Hierarchical Inference of the Lensing Convergence from Photometric Catalogs with Bayesian Graph Neural Networks

Park, Ji Won; Birrer, Simon; Ueland, Madison; Cranmer, Miles; Agnello, Adriano; Wagner-carena, Sebastian; Marshall, Philip J.; Roodman, Aaron; LSST Dark Energy Science Collaboration

Published in:
Astrophysical Journal

DOI:
10.3847/1538-4357/acdc25

Publication date:
2023

Document version
Publisher's PDF, also known as Version of record

Document license:
CC BY

Citation for published version (APA):
Park, J. W., Birrer, S., Ueland, M., Cranmer, M., Agnello, A., Wagner-carena, S., Marshall, P. J., Roodman, A., & LSST Dark Energy Science Collaboration (2023). Hierarchical Inference of the Lensing Convergence from Photometric Catalogs with Bayesian Graph Neural Networks. Astrophysical Journal, 953(2), [178]. https://doi.org/10.3847/1538-4357/acdc25
Hierarchical Inference of the Lensing Convergence from Photometric Catalogs with Bayesian Graph Neural Networks

Ji Won Park1,2, Simon Birrer1,2, Madison Ueland1, Miles Cranmer3, Adriano Agnello4, Philip J. Marshall1,2, and Aaron Roodman1,2
the LSST Dark Energy Science Collaboration

1 Kavli Institute for Particle Astrophysics and Cosmology, Department of Physics, Stanford University, Stanford, CA 94025, USA; jp281@slac.stanford.edu
2 SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA
3 Princeton University, Princeton, NJ 08544, USA
4 DARK, Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark

Received 2022 November 14; revised 2023 June 3; accepted 2023 June 5; published 2023 August 16

Abstract

We present a Bayesian graph neural network (BGNN) that can estimate the weak lensing convergence (κ) from photometric measurements of galaxies along a given line of sight (LOS). The method is of particular interest in strong gravitational time-delay cosmography (TDC), where characterizing the “external convergence” (κext) from the lens environment and LOS is necessary for precise Hubble constant (H0) inference. Starting from a large-scale simulation with a κ resolution of ~1′, we introduce fluctuations on galaxy–galaxy lensing scales of ~19″ and extract random sight lines to train our BGNN. We then evaluate the model on test sets with varying degrees of overlap with the training distribution. For each test set of 1000 sight lines, the BGNN infers the individual κ posteriors, which we combine in a hierarchical Bayesian model to yield constraints on the hyperparameters governing the population. For a test field well sampled by the training set, the BGNN recovers the population mean of κ precisely and without bias (within the 2σ credible interval), resulting in a contribution to the H0 error budget well under 1%. In the tails of the training set with sparse samples, the BGNN, which can ingest all available information about each sight line, extracts a stronger κ signal compared to a simplified version of the traditional method based on matching galaxy number counts, which is limited by sample variance. Our hierarchical inference pipeline using BGNNs promises to improve the κext characterization for precision TDC. The code is available as a public Python package, NODE TO JOY.

Unified Astronomy Thesaurus concepts: Cosmology (343); Hubble constant (758); Hierarchical models (1925); Bayesian statistics (1900); Neural networks (1933); Weak gravitational lensing (1797); Strong gravitational lensing (1643); Astronomical simulations (1857)

1. Introduction

Galaxies in the sky may appear to us distorted in shape and magnified (or de-magnified) in brightness. The responsible phenomenon is gravitational lensing: massive structures along our line of sight (LOS) curve spacetime, thereby gravitationally bending the path of light as it travels to us from the background source. When measured from a large sample of galaxies, the lensing distortions probe the total matter distribution in the Universe, both luminous and dark. A quantity of particular importance in cosmology and galaxy evolution studies is the convergence (κ), defined as the total integrated mass along an LOS weighted by the lensing efficiency, which peaks for mass located roughly midway between us and the source.

Reconstructing a reliable map of the density field κ across the sky, or mass mapping, can inform the relationship between luminous matter and dark matter, known as the galaxy–halo connection. Mass maps have been jointly analyzed with maps of stellar mass, galaxies, and galaxy clusters to constrain the galaxy bias—the statistical relation between the distribution of galaxies and matter (e.g., Chang et al. 2016; see Wechsler & Tinker 2018 for a review).

Mass maps can also improve weak lensing cosmology when used alongside shear maps. Traditional analyses using two-point correlations of the directly measurable shear field are limited to modeling zero-mean Gaussian random fields in the matter density. Extending to higher-order statistics can yield much more information about structure formation, including any non-Gaussianities in the early Universe that would propagate at large scales into the late Universe, as well as those that may arise at small scales later due to nonlinear gravitational collapse (Jeffrey et al. 2018). For reconstructing wide-field mass maps, variants of the direct shear-to-convergence inversion algorithm by Kaiser & Squires (1993), “KS,” are commonly used (Van Waerbeke et al. 2013; Chang et al. 2015; Liu et al. 2015; Vikram et al. 2015; Chang et al. 2018; Oguri et al. 2018). The KS method does not account for missing data or noise, so there have also been efforts to bring in different priors about the convergence field (e.g., Wiener 1949; Leonard et al. 2014; Lanusse et al. 2016). A related line of work deals with magnification rather than convergence (e.g., Gunnarsson et al. 2006; Ménard et al. 2010; Morrison et al. 2012).

Most sight lines in the Universe are only weakly lensed. But in the rare event that a halo lines up in front of a source, the lensing is strong enough that the source appears copied into multiple images. The photons from each image arrive at our telescope with relative time delays. If the strongly lensed
source is time-variable, e.g., a quasar or a supernova, the time delays can be measured and fed into a cosmographic analysis to measure the Hubble constant, $H_0$ (Refsdal 1964). Typically in time-delay cosmography (TDC), the external lensing perturbations are approximated as a quadrupole lens parameterized by external convergence ($\kappa_{\text{ext}}$) and shear. The external shear can be constrained by modeling the lenses from imaging data, but $\kappa_{\text{ext}}$ cannot—a manifestation of the mass-sheet degeneracy (MSD; Falco et al. 1985). We therefore require external tracers of the density field in addition to the strong lens modeling.

Suyu et al. (2010) and Fassnacht et al. (2011) pioneered the method of estimating $\kappa_{\text{ext}}$ based on galaxy number counts. In brief, the method involves (i) counting the galaxies around a lens system, (ii) comparing the resulting counts against those of a control field, e.g., the Cosmic Evolution Survey (COSMOS) field (Scoville et al. 2007) and the Canada–France–Hawaii Telescope Lensing Survey (Heymans et al. 2012), to obtain relative counts, and (iii) selecting lines of sight of similar relative counts, along with their associated convergence values, from a numerical $N$-body simulation with an assumed galaxy–halo connection. The matched $\kappa$ samples can then be interpreted as $\kappa_{\text{ext}}$ posterior samples for a sight line, where the data for the sight line has been reduced to number counts as a summary statistic. This method of matching summary statistics rather than evaluating a specific likelihood function can be cast as approximate Bayesian computation (ABC; Birrer et al. 2019). Note that the galaxy–halo connection encoded in the $N$-body simulation and the semi-analytic galaxy evolution model is a key source of uncertainty in the $\kappa$ constraints.

Since then, the method has evolved to accommodate summary statistics beyond simple galaxy counts. Suyu et al. (2013) used the external shear inferred from lens modeling as an additional constraint. Greene et al. (2013) explored schemes to weight number counts by the galaxy redshift, stellar mass, and projected separation from the LOS. The additional information improved the $\kappa_{\text{ext}}$ constraints for the most overdense lines of sight but helped little for the less-dense ones. They found that the underdense sight lines yielded the best $\kappa_{\text{ext}}$ constraints, at a residual uncertainty of $\sigma_{\kappa_{\text{ext}}} \approx 0.03$, corresponding to an uncertainty on $H_0$ comparable to that from lens modeling. Alternatively, Collett et al. (2013) reconstructed $\kappa_{\text{ext}}$ using a galaxy–halo model, calibrating for the effect of dark structures and voids. Similarly to Greene et al. (2013), they also reported the most precise estimates for environments with low $\kappa_{\text{ext}}$.

McCully et al. (2014) devised the “flexion shift” metric quantifying the contribution of a line-of-sight object to the lens potential. For the ground-truth $\kappa_{\text{ext}}$, all of these studies used the 1.7 deg$^2$ mock sky maps of $\kappa_{\text{ext}}$ obtained by raytracing through the Millennium Simulation (Springel et al. 2005; Hilbert et al. 2009). The analysis methods shed light on the spectroscopic data previously collected around strong lenses for line-of-sight assessment (e.g., Fassnacht et al. 2002, 2006; Momcheva et al. 2006).

To date, seven lensed quasars have undergone cosmographic analysis following a fairly standard procedure. The $\kappa_{\text{ext}}$ constraints were generated on an individual lens level by matching relative number counts, both with and without weighting by the inverse projected separation $(1/r)$; Greene et al. 2013; Rusu et al. 2017; Chen et al. 2019; Buckley-Geer et al. 2020; Rusu et al. 2020). The H0LiCOW and SHARP collaborations measured $H_0$ from six lenses with the median $\kappa_{\text{ext}}$ ranging from $-0.006$ (PG 1115+080) to 0.1 (B1608+656; Suyu et al. 2010, 2014; Wong et al. 2017; Birrer et al. 2019; Chen et al. 2019; Wong et al. 2020; Rusu et al. 2020). The uncertainty contribution of $\kappa_{\text{ext}}$ to the individual $H_0$ error budget ranged from $\sim$2.7% to $\sim$6.5%. The STRIDES collaboration measured $H_0$ to 3.9% precision from one additional lens inside a particularly underdense environment, of estimated $\kappa_{\text{ext}} \sim -0.04$ (Shajib et al. 2020).

The joint collaboration consisting of the H0LiCOW, COSMOCRAIL, STRIDES, SHARP, and COSMICLENS projects (hereafter the TDCOSMO collaboration) currently constrains $\kappa_{\text{ext}}$ for each lens independently when inferring $H_0$ from the combined lens sample. With this standard procedure, TDCOSMO achieved $\sim$ 2% precision on $H_0$ from the seven lenses (Chen et al. 2019; Shajib et al. 2019; Wong et al. 2020; Milon et al. 2020; Rusu et al. 2020). The uncertainty increased to $\sim$5% when the lens mass models were made significantly more flexible (Birrer et al. 2020). In relaxing assumptions on the lens mass model, Birrer et al. (2020) marginalized over the population offset in the mass sheet that is internal to the main deflector, which manifests as a transform of its mass profile. While these analyses did not hierarchically account for the external aspect of the mass sheet due to line-of-sight structure, the interpretation of the mass sheet can be extended to include external effects.

Hierarchically inferring the population mean of $\kappa_{\text{ext}}$ and marginalizing over it is important, because any bias in the assumed population mean of $\kappa_{\text{ext}}$ directly biases $H_0$. Note that the mean $\kappa_{\text{ext}}$ may not necessarily vanish for an ensemble of lenses due to selection effects, e.g., lenses tend to lie in groups (Blandford et al. 2001), causing a slight preference for lens systems with overdense lines of sight (Fassnacht et al. 2011; Collett & Cunnungan 2016). Any discrepancy between our prior and the actual $\kappa_{\text{ext}}$ population can result in bias, so we must hierarchically infer the actual population statistics and account for it in TDC analysis. The Legacy Survey of Space and Time (LSST; Ivezić et al. 2019) at the Vera Rubin Observatory is predicted to discover tens of thousands of lenses (Collett & Cunnungan 2016), among them $\sim$ 8000 lensed quasars (Oguri & Marshall 2010). As we scale up to such a large sample, hierarchical inference of $\kappa_{\text{ext}}$ can combine the information from all sight lines, each with limited signal on $\kappa_{\text{ext}}$, to minimize systematic bias on $H_0$.

Although uncertainties in other components of TDC, such as lens modeling and time-delay measurements, have reduced over time, $\kappa_{\text{ext}}$ remains a significant source of uncertainty on $H_0$. The primary drawback of matching relative number counts to estimate $\kappa_{\text{ext}}$ is that the wealth of photometric observations for each galaxy—such as the measured magnitudes in each bandpass filter, positions, and shapes—becomes reduced to a few numbers for the whole sight line. We seek to extract information more efficiently by enlisting a neural network to process all available observations. Namely, we design a Bayesian graph neural network (BGNN) that can take in the measurements of a variable number of galaxies observed around a given LOS and output the full posterior probability density function (PDF) over $\kappa$. Training the BGNN on a galaxy catalog with associated $\kappa$ labels allows it to implicitly learn the distribution of dark matter in galaxies and clusters encoded in

\footnote{www.tdcosmo.org}
the underlying simulation. This precludes the need to manually
correct for the mass that is not in the galaxy or cluster halos or
assume overly simple models for the lensing mass, as was done
in Collett et al. (2013).

This paper seeks to improve the \( \kappa_{\text{ext}} \) estimation on three
fronts: (i) proposing the BGNN as a novel \( \kappa_{\text{ext}} \) estimation
engine, (ii) combining the BGNN constraints into hierarchical
inference of the population \( \kappa_{\text{ext}} \), and (iii) enhancing the lensing
scales of large-scale simulations to adequately describe lensing
effects on the scale of galaxy-scale strong lenses. We ask the
following questions:

1. Is the BGNN capable of accurately and precisely
   estimating \( \kappa \) for individual sight lines?
2. When propagated into a hierarchical inference frame-
   work, do the BGNN-inferred posteriors lend themselves
to precise and accurate recovery of hyperparameters
governing \( \kappa \) in the population? This investigation seeks
to quantify population-level selection effects in regimes
where insufficient information is available on an
individual sight-line basis.
3. How does the BGNN compare with the ABC technique
   of matching number counts?
4. Which environments lend themselves to better \( \kappa \)
   constraints? At what extremes does this method break?

To answer these questions, we train our BGNN on a realistic
galaxy catalog with associated \( \kappa \) labels and design a suite of
numerical experiments to validate our approach.

We organize this paper as follows. Section 2 provides an
introduction to TDC, with a focus on the impact of \( \kappa_{\text{ext}} \) on \( H_0 \)
inference. Section 3 details our hierarchical inference pipeline.
Section 4 reports our \( \kappa \) recovery results on 100 test sight lines.
Section 5 situates our findings within the context of TDC and
suggests next steps.

We reserve a bulk of the details about our simulations to
Appendices A and B. In Section A, we start with a primer on
the lensing formalism and describe how we use multiplane
raytracing to post-process an existing galaxy catalog and \( \kappa \)
map, produced in the weak lensing resolution, for strong
lensing effects on the scale of galaxy-scale strong lenses. We ask the
following questions:

1. Is the BGNN capable of accurately and precisely
   estimating \( \kappa \) for individual sight lines?
2. When propagated into a hierarchical inference frame-
   work, do the BGNN-inferred posteriors lend themselves
to precise and accurate recovery of hyperparameters
governing \( \kappa \) in the population? This investigation seeks
to quantify population-level selection effects in regimes
where insufficient information is available on an
individual sight-line basis.
3. How does the BGNN compare with the ABC technique
   of matching number counts?
4. Which environments lend themselves to better \( \kappa \)
   constraints? At what extremes does this method break?

To answer these questions, we train our BGNN on a realistic
galaxy catalog with associated \( \kappa \) labels and design a suite of
numerical experiments to validate our approach.

We organize this paper as follows. Section 2 provides an
introduction to TDC, with a focus on the impact of \( \kappa_{\text{ext}} \) on \( H_0 \)
inference. Section 3 details our hierarchical inference pipeline.
Section 4 reports our \( \kappa \) recovery results on 100 test sight lines.
Section 5 situates our findings within the context of TDC and
suggests next steps.

We reserve a bulk of the details about our simulations to
Appendices A and B. In Section A, we start with a primer on
the lensing formalism and describe how we use multiplane
raytracing to post-process an existing galaxy catalog and \( \kappa \)
map, produced in the weak lensing resolution, for strong
lensing applications. Section B describes the photometric
catalog input to our simulations.

Throughout Sections 4.1 and 4.2, we will present compar-
isons with the method of matching summary statistics, in order
to highlight the difference between a neural net and an ABC-
based algorithm. Our implementation of the summary statistics
matching does not represent the TDCOSMO implementation.
TDCOSMO uses orders-of-magnitude more sight lines, allowing
them to match simultaneously to two sets of summary
statistics, whereas we match to one summary statistic at a time.
Additionally, TDCOSMO incorporates external shear information
from the lens modeling to further constrain their \( \kappa_{\text{ext}} \). This is
not possible within our context, because we only operate on
sight lines without strong lensing.

We release with this publication the open-source Dark
Energy Science Collaboration (DESC) Python package NODE
TO JOY\(^5\) (Park et al. 2023). This package implements the
complete pipeline, including the structure-enhanced raytracing,
training and evaluation of the BGNN, comparison with
summary statistics matching, and the hierarchical inference
analysis.

\(^5\) http://github.com/jwontcpark/node-to-joy

2. Strong Lensing Time-delay Cosmography

Let us begin by reviewing the basic principles of TDC
(Refsdal 1964). When light rays from a distant source are
deflected by some foreground lens, the travel time from the
source to the observer depends on both their path lengths and
the gravitational potential they traverse. Assuming a single thin
lens, the excess time delay of an image at position \( \theta \) originating
from a source at position \( \beta \) relative to an unperturbed path is

\[
t(\theta, \beta) = \frac{D_{\Delta t}}{c} \phi(\theta, \beta).
\]

Here,

\[
\phi(\theta, \beta) = \frac{(\theta - \beta)^2}{2} - \psi(\theta)
\]

is the Fermat potential (Schneider 1985; Blandford &
Narayan 1986), defined for the lensing potential \( \psi(\theta) \), a two-
dimensional projection of the three-dimensional Newtonian
potential \( \Phi(\eta \theta, \eta) \):

\[
\psi(\theta) = 2 \int_0^\eta d\eta \frac{D(\eta - \eta')}{D(\eta)D(\eta')} \Phi(\eta \theta, \eta').
\]

\( D_{\Delta t} \) is the time-delay distance:

\[
D_{\Delta t} = (1 + z_d) \frac{D_{d} D_{s}}{D_{ds}},
\]

where \( z_d \) is the redshift of the deflector galaxy, and \( D_0, D_{\alpha}, \) and
\( D_{ds} \) are the angular diameter distances to the deflector, to the
source, and between the deflector and the source, respectively.

In the strong lensing regime, the deflection is strong enough
that we observe multiple images of the same source. Whereas
the time delay with respect to an unperturbed path in
Equation (1) cannot be measured, we can observe the relative
time delays between a pair of images, at positions \( \theta_1 \) and \( \theta_2 \), and measure

\[
\Delta t_{\theta_2} := t(\theta_1, \beta) - t(\theta_2, \beta)
\]

\[
= \frac{D_{\Delta t}}{c} (\phi(\theta_1, \beta) - \phi(\theta_2, \beta)).
\]

The time-delay distance \( D_{\Delta t} \) corresponds to the absolute scale
of the Universe, which is inversely proportional to the \( H_0 \), i.e.,
\( D_{\Delta t} \propto 1/H_0 \).

Additionally, structures along the LOS to the strong lens
introduce additional weak lensing perturbations. The resulting
convergence affects the time delays while keeping imaging
observables the same under linear transformations of the lens
equation—the so-called mass-sheet degeneracy (MSD). As
shown by Falco et al. (1985), remapping the reference mass
distribution \( \kappa \), for any scalar \( \lambda \), as

\[
\kappa_{\text{ext}}(\theta) = \lambda \kappa(\theta) + (1 - \lambda),
\]

while isotropically scaling the source plane as \( \beta \rightarrow \lambda \beta \) results
in the same dimensionless observables (image pixel values) but
different time delays. The infinite family of solutions to the lens
equation results in a range of inferred properties of the deflector
and the source.

The mass-sheet parameter \( \lambda \) in Equation (7) may be internal
to the main deflector, affecting its kinematics, or it may stem
from the external line-of-sight structure. It is common practice
to express the external portion \( \lambda_{\text{ext}} \) as an equivalent additional
mass sheet at the redshift of the main deflector with uniform surface mass density, called external convergence, and denoted \( \kappa_{\text{ext}} \) (Keeton et al. 1997; Schneider 1997). In terms of \( \lambda_{\text{ext}} \) we have \( \kappa_{\text{ext}} = 1 - \lambda_{\text{ext}} \).

If we were to use a model not accounting for \( \kappa_{\text{ext}} \) (i.e., fixing \( \kappa_{\text{ext}} = 0 \)) to infer the time-delay distance \( D_{\Delta t}^{(0)} \), then the true time-delay distance can be recovered using a separately constrained \( \kappa_{\text{ext}} \) as

\[
D_{\Delta t} = \frac{D_{\Delta t}^{(0)}}{1 - \kappa_{\text{ext}}}.
\]

In terms of \( H_0 \), this relation is

\[
H_0 = (1 - \kappa_{\text{ext}})H_0^{(0)}.
\]

Underestimating \( \kappa_{\text{ext}} \) thus leads to an upward bias on \( H_0 \) and vice versa.

The external convergence \( \kappa_{\text{ext}} \) is a product of three different convergence values (Birrer et al. 2020; Fleury et al. 2021):

\[
1 - \kappa_{\text{ext}} = (1 - \kappa_d)(1 - \kappa_s),
\]

where \( \kappa_d \) and \( \kappa_s \) correspond to the integrated convergence along the strong lens LOS from the observer to the source, from the observer to the strong lens, and from the strong lens to the source, respectively. These component convergence values transform the background angular diameter distances \( D_{\text{BG}} \) from the homogeneous background metric given by the cosmological model, without any perturbations, into the angular diameter distances along the specific LOS of the strong lens \( D_{\lambda/d/ds} \). That is,

\[
D_{\lambda/d/ds}^{\text{SL}} = (1 - \kappa)D_{\lambda/d/ds}^{\text{BG}}.
\]

We apply our method only to \( \kappa_s \) to account for the MSD, but it can be generalized to accommodate all three angular diameter distances.

Because each sight line contains limited information about \( \kappa_{\text{ext}} \), the inference of \( \kappa_{\text{ext}} \) necessitates a hierarchical approach. For a sample of \( N_{\text{lens}} \) lenses, we can apply the simplified and analytical error propagation in Birrer et al. (2022) to evaluate the contribution of \( \kappa_{\text{ext}} \) to the overall \( H_0 \) error budget. We decompose the \( H_0 \) uncertainty into two terms: one term capturing global shifts in the inferred \( H_0 \) due to the mass sheet \( \lambda \) and another term incorporating uncertainties introduced by individual lenses, including uncertainties from the time-delay measurements and the inferred Fermat potentials of main deflectors. The fractional uncertainty on \( H_0 \) can be written as

\[
\left( \frac{\sigma(H_0)}{H_0} \right)^2 \approx \left( \frac{\sigma(\bar{\lambda})}{\bar{\lambda}} \right)^2 + \left( \frac{\sigma(H_0^{(0)})}{H_0^{(0)}} \right)^2,
\]

where \( \bar{\lambda} \) refers to the population mean in \( \lambda \) and \( \sigma(\chi) \) to the 1σ uncertainty on the quantity \( \chi \).

Note that \( \bar{\lambda} \) of Equation (12) includes both the internal and external mass sheets. Other probes that are sensitive to both can also constrain \( \bar{\lambda} \) but would not allow such a separation in the physical interpretation. The contribution of the external portion \( \lambda_{\text{ext}} \) to term \( \odot \) of Equation (12) is

\[
\left( \frac{1 - \sigma(\bar{\lambda}_{\text{ext}})}{\bar{\lambda}_{\text{ext}}} \right)^2 = \left( \frac{\sigma(\bar{\kappa}_{\text{ext}})}{1 - \bar{\kappa}_{\text{ext}}} \right)^2,
\]

This represents the main contribution of \( \kappa_{\text{ext}} \) to the overall \( H_0 \) error budget. There is a smaller, second-order contribution that enters term \( \oslash \) in Equation (12):

\[
\left( \frac{\sigma(H_0^{(0)})}{H_0^{(0)}} \right)^2,
\]

where \( \sigma(\bar{\kappa}_{\text{ext}}) \) denotes the 1σ scatter in the population \( \kappa_{\text{ext}} \). This contribution scales inversely with \( N_{\text{lens}} \) and becomes subdominant to other sources of uncertainty, especially in the large-\( N_{\text{lens}} \) setting.

Suppose we estimate \( \bar{\kappa}_{\text{ext}} \) for the population mean in \( \kappa_{\text{ext}} \) but the truth lies offset from this, at \( \bar{\kappa}_{\text{ext,true}} \). It follows from Equation (9) that the fractional bias on \( H_0 \) introduced by an erroneous estimate \( \bar{\kappa}_{\text{ext}} \) is

\[
\text{Frac. bias on } H_0 = \frac{\bar{\kappa}_{\text{ext,true}} - \bar{\kappa}_{\text{ext}}}{1 - \bar{\kappa}_{\text{ext,true}}}.
\]

That is, the error on \( \bar{\kappa}_{\text{ext}} \) biases \( H_0 \) almost directly.

In this paper, we perform hierarchical inference to obtain the full posterior PDF over the population \( \kappa_{\text{ext}} \) statistics, \( \bar{\kappa}_{\text{ext}} \) and \( \sigma(\bar{\kappa}_{\text{ext}}) \) (see Section 3.4). Equation (13) and, to a lesser extent, Equation (14) allow us to contextualize our hierarchical constraints within the overall \( H_0 \) error budget.

## 3. Methods

In this section, we outline the methodology for the various stages of our \( \kappa \) inference pipeline. Figure 1 illustrates our pipeline as a flowchart and probabilistic graphical models (PGMs) for training and inference. In Section 3.1, we describe the simulated training data. In particular, we state the numerical choices made when postprocessing the existing galaxy catalog and a map of weak lensing \( \kappa \) to generate the training set suitable for strong lensing applications. Then, in Section 3.2, we explain how the BGNN extracts the \( \kappa \) posterior from individual sight lines. Alternatively, the traditional summary statistics method can be used to obtain the posteriors, as described in Section 3.3. Lastly, the \( \kappa \) posteriors, whether from the BGNN or the summary statistics matching, enter the hierarchical inference framework to yield constraints on hyperparameters governing the population; Section 3.4 describes this process.

### 3.1. Simulated Data

Because our problem concerns the inference of weak lensing \( \kappa \), our training set consists of a galaxy catalog queried around random weak lensing sight lines and the associated \( \kappa \) labels. Both the input catalog and target labels are derived from COSMOC202, a catalog based on a cosmological N-body simulation and a semianalytic galaxy evolution model (Korytov et al. 2019). The catalog contains the \( \kappa_{\text{WL}} \) computed in the weak lensing (WL) regime \( (\delta \sim 1\%) \). We perform additional raytracing on top of the COSMOC202 halos to introduce fluctuations on a finer angular \( (\delta \sim 1\%) \) resolution required for strong lensing studies. Appendix A.4.1 details this structure-enhanced raytracing procedure. The surface mass densities were estimated by rendering halos with the Navarro–Frenk–White mass profile (Navarro et al. 1997), described in Section A.4.2. Appendix B briefly summarizes the semianalytic
models used to generate the galaxy catalogs. We train our BGNN on the photometric information associated with COSMOC2 sight lines and the \( \kappa \) values that we postprocessed to the strong lensing resolution. The source redshift was fixed at \( z_{\text{src}} = 2 \), typical of lenses expected to be discoverable in upcoming wide-area surveys (Collett 2015). This was a choice also made in Li et al. (2021). Our network, as presented here, must be retrained for \( z_{\text{src}} \) departing significantly from 2. It is possible to condition on \( z_{\text{src}} \) or train in a multitask setting for the network to accommodate multiple values of \( z_{\text{src}} \); we leave these extensions to future work.

The goal of this paper is to validate our hierarchical \( \kappa \) inference pipeline within a toy testbed, by explicitly treating the training distribution as a prior and conducting retrieval tests using carefully chosen test sets. We choose a broad Gaussian distribution for the training \( \kappa \) to enable an analytical treatment of the BGNN prior. As we will see in Section 3.4, Gaussian priors are especially convenient in our hierarchical inference context, because it has support everywhere in \( \mathbb{R} \). We subsampled 200,000 sight lines from the phenomenological set of 600,000 COSMOC2 sight lines to follow \( \kappa \sim \mathcal{N}(0.01, 0.04^2) \) where \( \mathcal{N}(\mu, \sigma^2) \) is a normal distribution with mean \( \mu \) and standard deviation \( \sigma \). The validation set contains 1000 examples, drawn from the same distribution as the training set.

Our primary target label for each sight line is \( \kappa \). We additionally design per-galaxy labels, the redshift (\( z \)) and stellar mass (\( M_* \)) of each galaxy included in the COSMOC2 catalog. We are motivated by the idea that training the BGNN to simultaneously predict \( z, M_* \) of each galaxy as well as \( \kappa \) may improve the inductive bias of the network for \( \kappa \) inference. We anticipate that it is important for the network to understand galaxy redshifts well, so that it can predict the relative \( \kappa \) contribution of the associated halos to the overall \( \kappa \) for the sight line, i.e., the lensing efficiency. The network must also internally learn to exclude a galaxy/halo from consideration if it is behind the source redshift of \( z_{\text{src}} = 2 \). We included stellar masses as a per-galaxy label to serve as proxies for halo masses, which were only available for central galaxies (populating main halos) in the COSMOC2 catalog and not for satellite galaxies (populating subhalos).

Our input is the photometric catalog of galaxies queried within an aperture of radius \( 1' \) around a sight line. The input features of each galaxy were derived from the COSMOC2 sky position and magnitudes in \( ugrizY \). We compute the relative sky position of each galaxy from the central LOS located at RA, dec, and transform the distances to the flat sky.

Gaussian photometric errors corresponding to 5 yr LSST conditions were simulated and added to the true \( ugrizY \) magnitudes. We adopted the analytic magnitude-dependent model for the 1\( \sigma \) uncertainties described in Ivezić et al. (2019). The model includes dependencies on true magnitudes as well as band-dependent parameters like sky brightness, seeing, atmospheric extinction, instrumental noise, and the number of visits. Figure 2 shows that our error model agrees with the estimated photometric uncertainties in the DC2 catalog (LSST DESC et al. 2021), which simulates 5 yr of the planned 10 yr LSST survey. In addition, we applied a detection cut of \( i < 25.3 \), corresponding to the 10 yr i-band gold sample (Gorecki et al. 2014). Ongoing follow-up study is investigating the effect of varying signal-to-noise ratio as well as the effect of removing or adding particular input features, such as the galaxy redshifts.

In summary, the input to our model was a variable-sized set of galaxies located within \( 1' \) of a sight line. Each galaxy carried eight features: the relative positional offset from the central LOS in RA, decl., and the \( ugrizY \) magnitudes with photometric noise added.

3.2. Bayesian Graph Neural Network

Bayesian neural networks (BNNs) are probabilistic extensions to standard neural networks (Denker & LeCun 1991). As outlined in Jospin et al. (2020), designing a BNN entails
choosing a stochastic model, which defines the prior on the network weights and the predictive distribution on the target quantities, and a functional model, which defines the network architecture. We describe the stochastic model in Section 3.2.1 and the functional model in Section 3.2.3.

3.2.1. Posterior Inference with Bayesian Neural Networks

It is instructive to decompose the uncertainty estimated by BNNs into two types: aleatoric (statistical, irreducible) and epistemic (systematic, reducible).

Originating from the intrinsic randomness in the data-generating process, the aleatoric uncertainty persists in the limit of infinite training data. This type of uncertainty is explicitly modeled as the width of the distribution over the target quantities. Concretely, let a sight line with photometric observations \( d \) be given. Suppose it contains \( L \) galaxies, indexed \( l = 1, \ldots, L \). Our targets are \( \kappa \) and the set of redshifts and stellar masses of the individual galaxies, \( \{z, M_*\} \). The aleatoric portion of our posterior is thus \( p(\kappa, \{z, M_*\} | d, W) \), for a set of network weights, \( W \).

In this paper, we adopt an independently Gaussian predictive distribution for \( \kappa \) and \( \{z, M_*\} \), i.e.,

\[
\log p(\kappa, \{z, M_*\} | d, W) \\
\propto \log \mathcal{N}(\kappa | m(d, W), s(d, W)) \\
+ \sum_l \log \mathcal{N}(z_l | t_l(d, W), u_l(d, W)) \\
+ \sum_l \log \mathcal{N}(M_{*l} | v_l(d, W), w_l(d, W)),
\]

where \( \mathcal{N}(\mu, \sigma) \) denotes the PDF of a Gaussian with mean and standard deviation \( \mu, \sigma \in \mathbb{R} \). The BNN predicted the \( m \), \( s \), and \( t_l \), \( u_l \), \( v_l \), \( w_l \) \( l = 1, \ldots, L \) for each sight line, where \( l \) indexes the galaxies in a sight line. We chose the Gaussian as a first-order approximation to the true posterior to focus on simple \( \kappa \) recovery tests. To control for effects from the prior, our training set has been subsampled from the original simulation to follow a Gaussian distribution in \( \kappa \) (Section 3.1). We train the network to jointly infer the per-galaxy properties \( \{z, M_*\} \) along with \( \kappa \) in order to improve the inductive bias of the BGN. Battaglia et al. 2018. The idea is that explicitly training the network on \( \{z, M_*\} \) labels will guide its understanding of the per-galaxy contribution to the overall \( \kappa \) for the sight line.

Epistemic uncertainty stems from our limited knowledge of the perfect model. It is reducible in the sense that it can be explained away with sufficient training data. This type of uncertainty is often described as a distribution over the network weights \( W \). Each realization of the weights corresponds to an alternative model, so integrating over this learned weight posterior amounts to Bayesian model averaging. Folding in the epistemic uncertainty, we have the full predictive distribution for \( \kappa \) of a given sight line:

\[
p(\kappa | d, \Omega_{\text{train}}) = \int p(\kappa | d, W)p(W | \Omega_{\text{train}}) \, dW,
\]

where we have made explicit the dependence on the training set by conditioning on the hyperparameters governing the training prior, \( \Omega_{\text{train}} \). Not modeling the epistemic uncertainty at all reduces to simple density estimation, where the weight posterior \( p(W | \Omega_{\text{train}}) \) is a delta function. In standard neural networks, which only give point estimates for the target parameters, both \( p(\kappa | d, W) \) and \( p(W | \Omega_{\text{train}}) \) are delta functions, so the predictive distribution in Equation (17) collapses to a delta function.

Exact evaluation of the integral in Equation (17) requires averaging over all of the weight configurations allowed by \( p(W | \Omega_{\text{train}}) \), which is intractable. There exist several approximations (see, e.g., Charnock et al. 2020 for a review). Among them, we opt for Monte Carlo (MC) dropout, because it precludes the need to train an ensemble of BNNs (Gal & Ghahramani 2016; Kendall & Gal 2017). MC dropout is a Bayesian interpretation of regular dropout, whereby nodes are randomly dropped out, or set to zero, with some tuned probability. Mathematically, it replaces the true weight posterior \( p(W | \Omega_{\text{train}}) \) with a variational distribution \( q_W(W | \Omega_{\text{train}}) \) parameterized by \( \theta \) (Gal & Ghahramani 2016):

\[
q_W(W | \Omega_{\text{train}}) = \prod_i q_W(W_i | \Omega_{\text{train}})
\]

\[
W_i = W_i \cdot \text{diag}(z_{i,j})_{j=1}^L
\]

where \( i \) indexes layers of the network, and \( j \) indexes the nodes in a given layer. Here, \( J_i \) denotes the number of nodes in layer \( i \), such that the weight matrix for layer \( i \) is \( W_i \in \mathbb{R}^{J_i \times J_{i-1}} \). When \( z_{i,j} = 0 \), the input node \( j \) in layer \( i \) is dropped out. In practice, the per-layer dropout probability \( p_i \) is replaced with a global dropout probability \( p_{\text{drop}} \).

The full predictive posterior with the variational approximation is thus

\[
p(\kappa | d, \Omega_{\text{train}}) = \int p(\kappa | d, W)q_W(W | \Omega_{\text{train}}) \, dW. \tag{19}
\]

3.2.2. Optimization

The loss function is the negative log evidence lower bound over the \( N = 200,000 \) examples in the training set, \( \{d^{(n)}, \kappa^{(n)}, \{z, M_*\}^{(n)}\}_{n=1}^N \):

\[
\mathcal{L}(W) = \sum_{n=1}^N \int \log p(\kappa^{(n)} | d^{(n)}, \{z, M_*\}^{(n)} | W)q_W(W | \Omega_{\text{train}})dW \\
+ KL(q_W(W | \Omega_{\text{train}}) || p(W)),
\]

where \( p(W) \) is a prior on the network weights. To evaluate the first term in an unbiased way, we take a single MC sample \( W \sim q_W(W | \Omega_{\text{train}}) \). Then \( W \) can be updated via gradient descent with respect to the realized sample. The second KL term is the "regularization" term that prevents the weights from deviating too far from our prior. This is intractable in its exact form, but reduces to \( L_2 \) regularization,

\[
KL(q_W(W | \Omega_{\text{train}}) || p(W)) \propto \frac{h^2(1 - p_i)}{2N} ||W||^2, \tag{21}
\]

where we assume a weight prior that can be factorized into a product of Gaussian priors in each layer. The length scale \( h \) is a hyperparameter that determines the width of the prior. Note that the dropout probability is also a hyperparameter in the formulation introduced here. It is not optimized along with \( W \) during training but rather tuned manually as part of the hyperparameter search. We assume the same dropout probability \( p_i = p_{\text{drop}} \) for every layer \( i \).
For a given choice of $p_{\text{drop}}$, $h$ can be folded into the hyperparameter $\lambda_{\text{reg}} = h^2(1 - p)/(2N)$ controlling the $L_2$ regularization strength. We tune $p_{\text{drop}}$, the initial learning rate, batch size, $\lambda_{\text{reg}}$, and the number of BGNN layers (network depth) on a sparse log-uniform random grid with the respective ranges $[0.01, 0.1]$, $[10^{-4}, 10^{-2}]$, $[128, 1024]$, and $[10^{-5}, 10^{-2}]$ as part of our hyperparameter search. We use a held-out validation set and do not perform cross validation.

We train our BNN via minibatch gradient descent using the ADAM optimizer with a weight decay of $\lambda_{\text{reg}} = 10^{-4}$ and batch size of 1024. The learning rate, initially $10^{-3}$, is halved whenever the validation loss fails to decrease for five consecutive epochs. We stop training when the validation loss does not increase for 10 consecutive epochs.

It is common practice to transform the training input and labels so that they fall into a predefined range. This preprocessing step has the effect of facilitating optimization, as it promotes the numerical stability of the network’s hidden units and their gradients. The input $d$ and target $\kappa$ labels are normalized so that they have a mean of 0 and standard deviation of 1 across the entire training set.

**Figure 2.** We simulated Gaussian photometric errors, with $1\sigma$ uncertainties plotted as white dashed lines. Our error model agreed with the estimated uncertainty values in the DC2 catalog simulating 5 yr of the LSST survey, shown in the heatmap. We also applied a cut of $i < 25.3$ (yellow), corresponding to the $i$-band gold sample.

**Figure 3.** Our network takes as input the features of $V$ nodes comprising an LOS, denoted $\{x^{(0)}_v, \ldots, x^{(V-1)}_v\}$, and passes through five residual blocks. In each residual block $b$, the local (node) encodings $\{x^{(b-1)}_v, \ldots, x^{(V-1)}_{V-1}\}$ and global (graph) encoding $u^{(b-1)}$ are processed together through a series of residual MLPs to yield updated local and global encodings. The final encodings enter the projector MLP, which produces the posterior PDFs over the local target quantities ($z, M$ for each node) and the global target quantity $\kappa$. 
increase linearly with decreasing $\kappa$. Our network takes as input the features of the LOS, denoted $\{x^{(b)}_i, \ldots, x^{(b)}_V\}$, and passes them through five residual blocks (He et al. 2016). In each residual block, indexed $b$, the constituent local (node-level) encodings $\{x^{(b-1)}_i, \ldots, x^{(b-1)}_V\}$ and global (graph-level) encoding $\bar{x}^{(b-1)}$ are processed together through a series of multilayer perceptrons (MLPs) with residual skip connections to yield updated local and global encodings. The final encodings enter the last MLP, which produces the posterior PDFs over the local target quantities ($z, M^*$ for each node) and the global target quantity $\kappa$.

The operations on the nodes (galaxies) are invariant to the permutations of the nodes, making our model a deep set (Zaheer et al. 2017; Battaglia et al. 2018) with a custom architecture. We do not supply edge information in our implementation, but it is possible to model features like the projected sky distance between galaxy pairs as edges and apply graph convolutions.

3.3. Summary Statistics Matching

Matching summary statistics can be considered an ABC-based method, whereby summary statistics of a test sight line are matched against those of training sight lines with known $\kappa$, which define the prior (Birrer et al. 2019). We can compare the BGNN constraints with constraints produced by matching two types of summary statistics. One is the unweighted number counts ($N_{\text{simp}}$), defined simply as the number of galaxies observed around the sight line. The other is the inverse-distance-weighted number counts ($N_{1/r,\text{simp}}$). For $N_{\text{simp}}$ observed galaxies in a field of view,

$$N_{1/r,\text{simp}} = \frac{N_{\text{simp}}}{\sum_{i=1}^{N_{\text{simp}}} \frac{1}{r_i + 10^{-5}}},$$  \hspace{1cm} (22)

where $r_i$ is the projected flat-sky distance of galaxy $i$ from the central LOS in arcseconds. The small value of $10^{-5}$ was added to the denominator for numerical stability. Figure 4 illustrates the training sight lines with varying $\kappa$. Overall, higher $\kappa$ correlates with higher $N$ and $N_{1/r}$.

The subscript “simp” in $N_{\text{simp}}$ and $N_{1/r,\text{simp}}$ indicates that these are simplified statistics compared to those employed by TDCOSMO (e.g., Greene et al. 2013; Rusu et al. 2017). We did not consider more complex, model-dependent weighting schemes that incorporate inferred stellar masses and spectroscopic redshifts, e.g., $z, L, \text{and } M$ in Greene et al. (2013), in order to focus on those that could be computed directly from photometric observations. We also did not normalize the summary statistics with respect to the average in the simulations, because all of the sight lines in this study were derived from the same simulations.

The summary statistics matching was implemented as follows. The values of $N_{\text{simp}}$ ($N_{1/r,\text{simp}}$) were computed for all of the sight lines in the training and test sets. Then, for each test sight line, we queried the training sight lines with $N_{\text{simp}}$ ($N_{1/r,\text{simp}}$) that matched the test sight line’s own $N_{\text{simp}}$ ($N_{1/r,\text{simp}}$) within some closeness threshold, which we denote $\Delta N_{\text{simp}}$ ($\Delta N_{1/r,\text{simp}}$). The $\kappa$ corresponding to the matched training sight lines could then be interpreted as a posterior on $\kappa$, i.e., $p(\kappa|d, \Omega_{\text{train}})$. The threshold $\Delta N_{\text{simp}}$ ($\Delta N_{1/r,\text{simp}}$) determines how much information is captured in the matched samples. As a rule of thumb, it should be kept as small as possible provided that it yields sufficient matched samples. A large threshold yields more samples, but the quality of the matching may be poor. A small threshold sacrifices the number of samples for a closer match to the test sight line’s summary statistic and thus potentially its $\kappa$. At one extreme, $\Delta N_{\text{simp}} = 0$ means that a training sample must have the same exact $N_{\text{simp}}$ as the test sample in order to be considered “matched” and contribute to the posterior. For a given test sight line, $\Delta N_{\text{simp}}$ was chosen to be the smallest among $\{0, 1, 2, 4, 8\}$ that yielded more than 100 matched samples. Likewise, $N_{1/r,\text{simp}}$ was chosen to be the smallest among $\{1, 2, 4, 8\}$ that yielded more than 100 matched samples. This resulted in $\Delta N_{\text{simp}} = 1$ for most sight lines and an average $\Delta N_{1/r,\text{simp}}$ of 1.3.

We stress that our matching scheme differs from that implemented by Greene et al. (2013) and Rusu et al. (2017) in two main ways. First, these studies binned the range of each summary statistic and reweighted the $\kappa$ contribution of each matched sight line in the final posterior by the inverse of the bin count, so that each bin contributed equal weight. The reweighting was intended to prevent the $\kappa$ posterior from being dominated by a large number of sight lines in a given $N_{\text{simp}}$ bin. We do not enforce such a reweighting with respect to the summary statistic, but reweight by the training distribution in $\kappa$ when we hierarchically infer the hyperparameters (see Equation (25)). Second, the $N_{\text{simp}}$ and $N_{1/r,\text{simp}}$ were matched jointly, rather than independently. We chose not to apply the joint constraint, because our small training set size of 200,000 sight lines made it difficult to achieve sufficient sample statistics. Rusu et al. (2017) used a joint constraint from four
The Astrophysical Journal, 953:178 (19pp), 2023 August 20

Park et al.

Figure 5. Visualization of the test set distributions. Dashed lines refer to the representative test set distribution that is identical to the distributions of training and validation sets. The B group of experiments (blue, solid lines) probe varying $\mu_{\text{test}}$ with broad $\sigma_{\text{test}}$, and likewise with the C group (black, solid lines) with narrow $\sigma_{\text{test}}$. Note that C4, in particular, has very little overlap with the training set.

Table 1

| Parameter                        | Prior                        | Description                                                                 |
|----------------------------------|------------------------------|-----------------------------------------------------------------------------|
| Individual lines of sight $\kappa$ | $\mathcal{N}(0.01, 0.04^2)$ | Posterior PDFs estimated by BGNN or summary statistics matching              |
| Gaussian test population $\mu_{\text{test}}$ | $\mathcal{U}(-0.5, 0.5)$    | Gaussian mean                                                               |
| $\log \sigma_{\text{test}}$          | $\mathcal{U}(-7, 0)$        | Natural log of the Gaussian standard deviation                              |

types of summary statistics on $10^9$ training sight lines and noted that using more than four may result in sample sparsity.

The $\kappa$ values of the matched training sight lines can be interpreted as posterior samples. We denote the matched samples and the posterior PDFs represented by these samples as

$$
\kappa_{\text{matched}} \sim p_{\kappa}(\kappa; \Omega_{\text{train}})
$$

and

$$
\kappa_{\text{matched,}\nu_j} \sim p_{\kappa}(\kappa; \Omega_{\text{train}}),
$$

for consistency with the notation of the BGNN posterior in Equation (19). We condition on $\Omega_{\text{train}}$ everywhere, because both the BGNN and summary statistics techniques depend on the distributions of sight lines in the training set.

3.4. Hierarchical Inference

We designed eight test sets, each containing 1000 sight lines drawn from normal distributions with varying $\mu_{\text{test}}$ and $\sigma_{\text{test}}$ (see Figure 5). Test set A is representative of the training set, i.e., follows the same distribution in $\kappa$. The B group of test sets has wide distributions in $\kappa$, with $\sigma_{\text{test}} = 0.02$, and the C group has artificially narrow distributions, with $\sigma_{\text{test}} = 0.005$. We include shifted test distributions with particularly overdense (high-$\kappa$) sight lines, where strong lenses tend to occur (B3, C3, and C4). In particular, C4 serves as an extreme stress test, as it has very little overlap with the training $\kappa$ distribution. See Figure 5 for a visualization of the eight test distributions and their varying degrees of overlap with the training distribution. The size of 1000 was chosen, because it was large enough to avoid small sample statistics but allowed experimental turn-around within a reasonable time (see Section 4.3).

The $\kappa$ posteriors in Equations (19), (23), and (24) are conditioned on $\Omega_{\text{train}}$, the set of assumptions governing the training set. Here, $\Omega_{\text{train}}$ includes the particular choices of the underlying N-body simulation, semianalytic models used to paint galaxies onto the halos, and the observation noise in the galaxy positions and magnitudes. The test set, in principle, may differ in any of these aspects. In this paper, we focus on training-test mismatches in the $\kappa$ distributions, by designing test sets from the same simulation suite as the training set but subsampled to follow shifted distributions in $\kappa$. As described in Section 3.1, we assume Gaussian distributions for both the training and test distributions.

We hierarchically infer the hyperparameters governing the Gaussian test distribution, $\mu_{\text{test}}, \sigma_{\text{test}}$, using importance sampling (Hogg et al. 2010; Foreman-Mackey et al. 2014; Wagner-Carena et al. 2021). The posterior on $\mu_{\text{test}}, \sigma_{\text{test}}$ takes the form

$$
p(\mu_{\text{test}}, \log \sigma_{\text{test}} | \{d\}) \propto p(\mu_{\text{test}}, \log \sigma_{\text{test}}) \times \prod_{i=1}^{N_{\text{test}}} \frac{1}{M} \sum_{k} \frac{p(\kappa | \mu_{\text{test}}, \log \sigma_{\text{test}})}{p(\kappa | \Omega_{\text{train}})}. 
$$

where $p(\mu_{\text{test}}, \log \sigma_{\text{test}})$ is our hyperprior and $i$ indexes the test set. Note that we work in the natural log space of $\sigma_{\text{test}}$ to restrict our sampling to positive values (see, e.g., Wagner-Carena et al. 2021, for the derivation of Equation (25)). The second term lends itself to Markov Chain Monte Carlo (MCMC) sampling. To evaluate this MCMC objective for a candidate $\mu_{\text{test}}, \sigma_{\text{test}}$, we first draw $M$ samples from the individual posteriors conditioned on the training set. We can do this efficiently with a BGNN or use the matched samples themselves with the summary statistics method. We then evaluate the density of the training prior $p(\kappa | \Omega_{\text{train}})$ and our target Gaussian distribution at those $\kappa$ samples and take the mean of the density ratio across samples. The weighting by $1/p(\kappa | \Omega_{\text{train}})$ assigns more relative
importance to $\kappa$ underrepresented by the training set. Note that division by zero will result whenever $p(\kappa|\mu_{\text{test}}, \sigma_{\text{test}})$ assigns density to a $\kappa$ value not encompassed by $p(\kappa|\Omega_{\text{train}})$. We choose a Gaussian distribution, with support in $\mathbb{R}$, as the training prior, and ensure that it is sufficiently broad to make evaluation of the MCMC objective numerically stable. Because Equation (25) is not guaranteed to be unbiased for finite $M$, we carried out convergence tests on the hierarchical results to determine our $M = 20,000$. This is a conservative value, given that there are only two hyperparameters. We summarize the model (hyper) parameters and their (hyper)priors in Table 1.

We interpret the output of the BGNN and summary statistics matching as posterior PDFs, with the training prior applied. For evaluating $\kappa$ recovery on individual sight lines, we work with likelihoods by dividing by the training prior:

$$p(d|\kappa, \Omega_{\text{train}}) \propto p(d|\kappa, \Omega_{\text{train}})/p(\kappa|\Omega_{\text{train}}),$$

where $p(\kappa|\Omega_{\text{train}}) = \mathcal{N}(0.01, 0.04^2)$ in our setup.

4. Results and Discussion

We organize our results as follows. In Section 4.1, we evaluate the BGNN’s $\kappa$ recovery performance on individual sight lines. Having established that the individual constraints are reasonable, we proceed to present the hierarchical inference results of the test population as a whole in Section 4.2.

4.1. Individual $\kappa$ Recovery

Let us first compare the accuracy of $\kappa$ recovery across the BGNN and summary statistics methods for a range of $\kappa$ spanning the training distribution. In Figure 6, we bin the representative test set A into 17 $\kappa$ bins of fixed bin width 0.01 and plot the mean and standard deviation of the recovered $\kappa$ in each bin, weighted by the estimated uncertainties for the individual sight lines. We subsample 148 sight lines from each bin to obtain uniform bin counts. That is, given the central estimate ($\kappa_i$) and the associated $1\sigma$ uncertainty ($\sigma_i$) of the likelihoods for each sight line $i$, we computed the weighted mean and standard error (SE) using $1/\sigma_i^2$ as the weights.

Explain the methodology and results.

Explicitly, we computed for each bin:

$$\text{weighted mean} = \bar{\kappa} = \frac{\sum_{i=1}^{B} \kappa_i}{\sum_{i=1}^{B} \sigma_i^2 / W}$$

and

$$\text{weighted SE} = \frac{\sum_{i=1}^{B} (\kappa_i - \bar{\kappa})^2 / \sigma_i^2}{\sum_{i=1}^{B} 1} W = \frac{V}{W},$$

where $B$ was the total number of sight lines in the bin, $W = \sum_{i=1}^{B} 1/\sigma_i^2$ was the sum of the weights, and the $V = W_2/W^2$ where $W_2 = \sum_{i=1}^{B} 1/\sigma_i^4$. If all weights $1/\sigma_i^2$ are equal, the factor $V$ reduces to $1/B$, so Equation (28) becomes the conventional (unweighted) SE that scales with $1/\sqrt{B}$. In Figure 6, the dots are computed from Equation (27) and the sizes of the error bars from Equation (28). Overall, our BGNN is more accurate than both $N_{\text{simp}}$ and $N_{1/r_{\text{simp}}}$ in all $\kappa$ bins. As expected, it boasts the highest accuracy in regions of $\kappa$ best represented in the training distribution $\sim \mathcal{N}(0.01, 0.04^2)$. On the other hand, all of the methods reveal signs of upward bias in $\kappa < -0.05$ and downward bias in $\kappa > 0.06$. The bias is the worst for $N_{\text{simp}}$, followed by $N_{1/r_{\text{simp}}}$ and the BGNN. The bin count was 148 for all bins. Note that $N_{\text{simp}}$ and $N_{1/r_{\text{simp}}}$ are simplified versions of the summary statistics employed by TDCOSMO (e.g., Rusu et al. 2017), and our analysis operates on far fewer training sight lines ($2 \times 10^5$ compared to TDCOSMO’s $10^5$).

Figure 6. The binned $\kappa$ recovery based on the likelihoods modeled by BGNN and the two summary statistics. The dots are computed from Equation (28), and the sizes of the error bars are computed from Equation (29). The BGNN is more accurate than both $N_{\text{simp}}$ and $N_{1/r_{\text{simp}}}$ in all $\kappa$ bins. It boasts the highest accuracy in regions of $\kappa$ best represented in the training distribution $\sim \mathcal{N}(0.01, 0.04^2)$. On the other hand, all of the methods reveal signs of upward bias in $\kappa < -0.05$ and downward bias in $\kappa > 0.06$. The bias is the worst for $N_{\text{simp}}$, followed by $N_{1/r_{\text{simp}}}$ and the BGNN. The bin count was 148 for all bins. Note that $N_{\text{simp}}$ and $N_{1/r_{\text{simp}}}$ are simplified versions of the summary statistics employed by TDCOSMO (e.g., Rusu et al. 2017), and our analysis operates on far fewer training sight lines ($2 \times 10^5$ compared to TDCOSMO’s $10^5$).
The binned $\kappa$ recovery based on the likelihoods modeled by BGNN and the two summary statistics. The dots are computed from Equation (27), and the sizes of the error bars are computed from Equation (29). The BGNN predictions are consistent with the true $\kappa$ for all $\kappa$ values. On the other hand, both the $N_{\text{simp}}$ and $N_{1/r_{\text{simp}}}$ summary statistics reveal signs of upward bias in $\kappa < -0.04$ and downward bias in $\kappa > 0.09$. The bin count was 148 for all bins. Note that $N_{\text{simp}}$ and $N_{1/r_{\text{simp}}}$ are simplified versions of the summary statistics employed by TDCOSMO (e.g., Rusu et al. 2017), and our analysis operates on far fewer training sight lines ($2 \times 10^5$ compared to TDCOSMO’s $10^6$).

Figure 7. The binned $\kappa$ recovery based on the likelihoods modeled by BGNN and the two summary statistics. The dots are computed from Equation (27), and the sizes of the error bars are computed from Equation (29). The BGNN predictions are consistent with the true $\kappa$ for all $\kappa$ values. On the other hand, both the $N_{\text{simp}}$ and $N_{1/r_{\text{simp}}}$ summary statistics reveal signs of upward bias in $\kappa < -0.04$ and downward bias in $\kappa > 0.09$. The bin count was 148 for all bins. Note that $N_{\text{simp}}$ and $N_{1/r_{\text{simp}}}$ are simplified versions of the summary statistics employed by TDCOSMO (e.g., Rusu et al. 2017), and our analysis operates on far fewer training sight lines ($2 \times 10^5$ compared to TDCOSMO’s $10^6$).

Table 3 summarizes the bias and uncertainty of $\mu_{\text{test}}$ constraints for all of the experiments. Treating the sight lines in each experiment as a population of strong lensing sight lines, we can interpret our maximum-likelihood estimate of $\mu_{\text{test}}$ as the central estimate of the external convergence, $\bar{c}_{\text{test}}$, and the uncertainty on $\mu_{\text{test}}$ as $\sigma(\bar{c}_{\text{test}})$. Equation (13) then allows us to compute the contribution of the external environment to the overall fractional uncertainty on $H_0$.

For all experiments, the BGNN introduces lower bias on $H_0$ (last column) than does either summary statistic. With the exception of the extremely overdense population C4, which introduces a 1.6% bias on $H_0$, the bias level remains at the subpercent level for the BGNN.

The performances of all three methods follow patterns expected from the individual recovery results in Section 4.1—the best constraints on $\mu_{\text{test}}$ come from environments best represented by the training set, i.e., B2, C2 that have $\kappa \sim 0$. Notably, the percent bias on $H_0$ is only $-0.1\%$ on B2 and $-0.2\%$ on C2 for the BGNN.

The bias increases toward underdense and overdense populations in the tails of the training distribution, but the BGNN is significantly more robust than the summary statistics. Whereas the bias level only increased to $-0.8\%$ from C2 to C1/C3 for the BGNN, it increased to $\sim 2\%$ for $N_{\text{simp}}$ and $\sim 1.5\%$ for $N_{1/r_{\text{simp}}}$. A similar pattern holds for the broad B group of test sets as well, although the degree of deterioration is slightly worse for the B group because it contains more of the extreme sight lines.

We find that the $1\sigma$ uncertainty on $H_0$ (second-to-last column) is smaller than the bias on $H_0$ across the board, pointing to a global underestimation of the uncertainty on $\mu_{\text{test}}$. The only exception is the BGNN for B2, where the bias is small enough ($-0.1\%$) that the estimated $1\sigma$ uncertainty of 0.14% can account for it. Even so, relative to the summary statistics, the BGNN estimates of the uncertainty are better, i.e., come closer to covering the bias.

The hierarchical inference constraints for C3 are shown in Figure 8. This test set represents a particularly challenging environment with $\mu_{\text{test}}$ shifted high and $\sigma_{\text{test}}$ artificially narrow. Our BGNN can accurately recover the high $\mu_{\text{test}}$ and narrow $\sigma_{\text{test}}$ within its $1\sigma$ credible interval. The BGNN constraints translate into a 0.3% contribution to the $H_0$ uncertainty without evidence of bias. On the other hand, both the $N_{\text{simp}}$ and
Table 2

| \( \kappa \)  | Method   | \( \log p^{\text{ab}} \) (Accuracy) | \( \text{MAE}^{\text{ac}} \) (Precision) | \( \kappa \)  | Method   | \( \log p^{\text{ab}} \) (Accuracy) | \( \text{MAE}^{\text{ac}} \) (Precision) |
|-----------|---------|-----------------------------------|---------------------------------|-----------|---------|-----------------------------------|---------------------------------|
| \([-0.06, -0.04)\] | BGNN    | 2.6 ± 0.3                         | 0.02 ± 0.01                     | \([0.02, 0.04)\] | BGNN    | 1.9 ± 0.2                         | 0.04 ± 0.01                     |
|           | \( N_{\text{simp}} \) | 2.3 ± 0.4                         | 0.03 ± 0.01                     |           | \( N_{\text{simp}} \) | 1.3 ± 0.4                         | 0.07 ± 0.02                     |
|           | \( N_{1/r,\text{simp}} \) | 2.1 ± 0.2                         | 0.03 ± 0.01                     |           | \( N_{1/r,\text{simp}} \) | 1.6 ± 0.4                         | 0.05 ± 0.02                     |
| \([-0.04, -0.02)\] | BGNN    | 2.4 ± 0.2                         | 0.02 ± 0.01                     | \([0.04, 0.06)\] | BGNN    | 1.8 ± 0.2                         | 0.04 ± 0.01                     |
|           | \( N_{\text{simp}} \) | 2.1 ± 0.3                         | 0.03 ± 0.01                     |           | \( N_{\text{simp}} \) | 1.3 ± 0.3                         | 0.07 ± 0.02                     |
|           | \( N_{1/r,\text{simp}} \) | 2.0 ± 0.1                         | 0.03 ± 0.01                     |           | \( N_{1/r,\text{simp}} \) | 1.4 ± 0.4                         | 0.06 ± 0.02                     |
| \([-0.02, 0.0)\]  | BGNN    | 2.2 ± 0.2                         | 0.03 ± 0.01                     | \([0.06, 0.08)\] | BGNN    | 1.7 ± 0.4                         | 0.05 ± 0.01                     |
|           | \( N_{\text{simp}} \) | 1.8 ± 0.3                         | 0.05 ± 0.01                     |           | \( N_{\text{simp}} \) | 1.3 ± 0.4                         | 0.06 ± 0.02                     |
|           | \( N_{1/r,\text{simp}} \) | 2.0 ± 0.1                         | 0.03 ± 0.01                     |           | \( N_{1/r,\text{simp}} \) | 1.4 ± 0.4                         | 0.06 ± 0.02                     |
| \([0.0, 0.02)\]  | BGNN    | 2.0 ± 0.2                         | 0.03 ± 0.01                     | \([0.08, 0.1)\] | BGNN    | 1.6 ± 0.5                         | 0.05 ± 0.02                     |
|           | \( N_{\text{simp}} \) | 1.6 ± 0.4                         | 0.05 ± 0.02                     |           | \( N_{\text{simp}} \) | 1.2 ± 0.5                         | 0.07 ± 0.03                     |
|           | \( N_{1/r,\text{simp}} \) | 1.9 ± 0.3                         | 0.04 ± 0.01                     |           | \( N_{1/r,\text{simp}} \) | 1.4 ± 0.4                         | 0.06 ± 0.02                     |

Notes.

\( a \) Values listed are the median of the metrics across sight lines in a given bin. Errors listed are the standard deviations in a given bin.

\( b \) Defined in Equation (30). Higher is better.

\( c \) Defined in Equation (31). Lower is more accurate.

\( d \) Defined in Equation (32). Lower is more precise.

\( N_{1/r,\text{simp}} \) summary statistics lead to downward biases on \( \mu_{\text{test}} \). They also overestimate \( \sigma_{\text{test}} \), which happens with underestimated uncertainties on individual \( \kappa \).

Similarly, the constraints for the broad, overdense test set (Experiment B3) are shown in Figure 9. Out of all of our test distributions, B3 best characterizes the \( \kappa_{\text{ext}} \) distributions in the seven TDCOSMO lenses. Our BGNN can accurately recover the high \( \mu_{\text{test}} \) within its 2\( \sigma \) credible interval. Again, both of the summary statistics underestimate \( \mu_{\text{test}} \) and overestimate \( \sigma_{\text{test}} \).

Our hierarchical inference pipeline reaches a numerical breaking point, however, with a test set shifted far away from the training set, e.g., Experiment C4, which lies at the upper tail of the training distribution (Figure 9). The BGNN struggles because it has not been exposed to many sight lines with \( \kappa \sim 0.08 \) in the training set. The low training density in this region of high \( \kappa \) affects the summary statistics methods more directly, because they run out of training samples to match. See Figure 5 for a visualization of the \( \kappa \) distribution in Experiment C4 compared to that of the training set.

Even at this breaking point, however, the true \( \mu_{\text{test}} \) falls within the BGNN’s 2\( \sigma \) credible interval. For the summary statistics, the downward biases on \( \mu_{\text{test}} \) become more dramatic than in Experiments C3 or B3. The uncertainties are underestimated for the BGNN in this regime, resulting in overestimation of \( \sigma_{\text{test}} \). The summary statistics methods do recover \( \sigma_{\text{test}} \) within their 2\( \sigma \) credible interval, however.

Based on a simulated data set consisting of 200,000 sight lines following \( N(0.01, 0.04) \), we have demonstrated the ability of BGNNs to effectively model the mapping between photometric observations and the continuous \( \kappa \) space. Summary statistics matching, on the other hand, operate on discrete samples, so it cannot extrapolate beyond the \( \kappa \) samples present in the data set. When there are only 200,000 sight lines, the poor matching causes posteriors to revert to the prior, and when the prior is shifted from the actual population, bias results. The BGNN also tends toward the training distribution when trained on a small data set and also leads to bias, but it is more robust to the train-test mismatch. For both approaches, we expect the hierarchical inference performance to improve with bigger training sets that better sample high- and low- \( \kappa \) environments. Further tests are needed to probe the relative performance of the BGNN and summary statistics methods with increasing training sets. We postpone this exercise to future work.

We emphasize that the comparison between BGNN and summary statistics presented in this work is not to be viewed as an evaluation of the TDCOSMO analysis of \( \kappa_{\text{ext}} \). We include the comparison to highlight the difference between a neural net that operates on the continuous \( \kappa \) space and an ABC method that either accepts or rejects discrete samples. Our implementation of the summary statistics matching differs in significant ways from the TDCOSMO implementation. TDCOSMO uses orders-of-magnitude more sight lines (~billion compared to our 200,000; e.g., Greene et al. 2013; Rusu et al. 2017). Their sight lines follow the phenomenological distribution from the Millennium Simulation, which extends out to high \( \kappa \). Thanks to the data set size, they are able to match to both \( N_{\text{simp}} \) and \( N_{1/r,\text{simp}} \), whereas we have matched to either \( N_{\text{simp}} \) or \( N_{1/r,\text{simp}} \) in our work. Additionally, TDCOSMO incorporates external shear information from the lens modeling to further constrain their \( \kappa_{\text{ext}} \). This could not be done here, because we focused on sight lines without strong lensing.

A key takeaway from our toy hierarchical tests is the importance of systematically testing our assumptions about the \( \kappa \) prior. Regardless of the \( \kappa \) inference algorithm used, neural network or ABC-based, we must infer the target \( \kappa \) population.
and account for any difference from the prior. If we instead derive $\kappa$ posteriors from individual sight lines based on a chosen $\kappa$ prior and let them independently enter the joint downstream $H_0$ analysis, we effectively multiply by the prior $N$ times, where $N$ is the number of sight lines. As $N$ increases, the particular choice of $\kappa$ prior will become more significant.

### 4.3. Computation Time

The total computation time of our hierarchical $\kappa$ inference pipeline can be broken down into the BGNN training/inference time and the hierarchical inference time. Training the BGNN with the configuration detailed in Section 3.2 took about 12 hr. The model was trained for 118 epochs on an Intel Xeon Gold 6148 GPU on the Cori system available at the National Energy Research Scientific Computing Center. Generating BGNN predictions on the 1000 sight lines in each test set took seconds. For a given training set size, the BGNN training and inference time can be considered fixed with respect to the number of test sight lines. The last step of hierarchical inference involves MCMC sampling, which took 2–3 hr to converge on 4 CPU cores for 1000 sight lines. In our current implementation, the matching time scaled linearly with the number of sight lines and with the size of the “closeness threshold grid.” For each sight line, we carried out matching according to each threshold on this prespecified grid and took the smallest threshold with more than 100 matched samples at the end. We used a grid of size 20 for both $N_{\text{simp}}$ and $N_{1/r_{\text{simp}}}$. In the hierarchical inference step, the MCMC sampling with the summary statistics likelihoods also took 2–3 hr to converge on four CPU cores for 1000 sight lines—similarly as with the BGNN likelihoods.

To generate our data sets, we performed additional raytracing on top of the COSMOC2 values, as detailed in Section A.4.1, to compute the $\kappa$ labels and queried the galaxy catalog to construct the input. We simulated 850,000 sight lines and subsampled smaller Gaussian subsets to conduct our analysis. Out of the 850,000 sight lines, we reserved 600,000 sight lines for subsampling the Gaussian training set of 200,000 sight lines and used the remaining 250,000 sight lines to subsample the Gaussian validation and test sets. The raytracing took 85 hr for 850,000 sight lines on 18 CPU cores. The input construction took 102 hr for the same sight lines on 18 CPU cores.

### 5. Conclusion

In this paper, we introduced a novel graph-based neural network architecture that can infer $\kappa$ from any number of observable galaxy properties and a first hierarchical pipeline for

| Exp. | Method | Central $\mu_{\text{test}}$, Pred $\bar{r}_{\text{ext}}$ | Bias on $\mu_{\text{test}}$ $\bar{r}_{\text{ext}} - \mu_{\text{test,true}}$ | 1σ Uncert. on $\mu_{\text{test}}$ $\sigma(\bar{r}_{\text{ext}})$ | % Contrib to $H_0$, err$^a$ $r_{\text{test}} / r_{\text{ext}}$ | % Bias on $H_0^b$ $\mu_{\text{test,true}} - r_{\text{test}}$ |
|------|--------|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|
| A    | BGN   | 0.006                                          | -0.004                                          | 0.002                                          | 0.157                                          | 0.417                                          |
|      | $N_{\text{simp}}$ | 0.002                                          | -0.008                                          | 0.002                                          | 0.183                                          | 0.826                                          |
|      | $N_{1/r_{\text{simp}}}$ | 0.005                                          | -0.005                                          | 0.002                                          | 0.188                                          | 0.485                                          |
| B1   | BGN   | -0.032                                         | 0.008                                           | 0.001                                          | 0.088                                          | -0.814                                         |
|      | $N_{\text{simp}}$ | -0.019                                         | 0.021                                           | 0.001                                          | 0.063                                          | -2.010                                         |
|      | $N_{1/r_{\text{simp}}}$ | -0.022                                         | 0.018                                           | 0.001                                          | 0.068                                          | -1.739                                         |
| B2   | BGN   | 0.001                                          | 0.001                                           | 0.001                                          | 0.143                                          | -0.108                                         |
|      | $N_{\text{simp}}$ | -0.004                                         | -0.004                                          | 0.001                                          | 0.116                                          | 0.399                                          |
|      | $N_{1/r_{\text{simp}}}$ | -0.003                                         | -0.003                                          | 0.001                                          | 0.104                                          | 0.323                                          |
| B3   | BGN   | 0.031                                          | -0.009                                          | 0.002                                          | 0.163                                          | 0.950                                          |
|      | $N_{\text{simp}}$ | 0.018                                          | -0.022                                          | 0.001                                          | 0.094                                          | 2.273                                          |
|      | $N_{1/r_{\text{simp}}}$ | 0.025                                          | -0.015                                          | 0.001                                          | 0.090                                          | 1.597                                          |
| C1   | BGN   | -0.032                                         | 0.008                                           | 0.001                                          | 0.120                                          | -0.762                                         |
|      | $N_{\text{simp}}$ | -0.021                                         | 0.019                                           | 0.001                                          | 0.053                                          | -1.866                                         |
|      | $N_{1/r_{\text{simp}}}$ | -0.024                                         | 0.016                                           | 0.001                                          | 0.060                                          | -1.528                                         |
| C2   | BGN   | 0.002                                          | 0.002                                           | 0.001                                          | 0.139                                          | -0.241                                         |
|      | $N_{\text{simp}}$ | -0.003                                         | -0.003                                          | 0.001                                          | 0.146                                          | 0.322                                          |
|      | $N_{1/r_{\text{simp}}}$ | -0.004                                         | -0.004                                          | 0.001                                          | 0.099                                          | 0.415                                          |
| C3   | BGN   | 0.033                                          | -0.007                                          | 0.002                                          | 0.164                                          | 0.750                                          |
|      | $N_{\text{simp}}$ | 0.020                                          | -0.020                                          | 0.001                                          | 0.094                                          | 2.125                                          |
|      | $N_{1/r_{\text{simp}}}$ | 0.027                                          | -0.013                                          | 0.001                                          | 0.084                                          | 1.404                                          |
| C4   | BGN   | 0.065                                          | -0.015                                          | 0.003                                          | 0.271                                          | 1.656                                          |
|      | $N_{\text{simp}}$ | 0.029                                          | -0.051                                          | 0.001                                          | 0.103                                          | 5.579                                          |
|      | $N_{1/r_{\text{simp}}}$ | 0.040                                          | -0.040                                          | 0.001                                          | 0.086                                          | 4.388                                          |

Note.

$^a$ Primary contribution to the fractional $H_0$ error budget based on $\mu_{\text{test}}$ constraints, defined in Equation (13).

$^b$ Bias introduced in $H_0$ due to bias on $\mu_{\text{test}}$, defined in Equation (15).
we have demonstrated the \( \sigma \) and \( \mu \) contours are 68% and 95% credible intervals. BGNN can accurately recover \( \mu_{\text{test}} \) within its 1σ credible interval. Both the \( N_{\text{simpl}} \) and \( N_{1/r,\text{simpl}} \) summary statistics lead to downward biases on \( \mu_{\text{test}} \). They also overestimate \( \sigma_{\text{test}} \). Note that \( N_{\text{simpl}} \) and \( N_{1/r,\text{simpl}} \) are simplified versions of the summary statistics employed by TDCOSMO (e.g., Rusu et al. 2017), and our analysis operates on far fewer training sight lines (2 \( \times \) 10^9 compared to TDCOSMO’s 10^10).

Figure 8. Constraints on the population hyperparameters for the narrow, overdense test set (Experiment C3) with truth \( \mu_{\text{test}} = 0.04 \) and \( \sigma_{\text{test}} = 0.005 \). Contours are 68% and 95% credible intervals. BGNN can accurately recover the \( \mu_{\text{test}} \) within its 1σ credible interval. Both the \( N_{\text{simpl}} \) and \( N_{1/r,\text{simpl}} \) summary statistics lead to downward biases on \( \mu_{\text{test}} \). They also overestimate \( \sigma_{\text{test}} \). Note that \( N_{\text{simpl}} \) and \( N_{1/r,\text{simpl}} \) are simplified versions of the summary statistics employed by TDCOSMO (e.g., Rusu et al. 2017), and our analysis operates on far fewer training sight lines (2 \( \times \) 10^9 compared to TDCOSMO’s 10^10).

Figure 9. Constraints on the population hyperparameters for the broad, overdense test set (Experiment B3) with truth \( \mu_{\text{test}} = 0.04 \) and \( \sigma_{\text{test}} = 0.02 \). Contours are 68% and 95% credible intervals. Accurate \( \mu_{\text{test}} \) recovery is critical to unbiased \( H_0 \) inference and the impact of \( \sigma_{\text{test}} \) recovery is second order (Section 2). The BGNN can accurately recover \( \mu_{\text{test}} \) within its 2σ credible interval. Both the \( N_{\text{simpl}} \) and \( N_{1/r,\text{simpl}} \) summary statistics lead to downward biases on \( \mu_{\text{test}} \). Note that \( N_{\text{simpl}} \) and \( N_{1/r,\text{simpl}} \) are simplified versions of the summary statistics employed by TDCOSMO (e.g., Rusu et al. 2017), and our analysis operates on far fewer training sight lines (2 \( \times \) 10^9 compared to TDCOSMO’s 10^10).

Constraining the population \( \kappa \) statistics. To probe a range of numerics in the recovery of hyperparameters in the target population, we have designed toy testbeds where the training \( \kappa \) distribution (our prior) and test populations are mismatched to varying degrees. We present comparisons between BGNN and summary statistics matching throughout to showcase the distinction between a neural network that operates in a continuous \( \kappa \) space and an ABC-based method that either accepts or reject discrete samples. Note, however, that our implementation of summary statistics matching does not represent TDCOSMO’s, which is more advanced and factors in more information.

We conclude the following:

1. Our BGNN can yield accurate and precise \( \kappa \) posterior PDFs by processing all available photometric observations (e.g., the positions and magnitudes of individual galaxies around an LOS). On average, it is 60% more accurate than matching our simplified number counts \( N_{\text{simpl}} \) and 30% more accurate than matching our simplified distance-weighted number counts \( N_{1/r,\text{simpl}} \).

2. When propagated into hierarchical inference, the BGNN-inferred posterior PDFs lend themselves to precise and accurate recovery of hyperparameters for a range of test populations shifted from the training distribution. On a population fully encompassed by the training distribution \( \kappa \sim N(0, 0.02^2) \), the BGNN can recover the population mean on \( \kappa \) precisely and accurately, translating into only a 0.1% 1σ uncertainty contribution to \( H_0 \) without evidence of bias. The \( N_{\text{simpl}} \) and \( N_{1/r,\text{simpl}} \) summary statistics, on the other hand, underestimates \( \kappa \), which would lead to a slight upward bias on \( H_0 \) even on this representative test set.

3. Both the BGNN and the simplified summary statistics methods became more biased as the test populations sampled the tails of the training distribution. The BGNN is found to be more robust to this training-test mismatch, however; on a particularly challenging shifted and narrow population with \( \kappa \sim N(0.04, 0.005^2) \), the bias on \( H_0 \) was 0.8% for the BGNN as compared to 2% and 1.5% for the \( N_{\text{simpl}} \) and \( N_{1/r,\text{simpl}} \) summary statistics, respectively.

4. Based on a simulated data set consisting of 200,000 sight lines following \( N(0.01, 0.04^2) \), we have demonstrated the ability of BGNNs to effectively model the mapping between photometric observations and the continuous \( \kappa \) space. On the other hand, the simplified summary statistics matching as implemented in our paper cannot extrapolate beyond the discrete \( \kappa \) samples in the data set, resulting in poor generalizability and accuracy. For both approaches, we expect the hierarchical inference performance to improve with bigger training sets that better sample high- and low- \( \kappa \) environments. Further tests are needed to probe the relative performance of the BGNN and summary statistics methods with bigger training sets.

We have validated the BGNN method of \( \kappa \) inference within the hierarchical Bayesian framework using simplified, well-controlled experiments. Being the first to infer the population \( \kappa \) statistics, we took a pedagogical angle and focused on “toy testbeds” for our method. We assumed convenient Gaussian parameterizations throughout in order to conduct simple...
recovery tests, but it remains to probe the numerics involved with inferring non-Gaussian population distributions, particularly those with a right skew. In addition, our test sets were drawn from the same simulation as the training set. In order to assess the impact of training-test mismatches in the assumed galaxy–halo connection, it will be important to test on sight lines derived from other simulations, such as the Millennium Simulation (Springel et al. 2005; Hilbert et al. 2009). Ultimately, we can better control the associated bias by inferring extra hyperparameters governing various aspects of the galaxy–halo connection. These extensions and further validation tests are not limited to the BGNN method and apply to the summary statistics matching (ABC methods) as well.

**Acknowledgments**

This paper has undergone internal review in the LSST Dark Energy Science Collaboration. We would like to thank the internal reviewers Adriano Agnello, Elizabeth Buckley-Geer, and Christopher Fassnacht for their insightful comments.

The NODE TO JOY package (Park et al. 2023) is heavily based on the Python libraries PYTORCH (Paszke et al. 2019), PYTORCH GEOMETRIC (Fey & Lenssen 2019), SCIPY (Virtanen et al. 2020), and NUMPY (Harris et al. 2020). We thank the developers of these libraries for making the code publicly available.

J.W.P. developed the NODE TO JOY package (Park et al. 2023) containing the data simulation, model training, and hierarchical inference; performed the analyses; and wrote the main text. S.B. contributed to the design, scope, analysis, and writing. M.U. implemented the photometric noise model and contributed to the design, scope, analysis, and writing. M.C. provided input on hierarchical inference. S.W.C. was supported by the National Science Foundation to perform the analyses; and wrote the main text. A.A. advised on method development and model architecture. A.R. advised on interpretation and presentation of results.

The DESC acknowledges ongoing support from the Institut National de Physique Nucléaire et de Physique des Particules in France; the Science & Technology Facilities Council in the United Kingdom; and the Department of Energy, the National Science Foundation, and the LSST Corporation in the United States. DESC uses resources of the IN2P3 Computing Center (CC-IN2P3–Lyon/Villeurbanne—France) funded by the Centre National de la Recherche Scientifique; the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231; STFC DiRAC HPC Facilities, funded by UK BIS National E-infrastructure capital grants; and the UK particle physics grid, supported by the GridPP Collaboration. This work was performed in part under DOE Contract DE-AC02-76SF00515.

**Appendix A**

**Lensing Formalism**

In this section, we review the gravitational lensing formalism for multiple lens planes, closely following the treatment in Das & Bode (2008). We begin with a general description in Section A.1, proceed to a widely used numerical approximation in Section A.2, and draw connections to our lensing quantities of interest in Section A.3.

**A.1. General Description of Lensing**

Consider a photon that reaches the observer at an angular direction of $\theta$ on the image plane. Suppose the source is located at a radial comoving distance $\eta$ relative to the observer and at an angular position $\beta$ on the source plane. The so-called lens equation maps $\theta$ to $\beta$:

$$\beta(\theta) = \theta - \alpha(\theta).$$  \hspace{1cm} (A1)

Here, $\alpha(\theta)$ is the deflection angle due to lensing, which satisfies

$$\alpha(\theta) = \nabla \psi(\theta),$$  \hspace{1cm} (A2)

where $\psi(\theta)$ is the effective lensing potential out to comoving distance $\eta$. Defining $D(\eta)$ as the comoving angular diameter distance corresponding to $\eta$, we can express $\psi(\theta)$ as a two-dimensional projection of the Newtonian potential $\Psi$ (Equation 3). Then Equation (A1) can be rewritten as

$$\beta(\theta) = \theta - 2 \int_0^\eta \frac{D(\eta_0 - \eta')}{D(\eta')} \nabla_\theta \psi(\eta_0, \eta').$$  \hspace{1cm} (A3)

**A.2. Multiplane Raytracing**

The integral in Equation (A3) is numerically solved by dividing the radial interval between the observer and the source into a finite number of concentric shells, via multiplane raytracing (Blandford & Narayan 1986; Schneider et al. 1992; Seitz & Schneider 1994; Jain et al. 2000). Suppose there are $K$ such shells, indexed by $k$ in increasing order from the observer to the source. Denote the lower and upper comoving distances of shell $k$ as $\eta_L(k)$ and $\eta_U(k)$, respectively, such that $\eta_L(0) = 0$ and $\eta_U(K-1) = \eta$. The continuous deflection $\alpha(\theta)$ experienced by the photon becomes approximated as discrete deflections $\alpha(k)(\theta(k))$.

Equation (A2) turns into

$$\alpha(k)(\theta(k)) = \nabla \psi(k)(\theta(k)),$$  \hspace{1cm} (A4)

where the discrete analog of the lensing potential $\psi(\theta)$ in Equation (3) is now

$$\psi(k)(\theta(k)) = \frac{2}{D(\eta(k))} \int_{\eta(k) - \frac{\Delta \eta}{2}}^{\eta(k) + \frac{\Delta \eta}{2}} d\eta' \psi(\eta', \eta').$$  \hspace{1cm} (A5)

Given the discretized deflection angles $\alpha(k)(\theta(k))$, we can raytrace the observed angular position $\theta(0)$ back through the series of lens planes:

$$\theta(k) = \theta(0) - \sum_{i=0}^{k-1} \alpha(i)(\theta(i)),$$  \hspace{1cm} (A6)

for $k = 1, \cdots, K$. In particular, for the source located at shell $k = K$, we have

$$\beta = \theta(K) = \theta(0) - \sum_{i=0}^{K-1} \alpha(i)(\theta(i)).$$  \hspace{1cm} (A7)

Naively applying the iterative scheme in Equation (A7) is computationally prohibitive for large $K$. For deep, full-sky simulations, there exist iterative techniques that reduce the number of arithmetic operations and memory.
A.3. Lensing Quantities

Our target lensing quantities are the convergence and shear, which describe the strength and direction, respectively, of the lensing effect along the entire LOS. They can be identified in the distortion matrix $\Gamma$, obtained by differentiating Equation (A3) with respect to $\theta$ (Schneider et al. 1992):

$$\Gamma_\theta(\theta) \equiv \frac{\partial \beta_\theta(\theta)}{\partial \theta_j}.$$  \hspace{1cm} (A8)

The lensing distortion can thus be decomposed into an isotropic change in area, parameterized by the convergence $\kappa = \kappa(\theta)$, and an area-preserving change in shape, parameterized by the shear components $\gamma_{1,2} = \gamma_{1,2}(\theta)$ forming the complex shear $\gamma = \gamma_1 + i\gamma_2$. In particular, $\psi$ satisfies the Poisson equation, i.e.,

$$\kappa(\theta) = \frac{1}{2} \nabla^2 \psi(\theta).$$ \hspace{1cm} (A9)

Predicting the lensing signal of halos entails relating their mass density to $\kappa$. In the Born approximation limit, $\kappa$ is the weighted sum of the projected surface mass densities of individual halos. Denoting the three-dimensional mass density of the individual halos, indexed by $n$, as $\rho_0(\eta, \theta)$, we have

$$\kappa(\theta) = \frac{4\pi G \rho_0}{c^2} \int_0^{n} d\eta \int d^2 \theta \rho_0(\eta', \theta').$$ \hspace{1cm} (A10)

Note that the Born approximation is not strictly valid in the strong lensing regime, as the light paths undergo significant perturbations (Bar-Kana 1996; Birrer et al. 2017). Throughout the paper, we assume the best-fit WMAP-7 cosmology (Komatsu et al. 2011) given by $\omega_{\text{cdm}} = 0.1109$, $\omega_b = 0.02258$, $n_s = 0.963$, $h = 0.71$, $\sigma_8 = 0.8$, and $w = -1.0$.

A.4. Raytracing Numerics

Note that $\kappa$ and $\gamma_{1,2}$ contain second-order gradients of the lensing potential $\psi$. In full-sky simulations, we can obtain these vector fields on the sphere by generating maps of the gradients with respect to the spherical harmonics. The computation can be performed rapidly using the fast Fourier transform on sky pixelization schemes such as Hierarchical Equal Area iso-Latitude Pixelation of a sphere (HEALPIX; Hockney & Eastwood 1988; Lewis 2005). The deflection field $\alpha(\theta)$ and distortion matrix $\Gamma(\theta)$ are first evaluated on a mesh grid, after which multiple rays can be traced backward in parallel. The choice of mesh spacing, which we will call $\delta$, determines the spatial resolution of the projected matter density.

In principle, all halos in the foreground of the source contribute to lensing, but the effect becomes smaller with increasing projected distance from the source. A cutoff in the multipole ($l_{\text{max}}$) smoothes out the effects within these angular scales. The cutoff determines the “field of view” of the light cone, or the sky area around an angular position $\theta'$ that contributes lensing potential to the calculation of $\Gamma(\theta')$. Let us denote the average field of view as $D$. COSMODC2 uses $l_{\text{max}} = 8000$, which corresponds to $D \sim 1.5'$. For smaller light cones with $D$ spanning a few arcminutes, flat-sky approximations may suffice. The field of view is divided into a rectilinear grid, from which the gradients are computed numerically using finite differences. Though not exact, an analog of $\delta$ in this approximation will be the grid spacing. Similarly, the size of the flat field is akin to $D$ operationally defined above.

A.4.1. Structure-enhanced Raytracing

Defining the gravity-only component of our mock universe is a large-scale cosmological N-body simulation called the Outer Rim (Heitmann et al. 2019). It sampled a volume of $(4.225 \text{ Gpc})^3$ with $10,240^3$ particles, yielding a mass resolution of $m_p = 2.6 \times 10^9 M_\odot$. The COSMODC2 galaxy catalog was built on top of the Outer Rim particles using semianalytic galaxy models (Korytov et al. 2019).

In addition to galaxy and halo information, COSMODC2 characterizes each galaxy with weak lensing shear and convergence. The lensing pipeline of COSMODC2 computed these quantities using a multiplane raytracing algorithm outlined in Section A.2. In particular, the surface densities were evaluated on an HEALPIX (Gorski et al. 2005) grid of $\text{NSIDE} = 4096$, which corresponds to a resolution of $\delta \sim 51''$ (P. Larsen et al. 2019, in preparation).

For galaxy-scale strong lensing studies, we instead require lensing statistics that vary on smaller, 1'' scales. At the same time, we must include the impact of large-scale structure at large angular scales, which are already captured in the COSMODC2 $\kappa$, $\gamma_{1,2}$ values. To obtain the structure-enhanced $\kappa_{\text{ext}}$, we raytraced through the Outer Rim halos along the sight line of each source galaxy at a resolution of $\delta = 1''$, while subtracting out the effect of any extra mass we were adding in the process. This subtraction scheme ensures that the mean curvature of the universe equals that imposed by the background, which is true if and only if the mean convergence of all angular directions to all redshifts is zero (Birrer et al. 2017). That is, our structure-enhanced convergence values must satisfy

$$\int d\eta \int d\theta \kappa(\theta; \eta) = 0,$$ \hspace{1cm} (A11)

where $\kappa(\theta; \eta)$ denotes the convergence evaluated at the angular position $\theta$ and comoving distance $\eta$. We assume that the COSMODC2 convergence values already satisfy this condition. In the following, we describe the process of structure-enhanced raytracing in detail.

Let us index each sight line by $s$. For a given source galaxy at angular position $\theta_s$ and redshift $z_{\text{src}}$, defining the sight line, we first queried halos with redshifts $z < z_{\text{src}}$ and masses $M_{200} > 10^{11} M_\odot$, located inside an aperture of fixed radius 1.5'. These halos constituted the light cone. The aperture size was chosen to match the COSMOCD2 field of view, $D \sim 1.5'$. The mass cut is a numerical choice, built in to reduce the computation time raytracing over halos with insignificant masses. We determined the mass threshold of $10^{11} M_\odot$ via a convergence test; it was increased from the base value of $10^{10} M_\odot$ until the $\kappa_{\text{ext}}$ value calculated within the aperture remained constant. On average, there were about 500 halos per light cone.

We then performed full multiplane raytracing from the source plane $z_{\text{src}}$ through each of the deflector halo planes using the strong lensing simulation package LENSTRONOMY (Birrer & Amara 2018). Surface densities were evaluated at a resolution of 1'', representing a structural enhancement of a factor of $\sim 51$ compared to COSMODC2. We then evaluated the...
convergence at the center of our aperture, which we denote \( \kappa_{\text{render}} \).

Adding \( \kappa_{\text{render}} \) to the original COSMODC2 convergence \( \kappa_{\text{DC2}} \), would capture both small- and large-scale fluctuations, as desired, but erroneously double count the Outer Rim particles already included in the \( \kappa_{\text{DC2}} \) computation. To preserve the mean mass in the universe, we employ the following calibration scheme. We render the halos at random positions within the aperture, compute the resulting convergence at the aperture center for each realization \( i \), which we denote \( \kappa_s^{(i)} \), and compute the mean across \( N \) realizations:

\[
\langle \kappa_s^{(i)} \rangle = \frac{1}{N} \sum_{i=1}^{N} \kappa_s^{(i)}. \tag{A12}
\]

Moving the halos around corresponds to an MC integration over all angular positions \( \theta \), i.e., \( \int d\theta \approx \frac{1}{N} \sum_{i=1}^{N} \). Our final, calibrated convergence value for a sight line \( s \) results from subtracting off this quantity:

\[
\kappa_s = \kappa_{\text{DC2}} + \kappa_{\text{render}} - \langle \kappa_s^{(i)} \rangle. \tag{A13}
\]

Assuming that we sample our sight lines uniformly across the sky, we have \( \sum_s \propto \int d\theta \), and since we assumed that the COSMODC2 values satisfied Equation (A11), we can claim

\[
\sum_s \kappa_s \approx \sum_s \kappa_{\text{DC2}} = 0. \tag{A14}
\]

Combining Equation (A13) and (A14) gives an important validation test of our calibration scheme

\[
\sum_s \kappa_{\text{render}} \approx \sum_s \langle \kappa_s^{(i)} \rangle > 1. \tag{A15}
\]

A.4.2. Halo Rendering

To generate the halo catalogs, Korytov et al. (2019, henceforward “COSMODC2 team”) ran a parallel, tree-based friends-of-friends (FoF) halo finder on the Outer Rim simulation with dimensionless linking length \( b = 0.168 \) and minimum requirement of 20 particles per halo. All particles, not just those in the original FoF halo, were counted in radial shells centered on the point of minimum potential. They then constructed halo merger trees using a particle-membership algorithm (Rangel et al. 2017). Their halo light cone was the product of tiling the simulation box in space to build a greater volume and applying a parallel solver that linearly interpolated halo positions between adjacent snapshot positions. The light cone filled one octant (\(~5000\text{deg}^2\)) of the sky and had a depth of \( z = 3 \). Although the original COSMODC2 lensing pipeline evaluated surface densities directly from the Outer Rim particles, we take as input the auxiliary halo catalog generated from running a halo finder on the particles. We treat each halo in the catalog as a Navarro–Frenk–White (NFW) mass distribution (Navarro et al. 1997). Its density can be written as

\[
\rho(r) = \frac{\delta_c \rho_c}{\left( \frac{r}{r_s} \right)^{1 + \gamma}}, \tag{A16}
\]

where \( \rho_c = 3H^2(\Omega)/8\pi G \) is the critical density, for the Hubble parameter \( H(z) \) at the halo redshift \( z \). The dimensionless characteristic density parameter \( \delta_c \) sets the normalization. The scale radius \( r_s \) is defined as the radius where the logarithmic profile slope \( n_{\text{eff}} = d \ln \rho/d \ln (r/r_s) = -2 \). Note that the slope falls off as \( n_{\text{eff}} \rightarrow -3 \) for \( r/r_s \gg 1 \). Here, \( \delta_c, r_s \) are the two free parameters that characterize each halo.

An alternative, and arguably more interpretable, parameterization uses the halo concentration and the integrated halo mass, for some choice of the dimensionless overdensity parameter \( \Delta \). The halo concentration \( c_\Delta \) is a dimensionless shape parameter defined as

\[
c_\Delta \equiv \frac{r_s}{r_c}, \tag{A17}
\]

where \( r_c \) is the radius inside of which the halo mass density is \( \Delta \times \rho_c \). Suppose we have the mass contained within \( r_\Delta \), i.e., \( M_\Delta \equiv \int_0^{r_\Delta} 4\pi r^2 \rho(r)dr \). Given \( M_\Delta \), the value of \( r_\Delta \) can be determined using

\[
M_\Delta = \frac{4}{3} \pi r_\Delta^3 \rho_c \Delta 

\Rightarrow r_\Delta = \left( \frac{3}{4\pi \rho_c \Delta} M_\Delta \right)^{1/3}. \tag{A18}
\]

The halo concentration \( c_\Delta \) is related to \( \delta_c \) as

\[
\delta_c = \frac{3}{4\pi \Delta} \ln(1 + c_\Delta) - \frac{c_\Delta}{1 + c_\Delta}. \tag{A19}
\]

The new parameterization is thus

\[
\rho(r) = \frac{\Delta \rho_c}{3} \left[ \ln(1 + c_\Delta) - \frac{c_\Delta}{1 + c_\Delta} \right]^{-1}

\times \left( \frac{r}{r_\Delta} \right)^{-1} \left( \frac{1}{c_\Delta} + \frac{r}{r_\Delta} \right)^2. \tag{A20}
\]

Taking Equations (A18) and (A20) together, the NFW profile with parameters \( r_c, \delta_c \) in Equation (A16) can be described completely by \( c_\Delta, M_\Delta \) instead. We make the conventional choice of \( \Delta = 200 \) in this paper.

The COSMODC2 catalog lists the FoF mass with dimensionless linking length \( b = 0.168 \) (\( M_{\text{FoF},0.168} \)) so we approximate \( M_{200} \) with this mass value, since an exact conversion between \( M_{\text{FoF},0.168} \) and \( M_{200} \) does not exist. To assign the concentration \( c_{200} \), we use the \( c_{200} = M_{200}/M_\ast \) fit derived from the Outer Rim halos (Child et al. 2018):

\[
c_{200} = A \left[ \left( \frac{M_{200}/M_\ast}{b} \right)^m \left( 1 + \left( \frac{M_{200}/M_\ast}{b} \right)^m \right) - 1 \right] + c_0

b = \frac{M_r}{M_\ast}. \tag{A21}
\]

where the fit parameters were \( m = -0.10, A = 3.44, b = 430.49 \), and \( c_0 = 3.19 \). The individual dispersion in \( c_{200} \) was \( c_{200}/3 \).

Granted, describing individual halos as spherical NFW profiles is an idealization. A more realistic description would be a prolate ellipsoid with a major axis length roughly twice as long as the minor axis (Jing & Suto 2000). Halo shapes and profiles are also highly variable, depending in part on whether the halos are dynamically relaxed (White 2002; Lukić et al. 2009).
Appendix B
Photometric Catalog

B.1. Galaxy–Halo Connection

First, we briefly summarize how galaxies were painted on top of the halos in COSMODC2 to yield the galaxy catalog containing the broadband filter magnitude information. Readers are referred to Korytov et al. (2019) for further details. The galaxy–halo connection encoded in COSMODC2 is one that the BGNN, as well as the summary statistics matching, implicitly learns in order to map the photometric observation space to the \( \kappa \) space.

Producing the accompanying galaxy catalog from the halo catalog involved modeling the galaxy–halo connection with a hybrid of empirical and semianalytic models. The UniverseMachine synthetic galaxy catalog provided the empirical model for predicting the star formation history of galaxies (Behroozi et al. 2019). It was chosen because it captures a wide range of statistics summarizing the observed galaxy distribution across redshift, including stellar mass functions, quenched fractions, and the dependence of two-point clustering on the star formation rate. Using the GalSampler technique, the COSMODC2 team matched every halo in the halo light cone to a suitable UniverseMachine (Behroozi et al. 2019) galaxy based on the halo mass \((M_{\text{halo}})\) and assigned the galaxy’s star formation rate (SFR) and total stellar mass \((M_*\)) to the halo. By construction, this technique preserved the halo mass dependence of the two assigned properties, \(P(\text{SFR}, M_*/M_{\text{halo}})\), as well as that of the UniverseMachine halo occupation statistics, \(p(N_{\text{halo}}|M_{\text{halo}})\). For the UniverseMachine \(M_* - M_{\text{halo}}\) relation for redshifts \(z = 0\) to \(z = 10\), see Behroozi et al. (2019).

To assign more complex properties, such as those due to galaxy mergers or metal production, the empirically modeled galaxies were matched to those in the Galacticus library (Benson 2012). Each galaxy was modeled as a mixture of two Sérsic components, i.e., a de Vaucouleurs bulge plus exponential disk.

B.2. Galaxy Catalog Properties

The COSMODC2 galaxy catalog covered a 440 deg\(^2\) field spanning \(0 < z < 3\). Each galaxy was characterized by various properties governing their surface brightness profiles. Of these properties, we took the projected coordinates (R.A./decl.) and LSST filter magnitudes \((ugrizY)\) to construct the photometric catalog that was input to the network. More precisely, we drew sight lines at the positions of random source galaxies in COSMODC2 with fixed redshift \(z_{\text{src}} \approx 2\) and, for an aperture of fixed size around a given sight line, compiled the catalog information of galaxies within the aperture except the source galaxy. We included galaxies in the foreground of the source \((z < z_{\text{src}})\), which contribute lensing, as well as those in the background \((z > z_{\text{src}})\). The resulting subcatalog represented the astrometry and broadband filter magnitudes of luminous tracers along an LOS being queried.

To simulate photometric noise, we added magnitude-dependent LSST Y5 noise to the ground-truth \(ugrizY\) magnitudes (Ivezić et al. 2019). The noise model is plotted in Figure 2. We applied a depth cut of \(i < 25.3\), corresponding to the LSST \(i\)-band gold sample (Gorayek et al. 2014).

ORCID iDs

Ji Won Park @ https://orcid.org/0000-0002-0692-1092
Simon Birrer @ https://orcid.org/0000-0003-3195-5507
Miles Cranmer @ https://orcid.org/0000-0002-6458-3423
Adriano Agnello @ https://orcid.org/0000-0001-9775-0331
Sebastian Wagner-Carena @ https://orcid.org/0000-0001-5039-1685
Philip J. Marshall @ https://orcid.org/0000-0002-0113-5770
Aaron Roodman @ https://orcid.org/0000-0001-5326-3486

References

Bar-Kana, R. 1996, Apj, 468, 17
Battaglia, P. W., Hamrick, J. B., Bapst, V., et al. 2018, arXiv:1806.01261
Behroozi, P., Wechsler, R. H., Hearin, A. P., & Conroy, C. 2019, MNRAS, 488, 3143
Benson, A. J. 2012, NewA, 17, 175
Bevington, P. R., & Robinson, D. K. 2003, (New York: McGraw-Hill)
Birrer, S., & Amara, A. 2018, PDU, 22, 189
Birrer, S., Dhawan, S., & Shajib, A. J. 2022, ApJ, 924, 2
Birrer, S., Shajib, A., Galan, A., et al. 2020, A&A, 643, A165
Birrer, S., Treu, T., Rusu, C., et al. 2019, MNRAS, 484, 4726
Birrer, S., Welschen, C., Amara, A., & Refregier, A. 2017, ICAP, 2017, 049
Blandford, R., & Narayan, R. 1986, Apj, 310, 568
Blandford, R., Surpi, G., & Kundic, T. 2001, in ASP Conf. Ser. 237, Gravitational Lensing: Recent Progress and Future Goals, ed. T. G. Brainerd & C. S. Kochanek (San Francisco, CA: ASP), 65
Buckley-Geer, E., Lin, H., Rusu, C., et al. 2020, MNRAS, 498, 3241
Chang, C., Pujol, A., Gaztanaga, E., et al. 2016, MNRAS, 459, 3203
Chang, C., Pujol, A., Mawdsley, B., et al. 2018, MNRAS, 475, 3165
Chang, C., Vikram, V., Jain, B., et al. 2015, PhRvL, 115, 051301
Chamock, T., Perreault-Levasseur, L., & Lanusse, F. 2020, arXiv:2006.01490
Chen, G. C., Faussnacht, C. D., Suyu, S. H., et al. 2019, MNRAS, 490, 1743
Child, H. L., Habib, S., Heitmann, K., et al. 2018, ApJ, 859, 55
Collett, T. E. 2015, ApJ, 811, 20
