable and probabilistic programming**, *e.g.* JAX, NumPyro.
3. **Scalable MCMC** that exploit gradients, *e.g.* NUTS.
4. **Decoupled and scalable Bayesian model selection**, *e.g.* learned harmonic mean that leverages normalizing flows (McEwen *et al.* 2021, Spurio Mancini *et al.* 2022, Polanska *et al.* 2024, Piras *et al.* 2024; `harmonic` code) .

Learned harmonic mean estimator for Bayesian evidence

- Requires **posterior samples only**
  - ⇝ Evidence almost for free
- **Agnostic to sampling** technique
  - ⇝ Leverage efficient samplers
  - ⇝ Simulation-based inference (SBI)
  - ⇝ Variational inference
- Scale to **high-dimensions**
  - ⇝ Normalizing flows

Accelerated Bayesian inference (Piras *et al.* 2024)

37 parameter cosmic shear analysis of LCDM vs $w_0 w_a$CDM

- CAMB + PolyChord ⇝ 8 months on 48 CPU cores
- CosmoPower-JAX + NumPyro/NUTS + **Harmonic**
  ⇝ 2 days on 12 GPUs

157 parameter 3x2pt analysis of LCDM vs $w_0 w_a$CDM

- CAMB + PolyChord ⇝ 12 years on 48 CPUs (projected)
- CosmoPower-JAX + NumPyro/NUTS + **Harmonic**
  ⇝ 8 days on 24 GPUs

- Requires **posterior samples only**
  - ⇝ Evidence almost for free
- **Agnostic to sampling** technique
  - ⇝ Leverage efficient samplers
  - ⇝ Simulation-based inference (SBI)
  - ⇝ Variational inference
- Scale to **high-dimensions**
  - ⇝ Normalizing flows

Accelerated Bayesian inference (Piras *et al.* 2024)

37 parameter cosmic shear analysis of LCDM vs $w_0 w_a$CDM

- CAMB + PolyChord ⇝ 8 months on 48 CPU cores
- CosmoPower-JAX + NumPyro/NUTS + **Harmonic**
  ⇝ 2 days on 12 GPUs

157 parameter 3x2pt analysis of LCDM vs $w_0 w_a$CDM

- CAMB + PolyChord ⇝ 12 years on 48 CPUs (projected)
- CosmoPower-JAX + NumPyro/NUTS + **Harmonic**
  ⇝ 8 days on 24 GPUs

See poster

Alicja Polanska

Matt Price

Denoising Diffusion MCMC for Imaging

Classical high-dimensional imaging problems often consider Gaussian likelihood and sparsity-promoting prior (e.g. in wavelet representation Ψ):

$$p(y \mid x) \propto \exp\left(-\|y - \Phi x\|_2^2 / (2\sigma^2)\right) \qquad p(x) \propto \exp\left(-\|\Psi^\dagger x\|_1\right)$$

Likelihood Prior

Often compute **MAP estimator** (variational regularisation) by convex optimization:

$$\arg\max_x \log p(x \mid y) = \arg\min_x \left[\|y - \Phi x\|_2^2 + \lambda \|\Psi^\dagger x\|_1 \right]$$

Data fidelity Regulariser

$\Rightarrow$ Alternatively, sample posterior to quantify uncertainties (parameter estimation and model selection).

Classical high-dimensional imaging problems often consider Gaussian likelihood and sparsity-promoting prior (e.g. in wavelet representation Ψ):

$$p(y \mid x) \propto \exp\left(-\|y - \mathbf{\Phi} x\|_2^2 / (2\sigma^2)\right) \qquad p(x) \propto \exp\left(-\|\Psi^\dagger x\|_1\right)$$

Likelihood — Prior

Often compute **MAP estimator** (variational regularisation) by convex optimization:

$$\arg\max_x \log p(x \mid y) = \arg\min_x \left[\|y - \mathbf{\Phi} x\|_2^2 + \lambda \|\Psi^\dagger x\|_1 \right]$$

Data fidelity — Regulariser

$\Rightarrow$ Alternatively, sample posterior to quantify uncertainties (parameter estimation and model selection).

Classical high-dimensional imaging problems often consider Gaussian likelihood and sparsity-promoting prior (e.g. in wavelet representation Ψ):

$$p(y \mid x) \propto \exp\left(-\|y - \Phi x\|_2^2/(2\sigma^2)\right) \qquad p(x) \propto \exp\left(-\|\Psi^\dagger x\|_1\right)$$

Likelihood Prior

Often compute **MAP estimator** (variational regularisation) by convex optimization:

$$\arg\max_x \log p(x \mid y) = \arg\min_x \Big[\underbrace{\|y - \Phi x\|_2^2}_{\text{Data fidelity}} + \underbrace{\lambda \|\Psi^\dagger x\|_1}_{\text{Regulariser}} \Big]$$

$\Rightarrow$ Alternatively, sample posterior to quantify uncertainties (parameter estimation and model selection).

Proximal nested sampling (Cai, McEwen & Pereyra 2021):

- Constrained nested sampling formulation;
- Langevin diffusion MCMC sampling;
- Proximal calculus Moreau-Yosida approximation of constraint.

Proximal nested sampling (Cai, McEwen & Pereyra 2021):

- Constrained nested sampling formulation;
- Langevin diffusion MCMC sampling;
- Proximal calculus Moreau-Yosida approximation of constraint.

Proximal nested sampling Markov chain:

$$x^{(k+1)} = x^{(k)} + \frac{\delta}{2}\nabla \log \pi(x^{(k)}) - \frac{\delta}{2\lambda}\left[x^{(k)} - \text{prox}_{\chi_{\mathcal{B}_\tau}}(x^{(k)})\right] + \sqrt{\delta}w^{(k+1)} .$$

Handcrafted priors (*e.g.* promoting sparsity in a wavelet basis) are **not expressive enough**
$\Rightarrow$ learn data-driven prior given by denoising model (McEwen *et al.* 2023).

Handcrafted priors (*e.g.* promoting sparsity in a wavelet basis) are **not expressive enough**
$\Rightarrow$ learn data-driven prior given by denoising model (McEwen *et al.* 2023).

Tweedie's formula

Consider noisy observations $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{x}, \sigma^2 I)$ of $\boldsymbol{x}$ sampled from some underlying prior.

Tweedie's formula gives the posterior expectation of $\boldsymbol{x}$ given $\boldsymbol{z}$ as

$$\mathbb{E}(\boldsymbol{x} \,|\, \boldsymbol{z}) = \boldsymbol{z} + \sigma^2 \nabla \log p(\boldsymbol{z}),$$

where $p(z)$ is the marginal distribution of z.

Handcrafted priors (*e.g.* promoting sparsity in a wavelet basis) are **not expressive enough** $\Rightarrow$ learn data-driven prior given by denoising model (McEwen *et al.* 2023).

Tweedie's formula

Consider noisy observations $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{x}, \sigma^2 I)$ of $\boldsymbol{x}$ sampled from some underlying prior.

Tweedie's formula gives the posterior expectation of $\boldsymbol{x}$ given $\boldsymbol{z}$ as

$$\mathbb{E}(\boldsymbol{x} \,|\, \boldsymbol{z}) = \boldsymbol{z} + \sigma^2 \nabla \log p(\boldsymbol{z}),$$

where $p(z)$ is the marginal distribution of z.

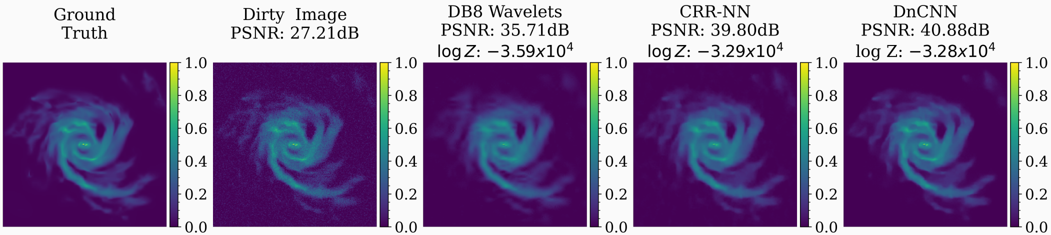
- Can be interpreted as a denoising strategy.
- Score of regualised prior related to learned denoiser by

$$\nabla \log \pi_\epsilon(\boldsymbol{x}) = \epsilon^{-1}(D_\epsilon(\boldsymbol{x}) - \boldsymbol{x}).$$

Hand-crafted vs data-driven priors

Consider simple Galaxy denoising inverse problem with:

- ▷ **hand-crafted prior** based on sparsity-promoting wavelet representation;
- ▷ **data-driven priors** based on a deep neural networks.

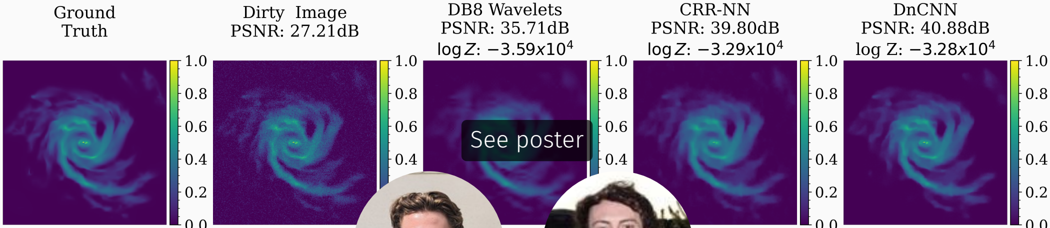


Which model best?

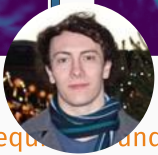
- ▷ SNR ⇒ **data-driven priors best** but require ground-truth;
- ▷ Bayesian evidence ⇒ **data-driven priors best** (no ground-truth knowledge).

Consider simple Galaxy denoising inverse problem with:

- ▷ **hand-crafted prior** based on sparsity-promoting wavelet representation;
- ▷ **data-driven priors** based on a deep neural networks.



See poster



Which model best?

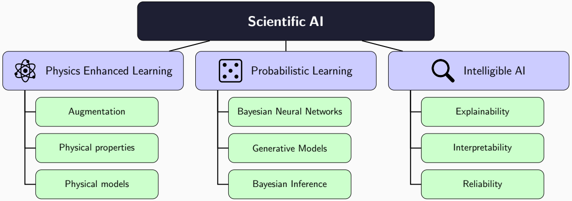
- ▷ SNR ⇒ **data-driven prio** ... but requ ... und-truth;
- ▷ Bayesian evidence = ... prio ... (no ground-truth knowledge).

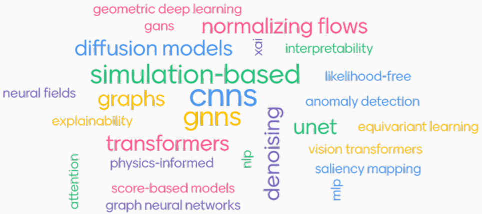
Henry Aldridge

Matt Price

Summary

Summary





SciAI UCL

UCL CDT DIS