The Astrophysical Journal, 910:39 (22pp), 2021 March 20 https://doi.org/10.3847/1538-4357/abdfc4

Large-scale Gravitational Lens Modeling with Bayesian Neural Networks for Accurate and Precise Inference of the Hubble Constant

Ji Won Park[1,2], Sebastian Wagner-Carena[1,2], Simon Birrer[1,2], Philip J. Marshall[1,2], Joshua Yao-Yu Lin[3], and Aaron Roodman[1,2]
(The LSST Dark Energy Science Collaboration)
[1] Kavli Institute for Particle Astrophysics and Cosmology, Department of Physics, Stanford University, Stanford, CA 94305, USA
[2] SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA
[3] University of Illinois at Urbana-Champaign, Champaign, IL 61820, USA
Received 2020 December 3; revised 2021 January 12; accepted 2021 January 23; published 2021 March 24

Abstract

We investigate the use of approximate Bayesian neural networks (BNNs) in modeling hundreds of time delay gravitational lenses for Hubble constant (H_0) determination. Our BNN was trained on synthetic Hubble Space Telescope quality images of strongly lensed active galactic nuclei with lens galaxy light included. The BNN can accurately characterize the posterior probability density functions (PDFs) of model parameters governing the elliptical power-law mass profile in an external shear field. We then propagate the BNN-inferred posterior PDFs into an ensemble H_0 inference, using simulated time delay measurements from a plausible dedicated monitoring campaign. Assuming well-measured time delays and a reasonable set of priors on the environment of the lens, we achieve a median precision of 9.3% per lens in the inferred H_0. A simple combination of a set of 200 test lenses results in a precision of 0.5 $\mathrm{km\,s^{-1}\,Mpc^{-1}}$ (0.7%), with no detectable bias in this H_0 recovery test. The computation time for the entire pipeline—including the generation of the training set, BNN training and H_0 inference—translates to 9 minutes per lens on average for 200 lenses and converges to 6 minutes per lens as the sample size is increased. Being fully automated and efficient, our pipeline is a promising tool for exploring ensemble-level systematics in lens modeling for H_0 inference.

Unified Astronomy Thesaurus concepts: Hubble constant (758); Cosmology (343); Bayesian statistics (1900); Hierarchical models (1925); Strong gravitational lensing (1643); Publicly available software (1864)

1. Introduction

The recent widening of the "Hubble tension" signals the need for rigorous tests of systematics in all cosmographic probes. The discrepancy in Hubble constant (H_0) measurements between early- and late-universe probes now lies at the 4–6σ level (Verde et al. 2019). Particularly valuable in this context are strong gravitational time delays—observed when light from a variable source is lensed by a massive foreground object, creating multiple images with relative delays in photon arrival times (Refsdal 1964). As time delay cosmography is fully independent of H_0 determination methods using the local distance ladder and the cosmic microwave background (CMB), it can serve as a check against sources of bias that may be affecting either method.

The H0 Lenses in COSMOGRAIL's Wellspring (H0LiCOW) Collaboration inferred H_0 to 2.4% precision using six lenses in the flat Lambda cold dark matter (ΛCDM) cosmology (Wong et al. 2019). The uncertainty increases to 7%, however, when the assumptions on the radial mass density profile of the lenses are relaxed and one additional lens is included (Birrer et al. 2020). Further folding in the external information from 33 Sloan Lens Advanced Camera for Surveys galaxy–galaxy lenses without time delays (Bolton et al. 2008; Auger et al. 2009; Shajib et al. 2020b), the precision improves to 5% assuming that the deflector galaxies follow the same population statistics. According to Birrer & Treu (2020), a sample size of 40 time delay lenses and 200 galaxy–galaxy lenses can enable 1.2%–1.5% precision necessary to resolve the H_0 tension.

The current modeling cycle in time delay cosmography does not scale well to the prospects of upcoming large-scale surveys. The Legacy Survey of Space and Time (LSST) at the Vera Rubin Observatory is expected to discover tens of thousands of lens systems, among them hundreds of lensed quasars (Oguri & Marshall 2010; Collett 2015). To date, time delay cosmography has relied on a time-consuming and manual forward modeling of observations. This approach takes several months under expert monitoring. With automation efforts, which are underway, the time may be reduced to several weeks (Shajib et al. 2019).

The efficiency issues aside, the current method of fine-tuning each lens model on a case-by-case basis makes it difficult to conduct global sensitivity tests on the model assumptions. See Shajib et al. (2019) for a uniform forward modeling of 13 quadruply lensed quasars (quads), among the first efforts to capitalize on the self-similarity of quads for automated (and thus consistent) lens modeling. A joint inference over hundreds of lenses requires a computationally efficient method with a uniform approach to modeling, so that systematics can be probed in an ensemble of lenses within reasonable time.

Bayesian neural networks (BNNs) offer an efficient alternative to forward modeling (Denker & LeCun 1991). They are a probabilistic variant of deep neural networks, which have demonstrated state-of-the-art performance in extracting highly abstract information from complex image data. Hezaveh et al. (2017) and Levasseur et al. (2017) demonstrated the efficacy of BNNs in accurately and precisely characterizing the lens model parameter posterior probability density functions (PDFs) for individual lenses, assuming a singular isothermal ellipsoid (SIE) lens model. Not only do BNN-based methods preclude the need for human supervision, once trained, a BNN model

can be applied to thousands of lens systems within seconds on a single GPU.

This paper connects the progress in BNN-based lens modeling to the H_0 inference stage, by extending BNN lens modeling to use all the features in a time delay lensed active galactic nucleus (AGN) system and combining the posterior PDFs from that modeling in an industry-standard joint inference of H_0 from a plausible near-future ensemble.

We are guided by the following questions:

1. Can the BNN accurately characterize the individual lens model posterior PDFs, given our model assumptions?
2. If so, do the BNN-inferred lens model posterior PDFs enable unbiased H_0 recovery when propagated into a joint H_0 inference over 200 lenses?
3. How sensitive are the H_0 predictions to the factors that are often considered when selecting lenses for follow-up, namely, the exposure time, the lensed image configuration, and the Einstein ring brightness?
4. Is our method efficient enough to handle large-scale tests of systematics? What is the net speed increase over traditional methods?

The goal of accurate and precise H_0 recovery places extra demands on lens modeling. This drives us to relax some of the assumptions made in the previous literature on lens models that are input to neural networks. We adopt the power-law elliptical lens mass distribution (PEMD) (Barkana 1998), a more complex form of the mass density profile than the fixed-slope SIE model used by Hezaveh et al. (2017), Pearson et al. (2019), and Schuldt et al. (2021) for their neural networks. PEMD is the model family currently used by the H0LiCOW and Time-Delay Cosmography (TDCOSMO) collaborations in their time delay cosmography analyses (Wong et al. 2019; Birrer et al. 2020).

H_0 inference also requires precise source position recovery. As discussed in Birrer & Treu (2019), the precision required on the source position is on the order of milliarcseconds. The ability of BNNs to constrain source positions to this level of precision has not yet been tested, but the accuracy of the predicted time delays (and hence the inferred H_0 value) will depend critically on this.

Lastly, we include the lens light in the images. In Hezaveh et al. (2017), the lens light was removed from the images via independent component analysis before the images were passed into the neural network for training. Pearson et al. (2019) saw a 34% reduction in accuracy of lens model recovery for images with lens light included, but report that multiband imaging could alleviate the decrease in performance. In the present study we restrict ourselves to a single Hubble Space Telescope (HST) infrared (IR) band and postpone the investigation of multiple bands to further work.

In this paper, we demonstrate that BNN lens modeling successfully meets the above performance requirements defined by time delay cosmography. Given our model assumptions, BNNs can, in fact, characterize the posterior PDF with sufficient accuracy, so as to recover H_0 without bias from a joint 200-lens inference. The source code we developed for our work has also been released for public use. The methodology and software presented in this paper are inherently versatile and allow extensions in many directions, including the hierarchical inference setup we developed in Wagner-Carena et al. (2020). They promise to become core infrastructure in time delay cosmography, as the cosmology community prepares to beat down systematics for a large sample of lenses due to be available in a few years' time.

This paper is organized as follows. Section 2 details each step of our automated H_0 inference pipeline. In Section 3, we report H_0 recovery results on a set of 200 test lenses under varying noise levels, image configurations (double versus quad), and Einstein ring brightness. Section 5 places our findings within the larger context of BNN-aided time delay cosmography and outlines next steps.

2. Methods

This section details the steps used for constructing an H_0 inference pipeline with a BNN as the lens modeling engine, beginning with a brief theoretical background of time delay cosmography in Section 2.1. In Section 2.2, we state our assumptions about the lens population, instrument optics, observation conditions, and cosmology—all of which we used to simulate the lensed AGN images and time delays. Then, in Section 2.3, we explain how the BNN models the individual lens model posteriors. The BNN-inferred lens model posterior becomes propagated into H_0 inference; Section 2.4 describes this process on an individual lens level and Section 2.5 on the joint-sample level. The entire pipeline is illustrated in Figure 1 as a flowchart and a probabilistic graphical model (PGM).

The implementation of the whole pipeline, including the BNN lens modeling and H_0 inference, is available in the open-source Dark Energy Science Collaboration (DESC) Python package H0RTON.[4] To generate the training set, we developed another DESC Python package BAOBAB,[5] which wraps around the multipurpose lens modeling package LENSTRONOMY[6] (Birrer & Amara 2018) to render the images and compute the time delays.

2.1. Time Delay Cosmography

Let us begin by reviewing the basic principles of time delay cosmography (Refsdal 1964). Readers are referred to recent reviews, e.g., Treu & Marshall (2016), for more details. When light rays from a background source are deflected by some foreground lens, the light travel time from the source to the observer depends on both their path length and the gravitational potential they must traverse. Assuming a single thin, isolated lens, the excess time delay of an image at position $\boldsymbol{\theta}$ originating from a source at position $\boldsymbol{\beta}$ relative to an unperturbed path is

$$t(\boldsymbol{\theta}, \boldsymbol{\beta}) = \frac{D_{\Delta t}}{c}\phi(\boldsymbol{\theta}, \boldsymbol{\beta}), \tag{1}$$

where

$$\phi(\boldsymbol{\theta}, \boldsymbol{\beta}) = \left[\frac{(\boldsymbol{\theta} - \boldsymbol{\beta})^2}{2} - \psi(\boldsymbol{\theta})\right] \tag{2}$$

is the Fermat potential (Schneider 1985; Blandford & Narayan 1986) defined for the lensing potential $\psi(\boldsymbol{\theta})$, and $D_{\Delta t}$ is the time delay distance (Refsdal 1964; Schneider et al. 1992; Suyu et al. 2010). The time delay distance is defined as

$$D_{\Delta t} \equiv (1 + z_{\rm lens})\frac{D_{\rm d} D_{\rm s}}{D_{\rm ds}}, \tag{3}$$

[4] http://github.com/jiwoncpark/h0rton
[5] http://github.com/jiwoncpark/baobab
[6] https://github.com/sibirrer/lenstronomy

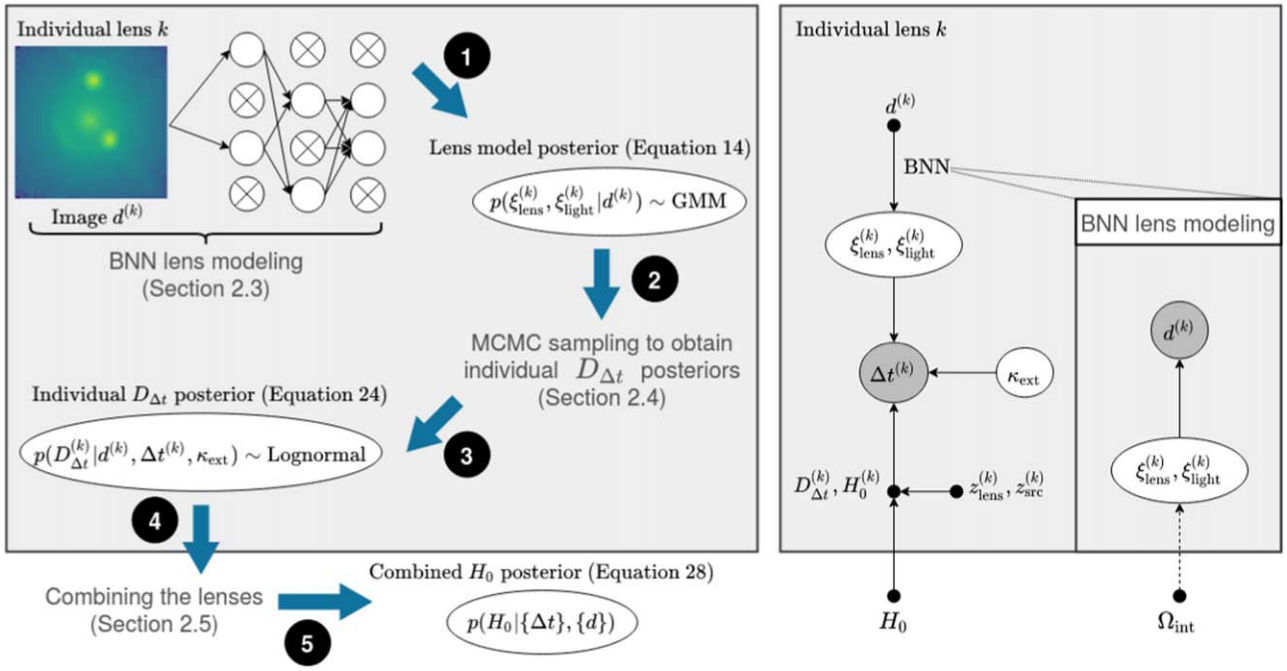

Figure 1. Left: illustration of the H_0 inference pipeline in the form of a flowchart. Right: the dependence relation shown as a PGM. Dots refer to delta functions, or fixed values; shaded ovals refer to observed values, or data; and unshaded ovals refer to random variables.

where $z_{\rm lens}$ is the lens redshift and $D_{\rm d}$, $D_{\rm s}$, $D_{\rm ds}$ are the angular diameter distances from the observer to the lens, the observer to the source, and from the lens to the source, respectively.

If the background source and the foreground lens are well aligned, we observe multiple images of the same background source. The position of the source with respect to the inner caustic determines whether there are two images, making the lensing system a "double," or four images, making it a "quad." The time delay between any pair of such lensed images is the difference of their excess time delays in Equation (1):

$$\Delta t_{i,j} = \frac{D_{\Delta t}}{c}[\phi(\boldsymbol{\theta}_i, \boldsymbol{\beta}) - \phi(\boldsymbol{\theta}_j, \boldsymbol{\beta})], \tag{4}$$

where $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_j$ are the positions of images i, j in the image plane.

If the source is variable, like an AGN, it is possible to measure the relative time delay $\Delta t_{i,j}$ by monitoring the fluxes of the images (Vanderriest et al. 1989; Schechter et al. 1997; Fassnacht et al. 1999; Kochanek et al. 2006; Courbin et al. 2011). The lensing potentials at the two image positions $\psi(\boldsymbol{\theta}_i)$, $\psi(\boldsymbol{\theta}_j)$ and the source position $\boldsymbol{\beta}$ can be determined by lens modeling, yielding a model of the relative Fermat potential $\Delta\phi_{i,j}$. Given the measured relative time delay and the constrained relative Fermat potential, we can constrain the time delay distance via

$$D_{\Delta t} = \frac{c\Delta t_{i,j}}{\Delta\phi_{i,j}}. \tag{5}$$

Being inversely proportional to the absolute distance scale, H_0 scales with $D_{\Delta t}$ as

$$H_0 \propto D^{-1}_{\Delta t}. \tag{6}$$

2.2. Simulated Data Set and Model Assumptions

The BNN, our lens modeling tool, requires a large training set that spans the target parameter space with sufficient density. The training set is necessarily synthetic because (1) fewer than 100 lensed AGN have been discovered to date and (2) it defines the models we assume for the lens mass, lens light, and source profiles during the inference stage. Our training set consists of 512,000 images, and we validate and test independent and identically distributed sets of 512 and 200 lenses, respectively.

2.2.1. Profile Assumptions

This study requires model profiles that are flexible enough to describe plausible lensing systems well but not too complex, so as to allow for simple interpretations in the basic parameter recovery tests. Earlier work on neural network-based lens modeling had focused on the SIE lens mass profile (Hezaveh et al. 2017; Levasseur et al. 2017; Pearson et al. 2019; Schuldt et al. 2021). As an update to this model, we allow the 3D power-law mass slope $\gamma_{\rm lens}$ to vary by adopting the PEMD (Barkana 1998). Note that the precision in $\gamma_{\rm lens}$ roughly translates to the precision in H_0, so demonstrating that we can recover $\gamma_{\rm lens}$ is crucial. The PEMD profile can be written in terms of six parameters as

$$\kappa(x, y) = \frac{3 - \gamma_{\rm lens}}{2}\left(\frac{\theta_E}{\sqrt{q_{\rm lens}x^2 + y^2/q_{\rm lens}}}\right)^{1-\gamma_{\rm lens}}, \tag{7}$$

where $q_{\rm lens}$ is the projected axis ratio and θ_E is the Einstein radius chosen such that it encloses the mean surface density in the spherical limit of $q_{\rm lens} = 1$. The coordinates (x, y) are the result of rotating the sky coordinates by the lens orientation angle $\phi_{\rm lens}$, so that the x-axis and the major axis of the lens align, and then centering them at the lens position $(x_{\rm lens}, y_{\rm lens})$. We also include the external shear component, parameterized by the shear modulus $\gamma_{\rm ext}$ and the shear angle $\phi_{\rm ext}$.

The lens galaxy light and the host galaxy light in our simulations follow the elliptical Sérsic distribution, which can

be expressed in terms of seven parameters as

$$I(x, y) = I_* \exp\left[-k\left\{\left(\frac{\sqrt{x^2 + y^2/q_*^2}}{R_*}\right)^{1/n} - 1\right\}\right], \quad (8)$$

where I_e is the surface brightness amplitude at the half-light radius R, k is a constant depending on the Sérsic index n such that R encloses half of the light, and q_* is the axis ratio. The coordinates (x, y) are as defined for Equation (7). As a simple approximation, we assume the lens light to share the centroid with the lens center. We parameterize the surface brightness amplitude I_* in terms of the magnitude m_* and convert into amplitude units using the instrument zero-point in order to render the image.

The AGN was modeled as an unresolved point source. To simulate microlensing, we added 10% Gaussian errors to the magnifications of the lensed AGN images.

The distribution of the model parameters in our training set serves as the implicit prior for our BNN. For the PEMD lens mass, external shear, and Sérsic lens light, we chose parameter distributions slightly broader than those in the Time Delay Lens Modeling Challenge (TDLMC; Ding et al. 2018). The distribution of the AGN host galaxy parameters was based on the estimates of source galaxy populations in galaxy–galaxy lenses presented in Collett (2015). See Table 1 for the specific choice of hyperparameters defining the implicit prior.

When the true input model parameters were drawn from the implicit prior, the ellipticities were parameterized in terms of the axis ratio $q_{\rm lens}$ and complex orientation angle $\phi_{\rm lens}$ as defined above. Similarly, the external shear was parameterized in terms of the shear modulus $\gamma_{\rm ext}$ and complex shear angle $\phi_{\rm ext}$. But the 2π-periodic property of the angles introduces degeneracies in target space that makes the BNN prediction task ill-defined. For training the BNN, we thus parameterized the target lens mass ellipticity and external shear in terms of the coordinate values in their respective spaces:

$$\begin{aligned} e_1 &= \frac{1 - q_{\rm lens}}{1 + q_{\rm lens}} \cos(2\phi_{\rm lens}) \\ e_2 &= \frac{1 - q_{\rm lens}}{1 + q_{\rm lens}} \sin(2\phi_{\rm lens}) \\ \gamma_1 &= \gamma_{\rm ext} \cos(2\phi_{\rm ext}) \\ \gamma_2 &= \gamma_{\rm ext} \sin(2\phi_{\rm ext}). \end{aligned} \quad (9)$$

Whereas there are known empirical covariances between subsets of our model parameters, such as a positive correlation between the ellipticities of lens mass and lens light, we assume the parameters to be a priori independent. This has the effect of reducing the efficiency of our training set by including some less plausible lenses in the BNN training. If the trained BNN were to be tested on real data, there would indeed be greater motivation to encode some covariance in the training set. The independence assumption is a safe choice for the purposes of our study, however, as it prevents the BNN from relying on the prescribed covariances when it generates its predictions. Even in applications when a more realistic training set is necessary, one should exercise caution when encoding the covariances; implicit priors that are too tight can introduce bias in the BNN parameter inference as well as hierarchical inference. In fact, as we demonstrated in Wagner-Carena et al. (2020), broad implicit priors are generally advised because the wide support aids in numerical stability when performing importance sampling for hierarchical inference.

Table 1
Parameter Distributions

Parameter	Distribution
Lens redshift	$z_{\rm lens} \sim N(0.5, 0.2)$
Source redshift	$z_{\rm src} \sim N(2, 0.4)$
Lens Galaxy	
Elliptical power-law mass	
Lens center ($''$)	$x_{\rm lens}, y_{\rm lens} \sim N(0, 0.07)$
Einstein radius ($''$)	$\theta_E \sim N(1.1, 0.1)$
Power-law slope	$\gamma_{\rm lens} \sim N(2.0, 0.1)$
Axis ratio	$q_{\rm lens} \sim N(0.7, 0.15)$
Orientation angle (rad)	$\phi_{\rm lens} \sim U(-\pi/2, \pi/2)$
Elliptical Sérsic light	
Magnitude	$m_{\rm lens*} \sim U(19, 17)$
Half-light radius ($''$)	$R_{\rm lens*} \sim N(0.8, 0.15)$
Sérsic index	$n_{\rm lens*} \sim N(3, 0.55)$
Axis ratio	$q_{\rm lens*} \sim N(0.85, 0.15)$
Orientation angle (rad)	$\phi_{\rm lens*} \sim U(-\pi/2, \pi/2)$
Environment	
External shear modulus	$\gamma_{\rm ext} \sim U(0, 0.05)$
Orientation angle (rad)	$\phi_{\rm ext} \sim U(-\pi/2, \pi/2)$
External convergence	$\kappa_{\rm ext} \sim N(0, 0.025)$
Host Galaxy	
Elliptical Sérsic light	
Host center ($''$)	$x_{\rm src}, y_{\rm src} \sim U(-0.2, 0.2)$
Host magnitude	$m_{\rm src} \sim U(25, 20)$
Half-light radius ($''$)	$R_{\rm src} \sim N(0.35, 0.05)$
Sérsic index	$n_{\rm src} \sim N(3, 0.5)$
Axis ratio	$q_{\rm src} \sim N(0.6, 0.1)$
Orientation angle (rad)	$\phi_{\rm src} \sim U(-\pi/2, \pi/2)$
AGN	
Point source	
AGN magnitude	$m_{\rm AGN} \sim U(22.5, 20)$

Note. The distribution of input parameters in the training, validation, and test data. $N(\mu, \sigma)$ denotes a normal distribution with mean μ and standard deviation σ and $U(a, b)$ denotes a uniform distribution with bounds a and b.

Throughout this paper, all magnitudes are given in the AB system with the WFC3/F160W filter zero-point of 25.9463.

2.2.2. Instrument and Observation Conditions

We simulated images obtained with the HST using the Wide Field Camera 3 (WFC3) IR channel in the F160W band, following the design of TDLMC. Dust extinction was not included, as it only has a weak effect in this filter. For simplicity, we approximated the point-spread function (PSF) drizzling process by convolving the unconvolved image with the drizzled HST PSF template provided as part of the TDLMC Rung 1. The effective pixel size of this drizzled PSF was $0.08''$ and we fixed the image size to 64×64 pixels. The PSF FWHM was in the range of $0.14'' \sim 0.16''$.

The PSF-convolved, noiseless images were stored so that noise could be added on the fly during training and testing. This setup exposes the network to different noise realizations of the

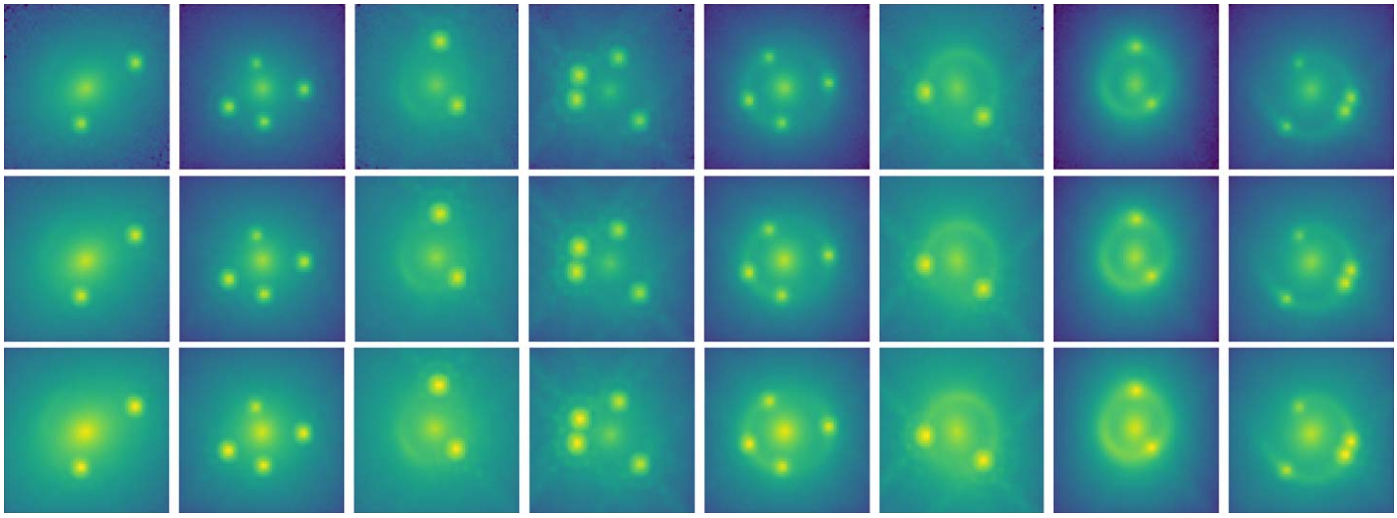

Figure 2. Gallery of eight images from our simulation, in log intensity scale. From the top row to the bottom, the exposure time varies as 0.5, 1, and 2 HST orbit(s). The columns are ordered such that, from left to right, the magnitude of the Einstein ring decreases. The eight images have been sampled from the test set, but this also serves as a visualization of the training set, as the test set and training set images are drawn from the same distribution in our study. The same color scale is used across the exposure times, for each lens.

same underlying system, which is known to help with generalization. It also precludes the need to generate new images for different noise levels. During training and testing, the stored noiseless images were scaled appropriately to simulate a new exposure time and BAOBAB efficiently computed the noise map on the GPU. The noise model included the background, readout, and Poisson CCD noise. We used the read noise of 4 e^- and CCD gain of 2.5 e^-/ADU, following the mean instrument statistics reported for WFC3/IR F160W (Dressel 2019). The sky brightness was calculated to be 22 mag arcsec^{-2} based on the zodiacal light estimation in Giavalisco et al. (2002), for the effective F160W filter wavelength of 1526.91 nm. The median signal-to-noise ratios (S/Ns) for the 0.5, 1, and 2 HST orbits were 4, 9, and 20, respectively, where signal was taken to be the sum of the pixels of the image with the lens light subtracted. Figure 2 displays images in the training set, with a range of exposure times and Einstein ring brightness.

2.2.3. Assumptions beyond the Images

The lens model parameters can be constrained from the simulated imaging observables alone, but in order to perform cosmological inference, we need to assign additional information to each lensing system: the redshifts, density of matter in the environment, and the time delay measurements.

The lens and source redshifts were drawn independently from Gaussian distributions centered at 0.5 and 2, respectively, as presented in Table 1. We assumed the availability of spectroscopic redshifts such that, during inference, the true lens and source redshifts were assumed to be known.

In principle, all masses in the lens environment and line of sight contribute lensing effects. We approximate the entire set of lensing mass as a single strong PEMD perturber plus external shear and convergence ($\kappa_{\rm ext}$). Effectively the density of a uniform mass sheet at the redshift of the main deflector, $\kappa_{\rm ext}$, affects the observed time delays but cannot be constrained by the image positions and fluxes—a phenomenon called "mass sheet degeneracy" (MSD) (Falco et al. 1985). For completeness, it should be noted that there is a separate aspect of MSD that is internal to the main deflector's mass profile, which can be constrained by kinematic tracers of the gravitational potential (Koopmans 2004; Saha & Williams 2006; Schneider & Sluse 2013; Birrer et al. 2016, 2020; Shajib et al. 2020a). We do not consider this internal mass sheet in our paper.

Failure to account for $\kappa_{\rm ext}$ can bias the H_0 inference. The effect of $\kappa_{\rm ext}$ on $D_{\Delta t}$ and H_0 is as follows:

$$D_{\Delta t} \propto \frac{1}{H_0} \propto \frac{1}{1-\kappa_{\rm ext}}. \tag{10}$$

In time delay cosmography, $\kappa_{\rm ext}$ is often estimated to a few-percent level using tracers of the large-scale structure, such as galaxy number counts (Rusu et al. 2017) or weak lensing of distant galaxies by all the mass along the line of sight (Tihhonova et al. 2018). Given our focus on assessing the impact of BNN lens modeling on H_0, however, we simply place a prior on $\kappa_{\rm ext}$. The images are generated with $\kappa_{\rm ext} = 0$ and we draw a true $\kappa_{\rm ext}$ from

$$\frac{1}{1-\kappa_{\rm ext}} \sim N(1, 0.025), \tag{11}$$

which, to first approximation, corresponds to

$$\kappa_{\rm ext} \sim N(0, 0.025) \tag{12}$$

and translates to an uncertainty of 2.5% on $D_{\Delta t}$. During inference, we use the exact input distribution in Equation (11) as the $\kappa_{\rm ext}$ prior. While Equations (11) and (12) are similar distributions in $\kappa_{\rm ext}$, we chose the former because it amounts to a Gaussian convolution in the $D_{\Delta t}$ posterior, by the relation in Equation (10), whereas the latter introduces non-Gaussianities in the $D_{\Delta t}$ posterior. Note that this choice is strictly numerical and not motivated by the physics. In Section 3, we discuss further the impact of non-Gaussianities in the individual $D_{\Delta t}$ posteriors on the combined H_0. Centering $\kappa_{\rm ext}$ at zero is also an artificial choice. The mean $\kappa_{\rm ext}$ for real lines of sight likely does not vanish for an ensemble of systems due to selection effects, e.g., lens galaxies tend to lie in groups (Blandford et al. 2001),

causing a slight preference for systems with overdense lines of sight (Collett & Cunnington 2016).

To simulate measurements of time delays, we artificially added Gaussian errors of 0.25 day to the true time delays, corresponding to zero bias and the smallest possible random errors under current monitoring strategies (Ding et al. 2018). We assumed such an optimistic scenario in time delay measurements so that the H_0 inference precision would be dominated by the capabilities of the BNN lens modeling rather than the time delay measurements.

The true cosmology was a flat ΛCDM cosmology with $H_0 = 70\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$ and $\Omega_\mathrm{m} = 0.3$. Throughout this study, Ω_m was assumed to be known and fixed so only H_0 was inferred.

2.3. Automated Lens Modeling with BNNs

What sets our method apart from the H0LiCOW method is that the lens model is estimated by the BNN rather than by forward modeling the images. As indicated in Figure 1, the trained BNN takes a test image and outputs the posterior over the target model parameters. The resulting lens model posterior is propagated into H_0 inference. There were 11 target model parameters: the six PEMD parameters, the two external shear parameters, the source position coordinates, and the host galaxy size R_src. Though not necessary for time delay cosmography, R_src was included in our predictions to allow the BNN to explicitly capture its known degeneracy with γ_lens.

In Section 2.3.1, we review the statistical framework of BNN posterior inference in the context of lens modeling. Section 2.3.2 briefly describes our choices in designing the network architecture and training. More implementation details are available in Appendix A.1.

2.3.1. Posterior Inference

BNNs represent a family of probabilistic neural networks that extends standard neural networks with posterior inference over the network weights (Denker & LeCun 1991). The uncertainty estimated by BNNs can be decomposed into two types: aleatoric and epistemic. Aleatoric uncertainty exists due to the intrinsic randomness in the underlying process. This type of uncertainty would persist even in the limit of infinite training data, i.e., perfect knowledge of the parameter-to-image mapping because various combinations of parameters may be capable of explaining a given test image. It encodes $\gamma_\mathrm{lens} - R_\mathrm{src}$ degeneracy; for instance, a thick Einstein ring in a test image may be explained by a shallow lens slope or a bigger source.

Aleatoric uncertainty is explicitly modeled as the width of the distribution over the target parameters. Improving on the work of Hezaveh et al. (2017) and Levasseur et al. (2017), who had used a Gaussian distribution with a diagonal covariance matrix, we adopt a mixture of two Gaussians (henceforward, Gaussian mixture model (GMM), each with a full covariance matrix, as we have done in Wagner-Carena et al. (2020). Explicitly, for a given test lens, we assumed the following form for the distribution over the target lens and light parameters $\xi^\star_\mathrm{lens}$, $\xi^\star_\mathrm{light}$ given the image $d^\star$ and a set of network weights W:

$$\begin{aligned} p(\xi^\star_\mathrm{lens}, \xi^\star_\mathrm{light}|d^\star, W) &= w_1(d^\star, W)\phi(\cdot|\mu_1(d^\star, W), \\ &\quad \Sigma_1(d^\star, W)) \\ &+ (1 - w_1(d^\star, W))\phi(\cdot|\mu_2(d^\star, W), \Sigma_2(d^\star, W)), \end{aligned} \tag{13}$$

where $\phi(\cdot\,|\mu, \Sigma)$ denotes the PDF of a p-dimensional Gaussian with mean $\mu \in \mathbb{R}^p$ and covariance $\Sigma \in \mathbb{R}^{p\times p}$ and the weight on the first Gaussian $w_1 \in (0, \frac{1}{2}]$. The BNN predicted μ_1, Σ_1, μ_2, Σ_2, and w_1 so the size of the output dimension was $p_\mathrm{out} = 2 \times \left[p + \frac{p(p+1)}{2}\right] + 1$ for the p target parameters. We had $p = 11$, so $p_\mathrm{out} = 155$.

Epistemic uncertainty, on the other hand, originates from limited training data or the choice of an imperfect model. It comes into play when the network attempts to generalize to regions outside the training set. In the context of machine learning, it is often described as a distribution over the network weights, post-training. Each realization of the weights corresponds to an alternative model, so integrating over this learned weight posterior is akin to Bayesian model averaging. Folding in the epistemic uncertainty, we have the full predictive distribution:

$$\begin{aligned} &p(\xi^\star_\mathrm{lens}, \xi^\star_\mathrm{light}|d^\star, \Omega_\mathrm{int}) \\ &= \int p(\xi^\star_\mathrm{lens}, \xi^\star_\mathrm{light}|d^\star, W)p(W|\Omega_\mathrm{int})\, dW, \end{aligned} \tag{14}$$

where we have made explicit the dependence on the specific training set by appropriately conditioning on the hyperparameters governing the implicit prior, Ω_int. Not modeling the epistemic uncertainty at all amounts to a simple conditional density estimation, where the weight posterior $p(W|\Omega_\mathrm{int})$ is a delta function. In standard neural networks, which only give point estimates for the target parameters, both $p(\xi^\star|d^\star, W)$ and $p(W|\Omega_\mathrm{int})$ are delta functions, so the predictive distribution in Equation (14) collapses to a delta function.

Consider the integral in Equation (14). An exact evaluation of this integral is intractable, as it requires averaging over all the weight configurations allowed by $p(W|\Omega_\mathrm{int})$. There exist several workarounds, including the Kronecker-factored Approximate Curvature (K-FAC) Laplace approximation (MacKay 1992; Ritter et al. 2018); Bayes by backprop (Blundell et al. 2015); stochastic Markov chain Monte Carlo (MCMC; Welling & Teh 2011); deep ensembles (Lakshminarayanan et al. 2017); and stochastic weight averaging Gaussian variants (Maddox et al. 2019; Wilson & Izmailov 2020). We opt for Monte Carlo (MC) dropout (Gal & Ghahramani 2016; Kendall & Gal 2017), however, for consistency with Wagner-Carena et al. (2020) and simplicity of implementation. In MC dropout, the weight posterior $p(W|\Omega_\mathrm{int})$ is replaced with the variational distribution $q_\theta(\hat{W}|\Omega_\mathrm{int})$ parameterized by θ:

$$\begin{aligned} q_\theta(\hat{W}|\Omega_\mathrm{int}) &= \prod_{i=1}^{L} q_\theta(\hat{W}_i|\Omega_\mathrm{int}) \\ \hat{W}_i &= W_i \cdot \mathrm{diag}(z_{i,j})_{j=1}^{K_i} \\ z_{i,j} &\sim \mathrm{Bernoulli}(p_i) \\ \theta &\equiv \{W_i, p_i\}_{i=1}^{L}, \end{aligned} \tag{15}$$

where i indexes the layer of the L-layer network and j the node in a given layer. Here, K_i denotes the number of nodes at layer i, such that the weight matrix for layer i is $W_i \in \mathbb{R}^{K_i \times K_{i-1}}$. When $z_{i,j} = 0$, the input node j in layer i is dropped out, i.e., set to zero. This form of the variational distribution arises from a mathematical result that a network with randomly dropped

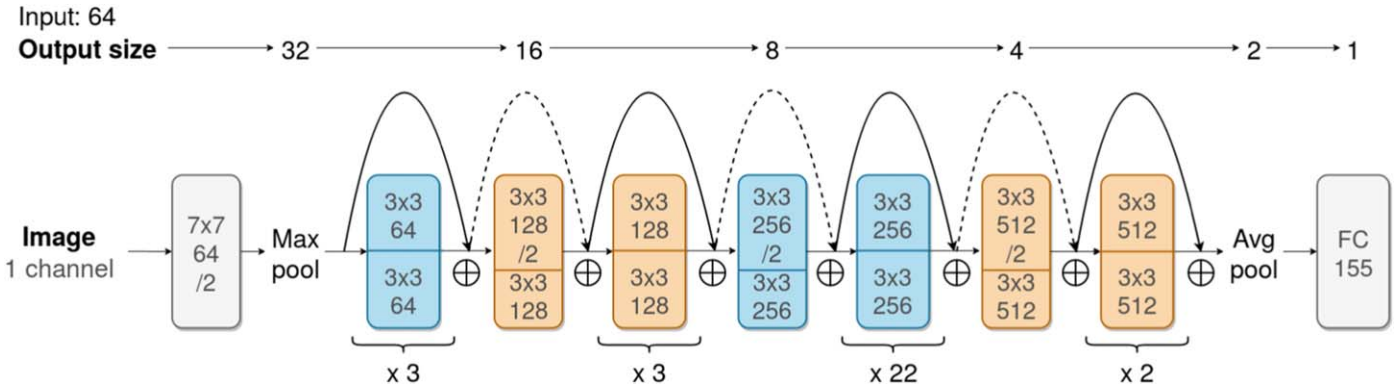

Figure 3. The `ResNet101` network architecture used for the convolutional engine of the BNN. The size of the square feature maps evolves through the layers as indicated on the top. Rectangular boxes contain convolutions of the indicated kernel size and channel number (width). Strides of 2 are denoted as /2. Note that blue and orange boxes are two stacked convolutions. Curved arrows indicate shortcut connections; the solid ones preserve the input feature dimension and dotted ones double the number of channels and halve the feature map resolution. Not shown are the 1D dropout layers, which were inserted before every convolutional layer and before the final fully connected layer. Batch normalization and the rectified linear unit (ReLU) layers followed each convolution as well.

weights is equivalent to a deep Gaussian process (Damianou & Lawrence 2013); see Gal & Ghahramani (2016) for the derivation.

To optimize θ, we minimize the Kullback–Leibler (KL) divergence between the true weight posterior $p(W|\Omega_{\rm int})$ and the variational approximation $q_\theta(\hat{W}|\Omega_{\rm int})$. Equivalently, the BNN minimizes the log evidence lower bound (ELBO) over the number (N) of examples in the training set $\{d^{(n)}, \xi_{\rm lens}^{(n)}, \xi_{\rm light}^{(n)}\}_{n=1}^{N}$:

$$\mathcal{L}(W) = -\sum_{n=1}^{N} \int q_\theta(\hat{W}|\Omega_{\rm int}) \log p(\xi_{\rm lens}^{(n)}, \xi_{\rm light}^{(n)}|d^{(n)}, \hat{W}) d\hat{W} + {\rm KL}(q_\theta(\hat{W}|\Omega_{\rm int})||p(\hat{W})), \quad (16)$$

where $p(W)$ is a prior on the network weights. To evaluate the first term in an unbiased way, we approximate each entry in the sum by MC integration with a single sample $\hat{W} \sim q_\theta(\hat{W}|\Omega_{\rm int})$. Then W can be updated via stochastic 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

$${\rm KL}(q_\theta(\hat{W}_i|\Omega_{\rm int})||p(\hat{W}_i)) \propto \frac{l^2(1-p_i)}{2N}||\hat{W}_i||^2 \quad (17)$$

when we assume a prior that can be factorized into a product of Gaussian priors in each layer. The length scale l is a hyperparameter that determines the width of the prior. Note that the dropout probability p is also a hyperparameter in the formulation introduced here. It is not optimized along with W during training and must be tuned manually as part of the hyperparameter search. We assume the same dropout probability $p_i = p_{\rm drop}$ for every layer i. For a given choice of $p_{\rm drop}$, l can be folded into the L_2 regularization strength hyperparameter $\lambda = l^2(1-p)/(2N)$.

The full predictive posterior with the variational approximation is thus

$$p(\xi_{\rm lens}^\star, \xi_{\rm light}^\star|\Omega_{\rm int}) = \int p(\xi_{\rm lens}^\star, \xi_{\rm light}^\star|d^\star, W) \times q_\theta(W|\Omega_{\rm int})\, dW. \quad (18)$$

In order to propagate this into an MCMC-based H_0 inference procedure, we need to be able to evaluate it. We do so via MC integration, i.e., by taking some S number of MC dropout iterates and averaging the resulting aleatoric portions of the posterior.

$$\hat{W}^{(s)} \sim q_\theta(\hat{W}|\Omega_{\rm int}), \quad s = 1, ..., S$$
$$p(\xi_{\rm lens}^\star, \xi_{\rm light}^\star|d^\star, \Omega_{\rm int}) \approx \frac{1}{S}\sum_{s=1}^{S} p(\xi_{\rm lens}^\star, \xi_{\rm light}^\star|d^\star, \hat{W}^{(s)}). \quad (19)$$

The value of S is determined by a convergence test—that is, it is increased from base values until the full predictive distribution no longer changes. The resulting approximation to the full predictive distribution is a mixture of $S \times 2$ Gaussians, where the factor of S comes from the epistemic MC dropout iterates and 2 from the aleatoric double-Gaussian parameterization. In particular, the predictive mean is

$$\mathbb{E}[\xi_{\rm lens}^\star, \xi_{\rm light}^\star|d^\star, \Omega_{\rm int}] \approx \frac{1}{S}\sum_{s=1}^{S} \mathbb{E}[\xi_{\rm lens}^\star, \xi_{\rm light}^\star|d^\star, \hat{W}^{(s)}]. \quad (20)$$

2.3.2. Network Architecture and Training

The convolutional engine of the BNN had the `ResNet101` architecture (He et al. 2015), modified from the TORCHVISION implementation (Marcel & Rodriguez 2010). The specific network architecture used for this paper is illustrated in Figure 3. See Appendix A.1.3 for more details on our choice of architecture.

It is common practice to transform the training images 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 target labels for the model parameters were normalized so that each parameter had a mean of 0 and standard deviation of 1 across the entire training set. Each input image $d \in \mathbb{R}^{64\times64}$ was also pixelwise transformed according to $\log(1+d_i)$, where d_i represents each pixel intensity value of d and is rescaled to the range [0, 1]. The log transformation was adopted so that the bright pixels in the cusp of the Sérsic or in the AGN images would not overwhelm the informative pixels in the Einstein ring.

The network was trained with the `ADAM` optimizer (Kingma & Ba 2014) for 50 epochs with the weight decay parameter

$\lambda = 1e-6$, batch size $B = 1024$, and initial learning rate of $\epsilon_0 = 5e-4$. Although the network was allowed to train for 50 epochs, we only saved the checkpoint with the best validation performance, which plateaued at 37–45 epochs for our experiments. The learning rate was reduced by a factor of 2 whenever the validation loss did not decrease for 50 minibatch updates, until it reached 1e − 5. The hyperparameters λ, B, and ϵ_0 were tuned via a random search on the validation set.

2.3.3. Calibration Metric

Recall that the MC dropout probability $p_{\rm drop}$ is also a hyperparameter that is tuned rather than optimized. Among the values 0.5%, 0.1%, and 0%, the value of 0.1% was found to be optimal for all three exposure times in our study—0.5, 1, and 2 HST orbits. To select a particular dropout probability, we used the confidence-frequency calibration, a semiquantitative metric introduced in Wagner-Carena et al. (2020). We reproduce the definition of this calibration metric here and state our own choices in using this metric.

Denote the N parameter samples drawn from the BNN posterior for some lens k as $\{\xi_n^{(k)}\}_{n=1}^{N}$. The true parameter value is $\xi_{\rm true}^{(k)}$. The metric asks: for a given percentage of the BNN posterior probability volume, p_X, what percentage of the samples, p_Y, contains the truth within this volume? If the posterior is perfectly calibrated, we would expect p_X of the samples to encompass the truth $p_Y = p_X$ of the time, for every value of p_X. We can apply this metric on the validation set as a whole by averaging the p_Y values evaluated on individual lenses. To wit,

$$p_Y^{(k)}(p_X) = \mathbb{1}\left\{\frac{\sum_{n=1}^{N}\mathbb{1}\{d(\xi_n^{(k)}) < d(\xi_{\rm true}^{(k)})\}}{N} < p_X\right\}$$

$$p_Y^{\rm val}(p_X) = \frac{1}{N^{\rm val}}\sum_{k=1}^{N^{\rm val}} p_Y^{(k)}(p_X), \tag{21}$$

where $1\{\cdot\}$ is an indicator function that evaluates to 1 when the argument is true and 0 otherwise, and $d(\xi)$ is a measure of distance of a particular point ξ from the posterior predictive mean given the posterior width.

Plotting $p_Y^{\rm val}$ for a grid of p_X values yields the calibration curve, to be presented and discussed in Section 3.1.2 in the context of evaluating the statistical consistency of BNN lens modeling. Regions of the curve with $p_Y^{\rm val} < p_X$ speak to an overconfident BNN, because there are not as many lenses with truth within the posterior volume p_X than there should be. Conversely, regions with $p_Y^{\rm val} > p_X$ indicate underconfidence.

There are many choices for the distance measure d. We use the Mahalanobis distance, a multidimensional generalization of the standard score measuring how many standard deviations away a point is from the mean.[7]

$$\begin{aligned} d(\xi^{(k)}) &\equiv \sqrt{(\xi^{(k)} - \mu^{(k)})^T[\Sigma^{(k)}]^{-1}(\xi^{(k)} - \mu^{(k)})} \\ \mu^{(k)} &\equiv \mathbb{E}[\xi^{(k)}|d^{(k)}, \Omega_{\rm int}] \quad \text{from Equation (20)} \\ \Sigma^{(k)} &\equiv {\rm Cov}_n\{\xi_n^{(k)}\}. \end{aligned} \tag{22}$$

[7] Levasseur et al. (2017) calibrated their $p_{\rm drop}$ using the 1D standard score. This calibration method was more appropriate for their choice of a diagonal Gaussian as the aleatoric portion of the posterior. For our study using the GMM parameterization, we require the multidimensional distance metric so that the parameter covariances can be taken into account.

2.4. Individual H_0 Inference

At the stage of inferring H_0, we use the lens model posterior obtained from the BNN to properly account for the uncertainties in the lens model parameters. In this section, we describe the inference procedure on the individual lens level: the arrow labeled "2" of the flowchart in Figure 1.

We make some simplifying assumptions in our inference. In order to focus on basic H_0 recovery, we fix $\Omega_{\rm m} = 0.3$ and infer only H_0 for each test lens. In doing so, we only use simulated images and time delay measurements, and do not include velocity dispersion or line-of-sight measurements in our modeling. Recall also that our training and test data were drawn from the same distribution. In terms of the conventions we introduced in Wagner-Carena et al. (2020), this setup translates to the assumption that the set of hyperparameters implicit in the training set equals that governing the test prior, i.e., $\Omega_{\rm int} = \Omega$. In addition, we do not place population-level hyperpriors on the individual model parameters; they are assumed to be known and accurate, and thus not varied in a hierarchical manner. This is the approach taken by the H0LiCOW Collaboration.

The posterior on the test set hyperparameters Ω can be written as

$$\begin{aligned} &p(\Omega|\Delta t^{(k)}, d^{(k)}) \\ &\propto p(\Omega)p(\Delta t^{(k)}, d^{(k)}|\Omega) \\ &\propto p(\Omega)\int p(\Delta t^{(k)}|\Omega, \xi_{\rm lens}^{(k)}, \kappa_{\rm ext}^{(k)}) \\ &\quad \times p(\xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}|d^{(k)}, \Omega_{\rm int}) \times \frac{p(\xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}|\Omega)}{p(\xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}|\Omega_{\rm int})} \\ &\quad \times p(\kappa_{\rm ext}^{(k)})d(\xi_{\rm lens}^{(k)})d(\xi_{\rm light}^{(k)})d\kappa_{\rm ext}^{(k)}, \end{aligned} \tag{23}$$

where k denotes a single test lens. The integral in Equation (23) represents the total likelihood for this lens. The data are the observed time delay(s) $\Delta t^{(k)}$, where $\Delta t^{(k)} \in \mathbb{R}^1$ for a double and $\Delta t^{(k)} \in \mathbb{R}^3$ for a quad, and the image $d^{(k)}$. The first term in the integral is the time delay likelihood, assumed to be diagonal Gaussian with an uncertainty of 0.25 day. The second-to-last line is the importance-weighted BNN-inferred lens model posterior, which serves as a prior in this level of inference. When the implicit prior differs from the test prior, the BNN posterior $p(\xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}|d^{(k)}, \Omega_{\rm int})$ must be divided out by the implicit prior the BNN was trained on and multiplied by the test prior. See Foreman-Mackey et al. (2014) and Wagner-Carena et al. (2020) for the derivation of importance weighting. The external convergence $\kappa_{\rm ext}$ and the lens model parameters $\xi_{\rm lens}^{(k)}$, $\xi_{\rm light}^{(k)}$ are nuisance hyperparameters that must be integrated out to obtain the population likelihood.

Applying our assumption of $p(\xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}|\Omega) = p(\xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}|\Omega_{\rm int})$ and paring down the target Ω to just H_0, Equation (23) can be greatly simplified to

$$\begin{aligned} &p(H_0|\Delta t^{(k)}, d^{(k)}) \\ &\propto p(H_0)p(\Delta t^{(k)}, d^{(k)}|H_0) \\ &\propto p(H_0)\int p(\Delta t^{(k)}|D_{\Delta t}^{(k)}(H_0), \xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}, \kappa_{\rm ext}^{(k)}) \\ &\quad \times p(\xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}|d^{(k)}) \\ &\quad \times p(\kappa_{\rm ext}^{(k)})d(\xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)})d\kappa_{\rm ext}^{(k)}. \end{aligned} \tag{24}$$

The notation $D_{\Delta t}(H_0)$ simply makes explicit that, with other cosmological parameters fixed, $D_{\Delta t}$ and H_0 have a one-to-one relation for a given lens.

The individual posterior in Equation (24) is difficult to evaluate due to the complicated dependence structure of $\Delta t^{(k)}$ but lends itself to sampling. We performed MCMC sampling over $D_{\Delta t}$ jointly with $\xi^{(k)}_{\rm lens}$, $\xi^{(k)}_{\rm light}$, with the following objective evaluated at each MCMC iteration:

$$p(\Delta t^{(k)}|D^{(k)}_{\Delta t}(H_0), \xi^{(k)}_{\rm lens}, \xi^{(k)}_{\rm light}, \kappa_{\rm ext}=0) \times p(\xi^{(k)}_{\rm lens}, \xi^{(k)}_{\rm light}|d^{(k)}). \quad (25)$$

Recall that we do not constrain $\kappa_{\rm ext}$ from data and instead assign a global prior of the form in Equation (11). To reduce the sampling space, $\kappa_{\rm ext}$ was assumed to be zero during MCMC sampling, and accounted for only in post-processing by multiplying each $D_{\Delta t}$ sample by $\frac{1}{1-\kappa_{\rm ext}}$ for multiple realizations of $p(\kappa_{\rm ext})$.

The first term in Equation (25), the time delay likelihood, is a one- or three-dimensional diagonal Gaussian PDF (depending on whether the lens is a double or a quad). The second term, the BNN-inferred lens model posterior in Equation (14), does not allow for an exact evaluation but can be approximated using MC integration with S dropout samples, as described in Equation (19). We used $S=12$, making the approximation a mixture of 24 Gaussians.

To minimize burn-in time, we initialized the walkers at the positions of the BNN posterior samples along the $\xi^{(k)}_{\rm lens}$, $\xi^{(k)}_{\rm light}$ dimensions. Along the $D_{\Delta t}$ dimension, the allowed range of the walkers was between 0 and 15,000 Mpc and the initial positions were also uniformly sampled in this range. We found that 18,000 samples gave good coverage of the 12 dimensional sample space (11 for $\xi^{(k)}_{\rm lens}$, $\xi^{(k)}_{\rm light}$ and 1 for $D^{(k)}_{\Delta t}$). Our implementation uses MCMC sampling modules in LENSTRONOMY, which uses EMCEE (Foreman-Mackey et al. 2013) to run the sampler. See Table 2 for a summary of the model parameters and their priors in the cosmological sampling stage.

Once the $D_{\Delta t}$ MCMC samples were generated this way for each lens, we stored them for the next step of joint-lens inference (Section 2.5). They were, effectively, samples from the individual $D_{\Delta t}$ posteriors $p(D^{(k)}_{\Delta t}|\Delta t^{(k)}, d^{(k)}) \propto p(D^{(k)}_{\Delta t})p(\Delta t^{(k)}, d^{(k)}|D^{(k)}_{\Delta t})$, when assuming a broad uniform prior $p(D^{(k)}_{\Delta t})$ in the range of 0–15,000 Mpc.

To obtain the individual H_0 posterior, the $D_{\Delta t}$ MCMC samples were converted into H_0 using the lens and source redshifts, assumed to be known. Then, we applied the uniform H_0 prior in the range 50–90 km Mpc^{-1} s^{-1}. A Gaussian fit to the resulting H_0 samples gave an estimate of the center and spread of H_0 posterior for each lens:

$$H^{(k)}_0 \sim N(\mu^{(k)}, \sigma^{(k)}). \quad (26)$$

2.5. Joint H_0 Inference

To perform joint inference on a sample of lenses, we combined the information from the individual $D_{\Delta t}$ posteriors, as indicated in arrow 5 of the flowchart in Figure 1. The H_0 posterior from a joint sample is

$$\begin{aligned} & p(H_0|\{\Delta t\}, \{d\}) \\ & \propto p(H_0)p(\{\Delta t\}, \{d\}|H_0) \\ & \propto p(H_0)\prod_k \int p(\Delta t^{(k)}|D^{(k)}_{\Delta t}(H_0), \xi^{(k)}_{\rm lens}, \xi^{(k)}_{\rm light}, \kappa^{(k)}_{\rm ext}) \\ & \quad \times p(\xi^{(k)}_{\rm lens}, \xi^{(k)}_{\rm light}|d^{(k)}) \\ & \quad \times p(\kappa^{(k)}_{\rm ext}) \quad d(\xi^{(k)}_{\rm lens}, \xi^{(k)}_{\rm light}) d\kappa^{(k)}_{\rm ext}, \end{aligned} \quad (27)$$

where k indexes the set of test lenses. This is identical to Equation (24) in form, except that our data now consist of the entire test set of observed time delays $\{\Delta t^{(k)}\}_{k=1}^{200}$ and images $\{d^{(k)}\}_{k=1}^{200}$. More simply, we can express this joint-sample posterior as

$$p(H_0|\{\Delta t\}, \{d\}) \propto p(H_0)\prod_k p(\Delta t^{(k)}, d^{(k)}|D^{(k)}_{\Delta t}(H_0)). \quad (28)$$

To generate H_0 samples from this joint-sample posterior, we MCMC sampled over H_0 by evaluating the following likelihood objective at each MCMC iteration:

$$\prod_k p(\Delta t^{(k)}, d^{(k)}|D^{(k)}_{\Delta t}(H_0)). \quad (29)$$

Doing so required likelihoods that could be evaluated. Recall from Section 2.4, that we stored the MCMC samples from individual $D_{\Delta t}$ posteriors. We had applied a broad, uniform prior on $D_{\Delta t}$, so that these samples could be reinterpreted as samples from the likelihoods $p(\Delta t^{(k)}, d^{(k)}|D^{(k)}_{\Delta t}(H_0))$. What remained was to fit appropriate distributions on these stored samples, so the likelihood could be evaluated. For our main analysis, we adopted the kernel-density estimate (KDE) using Gaussian kernels for its flexibility. The binning scheme followed Scott's normal reference rule (Scott 2015).

To assess the effect of the fit distribution on the joint-sample inference of H_0, we also experimented with two other, less flexible parameterizations of the $D_{\Delta t}$ likelihood. One was the Gaussian parameterization, by which the stored $D_{\Delta t}$ samples for each lens were interpreted to follow a Gaussian distribution and the two Gaussian parameters were fit. That is, we assumed

$$D^{(k)}_{\Delta t} \sim N(\mu^{(k)}_{D_{\Delta t}}, \sigma^{(k)}_{D_{\Delta t}}). \quad (30)$$

Table 2
Summary of Model Parameters

Parameter	Prior	Description
Flat ΛCDM cosmology		
H_0 (km s^{-1} Mpc^{-1})	$U(50, 90)$	H_0
$\Omega_{\rm m}$	$\delta(0.3)$	Mass density
Mass profile $\xi^{(k)}_{\rm lens}$, $\xi^{(k)}_{\rm light}$	BNN-inferred lens model posterior (see Equation (19))	PEMD, external shear, source position/size
Line of sight		
$\kappa_{\rm ext}$	$\frac{1}{1-\kappa_{\rm ext}} \sim N(1, 0.025)$	External convergence

Table 3
Summary of Experiments

Label	Exposure Time (HST Orbit)	Image Configuration	Number of Test Lenses	Number of Validation Lenses	Training Time (hr)	Inference Time (minutes lens^{-1})
A1	0.5	Both	200	512	5	6
A2	1	Both	200	512	5	6
A3	2	Both	200	512	5	6
B1	1	Doubles only	89	222	⋯	11
B2	1	Quads only	89	222	⋯	6

Note. Summary of experiments defined by lenses with varying exposure times (block A) and image configurations (block B). We report the median inference time across the sample of lenses.

Table 4
Prediction Accuracy of Individual Parameters

Experiment	γ_1	γ_2	$x_{\rm lens}(")$	$y_{\rm lens}(")$	e_1	e_2	$\gamma_{\rm lens}$	$\theta_E(")$	$x_{\rm src}(")$	$y_{\rm src}(")$	$R_{\rm src}(")$
0.5 HST orbit	0.012	0.013	0.002	0.001	0.024	0.025	0.056	0.006	0.006	0.007	0.03
1 HST orbit	0.012	0.013	0.002	0.002	0.025	0.025	0.056	0.006	0.007	0.006	0.03
2 HST orbits	0.012	0.013	0.002	0.002	0.023	0.024	0.055	0.006	0.006	0.007	0.03
Doubles	0.011	0.015	0.002	0.001	0.022	0.025	**0.064**	0.006	**0.008**	**0.010**	0.03
Quads	0.013	0.013	0.002	0.002	0.025	0.024	**0.050**	0.006	**0.005**	**0.005**	0.03

Note. Reported values are the MAE on the validation set (Equation (32)). Significant differences within each experiment group are denoted in bold.

The other was the lognormal parameterization:

$$\log D_{\Delta t}^{(k)} \sim N(m_{D_{\Delta t}}^{(k)}, s_{D_{\Delta t}}^{(k)}). \tag{31}$$

Post-MCMC sampling, we applied our uniform H_0 prior to obtain our final, combined H_0 posterior. Our implementation of the joint-sample MCMC sampling heavily borrows from HIERARC[8] (Birrer et al. 2020), which uses ASTROPY (Astropy Collaboration et al. 2013) to compute cosmological quantities and EMCEE to run the sampler.

3. Results

We organize our results as follows. Before we refer to the H_0 inference results, in Section 3.1, we assess the precision, accuracy, and statistical consistency of the first step in our pipeline: the BNN lens modeling. Having established that the individual BNN-inferred lens model posteriors are reasonable, in Section 3.2, we proceed to interpret the H_0 estimates obtained from individual lenses in the context of the TDLMC metrics. In Section 3.3, we report on the combined H_0 predictions and discuss the potential challenges associated with combining information from hundreds of lenses. Section 4.5 describes the computational efficiency of our pipeline as compared with traditional forward-modeling approaches, with cost projections for possible future applications.

See Table 3 for a summary of our experiments. The first block of experiments, labeled A, tests the sensitivity of our method to the pixel noise level. We retrain the BNN on images rendered with three different exposure times of 2700 s (0.5 HST orbit), 5400 s (1 HST orbit), and 10,800 s (2 HST orbits) for these experiments. The second block B takes either the doubles or the quads from the run with the longest exposure time of 2 HST orbits. There were 89 quads and 111 doubles in our set of 200 test lenses. To control for the sample size, we took all the 89 quads and randomly sampled 89 doubles.

[8] https://github.com/sibirrer/hierArc

3.1. Individual Parameter Recovery

3.1.1. Accuracy

To evaluate the accuracy of BNN parameter recovery, we adopt the median absolute error (MAE) metric, defined as the median of the absolute-valued difference between the predictive mean (Equation (20)) and the true parameter value across all the lenses in the experiment group, i.e.,

$$\text{MAE} \equiv \text{median}\{|\mathbb{E}[\xi^{(k)}|d^{(k)}, \Omega_{\rm int}] - \xi_{\rm true}^{(k)}|\} \tag{32}$$

for each parameter $\xi^{(k)} \in \mathbb{R}$ for lens k. Table 4 lists the MAE values evaluated on the validation set.

Overall, the BNN yields accurate posteriors. In particular, we can retrieve $\gamma_{\rm lens}$ to 3% accuracy. Surprisingly, the accuracy does not seem to vary across the exposure times. We investigate the apparent insensitivity to the exposure time in Section 4.1. The 6–7 mas accuracy in the source position is contextualized further in Section 4.2.

3.1.2. Statistical Consistency

To probe the statistical consistency of BNN-inferred lens model posteriors with the truth values, we use the calibration metric presented in Section 2.3.3. It is this metric that we used to select our final MC dropout rate $p_{\rm drop} = 0.1\%$ as well, from the values 0%, 0.1%, and 0.5%.

The calibration curves for the exposure time of 0.5 HST orbit and dropout probabilities of 0% (no dropout), 0.1%, and 0.5% are shown in Figure 4. The curves for longer exposure times looked qualitatively similar. The no-dropout and 0.1% dropout curves are almost indistinguishable; not modeling the epistemic uncertainty at all (i.e., only performing simple conditional density estimation) does not make the model significantly more confident. The epistemic uncertainties are small, most likely because we have a large training set of 512,000 lenses and a relatively flexible GMM parameterization for the aleatoric

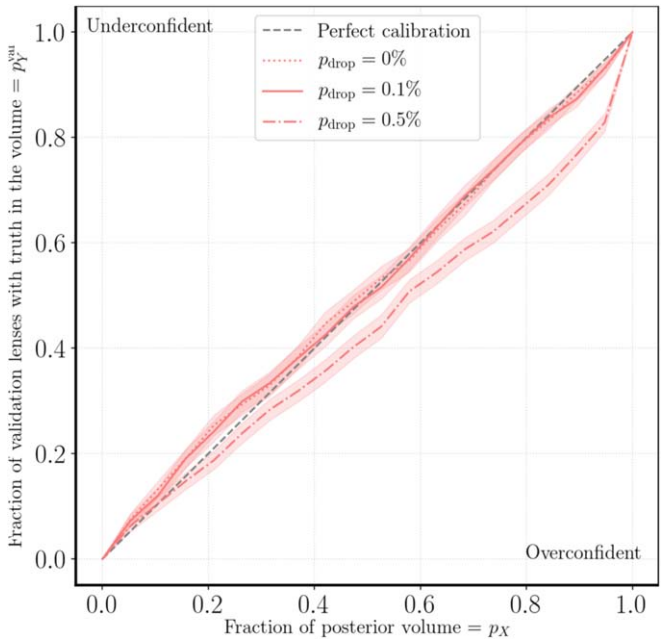

Figure 4. The calibration curve $p_Y^{\rm val}$ vs. p_X for the BNN lens model posterior as defined in Equation (21), for an exposure time of 0.5 HST orbit and dropout probabilities of 0% (no dropout), 0.1%, and 0.5%. There were $N^{\rm val} = 512$ lenses in the validation set. The aleatoric portion of the BNN lens model posterior seems to be capturing most of the uncertainty. Not modeling the epistemic uncertainty at all ($p_{\rm drop} = 0\%$) does not affect the calibration significantly. A slightly higher dropout probability of $p_{\rm drop} = 0.5\%$ leads to overconfidence because the model is underfit. We choose $p_{\rm drop} = 0.1\%$.

uncertainty. As mentioned in Section 2, we opt for $p_{\rm drop} = 0.1\%$ in this paper for all exposure times. We keep the MC dropout parameterization in case the MC dropout captures higher-order epistemic uncertainties that the calibration metric would miss.

If we increase dropout to $p_{\rm drop} = 0.5\%$, the BNN posterior becomes overconfident. This result may be counterintuitive at first glance; a higher dropout probability corresponds to a higher assigned epistemic uncertainty, so we would expect the BNN posterior to become less confident. One possible explanation is the bias-variance trade-off. A higher dropout means more regularization (lower variance), but regularization can hamper optimization and lead to a larger training error (higher bias). The MAE values for the $p_{\rm drop} = 0.5\%$ model were, in fact, higher than those for the $p_{\rm drop} = 0.1\%$ model by 30%. To some extent, this pattern speaks to the limitation of MC dropout as a method of quantifying epistemic uncertainties. It would be worthwhile to explore other methods, such as ensemble-based ones, that are not associated with underfitting.

3.1.3. Precision

Having established that the BNN is accurate and well calibrated, we proceed to report its precision. The parameter uncertainty values in Table 5 are the predictive standard deviation, obtained by taking the standard deviation of the posterior samples for each parameter. Similarly as with the MAEs, we report the median uncertainty over the validation set. More concretely, we defined the parameter uncertainty as

$$\xi^{(i)(k)} \sim p(\cdot|d^\star, \Omega_{\rm int})$$
$$\text{median}_k\{\sqrt{\text{Var}_i\{\xi^{(i)(k)}\}}\}, \quad (33)$$

where $\xi^{(i)(k)} \in \mathbb{R}$ refers to the ith posterior sample of this parameter for each lens k.

Taking the MAE and uncertainty together, the predictions for $\gamma_{\rm lens}$ and $x_{\rm src}$, $y_{\rm src}$ seem to be more accurate and precise for the quads compared to the doubles. For $\gamma_{\rm lens}$, the accuracy is better by 30% and precision by 20%. The source position errors are smaller by 3–5 mas and uncertainty smaller by 2–4 mas. The two extra AGN images in the quads likely offers additional information about the position and orientation of the Einstein ring. Whereas the accuracy on the lens ellipticity parameters e_1, e_2 is similar between the doubles and quads, the BNN is 20% more uncertain about these parameters for the quads on average. The reason for this is not clear and merits further exploration.

3.2. Individual H_0 Recovery

How does the BNN lens modeling performance, validated previously in Section 3.1, translate to H_0 ($D_{\Delta t}$) for individual lenses? The BNN was sufficiently accurate that, at the level of inferring H_0, none of the 200 test lenses were discarded. In this section, we visualize the H_0 posteriors for a few lenses and summarize the per-lens H_0 recovery for the entire set of 200 test lenses.

To guide our interpretation of the individual H_0 posteriors, let us first introduce a useful benchmark. Aside from the BNN-inferred lens model, two more ingredients affect our H_0 inference for a given lens: the time delays and the external convergence. We had assumed small time delay uncertainties and measurement errors (both 0.25 day) so that the relative Fermat potential from the BNN lens modeling would dominate the H_0 uncertainty budget (see Equation (4)). We also assumed a narrow distribution in the environment mass densities that would shift the inferred H_0 at a level of 2.5%. Whereas the effects of time delays and convergence are small on average, it is instructive to completely isolate the effect of BNN lens modeling on the individual lens level. To this end, we define the "time delay precision ceiling," a reference H_0 "posterior" that fixes the lens model posterior at the delta-function truth:

$$\begin{aligned} &p_{\rm precisionceiling}(H_0|\Delta t^{(k)}, d^{(k)}) \\ &\propto p(H_0) \int p(\Delta t^{(k)}|D_{\Delta t}^{(k)}(H_0), \xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}, \kappa_{\rm ext}) \\ &\quad \times \delta(\xi_{\rm lens}^{(k)} - \xi_{\rm lens,true}^{(k)})\delta(\xi_{\rm light}^{(k)} - \xi_{\rm light,true}^{(k)}) \\ &\quad \times p(\kappa_{\rm ext}) \quad d(\xi_{\rm lens}^{(k)}, \xi_{\rm light}^{(k)}) d\kappa_{\rm ext}. \end{aligned} \quad (34)$$

The time delay precision ceiling represents the precision ceiling in the theoretical case of a perfectly known lens model.

The $D_{\Delta t}$ posterior under the time delay precision ceiling is exactly Gaussian, by design. To see this, note that

$$D_{\Delta t} = \frac{c\Delta t_{\rm true}}{\Delta\phi}, \quad (35)$$

$$\Delta t_{\rm true} = \frac{\Delta t_{\rm obs}}{1 - \kappa_{\rm ext}}, \quad (36)$$

where $\Delta\phi$ is the Fermat potential difference between the images under consideration. The likelihood of the observed time delay $\Delta t_{\rm obs}$ is Gaussian and so is the prior on $\frac{1}{1-\kappa_{\rm ext}}$ (Equation (11)), so when the lens model is fixed at the truth, $D_{\Delta t}$ being modeled is simply a convolution of Gaussians.

Any difference between the BNN-inferred posterior and the time delay precision ceiling can be attributed to the BNN lens

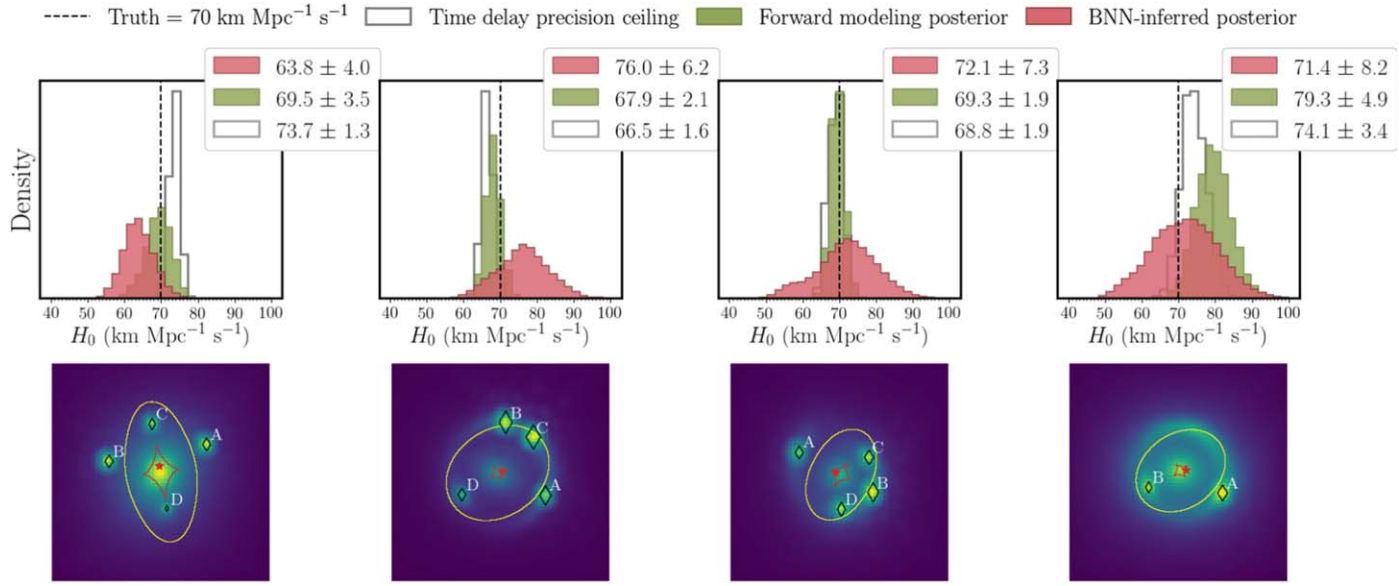

Figure 5. Example lenses from the test set with increasing H_0 uncertainties from left to right. Top: the BNN-inferred H_0 posterior for 2 HST orbits and time delay precision ceiling (defined in Equation (34)) for each lens, with the mean and standard deviation (68% credible interval) of the Gaussian fit (Equation (26)). Note that the BNN-inferred posterior and the time delay precision ceiling share the effect of time delay errors and external convergence, so any difference between them is purely due to BNN lens modeling. Bottom: noiseless images of the lenses, overlaid with the true source position (red star), caustics (red), and critical curves (yellow). AGN image positions are labeled A-D or A-B, with the size of the diamond marker indicating the magnification.

Table 5
Uncertainty Assigned by the BNN on Individual Parameters

Experiment	γ_1	γ_2	$x_{\rm lens}('')$	$y_{\rm lens}('')$	e_1	e_2	$\gamma_{\rm lens}$	$\theta_{\rm E}('')$	$x_{\rm src}('')$	$y_{\rm src}('')$	$R_{\rm src}('')$
0.5 HST orbit	0.020	0.020	0.004	0.004	0.039	0.039	0.080	0.011	0.011	0.012	0.04
1 HST orbit	0.020	0.020	0.005	0.005	0.039	0.040	0.077	0.011	0.012	0.012	0.04
2 HST orbits	0.020	0.020	0.05	0.005	0.039	0.039	0.076	0.011	0.012	0.012	0.04
Doubles	0.020	0.020	0.005	0.005	**0.036**	**0.036**	0.074	0.011	**0.013**	**0.014**	0.04
Quads	0.020	0.020	0.005	0.005	**0.044**	**0.044**	0.078	0.011	**0.011**	**0.010**	0.04

Note. Definition is given in Equation (33). Significant differences within each experiment group are denoted in bold.

modeling. To illustrate this concept on the individual lens level, in Figure 5, we display the BNN-inferred H_0 posterior and time delay precision ceiling of four test lenses along with their images. The H_0 precision of these four lenses is representative of that in the test set as a whole; the set of 200 test lenses had been divided into four bins of increasing H_0 uncertainty and the four lenses sampled randomly from the bins. Our predictions for the rightmost lens, a double, is the least precise partly because, being spherical, the lens has a very small caustic and hence a short relative time delay. On the other hand, the leftmost, most precise lens is a fold quad.

Looking at the joint posteriors over key BNN-predicted parameters and $D_{\Delta t}$ for individual lenses, obtained through the MCMC sampling procedure described in Section 2.4, we can determine whether $D_{\Delta t}$ was sensitive to any particular parameter. Figures 6 and 7 show the posterior over key BNN-predicted parameters and $D_{\Delta t}$ for the leftmost and rightmost lenses in Figure 5. For both lenses, the BNN lens modeling was accurate; the truth falls within the 68% contour of the inferred posterior. The pairwise correlations between $D_{\Delta t}$ and each parameter reveal that, for the lens in Figure 6, $D_{\Delta t}$ was mainly sensitive to $\gamma_{\rm lens}$, lens ellipticity, and $y_{\rm src}$. The lens in Figure 7 was sensitive to $\gamma_{\rm lens}$, e_2, and $x_{\rm src}$.

In addition, we can identify pairwise parameter degeneracies modeled by the BNN. Notice that the BNN effectively captures the $e_1 - \gamma_1$ and $e_2 - \gamma_2$ degeneracies. Because the $R_{\rm src}$ posterior is no better than the implicit prior defined by the training set, however, we do not observe any degeneracy between $\gamma_{\rm lens}$ and $R_{\rm src}$. The BNN does not constrain the external shear parameters γ_1, γ_2 very well beyond the implicit prior, but it assigns sufficiently large uncertainties so as not to bias the $D_{\Delta t}$ inference.

So far, we have looked at the H_0 posterior for one lens at a time. To summarize our method's H_0 retrieval performance on individual lenses for the test set as a whole, we adopt some of the metrics introduced in the TDLMC: precision, accuracy, and goodness (Ding et al. 2018). Their definitions are reproduced here. Precision (P) is the average fractional H_0 uncertainty across the $N_{\rm test}$ test lenses in the experimental group. Letting $\mu^{(k)}$, $\sigma^{(k)}$ denote the assigned mean and uncertainty of the H_0 prediction for lens k from Equation (26),

$$P \equiv \frac{1}{N_{\rm test}} \sum_k \frac{\sigma^{(k)}}{H_0}. \tag{37}$$

Accuracy (A) is the average fractional bias across the lenses. If the true H_0 value is H_0,

$$A \equiv \frac{1}{N_{\rm test}} \sum_k \frac{\mu^{(k)} - H_0}{H_0}. \tag{38}$$

Recall $H_0 = 70\,{\rm km\,s^{-1}\,Mpc^{-1}}$. Finally, goodness ($\chi^2$) is the standard reduced χ^2 that evaluates the goodness of the assigned

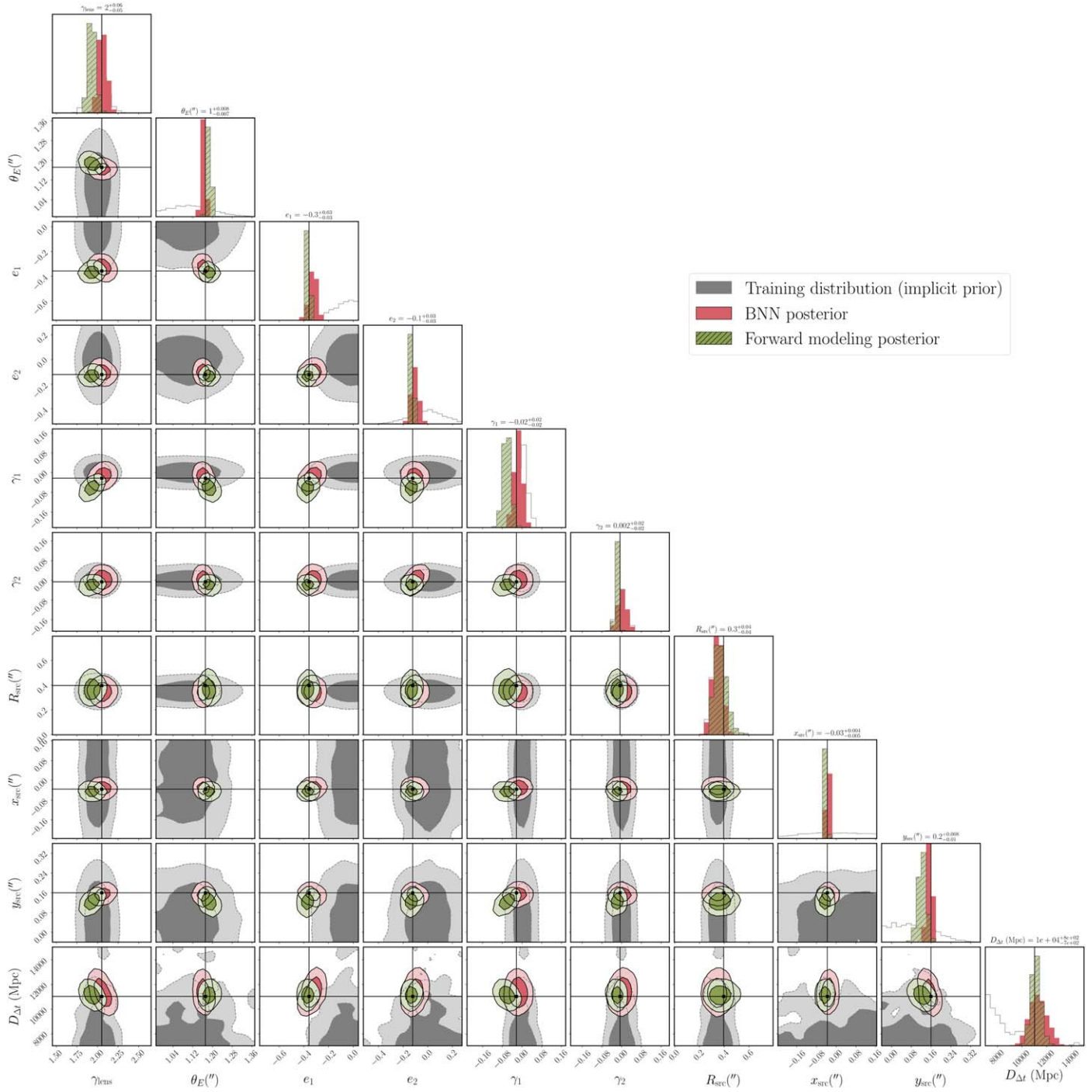

Figure 6. The joint posterior over key BNN-predicted parameters and $D_{\Delta t}$ for the leftmost lens in Figure 5. Contours are the 68% and 95% credible intervals.

uncertainties across the lenses.

$$\chi^2 \equiv \frac{1}{N_{\rm test}} \sum_i \left(\frac{\mu^{(k)} - H_0}{\sigma^{(k)}} \right)^2. \tag{39}$$

The TDLMC metrics for all our experiments in Table 3 are plotted in Figure 8. Overlaid are the target ranges for the metrics, which were determined following Ding et al. (2018). The target for P is based on the best forward-modeling results reported in the literature by the start of the challenge:

$$P < 6\%. \tag{40}$$

Our precision of 9%–10% does not meet this target. This is expected, as classical forward modeling would be more precise than the BNN-based inference, even for our simple model assumptions.

The accuracy target of

$$|A| < 1\% \tag{41}$$

expresses the goal of sub-percent accuracy. All the BNN experiments are well within sub-percent accuracy.

The target range for χ^2 corresponds to the 1 and 99 percentiles of the χ^2 distribution for 200 degrees of freedom

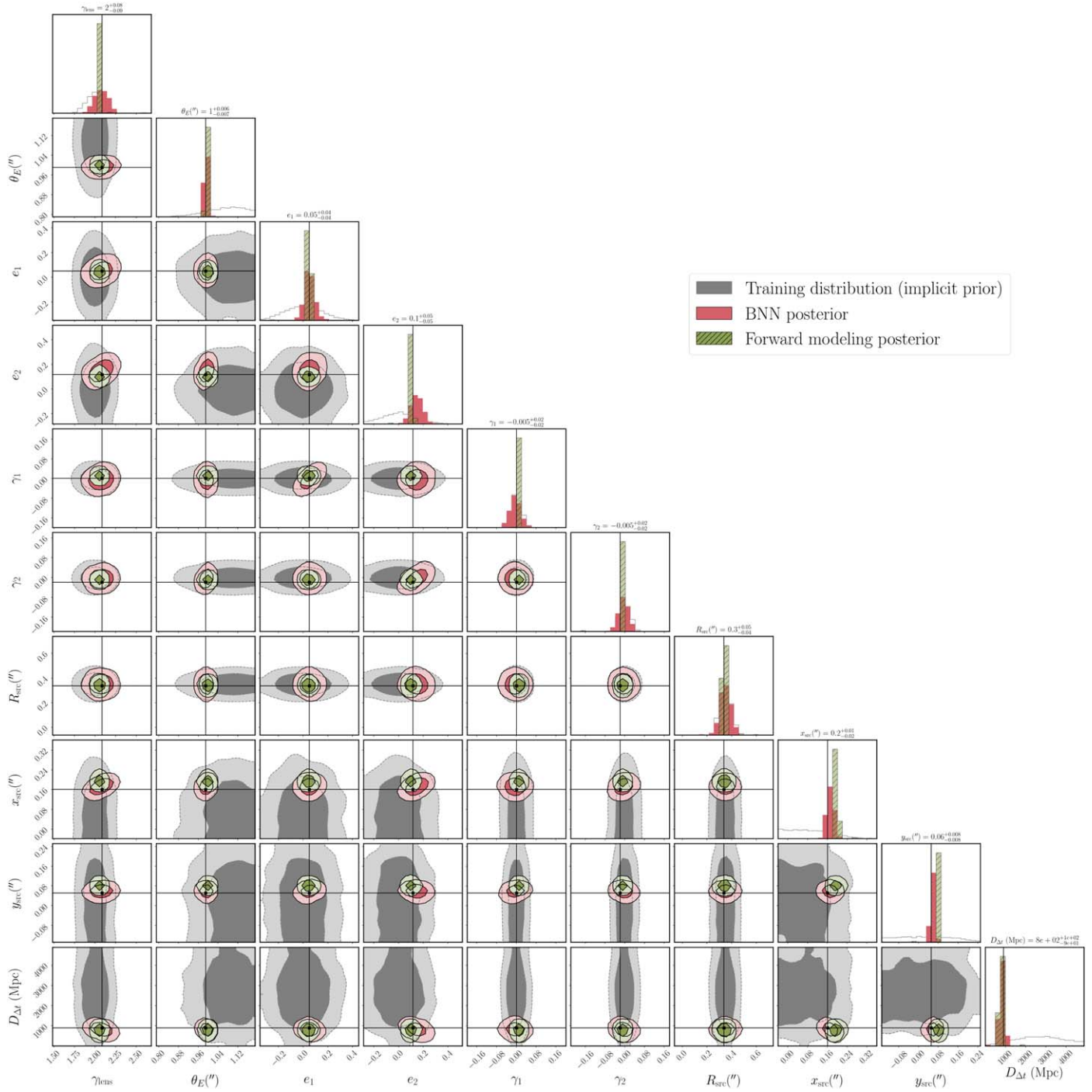

Figure 7. The joint posterior over key BNN-predicted parameters and $D_{\Delta t}$ for the rightmost lens in Figure 5. Contours are the 68% and 95% credible intervals.

(for the 200 lenses in our test set).

$$0.8 < \chi^2 < 1.2. \tag{42}$$

When taking the 89 doubles or quads only, adjusting the degrees of freedom gives

$$0.7 < \chi^2 < 1.4. \tag{43}$$

All except the doubles meet the goodness target, but this does not necessarily imply statistical inconsistency for the doubles, given that the TDLMC metrics weight the lenses equally. In order to account for the varying information content across the lenses, we analyze the combined posteriors in the next section.

3.3. *Combined* H_0 *Recovery*

The true efficacy of our pipeline lies in the joint inference over hundreds of lenses. In Figure 9, we overlay the combined H_0 posterior for the 200 test lenses along with the 200 individual H_0 posteriors, assuming a broad uniform prior in H_0 everywhere. We report a final precision of 0.5 $\mathrm{km\,s^{-1}\,Mpc^{-1}}$ (0.7%). There is no detectable bias; the combined posterior is consistent with the truth. As an additional test of statistical

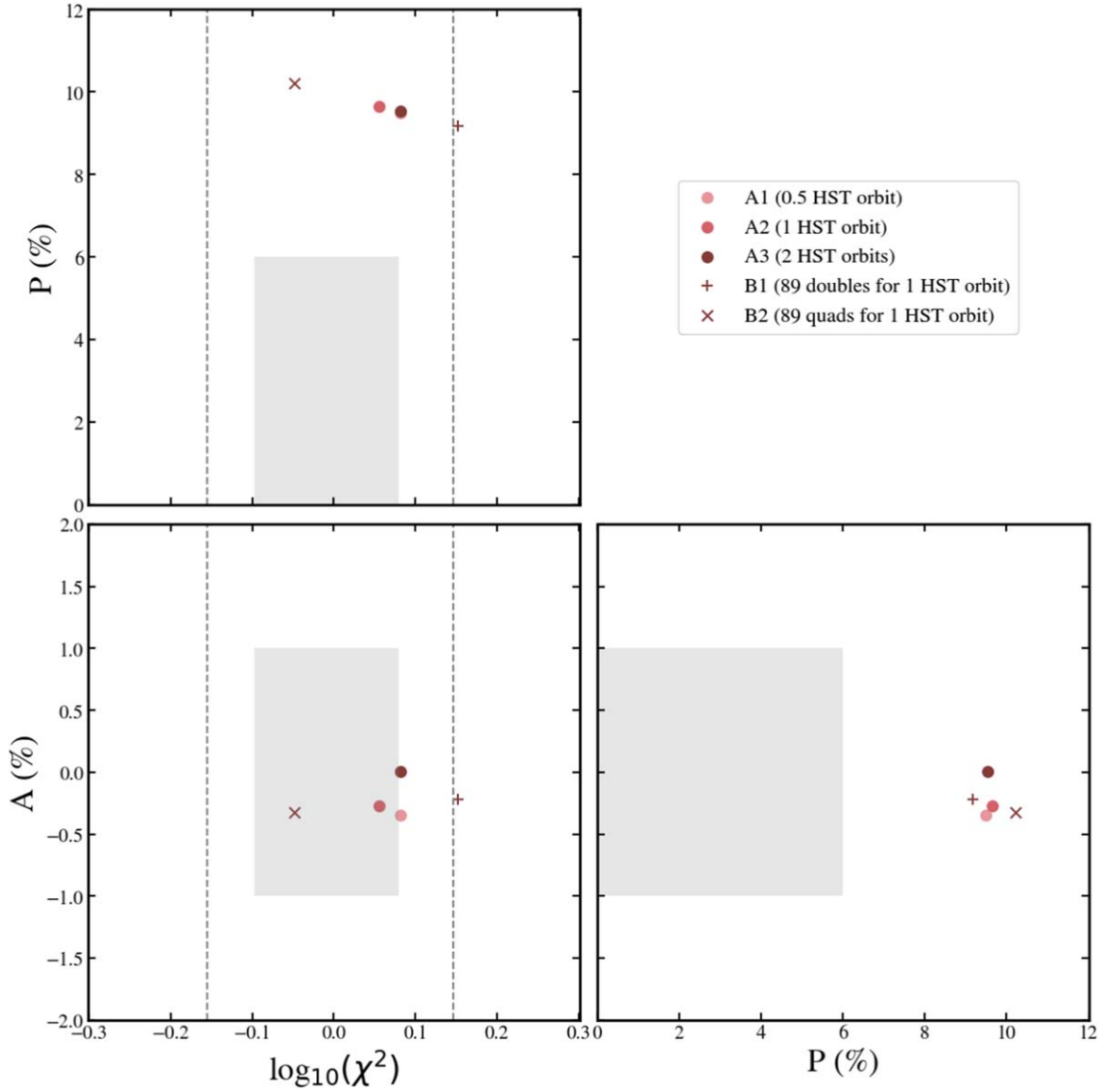

Figure 8. The TDLMC metrics evaluated on the test sample (see Equations (37)–(39) for the definitions). The shaded region corresponds to the target region for a sample of 200 lenses (Equations (40)–(42)). For the doubles and quads, the dotted lines demarcate the goodness (χ^2) target range for the sample size of 89 (Equation (43)). Our precision of 9%–10% lies outside the target of 6% expected for the best-performing forward-modeling approaches. All experiments meet the sub-percent accuracy target. All except the doubles meet the goodness target, but note that the TDLMC metrics weight the lenses equally, so a few outlying lenses could have skewed the metric.

consistency, for each lens, we computed the credible interval at which the truth H_0 lies. The confidence matched the frequency closely; the truth fell within the 68.3% credible interval in 65% of the lenses, within the 95.5% credible interval in 95.5% of the lenses, and within 99.7% credible interval in 99.0% of the lenses.

To assess the impact of the exposure time and image configuration on the joint-sample inference, the combination was done for each of the experiment groups in Table 3. Figure 10 shows the combined estimate for experiments A1–A3, i.e., 200 lenses in the exposure times of 0.5, 1, and 2 HST orbit(s). The precision of the combined posterior does not vary with image depth, as expected from the lack of such a trend in the individual parameter recovery (Section 3.1) and individual H_0 recovery (Section 3.2) with the image depth.

Figure 11 shows the combined H_0 posteriors for the 89 doubles and 89 quads separately (groups B1, B2 in Table 3). For both doubles and quads, the precision is comparable, at 0.7 km s^{-1} Mpc^{-1} (1%), and there is no bias. Again, there is no significant trend with image depth.

Systematics that appear small on an individual level can figure prominently in the combined posterior when the sample size is large. We find that the form of the fit distribution for the $D_{\Delta t}$ posterior merits careful consideration. See Section 4.4 for a discussion of this issue.

4. Discussion

4.1. Pixel-level Information Processed by the BNN

In this section, we explore the patterns in the BNN predictions with the exposure time and various properties of the lens, to confirm that the BNN-inferred posteriors are reasonable. We also compare the BNN lens modeling with traditional forward modeling on a lens-by-lens basis.

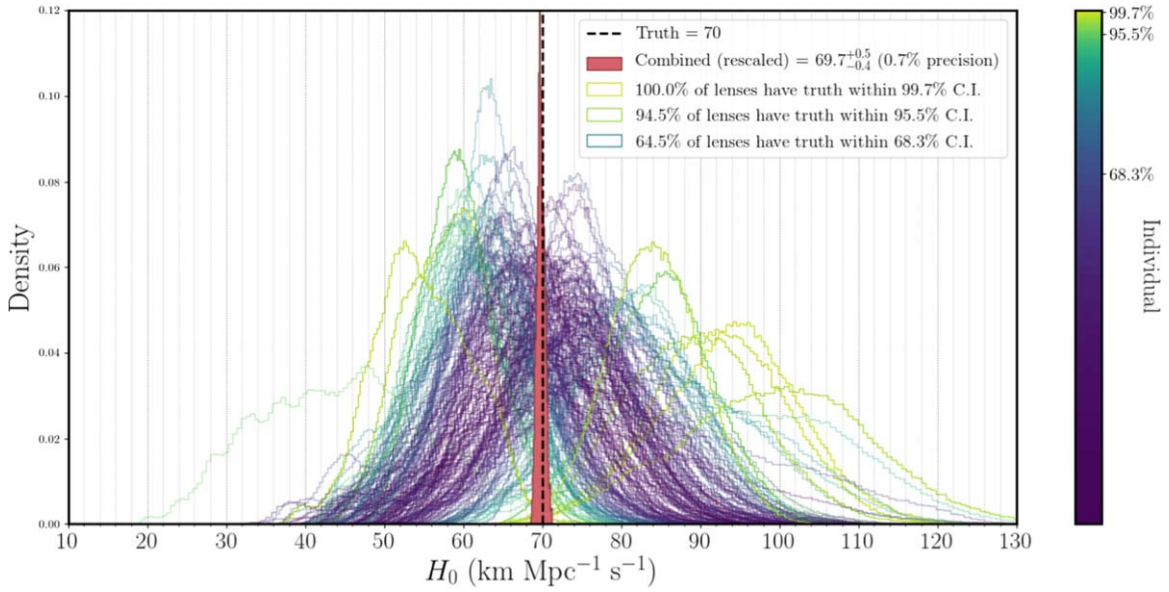

Figure 9. The BNN-inferred H_0 posteriors for 200 lenses were combined to yield an unbiased H_0 estimate of precision 0.5 km s^{-1} Mpc^{-1} (0.7%). We overlaid the 200 individual H_0 posteriors with the joint-sample posterior. The colors of the individual H_0 posteriors represent the credible interval in which the truth H_0 value falls. The colors corresponding to 68.3%, 95.5%, and 99.7% credible intervals are indicated in the legend.

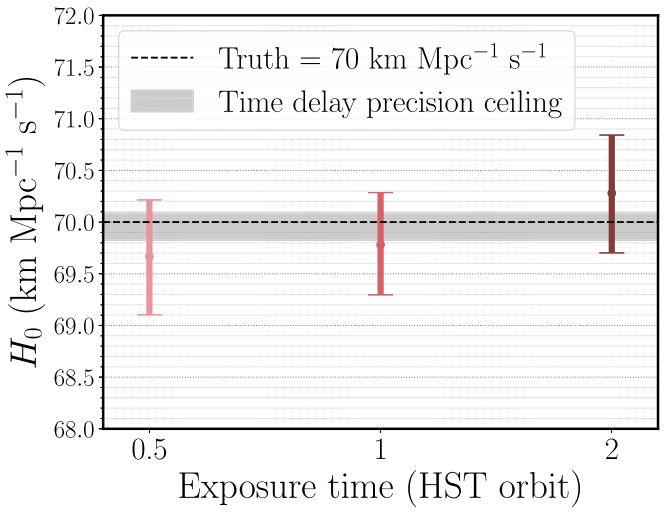

Figure 10. Combined H_0 prediction on a set of 200 test lenses (solid). Error bars represent 68% credible intervals. The final precision is 0.5 km s^{-1} Mpc^{-1} (0.7%) and there is no detectable bias.

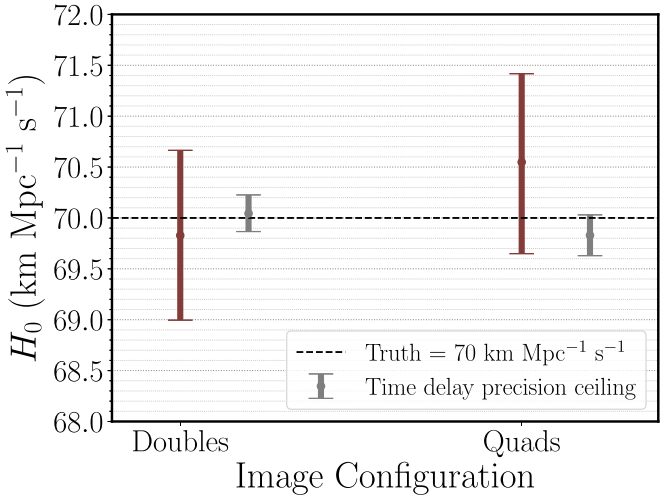

Figure 11. Combined H_0 prediction on 89 doubles and 89 quads (solid). Error bars represent 68% credible intervals. For both doubles and quads, the final precision is around 0.7 km s^{-1} Mpc^{-1} (1%) and there is no detectable bias. Again, the combined posteriors do not vary significantly with image depth.

According to the MAE values in Table 4 and BNN-assigned parameter uncertainty in Table 5, the BNN appears to be insensitive to the exposure time. To investigate this surprising behavior, we take a closer look at the BNN constraints on the $\gamma_{\rm lens}$ parameter, whose information is largely contained in pixels comprising the Einstein ring. In Figure 12, we plot the weighted mean of the absolute bias in $\gamma_{\rm lens}$ binned by the surface brightness of the Einstein ring, where the weights are the inverse of the predictive variances in $\gamma_{\rm lens}$. The surface brightness of the Einstein ring was computed by rendering an image of the source galaxy, without the lens light, and summing up the pixel values. The error bars indicate the weighted standard deviation of the absolute bias. We confirm that the bias is consistent with zero, so that $\gamma_{\rm lens}$ does not bias our H_0 predictions downstream. Yet we do not detect a strong correlation in the center and spread of the $\gamma_{\rm lens}$ bias with the ring brightness, nor with the exposure time.

Does the apparent lack of dependence on the exposure time originate from the data or the BNN? To explore this question, we directly compare the BNN to forward modeling. Because running forward modeling on all 200 test lenses is computationally prohibitive, we examine one lens at a time. In Figures 6 and 7, we overlaid the BNN-inferred posterior with the forward-modeling posterior. Forward modeling achieves tighter constraints on most parameters by a factor of 2–3 compared to the BNN. As shown in Figure 5, we can see that forward modeling can even approach the time delay precision ceiling on H_0 on some lenses, whereas the BNN precision trails behind. The BNN is indeed limited in its ability to extract all the information out of the pixels. The strength of the BNN method lies not in the per-lens precision but in the accuracy and computational efficiency, which enable joint inference over hundreds of lenses.

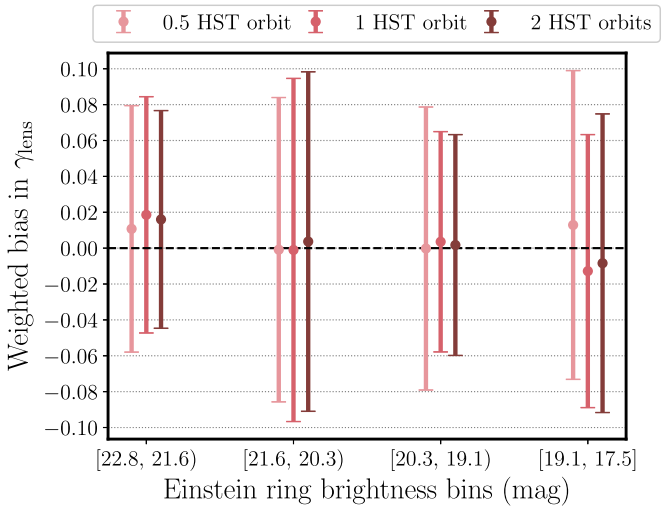

Figure 12. The weighted mean and standard deviation of the absolute bias in $\gamma_{\rm lens}$, binned by the Einstein ring brightness. The bias is consistent with zero. There is no strong pattern in the center and spread of the bias with the ring brightness in any of the exposure times.

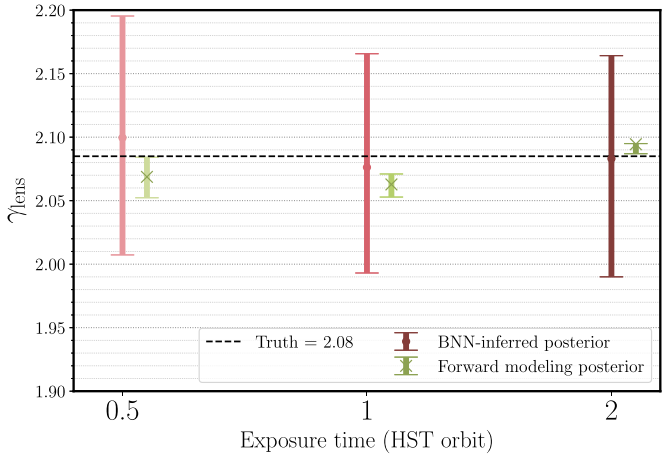

Figure 13. The marginal posteriors on $\gamma_{\rm lens}$ from forward modeling and BNN for the rightmost lens in Figure 5. Forward modeling generates tighter constraints on deeper images, whereas the BNN precision remains similar.

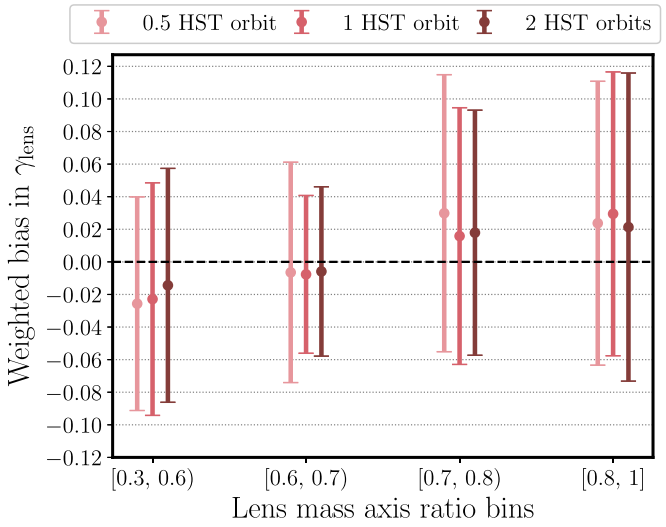

Figure 14. The weighted mean and standard deviation of the absolute bias in $\gamma_{\rm lens}$ binned by the Einstein ring brightness. The spread of the bias clearly increases for more spherical lenses.

There is more information in the pixels than the BNN can harness at each exposure time. In addition, with increasing exposure time, the forward-modeling constraints become tighter, whereas the BNN constraints remain at the same level. Figure 13 overlays the constraints on $\gamma_{\rm lens}$ from the BNN and forward modeling across the exposure times, for a single lens. Forward modeling is able to extract the extra information from deeper images, achieving higher precision by a factor of 6, 8, and 13 compared to the BNN for exposure times of 0.5, 1, and 2 HST orbits, respectively. The inner workings of the BNN's response to image depth requires further investigation. We postpone this to future work.

Given that the pixel intensity values do not significantly affect the BNN, we proceed to establish whether the BNN predictions follow expected trends with more geometric features of the image. In Figure 14, we plot the weighted mean of the absolute bias in $\gamma_{\rm lens}$ binned by the lens axis ratio, where the weights are the inverse of the predictive variances in $\gamma_{\rm lens}$. We see a trend of smaller uncertainty in $\gamma_{\rm lens}$ with more elliptical lenses, most likely because the elongated shape of the critical curves make the relative Fermat potential differences more dramatic. Because doubles are generally more spherical than quads, this trend may partly explain why the parameter constraints on $\gamma_{\rm lens}$ was 30% more accurate and 20% more precise for quads compared to doubles (see Tables 4 and 5). Similarly, in Figure 15, we bin by the image separation and find that the spread of the bias reduces for lenses with bigger separation, as we expected.

4.2. Astrometric Requirements

We can put the BNN's source position predictions in the context of time delay cosmography by propagating astrometric errors through the relative Fermat potential, into the H_0 inference (see Equation (5)). Birrer & Treu (2019) derived the following approximate requirement for the astrometric uncertainty to be subdominant to the time delay uncertainty:

$$\theta_{i,j}\sigma_\beta \lesssim \sigma_{\Delta t_{i,j}}\frac{c}{D_{\Delta t}}, \tag{44}$$

where $\theta_{i,j}$ is the separation between images i and j; $\sigma_{\Delta t_{i,j}}$ the uncertainty in the measurement of the relative time delay, $\Delta t_{i,j}$; and σ_β the astrometric uncertainty in the source plane. For our galaxy-scale lenses, $\theta_{i,j}\sim 1''$. Further assuming that the $z_{\rm lens}\sim 0.5$, $z_{\rm src}\sim 2$, and $\Delta t_{i,j}\sim 4$ days, we can estimate that the astrometric uncertainty will have to be beaten down to 3 mas. The BNN's astrometric uncertainties were 6–7 mas, as estimated from adding in quadrature the median BNN-assigned uncertainties on the $x_{\rm lens}/y_{\rm lens}$ and $x_{\rm src}/y_{\rm src}$—because the BNN was trained to predict $x_{\rm src}/y_{\rm src}$, defined as the source's offset from the lens centroid $x_{\rm lens}/y_{\rm lens}$. For the BNN, the astrometric uncertainties will therefore dominate the uncertainty error budget. Conversely, for $\sigma_\beta\sim 7$ mas to be subdominant, the time delay measurement would have been degraded to $\sigma_{\Delta t_{i,j}}\sim 0.6$ day.

4.3. Impact of the Lens Light in BNN Lens Modeling

By comparing these MAE values with the values reported in Wagner-Carena et al. (2020), where the BNN was trained on

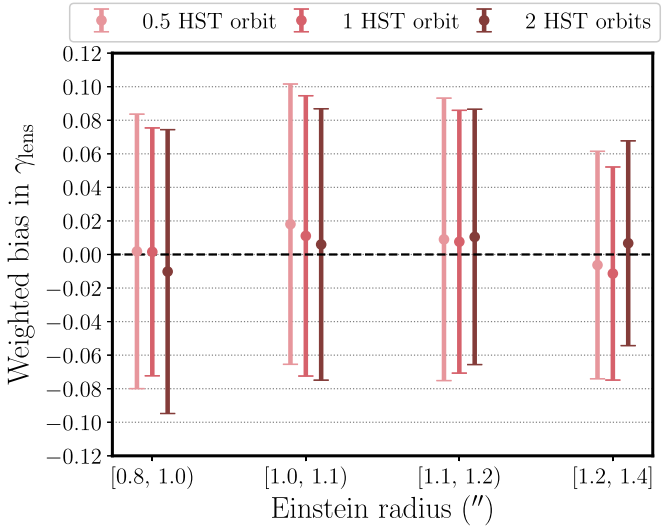

Figure 15. The weighted mean and standard deviation of the absolute bias in $\gamma_{\rm lens}$, binned by the Einstein radius. Bigger separation lenses lead to a small spread in bias, as expected.

lens-subtracted images, we can draw rough conclusions about the impact of including the lens light on parameter recovery. For one, the lens light seems to have aided in the prediction of the lens centroid because we artificially centered the lens light with the lens mass. Our predictions for the $x_{\rm lens}$, $y_{\rm lens}$ were accurate to 1–3 mas, compared to 5–6 mas on the lens-subtracted images for the best-performing BNN model in Wagner-Carena et al. (2020) (Gaussian mixture model, 0.1% dropout). The partial mixing of the Einstein ring with the bright lens light, however, appears to have hurt the prediction accuracy for most other parameters, notably, $\gamma_{\rm lens}$ which had double the MAE. Other parameters saw an MAE increase of 25%–50%. The interpretation that the lens light hurts more than helps agrees with previous experiments with non-Bayesian convolutional neural nets, in which the lens light reduced parameter accuracy by 30%–40% (Pearson et al. 2019). Note, however, that this comparison is approximate at best, as it does not control for the size of the training set (our 512,000 versus the previous 400,000), the width of the training distribution (the previous was almost twice as wide), and the volume of the target parameter space (our 11 dimensions versus the previous 8).

4.4. Choice of Fit Distribution

When the uncertainty in the relative Fermat potential is the precision bottleneck and is approximately Gaussian, it follows from Equation (35) that $D_{\Delta t} \propto 1/\text{Gaussian}$. The PDF for the inverse of a Gaussian random variable does not exist (Robert 1991), but the resulting distribution has a heavy upper tail. We opted to run KDE on the MCMC samples from the $D_{\Delta t}$ posterior (as stated in Equation (31)) so as to explicitly assign weight to this upper tail. Gaussian or lognormal distributions, on the other hand, will not be appropriate. As a simple illustrative example, a Gaussian fit will underestimate $D_{\Delta t}^{(k)}$. See Figure 16 for a visual comparison of lognormal and normal fits on individual $D_{\Delta t}^{(k)}$ samples. The normal fit always lies to the left of the lognormal fit. The difference is qualitatively small for some lenses with a well-constrained Fermat potential, e.g., the leftmost lens, but is consistent.

In Figure 17, we confirm that the Gaussian is an inadequate choice of the fit distribution when the sample size is 200; seemingly small fit errors on individual lenses result in a significant upward bias in the combined H_0. Fitting a lognormal instead brings the combined H_0 posterior to a level more consistent with the truth. The KDE parameterization agrees the best with the truth.

As we look to the prospect of hundreds, maybe thousands, of lens samples, making statements at a 0.1% level will require knowing the shape of the $D_{\Delta t}$ posterior with great accuracy, including the regions toward its tails. We have chosen the KDE with Gaussian kernels for this proof-of-concept study because of its flexibility, but it may be worthwhile to experiment with distributions that can accommodate a positive skew, such as the skew normal distribution (Suyu et al. 2010; Suyu 2012).

4.5. Computational Efficiency

Computational efficiency is an important metric for the inference pipeline we present in this paper because its use case lies in a joint-sample inference over many lenses. The total CPU time of our inference pipeline can be broken down into the data generation time, the BNN training time, and the H_0 inference time. Generating the training and validation sets consisting of 512,000 and 512 lenses, respectively, took a total of 6 hr on 8 CPU cores. Training the BNN with the architecture and training configuration, detailed in the Appendix, took less than 5 hr on a 16 GB NVIDIA Tesla P100 GPU.

The cosmological inference includes evaluating the BNN lens model on the set of test lenses and sampling from the $D_{\Delta t}^{(k)}$ posterior for individual lenses k. The former step can be done within seconds on a CPU or GPU across all the lenses at once. The latter step requires MCMC sampling, which dominates the total inference time. The median sampling time across 4 CPU cores was about 6 minutes per lens. The sampling time for a given lens depends on the shape of the caustic, which determines the stability of the lens equation solver called every iteration to solve for the image positions. Even for a fixed number of iterations, the MCMC sampling time varied greatly, from 3–50 minutes, across the lenses. Interestingly, of the eight lenses for which the MCMC sampling took more than 30 minutes, seven were doubles.

Taken together, the total computational time required to obtain the H_0 posterior from each test lens was 9 minutes. For a set of 200 test lenses, this translates to around 30 hr—which is promising, as complete experiments can be performed on shorter than a 1.25 day development cycle. Note also that the data set generation and training time is a fixed investment that does not vary with the test set size. For 2000 test lenses, for instance, the pipeline would only take 6.3 minutes per lens.

It is useful to compare our computational efficiency with that of the traditional forward-modeling approach. Given the same set of simple model assumptions, `Lenstronomy` takes 0.1 s per likelihood evaluation. It will yield a reasonable but unconverged H_0 posterior in 100,000 MCMC samples, or 3 hr, and will fully converge within 200,000 MCMC samples, or 6 hr. The 9 minute inference time of our method thus represents a speed-up of a factor of 20–40, with the range reflecting the degree of uncertainty in the `Lenstronomy` output.

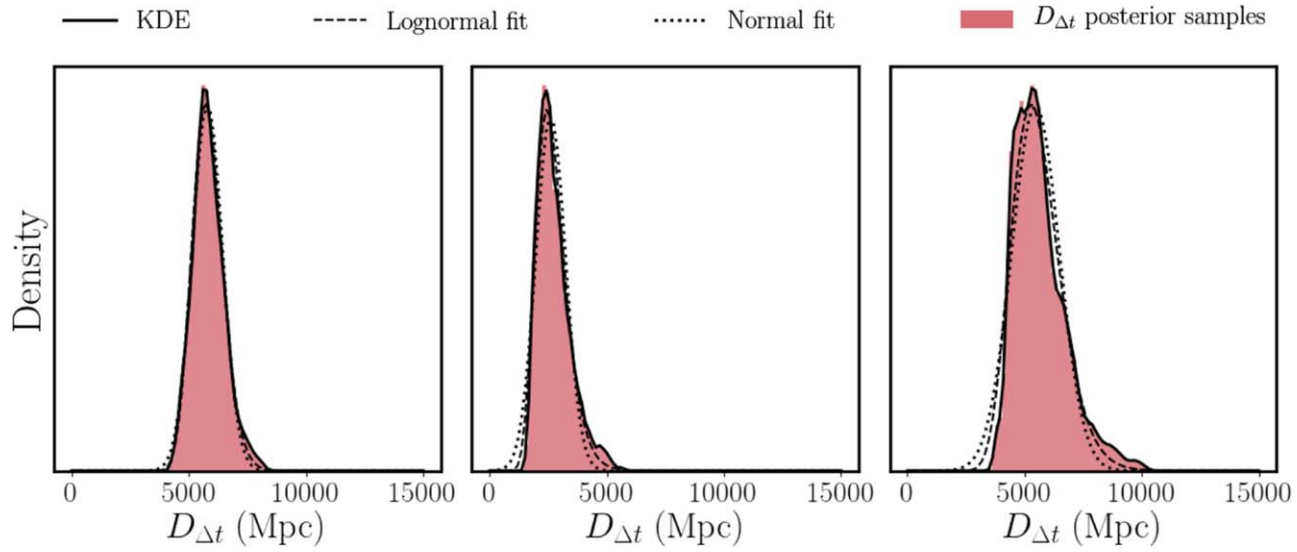

Figure 16. A simple illustration of the importance of the fit distribution. When the dominant source of uncertainty in $D_{\Delta t}^{(k)}$ is the relative Fermat potential, the $D_{\Delta t}^{(k)}$ posterior will take on a heavy upper tail. KDE can capture the tail well. On the other hand, the lognormal (Equation (31)) tail is not heavy enough for some lenses. Normal approximation (Equation (30)) will always underestimate $D_{\Delta t}^{(k)}$, even compared to lognormal.

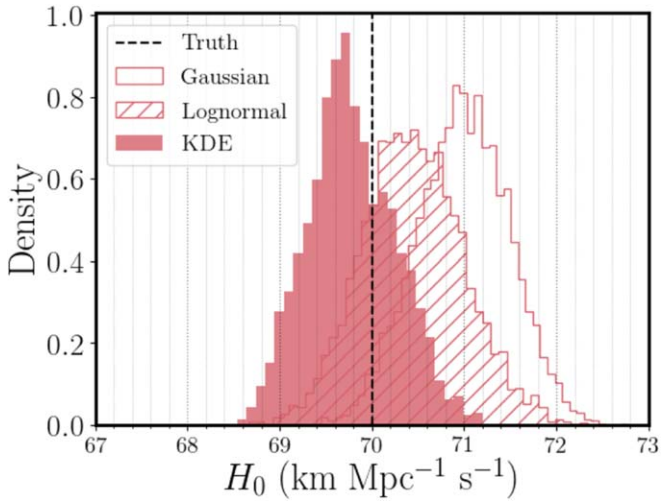

Figure 17. Small fit errors on individual $D_{\Delta t}$ posteriors can be amplified into a substantial bias in the combined H_0 when the sample size is 200. For a simple illustration, we overlaid the combined H_0 from using a Gaussian fit distribution —and observed a significant upward bias. Lognormal does better because it can still capture some of the upper tail in $D_{\Delta t}$. KDE agrees the best with the truth.

4.6. Limitations and Future Directions

Having demonstrated proof of concept, we can now move onto testing the robustness of our approach to systematic errors, particularly those arising from a nonrepresentative training set. We intend to apply the BNN hierarchical inference framework developed in Wagner-Carena et al. (2020) to the problem of inferring H_0 from large samples of lenses and recovering the population prior over the lens model parameters in a follow-up study.

In this method paper, we did not test the response of the BNN to line-of-sight objects and image artifacts like cosmic rays. Such systematics tests must be performed before our method is applied to real data. One way to improve the BNN's robustness is to augment the training data. Hezaveh et al. (2017) added faint cosmic rays, hot pixels, and randomly distributed circular masks to their training images and reported good performance on real HST images.

The BNN lens modeling with lens light in the images is expected to improve with multiband imaging, as demonstrated by experiments with non-Bayesian convolutional neural networks (CNNs; Pearson et al. 2019). Encoding the color difference between the lens and source in the images will alleviate the contaminant effect of the deflector lens light.

Another interesting avenue for exploration is advancing to more realistic sources. For the demonstrated method to be applicable to real systems, we must be able to handle more complex host galaxy profiles, including those of spiral galaxies, and possibly more than one source. The BNN can be put to the test of inferring the posterior over the coefficients of a shapelet decomposition (Birrer et al. 2016).

Similarly, the lensing mass distribution can be made more complex. Whereas we have considered the total density profile in this work, we can probe the detailed structure of the lensing mass by adopting two-component models with explicitly assigned stellar and dark matter halo profiles. Disentangling the stellar and dark contributions would be particularly instructive for galaxy evolution studies. Other natural extensions include additional angular modes beyond elliptical symmetry and multiplane, multi-deflector lensing.

As discussed in Section 2.2.3, we have not addressed the internal and external mass sheets, which are potential sources of bias in H_0. These aspects will need to be investigated using separate data sets. The former can be probed with galaxy kinematics data, and the latter with photometry and spectroscopy of the environment. Our lens modeling pipeline thus requires an accompanying method that can characterize the mass sheets accurately from individual systems, so that biases can be mitigated hierarchically. The method must also be scalable, so as not to severely bottleneck the computation time.

While parameterizing the lens model posterior distribution as a mixture of two Gaussians served our simple profile assumptions, the training time and GPU memory requirement may not scale well to more complex lenses and source profiles, given that the output dimension of the BNN increases

exponentially with the number of model parameters. More complex models may also demand more flexibility in the shape of the posterior. Likelihood-free inference methods such as flow-based generative models are likely to be interesting in that regard.

Lastly, our method can be applied to time delay cosmography with lensed supernovae (SNe) as well. The LSST is expected to discover 3500 lensed SNe over the course of its 10 yr survey (Goldstein et al. 2019). With follow-up spectroscopy and time delay monitoring, the sample of "cosmo-grade" lensed SNe will be significantly increased and will benefit from the efficiency of BNN lens modeling.

5. Conclusions

In this paper, we introduced an automated pipeline for gravitational lens modeling and H_0 estimation that takes as input a high-resolution image, derives an approximate lens model parameter posterior PDF via a BNN, and then propagates the posterior PDFs from multiple lenses into an estimate of the H_0 following the H0LiCOW/TDCOSMO project approach. The computational efficiency of our pipeline enables various sensitivity and robustness checks, from which we draw the following conclusions:

1. BNNs can yield accurate and well-calibrated posterior PDFs over the lens model parameters and the source position required for time delay cosmography.
2. A simple combination of 200 mock test lenses yields a precision of 0.5 $\mathrm{km\,s^{-1}\,Mpc^{-1}}$ (0.7%) and no detectable bias in H_0. For our choice of network architecture and and optimization strategy, the BNN lens modeling and the inferred H_0 are insensitive to varying image depth.
3. Our inference pipeline takes around 9 minutes per lens, including the time taken to generate the training set, train the network, and run the cosmological sampling. It is automated and requires no expert supervision. This represents a 20–40x speed-up compared to the traditional forward-modeling method. The computational efficiency makes the pipeline a promising method to handle large-scale sensitivity tests.

The methodology and software presented in this paper promise to become core infrastructure in time delay cosmography, as the cosmology community prepares to beat down systematics for a large sample of lenses due to be available in a few years' time.

The BNN-based H_0 inference pipeline presented in this paper provides a route to rapid inference of lens model parameters for a large sample of lenses. We have demonstrated that BNN lens modeling can accurately characterize the individual lens posterior PDFs and leads to an unbiased estimate of H_0 on a 200-lens sample, given simple assumptions on the lens model, time delay measurements, and the lens environment. The accuracy and speed make it a promising tool for the exploration of various systematics that may enter the H_0 analysis, where traditional forward-modeling approaches could be slow and intractable. Given the large volumes of data expected from upcoming surveys, our pipeline can play a crucial role in time delay cosmography.

This paper has undergone internal review in the LSST Dark Energy Science Collaboration. We would like to thank the internal reviewers Thomas Collett and Remy Joseph for their insightful comments.

J.W.P. developed the BAOBAB and H0RTON packages, used to perform the analyses in this paper, and wrote the main text. S.W.C. provided input on the BNN training and calibration, helped interpret the results, and contributed to the text. S.W.C. was supported by the KIPAC-Chabolla fellowship and NSF Award DGE-1656518. S.B. advised on motivation, scope, training set generation and analysis, and contributed to the text. J.Y.L. collaborated on the TDLMC submission by contributing code to the early versions of the image simulation and BNN training pipelines. P.J.M. advised on motivation, scope, experimental design, and analysis. A.R. advised on scope and experimental design.

LSST 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. LSST DESC uses the resources of the IN2P3/CNRS Computing Center (CC-IN2P3–Lyon/Villeurbanne—France) funded by the Centre National de la Recherche Scientifique; the Univ. Savoie Mont Blanc—CNRS/IN2P3 MUST computing center; 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.

The authors thank the Google Cloud research credits program for providing the computing resources of the Google Cloud Platform.

This work used the following public software packages: H0RTON (this work), BAOBAB (this work), LENSTRONOMY Birrer & Amara (2018), EMCEE (Foreman-Mackey et al. 2013), CORNER (Foreman-Mackey 2016), ASTROPY (Astropy Collaboration et al. 2013), FASTELL (Barkana 1998), and the standard Python libraries.

Appendix

A.1. BNN Implementation Details

A.1.1. Scalability of the BNN with Increasing Target Parameters

More complex lens profiles will likely be described by more parameters. Yet the size of the output dimension scales exponentially with the number of model parameters, an oft-criticized trait of BNNs. This effect can be mitigated by parameterizing the covariance matrix as positive diagonal elements plus a low-rank (rank r) matrix, in which case the $p_{\mathrm{out}} = 2 \times (2 + r)p + 1$ and the scaling is instead linear. See the `LowRankGaussianNLL` and `DoubleLowRankGaussianNLL` classes in the H0RTON repo for the implementation.

A.1.2. Numerical Stability of the Loss Function

As presented in Section 2.3.1, the ELBO objective is the negative log of the posterior given in Equation (13) plus an L_2 weight regularization term with strength λ. Given our double-Gaussian parameter posterior assumption, this can be written

more concretely as

$$\begin{aligned}\mathcal{L}(W) = &-\log\left(w_1(d;\,W)\right) - \log(1 - w_1(d;\,W)) \\ &+ \frac{1}{2}\log|\Sigma_1(d;\,W)| + \frac{1}{2}\log|\Sigma_2(d;\,W)| \\ &+ (m_1(d;\,W) - \mu_1)^T\Sigma_1(d;\,W)^{-1}(m_1(d;\,W) - \mu_1) \\ &+ (m_2(d;\,W) - \mu_2)^T\Sigma_2(d;\,W)^{-1}(m_2(d;\,W) - \mu_2) \\ &+ \lambda||W||^2, \end{aligned} \tag{45}$$

where m_1, m_2 represent the network-predicted posterior means for the two Gaussians. The dependence of each posterior parameter on W and the input training image d is made explicit here. To ensure that the optimization is numerically stable and well defined, each covariance matrix was parameterized as the log Cholesky decomposition of its inverse (the precision matrix), i.e., for $\Sigma(d;\,W) = \Sigma_{1/2}(d;\,W)$,

$$\begin{aligned}\Sigma^{-1}(d;\,W) &= L(d;\,W)L(d;\,W)^T \\ L(d;\,W) &= \begin{bmatrix} \exp l_1(d;\,W) & 0 & 0 \\ L_{21}(d;\,W) & \exp l_2(d;\,W) & 0 \\ L_{31}(d;\,W) & L_{32}(d;\,W) & \exp l_3(d;\,W) \end{bmatrix}. \end{aligned} \tag{46}$$

Note that the BNN predicts the log of the diagonal entries, so that the diagonal entries are positive. This extra requirement of the log Cholesky parameterization guarantees that Σ is positive definite, whereas the regular Cholesky parameterization only guarantees that Σ is positive semidefinite and can thus lead to a nonunique L. Also, we parameterized w_1 as the half-sigmoid of the BNN-predicted logit ω to get it in the range of $\left(0,\,\frac{1}{2}\right)$.

$$w_1(d;\,W) = \sigma(\omega(d;\,W)) \equiv \frac{1}{1 - \exp(-\omega(d;\,W))}. \tag{47}$$

The source positions $x_{\rm src}$, $y_{\rm src}$ were parameterized in terms of their offsets from the lens positions $x_{\rm lens}$, $y_{\rm lens}$.

A.1.3. Deep Residual Networks with MC Dropout

ResNets, or deep residual networks, address the problems of vanishing/exploding gradients (Bengio et al. 1994) and degradation of training accuracy known to plague deep networks. They do so by inserting so-called "shortcut connections" between the inputs and outputs of a few stacked convolutional layers (He et al. 2015). The idea is that, instead of expecting a set of stacked layers to learn the mapping H between the input x and output $H(x)$, we require it to learn the difference, i.e., the residual mapping $F(x) \equiv H(x) - x$. The original mapping is then recovered as $F(x) + x$. The shortcut connections implement precisely this addition operation. See Figure 3 for the architecture of `ResNet101` as applied to our images. We inserted 1D dropout before every convolution, including the first convolution prior to max pooling. We also had batch normalization and ReLU activation after every convolution. `ResNet101` has 44 million trainable parameters and 347 layers.

The depth and width of the architecture were also tunable hyperparameters. We chose `ResNet101` among ResNet variants with different depths and widths—the other candidates being `ResNet50`, which was shallower but equally wide, and `ResNet56`, which was shallower but much wider—based on the validation set performance.

ORCID iDs

Ji Won Park https://orcid.org/0000-0002-0692-1092
Sebastian Wagner-Carena https://orcid.org/0000-0001-5039-1685
Simon Birrer https://orcid.org/0000-0003-3195-5507
Joshua Yao-Yu Lin https://orcid.org/0000-0003-0680-4838
Aaron Roodman https://orcid.org/0000-0001-5326-3486

References

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Auger, M., Treu, T., Bolton, A., et al. 2009, ApJ, 705, 1099
Barkana, R. 1998, ApJ, 502, 531
Bengio, Y., Simard, P., & Frasconi, P. 1994, IEEE Trans. on Neural Networks and Learning Systems, 5, 157
Birrer, S., & Amara, A. 2018, PDU, 22, 189
Birrer, S., Amara, A., & Refregier, A. 2016, JCAP, 2016, 020
Birrer, S., Shajib, A., Galan, A., et al. 2020, A&A, 643, A165
Birrer, S., & Treu, T. 2019, MNRAS, 489, 2097
Birrer, S., & Treu, T. 2020, arXiv:2008.06157
Blandford, R., & Narayan, R. 1986, ApJ, 310, 568
Blandford, R., Surpi, G., & Kundic, T. 2001, in ASP Conf. Proc. 237, Gravitational Lensing: Recent Progress and Future Goals, ed. T. G. Brainerd & C. S. Kochanek (San Francisco, CA: ASP), 65
Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. 2015, PMLR, 37, 1613
Bolton, A. S., Burles, S., Koopmans, L. V., et al. 2008, ApJ, 682, 964
Collett, T. E. 2015, ApJ, 811, 20
Collett, T. E., & Cunnington, S. D. 2016, MNRAS, 462, 3255
Courbin, F., Chantry, V., Revaz, Y., et al. 2011, A&A, 536, A53
Damianou, A., & Lawrence, N. 2013, PMLR, 31, 207
Denker, J. S., & LeCun, Y. 1991, in Proc. of the 3rd Int. Conf. on Neural Information Processing Systems (San Francisco, CA: Morgan Kaufmann), 853
Ding, X., Treu, T., Shajib, A. J., et al. 2018, arXiv:1801.01506
Dressel, L. 2019, Wide Field Camera 3 Instrument Handbook, Version 11.0 (Baltimore, MD: STScI)
Falco, E., Gorenstein, M., & Shapiro, I. 1985, ApJL, 289, L1
Fassnacht, C., Pearson, T., Readhead, A., et al. 1999, ApJ, 527, 498
Foreman-Mackey, D. 2016, JOSS, 1, 24
Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306
Foreman-Mackey, D., Hogg, D. W., & Morton, T. D. 2014, ApJ, 795, 64
Gal, Y., & Ghahramani, Z. 2016, PMLR, 48, 1050
Giavalisco, M., Sahu, K., & Bohlin, R. C. 2002, STScI Instrument Science Report WFC3-ISR 2002-12
Goldstein, D. A., Nugent, P. E., & Goobar, A. 2019, ApJS, 243, 6
He, K., Zhang, X., Ren, S., & Sun, J. 2015, arXiv:1512.03385
Hezaveh, Y. D., Levasseur, L. P., & Marshall, P. J. 2017, Natur, 548, 555
Kendall, A., & Gal, Y. 2017, arXiv:1703.04977
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Kochanek, C., Morgan, N., Falco, E., et al. 2006, ApJ, 640, 47
Koopmans, L. 2004, arXiv:astro-ph/0412596
Lakshminarayanan, B., Pritzel, A., & Blundell, C. 2017, in Proc. of the 31st Int. Conf. on Neural Information Processing Systems (NIPS17) (Red Hook, NY: Curran Associates), 6405
Levasseur, L. P., Hezaveh, Y. D., & Wechsler, R. H. 2017, ApJL, 850, L7
MacKay, D. J. 1992, PhD Thesis, California Institute of Technology
Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., & Wilson, A. G. 2019, Advances in Neural Information Processing Systems 32 (NeurIPS 2019), 32, 1
Marcel, S., & Rodriguez, Y. 2010, in Proc. XVIII ACM Int. Conf. Multimedia (New York: Association for Computing Machinery), 1485
Oguri, M., & Marshall, P. J. 2010, MNRAS, 405, 2579
Pearson, J., Li, N., & Dye, S. 2019, MNRAS, 488, 991
Refsdal, S. 1964, MNRAS, 128, 307
Ritter, H., Botev, A., & Barber, D. 2018, in Int. Conf. on Learning Representations (ICLR 2018) (Vancouver: ICLR)
Robert, C. 1991, Statistics & Probability Letters, 11, 37
Rusu, C. E., Fassnacht, C. D., Sluse, D., et al. 2017, MNRAS, 467, 4220
Saha, P., & Williams, L. L. 2006, ApJ, 653, 936
Schechter, P. L., Bailyn, C. D., Barr, R., et al. 1997, ApJL, 475, L85
Schneider, P. 1985, A&A, 143, 413

Schneider, P., Ehlers, J., & Falco, E. E. 1992, Gravitational Lenses (Berlin: Springer)
Schneider, P., & Sluse, D. 2013, A&A, 559, A37
Schuldt, S., Suyu, S., Meinhardt, T., et al. 2021, A&A, 646, A126
Scott, D. W. 2015, Multivariate Density Estimation: Theory, Practice, and Visualization (New York: Wiley)
Shajib, A., Birrer, S., Treu, T., et al. 2019, MNRAS, 483, 5649
Shajib, A., Birrer, S., Treu, T., et al. 2020a, MNRAS, 494, 6072
Shajib, A. J., Treu, T., Birrer, S., & Sonnenfeld, A. 2020b, arXiv:2008.11724
Suyu, S. 2012, MNRAS, 426, 868
Suyu, S., Marshall, P., Auger, M., et al. 2010, ApJ, 711, 201
Tihhonova, O., Courbin, F., Harvey, D., et al. 2018, MNRAS, 477, 5657
Treu, T., & Marshall, P. J. 2016, A&ARv, 24, 11
Vanderriest, C., Schneider, J., Herpe, G., et al. 1989, A&A, 215, 1
Verde, L., Treu, T., & Riess, A. G. 2019, NatAs, 3, 891
Wagner-Carena, S., Park, J. W., Birrer, S., et al. 2020, arXiv:2010.13787
Welling, M., & Teh, Y. W. 2011, Proc. of the 28th Int. Conf. on Machine Learning (Madison, WI: Omnipress), 681
Wilson, A. G., & Izmailov, P. 2020, arXiv:2002.08791
Wong, K. C., Suyu, S. H., Chen, G. C.-F., et al. 2019, MNRAS, 498, 1420