A&A 672, A51 (2023)
https://doi.org/10.1051/0004-6361/202243054

Astronomy & Astrophysics

Probabilistic mass-mapping with neural score estimation*

B. Remy[1], F. Lanusse[1], N. Jeffrey[2,3], J. Liu[4,5,6], J.-L. Starck[1], K. Osato[7,2], and T. Schrabback[8]

[1] AIM, CEA, CNRS, Université Paris-Saclay, Université Paris Diderot, Sorbonne Paris Cité 91191 Gif-sur-Yvette, France
e-mail: benjamin.remy@cea.fr
[2] Laboratoire de Physique de l'École Normale Supérieure, ENS, Université PSL, CNRS, Sorbonne Université, Université de Paris, Paris, France
[3] University College London, Gower St, London, UK
[4] Berkeley Center for Cosmological Physics, University of California, 341 Campbell Hall, Berkeley, CA 94720, USA
[5] Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 93720, USA
[6] Kavli IPMU (WPI), UTIAS, The University of Tokyo, Kashiwa, Chiba 277-8583, Japan
[7] Center for Gravitational Physics, Yukawa Institute for Theoretical Physics, Kyoto University, Kitashirakawa Oiwakecho, Sakyo-ku, Kyoto 606-8502, Japan
[8] Argelander-Institut für Astronomie, Universität Bonn, Auf dem Hügel 71, 53121 Bonn, Germany

Received 6 January 2022 / Accepted 29 April 2022

ABSTRACT

Context. Weak lensing mass-mapping is a useful tool for accessing the full distribution of dark matter on the sky, but because of intrinsic galaxy ellipticities, finite fields, and missing data, the recovery of dark matter maps constitutes a challenging, ill-posed inverse problem

Aims. We introduce a novel methodology that enables the efficient sampling of the high-dimensional Bayesian posterior of the weak lensing mass-mapping problem, relying on simulations to define a fully non-Gaussian prior. We aim to demonstrate the accuracy of the method to simulated fields, and then proceed to apply it to the mass reconstruction of the HST/ACS COSMOS field.

Methods. The proposed methodology combines elements of Bayesian statistics, analytic theory, and a recent class of deep generative models based on neural score matching. This approach allows us to make full use of analytic cosmological theory to constrain the 2pt statistics of the solution, to understand any differences between this analytic prior and full simulations from cosmological simulations, and to obtain samples from the full Bayesian posterior of the problem for robust uncertainty quantification.

Results. We demonstrate the method in the κTNG simulations and find that the posterior mean significantly outperfoms previous methods (Kaiser–Squires, Wiener filter, Sparsity priors) both for the root-mean-square error and in terms of the Pearson correlation. We further illustrate the interpretability of the recovered posterior by establishing a close correlation between posterior convergence values and the S/N of the clusters artificially introduced into a field. Finally, we apply the method to the reconstruction of the HST/ACS COSMOS field, which yields the highest-quality convergence map of this field to date.

Conclusions. We find the proposed approach to be superior to previous algorithms, scalable, providing uncertainties, and using a fully non-Gaussian prior.

Key words. cosmology: observations – methods: statistical – gravitational lensing: weak

1. Introduction

The weak gravitational lensing effect provides a direct probe of the large-scale matter distribution in the Universe. This lensing effect generates minute deformations of the apparent shapes of distant galaxies, a so-called shear, in the presence of massive structures along the line of sight. Due to its ability to directly probe the matter field, weak lensing is found at the heart of present and upcoming wide-field optical galaxy surveys, including the ESA Euclid mission (Laureijs et al. 2011), the *Vera C. Rubin* Observatory Legacy Survey of Space and Time (Ivezić et al. 2019), and the *Roman* Space Telescope (Spergel et al. 2015).

While most of the cosmological analysis of weak lensing focuses on 2pt functions, the reconstruction of maps of the matter distribution opens up alternative ways to analyze data, including giving access to higher-order statistics, such as the peak count statistic, which has been applied to most existing weak lensing surveys (Liu et al. 2015a,b; Kacprzak et al. 2016; Shan et al. 2018; Martinet et al. 2018; Harnois-Déraps et al. 2021). In addition, novel higher-order statistics such as the wavelet peak counts and ℓ_1-norm (Ajani et al. 2021), the scattering transform statistics (Cheng et al. 2020), or neural summaries (e.g., Ribli et al. 2019; Jeffrey et al. 2020a), have recently been shown to be even more sensitive to cosmology.

This process of reconstructing maps of the matter distribution from measured galaxy ellipticities is known as weak lensing mass-mapping. Because of noise and missing data, the weak lensing mass-mapping problem is ill-posed; in other words, the matter density field is not uniquely determined by the observed shear. This implies that any mass-mapping method will rely on different prior assumptions to regularize this problem and yield a different map estimate. The most standard approach is the Kaiser–Squires method (Kaiser & Squires 1993), which is based on a direct inversion of the lensing operator along with some amount of Gaussian smoothing. A number of other

* All codes and data products associated with this paper are available at https://github.com/CosmoStat/jax-lensing.

methods have since been proposed, using various types of regularization schemes such as maximum entropy (Marshall 2001), Gaussian prior (Wiener filter; Simon et al. 2012; Horowitz et al. 2019), and sparsity (Starck et al. 2006, 2021; Leonard et al. 2012; Lanusse et al. 2016; Price et al. 2018). These techniques usually yield a point estimate of a convergence map, and usually lack proper uncertainty quantification, which makes interpreting the resulting maps difficult.

A number of Bayesian methods have been proposed to recover a posterior probability estimate for the unknown mass map, but these are usually limited by restrictive prior assumptions. For instance, using hierarchical Bayesian modeling, Alsing et al. (2016) proposed an approach for sampling mass-maps, but this relied on a Gaussian prior for the unknown map. More recently, Porqueres et al. (2022) extended this work and demonstrated mass-map posterior sampling with a nonlinear prior defined by a forward gravity model, but limited to very low angular resolution. Other Bayesian posterior sampling approaches include the approach by Schneider et al. (2016), which relied on a Gaussian Process prior, the one by Price et al. (2018), which proposed a proximal Markov chain Monte Carlo (MCMC) approach to accommodate non-differentiable sparsity priors, and the one by Fiedorowicz et al. (2022), which relies on a log-normal prior.

More recently, with the rise of deep learning, a number of methods relying on deep neural networks have been proposed to address the mass-mapping problem. The strength of these approaches is that they provide a practical way to leverage simulations as a prior to solve the mass-mapping problem. In particular, the `DeepMass` method (Jeffrey et al. 2020b) uses a U-net (Ronneberger et al. 2015) to recover an estimate of the mean posterior convergence map, with a prior defined by a set of simulations. Additionally, Shirasaki et al. (2021) have proposed a model based on a generative adversarial network (GAN; Goodfellow et al. 2014), which is able to to denoise weak lensing mass-maps. Similarly, Yiu et al. (2022) also proposed sampling mass-maps using GANs, taking into account the spherical curvature of the sky.

In this paper, we propose a new approach to mass-mapping that combines elements of deep learning with Bayesian inference and provides a tractable way to sample from the full high-dimensional posterior distribution of convergence maps.

The limiting factor in all of the Bayesian approaches mentioned above are the simplifications needed to achieve a tractable prior (which lead to Gaussian, log-Normal, sparse, or simplified hierarchical priors). These approximations are needed because the distribution of mass-maps is not tractable analytically, and is only accessible as an implicit distribution, namely a distribution without an explicit likelihood, and that can only be sampled from. Sampling from such an implicit simulation is otherwise known as running a simulator.

Our `DeepPosterior` approach is based on learning a prior from samples drawn from such an implicit distribution (i.e., from simulated convergence maps), and using this prior to sample the full Bayesian posterior.

Instead of learning a full probability density function $p(x)$ over mass-maps with likelihood-based deep generative models such as Normalizing Flows (Jimenez Rezende et al. 2014) or PixelCNNs (van den Oord et al. 2016; Salimans et al. 2017), following recent developments in the field of diffusion-based models, we instead targeted the score function, $\frac{\partial \log p(x)}{\partial x}$. This score function is estimated using the denoising score matching (DSM) technique (Vincent 2011), which relies on training a neural network under a simple denoising task, and yet leads asymptotically to an unbiased estimate of the score function. With this neural score estimation in hand, we demonstrate that one can sample from the posterior of the mass-mapping problem using an efficient gradient-based sampling technique: an annealed Hamiltonian Monte Carlo. Contrary to most similar deep learning approaches to solve inverse problems, this method is stable and scalable, and we achieved one independent sample from the mass-mapping posterior for maps of size 360×360 pixels in 10 GPU minutes.

We demonstrate our proposed methodology by applying it to simulations, using the high-resolution κTNG convergence maps (Osato et al. 2021), based on the IllustrisTNG simulations (Nelson et al. 2018, 2019; Pillepich et al. 2018; Springel et al. 2018; Naiman et al. 2018; Marinacci et al. 2018). We compare the method against other standard methods (Kaiser–Squires, `DeepMass`, `GLIMPSE`; Lanusse et al. 2016, `MCALens`; Starck et al. 2021) and find improvements in terms of the pixel reconstruction error, the Pearson correlation, and the convergence power spectrum. We also investigate the interpretation of the recovered posterior, and demonstrate a direct correlation between signal-to-noise ratio (S/N) of structures such as galaxy clusters and the posterior distribution.

Following these validation tests to simulations, we applied the method to the reconstruction of the HST/COSMOS field (Scoville et al. 2007) based on the shape catalog from Schrabback et al. (2010). We obtained the highest-quality mass-map of the HST/COSMOS field, alongside uncertainty quantification. Our result improves over the previously published COSMOS map Massey et al. (2007), both from using a more recent shape catalog, and from our much improved methodology, revealing much finer structures and providing uncertainty quantification.

The paper is organized as follows. After providing some general background on gravitational lensing in Sect. 2, and describing a unified view of mass-mapping and related works in Sect. 3, we introduce the methodology in Sects. 4 and 5. We first describe how to build a prior from cosmological simulations and how to sample from the posterior distribution. Then in Sect. 6, we describe the simulations we used to train our model. In Sect. 7 we validate our method, showing improvement in point estimate reconstruction against other methods, and present a detection experiment using the posterior samples. Finally, in Sect. 8, we apply our method to real data, reconstructing a very high-quality map of the HST/ACS COSMOS field.

2. Weak gravitational lensing formalism

In this section, we present an overview of weak gravitational lensing and mass-mapping.

Observed galaxy shapes are affected by the gravitational shearing effect that occurs in the presence of massive structures, acting as lenses, along the line of sight. This distortion can be described by a coordinate transformation (Bartelmann 2010) between unlensed coordinates β and observed image coordinates θ:

$$\beta = \theta - \nabla\psi(\theta), \tag{1}$$

where ψ is known as the lensing potential, and is sourced by the projected matter density on the sky.

To first order, the resulting distortions that affect galaxy images can be described in terms of a simple linear Jacobian

matrix, $\mathbf{A}$, known as the amplification matrix:

$$\beta = \mathbf{A}\theta = (1-\kappa)\begin{pmatrix} 1-\gamma_1 & -\gamma_2 \\ -\gamma_2 & 1+\gamma_1 \end{pmatrix}\theta\,. \tag{2}$$

In this expression, which only holds in the weak lensing regime (i.e., $\kappa \ll 1$), the convergence, κ, translates into an isotropic dilation of the source, while the shear, γ, causes anisotropic stretching of the image.

This convergence κ can be directly related to the projected mass density on the sky, which leads to typically using the denomination convergence map or mass-map interchangeably. Indeed, considering a sample of lensing sources distributed in redshift according to some distribution $n(z)$, one can relate the convergence κ to the three-dimensional matter overdensity δ according to

$$\kappa(\theta) = \frac{3H_0^2\Omega_\mathrm{m}}{2c^2}\int_0^{\chi_\mathrm{lim}} \mathrm{d}\chi \frac{q(\chi)}{a(\chi)} f_K(\chi)\delta(f_K(\chi)\theta,\chi), \tag{3}$$

where H_0 is the Hubble constant, c is the speed of light, $q(\chi) = \int_\chi^\infty \mathrm{d}\chi'\, n(\chi')\frac{f_K(\chi'-\chi)}{f_K(\chi')}$ is the lensing efficiency, f_K is the comoving angular distance, δ is the over density, a is the scale factor, and Ω_m is the matter density (Kilbinger 2015).

While shear can be measured by the spatially coherent correlations it induces on galaxy shapes, convergence is typically not directly observable, as its magnification effect is much more difficult to disentangle from intrinsic galaxy sizes. Therefore, the problem of mass-mapping is generally recovering an estimate of the convergence κ from measurements of the shear γ. This is made possible by the following equations that tie convergence and shear to the lensing potential:

$$\kappa = \frac{1}{2}\Delta\psi \quad ; \quad \gamma_1 = \frac{1}{2}\left(\partial_1^2\psi - \partial_2^2\psi\right) \quad ; \quad \gamma_2 = \partial_1\partial_2\psi. \tag{4}$$

Combining these equations, one can recover a minimum variance estimator for the convergence as

$$\kappa = \Delta^{-1}\left((\partial_1^2 - \partial_2^2)\gamma_1 + 2\partial_1\partial_2\gamma_2\right), \tag{5}$$

which constitutes the basis for the Kaiser–Squires reconstruction technique. This equation can be solved most efficiently in practice using a Fourier transform in the flat sky limit, or a spherical harmonics transform in the spherical setting.

The Fourier solution of the Kaiser–Squires estimator can be written as

$$\tilde{\kappa} = \frac{k_1^2 - k_2^2}{k^2}\tilde{\gamma}_1 + \frac{2k_1k_2}{k^2}\tilde{\gamma}_2, \tag{6}$$

where $k^2 = k_1^2 + k_2^2$. It should be noted that the solution is not defined for $k = 0$, which means that the mean of the convergence field cannot be directly constrained from shear, which is usually known as the mass-sheet degeneracy.

One particularly remarkable property of the Kaiser–Squires estimator Eq. (6) is that it defines a unitary operation. In other words, we introduce the linear operator $\mathbf{P}$ as

$$\tilde{\kappa}_E + i\tilde{\kappa}_B = \left(\frac{k_1^2 - k_2^2}{k^2} + i\frac{2k_1k_2}{k^2}\right)(\tilde{\gamma}_1 + i\tilde{\gamma}_2) = \mathbf{P}\,(\tilde{\gamma}_1 + i\tilde{\gamma}_2)\,, \tag{7}$$

where $\tilde{\kappa} = \tilde{\kappa}_E + i\tilde{\kappa}_B$ and $\tilde{\gamma} = \tilde{\gamma}_1 + i\tilde{\gamma}_2$ are complex representations of the convergence E and B modes, and of the two shear components in Fourier space. Then the operator $\mathbf{P}$ verifies $\mathbf{P}^\dagger\mathbf{P} = \mathbf{I}_\mathrm{d}$.

3. A unified view of mass-mapping

This mass-mapping problem can be reformulated as a probabilistic inference problem from a Bayesian perspective:

$$p(\kappa|\gamma) = \frac{p(\gamma|\kappa)p(\kappa)}{p(\gamma)}, \tag{8}$$

where the posterior distribution $p(\kappa|\gamma)$ models the probability of the signals κ conditioned on the observations, γ, the likelihood distribution, $p(\gamma|\kappa)$, encodes the forward process of the model in equation Eq. (7), the prior distribution, $p(\kappa)$, encodes the knowledge about the signal κ, and the Bayesian evidence, $p(\gamma)$, is the marginal density of the observations. The evidence is a constant if we assume a given model, and will be ignored in the rest of this work as we do not consider Bayesian model comparison.

In our work, the forward process encoded in Eq. (7) returns a binned shear map γ, where each pixel value corresponds to the average shear in the pixel area, and therefore takes as input a pixelized convergence map κ. Working with real data, it is assumed that the measurement for each pixel is degraded by shape noise n_s, due to the finite average of galaxy intrinsic ellipticities in the bin. This noise is assumed to be white Gaussian (i.e., $n_\mathrm{s} \sim \mathcal{N}(0, \Sigma_\mathrm{n})$, where Σ_n is the noise covariance matrix of the shear map). Moreover, because of missing data due to survey measurement masks, we need to explicitly consider that there is no measurement in some pixel regions. In practice, we set a very high variance, such as 10^{10}, for these pixels in the covariance matrix Σ_n. More specific information on the strategy we followed to emulate the COSMOS shape catalog can be found in Sect. 6.3.

Thus, the log-likelihood takes the following form:

$$\log p(\gamma|\kappa) = -\frac{1}{2}(\gamma - \mathbf{F}^*\mathbf{P}\mathbf{F}\kappa)^\dagger\Sigma_\mathrm{n}^{-1}(\gamma - \mathbf{F}^*\mathbf{P}\mathbf{F}\kappa) + \text{constant}, \tag{9}$$

where $\mathbf{F}$ and $\mathbf{F}^*$ are, respectively, the direct and inverse Fourier transform.

All existing mass-mapping techniques can be understood under the lens of this Bayesian formulation and will generally differ mostly in their choice of prior, and in the specific algorithm used to recover a point estimate of the convergence map. As in practice this problem is ill-posed due to noise corruption and missing data in Eq. (7), the posterior $p(\kappa|\gamma)$ can be both wide and heavily prior dependent, which explains why all these different techniques yield different answers. Below we describe several methods that we use in this paper for comparison.

3.1. Kaiser–Squires reconstruction

The Kaiser–Squires method (Kaiser & Squires 1993) can be seen as a simple maximum likelihood estimate (MLE) of the convergence map, typically followed by a certain amount of Gaussian smoothing:

$$\check{\kappa}_\mathrm{ks} = \arg\min_\kappa \| \gamma - \mathbf{F}^*\mathbf{P}\mathbf{F}\kappa \|_2^2 = (\mathbf{F}^*\mathbf{P}\mathbf{F})^\dagger\gamma\,, \tag{10}$$

$$\kappa_\mathrm{ks} = s * \check{\kappa}_{ks}, \tag{11}$$

where s is a Gaussian smoothing kernel of a given scale, and $\mathbf{P}^\dagger$ is a pseudo-inverse of the operator $\mathbf{P}$, typically achieved by a direct Fourier inversion. While this method is the fastest, it does not take into account masks, and leads to leakage between E and B modes of the convergence field. For Kaiser–Squires, the heteroscedasticity does not impact the solution, whereas it can do for certain extensions such as the Generalized Kaiser–Squires

method (GKS) (Starck et al. 2021; Appendix B.1), in which an iterative approach with little regularization takes the mask and noise heterescedasticity into account.

3.2. Wiener filter

The Wiener filter approach assumes a Gaussian random field prior on κ and takes advantage of the fact that the power spectrum of the convergence can be analytically predicted from cosmological models, and accurately describes the field on large scales. This prior on the convergence can be expressed as a Gaussian distribution with a diagonal covariance matrix $\mathbf{S}$ in Fourier space:

$$p_{\text{Gaussian}}(\kappa) = \frac{1}{\sqrt{\det 2\pi\mathbf{S}}} \exp\left(-\frac{1}{2}\tilde{\kappa}^{\dagger}\mathbf{S}^{-1}\tilde{\kappa}\right), \tag{12}$$

where $\mathbf{S}$ is the convergence power spectrum.

The solution of the inverse problem can be formulated as

$$\hat{\kappa}_{\text{wiener}} = \arg\min_{\kappa} \| \Sigma^{-1/2}(\gamma - \mathbf{F}^{*}\mathbf{P}\mathbf{F}\kappa) \|_2^2 + \log p_{\text{Gaussian}}(\kappa). \tag{13}$$

This Wiener solution corresponds to the maximum a posteriori (MAP) solution under this Gaussian prior, and also matches the mean of the Gaussian posterior. An appealing property of this estimator is that the solution can be easily recovered analytically in cases where the noise is homoscedastic, as both signal and noise covariance matrices become diagonal in Fourier space. The Wiener filter reconstruction (Lahav et al. 1994; Zaroubi et al. 1995) is given in Fourier space by:

$$\tilde{\kappa} = \mathbf{S}\mathbf{P}^{\dagger}\left[\mathbf{P}\mathbf{S}\mathbf{P}^{\dagger} + \mathbf{N}\right]^{-1}\tilde{\gamma}, \tag{14}$$

where $\mathbf{S}$ and $\mathbf{N}$ are, respectively, the signal and noise covariance matrix in Fourier space.

In more complex cases, where the noise covariance is not diagonal in Fourier space (for instance because of a mask in pixel space), the solution can still be efficiently recovered by optimization, using the proximal method (Bobin et al. 2012; Starck et al. 2021), or its related messenger field alternative (Elsner & Wandelt 2013). One can also draw samples from the Wiener posterior with the messenger field algorithm (Jeffrey et al. 2018a).

3.3. Sparse priors

Convergence maps contain non-Gaussian features that are not well recovered with the methods described above. Several mass-mapping algorithms have been proposed, relying on a wavelet sparsity prior (Starck et al. 2006, 2021; Leonard et al. 2012; Lanusse et al. 2016; Price et al. 2018), which can be formulated as:

$$\log p(x) = \| \mathbf{\Phi}^{t} x \|_{\text{p}}, \tag{15}$$

with $p < 2$, and where $\mathbf{\Phi}$ is a wavelet dictionary and $\| \cdot \|_{\text{p}}$ is a sparsity promoting ℓ_{p} norm.

The convergence map is the solution of the following sparse recovery optimization problem:

$$\hat{\kappa} = \arg\min_{\kappa} \| \Sigma^{-1/2}(\gamma - \mathbf{F}^{*}\mathbf{P}\mathbf{F}\kappa) \|_2^2 + \lambda \| \mathbf{\Phi}^{t}\kappa \|_{\text{p}}, \tag{16}$$

where λ is the regularization parameter, weighting the sparse regularization constraint. The `GLIMPSE` method (Lanusse et al. 2016) additionally allows masks, non-uniform noise, and flexion data (if they are available) to be taken into account, and also does not require the shear catalog to be transformed on pixelized map.

3.4. `DeepMass`

While all the priors described above have closed-form expressions, it is also possible to design a mass-mapping method where the prior is defined implicitly. The first dark matter map reconstruction from weak lensing observational data using deep learning was shown in Jeffrey et al. (2020b). `DeepMass` is a convolutional neural network (CNN) trained on pairs of simulated pixelized shear and convergence maps. One can show that, under some assumptions, a deep learning model can estimate the mean of the posterior distribution $p(\kappa|\gamma)$. In a nutshell, the network needs to be trained to minimize the mean-squared-error (MSE) of the output convergence κ, and the training convergence and shear maps must be drawn, respectively, from the prior distribution $p(\kappa)$ and the likelihood distribution $p(\gamma|\kappa)$.

While the Wiener filter assumes a Gaussian prior over the convergence, simulated training data for `DeepMass` are drawn from the "true" prior $p(\kappa)$, thus improving the accuracy of the reconstruction. `DeepMass` is therefore able to recover the non-linear structures of the convergence better than the Wiener filter, and is able to reduce the MSE of the reconstruction.

Even if `DeepMass` reconstructs high-quality convergence maps, `DeepMass` alone only provides the mean posterior and cannot quantify the uncertainties of the reconstruction. Furthermore, as any direct inversion method based on neural networks, the likelihood Eq. (9) is learned implicitly by the model, and does not explicitly constrain the solution at inference time. This means that although we have not found obvious failures in our experiments, the CNN may in theory fail in ways that would lead to a map not actually consistent with observations, for instance creating spurious artifacts, or missing structures present in the true map. Another side effect of implicitly learning the likelihood during training is that the model is trained for a specific survey configuration, and retraining is required if either the mask or the noise is different.

In this work, we propose a new approach that estimates the full posterior distribution $p(\kappa|\gamma)$, being able to not only recover the posterior mean, but also to quantify the uncertainties of the reconstruction. In addition, in our method the likelihood is explicit, meaning that it does not require retraining for a new survey configuration.

4. Primer on neural score estimation and sampling

In this section, we review the technical aspects of the machine learning methodology that we will employ in the mass-mapping problem. We begin by detailing how the score function can be estimated for an implicit distribution. We then describe our strategy to sample any distribution from the knowledge of its score.

There are many classes of generative models in the machine learning litterature, such as Generative adversarial networks (GANs; Goodfellow et al. 2014), Variational autoencoders (VAEs; Kingma & Welling 2013), Normalizing Flows (Jimenez Rezende et al. 2014), and Energy-based models (EBMs; Lecun et al. 2006).

All of these methods aim to model the probability distribution underlying some data, but not in the same way. On one hand, GANs and VAEs implicitly learn the probability distribution, which does not fit into our framework because we want to leverage the closed-form expression of the likelihood distribution. On the other hand, Normalizing Flows and EBMs can learn explicit forms of probability distribution, but do not scale well in dimension. Therefore, we resort to a recent and promising class of generative models introduced in Song & Ermon (2019) based

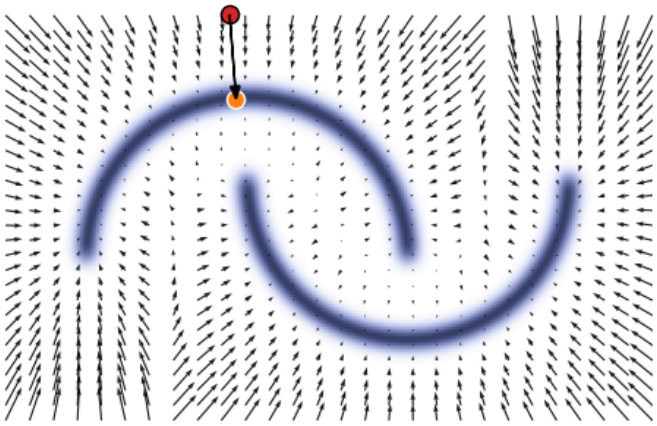


Fig. 1. Illustration of the score field (represented by the vector field) on the simple two-dimensional "two moons" distribution. The score field points in the direction of regions of higher probability. A noisy sample from the distribution is shown in red, while the denoised sample computed using equation Eq. (18) is shown in orange. This illustrates how the variance multiplied by the score function, represented by the black arrow, can project the sample onto the high density of the distribution.

on learning an explicit model of the gradient log-probability distribution, also known as the score function, which is all we need in our framework to build the posterior distribution, as described in Sect. 5.3. Score-based generative models have demonstrated a state-of-the-art quality level for image generation (Song et al. 2020; Nichol & Dhariwal 2021; Dhariwal & Nichol 2021), and particularly astrophysical images such as realistic galaxy image generation (Smith et al. 2022).

4.1. Denoising score matching

As originally identified by Vincent (2011) and Alain & Bengio (2013), the gradient with respect to the data of a log distribution, which is called the score function, can be modeled using a denoising autoencoder (DAE). This method is also known as denoising score matching (DSM).

We introduce an auto-encoding function $\boldsymbol{r}_\theta : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N, (\boldsymbol{x}', \sigma) \mapsto \tilde{\boldsymbol{x}}$, where N is the signal dimension, trained to reconstruct a true signal $\boldsymbol{x}$ following the probability distribution p, given a noisy version $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{n}$, with $\boldsymbol{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_N)$, under an ℓ_2 loss. In this case, the function $\boldsymbol{r}_\theta$ is called a denoiser, parametrized by a noise level σ, and is trained minimizing the following criterion:

$$\mathcal{L}_{\text{DAE}} = \mathbb{E}_{\boldsymbol{x}' \sim p_{\sigma^2}} \left[\|\boldsymbol{x} - \boldsymbol{r}_\theta(\boldsymbol{x}', \sigma)\|_2^2 \right] \tag{17}$$

where $p_{\sigma^2} = p * \mathcal{N}(0, \sigma^2)$, namely it corresponds to the data distribution we want to model, convolved by a multivariate Gaussian with a diagonal covariance matrix $\sigma^2 \mathbf{I}_N$.

Alain & Bengio (2013) showed that in this setting, an optimal denoiser $\boldsymbol{r}^\star$ would then be achieved for:

$$\boldsymbol{r}^\star(\boldsymbol{x}, \sigma) = \boldsymbol{x} + \sigma^2 \nabla_{\boldsymbol{x}} \log p_{\sigma^2}(\boldsymbol{x}) + o(\sigma^2), \quad \text{as} \quad \sigma^2 \to 0. \tag{18}$$

Figure 1 illustrates how one can denoise a two-dimensional point knowing the noise level and the score function. We can clearly see that the score multiplied by the variance gives the right shift between the noisy point (in red) and the high-density region of the distribution. Applying this shift to the noisy sample gives the projected sample in orange.

Rewriting this equation gives an estimator of the score function:

$$\nabla_{\boldsymbol{x}} \log p_{\sigma^2}(\boldsymbol{x}) = \frac{\boldsymbol{r}^\star(\boldsymbol{x}, \sigma) - \boldsymbol{x}}{\sigma^2} + o(1), \quad \text{as} \quad \sigma^2 \to 0. \tag{19}$$

The optimal denoiser is thus related to the score we wish to learn, and exactly matches the score of the probability distribution underlying data when the noise variance σ^2 goes to zero.

Following the loss function design in Lim et al. (2020), the optimization objective to train the denoiser is:

$$\mathcal{L}_{\text{AR-DAE}} = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x} \sim P \\ \boldsymbol{u} \sim \mathcal{N}(0, I) \\ \sigma \sim \mathcal{N}(0, s^2)}} \left[\| \boldsymbol{u} + \sigma \boldsymbol{r}_\theta(\boldsymbol{x} + \sigma \boldsymbol{u}, \sigma) \|_2^2 \right], \tag{20}$$

where $\boldsymbol{u}$ is sampled from a standard multivariate Gaussian, r_θ is a denoiser parametrized by neural network weights θ, optimized to estimate r^*. The modifications between Eqs. (17) and (20) resolve some numerical instabilities and reduce the error of approximation of the precedent objective. In particular, the learned denoiser $\boldsymbol{r}_\theta$ now directly models the score function, without the need to divide by the noise level (i.e., $\boldsymbol{r}_\theta(\boldsymbol{x}, \sigma^2) = \nabla_{\boldsymbol{x}} \log p_{\sigma^2}(\boldsymbol{x})$, as $\sigma^2 \to 0$). Moreover, rescaling by σ and decoupling the noise level from an isotropic Gaussian noise prevents the gradient from vanishing.

To summarize, DSM provides a tractable way of estimating a score function, from only having access to samples from an implicit distribution. We will now be able to use this score estimate to sample from said distribution, as detailed in the next section.

4.2. Score-based sampling

Given the score function $\nabla \log p(\boldsymbol{x})$ (either directly available, or learned by DSM), it is possible to sample from the distribution $p(\boldsymbol{x})$ by an MCMC. Indeed, the Langevin Dynamics (LD) or Hamiltonian Monte Carlo (HMC) updates, described in Neal (2011) and Betancourt (2017), only require the evaluation of the score function $\nabla \log p(\boldsymbol{x})$. For instance, the HMC update, based on the leapfrog integrator, requires the evaluation of the score function only:

$$\begin{aligned} \boldsymbol{m}_{t+\frac{\alpha}{2}} &= \boldsymbol{m}_t + \frac{\alpha}{2} \nabla \log p(\boldsymbol{x}_t) \\ \boldsymbol{x}_{t+\alpha} &= \boldsymbol{x}_t + \alpha \mathbf{M}^{-1} \boldsymbol{m}_{t+\frac{\alpha}{2}} \\ \boldsymbol{m}_{t+\alpha} &= \boldsymbol{m}_{t+\frac{\alpha}{2}} + \frac{\alpha}{2} \nabla \log p(\boldsymbol{x}_{t+\alpha}) \end{aligned} \tag{21}$$

where α is the step size, $\boldsymbol{m}$ is the auxiliary momentum, and $\mathbf{M}$ is a preconditioning matrix that could take into account the space metric, but in our case the identity matrix.

Following this procedure, HMC is supposed to sample from $p(\boldsymbol{x})$, but as explained in Betancourt (2017), the discretization induces a small error that will bias the resulting transition and requires a correction. In order to correct this bias, every sample is considered as a Metropolis–Hastings (MH; Metropolis et al. 1953; Hastings 1970) proposal and is accepted or rejected according to an acceptance probability. This acceptance probability is designed from the Hamiltonian transition and is also only score-dependent, as we show in Appendix A.

However, in most, if not all but the most trivial cases, sampling in high dimension using Langevin or Hamiltonian dynamics is made very difficult by the fact that the distribution manifold is never Euclidean, meaning that assuming a diagonal noise

covariance for LD, or a diagonal momentum matrix for HMC, leads to extremely inefficient sampling.

To take a concrete example, we consider the case of distribution of handwritten digits in the MNIST dataset, and imagine an LD chain exploring this distribution. To transition from a one to a seven, for instance, the chain running in pixel space will try to add some white Gaussian noise at each update. However, MNIST digits are binary, so any addition of noise is bound to kick the chain out of the data distribution, and be rejected in a Metropolis–Hastings step. Only the smallest step sizes ϵ (which also tunes the amount of noise applied on the chain) will have a nonzero chance of acceptance, meaning the chain is practically never moving.

To summarize, having access to the score function is all we need to sample from a distribution, but this remains difficult for nontrivial high-dimensional distributions. In the next subsection, we describe a strategy to circumvent this issue.

4.3. Efficient sampling in high dimensions with annealing

Various approaches have been suggested to increase the sampling efficiency of HMC and LD for complex distributions, which generally aim to reframe the chain in a space closer to the intrinsic manifold of the distribution, rather than pixel space. For instance, Girolami & Calderhead (2011) exploited the Riemannian geometry of parameters space to define a Metropolis–Hastings (Metropolis et al. 1953; Hastings 1970) proposal, enabling high-efficiency sampling in high dimensions.

In our work, we follow another direction that recently gained momentum from the literature on denoising diffusion models (Sohl-Dickstein et al. 2015; Song & Ermon 2019; Ho et al. 2020). Instead of trying to follow the non-Euclidean latent manifold of the distribution, we look at transforming this distribution so that it becomes easier to travel in pixel-space. This can be done very simply by convolving the data distribution with noise.

To go back to our MNIST thought experiment, if we convolve our binary handwritten digits with Gaussian noise, it becomes very easy for an LD chain to move in pixel space, as the noise added by the chain can be made to match the noise present in the distribution. The higher the noise, the larger the pixel-wise transition will be, and the faster the chain can transition from one digit to the other. Another view of this effect is that, given enough noise, all of the distinct modes of a distribution will begin to merge with each other. Figure 2 illustrates the same two-moons distribution convolved with varying amount of Gaussian noise. At high noise values (top left corner), the distribution becomes close to a diagonal Gaussian, and thus extremely easy to sample in order to explore all possible regions of the distribution.

We will call temperature T the variance σ^2 of this Gaussian kernel convolved with the target distribution.

Following our preliminary work (Remy et al. 2020), in this paper we adopt a two-step sampling procedure, based on an annealed version the HMC (Neal 2011; Betancourt 2017), similar to the annealed LD proposed by Song & Ermon (2019).

In the first step of our sampling procedure, we initialize a chain using white Gaussian noise with a high temperature σ_{T}^2. Leveraging the conditional noise property of the score function described in Sect. 4.1, we have direct access to the score function of the convolved distribution $\nabla \log p_{\sigma_{\mathrm{T}}^2}$, which we can use in a score-based HMC. We then let the chain evolve under Hamiltonian dynamics, and progressively lower the temperature σ^2 of the conditional score with a geometric schedule of common ratio equal to 0.98. Each time the temperature σ^2 is decreased, the MCMC chain thermalizes to the new temperature

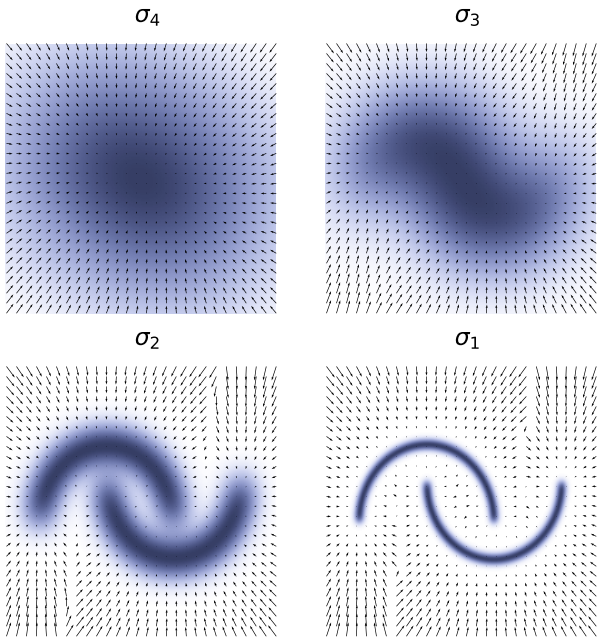


Fig. 2. Example of annealing on the two moons distribution. The distribution is convolved with a multivariate Gaussian of variance σ. This variance decreases over the sampling procedure so that the MCMC is always over high-density regions, i.e., $\sigma_4 > \sigma_3 > \sigma_2 > \sigma_1$. The chain is initialized from a wide multivariate Gaussian, as in the top left panel, and converges to the data distribution, as in the bottom right panel. In blue is the density of the convolved distribution and the arrows represent its score function.

in a few HMC steps. As described in Song et al. (2019), small enough steps are important to ensure nonzero transition probabilities between adjacent temperature values. Once the chain has reached a sufficiently low temperature (ideally $\sigma^2 = 0$), we stop the chain and only retrieve the last sample. Multiple independent samples are obtained by running multiple independent annealed HMC chains in parallel. One must not think that having one chain per sample makes the process much longer than having multiple samples per chain. It is indeed much more efficient, because in practice it is very long to ensure sample independence within the same chain.

In practice, however, it has proven to be difficult to anneal chains all the way to zero temperature. We find that, at low temperatures, the chains do not properly thermalize at each step and residual noise remains in our samples. This was also observed by other authors using annealed LD instead of HMC (Jolicoeur-Martineau et al. 2020; Song & Ermon 2019). In the application in this paper, we can reach $\sigma = 10^{-3}$ by fine-tuning our annealing scheme.

Therefore, in a second step of our sampling procedure, we propose a strategy to transport the annealed HMC samples obtained in step 1 at a finite temperature all the way to $\sigma^2 = 0$. To achieve this, we use remarkable results from Song et al. (2020), which establish a parallel between denoising diffusion models (generative models based on random walks) and stochastic differential equations (SDEs). We refer the interested reader to Appendix B for more mathematical details and derivations, but the main result from that paper that we use here is the following ordinary differential equation (ODE):

$$\mathrm{d}\boldsymbol{x} = -\frac{1}{2}\sqrt{\frac{\mathrm{d}}{\mathrm{d}t}\left(\sigma^2(t)\right)}\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\mathrm{d}t, \tag{22}$$

where $\sigma^2(t)$ is the increasing variance schedule of the Gaussian distribution with which we convolve the data distribution. In our particular implementation, we used a linear schedule $t \mapsto \sigma^2(t) = t$. This ODE describes a deterministic process $\{\boldsymbol{x}(t)\}_{t=0}^{\mathrm{T}}$ indexed by a continuous time variable $t \in [0, T]$, such that $\boldsymbol{x}(0) \sim p_0$ and $\boldsymbol{x}(T) \sim p_{\mathrm{T}}$, where p_0 denotes the data distribution and p_{T} the convolution between the data distribution and a multivariate Gaussian of variance T. This ODE can be solved by any black-box ODE solver, provided that the convolved score is available. In particular, it means that if we start the ODE at a point in p_{t}, where t is the intermediate temperature at which we have stopped our HMC chains, we can transport that point to p_0. This procedure very effectively removes residual noise while making sure we reach a point in the target distribution at zero temperature.

We summarize our full sampling procedure as follows. First, we initialize with white Gaussian noise $x_{\mathrm{init}} \sim \mathcal{N}(0, T \cdot \boldsymbol{I}_{\mathrm{d}})$. Next, we sample the posterior distribution with the annealed HMC algorithm, which, at each step, requires: (a) evaluating the convolved score $\nabla \log p_\sigma(\gamma|\kappa)$ using Eq. (19); (b) computing the MH proposal using Eq. (21) and accepting it according to Eq. (A.1); and (c) annealing if there are enough samples for a given temperature. Finally, we project the last sample on the posterior distribution with the ODE described in Eq. (22).

5. Mass-mapping by neural score estimation

Having laid out the machine learning notions needed to implement our method, we now describe how to apply these techniques to the weak lensing mass-mapping problem. Our strategy will be to use DSM to learn a high-fidelity prior over convergence maps, and use this prior in combination with the analytic likelihood function Eq. (9) to access the full posterior of the mass-mapping problem by annealed HMC sampling.

5.1. Hybrid analytic and generative modeling of the prior

One of the main limiting factors of previous mass-mapping approaches is the limited complexity of the prior used in the inversion problem (e.g., Gaussian, or sparse). Our first step toward our mass-mapping technique is to build a prior that can fully capture the non-Gaussian nature of convergence maps. To achieve this goal, we propose an hybrid prior relying on a Gaussian analytic model, complemented by a DSM approach to learn from simulations, the delta between this Gaussian prior and fully non-Gaussian convergence map distribution, accessible through simulations.

We indeed know that analytic cosmological models provide a reliable model of the 2 pt functions of the convergence field, and can thus be used to define a Gaussian prior, which enjoys a closed-form expression (Eq. (12)), as discussed in Sect. 3.2, in the context of the Wiener filter. Moreover, because of its tractable expression, we directly have access to its score function $\nabla \log p_{\mathrm{Gaussian}}(\kappa)$. This Gaussian prior is particularly adapted on large scales, where the matter distribution is well modeled by a Gaussian random field, but does not capture the significantly non-Gaussian nature of the convergence field on small scales.

On the other hand, we do have access to physical models that can capture the full statistics of the convergence field, in the form of numerical simulations. In these cases, however, the simulator gives us access to an implicit distribution, meaning we can only draw samples from the distribution, but we do not have access to the likelihood of a given sample under the model. We cannot, therefore, directly use black-box simulators as priors for sampling the Bayesian posterior. This is where we can leverage deep generative models to turn samples from an implicit distribution into a model with a tractable likelihood that can be used for inference.

In this work, we propose using DSM for this task, but we also want to capitalize as much as possible on the analytic Gaussian prior. Consequently, we aim to use the neural network to model the higher-order moments of the convergence field that the Gaussian prior cannot capture. In a DSM framework, it is straightforward to implement such a network in practice by feeding the neural network denoiser with: (1) the noisy input image, (2) the noise level, and (3) the Gaussian score. The full prior $p(\kappa)$ is then computed as

$$\nabla \log p(\kappa) = \nabla \log p_{\mathrm{Gaussian}}(\kappa) + \boldsymbol{r}_\theta\left(\kappa, \nabla \log p_{\mathrm{Gaussian}}(\kappa)\right), \tag{23}$$

where $\boldsymbol{r}_\theta$ is the DAE trained to model only the residuals from the Gaussian prior. During learning, a first step of denoising is performed using $\nabla \log p_{\mathrm{Gaussian}}(\kappa)$, then the neural network r_θ is optimized to denoise from this Gaussian guess, and is hence referred to as the residual score.

Just as for conventional DSM, we can train the model on simulations by adapting Eq. (20) to residual denoising score matching:

$$\mathcal{L}_{\mathrm{RDSM}} = \mathop{\mathbb{E}}_{\substack{\boldsymbol{u} \sim \mathcal{N}(0, I) \\ \sigma \sim \mathcal{N}(0, s^2)}} \| \boldsymbol{u} + \sigma * [\boldsymbol{r}(\boldsymbol{x} + \sigma * \boldsymbol{u}, \sigma, \nabla \log p_{\mathrm{G}}(\boldsymbol{x} + \sigma * \boldsymbol{u})) + \nabla \log p_{\mathrm{G}}(\boldsymbol{x} + \sigma * \boldsymbol{u})] \|^2, \tag{24}$$

where p_{G} is the Gaussian prior.

Figure 3 provides an illustration of this hybrid prior. From the noisy input map b, we first compute the Gaussian score function. Panel c shows the Gaussian denoiser estimate computed with Eq. (18) from the Gaussian score function. Then, feeding the DAE with the noisy input, the input noise level, and the Gaussian score function, we get the full score function. Panel d shows the complete denoising estimate computed with Eq. (18), again from the score function. Figure 3 exemplifies the ability of the neural network to capture complementary scales to the Gaussian prior.

Building the prior with this hybrid design reduces the complexity of the modeling problem, and limits the reach of the neural network model by only modeling a correction to the analytical prior.

5.2. Neural score estimator architecture for mass-mapping

The DSM approach described in Sect. 4.1 was so far completely generic and agnostic of the actual implementation of the parametric denoiser r_θ. We describe here the concrete neural network architecture that we will use throughout this work, fine-tuned to image data of size 360×360 pixels.

Following Eq. (19), we need to train a DAE in order to model the prior score function. We used a U-net, inspired from Ronneberger et al. (2015), with ResNet building blocks of convolutions from He et al. (2016) followed by batch normalization. We also used spectral normalization, using a power iteration method on the neural network weights Gouk et al. (2020), to improve the regularity of the network in out-of-distribution regions, where the score is not constrained by the data. Indeed, as discussed in Appendix C, regularizing the spectral norm lowers the Lipschitz constant of the network, and thus enforces the score field to be aligned for close inputs. It should be noted that the network is noise conditional, which means that it takes as

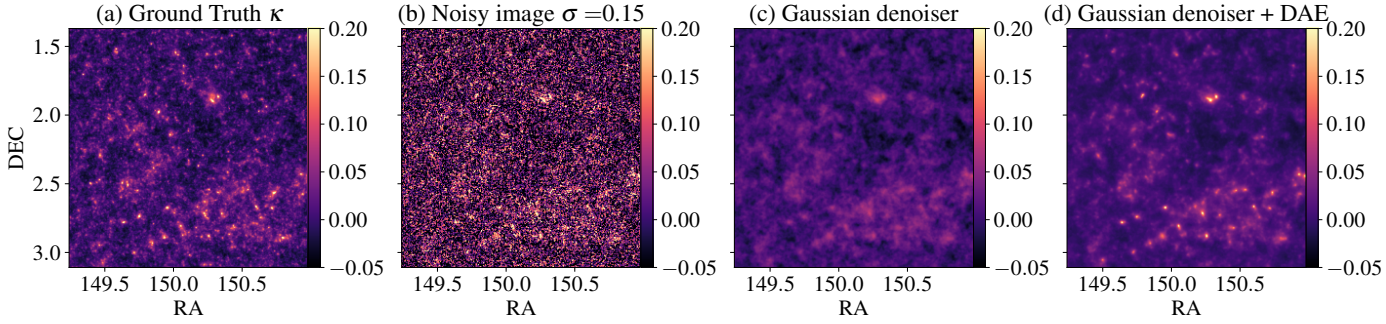


Fig. 3. Denoising of a $100 \times 100\,\mathrm{arcmin}^2$ simulated convergence map (a), which was corrupted with a noise level of $\sigma = 0.15$ per pixel (b). The Gaussian denoiser (c), computed with Eqs. (18) and (12), retrieves large scales of the convergence map. Adding the DAE residuals to the Gaussian denoiser (d) allows us to recover both large and small scales of the κ map. This illustrates the Gaussian (analytic) and non-Gaussian (learned) parts of our hybrid score function Eq. (19).

inputs both the image to denoise and the noise level. We distinguish two kinds of noise: the shape noise and the temperature noise. The latter is given as input to the network. The shape noise is encoded in the likelihood covariance matrix, and so is not taken into account in the network. The noise level was drawn from a normal distribution of standard deviation $\sigma = 0.2$. Our U-net denoiser was trained using the Adam optimizer (Kingma & Welling 2013), starting with a learning rate of 10^{-4} and decreasing up to 10^{-7} at the end. We used a batch size of 32 to train the neural network for 40 000 steps.

More details on the architecture, training, and regularization are given in Appendix C.

5.3. Sampling from the posterior

The prior learned with DSM is naturally defined as a convolved version of the data distribution. Besides, in order to use annealing on the posterior distribution, it is necessary to convolve the likelihood as well. Under the assumption that the likelihood is Gaussian, convolving with a multivariate Gaussian on $\boldsymbol{x}$ gives the following expression:

$$\log p_{\sigma_{\mathrm{L}}^2}(\boldsymbol{y}|\boldsymbol{x}) = -\frac{\|\boldsymbol{y} - \mathbf{P}\boldsymbol{x}\|_2^2}{2(\sigma_{\mathrm{n}}^2 + \sigma^2)} + \text{constant}, \tag{25}$$

where σ_{L}^2 is the variance of the convolved likelihood, σ_{n}^2 is the noise variance of the measurement, and σ^2 is the variance of the convolved Gaussian, also called the temperature. We provide a demonstration of Eq. (25) in Appendix D.

It is important to note that the MCMC algorithm does not navigate the posterior distribution convolved at temperature σ^2 at the same rate as the likelihood or the prior. Indeed, convolving the product of two distributions is not equivalent to convolving each distribution independently. However, we found empirically that the annealed HMC converges toward the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ at zero temperature. We first demonstrate it with a Gaussian posterior in Sect. 7.1 and then with the full posterior distribution in Sects. 7.2.1, 7.2.2, and 7.3.

This way, the complete sampling procedure consists of running an MCMC on a proxy of the posterior distribution, which is gradually annealed to low temperature, with chains progressively moving toward the posterior distribution. Each chain is initialized by sampling a multivariate normal distribution of unit variance and we leverage the usage of GPUs by running several chains in batches. Song & Ermon (2019) stress that random initialization guarantees obtaining independent samples from the distribution. They also demonstrate that this does not only allow us to sample different modes of the distribution, but also to recover the relative weights of these modes.

6. Simulations and data

In this section we describe the simulated data we used to validate our method, designed to emulate the COSMOS field on which we aim to apply our algorithm.

6.1. COSMOS shear catalog

The COSMOS survey (Scoville et al. 2007) is a contiguous $1.64\,\mathrm{deg}^2$ field imaged with the Advanced Camera for Surveys (ACS) in the F814W band. In this paper, we use the shape catalog obtained in Schrabback et al. (2010; hereafter S10) by reduction of this survey using the KSB+ method (Schrabback et al. 2007; Erben et al. 2001; Kaiser et al. 1995; Hoekstra et al. 1998), applying modeling for the variable HST/ACS point spread function using principal component analysis and a correction for multiplicative shear measurement bias, which was calibrated as a function of the galaxy signal-to-noise ratio (S/N) using image simulations.

The S10 catalog is divided into a bright $i^+ < 25$ (Subaru `SExtractor MAG_AUTO` magnitude) and a faint $i^+ > 25$ galaxy sample. For the bright sample, individual high-quality photometric redshifts are available by cross-matching against the COSMOS-30 catalog (Ilbert et al. 2009), while for the faint sample, Schrabback et al. (2010) proposed a functional form for the overall $n(z)$ based on an extrapolation to fainter magnitudes of the i_{814}–redshift relation observed in the $23 < i_{814} < 25$ range. The redshift distribution of both samples is illustrated in Fig. 4.

In this analysis, we combine both bright and faint galaxy samples into a single shape catalog, and will assume the combined redshift distribution illustrated in Fig. 4. The only cut we apply is to reject galaxies in the bright sample with $z_{\mathrm{phot}} < 0.6$ and $i^+ > 24$. The photometric redshift cut is motivated by S10 finding indications that many of these galaxies are truly at high redshifts. Thus, their inclusion would imply that the used estimate of the redshift distribution is inaccurate. This yields a total of 417 117 galaxies, which translates into a mean number of galaxies 64.2 per arcmin^2.

6.2. κTNG simulations

Having described the COSMOS field, we now present the κTNG suite of simulations we used to create mock data.

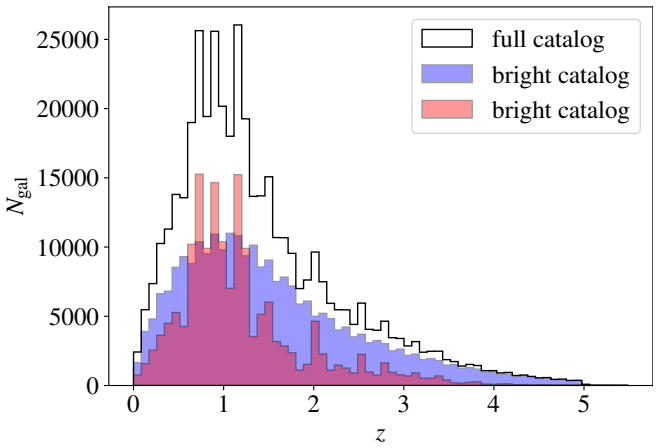


Fig. 4. Redshift distribution of the COSMOS catalog from Schrabback et al. (2010). The redshifts for bright galaxies ($i^+ < 25$) are derived from the COSMOS-30 catalog (Ilbert et al. 2009), while the $n(z)$ distribution for faint galaxies ($i^+ > 25$) is based on an extrapolation to fainter magnitudes as described in Schrabback et al. (2010).

We used the κTNG simulated convergence maps (Osato et al. 2021), generated from the IllustrisTNG (TNG) hydrodynamic simulations (Nelson et al. 2018, 2019; Pillepich et al. 2018; Springel et al. 2018; Naiman et al. 2018; Marinacci et al. 2018). The TNG simulations are a set of cosmological, large-scale gravity and magneto-hydrodynamical simulations, where baryonic processes such as stellar evolution, chemical enrichment, gas cooling, and supernovae and black hole feedback are incorporated as subgrid models.

The κTNG suite is generated through tracing the trajectories of light-rays from $z = 0$ to the target source redshift. A large number of realizations are generated by randomly translating and rotating the snapshots. Specifically, the full set includes 10 000 realizations of 5×5 deg^2 convergence maps for 40 source redshifts up to $z_s = 2.6$, at a 0.29 arcmin pixel resolution. A flat Λ-cold dark matter cosmology is assumed for both TNG and κTNG: Hubble constant $H_0 = 67.74\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$, baryon density $\Omega_b = 0.0486$, matter density $\Omega_m = 0.3089$, spectral index of scalar perturbations $n_s = 0.9667$, and amplitude of matter fluctuations at $8\,h^{-1}$ Mpc $\sigma_8 = 0.8159$. Because κTNG maps are computed by ray-tracing through the IllustrisTNG hydrodynamic simulations, there is no need for post-Born correction. Though the current κTNG simulations do not incorporate intrinsic alignment, we can address the effect using the halo or galaxy shape (Shi et al. 2021; Kurita et al. 2021).

6.3. Mock COSMOS data

To act as our prior for the reconstruction of the COSMOS field, and for validation tests, we emulate mock COSMOS lensing catalogs from κTNG simulations.

Using the binned galaxy distribution from the S10 catalog at the same 0.29 arcmin resolution, we define a binary mask that captures the limits of the survey, as well as missing data within the survey area due to bright stars or image artifacts that prevent the reliable measurement of galaxy shapes in some regions of the survey. This binning also yields noise variance maps, defined by the standard deviation of intrinsic galaxy ellipticities, rescaled by the number of galaxies per pixels N_i:

$$\Sigma_{(i,i)} = \begin{cases} \frac{\sigma_e^2}{N_i} & \text{if } N_i > 0 \\ 10^{10} & \text{otherwise} \end{cases} \tag{26}$$

for each pixel i, and $\sigma_e = \langle e, e \rangle$, with e ellipticities from the Schrabback et al. (2010) catalog. It it important to note that in empty pixels, we assume a large variance instead of infinity.

κTNG provides finely sampled convergence source planes in the range $0 < z_s < 2.6$. We combine these source planes to match the redshift distribution of the survey illustrated in Fig. 4. To handle source redshifts higher than $z_s \geq 2.6$, we resorted to a redshift recycling approach in which the last source plane at $z = 2.6$ was reused, with an appropriate weight designed to match the expected lensing kernel. This procedure yields a total of 10 000 convergence maps, now properly weighted to match the redshift distribution of the binned data.

Mock observations of COSMOS shear maps are obtained by first computing the shear from simulated convergence maps, sampling spatially variant noise according to Σ, and applying the binary mask to mask out unobserved regions.

7. Tests with simulated data

In this section, we validate our sampling procedure on mock COSMOS data generated as described in the previous section. We begin by demonstrating our sampling method in an analytically tractable Gaussian case, before testing full posterior sampling using the neural score matching approach introduced in Sect. 5.

7.1. Sampling validation with an analytic Gaussian prior

In this section, we assume that the convergence κ is a Gaussian random field, and therefore that it is associated to a Gaussian prior. We show that one can compute the Wiener filter estimate using our posterior sampling method.

Our method, based on the evaluation of the gradient of the log posterior, enables us to recover both the MAP and the mean posterior, which should match.

We used the halofit matter power spectrum from `jax-cosmo`[1], using the Planck Collaboration XIII (2015; Table 4, final column) results, to build the Gaussian prior covariance matrix. In the following, we also refer to it as the "fiducial" power spectrum. From the score function of the Gaussian posterior, (i.e., $\nabla \log p(\kappa|\gamma)$), one can compute the MAP using the gradient descent algorithm. As the posterior distribution is Gaussian, there is only one minimum, so we are sure to obtain the MAP. The MAP matches the posterior mean, so this also provides a check of the validity of our sampling procedure.

Figure 5 compares the Wiener filter computed with the messenger field method, from Elsner & Wandelt (2013), and our score-based MAP and posterior mean reconstructions for the same input shear field. The residuals between the MAP and the Wiener filter are very low (two orders of magnitude smaller than the signal), and the residuals are also unstructured, as expected; both are computed with the same input data and fiducial cosmology, and should mathematically match. We also show the relative error between the power spectrum of the Wiener filter and the MAP, and the relative error between the Wiener filter and the posterior mean, which is almost zero at every scale. The posterior mean is computed by averaging 1500 posterior samples, sampled with the annealed HMC presented in Sect. 4.2, reaching 10^{-3} temperature and then projected on the posterior distribution

[1] https://github.com/DifferentiableUniverseInitiative/jax_cosmo

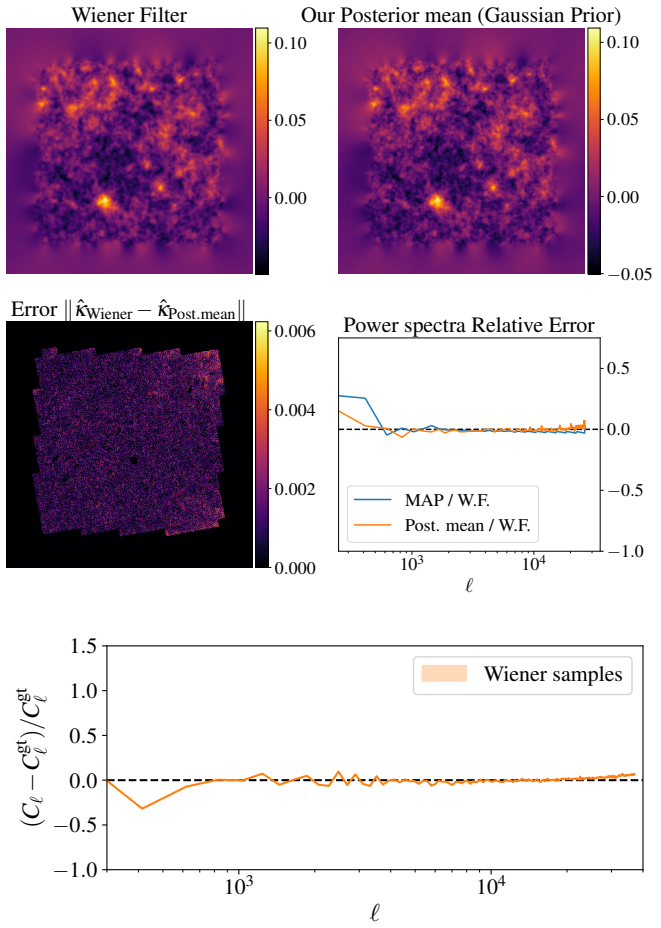


Fig. 5. Comparison between the Wiener filter (top left) and our score-based MAP (not shown) and posterior mean reconstructions (top right, average of 1500 samples), assuming a Gaussian prior. The MAP reconstruction matches the Wiener filter, both in terms of absolute pixel error and power spectrum ratio (bottom right, blue line). Our posterior mean is also in excellent agreement with the Wiener filter, as illustrated by the pixel absolute error (bottom left) and the power spectrum relative error (bottom right, orange line). The bottom panel shows the relative error between the Gaussian posterior samples' power spectra and the theoretical power spectrum.

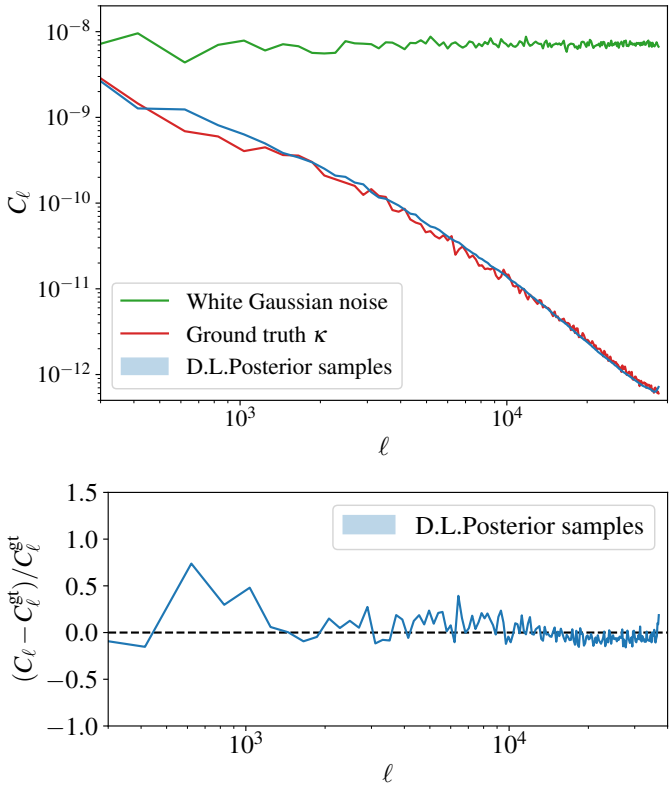


Fig. 6. Comparison between the κTNG samples' power spectrum and our posterior power spectrum. The red curve is computed from the κTNG simulated map. The green curve is the power spectra of a white Gaussian noise realization that we use to initialize the chains. The blue curve is the mean power spectrum of posterior samples, assuming our proposed hybrid prior, and the blue interval represents the standard deviation of the power spectra computed over 400 samples.

with the ODE sampler. The amount of samples was chosen to get the relative power spectrum to zero at all scales.

This experiment with a Gaussian prior validates that our sampling procedure is well adapted to sampling from the posterior distribution in this analytically tractable case.

7.2. Tests with a simulation-based prior

7.2.1. Sampling with a neural prior

In the last section, we showed that with the gradient of the log-Gaussian posterior, we can sample the posterior distribution, and that we can therefore recover classical results such as the Wiener filter estimate. In this section, we apply the same sampling procedure, using a simulation-based prior.

In Fig. 3 we saw that denoising a convergence map with a deep neural network trained on κTNG simulated maps (using the full prior) is much more effective for small-scale structure inference than using the Gaussian assumption alone. We will now show that the same results hold for posterior samples (using the full posterior) with a neural prior. The likelihood remains exactly the same as the one used in the previous section (i.e., from Eq. (25)). Likewise, the sampling procedure is unchanged, only the prior is extended with a neural network as defined in Eq. (23).

Figure 6 illustrates that the power spectra of our `DLPosterior` samples is consistent with the ground truth κTNG simulated convergence map power spectrum, validating that we are able to sample maps with the correct statistics. This is also apparent from visual inspection, and posterior samples shown in Fig. 7 are very similar to convergence maps from the κTNG simulations, illustrating that we are able to sample κ maps at the simulation resolution. In addition, Fig. 6 demonstrates that the prior choice has a strong influence on the map statistics. Using the Gaussian prior alone leads to sampling a map that matches the fiducial power spectrum, which has more power at high ℓ due to the limited resolution of the simulation.

Convergence maps sampled from the full posterior are shown in the second row of Fig. 7. One can observe two regimes in these samples. On one hand, sampling in the unobserved regions (outside of the white boundary of the survey) is only driven by the prior. These regions therefore have high variance between samples because of the different chain initialization maps. On the other hand, in the data region, the sampling is driven by both the likelihood and the prior. Therefore, there is a high correlation between the samples, and with the ground truth convergence map, within the white contours.

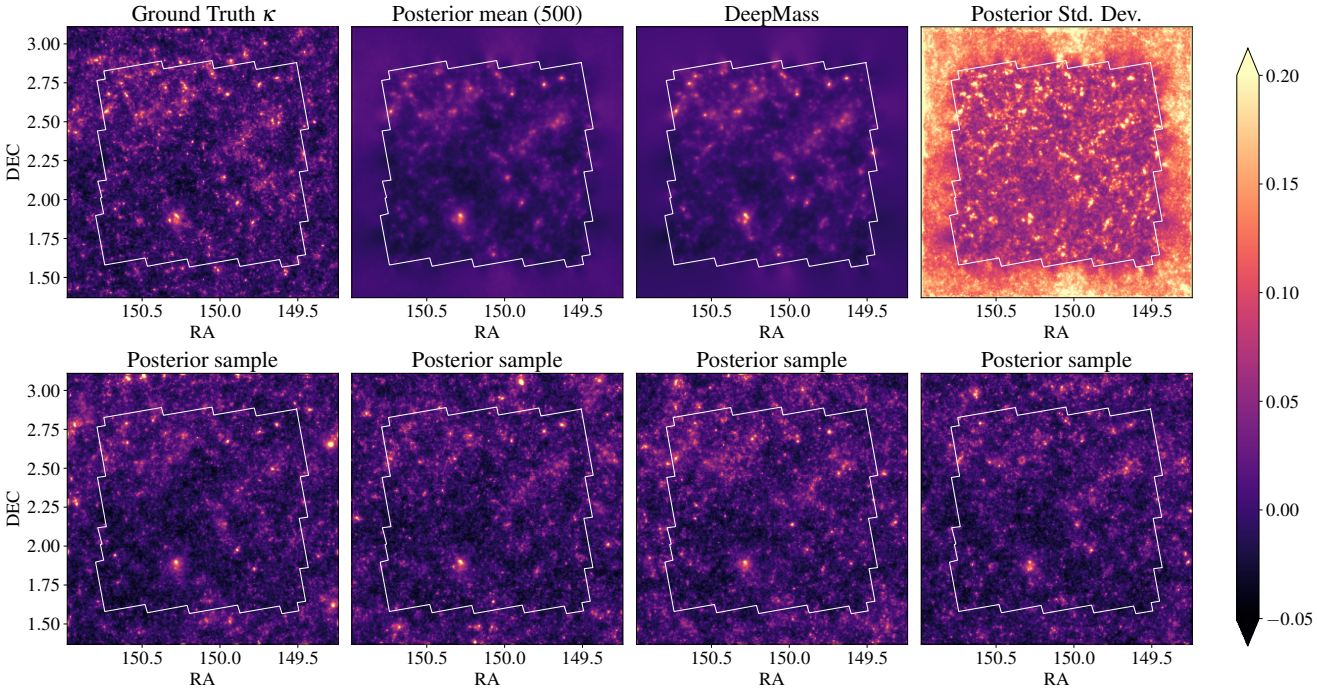


Fig. 7. Comparison of deep learning-based mass-mapping methods on one mock COSMOS field. First row: ground truth κTNG convergence map, the mean of our posterior samples (over 400 samples), DeepMass, and the standard deviation of our posterior samples. All the maps use the same color bar, except for the standard deviation, which is displayed in the range $[0, 0.035]$. Second row: samples from the posterior distribution, using the hybrid Gaussian-neural score prior. The input noise level is computed according Eq. (26) and the mask contours correspond to the COSMOS survey boundary mask.

7.2.2. Posterior mean

As for the Wiener filter, a way to validate the posterior distribution is to look at the posterior mean solution. In this section, we compare to the state-of-the-art method to date, `DeepMass`, which is another deep-learning based method. We trained `DeepMass` on examples drawn from our prior and likelihood model with the exact same U-net architecture as our denoiser. Using the convergence maps from our simulated dataset, we created mock noisy shear maps according to the likelihood in Eq. (25). `DeepMass` computes the posterior mean solution (Sect. 3.4), which can be recovered by averaging the posterior samples from our procedure, (i.e., $\int \kappa p(\kappa|\gamma) \mathrm{d}\kappa \approx \sum_{\kappa \sim p(\kappa|\gamma)} \kappa$). Even if `DeepMass` learns both the prior and the likelihood, the two posterior means should match since both methods were learned on the same simulations.

A qualitative comparison of the method is discussed in Fig. 7. Table 1 shows a quantitative comparison of the Kaiser–Squires, the Wiener Filter, `MCALens`, `GLIMPSE`, `DeepMass`, and our mass-mapping method, based on two metrics. The first metric is the root mean square error (RMSE), defined as:

$$\mathrm{RMSE}\left(\hat{\kappa}, \kappa^{\mathrm{gt}}\right) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\hat{\kappa}_i - \kappa_i^{\mathrm{gt}}\right)^2}, \tag{27}$$

where i is the pixel index, $\hat{\kappa}$ the mean-subtracted estimated convergence map, κ^{gt} the mean-subtracted ground truth convergence map we aim to recover. It should be noted that we use the mean-subtracted convergence map here. The reason for this is the mass-sheet degeneracy, which does not constrain the mean of the convergence map.

Table 1. Metrics comparison between different mass-mapping methods.

Method	RMSE ↓	r ↑
KS (σ_{smooth} = 1 arcmin)	2.40×10^{-2}	0.57
Wiener filter	2.31×10^{-2}	0.61
`GLIMPSE` (3)	2.84×10^{-2}	0.42
`MCAlens` (5)	2.19×10^{-2}	0.67
`DeepMass`	$\mathbf{2.18 \times 10^{-2}}$	**0.68**
`DLPosterior` mean	$\mathbf{2.16 \times 10^{-2}}$	**0.68**

Notes. RMSE (the lower the better) is computed according to Eq. (27) and the Pearson Correlation coefficient r (the higher the better) according to Eq. (28). For a fair comparison to Kaiser–Squires, we kept the smoothing coefficient that minimized the RMSE.

The second metric is the Pearson correlation coefficient r, defined as:

$$r\left(\hat{\kappa}, \kappa^{\mathrm{gt}}\right) = \frac{\mathrm{Cov}\left(\hat{\kappa}, \kappa^{\mathrm{gt}}\right)}{\sigma_{\hat{\kappa}} \sigma_{\kappa^{\mathrm{gt}}}}, \tag{28}$$

where Cov is the covariance and σ_κ is the standard deviation of the convergence map κ.

In the top row of figure Fig. 7, we display our posterior mean against `DeepMass`, the ground truth, and the Wiener filter. Qualitatively, we can clearly observe that both `DeepMass` and our posterior mean recover the convergence map with striking visual similarity, and at a much higher resolution than the Wiener filter applied to the same field (see Fig. 5). We can indeed identify the presence of clusters, which correspond to true clusters in the ground-truth map. For further visual comparison, Fig. G.1 shows Kaiser–Squires, Wiener filter, `MCALens`, `GLIMPSE`, `DeepMass`,

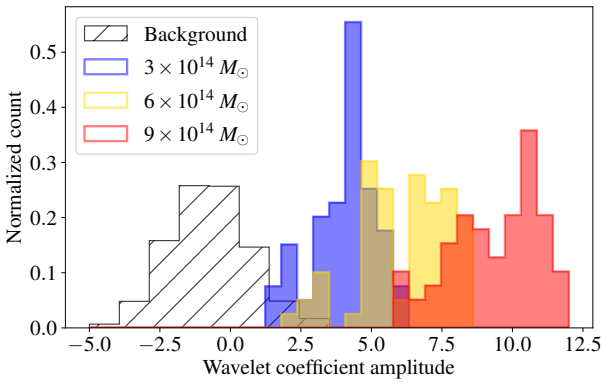


Fig. 8. Histograms of coefficient amplitudes. A wavelet coefficient amplitude corresponds to the coefficient computed in Eq. (30). Background corresponds to the coefficient of samples where we did not add any cluster. The background coefficients follow a Gaussian distribution, and we can thus define a standard deviation σ for detection. We can see that the lower the cluster mass, and thus the lower the S/N, the closer the histograms are to the background.

and our posterior mean and sample convergence estimates from the same input shear field.

Just as in the case of the Gaussian prior explored in the previous section, one can observe that computing the mean of the posterior samples cancels out the signal outside the survey mask, where the reconstruction is only constrained by the prior. This is expected since sampling with a random initialization yields a random sample on the posterior. Moreover, this is confirmed by the standard deviation map, showing that the uncertainty of the reconstruction is much higher in masked regions than in regions constrained by data.

Quantitatively, we also compare our reconstruction to methods based on other priors, which are `MCAlens` and `GLIMPSE`. These methods are based on sparse priors, which are described in Sect. 3.3. It turns out that `MCAlens`, `DeepMass`, and our posterior mean all reach the lowest RMSEs and highest Pearson correlations. This results are expected since both the `DeepMass` and `DLPosterior` mean are data-driven. `DeepMass` is explicitly trained to minimize the pixel reconstruction error, leading to an estimate of the posterior mean, which our method is also targeting. This result shows that our method reaches the state-of-the-art reconstruction of convergence maps.

The fact that `MCAlens` leads to metrics similar to the `DeepMass` and `DLPosterior` mean could be explained by the fact that `MCAlens` modeling (the convergence map being modeled as the sum of a Gaussian random field and a non-Gaussian one being sparse in the wavelet domain) is able to capture well enough the main properties of the convergence map.

7.3. Demonstration with cluster detection

In a Bayesian perspective, we expect that there is a relation between the number of times a cluster appears within posterior samples and its S/N. In this section, we show the extent to which having access to posterior samples allows us to quantify the uncertainty of a cluster detection and how it relates to the cluster S/N.

In order to compare the occurrence of a cluster in the posterior and its S/N, we insert a Navarro–Frenk–White (NFW) lensing signal, described in Navarro et al. (1997) and Takada & Jain (2003), of a given mass, redshift, and concentration, into a mock shear field. We consider a cluster at redshift $z_h = 0.5$, concentration $c = 1$, lensing a source at redshift s = 1. We run the experiments at different halo masses $\{9 \times 10^{13}, 3 \times 10^{14}, 6 \times 10^{14}\}\ M_\odot$, simulating an S/N increase for a given noise level in the data. We used the code from the `lenspack`[2] repository to build the NFW shear map, which is summed to the mock shear field from our test dataset.

In this section, all the maps are filtered with an aperture mass filter with starlets functions (Leonard et al. 2012). Starlets are well localized in real space, and thus it is well very adapted to cluster detection (Starck et al. 1998). In the following, we either use pixels or coefficients to refer to pixels of the filtered convergence maps.

Therefore, for a given NFW cluster, its intrinsic S/N, which we also call the input S/N, is defined as

$$\mathrm{S/N}_{\mathrm{input}} = \frac{\max\limits_{i,j} \check{\kappa}_{\mathrm{NFW}}[i,j]}{\mathrm{Std}\left(\check{\kappa}_{\mathrm{background}}\right)}, \tag{29}$$

where $\check{\kappa}_{\mathrm{NFW}}$ denotes the convergence NFW profile that has been filtered and the maximum is taken over the pixels of the map, $\check{\kappa}_{\mathrm{background}}$ is the background convergence map, namely the mock convergence field over which we add the NFW profile that is being filtered. The mock convergence field is computed from COSMOS-like data, similar to the fields described in Sect. 6.3.

We now describe the detection procedure. Given a shear field, we sample convergence maps with our method and each posterior sample is filtered with the aperture mass. Given one posterior sample, the detection criteria defined as:

$$\mathrm{Detection} = \begin{cases} \text{True if} & \dfrac{\max\limits_{i,j} \check{\kappa}_{\text{post. sample}}[i,j]}{\mathrm{Std}\left(\check{\kappa}_{\mathrm{background}}\right)} > 3, \\ \text{False} & \text{otherwise,} \end{cases} \tag{30}$$

which means that we consider that a cluster is detected when the convergence is over three times the noise background amplitude. The distribution of the convergence background is illustrated as the hatched histogram in Fig. 8. Similar histograms were computed from posterior samples with an added cluster in the input shear and plotted next to the background. In Fig. 8, we can see that when decreasing the input S/N, by reducing the halo mass, the distribution of the cluster maximum coefficient shifts toward lower levels, approaching the background distribution. The background distribution histogram in hatched black in Fig. 8 is the histogram of the $\check{\kappa}_{\mathrm{background}}$ map. Figure F.1 shows stamp coefficients after filtering.

Figure 9 shows the frequency of the cluster detection in our posterior samples as a function of the input S/N.

8. Reconstruction of the COSMOS field

In this section we apply our full methodology to the reconstruction of the COSMOS field, using the catalog described in Sect. 6.1. The likelihood convariance and the simulation-based prior remain the same as in the validation section.

As a baseline, we show the Kaiser–Squires reconstruction of the COSMOS convergence field in the top left panel of Fig. 10. We applied a Gaussian-smoothing with a variance $\sigma_{\mathrm{smooth}} = 1$ arcmin, chosen so that the RMSE is minimized and the Pearson correlation coefficient is maximized for simulated data. Although large-scale and small-scale structures can already be observed on this map, its power spectrum does not correspond to the fiducial matter power spectrum, and no feature at scales

[2] https://github.com/CosmoStat/lenspack

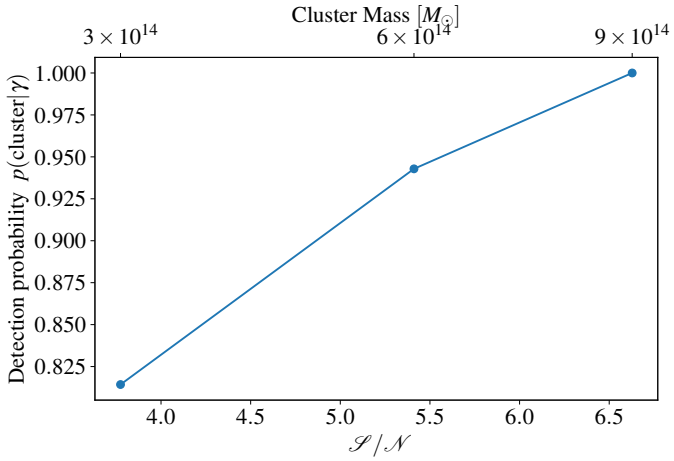


Fig. 9. Comparison of the probability of cluster detection given the input data and the detection method with respect to the S/N. The ratio of detection corresponds the number of samples above 5σ over the total number of samples. The S/N was computed by dividing the maximum κ value of the NFW profile by the noise in the κ field. We considered clusters at redshift $z_h = 0.5$ and concentration $c = 1$ varying the halo mass.

smaller that 1 arcmin can be observed. Moreover, there is not any uncertainty quantification on the reconstruction.

In Fig. 10 we also present our reconstruction of the COSMOS convergence field, alongside uncertainty quantification. The top-right panel shows a posterior sample that looks very similar to a posterior sample from simulated input from κTNG simulations. It can be noticed that, although there is only input shear within the white contours (i.e., in the survey footprint), a complete convergence map is sampled from the posterior distribution. We furthermore validate, in Fig. 11, the quality of COSMOS posterior sample reconstruction by showing that their power spectra are in good agreement with those of κTNG convergence maps.

As proposed in Sect. 3.4, we chose the posterior mean as our estimate for the convergence reconstruction. The bottom left panel of Fig. 10 shows the average of 400 samples making the `DLPosterior` mean of the COSMOS field. One can visually observe the similar large-scale structures of the field between the Kaiser–Squires and the `DLPosterior` mean, while the latter contains much more resolved features, especially concerning the cluster shapes.

Another way to examine our COSMOS convergence map is to compare it to another probe for cluster mass detection. Thus, in the bottom left and top right panels, we overlay a subset of X-ray clusters from the Finoguenov et al. (2007) catalog. Most of the X-ray clusters match to a resolved peak in the convergence field, which is, in a way, expected, since we selected the most massive X-ray clusters.

Alongside the `DLPosterior` mean, we also provide the `DLPosterior` standard deviation, in the bottom right panel of Fig. 10. One can observe again that the variance is the highest outside the survey contours, since there is no data. Another interesting behavior of the posterior is that the location where the uncertainty is also high is where the signal is the most intense.

The most recent mass-map of the COSMOS field was published in Massey et al. (2007), using an generalized version of the Kaiser–Squires method. When comparing the two maps, although we do not share the same shape catalog or redshift distribution, the maps clearly share similar mass distributions. However, the COSMOS-`DLPosterior` mean is much more resolved. In particular, we can identify several clusters in what looks like one coarse cluster in the Massey et al. (2007) mass-map at coordinates RA = 150.4, Dec = 2.5.

9. Discussion

Having presented the method and results, in this section we propose a discussion on some important points, including scientifically relevant use-cases, limitations, and possible extensions.

As mentioned earlier in this paper, taking the average over posterior samples should reduce to the `DeepMass` results. One may wonder about the tradeoff between the two approaches if the results should be the same. We would argue that the two methods are complementary. `DeepMass` is much faster at producing a convergence map, as it only requires a forward pass of the U-net. However, `DeepMass` remains an "amortized" solution, with no strong guarantees on the solution it recovers in practice, as it does not have an explicit likelihood. This also means that the entire model needs to be retrained for any change in the lensing catalog (to account for variations in the mask or noise).

We expect that the method presented in this paper will find its most compelling applications in the study of localized structures through the weak lensing effect, where the benefits of a full pixel-level posterior are the strongest. As a particularly relevant example, we mention the discussion surrounding the potential detection in the Abell 520 (A520) cluster of a dark clump (Jee et al. 2012; Clowe et al. 2012), a localized peak visible in mass-maps but with no optical counterpart. Quantifying the significativity of such a structure using nonlinear mass-mapping algorithms targeting a MAP solution (for instance, based on sparse regularization) is a difficult task. It was attempted for this particular field using different techniques in Peel et al. (2017) and Price et al. (2018), but without strong quantitative statements. The method developed in this paper, however, would be able to access the full posterior of the problem.

For cosmological applications, however, it is likely that the use cases of the method presented in this work will remain limited. Indeed, for higher-order statistics that rely on simulations to evaluate their likelihood, the particular mass-mapping technique used is not critical, as any systematics due to reconstruction errors induced by the algorithm are calibrated on simulations. A simple Kaiser–Squires inversion should, in principle, suffice in most cases of interest. From an information theoretic point of view, the information is preserved by a Kaiser–Squires inversion (in the presence of masks, keeping both E- and B-modes) while any posterior summary may discard some cosmological information. Nevertheless, the ability to sample constrained realizations may find very useful applications, such as inpainting masked regions for the purpose of facilitating the computation of 3pt functions, or void detection algorithms.

We also want to highlight that the method presented in this paper can be extended in multiple directions. In this configuration we would be limited to analyzing small-volume surveys. However, we could train a prior on spherical maps, having a spherical likelihood, and run analysis and then run large-volume survey analyses. We are considering here a prior at a fixed cosmology, primarily due to the high computational cost of high-resolution hydrodynamical simulations at different points in cosmological parameter space. Concretely, this means that our solution is heavily biased toward the fiducial cosmology on scales poorly constrained by the data. We note, however, that the hybrid deep learning + physical gaussian prior we have introduced in this work provides a natural framework

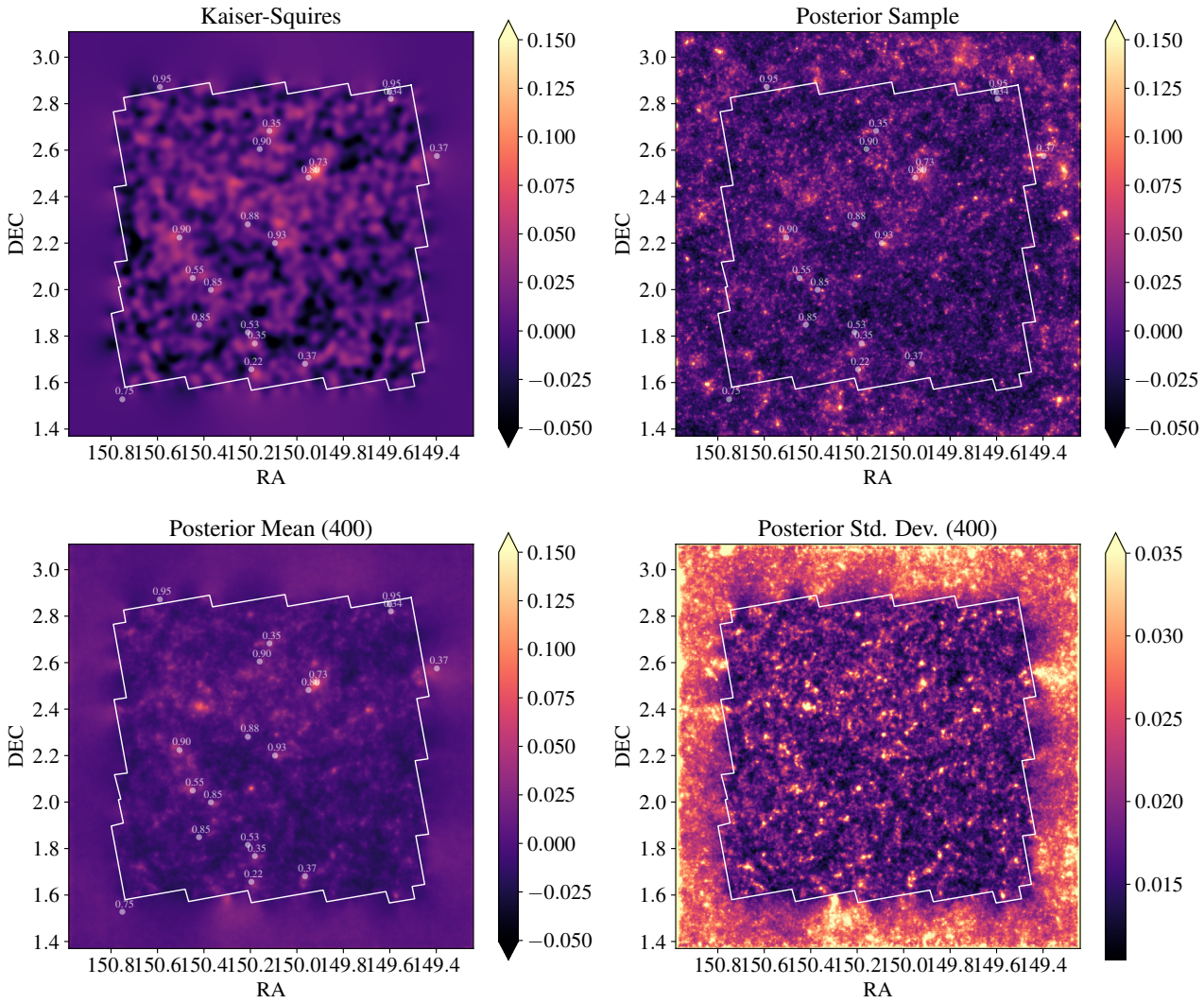


Fig. 10. HST/ACS COSMOS field reconstructions along known X-ray clusters from the *XMM-Newton* survey. Top left: Kaiser–Squires (with a Gaussian smoothing of $\sigma = 1$ arcmin). Top right: sample from the posterior distribution. Bottom left: mean of the posterior distribution. Bottom right: standard deviation of the posterior distribution (over 400 samples, shown in the clipped range [0, 0.035]).

for extending our method to include cosmological dependence. On large scales, the prior is mostly driven by the analytic power spectrum, and thus easy on condition on cosmology, while on small scales, only the non-Gaussian residuals are captured by the neural network. This implies that the cosmology-dependent part of the model that needs to be learned from simulations is mostly on small scales. This makes it very likely that in the close future, one could develop such a cosmology-dependent residual prior from suites of numerical simulations of smaller cosmological volume, but spanning a range of cosmological models. The recent multifield CAMELS simulations (Villaescusa-Navarro et al. 2022) would be an ideal dataset for this purpose. Another avenue for future work would be extending the method to the sphere, which is simply a matter of defining a U-net, for instance using a DeepSphere (Perraudin et al. 2019) approach for convolutions on a spherical domain.

10. Conclusion

In this paper, we have presented a unified view of the mass-mapping problem as a Bayesian posterior estimation problem. Most existing methods either rely on simple priors (i.e., a Gaussian prior for the Wiener filter) and/or only return a point estimate of the mass-mapping posterior. Instead, we have proposed a framework that allows us to: (1) use numerical simulations to provide a full non-Gaussian prior on the convergence field; and (2) sample from the full Bayesian posterior of the mass-mapping problem under this simulation-driven prior and a physical likelihood.

The proposed approach, dubbed `DLPosterior`, relies first on using a DSM technique to learn a prior from high-resolution hydrodynamical simulations (the κTNG dataset; Osato et al. 2021), an approach that has proven to be extremely scalable and easy to implement. And as a second ingredient, we have introduced an annealed HMC approach that allows us to sample the high-dimensional Bayesian posterior of the problem with high efficiency. We demonstrate that we are able to achieve, on average, an independent posterior sample in 10 GPU minutes on an Nvidia Tesla V100 GPU. Moreover, the annealing scheme ensures that we sample independent samples with the correct weights of the posterior modes.

We first validated the sampling approach in an analytically tractable, fully Gaussian case, where we recovered the Wiener filter. We then validated the entire method over mock

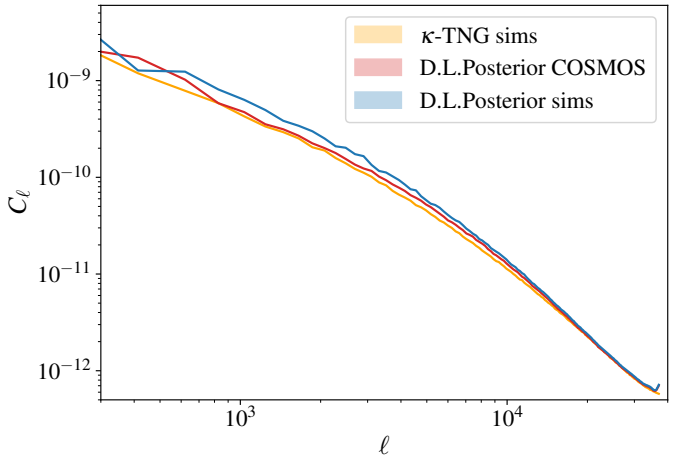


Fig. 11. Power spectra comparison between the COSMOS field reconstruction and simulations. We compare power spectra of posterior samples for the COSMOS field in blue and for a simulated field in red. We also display the power spectrum of samples from the training set in orange to show the matching.

observations, by comparing the posterior mean obtained by our method to a deep learning estimate of this posterior mean, using the `DeepMass` method. We found excellent agreement between these two independently estimated posterior summaries, with near identical Pearson correlation coefficients and RMSEs. This consistency gives us high confidence that the full posterior is correctly sampled even in the non-Gaussian case.

In further comparisons, `DLPosterior`, demonstrated a quantitative improvement on convergence reconstructions against other standard methods, based on a large class of priors such as the Kaiser–Squires inversion, the Wiener filter, or `GLIMPSE2D`. In addition, contrary to most of these methods, `DLPosterior` provides uncertainty quantification with the posterior samples, such as the posterior variance. In this respect, we have also shown that the recovered posterior can be interpreted to quantify uncertainties, by establishing a close correlation between posterior convergence values and the S/N for clusters that are artificially introduced into a field.

Finally, we applied the validated method to the reconstruction of the HST/ACS COSMOS field based on the shape catalog from Schrabback et al. (2010), and produced the highest-quality convergence map of this field to date.

In the spirit of reproducible and reusable research, all software, scripts, and analysis notebooks are publicly available[3].

Acknowledgements. The authors are grateful Zaccharie Ramzi for fruitful discussions on inverse problems and very helpful discussions on neural network architecture. Benjamin Remy acknowledges support by the Centre National d'Études Spatiales and the Université Paris-Saclay Doctoral Program in Artificial Intelligence (UDOPIA). KO is supported by JSPS Research Fellowships for Young Scientists. This work was supported by Grant-in-Aid for JSPS Fellows Grant Number JP21J00011. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF Grant No. ACI-1053575. This work was granted access to the HPC resources of IDRIS under the allocation 2021-AD011011554R1 made by GENCI (Grand Équipement National de Calcul Intensif). We thank New Mexico State University (USA) and Instituto de Astrofisica de Andalucia CSIC (Spain) for hosting the Skies and Universes site for cosmological simulation products. Software: Astropy (Astropy Collaboration 2013, 2018), IPython (Perez & Granger 2007), Jupyter (Kluyver et al. 2016), Matplotlib (Hunter 2007), Numpy (Harris et al. 2020), TensorFlow Probability (Dillon et al. 2017), JAX (Bradbury et al. 2018).

[3] https://github.com/CosmoStat/jax-lensing

References

Ajani, V., Starck, J.-L., & Pettorino, V. 2021, A&A, 645, L11
Alain, G., & Bengio, Y. 2013, in 1st International Conference on Learning Representations, ICLR 2013 - Conference Track Proceedings, 15, 3743
Alsing, J., Heavens, A., Jaffe, A. H., et al. 2016, MNRAS, 455, 4452
Anderson, B. D. O. 1982, Stochast. Process. Appl., 12, 313
Astropy Collaboration (Robitaille, T. P., et al.) 2013, A&A, 558, A33
Astropy Collaboration (Price-Whelan, A. M., et al.) 2018, AJ, 156, 123
Bartelmann, M. 2010, Class. Quant. Grav., 27, 233001
Betancourt, M. 2017, ArXiv e-prints [arXiv:1701.02434]
Bobin, J., Starck, J. L., Sureau, F., & Fadili, J. 2012, Adv. Astron., 2012, 703217
Bradbury, J., Frostig, R., Hawkins, P., et al. 2018, https://github.com/google/jax
Cheng, S., Ting, Y.-S., Ménard, B., & Bruna, J. 2020, MNRAS, 499, 5902
Clowe, D., Markevitch, M., Bradač, M., et al. 2012, ApJ, 758, 128
Dhariwal, P., & Nichol, A. 2021, ArXiv e-prints [arXiv:2105.05233]
Dillon, J. V., Langmore, I., Tran, D., et al. 2017, ArXiv e-prints [arXiv:1711.10604]
Elsner, F., & Wandelt, B. D. 2013, A&A, 549, A111
Erben, T., Van Waerbeke, L., Bertin, E., Mellier, Y., & Schneider, P. 2001, A&A, 366, 717
Fiedorowicz, P., Rozo, E., Boruah, S. S., Chang, C., & Gatti, M. 2022, MNRAS, 512, 73
Finoguenov, A., Guzzo, L., Hasinger, G., et al. 2007, ApJS, 172, 182
Girolami, M., & Calderhead, B. 2011, J. R. Statist. Soc.: Ser. B (Statist. Methodol.), 73, 123
Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al. 2014, ArXiv e-prints [arXiv:1406.2661]
Gouk, H., Frank, E., Pfahringer, B., & Cree, M. J. 2020, Mach. Learn., 110, 393
Harnois-Déraps, J., Martinet, N., Castro, T., et al. 2021, MNRAS, 506, 1623
Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Nature, 585, 357
Hastings, W. K. 1970, Biometrika, 57, 97
He, K., Zhang, X., Ren, S., & Sun, J. 2016, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770
Ho, J., Jain, A., & Abbeel, P. 2020, Adv. Neural Inf. Process. Syst., 33, 6840
Hoekstra, H., Franx, M., Kuijken, K., & Squires, G. 1998, ApJ, 504, 636
Horowitz, B., Seljak, U., & Aslanyan, G. 2019, J. Cosmol. Astropart. Phys., 10, 035
Hunter, J. D. 2007, Comput. Sci. Eng., 9, 90
Ilbert, O., Capak, P., Salvato, M., et al. 2009, ApJ, 690, 1236
Ivezić, Z., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111
Jee, M. J., Mahdavi, A., Hoekstra, H., et al. 2012, ApJ, 747, 96
Jeffrey, N., Abdalla, F. B., Lahav, O., et al. 2018a, MNRAS, 479, 2871
Jeffrey, N., Heavens, A. F., & Fortio, P. D. 2018b, Astron. Comput., 25, 230
Jeffrey, N., Alsing, J., & Lanusse, F. 2020a, MNRAS, 501, 954
Jeffrey, N., Lanusse, F., Lahav, O., & Starck, J.-L. 2020b, MNRAS, 492, 5023
Jimenez Rezende, D., Mohamed, S., & Wierstra, D. 2014, ArXiv e-prints [arXiv:1401.4082]
Jolicoeur-Martineau, A., Piché-Taillefer, R., des Combes, R. T., & Mitliagkas, I. 2020, ArXiv e-prints [arXiv:2009.05475]
Kacprzak, T., Kirk, D., Friedrich, O., et al. 2016, MNRAS, 463, 3653
Kaiser, N., & Squires, G. 1993, ApJ, 404, 441
Kaiser, N., Squires, G., & Broadhurst, T. 1995, ApJ, 449, 460
Kilbinger, M. 2015, Rep. Progr. Phys., 78, 086901
Kingma, D. P., & Welling, M. 2013, ArXiv e-prints [arXiv:1312.6114]
Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. 2016, Jupyter Notebooks - a Publishing Format for Reproducible Computational Workflows, 87
Kurita, T., Takada, M., Nishimichi, T., et al. 2021, MNRAS, 501, 833
Lahav, O., Fisher, K. B., Hoffman, Y., Scharf, C. A., & Zaroubi, S. 1994, ApJ, 423, L93
Lanusse, F., Starck, J. L., Leonard, A., & Pires, S. 2016, A&A, 591, A2
Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, ArXiv e-prints [arXiv:1110.3193]
Lecun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F.-J. 2006, A Tutorial on Energy-Based Learning (New York: MIT Press)
Leonard, A., Dupé, F.-X., & Starck, J.-L. 2012, A&A, 539, A85
Lim, J. H., Courville, A., Pal, C., & Huang, C.-W. 2020, ArXiv e-prints [arXiv:2006.05164]
Liu, J., Petri, A., Haiman, Z., et al. 2015a, Phys. Rev. D, 91, 063507
Liu, X., Pan, C., Li, R., et al. 2015b, MNRAS, 450, 2888
Marinacci, F., Vogelsberger, M., Pakmor, R., et al. 2018, MNRAS, 480, 5113
Marshall, P. J. 2001, Clusters of galaxies and the high redshift universe observed in X-rays, Recent results of XMM-Newton and Chandra, XXXVIth Rencontres de Moriond , XXIst Moriond Astrophysics Meeting, March 10-17, 2001 Savoie, France, eds. D. M. Neumann, J. T. T. Van, http://moriond.in2p3.fr, 47

Martinet, N., Schneider, P., Hildebrandt, H., et al. 2018, MNRAS, 474, 712
Massey, R., Rhodes, J., Ellis, R., et al. 2007, ApJS, 172, 239
Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. 1953, J. Chem. Phys., 21, 1087
Naiman, J. P., Pillepich, A., Springel, V., et al. 2018, MNRAS, 477, 1206
Navarro, J. F., Frenk, C. S., & White, S. D. M. 1997, ApJ, 490, 493
Neal, R. M. 2011, Handbook of Markov Chain Monte Carlo, 113
Nelson, D., Pillepich, A., Springel, V., et al. 2018, MNRAS, 475, 624
Nelson, D., Springel, V., Pillepich, A., et al. 2019, Comput. Astrophys. Cosmol., 6, 2
Nichol, A., & Dhariwal, P. 2021, ArXiv e-prints [arXiv:2102.09672]
Osato, K., Liu, J., & Haiman, Z. 2021, MNRAS, 502, 5593
Peel, A., Lanusse, F., & Starck, J.-L. 2017, ApJ, 847, 23
Perez, F., & Granger, B. E. 2007, Comput. Sci. Eng., 9, 21
Perraudin, N., Defferrard, M., Kacprzak, T., & Sgier, R. 2019, Astron. Comput., 27, 130
Pillepich, A., Nelson, D., Hernquist, L., et al. 2018, MNRAS, 475, 648
Planck Collaboration XIII 2015, A&A, 594, A13
Porqueres, N., Heavens, A., Mortlock, D., & Lavaux, G. 2022, MNRAS, 509, 3194
Price, M. A., Cai, X., McEwen, J. D., Pereyra, M., & Kitching, T. D. 2018, MNRAS, 489, 3236
Remy, B., Lanusse, F., Ramzi, Z., et al. 2020, ArXiv e-prints [arXiv:2011.08271]
Ribli, D., Pataki, B. Á., & Csabai, I. 2019, Nat. Astron., 3, 93
Ronneberger, O., Fischer, P., & Brox, T. 2015, ArXiv e-prints [arXiv:1505.04597]
Salimans, T., Karpathy, A., Chen, X., & Kingma, D. P. 2017, ArXiv e-prints [arXiv:1701.05517]
Schneider, M. D., Ng, K. Y., Dawson, W. A., et al. 2016, ApJ, 839, 25
Schrabback, T., Erben, T., Simon, P., et al. 2007, A&A, 468, 823
Schrabback, T., Hartlap, J., Joachimi, B., et al. 2010, A&A, 516, A63
Scoville, N., Abraham, R. G., Aussel, H., et al. 2007, ApJS, 172, 38
Shan, H., Liu, X., Hildebrandt, H., et al. 2018, MNRAS, 474, 1116
Shi, J., Kurita, T., Takada, M., et al. 2021, J. Cosmol. Astropart. Phys., 2021, 030
Shirasaki, M., Moriwaki, K., Oogi, T., et al. 2021, MNRAS, 504, 1825
Simon, P., Heymans, C., Schrabback, T., et al. 2012, MNRAS, 419, 998
Smith, M. J., Geach, J. E., Jackson, R. A., et al. 2022, MNRAS, 511, 1808
Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. 2015, ArXiv e-prints [arXiv:1503.03585]
Song, Y., & Ermon, S. 2019, ArXiv e-prints [arXiv:1907.05600]
Song, Y., & Ermon, S. 2020, ArXiv e-prints [arXiv:2006.09011]
Song, Y., Garg, S., Shi, J., & Ermon, S. 2019, ArXiv e-prints [arXiv:1905.07088]
Song, Y., Sohl-Dickstein, J., Kingma, D. P., et al. 2020, ArXiv e-prints [arXiv:2011.13456]
Spergel, D., Gehrels, N., Baltay, C., et al. 2015, ArXiv e-prints [arXiv:1503.03757]
Springel, V., Pakmor, R., Pillepich, A., et al. 2018, MNRAS, 475, 676
Starck, J.-L., Murtagh, F., & Bijaoui, A. 1998, Image Processing and Data Analysis: The Multiscale Approach (Cambridge: Cambridge University Press)
Starck, J.-L., Pires, S., & Réfrégier, A. 2006, A&A, 451, 1139
Starck, J. L., Themelis, K. E., Jeffrey, N., Peel, A., & Lanusse, F. 2021, A&A, 649, A99
Takada, M., & Jain, B. 2003, ApJ, 583, L49
van den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. 2016, ArXiv e-prints [arXiv:1601.06759]
Villaescusa-Navarro, F., Genel, S., Angles-Alcazar, D., et al. 2022, ApJS, 259, 61
Vincent, P. 2011, Neural Comput., 23, 1661
Yiu, T. W. H., Fluri, J., & Kacprzak, T. 2022, J. Cosmol. Astropart. Phys., 12, 013
Zaroubi, S., Hoffman, Y., Fisher, K. B., & Lahav, O. 1995, ApJ, 449, 446

Appendix A: Score-based Metropolis-Hastings

The discretized implementation of the HMC algorithm requires correcting for the discretization error. The chain update computed with Equation (21) is a proposal that is accepted with probability of the form $\min\{1, p(x_*)q(x_n|x_*)/p(x_n)q(x_*|x_n)\}$, where x_n is the last sample of the chain, x_* is the HMC proposal, p is the target density, and q is a proposal distribution from a random walk (i.e., $q(x_n|x_*) = \mathcal{N}(x_n|x_*, \mathbf{M})$ with $\mathbf{M}$ the HMC preconditionig matrix).

In our approach, we do not directly have access to the distribution p, but to its score function $\nabla \log p$. However, we can still approximate the log ratio needed to compute the MH acceptance probability from the scores using the path integral

$$\log p(x_*) - \log p(x_n) = \int_0^1 \nabla_x \log p(t*(x_*-x_n)+x_n)\cdot(x_*-x_n)dt, \tag{A.1}$$

which we evaluate in practice with a simple four points Simpson integration rule. This integral could be approximated to any precision at the cost of additional score evaluations.

Thus, with Equation A.1 at hand, we are able to implement a large class of MCMC algorithm, such as the HMC or the Metopolis adjusted Langevin algorithm, using the score function only.

Appendix B: Sampling by solving a stochastic differential equation

Recent work from Song et al. (2020) improves annealed sampling procedures by generalizing the process as an SDE. An SDE has the following form:

$$d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}, \tag{B.1}$$

where $f(\cdot, t) : \mathbb{R}^d \to \mathbb{R}$ is a vector valued function, $g(\cdot) : \mathbb{R} \to \mathbb{R}$ is a scalar function, and $\boldsymbol{w}$ is a Wiener process. This equation models a diffusion process $\{\boldsymbol{x}(t)\}_{t=0}^T$ indexed by a continuous time variable $t \in [0, T]$, such that $\boldsymbol{x}(0) \sim p_0$ and $\boldsymbol{x}(T) \sim p_T$, where p_0 denotes the data distribution, and p_T denotes the convolution between the data distribution and a multivariate Gaussian of variance T. The variance T being much larger than the support size of the data distribution, p_T can be seen as a wide multivariate Gaussian.

A result from Anderson (1982) shows that one can reverse the SDE in Equation B.1, starting from samples $\boldsymbol{x}(T) \sim p_T$ and obtaining samples $\boldsymbol{x}(0) \sim p_0$, by computing the reverse-time SDE, involving the score function:

$$d\boldsymbol{x} = \left[f(\boldsymbol{x}, t) - g(t)^2\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right]dt + g(t)d\bar{\boldsymbol{w}}, \tag{B.2}$$

where $\bar{\boldsymbol{w}}$ is a Wiener process with backward time, from T to 0.

Song et al. (2020) stress that the annealed LD is a discretized version of an SDE. Indeed, the chain update of the annealed LD is:

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\boldsymbol{z}_{i-1}, \tag{B.3}$$

with $i \in \{1, \dots, N\}$, where the $\{\sigma_i\}_{i=1}^N$ is the increasing variance of the Gaussian with which we convolve the data distribution, also called the temperature, and z_i are multivariate Gaussian realization. Then if $N \to \infty$, the temperature becomes a continuous function $\sigma(t)$ and $\boldsymbol{z}_i$ becomes a Gaussian process $\boldsymbol{w}(t)$. Thus, the process $\{\boldsymbol{x}(t)\}_{t=0}^T$ is given by the SDE:

$$d\boldsymbol{x} = \sqrt{\frac{d}{dt}\left(\sigma^2(t)\right)}d\boldsymbol{w}. \tag{B.4}$$

Finally, Song et al. (2020) demonstrate that for any reverse-time SDE expressed as in Equation B.2, there is a corresponding deterministic process whose trajectory has the same marginal probability densities $\{p_t(\boldsymbol{x})\}_{t=0}^T$, satisfying an ODE:

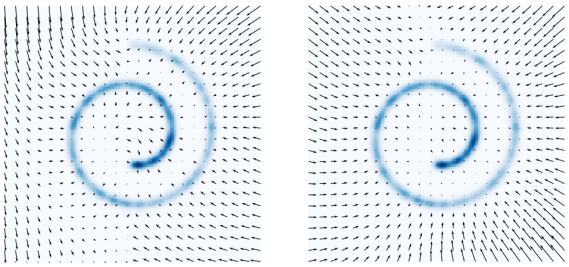
$$d\boldsymbol{x} = \left[f(\boldsymbol{x}, t) - \frac{1}{2}g(t)^2\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right]dt, \tag{B.5}$$

with the exact same notation as in the SDE equation. Thus, in order to sample from the distribution p_0, one can solve the corresponding ODE of the annealed LD:

$$d\boldsymbol{x} = -\frac{1}{2}\sqrt{\frac{d}{dt}\left(\sigma^2(t)\right)}\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})dt, \tag{B.6}$$

using any black-box ODE solver. We notice that, once again, the sampling procedure only depends on the score function. Thus, one can sample from the target distribution, denoted p_0 in this formalism, starting from random samples from a multivariate Gaussian p_T.

Appendix C: U-net architecture



(without SN) (with SN)

Fig. C.1. Comparison of the score function learning with and without spectral normalization. In blue is the density of the swiss roll distribution, and the black vector field is the learned score function evaluated on a grid.

We used a three-scale U-net architecture (Ronneberger et al. 2015) composed of residual blocks (He et al. 2016). Each block is composed of two convolutions followed by a batch normalization and three convolutions for the bottleneck. Each batch normalization is followed by a rectified linear unit (ReLU) nonlinearity, except the last one. As densigned in Ronneberger et al. (2015), we performed downsampling by average pooling and upsampling by interpolation. We used the sequence of channels [32, 64, 128, 128] over the different scales.

Sampling requires evaluating the score function in regions where the network did not necessarily observe data. As described in section 4.2, annealing is useful to smooth the distribution, and thus the associated score. Besides, we investigated the effect of spectral normalization on the network regularity in those regions. Indeed, as it can be seen in Figure C.1, spectral normalization smoothens the learned gradient map far from the high densities. Regularizing the spectral normal of a network lowers its Lipschitz constant, which prevents high variation between close points, thus aligning unconstrained gradients.

An important feature of the network is the noise condition. The input image is concatenated with a noise standard deviation map, as well as the input of the bottleneck of the network. We also followed the advice of Song & Ermon (2020) to divide the output of the network by the absolute value of the noise level. It is observed that the norm of the score function is inversely proportional to the noise power, and neural networks have issues in doing this rescaling automatically. This output normalization turns out to be necessary when we consider a large order of magnitude between the noise powers during annealing.

Appendix D: Convolution of the likelihood

Proposition: let $x \in \mathbb{R}^d$, $y \in \mathbb{R}^d$, $p_{\sigma_1}(y|x) \triangleq \mathcal{N}(y \mid \mathbf{P}x, \sigma_1^2 \mathbf{I}_d)$ be the likelihood and $p_{\sigma_2}(x) \triangleq \mathcal{N}(x \mid 0, \sigma_2^2 \mathbf{I}_d)$ a centered multivariate Gaussian distribution. We assume that the operator $\mathbf{P}$ is unitary (i.e., it verifies $\mathbf{P}^\dagger \mathbf{P} = \mathbf{I}_d$).

Then the convolution of the likelihood with the centered gaussian is:

$$p_{\sigma_1} \circledast p_{\sigma_2}(y|x) = \mathcal{N}(y \mid \mathbf{P}x, (\sigma_1^2 + \sigma_2^2)\mathbf{I}_d). \tag{D.1}$$

Proof: According to the definition of p_{σ_1} and p_{σ_2}, we have:

$$p_{\sigma_1} \circledast p_{\sigma_2}(y|x) = \int p_{\sigma_1}(y|x-t) p_{\sigma_2}(t) dt$$

$$= \frac{1}{(2\pi\sigma_1^2\sigma_2^2)^d} \int \exp\left(-\frac{1}{2\sigma_1^2}\left((y - \mathbf{P}(x-t))^\dagger (y - \mathbf{P}(x-t))\right)\right) \times \exp\left(\frac{t^\dagger t}{2\sigma_2^2}\right) dt,$$

$$= \frac{1}{(2\pi\sigma_1^2\sigma_2^2)^d} \int \exp\left(-\frac{1}{2\sigma_1^2}\Big(\|y - \mathbf{P}x\|_2^2 + 2y^\dagger \mathbf{P}t - 2x^\dagger \mathbf{P}^\dagger \mathbf{P}t + t^\dagger \mathbf{P}^\dagger \mathbf{P}t\Big)\right) \exp\left(\frac{t^\dagger t}{2\sigma_2^2}\right) dt$$

$$= \frac{1}{(2\pi\sigma_1^2\sigma_2^2)^d} \int \exp\left(-\frac{1}{2\sigma_1^2}\Big(\|y - \mathbf{P}x\|_2^2 + 2y^\dagger u - 2x^\dagger \mathbf{P}^\dagger u + u^\dagger u\Big)\right) \exp\left(\frac{t^\dagger t}{2\sigma_2^2}\right) dt.$$

We can use the change of variable $u = \mathbf{P}t$, and we notice that $du = |\det \mathbf{P}| dt = dt$ and $u^\dagger u = t^\dagger \mathbf{P}^\dagger \mathbf{P} t = t^\dagger t$, since $\mathbf{P}$ is unitary, so that:

$$= \frac{1}{(2\pi\sigma_1^2\sigma_2^2)^d} \int \exp\left(-\frac{1}{2\sigma_1^2}\left(\|y - \mathbf{P}x\|_2^2 + 2u^\dagger(y - \mathbf{P}x) + u^\dagger u\right)\right) \times \exp\left(\frac{u^\dagger u}{2\sigma_2^2}\right) du$$

$$= \frac{1}{(2\pi\sigma_1^2\sigma_2^2)^d} \exp\left(-\frac{\|y - \mathbf{P}x\|_2^2}{2\sigma_1^2}\right) \times \int \exp\left(-\frac{u^\dagger(y - \mathbf{P}x)}{\sigma_1^2} - u^\dagger u\left(\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2}\right)\right) du$$

$$= \frac{1}{(2\pi\sigma_1^2\sigma_2^2)^d} \exp\left(-\frac{\|y - \mathbf{P}x\|_2^2}{2\sigma_1^2}\right) \times \int \exp\left(-\frac{1}{2}\left(\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2}\right)\left[\frac{2u^\dagger(y - \mathbf{P}x)}{2\sigma_1^2}\left(\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2}\right)^{-1} + u^\dagger u\right]\right) du$$

$$= \underbrace{\exp\left(-\frac{\|y - \mathbf{P}x\|_2^2}{2\sigma_1^2}\right) \exp\left(\frac{\|y - \mathbf{P}x\|_2^2}{2\sigma_1^4}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1}\right)}_{(*)} \times \underbrace{\int \exp\left(-\frac{1}{2}\left(\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2}\right)\left[u + \underbrace{\frac{y - \mathbf{P}x}{2\sigma_1}\left(\left(\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2}\right)^{-1}\right)}_{\triangleq \mu}\right]^2\right) du}_{(**)}$$

$$= \frac{(2\pi)^{d/2}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{d/2}}{(2\pi\sigma_1^2\sigma_2^2)^d} \exp\left(-\frac{\|y - \mathbf{P}x\|_2^2}{2(\sigma_1^2 + \sigma_2^2)}\right)$$

$$= (2\pi)^{-d/2}(\sigma_1^2 + \sigma_2^2)^{-d/2} \exp\left(-\frac{\|y - \mathbf{P}x\|_2^2}{2(\sigma_1^2 + \sigma_2^2)}\right)$$

$$= (2\pi)^{-d/2}(\sigma_1^2 + \sigma_2^2)^{-d/2} \exp\left(-\frac{\|y - \mathbf{P}x\|_2^2}{2(\sigma_1^2 + \sigma_2^2)}\right)$$

$$= \mathcal{N}(y \mid \mathbf{P}x, (\sigma_1^2 + \sigma_2^2)\mathbf{I}_d)$$

□

$$(*) = \exp\left(-\frac{\|y - \mathbf{P}x\|_2^2}{2}\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_1^4}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)\right)\right)$$

$$= \exp\left(-\frac{\|y - \mathbf{P}x\|_2^2}{2(\sigma_1^2 + \sigma_2^2)}\right).$$

$$(**) = (2\pi)^{d/2}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{d/2} \times \underbrace{\frac{1}{(2\pi)^{d/2}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{d/2}} \int \exp\left(-\frac{1}{2}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)[u + \mu]\right) du}_{=1}.$$

Appendix E: Signal-to-noise ratio of the convergence field κ

In Figure E.1 we show the ratio between the fiducial power spectrum and the power spectrum of the input noise.

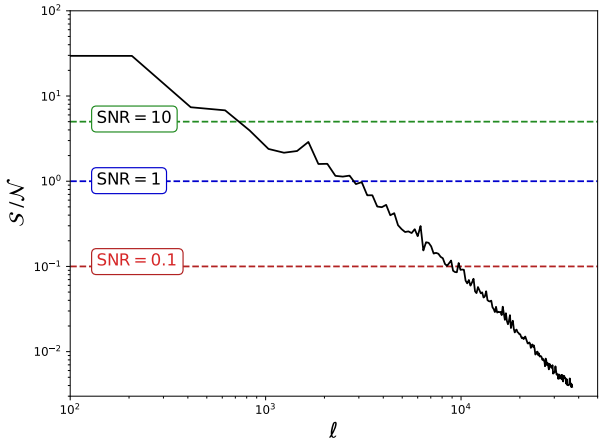


Fig. E.1. Signal-to-noise ratio of the input data. The S/N corresponds to the ratio between the fiducial power spectrum and the power spectrum of the input noise. This shows that the S/N is equal to 0.1 from $\ell = 10^4$, and therefore the reconstruction from this scale does not correspond to the input data, but is driven by the prior only.

Appendix F: Cluster detection

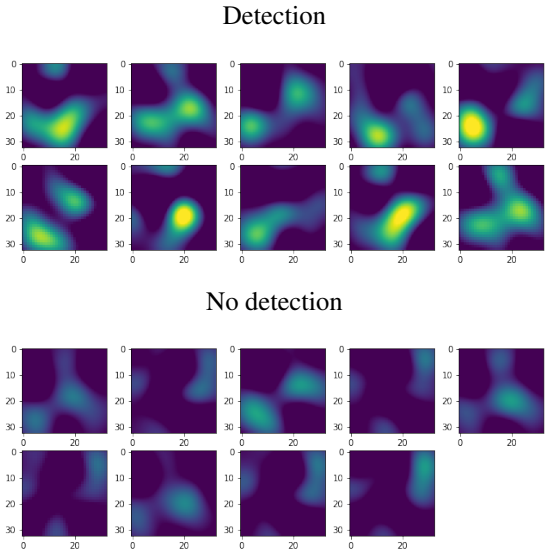


Fig. F.1. Filtered maps of the convergence posterior samples estimated by our method. Every cutout corresponds to the same region of the map. In the upper panels, a cluster was added in the input shear, but not in the lower panels. The upper panel cutouts of the filtered maps show the recovered cluster shape in the center of the map, while lower panel cutouts show the coefficient from structure nearby. Selection was done with a 3σ threshold. The axes indicate the pixel numbers.

In the detection experiment described in section 7.3, we added a NFW profile into the input shear map. We then ran our cluster detection procedure on the associated `DLPosterior` samples. Figure F.1 shows cutouts of those samples around the cluster location. The upper plot shows the samples that were selected with a 3σ threshold and the bottom plot shows the ones that were not selected. We can clearly recognize the shape of a cluster among the samples with positive detection (i.e., with maximum coefficient above 3σ).

Appendix G: Comparison of mass-mapping methods

In figure G.1, we show how the different methods described in section 3 recover a convergence map from the same input shear field used in section 7 and the COSMOS-like survey mask.

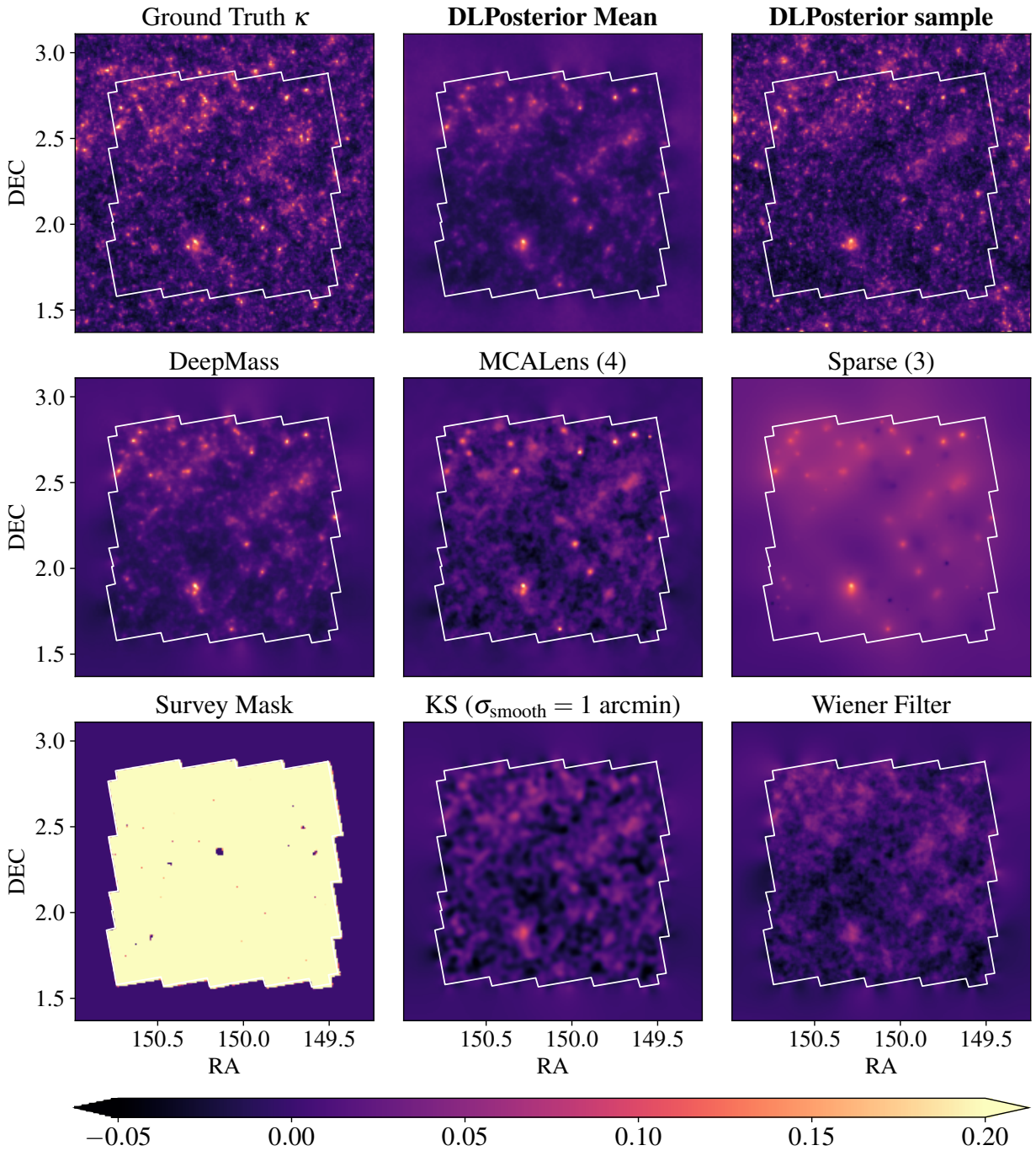


Fig. G.1. Comparison of mass-mapping methods. Ground truth corresponds to the convergence map κ that we aim to recover, taken from Osato et al. (2021), the `DLPosterior` mean and the `DLPosterior` sample are the results discussed in section 7, `DeepMass` is from Jeffrey et al. (2020b), `MCALens` with $\lambda = 4$ is from Starck et al. (2021), the `GLIMPSE` method with $\lambda = 3$ is from Lanusse et al. (2016), Survey Mask is the COSMOS catalog binary mask, Kaiser–Squires with Gaussian smoothing ($\sigma_{\rm smooth} = 1$ arcmin) is from Kaiser & Squires (1993), and the Wiener filte is from Elsner & Wandelt (2013), Jeffrey et al. (2018b).