Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **524,** 6167–6180 (2023) https://doi.org/10.1093/mnras/stad2262
Advance Access publication 2023 August 3

A simulation-based inference pipeline for cosmic shear with the Kilo-Degree Survey

Kiyam Lin ,★ Maximilian von Wietersheim-Kramsta,★ Benjamin Joachimi★ and Stephen Feeney
Department of Physics and Astronomy, University College London, Gower Street, London, WC1E 6BT, UK

Accepted 2023 July 18. Received 2023 June 26; in original form 2022 December 8

ABSTRACT

The standard approach to inference from cosmic large-scale structure data employs summary statistics that are compared to analytic models in a Gaussian likelihood with pre-computed covariance. To overcome the idealizing assumptions about the form of the likelihood and the complexity of the data inherent to the standard approach, we investigate simulation-based inference (SBI), which learns the likelihood as a probability density parameterized by a neural network. We construct suites of simulated summary statistics, exactly Gaussian distributed for validation purposes, for the most recent Kilo-Degree Survey (KiDS) weak gravitational lensing analysis and demonstrate that SBI recovers the full 12-dimensional KiDS posterior distribution with just under 10^4 simulations. We optimize the simulation strategy by initially covering the parameter space by a hypercube, followed by batches of actively learnt additional points. The data compression in our SBI implementation is robust to suboptimal choices of fiducial parameter values and of data covariance. Together with a fast simulator, SBI is therefore a competitive and more versatile alternative to standard inference.

Key words: gravitational lensing: weak – methods: data analysis – cosmological parameters.

1 INTRODUCTION

Cosmological weak lensing in the era of modern high-precision cosmology has proven itself to be an excellent probe of key parameters of the standard Lambda Cold Dark Matter (ΛCDM) Model. Most notably, it is able to constrain a degenerate combination of σ_8 and Ω_m, or alternatively Ω_m and S_8, a combined parameter typically taken as $S_8 = \sigma_8(\Omega_\mathrm{m}/0.3)^{0.5}$. In recent years, two weak-lensing surveys, the Kilo-Degree Survey (KiDS, Asgari et al. 2021) and the Dark Energy Survey (DES, Secco et al. 2022; Amon et al. 2022b), alongside other photometric galaxy surveys such as the Subaru Hyper Suprim-Cam (Sugiyama et al. 2022) have yielded results that are in agreement with each other despite very different methodologies (Asgari et al. 2021; Heymans et al. 2021; Busch et al. 2022; Secco et al. 2022; Amon et al. 2022b).

Interestingly, both KiDS and DES find values of S_8 of between 2σ and 3σ lower than the value inferred by *Planck*, a space-based experiment observing cosmic microwave background anisotropies (Ade et al. 2016; Aghanim et al. 2020). The consistent results from both KiDS and DES alongside their re-analysis suggests that it is unlikely for the tension to arise simply from some un-modelled systematic error (Amon et al. 2022a). If this S_8 tension cannot be resolved through the discovery of systematic errors, then it motivates the search of new physics.

The analysis of cosmic weak-lensing survey data, however, is fraught with challenges from not only the modelling side but also the limitations of traditional inference methods. To elaborate, the modelling must take into account many factors, such as baryon feedback and the intrinsic alignment of galaxies caused by matter-galaxy interactions (Kilbinger 2015; Mandelbaum 2018; Amon & Efstathiou 2022). Therefore, we may find a complex statistical problem to solve within the likelihood function, necessary for the task of cosmological parameter inference involving these stochastic forward modelling processes.

Traditional likelihood analysis requires a likelihood that can be evaluated, but the complete set of these factors makes it impossible to know the exact analytical model of the likelihood written in closed form. For many cases, an accurate likelihood model that takes into account all of the statistical features is essentially too expensive and thus intractable to evaluate (Jeffrey, Alsing & Lanusse 2021). To this end, it is routine in cosmological surveys to assume a Gaussian likelihood as an approximation to the true likelihood.

This Gaussian likelihood assumption is employed on summary two-point statistics (Asgari et al. 2021), which are sensitive to the underlying cosmology. However, in the use of two-point statistics such as the correlation function, we may find significant deviations from a Gaussian likelihood when we consider the two-point statistics' sensitivity to low multipoles (Schneider & Hartlap 2009; Sellentin, Heymans & Harnois-Déraps 2018). This is true even when the underlying lensing fields are Gaussian (Sellentin & Heavens 2018; Sellentin et al. 2018; Taylor et al. 2019; Upham, Brown & Whittaker 2021). Systematic effects could also introduce non-Gaussianity, with their relative importance increasing as surveys become more statistically powerful.

As such, to tackle the statistical side of cosmological analysis, there has been a growing number of forward simulation-based methods (Fluri et al. 2018; Gupta et al. 2018; Ribli et al. 2019; Taylor et al. 2019; Jeffrey et al. 2021; Fluri et al. 2022; Hahn et al.

★ E-mail: kiyam.lin.20@ucl.ac.uk (KL); maximilian.wietersheim-kramsta.19@ucl.ac.uk (MvW-K); b.joachimi@ucl.ac.uk (BJ)

2022). These seem attractive as they completely circumvent the need of evaluating or working with an explicit or computable form of the likelihood function. This then allows these simulation-based inference (SBI) methods to fully propagate all of the uncertainties and survey systematics from data to parameters through forward simulation. It should be noted, however, that SBI methods are not the only methods that circumvent the use of a Gaussian likelihood, for example, Bayesian Hierarchical Models (Porqueres et al. 2021a, b).

Most tantalizing, however, is that combining likelihood-free, SBI methods with recent advances in machine learning provides an inference methodology that is not only capable of freeing the analysis pipeline from intractable likelihoods, but also computationally cheaper when paired with a similarly fast simulator. For example, in Alsing et al. (2019), only 1000 simulations were needed to infer a posterior with the same constraints as a long Markov chain Monte Carlo (MCMC) run that required at least 10 000 likelihood evaluations to converge. These techniques have been explored in detail by others with high levels of success (Fluri et al. 2018; Gupta et al. 2018; Ribli et al. 2019; Jeffrey et al. 2021; Fluri et al. 2022).

Notably, Fluri et al. (2022) performed a full wCDM analysis of KiDS-1000 weak lensing using deep learning. Despite using the same data set as in this analysis, the Fluri et al. (2022) analysis is concerned with going beyond the standard weak-lensing framework by using field-level summary statistics in order to constrain cosmologies beyond the standard ΛCDM. In that analysis, a graph convolutional neural network is trained on the cosmology dependence of the simulated weak lensing maps with respect to four parameters. Using that as a summary statistic, the parameters are then inferred from the data using approximate Bayesian computation (ABC). In this work, the aim is instead to show the feasibility of applying density estimation SBI to a standard ΛCDM weak-lensing analysis with its full complexity in parameter space; such an analysis has not been conducted until now. We achieve this by demonstrating that this methodology allows us to infer all 12 of the standard KiDS-1000 parameters using mock data vectors that are Gaussian drawn to allow for validation.

Arguably the idea of SBI began with the seminal work of Rubin (1984) when describing the process of inferring a posterior in Bayesian analysis from a frequentist perspective. The core idea being that if the parameter values to a typically stochastic forward modelling process generates a data vector identical to the observed data vector, then those parameter values must make up part of the inferred posterior. This in turn means that they have a high probability of being the 'true' parameter values. In essence, a rejection sampling schema for the posterior.

This idea led to the birth of a class of SBI more commonly known today as ABC. However, one might immediately notice that for any realistic stochastic process, getting an exact match in the data vector is highly improbable, leading to Pritchard et al. (1999) introducing a notion of closeness to allow for imperfect matching, an idea that is also fraught with problems requiring ever increasingly complex notions of closeness.

Furthermore, with the low probability of obtaining a data vector that passes any such closeness criteria, one can imagine that the ABC methodology is computationally inefficient, requiring many forward simulations to find just one posterior parameter point. In light of this, many have tried to develop better sampling schemas to increase the efficiency of ABC, but even after many such improvements, Leclercq (2018) estimates that computational efficiency remains low. This ABC method, however, has met high levels of success (Pritchard et al. 1999; Marin et al. 2012; Akeret et al. 2015; Ishida et al. 2015; Jennings & Madigan 2017; Prangle 2017; Fluri et al. 2018; Leclercq 2018; Fluri et al. 2022).

There is a desire to make use of all of the information available from forward simulation, giving rise to an alternative SBI methodology in the form of density estimation likelihood-free inference (DELFI). In this schema, the probability density of the sampling distribution is learnt through the use of neural networks. After evaluating this probability density at a given mock or observed data vector, a likelihood and thus posterior can be recovered. This method has the advantage of not requiring an explicit and often simplified form of a likelihood, and also does not throw away the information from forward simulations that do not produce a data vector that passes any 'closeness' criteria.

In recent years, DELFI has shown itself to be capable of inferring tight constraints on the final posterior surface with an almost order of magnitude fewer forward simulations than traditional Bayesian methods (Papamakarios & Murray 2016; Alsing et al. 2019; Jeffrey et al. 2021; Hahn et al. 2022). In this paper, we apply this method of statistical inference to mock KiDS-1000 cosmic shear data. We validate the method using simplified simulations but with the full set of inferred parameters. We also optimize the method to explore how many simulations are needed in comparison to the number of parameters being varied and inferred. The setup involves a mixture of both data-sensitive parameters but also prior-driven parameters.

This paper is structured as follows: Section 2 details the specifics of the software used to perform DELFI SBI as well as the compression scheme that is applied on to the cosmological summary statistics. Section 3 provides an overview of the cosmological setup as well as the test simulations used to generate mock data vectors. In Section 4, we outline the process of validating our SBI methodology, testing for robustness as well as optimizing the method for the number of simulations and the learning process. Importantly, in this section, we demonstrate that the method can be easily made robust towards both poor choices of fiducial cosmology as well as being robust to sub-optimal compression.

2 SIMULATION-BASED INFERENCE

2.1 DELFI

The methodology of DELFI SBI is depicted in Fig. 1. First, both the observed data and simulated data are compressed into a set of informative summaries. The compressed simulated data are used to train a neural network, which after applying the compressed observed data yields an empirical likelihood. After multiplying this learned likelihood with the priors, one obtains posterior parameter constrains.

Alsing et al. (2019) outlined a variety of ways of performing DELFI. The methodology we use involves learning the sampling distribution of data vectors conditional on the input cosmology, $p(d|\theta)$ where d denotes data vector and θ cosmological parameters. By evaluating this learned sampling distribution at the observed data vector, one obtains the likelihood, which, after multiplication with a prior, yields the posterior through the use of Bayes' theorem (Lueckmann et al. 2019; Papamakarios, Sterratt & Murray 2019).

The biggest advantage of learning the sampling distribution of data as a function of parameters versus learning the joint distribution is that the networks do not have the prior embedded within them. This means that one can acquire forward simulations in regions of posterior interest without worrying about importance re-weighting issues (Alsing et al. 2019; Papamakarios et al. 2019). This methodology also means that different priors can be explored and changed a posteriori without similar re-weighting issues.

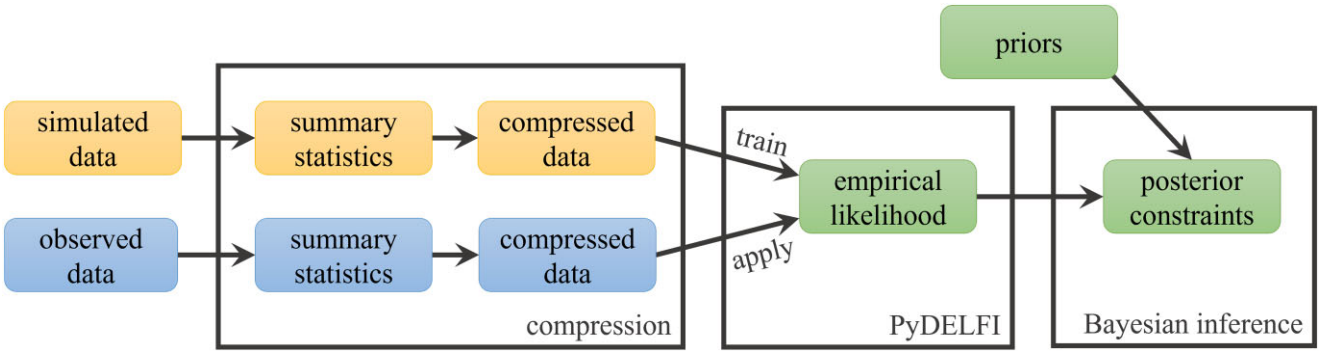

Figure 1. An overview of DELFI SBI using `PyDELFI`. First, both the observed data and simulated data are compressed into a set of informative summaries. The compressed simulated data are used to train `PyDELFI`'s neural networks. Applying the compressed observed data yields an empirical likelihood, which after application of the priors produces posterior parameter constraints.

The specific implementation of density estimation SBI used is that of the `PyDELFI` software package.[1] (Alsing et al. 2019). To account for any numerical anomalies or learning problems in the density learning process, a committee of neural density estimators (NDEs) is employed. This means that multiple NDEs are trained independently and later combined, weighted by how well they learned the target density. `PyDELFI` supports natively two different classes of NDEs, Mixture Density Networks (MDNs) and Masked Autoregressive Flows (MAFs).

We found that MDNs performed poorly for our problem, especially when considering high-dimensional parameter spaces, so we will not discuss them further (see Alsing et al. (2019) for details on the method). We make use of MAFs, which are NDEs constructed out of a chain of Masked Autoregressive Density Estimators (MADEs). As any probability distribution can be written as a chain of one-dimensional conditional probabilities, a MADE effectively learns a target distribution through a series of conditional probability distribution transformations, $p(t_i|\mathbf{t}_{1:i-1}, \theta)$, back to the unit normal, where $\mathbf{t}$ is the data vector of interest. The means and variances are parameterised by a neural network with weights, $\mathbf{w}$ (Germain et al. 2015; Uria et al. 2016; Alsing et al. 2019). A MADE therefore has a functional form of

$$p(\mathbf{t}|\boldsymbol{\theta};\mathbf{w}) = \prod_{i=1}^{\dim(\mathbf{t})} p(t_i|\mathbf{t}_{1:i-1}, \boldsymbol{\theta};\mathbf{w}). \tag{1}$$

where each conditional $p(t_i|\mathbf{t}_{1:i-1}, \boldsymbol{\theta};\mathbf{w})$ is conditioned on having observed the previous $\mathbf{t}_{1:i-1}$ data vector values. Alsing et al. (2019) write that any single MADE has two key limitations. One limitation is that a single MADE is sensitive to the order of the factorization whilst the other is that simple conditionals may not be flexible enough to learn complex target distributions. The information pertaining to suitability of conditionals or factorization order, however, is not available a priori.

To overcome these limitations, MAFs are employed. A MAF addresses these limitations by creating a stack of individual MADEs, where the output $\mathbf{u}$ of each MADE is used as the input distribution for the next MADE in the stack (Papamakarios, Pavlakou & Murray 2017). By creating an ensemble of MADEs with random re-ordering of the factorization order between each MADE, the limitation and sensitivity of factorization order is overcome. Papamakarios et al. (2019) write that these MAFs are very flexible NDEs and well suited to the task of likelihood-free inference. A MAF as a NDE can thus be expressed as

$$p(\mathbf{t}|\boldsymbol{\theta};\mathbf{w}) = \mathcal{N}[\mathbf{u}(\mathbf{t},\theta;\mathbf{w})|\mathbf{0},\mathbf{I}] \times \prod_{n=1}^{N_{\rm mades}} \prod_{i=1}^{\dim(\boldsymbol{t})} p_i^n(\mathbf{t},\theta;\mathbf{w}), \tag{2}$$

where $\mathbf{u}$ is the output from the final MADE. $\mathcal{N}[\mathbf{u}(\mathbf{t},\theta;\mathbf{w})|\mathbf{0},\mathbf{I}]$ denotes the aforementioned unit normal and $\prod_{i=1}^{\dim(\boldsymbol{t})} p_i^n(\mathbf{t},\theta;\mathbf{w})$ represents the chain of conditional probabilities factorized via the chain rule. The reason this expression picks up the $\mathcal{N}[\mathbf{u}(\mathbf{t},\theta;\mathbf{w})|\mathbf{0},\mathbf{I}]$ term is because we can think of a MADE, or a stack of MADEs in a MAF as learning the transformation of $\mathbf{t}$ back to the unit normal as hinted in Alsing et al. (2019).

To train these NDEs, `PyDELFI` minimizes the Kullback–Leibler divergence between the parametric density estimator and the target density. However, for the purposes of SBI, the target probability density is not known. As such, a Monte Carlo estimate of the Kullback–Leibler divergence is used instead for the target density. More details can be found in Alsing et al. (2019).

However, any application of machine-learning trained on a small training set by minimization of the loss function easily runs into the problem of potentially optimizing said density estimators for local minima that do not well represent the larger data set as a whole. As such `PyDELFI` employs the technique of training an ensemble of NDEs with a range of network architectures. By doing so, one constructs a stack of NDEs, with stacking weights (contribution weighting) of each NDE given by the relative likelihoods for each NDE. A stack of NDEs combined this way is reported to perform better than any single NDE (Smyth & Wolpert 1998, 1999). This give

$$p(\mathbf{t}|\theta;\mathbf{w}) = \prod_{\alpha=1}^{N_{\rm NDEs}} \beta_\alpha\, p_\alpha(\mathbf{t}|\theta;\mathbf{w}), \tag{3}$$

where β_α represents the stacking weight of each NDE with index α and $\sum\beta_\alpha = 1$.

It is possible to run `PyDELFI` in two different modes which we will refer to as *batch run* mode and *active learning* mode. In the *batch run* mode, simulations are run beforehand before being fed as one batch to `PyDELFI`. This means that to use the *batch run* mode setting, it is typically prudent to select parameter points at which to run forward simulations by sampling from the prior with an appropriate method, such as by using an equally spaced grid or a latin hypercube. The main drawback of this mode is that with any individual standalone run, it is hard to tell whether a sufficient number of forward simulations have been run without some ground truth to

[1] https://github.com/justinalsing/pydelfi

which the results can be compared. This is because the absolute value of the loss is dependent on the number of parameters being inferred, and there is not one target loss to aim for across all models and runs.

In the active learning mode, `PyDELFI` proposes new parameter values at which to run simulations after obtaining data vectors from a small initial set of simulations. This means that for the initial set of simulations one can choose to either randomly sample from the prior or make use of a latin hypercube that maximally covers the prior volume efficiently. We choose the latter. After training on this initial set of simulations, `PyDELFI` then acquires further sets of parameter-data pairs by sampling from a weighted mixture of the intermediate posterior and the prior, however, other parameter acquisition schemes may be used.

The performance of neural networks is typically also sensitive to their initialization. For our work, given that we know that the likelihood will be approximately Gaussian around its peak from the published results of the KiDS-1000 team (Asgari et al. 2021), we can make use of a Fisher matrix, **F**, multiplied by a factor of safety to initialize our ensemble of NDEs. We express the Fisher matrix as

$$\mathbf{F} = \nabla\boldsymbol{\mu}^{\mathrm{T}}\mathbf{C}^{-1}\nabla^{\mathrm{T}}\boldsymbol{\mu}, \tag{4}$$

where $\nabla\boldsymbol{\mu}$ is the derivative of the data vector at the fiducial cosmology and **C** is the data covariance assuming a Gaussian likelihood (See Alsing & Wandelt (2018) for more details). We have made use of the fact that the covariance is cosmology independent here.

The factor of safety is introduced to ensure that the NDEs are not initialized with too restrictive a volume. In practice, this means we initialize our NDEs with a Gaussian target distribution with their means equal to the chosen fiducial cosmology parameter values and a covariance equal to the inverse Fisher matrix multiplied by a constant acting as a factor of safety. To elaborate, the NDEs are initialized before training to $p(\mathbf{t}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{t}|\boldsymbol{\theta}, k\mathbf{F}^{-1})$, where k denotes the aforementioned factor of safety constant.

2.2 Parameter sampling

The process of SBI means that ideally the forward simulations that generate the data vectors are drawn from a set of parameters that maximally cover the prior volume. As such a latin hypercube covering the prior volume would be ideal as it is a method of maximally covering the volume of parameters of interest with minimal computational waste by not sampling any particular parameter values twice (Stein 1987; Park 1994; Loh 1996). As the KiDS prior contains a mix of top hat and Gaussian components, the hypercube generating algorithms provided by the `PyDOE`[2] package could be employed alongside `SciPy`[3] with only minor modifications.

Step one of the algorithm is to divide the prior volume into a lattice of equally spaced hypercuboids. Step two is to then randomly choose hypercuboids such that along each dimension no parameter interval is sampled twice. From here, either a parameter point can be picked randomly within the chosen intervals, or the centre of each interval can be chosen. Running the inference pipeline on either choice appears to make little difference, so for simplicity's sake the middle of each interval is chosen as the sampled value for our work. This random choosing of hypercuboids can be run multiple times. A measure of minimum Euclidean distance between any two points is used as a measure of how spread out the points are within the hypercube, and any random sampling that produces a larger minimum Euclidean distance is deemed to be a better sample. This process can be repeated an arbitrary number of times with the only caveat being an increase in computational cost.

The process ends here for parameters that have a flat prior, but for parameters that have a Gaussian prior there is an additional step. For the parameters that have a Gaussian prior, `SCIPY` is used to map the flat spread of parameter points on to a Gaussian target distribution through its inverse cumulative distribution function.

2.3 Score compression

The data vector we have chosen for this work is that of weak-lensing two-point correlation functions, which in the KiDS-1000 setup has length 270 (see Section 3). Using data vectors of such length directly within `PyDELFI` would be prohibitively expensive and difficult to fit due to its high dimensionality. Hence, massive data compression is necessary to compress the data vectors into highly informative summary statistics with minimal loss in information.

For this massive data compression task, there are a few methods to choose from. One option would be to use a neural network to try and maximize the information content automatically. For example, this can be done through the use of the software package `IMNN` (Charnock, Lavaux & Wandelt 2018), which uses a neural network to construct summaries that maximize the Fisher information. Another data compression method is MOPED (Heavens, Jimenez & Lahav 2000), a linear compression method built upon the more classic method of Karhunen–Loéve eigenvalue decomposition (Tegmark, Taylor & Heavens 1997). However, since we know from previous analyses that the parameter likelihood from cosmic shear two-point statistics will be approximately Gaussian close to the peak of the likelihood (Schneider & Hartlap 2009; Sellentin & Heavens 2018; Sellentin et al. 2018; Taylor et al. 2019; Upham et al. 2021), we make use of linear score compression as outlined in Alsing & Wandelt (2018).

Given a log-likelihood, $\mathcal{L}$, its Taylor expansion around a set of fiducial parameters, $\boldsymbol{\theta}_*$ with respect to $\delta\boldsymbol{\theta}$ can be written as

$$\mathcal{L} = \mathcal{L}_* + \delta\boldsymbol{\theta}^{\mathrm{T}}\nabla\mathcal{L}_* - \frac{1}{2}\delta\boldsymbol{\theta}^{\mathrm{T}}\mathbf{J}_*\delta\boldsymbol{\theta}, \tag{5}$$

where a $_*$ denotes evaluation at the fiducial parameter values, J_* is the observed information matrix, $\mathbf{J}_* \equiv -\nabla\nabla^T\mathcal{L}_*$. To linear order in the parameters, the data vector only couples to the parameters via the $\nabla\mathcal{L}_*$ term, commonly referred to as the score, with $\mathbf{s} \equiv \nabla\mathcal{L}$. By construction, this score function is a vector of length n, where n is the number of parameters. This is as derivatives of the log-likelihood are taken with respect to $\boldsymbol{\theta}$. As such, this provides a natural method for compressing any data vector of length N to a massively smaller vector of length n, sensitive to changes in the parameters to the first order. Alsing & Wandelt (2018) remark that this linear compression method generalizes the linear Karhunen–Loéve compression and MOPED schemas considered in Heavens et al. (2000) and Tegmark et al. (1997).

However, as already alluded to above, this method of compression does require some form of an approximate likelihood with respect to the parameters of interest: the more accurate the likelihood, the more optimal the compression. It should be emphasized that an inaccurate likelihood approximation would only result in lossy compression, and would not bias the inference, simply producing wider posterior contours. For our work, we choose a Gaussian form for the likelihood for which this method of compression saturates the Cramér–Rao bound (Alsing & Wandelt 2018).

[2] https://github.com/tisimst/pyDOE
[3] https://github.com/scipy/scipy

Moreover, we assume that the covariance is parameter-independent, which allows us to drop any partial derivatives with respect to the covariance. The contribution to the Fisher matrix by the parameter dependence of the covariance is suppressed for larger survey area (REF Tegmark, Taylor, Heavens). We note that the SBI analysis does fully account for the parameter dependence of statistical errors, as opposed to the standard Gaussian likelihood analysis. This leaves us with a greatly simplified linear score compression function, defined as

$$\boldsymbol{t} = \nabla\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{C}^{-1}(\boldsymbol{d} - \boldsymbol{\mu}), \tag{6}$$

where $\boldsymbol{t}$ now denotes the compressed data that will be used as the information-sufficient summary statistics to be fed into `PyDELFI`.

However, with the expression of score compression as given in equation (6), the numerical values of the compressed summary statistics are not directly informative. Through equation (5), Alsing & Wandelt (2018) show that by maximizing the Taylor-expanded log-likelihood, the score compression numbers can be mapped on to a quasi maximum-likelihood estimator through

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_* + \boldsymbol{F}_*^{-1}\nabla\mathcal{L}_* = \boldsymbol{\theta}_* + \boldsymbol{F}_*^{-1}\boldsymbol{t}_*, \tag{7}$$

where $\boldsymbol{F}_*$ denotes the Fisher matrix evaluated at the fiducial cosmology and $\boldsymbol{\theta}_*$, the fiducial cosmology parameter values. We use these quasi maximum-likelihood estimators as the summary statistics within `PyDELFI` to be able to make use of the Fisher initialization schema mentioned in Section 2.1.

Furthermore, it is in this compression step that we may choose to marginalize over certain parameters such as any nuisance parameters. For example, if we know that our data are only sensitive to a subset of the parameters varied through forward simulation, those parameters can be marginalized out of the analysis through the score. Alsing & Wandelt (2019) proposed a method of doing this whilst maximizing the information within the data vector's sensitivity to those parameters through use of the Fisher matrix, a method they call 'nuisance hardened score compression'.

For our cosmological setup, the parameters that we might choose to marginalize out contain little information, meaning that the Fisher matrix yielded little to no information pertaining to the marginalized parameters. This meant that for our work, we could marginalize our compressed summaries by simply truncating the score, removing the terms that corresponded to the parameters we wished to marginalize over. To mirror the analysis done by the KiDS-1000 team, however, no marginalization of the score is done throughout this work. It should be noted that if the covariance depended on parameters, then such a clean separation between statistic and parameter is impossible, and instead more mixing would be observed between compressed statistics and parameter values. This would also result in a more complex marginalization process beyond simply truncating the score.

3 SIMULATIONS OF COSMIC SHEAR DATA FROM KIDS-1000

For the generation of mock data from initial parameters, forward simulation is done using the `KCAP`[4] and `CosmoSIS`[5] packages (Zuntz et al. 2015). For this analysis, the output of the simulations are the shear 2PCFs, $\xi_{\pm}(\theta)$. To calculate the 2PCFs, first the cosmological pipeline assumes a spatially flat ΛCDM model. The linear matter power spectrum is calculated using `CAMB` (Lewis, Challinor & Lasenby 2000; Howlett et al. 2012) with its non-linear evolution calculated using `HMCode` (Mead et al. 2015). `HMCode` makes use of a halo model with baryonic feedback. The amplitude of the halo mass-concentration, a_{bary}, is allowed to vary freely, whilst the halo model bloating parameter η_0 is fixed in relation to a_{bary} (see Joachimi et al. 2021, for more details).

The effects of the intrinsic alignment of galaxies is factored in through the non-linear alignment model of Bridle & King (2007). Following the pipeline as set out by KiDS (Asgari et al. 2021), the Limber approximation is used to project the matter power spectrum along the line of sight to obtain $C_{\epsilon\epsilon}(\ell)$, which are the observed cosmic shear angular power spectrum that are dependent on multipole ℓ. The total shear angular power spectrum is the sum of contributions from gravitational lensing (G) and intrinsic alignments (I), giving,

$$C_{\epsilon\epsilon}(\ell) = C_{\mathrm{GG}}(\ell) + C_{\mathrm{GI}}(\ell) + C_{\mathrm{II}}(\ell). \tag{8}$$

The $C_{\epsilon\epsilon}(\ell)$ are subsequently transformed into 2PCFs, $\xi\pm$,

$$\xi\pm(\theta) = \int_0^{\infty} \frac{\mathrm{d}\ell\,\ell}{2\pi} \mathrm{J}_{0/4}(\ell\theta) C_{\epsilon\epsilon}(\ell), \tag{9}$$

with $\mathrm{J}_{0/4}$ denoting Bessel functions of the first kind and θ the angular separation on the sky. Following KiDS-1000, we assume zero contribution from the B-modes to the 2PCFs.

The source galaxies are split up into five redshift bins with bin boundaries [0.1, 0.3, 0.5, 0.7, 0.9]. All of the cross-correlation and autocorrelation pairs between respective redshift bins were taken into account, resulting in 15 redshift bin pairs. Following KiDS-1000, scale cuts are performed on the 2PCFs, only keeping angular separations of between 0.5 and 300 arcmin with a total of nine angular bins. This resulted in a data vector of length 270.

As the goal is to test the performance of the `PyDELFI` SBI pipeline, we generate our own mock data vectors by sampling the 2PCFs using the KiDS analytic covariance as derived in Joachimi et al. (2021). We do this to have full control over the results and to perform valid testing of the SBI methodology by way of comparison to that of traditional likelihood analysis. Furthermore, for the purposes of testing we generated a mock data vector that was used as the observed data vector for both traditional likelihood analysis and SBI.

In the future, we plan to apply this pipeline to a novel suite of physically informed forward-simulations of weak-lensing observables (von Wietersheim-Kramsta M, Lin K, Tessore N, Joachimi B, Loureiro A, Reichke R, Wright A, in preparation). For this full SBI analysis of KiDS-1000 data, the simulations will include all relevant systematics and physical effects which might induce non-Gaussianity into the likelihood.

Following the methodology of KiDS, we vary five cosmological parameters and two astrophysical nuisance parameters with flat priors identical to the ones used by KiDS (see section 3.3 of Joachimi et al. 2021). The cosmological parameters varied include σ_8, the present-day root-mean-square matter fluctuation averaged over a sphere of radius $8h^{-1}$Mpc; the density parameter for cold dark matter, $\omega_{\mathrm{c}} = \Omega_{\mathrm{c}}h_0^2$ and baryonic matter, $\omega_{\mathrm{b}} = \Omega_{\mathrm{b}}h_0^2$ multiplied by h_0, the dimensionless Hubble constant. The spectral index of the primordial power spectrum, n_{s}, is likewise varied.

The two astrophysical nuisance parameters are A_{IA}, the intrinsic alignment amplitude of galaxies and a_{bary}, the baryonic feedback amplitude. Furthermore, we define matter density as $\Omega_{\mathrm{m}} = \Omega_{\mathrm{c}} + \Omega_{\mathrm{b}} + \Omega_{\nu}$, where Ω_{ν} is the neutrino density. Finally, the shifts in the means of the redshift distribution bins follow a Gaussian prior with covariance that can be found in the latest KiDS data

[4] https://github.com/KiDS-WL/kcap
[5] https://bitbucket.org/joezuntz/cosmosis/wiki/Home

Table 1. The prior ranges for the parameters to be inferred alongside the parameter values used to generate the mock data vector and the fiducial cosmology for compression. The prior ranges for σ_8, ω_b, ω_c, n_s, h_0, a_{bary}, and A_{IA} are all top hats, whilst the prior for δ_z follows a correlated Gaussian distribution characterized by a covariance matrix C with mean of μ. The δ_z parameters encapsulate freedom in the mean of the redshift distribution bins whilst the other parameters are: σ_8, the root-mean-square matter fluctuation; ω_b, baryonic matter density; ω_c, cold dark matter density; n_s, scalar spectral index; h_0, Hubble constant; a_{bary}, baryonic feedback parameter; A_{IA}, galaxy intrinsic alignment amplitude. We set the equation of state parameter as $w = -1$, pick a flat curvature, $\omega_k = 0$, and fix a neutrino mass sum of $\Sigma m_\nu = 0.06\text{eV}/c^2$.

Parameter	Prior range	Mock data	Fiducial
σ_8	[0.6, 1.0]	0.8	0.811
ω_b	[0.019, 0.026]	0.0230	0.0224
ω_c	[0.07, 0.18]	0.120	0.120
n_s	[0.8, 1.15]	0.960	0.965
h_0	[0.6, 0.9]	0.674	0.674
a_{bary}	[2.0, 4.0]	3.10	3.13
A_{IA}	[−6.0, 6.0]	0.960	0.974
$\delta_z[5]$	$\mathcal{N}(\mu, C)$	0.0	0.0

release repository.[6] For our analysis, the mean shift in the redshift distribution is set to zero. See Table 1 for a summary of the parameters varied and their prior ranges.

It should be noted that the 2PCFs are only strongly sensitive to the parameters σ_8, ω_c and A_{IA}. This means that we expect the prior to dominate the posterior for all of the other parameters that are varied.

4 VALIDATION AND OPTIMIZATION

4.1 SBI methodology validation

To validate the methodology, using the setup described in Section 3, both a mock observed data vector and fiducial cosmology data vector were generated. We choose Planck 2018 cosmology values (Aghanim et al. 2020) as our fiducial cosmology but pick a slightly different set of cosmology values to generate the mock observed data vector. We test robustness and sensitivity to the choice of fiducial cosmology later in Section 4.2.

The generated data vectors are compressed following the schema outlined in Section 2.3. The compressed summary statistics are then fed into a `PyDELFI` pipeline that initializes the NDEs with the inverse Fisher matrix, as mentioned in Section 2.1, before training the ensemble of NDEs on forward simulated data-parameter pairs.

Importantly, as our data are only sensitive to a subset of the parameters, with the other parameters being highly prior-driven, we add this prior information to the inverse Fisher matrix used for NDE initialization via the method set out in Coe (2009). This will have no effect on the final inferred parameter posterior, but it helps regularise the NDE initialization making it perform more consistently.

In the end, the chosen ensemble of NDEs used included six MAFs comprised of three to eight MADEs, respectively. This choice was made after trying a variety of different combinations of NDEs, with this combination providing good performance without being too restrictive as would be the case if MAFs with fewer components were chosen.

The results of the SBI pipeline were compared each time to a standard likelihood inference pipeline using `emcee`[7] (Foreman-Mackey et al. 2013) that made use of 48 000 model evaluations, with convergence tested using the integrated correlation time as recommended by Foreman-Mackey et al. (2013). As mentioned previously, the results of the standard likelihood inference pipeline were treated as the ground truth for testing purposes. Fig. 2 shows the comparison between the traditional likelihood analysis versus the output of the SBI pipeline after running forward simulations with all 12 parameters varied. It is clear that the SBI pipeline is able to reproduce the ground truth posterior with all 12 cosmological parameters varied.

This particular posterior was obtained after 11 000 forward simulations with `PyDELFI` set to run in its active learning mode. The one-dimensional marginal posteriors differ slightly for h_0 and ω_b, which is due to the posterior being prior driven with a low sensitivity to the data. This means that we expect the posterior to be flat for these parameters.

In particular for ω_b, we can see that the posterior from SBI reflects the expected flat distribution better than the standard MCMC analysis that assumed a Gaussian likelihood. This shows us that the SBI methodology accurately reflects any deficiency in information within the data vector concerning parameter constraints.

Classifier two-sample tests (C2ST) are often used to determine how well a posterior has been learned, whereby a classifier is trained to see if it can distinguish between samples from the ground-truth distribution and samples from the learned distribution (Friedman 2003; Lopez-Paz & Oquab 2016). A value of 0.5 in the test would indicate the classifier cannot distinguish between the two distributions whilst a value of 1.0 would indicate the classifier can perfectly distinguish between the two distributions. We found that our methodology when tested with C2ST was competitive with what Miller et al. (2021) found the performance of `PyDELFI` to be, giving a value of 0.65 with $\mathcal{O}(10^4)$ simulations when only considering three-dimensional marginalized posteriors in σ_8, Ω_m, A_{IA} and a value of 0.6 when only a two-dimensional posterior is considered.

4.2 SBI sensitivity to choice of fiducial cosmology

Our SBI methodology requires a choice of fiducial cosmology both to initialize the NDEs as well as to do score compression. There is, however, no way to know what a good choice of fiducial cosmology is a priori, and importantly we would not want the choice of fiducial cosmology to bias the results. Instead, a poor choice of fiducial cosmology should only result in sub-optimal compression. Thus, it was important to test this methodology for robustness against choice of fiducial cosmology.

To test for this, we generated a set of 100 mock observed data vectors using varying cosmologies that spanned the prior in σ_8 and Ω_m. For this set of mock observed data vectors, the other parameter values were kept fixed to the values depicted in Table 1. For the sake of simplicity, instead of running the SBI pipeline in its active learning mode, a batch of 24 000 forward simulations were pre-run with their data vectors compressed using the schema outlined in Section 2.3 and all 12 parameters varied drawn from a latin hypercube. Separate runs of the SBI pipeline were then performed using each of the individual mock observed data vectors.

We first found that the inferred posterior is always consistent with the cosmology used to generate the mock observed data vectors, even

[6] https://github.com/KiDS-WL/Cat_to_Obs_K1000_P1/

[7] https://github.com/dfm/emcee

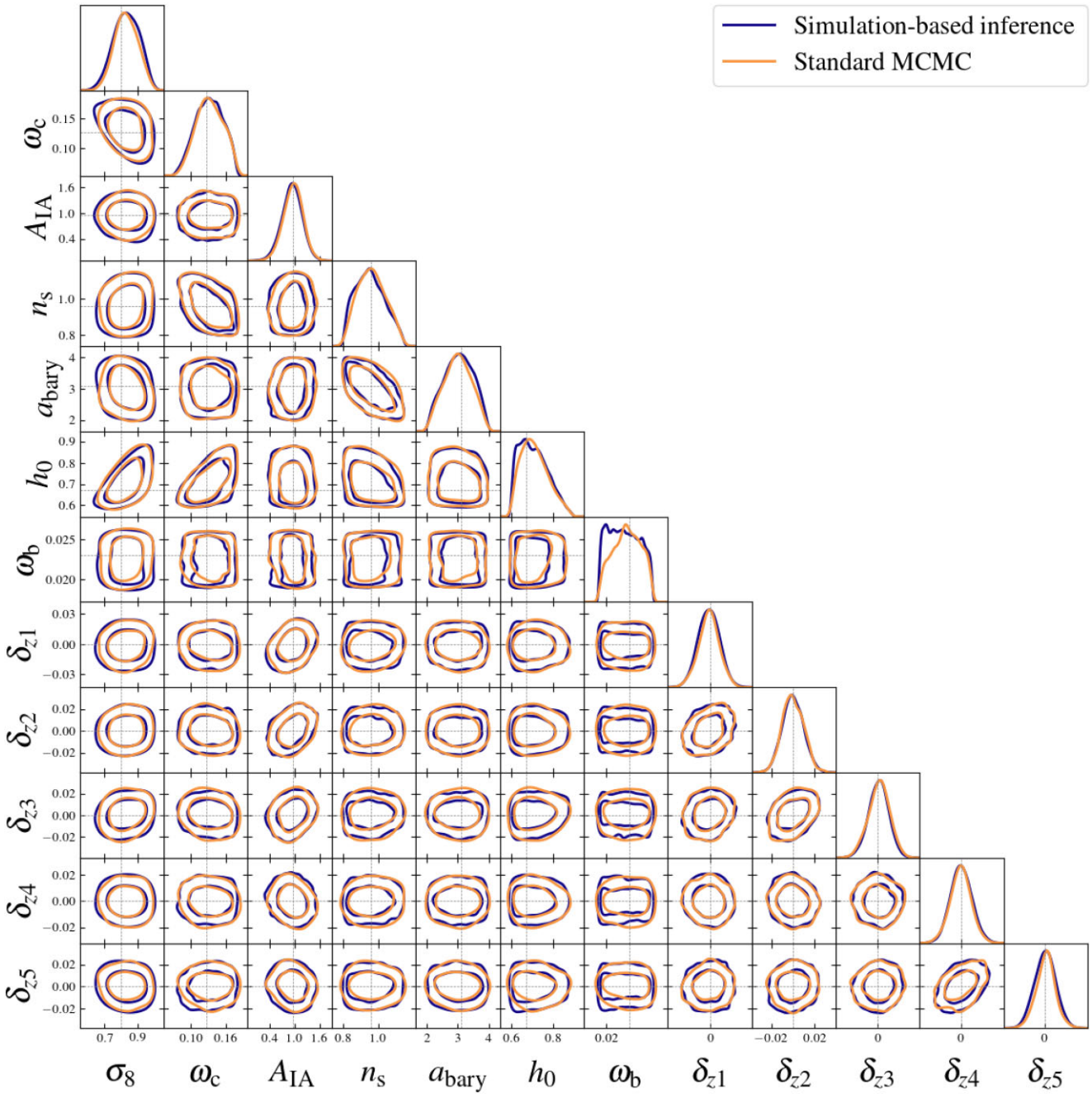

Figure 2. Posterior distributions of the full parameter set of KiDS-1000 obtained through a standard MCMC analysis (orange) versus the SBI pipeline (blue). Choices for the fiducial cosmology and the cosmology parameter values for the mock observed data are outlined in Table 1. The dashed grey lines depict the cosmology used to generate the mock observed data. The SBI contours were obtained with `PyDELFI` in its active learning mode that made use of 11 000 forward simulations. In comparison, the MCMC analysis made use of 48 000 model evaluations.

when the fiducial cosmology lay outside of the posterior. We then wished to see how the standard deviation in S_8 is affected by the choice in fiducial cosmology. Fig. 3 depicts the percentage change in standard deviation in the S_8 marginal posterior. We find that when the true S_8 and corresponding σ_8 value is larger than the fiducial S_8 and σ_8 value, the posterior in S_8 is artificially widened, whilst for the rest of parameter space in Ω_m and σ_8, there is no clear trend. For the majority of cases, the change in standard deviation is under 5 per cent; a small percentage for an inherently noisy process due to both cosmic variance changing with S_8 and the stochastic nature of NDE training. This indicates to us that after running an initial analysis with a fiducial cosmology that will yield parameter constraints consistent with the data cosmology, it would be prudent to re-compress the data vector with the newly inferred data cosmology.

In practice, this first involves performing inference with a fiducial cosmology. The maximum a posteriori (MAP) of this inference can be found with an optimizer such as Nelder–Mead (Nelder & Mead 1965), and we use the MAP parameter values to re-compress our data. We then re-perform `PyDELFI` inference with this once-iterated MAP cosmology to yield more accurate constraints.

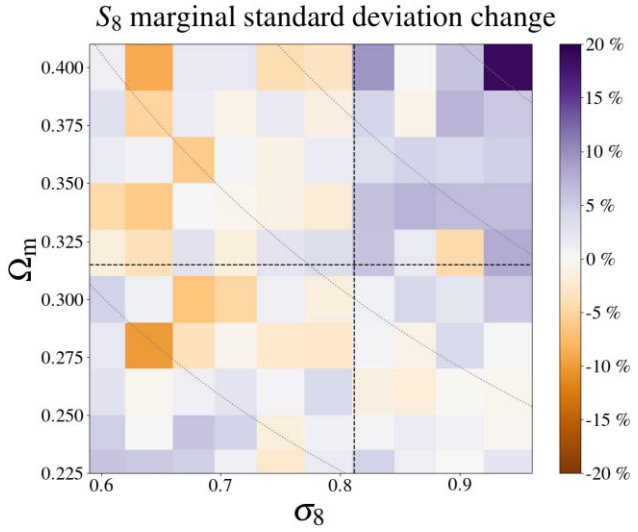

Figure 3. The relative size of the S_8 marginal posterior standard deviation compared to a fiducial analysis where the mock data vector was the same as the mock data vector. The Ω_m and σ_8 axes depict the cosmology used to generate the mock data vector, whilst the colour maps the percentage difference in the standard deviation of the S_8 marginal posterior. The black-dashed lines depict the fiducial cosmology values, whilst the dotted grey lines that span the figure diagonally depict lines of constant S_8. The standard deviation of the S_8 marginal posterior is up to 20 per cent different to the case where the data cosmology aligned with the fiducial cosmology.

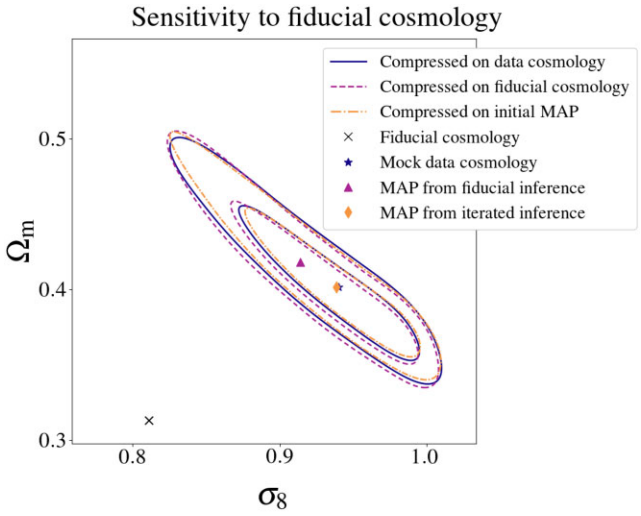

Figure 4. SBI analysis where the mock observed data were generated from a cosmology (blue star) that deviated far from the choice of fiducial cosmology (black cross). The MAP was found from this initial fiducial inference (fuschia triangle) with corresponding posterior contours (fuchsia dashed line), and used to re-compress the data vectors which resulted in new posterior contours (yellow dash–dotted line). The MAP from this once iterated inference is depicted by the yellow diamond. Inference performed with summaries that were compressed with the cosmology used to generate the mock observed data are shown by the solid blue line.

Fig.4 depicts this process using the cosmology depicted by the top right-hand corner in Fig. 3, where the difference in S_8 standard deviation between inference performed with compression on the fiducial cosmology and inference performed with compression on the mock data cosmology was 19 per cent. This was the worst-case scenario that we encountered in our testing. After compressing the data on the fiducial cosmology, we infer the MAP and re-perform the compression to obtain new posterior contours with an S_8 standard deviation that is now only 5 per cent different to that of inference performed with compression on the mock data cosmology. This process can be iterated several times if required. Furthermore, all of the MAP values have very similar S_8, but the MAP from the once-iterated inference is also very close to the true σ_8 and Ω_m values.

It should be noted that re-performing the `PyDELFI` inference with a new cosmology is computationally inexpensive as we can make use of all of the previously run simulations to perform the fiducial inference. There is only a small amount of computational cost associated with calculating data vector derivatives with respect to cosmology at the first iteration MAP, and also to re-calculate the score compression followed by training a new set of NDEs. As such, we would always recommend to perform at least one iteration and see if the MAP or standard deviation varies significantly, and to continue iterating until neither the MAP nor the standard deviation vary by much depending on the amount of noise present. This result shows that the SBI methodology can easily be made robust towards the choice of fiducial cosmology after just one inference iteration.

4.3 SBI sensitivity to quality of compression

As discussed in Section 2.3, a crucial step in the SBI pipeline is the compression. Whilst for our testing we were able to make use of optimal compression by using the same covariance to both draw data vector values and in the compression, this will not generally be the case. This is as an analytical covariance may not always be available and a covariance matrix would also be unable to completely capture the non-Gaussian features of forward simulated data. Therefore, there is a need to test the methodology in lieu of lossy compression, which we do by tampering with the covariance used in compression, artificially worsening it.

We tamper with the covariance in two ways. In the first method, as our analytic sample data covariance matrix follows a Wishart distribution (Wishart 1928; Taylor, Joachimi & Kitching 2013), we draw random samples of the covariance from a Wishart distribution constructed from our data covariance whilst varying the degrees of freedom. By reducing the degrees of freedom, we increase the amount of noise in the covariance like if we were estimating the covariance numerically from data samples, with lower degrees of freedom corresponding to fewer data samples. In the second method of tampering with the covariance, we suppress the off-diagonal elements of the covariance by a factor of 10^{-l}, where l denotes the diagonal distance from the diagonal element. This has the effect of destroying all of the cross-correlation information within the compression. This approach of tampering the covariance was chosen to ensure that the covariance structure would be significantly compromised but retaining its positive definiteness; it does not mimic any physical effect in the covariance modelling.

Fig.5 depicts the posterior contours obtained through the SBI pipeline using these artificially worsened compression methods. We can see from this figure that in all but the worst Wishart tampered case where the degrees of freedom was set to 312, just above the degrees of freedom limit of 270 to keep the covariance matrix invertible, the posteriors obtained through SBI do not differ greatly from the case with good compression. In the realistic Wishart tampered case, the degrees of freedom was set to 1000, a reasonable number of forward simulations one might perform to estimate a data covariance. However, even in the worst-case scenario, we can see that the effect is almost entirely posterior widening and mostly only on the A_{IA} parameter.

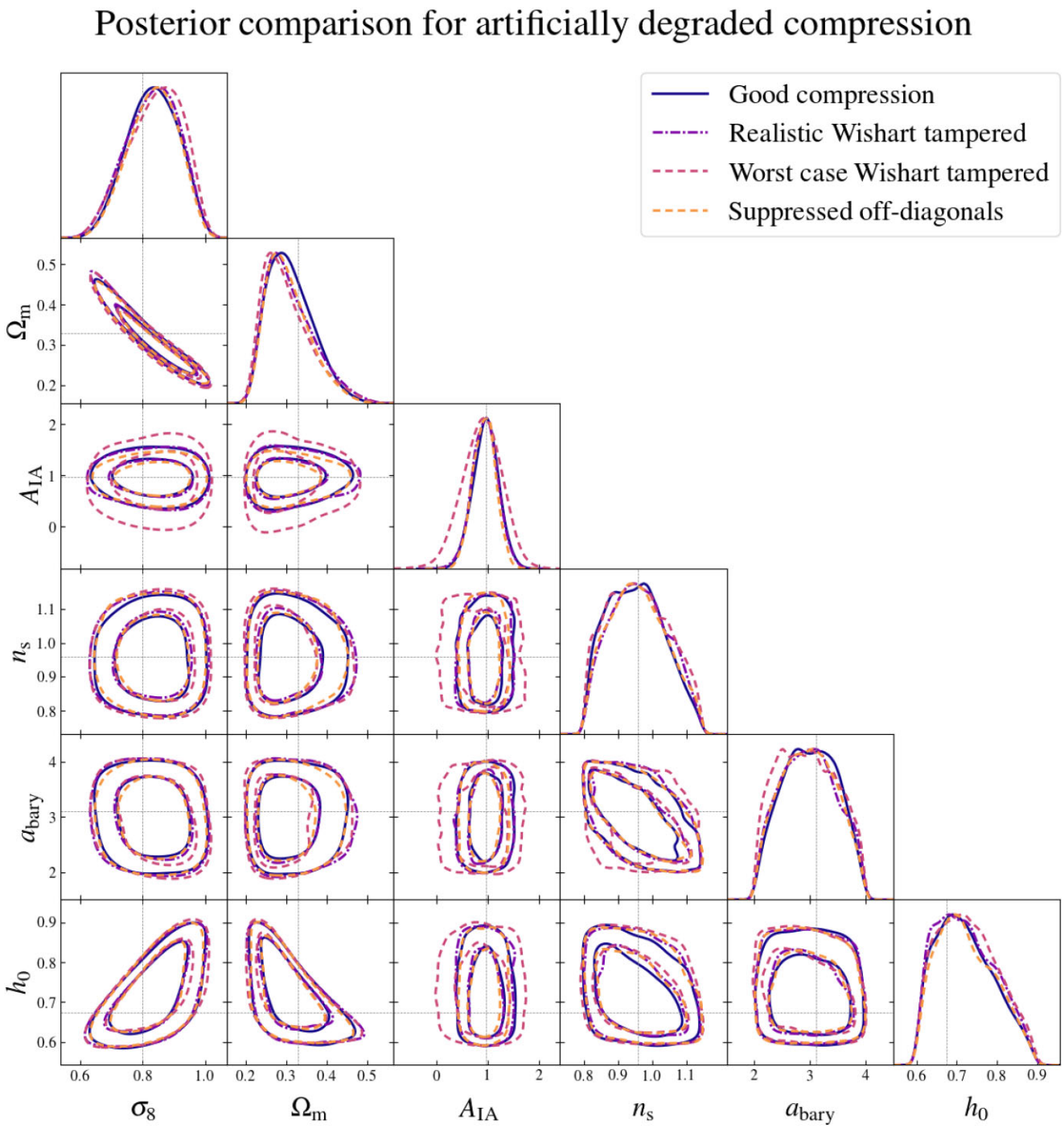

Figure 5. Posterior contours obtained through the SBI pipeline using three artificially worsened compression covariances plotted on top of the posterior obtained from using a good compression covariance (solid blue). In the realistic Wishart tampered case (purple dash–dotted), a covariance for use in compression was resampled from a Wishart distribution with the degrees of freedom set to 1000, a reasonable number of forward simulations one might perform to estimate a data covariance. In the worst-case Wishart tampered case (pink dashed), the Wishart degrees of freedom was set to 312, just above the limit of the degrees of freedom for our data covariance that is 270 to keep the matrix invertible (Taylor et al. 2013). In the suppressed off-diagonals case (orange dashed), the covariance had its off-diagonal elements suppressed by a factor of 10^{-l}, where l denotes the diagonal distance from the diagonal element. This strongly suppresses all of the cross-correlation information within the compression. The dashed grey lines depict the cosmology used to generate the mock observed data. It should be noted that in all but the worst Wishart-tampered case where only the A_{IA} contours widen, the SBI method is able to obtain a posterior almost identical to the good compression case.

It is clear that this SBI pipeline is robust towards lossy compression and can infer good posteriors under such circumstances. This sensitivity test also indicates that making use of a numerically estimated, and thus noisy covariance would suffice for purposes of re-analysing KiDS-1000.

4.4 Optimization

We wish to know how many potentially expensive forward simulations are required to successfully make use of this SBI methodology. We find that the active learning approach requires fewer simulations

to produce a well learned and stable posterior. However, there are downsides: the networks go through more rounds of training overall, as they are retrained each time a new set of simulations is acquired. In terms of training speed, however, retraining PyDELFI NDEs is faster than even the simplified simulations that we are making use of here. This indicates that they will be much faster than the more realistic simulations we will make use of in the re-analysis of KiDS-1000 data. To elaborate, the simulations we are running as outlined in Section 3 take place on the order of a minute per realization whilst training an entire PyDELFI model on a modern CPU with no GPU acceleration takes around 10 to 15 min. This shows us that the speed of analysis is dominated by the time it takes to run the forward simulations.

As we have mentioned previously in Section 2.2, we draw initial parameter points at which to run the forward simulations using a latin hypercube. We found that the key benefit of using a latin hypercube is that the tails of the parameter space are well explored. However, just relying on a latin hypercube led to the peaks of the posterior lacking the number of simulations required to converge. As such, the active learning mode of PyDELFI draws further parameter points at which to run forward simulations from a weighted mix of the intermediate posterior and prior. We wished to see the impact on the training by drawing parameter points in this manner, hence we compared the results of running the active learning approach against a single large latin hypercube with the same number of total simulations.

There are several metrics that we can use to determine the number of forward simulations we need to obtain good posterior contours. One method is to look at the spread in the learned S_8 posterior across each of the individual NDEs in the ensemble that was employed by PyDELFI. Another method is to check if the marginal S_8 posterior's standard deviation has converged. We found that both of these metrics were not strongly conclusive, and regardless of the number of simulations, neither metric appreciably deteriorated. As an alternative, we turned to the validation loss of the neural networks to see when the loss stopped decreasing when increasing the number of simulations. Fig. 6 depicts such loss curves, comparing both the validation loss from the active learning mode as well as a latin hypercube with the same number of samples. We find that the active learning approach always outperforms the latin hypercube batch run mode.

To test for the number of simulations required, however, we can look at where the steepness of the log loss curves starts to plateau. When the log loss plateaus, we can deduce that the networks are no longer able to glean much information from adding on further simulations. We can see that for the 12-parameter case, it is around 10 000 simulations that the loss curve starts to plateau. For the seven-parameter case, this takes place around 8000 simulations whilst for five parameters closer to 7000. This shows us that the number of simulations required scales with the number of parameters being inferred. However, as the scaling of simulations required versus parameters inferred is not polynomial, it is relatively cheap to infer more parameters.

Furthermore, it is of note that after the initial hypercube, the active learning method rapidly improves its quality of training. To graphically depict this, Fig. 7 plots the intermediate posteriors overlaid on top of each other. The δ_z parameters have been marginalized out just to make the plot more visually clear, but they were still learned in this particular run. We can see from this plot that the posterior quickly converges with small increases in simulation number, starting from a poor posterior constraints with the 2000 parameter points hypercube. Yet, the 9000 simulations intermediate posterior is almost identical

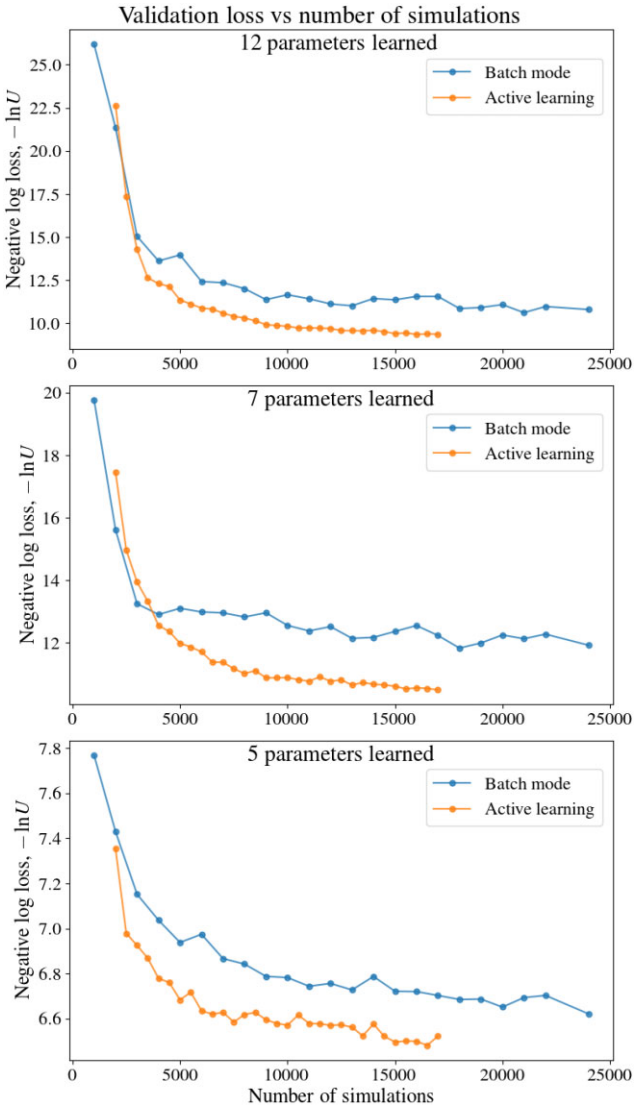

Figure 6. The plot depicts validation loss against the number of simulations for the two modes that PyDELFI can be run in. The orange points are the validation loss from *active learning* whilst the blue points depict the *batch run* mode, both modes are outlined in Section 2.1. For all simulations, all 12 parameters were varied. In the top panel, PyDELFI also inferred all 12 parameters, whilst in the middle panel PyDELFI only inferred seven parameters, which were $\{\sigma_8, \omega_c, A_{IA}, a_{bary}, n_s, h_0, \omega_b\}$. The bottom panel depicts a scenario where PyDELFI inferred five parameters, which were $\{\sigma_8, \omega_c, A_{IA}, a_{bary}, n_s\}$. For more than $\mathcal{O}(10^4)$ simulations, the gain in information becomes minimal, with the exact number of simulations dependent on the number of parameters inferred.

to the one obtained after 17 000 forward simulations. This serves to further reinforce what we see in the loss curves depicted in Fig. 6.

Within the setup of PyDELFI itself, however, we also tried tuning other hyperparameters, such as the number of epochs, i.e. the number of training rounds the NDEs undergo; the learning rate, a parameter that determines the amount the NDEs changes after each training epoch; the early stopping threshold, a parameter that determines when to stop training to guard from overfitting. We found the settings that PyDELFI uses by default to be good, with tweaks providing little to no improvement. The only hyperparameter we changed was the number of epochs. For high dimensionality inference, the number

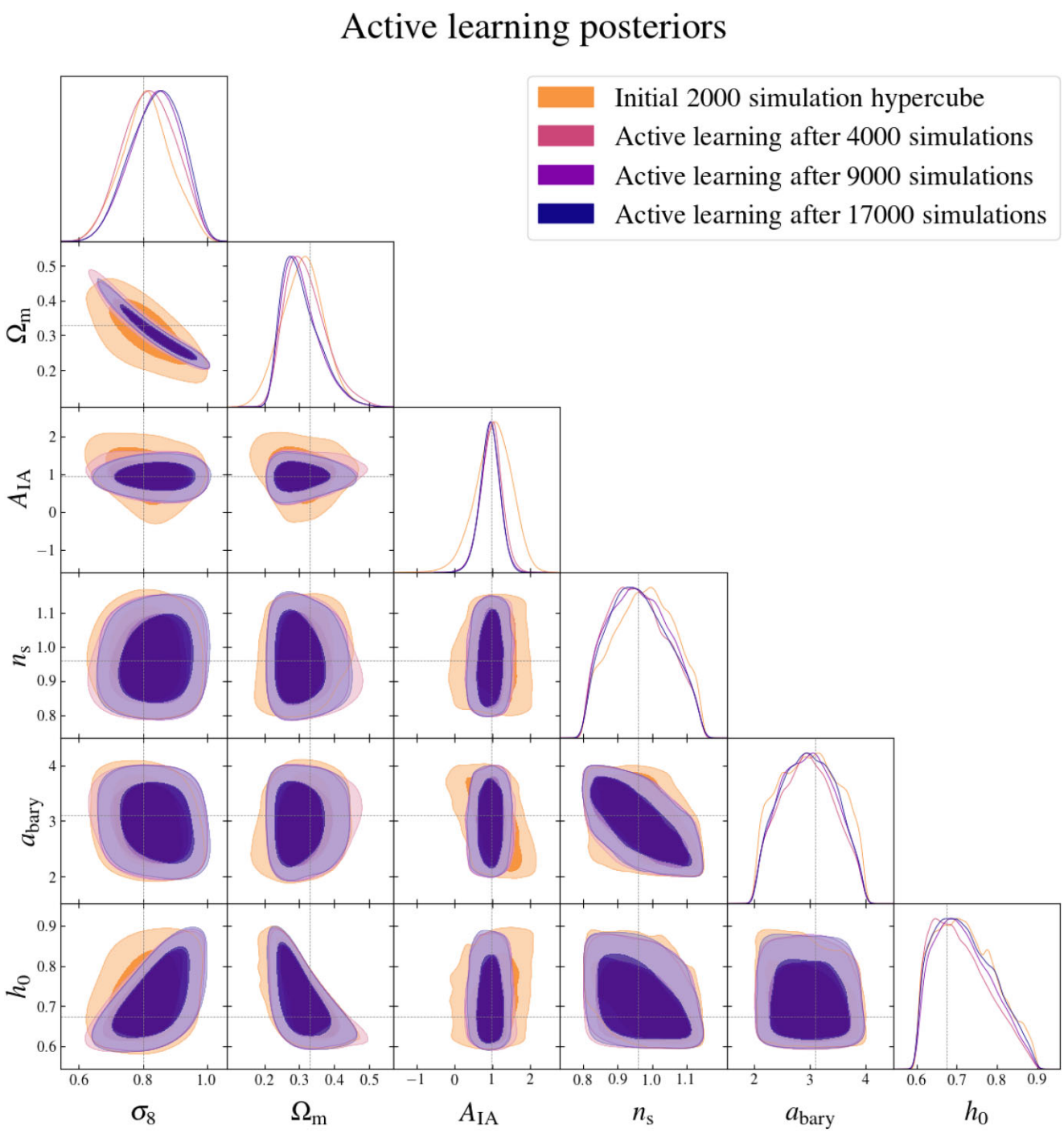

Figure 7. Intermediate posteriors from the active learning mode in `PyDELFI` for 2000 hypercube samples (orange), 4000 samples (fuchsia), 9000 samples (purple), and 17 000 samples (blue). The dashed grey lines depict the cosmology used to generate the mock observed data. The posterior is poorly approximated after the initial 2000 simulations, yet improves rapidly with just a doubling of simulation number, and after 9000 simulations are almost identical to the posterior obtained after 17 000 simulations.

of epochs needed to be set high enough to give the NDEs enough training rounds to learn the features of the data fully.

Throughout this optimization testing, the full set of KiDS-1000 parameters was varied; however, we also tried marginalizing out the other parameters in the compression step to see if that made the learning task easier. This is prudent to try as it both lowers the dimensionality of the problem and also guided by our knowledge of the cosmological setup, we know that many of the parameters are almost purely prior-driven. For our setup, however, given that there was little sensitivity to the data in the parameters that we wished to marginalize, performing nuisance-hardened compression as outlined in Alsing & Wandelt (2019) made no difference.

If the `MOPED` compression schema was used in its original Gram–Schmidt orthogonalization form Heavens et al. (2000), then the sampling distribution of summary statistics would have a covariance structure of that of a unitary matrix. This, in principle, should be

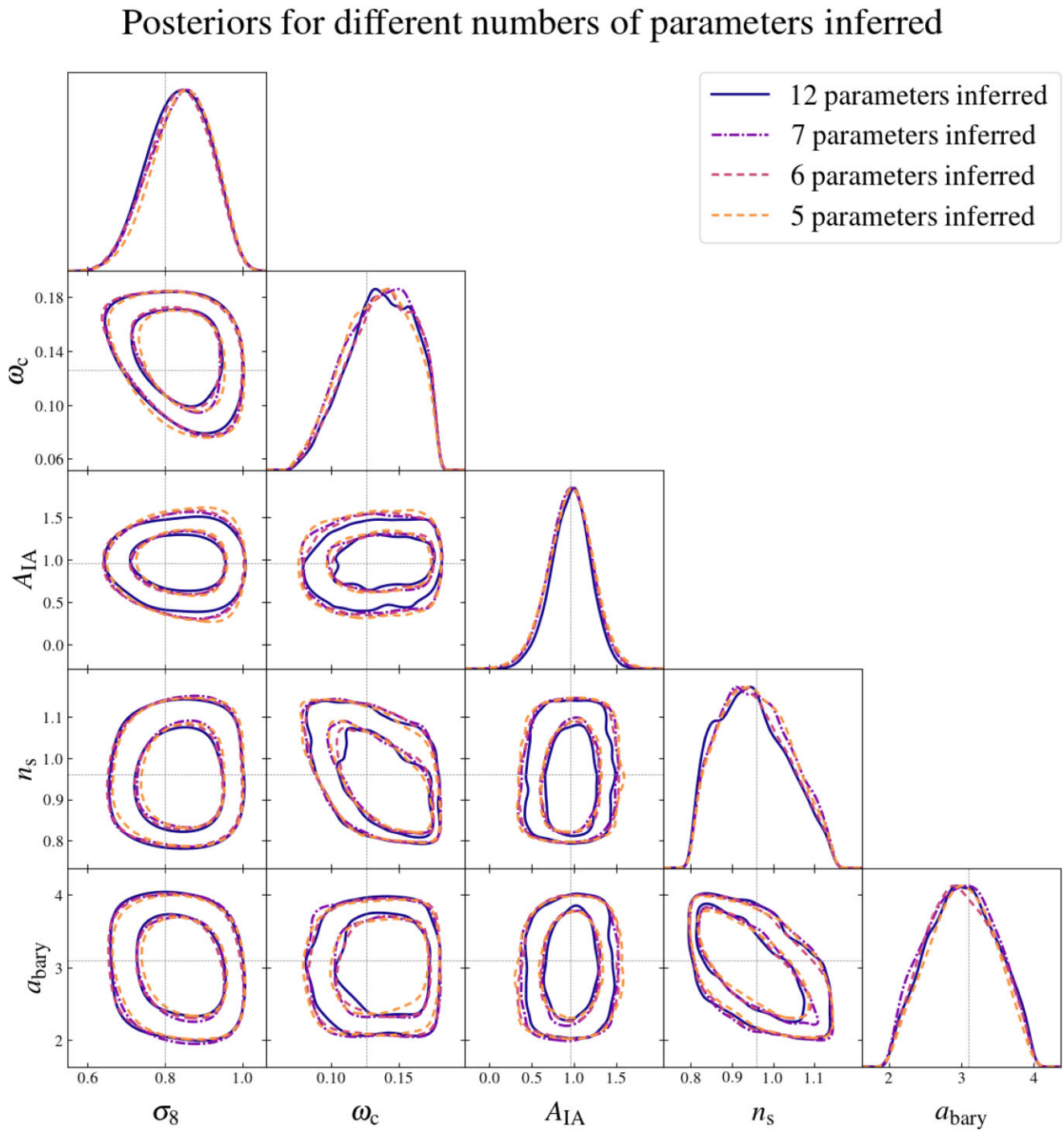

Figure 8. Marginal posteriors from the active learning mode in `PyDELFI` for varying numbers of parameters inferred. All 12 parameters were varied in the forward simulations, and also `PyDELFI` inferred all 12 parameters in the solid blue contour. The dashed grey lines depict the cosmology used to generate the mock observed data. For the seven-parameter inference (dash–dotted purple), the $\delta_z[5]$ parameters were not inferred, the six-parameter inference (fuchsia dashed) also dropped ω_b and the five-parameter inference further dropped h_0.

more straightforward for the NDEs to learn, but would require modifications to the current initialization schema that makes use of a Fisher matrix requiring the summary statistics to be cast into quasi maximum-likelihood estimators.

We also wished to test if reducing the number of parameters inferred would artificially narrow or widen the constraints on the parameters being inferred. We found that there was no such effect on the final parameter posteriors, meaning it is safe to reduce the number of parameters inferred but keep them varied in the forward simulations. The marginal posteriors for this testing are depicted in Fig. 8.

5 CONCLUSIONS

As we enter an era of high-precision cosmology, it will become increasingly difficult to reliably extract information from complex

data via analytic models and likelihoods. Therefore, we explored simulation-based inference (SBI) as a methodology that would enable forward-modelling and avoidance of a Gaussian likelihood assumption as is common in most cosmological analyses. We tested SBI on the full 12-dimensional parameter space of the most recent KiDS cosmological analysis of tomographic weak gravitational lensing data (KiDS-1000), assuming a Gaussian data vector to validate the SBI methodology, employing density estimation likelihood-free inference (DELFI) using the `PyDELFI` software package.

We demonstrated that our SBI method accurately recovers the full cosmological posterior of the KiDS-1000 analysis when applied to a mock data vector drawn from a Gaussian likelihood. This was achieved with under 10^4 forward simulations. Moreover, we showed that the necessary maximal data compression step in our method is robust to employing an inaccurate data covariance, and readily made robust towards the choice of fiducial parameter values. This suggests that our approach will still perform well when using approximate analytic covariances or noisy numerical covariance estimates.

Furthermore, we found the most computationally efficient mode in which to run `PyDELFI` to be an initial latin hypercube of parameter values followed by additional batches determined by *active learning*. Marginalizing parameters that are varied in the forward simulations in the score compression also does not bias the parameter constraints on the remaining parameters. This allows the number of forward simulations required to be further reduced if certain parameters are not of interest for the posterior.

The tests for robustness we have performed for our SBI method show that it is competitive for performing accurate parameter inference all whilst dropping the Gaussian likelihood assumption or being restricted to analytic models of the data. If paired with fast yet comprehensive simulations, SBI inference will also not dramatically increase computational requirements. In forthcoming work, we will back-end our SBI pipeline with realistic forward simulations of weak lensing and apply this inference pipeline to KiDS-1000 (von Wietersheim-Kramsta, Lin, et al., in preparation). While we have restricted our analysis to two-point statistics to be able to validate against the standard inference approach, SBI is readily applicable to any combination of summary statistics that are accurately represented in the simulations, which makes it a powerful tool to extract maximal amounts of information from next-generation cosmological surveys that contain more non-Gaussian information.

ACKNOWLEDGEMENTS

This work was partially enabled by funding from the UCL Cosmoparticle Initiative. MWK thanks the Science and Technology Facilities Council (STFC) for support in the form of a PhD Studentship. BJ acknowledges support by STFC Consolidated Grant ST/V000780/1. For the purpose of open access, the authors have applied a creative commons attribution (CC BY) licence to any author-accepted manuscript version arising.

DATA AVAILABILITY

The code will be made publicly available upon acceptance. The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

Ade P. A. et al., 2016, Astron. Astrophys., 594, A13
Aghanim N. et al., 2020, Astron. Astrophys., 641, A6
Akeret J., Refregier A., Amara A., Seehars S., Hasner C., 2015, J. Cosmol. Astropart. Phys., 2015, 043
Alsing J., Wandelt B., 2018, MNRAS, 476, L60
Alsing J., Wandelt B., 2019, MNRAS, 488, 5093
Alsing J., Charnock T., Feeney S., Wandelt B., 2019, MNRAS, 488, 4440
Amon A., Efstathiou G., 2022, MNRAS, 516, 5355
Amon A. et al., 2022a, MNRAS, 518, 477
Amon A. et al., 2022b, Phys. Rev. D, 105, 023514
Asgari M. et al., 2021, Astron. Astrophys., 645, A104
Bridle S., King L., 2007, New J. Phys., 9, 444
Busch J. et al., 2022, Astron. Astrophys., 664, 1
Charnock T., Lavaux G., Wandelt B. D., 2018, Astrophysics Source Code Library, recordascl:1804
Coe D., 2009, preprint (arXiv:0906.4123)
Fluri J., Kacprzak T., Refregier A., Amara A., Lucchi A., Hofmann T., 2018, Phys. Rev. D, 98, 123518
Fluri J., Kacprzak T., Lucchi A., Schneider A., Refregier A., Hofmann T., 2022, Phys. Rev. D, 105, 083518
Foreman-Mackey D., Hogg D. W., Lang D., Goodman J., 2013, Publ. Astron. Soc. Pac., 125, 306
Friedman J. H., 2003, Statistical Problems in Particle Physics, Astrophysics, and Cosmology, 1, Citeseer, p. 311
Germain M., Gregor K., Murray I., Larochelle H., 2015, International Conference on Machine Learning. PMLR, p. 881
Gupta A., Matilla J. M. Z., Hsu D., Haiman Z., 2018, Phys. Rev. D, 97, 103515
Hahn C. et al., 2022, J. Cosmol. Astropart. Phys., 2023, 31
Heavens A. F., Jimenez R., Lahav O., 2000, MNRAS, 317, 965
Heymans C. et al., 2021, Astron. Astrophys., 646, A140
Howlett C., Lewis A., Hall A., Challinor A., 2012, J. Cosmol. Astropart. Phys., 2012, 027
Ishida E. E. et al., 2015, Astron. Comput., 13, 1
Jeffrey N., Alsing J., Lanusse F., 2021, MNRAS, 501, 954
Jennings E., Madigan M., 2017, Astron. Comput,, 19, 16
Joachimi B. et al., 2021, Astron. Astrophys., 646, A129
Kilbinger M., 2015, Rep. Prog. Phys., 78, 086901
Leclercq F., 2018, Phys. Rev. D, 98, 063511
Lewis A., Challinor A., Lasenby A., 2000, ApJ, 538, 473
Loh W.-L., 1996, Ann. Stat., 24, 2058
Lopez-Paz D., Oquab M., 2016, preprint (arXiv:1610.06545)
Lueckmann J.-M., Bassetto G., Karaletsos T., Macke J. H., 2019, Symposium on Advances in Approximate Bayesian Inference, PMLR, p. 32
Mandelbaum R., 2018, Annu. Rev. Astron. Astrophys., 56, 393
Marin J.-M., Pudlo P., Robert C. P., Ryder R. J., 2012, Stat. Comput., 22, 1167
Mead A. J., Peacock J. A., Heymans C., Joudaki S., Heavens A. F., 2015, MNRAS, 454, 1958
Miller B. K., Cole A., Forré P., Louppe G., Weniger C., 2021, Truncated marginal neural ratio estimation, Advances in Neural Information Processing Systems, vol. 34, p. 129
Nelder J. A., Mead R., 1965, Comput. J., 7, 308
Papamakarios G., Murray I., 2016, Fast ε-free inference of simulation models with bayesian conditional density estimation, Advances in Neural Information Processing Systems, 29
Papamakarios G., Pavlakou T., Murray I., 2017, Advances in Neural Information Processing Systems, 30
Papamakarios G., Sterratt D., Murray I., 2019, The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, p. 837
Park J.-S., 1994, J. Stat. Plan. Inference, 39, 95
Porqueres N., Heavens A., Mortlock D., Lavaux G., 2021a, MNRAS, 502, 3035
Porqueres N., Heavens A., Mortlock D., Lavaux G., 2021b, MNRAS, 509, 3194
Prangle D., 2017, Bayesian Anal., 12, 289
Pritchard J. K., Seielstad M. T., Perez-Lezaun A., Feldman M. W., 1999, Mol. Biol. Evol., 16, 1791
Ribli D., Pataki B. Á., Zorrilla Matilla J. M., Hsu D., Haiman Z., Csabai I., 2019, MNRAS, 490, 1843

Rubin D. B., 1984, Ann. Stat., 12, 1151
Schneider P., Hartlap J., 2009, Astron. Astrophys., 504, 705
Secco L. et al., 2022, Phys. Revi. D, 105, 023515
Sellentin E., Heavens A. F., 2018, MNRAS, 473, 2355
Sellentin E., Heymans C., Harnois-Déraps J., 2018, MNRAS, 477, 4879
Smyth P., Wolpert D. H., 1998, An Evaluation of Linearly Combining Density Estimators via Stacking. Information and Computer Science, University of California, Irvine
Smyth P., Wolpert D., 1999, Mach. Learn., 36, 59
Stein M., 1987, Technometrics, 29, 143
Sugiyama S. et al., 2022, Phys. Rev. D, 105, 123537
Taylor A., Joachimi B., Kitching T., 2013, MNRAS, 432, 1928
Taylor P. L., Kitching T. D., Alsing J., Wandelt B. D., Feeney S. M., McEwen J. D., 2019, Phys. Rev. D, 100, 023519
Tegmark M., Taylor A. N., Heavens A. F., 1997, ApJ, 480, 22
Upham R. E., Brown M. L., Whittaker L., 2021, MNRAS, 503, 1999
Uria B., Côté M.-A., Gregor K., Murray I., Larochelle H., 2016, J. Mach. Learn. Res., 17, 7184
Wishart J., 1928, Biometrika, 32 20A
Zuntz J. et al., 2015, Astron. Comput., 12, 45

This paper has been typeset from a TEX/LATEX file prepared by the author.