The Astrophysical Journal, 909:187 (25pp), 2021 March 10 https://doi.org/10.3847/1538-4357/abdf59

Hierarchical Inference with Bayesian Neural Networks: An Application to Strong Gravitational Lensing

Sebastian Wagner-Carena[1,2], Ji Won Park[1,2], Simon Birrer[1,2], Philip J. Marshall[1,2], Aaron Roodman[1,2], and Risa H. Wechsler[1,2]
(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
Received 2020 November 9; revised 2021 January 20; accepted 2021 January 22; published 2021 March 17

Abstract

In the past few years, approximate Bayesian Neural Networks (BNNs) have demonstrated the ability to produce statistically consistent posteriors on a wide range of inference problems at unprecedented speed and scale. However, any disconnect between training sets and the distribution of real-world objects can introduce bias when BNNs are applied to data. This is a common challenge in astrophysics and cosmology, where the unknown distribution of objects in our universe is often the science goal. In this work, we incorporate BNNs with flexible posterior parameterizations into a hierarchical inference framework that allows for the reconstruction of population hyperparameters and removes the bias introduced by the training distribution. We focus on the challenge of producing posterior PDFs for strong gravitational lens mass model parameters given Hubble Space Telescope–quality single-filter, lens-subtracted, synthetic imaging data. We show that the posterior PDFs are sufficiently accurate (statistically consistent with the truth) across a wide variety of power-law elliptical lens mass distributions. We then apply our approach to test data sets whose lens parameters are drawn from distributions that are drastically different from the training set. We show that our hierarchical inference framework mitigates the bias introduced by an unrepresentative training set's interim prior. Simultaneously, we can precisely reconstruct the population hyperparameters governing our test distributions. Our full pipeline, from training to hierarchical inference on thousands of lenses, can be run in a day. The framework presented here will allow us to efficiently exploit the full constraining power of future ground- and space-based surveys (https://github.com/swagnercarena/ovejero).

Unified Astronomy Thesaurus concepts: Strong gravitational lensing (1643); Cosmology (343); Computational methods (1965); Convolutional neural networks (1938); Hierarchical models (1925)

1. Introduction

As light from a distant source passes by a sufficiently massive foreground lens, multiple rays of light can be refocused onto the same observer in an effect known as strong gravitational lensing. As an astrophysical probe, strong lenses are directly sensitive to the gravitational potential of the lens (or deflector), the large-scale structure along the line of sight, and the metric of the universe. These are the very regimes where some of the most interesting questions about the nature of dark matter and the geometry of our universe can be probed. Over the past three decades, the number of observed strong gravitational lenses has increased by well over an order of magnitude (Blandford & Narayan 1992; Sonnenfeld et al. 2013), to roughly 1000 currently known systems. The next generation of wide-field optical imaging surveys, particularly the Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST)[3] and the surveys carried out by ESA's Euclid mission[4] and NASA's Nancy Grace Roman Space Telescope,[5] will push the number of measured strong lenses into the tens of thousands (Collett 2015). Even imposing stringent sample selection criteria, for example focusing on quadruply imaged quasars with well-measured time delays, will leave us with hundreds of viable lenses to analyze (Oguri & Marshall 2010). The combination of an order-of-magnitude increase in the quantity of the data and a high sensitivity to the salient physics gives strong-lensing science enormous discovery potential.

Perhaps the most pressing application of strong-lensing science today is in constraining the expansion of our universe. The recent tension between early-universe probes like the Planck cosmic microwave background (CMB) measurements (Planck Collaboration 2020) and late-universe probes like the Type Ia supernovae measurements by the SH_0ES team (Riess et al. 2019) has placed increased attention on tests of the Hubble constant (H_0). Further complicating our understanding, alternative late-universe measurements by the Carnegie Chicago Hubble Program (CCHP; Freedman et al. 2019, 2020) do not find the same discrepancy, despite sharing physical uncertainties with the SH_0ES measurement. In the "late" universe, strong gravitational lens time delays offer an essential complementary probe. The rays of each lensed image of the source take different paths with different physical distances, producing a "time delay" between the source images. By connecting this time delay to a "time-delay distance," we can construct a probe that is sensitive to the mass distribution of the lens, the mass distribution along the line of sight, and H_0. Because the physical uncertainties associated with strong-lensing measurements are independent of those in the SH_0ES, Planck, and CCHP data, time-delay cosmography is uniquely suited to help constrain systematics and new physical models (Verde et al. 2019).

[3] https://www.lsst.org
[4] https://sci.esa.int/web/euclid
[5] https://roman.gsfc.nasa.gov/

The work by the H0LiCOW collaboration (Wong et al. 2019) measured a value of the Hubble constant at 2.4% precision from six lenses. Work by Shajib et al. (2018) forecasts that improving these constraints to below the 1% level will require a joint analysis of at least 40 lenses. However, both these analyses utilize independent priors on a lens-by-lens basis and do not consider significant covariance in the systematics of the modeling. More recent work by the TDCOSMO collaboration (Birrer et al. 2020) has shown that relaxing radial mass profile assumptions and conducting hierarchical inferences of the lenses results in an 8% measurement from the sample of seven TDCOSMO time-delay lenses alone. When combined with 23 lenses with kinematics information from the SLACS sample, the uncertainty reduces to 5%. Using a hierarchical approach, Birrer & Treu (2020) forecast that even with drastically relaxed assumptions on the radial mass profile, a sample of 50 time-delay lenses together with 200 non-time-delay lenses can reach 1% level precision constraints on the Hubble constant.[6]

Strong lensing has also been instrumental in developing our understanding of dark matter at both galactic and subgalactic scales. Work on early-type galaxies combining strong-lensing measurements with constraints from stellar kinematics has been able to experimentally verify the presence of dark matter halos, probe deviations from an isothermal mass profile, and quantify the redshift evolution of the mass-to-light ratio (Treu & Koopmans 2004; Koopmans et al. 2006; Bolton et al. 2008). Sonnenfeld et al. (2015) apply a fully hierarchical ensemble analysis to a set of ∼80 lenses from the Strong Lensing Legacy Survey and the Sloan ACS Lens Survey (Bolton et al. 2006). Their work simultaneously models the stellar initial mass function (IMF) and the dark matter halo while accounting for strong-lensing selection effects. The analysis produces constraints on the density and slope of dark matter in the inner halo and demonstrates the value of a joint analysis even on a smaller sample of lenses. As with the time-delay measurements, reproducing this work on thousands of lenses will require new modeling frameworks. A large-scale analysis also holds a great deal of promise for measurements of cosmic shear. Projections with an LSST-sized data set suggest that strong lensing can offer shear constraints competitive with current weak lensing surveys (Birrer et al. 2018).

While these examples of strong-lensing science are only a subset of the work that has been done in the field,[7] they already suggest the need for a new modeling methodology that is capable of producing consistent and accurate predictions on thousands of lenses. One potential tool is a class of models known as Bayesian Neural Networks (BNNs). Unlike conventional neural networks, BNNs seek to go beyond accurate parameter predictions by producing a full posterior of the output parameters that includes modeling uncertainty. Gal & Ghahramani (2016) demonstrate that using Monte Carlo dropout during training and testing yields an approximate BNN[8] that is computationally tractable and more robust than traditional neural networks. Kendall & Gal (2017) extend this work to imaging data and show that BNN-predicted posteriors are statistically sound and precise. Since then, BNNs have been used with success in data problems ranging from semantic image segmentation (Kampffmeyer et al. 2016), to disease detection (Leibig et al. 2017), to active learning (Gal et al. 2017b). For a more detailed review of BNNs in the astrophysical literature, see Charnock et al. (2020).

Within the field of strong gravitational lensing, Perreault Levasseur et al. (2017) have applied BNNs to lenses drawn from a singular isothermal ellipsoid (SIE) profile and produced well-calibrated one-dimensional marginal posteriors of the lens parameters. Their work demonstrates that the BNN approach can return accurate, fully automated predictions several orders-of-magnitude faster than more traditional modeling. More recently, Schuldt et al. (2021) have also successfully applied neural networks to estimate the maximum likelihood parameters of simulated Hyper Suprime-Cam strong lenses. However, there are a few notable limitations that must be addressed before these types of results can be used to model the mass profiles of real lenses. The SIE profile assumption used for both training and testing in Perreault Levasseur et al. (2017) and Schuldt et al. (2021) is equivalent to assuming a power-law elliptical mass distribution (PEMD) with a fixed value of slope $\gamma = 2.0$. Traditionally, the slope is allowed to vary (Wong et al. 2019; Shajib et al. 2020), and its uncertainty is a dominant contribution to the uncertainty in the inference of cosmological and astrophysical parameters like H_0 (Suyu et al. 2013). Additionally, extending the single Gaussian marginal posteriors used in Perreault Levasseur et al. (2017) to a full posterior would be overly simplistic; there are known covariances between the mass profile parameters. Finally, it is not sufficient to evaluate the calibration of our BNN modeling on test examples drawn from the same lens parameter hyperdistributions as our training set. This final point touches on a limitation of BNNs more broadly: the training distribution becomes an interim prior for our BNN's posteriors. Because we cannot train our networks on examples drawn from the same underlying physical distribution that governs the objects in the sky, this interim prior can bias our inference.

BNNs are not the only modeling technique that has been proposed as an alternative to traditional forward modeling on strong lenses. Work by Chianese et al. (2020) has also shown that using Variational Autoencoders for source generation can improve the flexibility of strong-lensing parameter estimation, albeit with only a small improvement in computational time. Brehmer et al. (2019) use simulation-based inference to circumvent an intractable likelihood function and infer the posteriors on population hyperparameters of lensing substructure. Similarly, neural-network-based approaches have been proposed as a method for detecting the presence of individual substructure in strong-lensing images (Diaz Rivero & Dvorkin 2020; Ostdiek et al. 2020a, 2020b). There has also been work toward automated modeling of strong-lensing quads (Shajib et al. 2019), although this approach still takes 50–500 CPU hr and 3 hr of expert time per lens.

In this paper, we are interested in answering the following questions:

1. Can BNN predictions be made robust to the distribution used to generate a test set without retraining the BNN to that specific distribution?
2. Relatedly, can a BNN be used to reconstruct the population hyperparameters that govern the distribution of objects in our universe?

[6] If spatially resolved kinematics can be obtained for the 50 time-delay lenses, modeling the 200 non-time-delay lenses is not required.

[7] For a more in-depth review, see Treu (2010).

[8] For consistency with the literature, we will use the acronym BNN for our approximate BNNs.

3. For the case of strong lensing, are BNNs capable of producing posteriors on PEMD parameters that are statistically consistent with the truth? How do these posteriors compare to those generated by a traditional forward-modeling approach?
4. How flexible do the posteriors predicted by our BNN need to be to perform well on our simulated strong lenses? Under what conditions do the assumptions that go into BNN-based inference begin to break down?
5. Is it possible to construct an inference pipeline using BNNs that can extract accurate constraints from an LSST-sized data set on short (hours or days) timescales?

This work demonstrates that, given reasonable requirements on the training set, all five of these challenges can be robustly addressed. We present a BNN-based modeling framework that is fast, automated, and capable of returning unbiased representative posteriors. We start from a training set with a broad sampling of lenses drawn from PEMDs; our models then build on the work in Perreault Levasseur et al. (2017) by extending the inferred posterior to include potential covariances and bimodalities. Through a set of validation metrics, we find that both a multivariate Gaussian and a mixture of Gaussians are capable of capturing the complexity in the strong-lensing posterior. To our knowledge, our work is the first to demonstrate that BNNs can return statistically consistent full posteriors on strong lenses. We then continue to extend our models by deploying them on test distributions that are statistically distinct from our training distribution (while still employing the same lens model family in both the training and test sets). While a simplistic application of our models returns a biased parameter inference, we develop and test a hierarchical framework that removes this bias. Our final ensemble approach not only returns accurate constraints on the true underlying distribution of our sky, but is capable of returning corrected posteriors on subsecond timescales. The BNN approach requires less expert intervention than traditional forward modeling, and this several orders-of-magnitude improvement in computational time allows for faster model iteration and drastically reduces computational costs.

The paper is organized as follows. In Section 2, we offer a summary of the approximate BNN framework and a detailed description of our hierarchical modeling framework. Section 3 includes a description of our BNN implementation and the pipeline we used to generate our simulated lenses. We then train a set of models with different posterior parameterizations and modeling priors to compare their performance in Section 4.1. To probe the bias induced by our training distribution, in Section 4.2 we introduce a group of "true sky" test sets and apply our hierarchical framework to infer population hyperparameters. Finally, in Section 5, we discuss possible further directions and the implications of our results for future BNN modeling in the strong-lensing literature.

As part of this publication, we are releasing our strong-lensing analysis package OVEJERO. The package includes all of the code and dependencies necessary to reproduce the results in this paper along with a set of comprehensive JUPYTER notebooks that are meant to help familiarize users with the code. The source code, documentation, demo notebooks, and plotting code used for the graphics in this publication can be found at https://github.com/swagnercarena/ovejero.

2. Hierarchical Bayesian Computations with BNNs

2.1. Approximate BNNs

BNNs offer a framework to generate parameter posteriors that incorporate both the uncertainty inherent to the data (aleatoric uncertainty) and the uncertainty in the modeling (epistemic uncertainty). Written in more concrete terms, BNNs seek to predict the posterior on one object's parameters $\xi^\star$ given its corresponding data $d^\star$ and the training set of parameter-image pairs $\{\Xi, D\}$:

$$p(\xi^\star|d^\star, \Xi, D) = \int p(\xi^\star|d^\star, W)p(W|\Xi, D)\, dW. \quad (1)$$

Here, W are the weights of the BNN. For our work, $d^\star$ will be the strong-lens image and $\xi^\star$ will be the PEMD parameters. However, the derivations in this section are general to any set of objects with an underlying hyperdistribution. The aleatoric uncertainty is captured in the distribution $p(\xi^\star|d^\star, W)$ and expresses the fact that even if we knew the weights for our model perfectly, there would still be a limit to the constraining power from one image. The epistemic uncertainty is represented by the distribution $p(W|\Xi, D)$ and captures the fact that, without infinite training data, there will exist some uncertainty on the functional form of our network. In this framework, the aleatoric uncertainty is parameterized as a function of the BNN outputs. For example, in the case of a mixture of Gaussians, this would be

$$p(\xi^\star|d^\star, W) = \sum_i n_i(d^\star, W)\mathcal{N}(\xi|\mu_i(d^\star, W), \Sigma_i(d^\star, W)), \quad (2)$$

where the variables n_i, μ_i, and Σ_i are written as functions of the input image $d^\star$ and the model weights W because they are the final outputs of our BNN. Here, $0 \leqslant n_i \leqslant 1$ is the weight of the ith Gaussian, μ_i is the mean of the ith Gaussian, and Σ_i is the covariance of the ith Gaussian. The second term in Equation (1) is intractable. Gal & Ghahramani (2016) suggest approximating $p(W|\Xi, D)$ using a variational distribution $q(W|\Omega_{\rm int})$, where $\Omega_{\rm int}$ is the interim distribution from which the training data $\{\Xi, D\}$ is drawn. The introduction of this approximation is what distinguishes the Gal & Ghahramani (2016) approximate BNN from a true BNN. Note that we introduce a conditional on $\Omega_{\rm int}$ to emphasize that the results of our training are dependent on the training set distribution; this will be an important consideration when we discuss our hierarchical formalism in Section 2.2. Gal & Ghahramani (2016) propose the distribution:

$$q(W|\Omega_{\rm int}) = \prod_k q(W_k|\Omega_{\rm int}) \quad (3)$$

$$= \prod_k (1 - p_k)\mathcal{N}(M_k, \sigma^2 I) + p_k\mathcal{N}(0, \sigma^2 I), \quad (4)$$

where k indexes the layers of our BNN. Both the matrix M_k and the constant p_k are free parameters of our variational distribution. M_k is equivalent to the weights of layer k of a traditional neural network, and p_k is equivalent to the dropout probability of a layer. This specific variational formulation is appealing because it is computationally easy to sample from; for $\sigma \to 0$, it is equivalent to applying a Bernoulli mask on the

weights M_k, an operation known as dropout. The parameters in the variational distribution are then optimized by minimizing the Kullback–Leibler (KL) divergence between $q(W|\Omega_{\rm int})$ and $p(W|\Xi, D)$. This is equivalent to minimizing the negative log Evidence Lower Bound (ELBO) loss $\mathcal{L}$:

$$\mathcal{L} = -\int q(W|\Omega_{\rm int}) \log p(\Xi|D, W) dW + {\rm KL}(q(W|\Omega_{\rm int})||p(W)) \quad (5)$$

$$= -\prod_{j\in\{\Xi,D\}} \int q(W|\Omega_{\rm int}) \log p(\xi_j|d_j, W) dW + {\rm KL}(q(W|\Omega_{\rm int})||p(W)), \quad (6)$$

where $p(W)$ is a prior on our weights and $\prod_{j\in\{\Xi,D\}}$ is a product over the examples in our training set. The first term can be minimized in an unbiased manner by sampling over $q(W|\Omega_{\rm int})$ and then updating M_i using stochastic gradient descent. The remaining KL term cannot be analytically evaluated, but Gal & Ghahramani (2016) show that, in the case of a discrete quantized Gaussian prior on each W_i with mean 0 and covariance $\sigma_W \mathbb{I}$, it can be approximated by

$${\rm KL}(q(W_i|\Omega_{\rm int})||p(W_i)) \propto -\frac{l^2(1-p_i)}{2N}||M_i||^2, \quad (7)$$

where $l \propto \frac{1}{\sigma_W}$ is the length scale of the weight prior and N is the number of parameter-image pairs in the training set. With this simplification, we can then also take gradients over the second KL term in Equation (6) and optimize our weights.

Running a BNN is therefore a two-step process. During the training phase, we minimize the loss in Equation (6) to fit a variational distribution $q(W|\Omega_{\rm int})$ that matches $p(W|\Xi, D)$. Then, conducting inference on a new example is just a matter of sampling from $q(W|\Omega_{\rm int})$:

$$p(\xi^\star|d^\star, \Xi, D) = \int p(\xi^\star|d^\star, W) q(W|\Omega_{\rm int}) dW. \quad (8)$$

$$p(\xi^\star|d^\star, \Omega_{\rm int}) \approx \frac{1}{N} \sum_{w\sim q(W|\Omega_{\rm int})} p(\xi^\star|d^\star, w), \quad (9)$$

where in the last line we have replaced $p(\xi^\star|d^\star, \Xi, D)$ with $p(\xi^\star|d^\star, \Omega_{\rm int})$ to highlight the component of the conditioning on $\{\Xi, D\}$ that will become important in our hierarchical modeling.

2.2. Hierarchical Inference

While no explicit prior term is imposed on the distribution of the lens parameters during the BNN's training, the distribution used to generate the training set becomes an implicit, "interim" prior, $\Omega_{\rm int}$, in the model's inference. Any discrepancy between this interim prior and the true distribution generating the lenses in the sky can become a source of bias. Even if our model is perfectly calibrated to the training distribution, rather than giving $p(\{\xi\}|\{d\})$, it will output $p(\{\xi\}|\{d\}, \Omega_{\rm int})$, where $\{\xi\}$ is the set of inferred parameters and $\{d\}$ is the set of images in our true sky or test set. As we demonstrate in our experiments in Section 4.2, realistic distributions for a test set can lead to substantial bias. However, given sufficient lenses, knowledge of the interim prior $\Omega_{\rm int}$, and the ability to sample from $p(\{\xi\}|\{d\}, \Omega_{\rm int})$, a hierarchical inference framework can be used to extract an unbiased sampling of the lens parameters. In the process, we also reconstruct the hyperparameters that define the test set lens distribution Ω. We start by considering the probability of a specific test set distribution given the set of test images $\{d\}$, for which the full derivation can be found in Appendix C:

$$p(\Omega|\{d\}) = \underbrace{p(\Omega)}_{\Omega\ \text{prior}} \times \underbrace{\prod_k \frac{p(d_k|\Omega_{\rm int})}{p(\{d\})}}_{\text{normalizing factor}} \times \underbrace{\prod_k \frac{1}{N_{\rm imp}} \sum_{\xi_k\sim p(\xi_k|d_k,\Omega_{\rm int})} \frac{p(\xi_k|\Omega)}{p(\xi_k|\Omega_{\rm int})}}_{\text{MCMC with re-weighting}}. \quad (10)$$

Here, k is an index over all lenses in the test set, ξ_k and d_k represent the parameters and data of each individual lens, respectively, and $N_{\rm imp}$ is the number of samples drawn from $p(\xi_k|d_k, \Omega_{\rm int})$. Our BNN allows us to efficiently sample from $p(\xi_k|d_k, \Omega_{\rm int})$, so the third term in Equation (10) can be calculated given an analytic expression for $p(\xi_k|\Omega)$—the likelihood of a lens parameter ξ_k for a fixed test distribution Ω. The type of reweighting being done in Equation (10) is also known as importance sampling. There are a few properties worth noting about this equation. The MCMC reweighting term provides the needed division by the interim prior; this operation is ill defined in regions where $p(\xi_k|\Omega_{\rm int})$ is zero. This limits our model to inferring distributions contained within the interim prior $\Omega_{\rm int}$, motivating our choice of broad training priors. This same $\frac{1}{p(\xi_k|\Omega_{\rm int})}$ term will also assign a large weight to examples that are underrepresented by our training distribution; this will allow our hierarchical model to remove bias introduced by offsets between our interim prior and the test (or true sky) distribution. Finally, because the MCMC reweighting term is a sum rather than a product over BNN samples, distributions Ω that assign little to no probabilistic weight to a sample are not excluded. This will become important when we wish to infer a distribution Ω that is narrower than the uncertainty of our BNN. For previous examples of the use of importance sampling in the literature, see Foreman-Mackey et al. (2014) or Hogg et al. (2010). As mentioned in both of these works, Equation (10) is not guaranteed to be an unbiased estimator in the limit of finite samples. In this work, we have been conservative in our number of BNN samples and checked for the convergence of our hierarchical results.

The next step is to calculate the unbiased posterior of a single lens given the full data set, $p(\xi_k|\{d\})$. The full derivation can be found in Appendix C; here we quote the final result:

$$p(\xi_k|\{d\}) \propto p(\xi_k|d_k, \Omega_{\rm int}) \frac{1}{N} \sum_{\Omega\sim p(\Omega|\{d\})} \frac{p(\xi_k|\Omega)}{p(\xi_k|\Omega_{\rm int})}. \quad (11)$$

N is the number of samples being drawn from $p(\Omega|\{d\})$. As with Equation (10), Equation (11) uses a reweighting term equivalent to importance sampling, although now the summation is over the space of possible test distributions. Note that given the distribution $p(\Omega|\{d\})$, Equation (11) depends only on the lens k. Therefore, calculating $p(\xi_k|\{d\})$ can be broken up into two parts:

Table 1
Configuration of the BNN Used for All of the Results Presented in This Paper

Layer Type	Input Shape	Output Shape
2D Convolutional	($N_{\rm batch}$, 64,64,1)	($N_{\rm batch}$, 30,30,64)
Max Pooling	($N_{\rm batch}$, 30,30,64)	($N_{\rm batch}$, 15,15,64)
2D Convolutional	($N_{\rm batch}$, 15,15,64)	($N_{\rm batch}$, 15,15,192)
Max Pooling	($N_{\rm batch}$, 15,15,192)	($N_{\rm batch}$, 8,8,192)
2D Convolutional	($N_{\rm batch}$, 8,8,192)	($N_{\rm batch}$, 8,8,384)
2D Convolutional	($N_{\rm batch}$, 8,8,384)	($N_{\rm batch}$, 8,8,384)
2D Convolutional	($N_{\rm batch}$, 8,8,384)	($N_{\rm batch}$, 8,8,256)
Max Pooling	($N_{\rm batch}$, 8,8,256)	($N_{\rm batch}$, 4,4,256)
Reshape	($N_{\rm batch}$, 4,4,256)	($N_{\rm batch}$, 4096)
Fully Connected	($N_{\rm batch}$, 4096)	($N_{\rm batch}$, 4096)
Fully Connected	($N_{\rm batch}$, 4096)	($N_{\rm batch}$, 4096)
Fully Connected	($N_{\rm batch}$, 4096)	($N_{\rm batch}$, $N_{\rm outputs}$)

Note. The configuration we use is a modification of the ALEXNET model (Krizhevsky et al. 2012). All convolutional and fully connected layers have dropout performed on their input with a single dropout rate $p_{\rm drop}$ for the entire network. The dropout rate used in our models is discussed in Section 3.3. The number of outputs depends on what parameterization of the posterior is used. The optimal batch size is dependent on the memory available to the GPU.

1. Calculate $p(\Omega|\{d\})$ using Equation (10). This only needs to be done once per data set.
2. Draw samples from $p(\Omega|\{d\})$ to reweight the posterior $p(\xi_k|d_k, \Omega_{\rm int})$ given by our BNN using Equation (11).

With this, we have the numerical framework necessary to turn our BNN samples $p(\{\xi\}|\{d\}, \Omega_{\rm int})$ into samples from the training independent distribution $p(\{\xi\}|\{d\})$.

3. Methods

3.1. BNN Implementation

All of the BNNs we present in this work are modifications of the original ALEXNET model (Krizhevsky et al. 2012) implemented using the TENSORFLOW (Abadi et al. 2016) module in Python. The exact network architecture is outlined in Table 1. In line with the BNN approach introduced in Gal & Ghahramani (2015, 2016), the input to each convolutional and fully connected layer is first passed through a dropout layer. All of the dropout layers in our model share a single dropout rate $p_{\rm drop}$, and dropout is applied both at training and test time. It is also possible to formulate the BNN to have a trainable dropout parameter (Gal et al. 2017a); however, our models with trainable dropout suffered from issues with training convergence that led to extremely poor performance. We have therefore excluded them from our analysis.

The shape of the final output of our model is dependent on the aleatoric posterior parameterization. We use three parameterizations in this work: a diagonal Gaussian (16 degrees of freedom), a full covariance Gaussian (44 degrees of freedom), and a mixture of two Gaussians (89 degrees of freedom). The precise breakdown of the output parameters is described in Table 2. Note that we do not directly map our BNN outputs, which are allowed to range from $(-\infty, \infty)$, to the free parameters of our posterior. Instead, for each posterior, we have selected a mapping that is bijective—every possible BNN output maps to a valid configuration of the posterior, and each possible configuration of the posterior is mapped to by one, and only one, BNN output. A bijective mapping between the BNN outputs and the free parameters of each posterior stabilizes learning and prevents the BNN from proposing invalid posteriors (i.e., a covariance matrix that is not positive semidefinite) that will break training and inference. At the same time, because these mappings are complicated and nonlinear, the Gaussian prior imposed on the weight matrices can have complicated implications for the free parameter configurations that are favored. However, as we will demonstrate through our calibration metric in Section 4, we find that even with these nonlinear mappings, the posteriors returned by our BNNs are statistically sound. It is also worth noting that some recent work has added a normalizing flow to the final layer of a BNN to give the posterior more flexibility (Hortúa et al. 2020a; Hortúa et al. 2020b). While it is not yet clear how to incorporate this normalizing flow into the Bayesian framework, this is an interesting potential avenue for future work.

3.2. Simulated Data Set

The simulated data set consists of 400,000 PEMD lenses with external shear. The PEMD profile (Kormann et al. 1994; Barkana 1998) is given by

$$\kappa(x, y) = \frac{3 - \gamma_{\rm lens}}{2}\left(\frac{\theta_E}{\sqrt{q_{\rm lens}x^2 + y^2/q_{\rm lens}}}\right)^{\gamma_{\rm lens}-1}, \quad (12)$$

where $q_{\rm lens}$ is the axis ratio of the lens, θ_E is the Einstein radius, and $\gamma_{\rm lens}$ is the logarithmic slope. In this profile definition, x and y are defined in a coordinate system aligned with the major and minor axes of the lens. This requires three additional parameters, a rotation angle $\phi_{\rm lens}$ and the lens center position $(x_{\rm lens}, y_{\rm lens})$. The external shear is characterized by an orientation angle $\phi_{\rm ext}$ and modulus $\gamma_{\rm ext}$. Using an angle to parameterize the profile creates a cyclic parameter. In order to avoid dealing with the complications introduced by a cyclic boundary condition, we will often work in the eccentricity/Cartesian coordinates for our ellipticity/shear:

$$e_1 = \frac{1 - q_{\rm lens}}{1 + q_{\rm lens}}\cos(2\phi_{\rm lens}) \quad (13)$$

$$e_2 = \frac{1 - q_{\rm lens}}{1 + q_{\rm lens}}\sin(2\phi_{\rm lens}) \quad (14)$$

$$\gamma_1 = \gamma_{\rm ext}\cos(2\phi_{\rm ext}) \quad (15)$$

$$\gamma_2 = \gamma_{\rm ext}\sin(2\phi_{\rm ext}). \quad (16)$$

Each image contains the lensed light from a source with a Sérsic light profile. This profile is defined by

$$I_{\rm S}(x, y) = I_{\rm eff}\exp\left[-b\left(\left[\frac{\sqrt{x^2 + y^2/q_{\rm s}^2}}{R_{\rm eff}}\right]^{\frac{1}{n_{\rm src}}} - 1\right)\right], \quad (17)$$

where $R_{\rm eff}$ is the effective half-light radius, $I_{\rm eff}$ is the amplitude at $R_{\rm eff}$, n_s is the Sérsic index, and $q_{\rm s}$ is the source axis ratio. The parameter b is set such that half of the luminosity is contained within $R_{\rm eff}$. Here (x, y) are defined on the coordinate system set by the source's major and minor axis. This gives us our final three parameters: the rotation angle $\phi_{\rm s}$ and the source center coordinates $(x_{\rm src}, y_{\rm src})$. Note that we do not draw values of $I_{\rm eff}$ but rather draw the source magnitude $m_{\rm src}$ and set $I_{\rm eff}$

Table 2
Mapping Between BNN Outputs and Degrees of Freedom of Posterior Parameterizations

Posterior	N_{Outputs}	Mapping Details
Diagonal Gaussian	16	8 outputs mapped to the mean of the Gaussian 8 outputs mapped to the log of the diagonal entries of the covariance matrix
Full Gaussian	44	8 outputs mapped to the mean of the Gaussian 8 outputs mapped to the log of the diagonal entries of the lower triangular matrix 28 outputs mapped to off-diagonal entries of the lower triangular matrix Note: the lower triangular matrix specifies the precision matrix of the Gaussian using a log-Cholesky parameterization
GMM	89	16 outputs mapped to the means of two Gaussian (8 each) 16 outputs mapped to the log of the diagonal entries of the lower triangular matrices (8 each) 56 outputs mapped to off-diagonal entries of the lower triangular matrices (28 each) 1 output mapped to the weight on the first Gaussian, w_1, by $w_1 = 1 + \sigma(\text{output})/2$, where σ is the sigmoid function. Note: lower triangular matrix used as before. The weight on the second Gaussian, w_2, is specified by $w_2 = 1 - w_1$.

Note. All of the mapping selected here are bijective between the BNN outputs and the space of possible posterior configurations. This means both that each unique BNN output maps to a unique posterior configuration and that all possible posterior configurations are mapped to.

accordingly. There is no light attributed to the lens galaxy (i.e., the lens is perfectly subtracted from the image).

Our images are simulated to match the quality of the Hubble Space Telescope (HST) Wide Field Camera 3 (WFC3) IR channel with the F160W filter. We use AB magnitudes with a zero point of 25.9463. Our point-spread function (PSF) is set to the drizzled PSF map used by Rung 1 of the Time Delay Lens Modeling Challenge (Ding et al. 2018), which itself was designed to model the WFC3/F160W PSF. We assume the images are 64×64 postage stamps with a pixel size of $0\farcs08$. The distributions for each of the lens and source parameters can be found in Table 3. The large-scale training and test set generation was done using the package BAOBAB[9] (Park et al. 2020), which extends the lens modeling package LENSTRONOMY[10] (Birrer & Amara 2018).

Noise for our images was added on the fly during training in order to augment the quality of our data set. We utilized the noise functionality of the BAOBAB package. The sky brightness was set to 22 mag arcsec^{-2} based on the estimates from Giavalisco et al. (2002). We selected an exposure time of 5400 s to correspond to one HST orbit. Using the mean reported instrument statistics for WFC3/F160W (Dressel 2019), we set the CCD gain to 2.5 e^-/ADU and the read noise to $4e^-$. The BAOBAB configuration files to recreate our training data set can be found in the OVEJERO module files.

In addition to our training set, we also generated a 512 image validation set and a 512 image test set. Together, these two sets are used to select between BNN hyperparameters and evaluate the performance of our network. The validation set and test set were both drawn from the same interim prior described in Table 3 and with the same detector properties as the training set.

Table 3
The Interim Prior Ω_{int}

Component	Distribution
Lens: PEMD	
x-coordinate lens center ($''$)	$x_{\text{lens}} \sim \mathcal{N}(\mu\text{: } 0, \sigma\text{: } 0.102)$
y-coordinate lens center ($''$)	$y_{\text{lens}} \sim \mathcal{N}(\mu\text{: } 0, \sigma\text{: } 0.102)$
Einstein radius ($''$)	$\theta_E \sim \mathcal{N}_{\log}(\mu\text{: } 0.0, \sigma\text{: } 0.1)$
Power-law slope	$\gamma_{\text{lens}} \sim \mathcal{N}_{\log}(\mu\text{: } 0.7, \sigma\text{: } 0.1)$
x-direction ellipticity eccentricity	$e_1 \sim \mathcal{N}(\mu\text{: } 0, \sigma\text{: } 0.2)$
xy-direction ellipticity eccentricity	$e_2 \sim \mathcal{N}(\mu\text{: } 0, \sigma\text{: } 0.2)$
External shear	
Shear modulus	$\gamma_{\text{ext}} \sim \mathcal{N}_{\log}(\mu\text{: } -2.73, \sigma\text{: } 1.05)$
Orientation angle	$\phi_{\text{ext}} \sim \text{Unif}(-\frac{\pi}{2}, \frac{\pi}{2})$
Source: elliptical Sérsic light	
Source magnitude	$m_{\text{src}} \sim \text{Unif}(-25, -22)$
Half-light radius ($''$)	$R_{\text{eff, src}} \sim N_{\log}(\mu\text{: } -0.7, \sigma\text{: } 0.4)$
Sérsic index	$n_{\text{src}} \sim N_{\log}(\mu\text{: } 0.7, \sigma\text{: } 0.4)$
x-coordinate src center ($''$)	$x_{\text{src}} \sim \mathcal{N}_{\text{gen}}(\mu\text{: } 0.0, \alpha\text{: } 0.4, \beta\text{: } 10.0)$
y-coordinate src center ($''$)	$y_{\text{src}} \sim \mathcal{N}_{\text{gen}}(\mu\text{: } 0.0, \alpha\text{: } 0.4, \beta\text{: } 10.0)$
x-direction ellipticity eccentricity	$e_1 \sim \mathcal{N}(\mu\text{: } 0, \sigma\text{: } 0.2)$
xy-direction ellipticity eccentricity	$e_2 \sim \mathcal{N}(\mu\text{: } 0, \sigma\text{: } 0.2)$

Note. $\mathcal{N}$ is the normal distribution, $\mathcal{N}_{\log}$ is the log-normal distribution, and $\mathcal{N}_{\text{gen}}$ is the generalized normal distribution. The mean of a log-normal distribution is set by $\exp(\mu + \frac{\sigma^2}{2})$, meaning that the mean value of the Einstein radius is $1\farcs01$, the power-law slope is 2.02, and the shear modulus is 0.11. For a discussion of parameter definitions, see Section 3.2. All of the distributions used here are intentionally much broader than our expectations from empirical evidence.

3.3. Training Procedure

All the models presented in this work were trained on an NVIDIA Tesla P100 GPU for 400 epochs (i.e., 400 passes over the training data) with a batch size of 512. The TENSORFLOW implementation of the ADAM optimizer was used with the learning rate set to 1×10^{-5} and the ADAM parameters kept at their default values of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-7}$. While the training was allowed to continue for the full 400 epochs for all models, only the model weights that achieved the lowest validation loss during the training run were kept. After 400 epochs, the validation loss on all nine models presented in this work had plateaued. The total training time per model was around 16–24 hr.

In order to improve the stability of training, two types of normalizations were conducted on the training data. Each input image was normalized to have a standard deviation of 1, and the target lens parameters were normalized such that they had mean 0 and standard deviation 1 over the training set. The

[9] https://github.com/jiwoncpark/baobab
[10] https://github.com/sibirrer/lenstronomy

constants used for this normalization were saved so that the normalization could be undone for the purposes of inference. Additionally, to help extend the robustness of the training set, a new noise realization was drawn on the fly each time an image was passed to the BNN.

4. Results

In this section, we present the full results of our combined BNN and hierarchical inference methodology. First, we compare the performance of different modeling choices on the validation and test sets, and show that both a multivariate Gaussian posterior and a Gaussian mixture model posterior with a 0.1% dropout rate produce the best combination of calibration, precision, and agreement with traditional forward modeling (Section 4.1). We then introduce a new set of test sets whose distributions mimic important biases and covariances we expect to find in an LSST-sized data set (Section 4.2). Using these test sets, we demonstrate the ability of our hierarchical inference framework to extend our BNN models beyond the training distribution and recover the population hyperparameters, with special attention given to those of the power-law slope $\gamma_{\rm lens}$.

4.1. Training and Validation

As discussed in Section 3.2, our BNNs are trained on 400,000 synthetic images of PEMD model lenses with external shear drawn from the distributions specified in Table 3. Here we will explore the effects of changing the form of our predicted posterior (which controls the aleatoric uncertainty) and tuning the dropout rate used (which controls the epistemic uncertainty). We focus on three posterior parameterizations:

1. Diagonal Gaussian (Diagonal): the predicted posterior is a multidimensional Gaussian with a diagonal covariance matrix. This resembles the choice made by Perreault Levasseur et al. (2017).
2. Full Gaussian (Full): the predicted posterior is a multidimensional Gaussian with all off-diagonal elements of the covariance matrix included. For details on how the covariance matrix is parameterized, see Section 3.1
3. Gaussian Mixture Model (GMM): the predicted posterior is a weighted sum of two multidimensional Gaussians with a full covariance matrix.

For each of the different posterior parameterizations, we present three dropout rates for a total of nine models. For the GMM and full posterior models, we also present the results with no dropout. The dropout rates presented here were experimentally chosen to span a range of calibrations—from overconfident to underconfident. Along with the dropout rate and the posterior parameterization, the BNN formalism allows for freedom in the length scale used to set the prior on the model weights. From our own tests, we found the inference results to be fairly insensitive to the value of the length scale. We have therefore used one fixed length scale value for all the models in this paper, which is equivalent to keeping the weight prior fixed. The precise parameters for each model are given in Table 4.

To assess the relative quality of our models, we conduct three tests:

Table 4
The Parameters for Each BNN

	BNN Parameters	
Model	Dropout Rate $p_{\rm drop}$	Length Scale l
Diagonal 5%	0.05	1
Diagonal 10%	0.1	1
Diagonal 30%	0.3	1
Full 0%	0	1
Full 0.1%	0.001	1
Full 0.5%	0.005	1
Full 1%	0.01	1
GMM 0%	0	1
GMM 0.1%	0.001	1
GMM 0.5%	0.005	1
GMM 1%	0.01	1

Note. In the training loss, the length scale l appears in a weight regularization term $\lambda_W||W||^2$ with $\lambda_W = \frac{l^2 p_{\rm drop}}{2N}$ (see Section 2.1 for details).

1. The calibration of the model—if a posterior contour contains $x\%$ of the probability volume, the truth should fall within that volume $x\%$ of the time (Section 4.1.1).
2. The median absolute error (MAE) between the posterior mean and the true value (Section 4.1.2).
3. A spot check comparison to results from the forward-modeling approach (Section 4.1.3).

All of the evaluations presented in this subsection are done on the validation set.

4.1.1. Model Calibration

Our main concern with our BNN models is that the posteriors be well calibrated. A well-calibrated posterior is representative of the truth—$x\%$ of the probability volume contains the true value $x\%$ of the time. Our performance on this calibration metric is our principal cut. A model that is accurate in its mean but proportionally overconfident in its uncertainties cannot be used for scientific constraints; if the truth falls well outside the predicted posterior, we would expect catastrophic errors in inference. We will demonstrate the validity of this intuition when we attempt to reconstruct the population hyperparameters of a test set in Section 4.2.

However, measuring the calibration of a BNN in a high-dimensional posterior space is a nontrivial task. While we have a parameterized form for the aleatoric uncertainty, the epistemic uncertainty can only be sampled from. Therefore, we cannot use a calibration metric that requires evaluating the cumulative distribution function of our posterior. Another issue is that, while we can sample from the predicted posterior as much as we would like, we only have one sample from the true posterior—the "true" parameter value used to generate the image. It is not statistically meaningful to ask how well our posterior represents a single point; therefore, we must use a metric that can be averaged over all the lenses in our training set.

One option, as was done by Perreault Levasseur et al. (2017), is to simplify the calibration problem to the one-dimensional marginal posteriors. If we then approximate the

one-dimensional posteriors as Gaussians, we can define 68%, 95%, and 99.7% of the probability volume as being one, two, and three standard deviations from the mean. However, this approach is insensitive to higher-order statistics like parameter covariances. Instead, we employ a calibration metric on the full posterior that builds on the one-dimensional approach. For each lens i, we take $N_{\rm samps}$ samples from its posterior. We then define a distance metric for a posterior sample $\boldsymbol{x}$ as

$$d(x)_i = (\boldsymbol{x} - \mu_i) \cdot \Sigma_{\rm data} \cdot (\boldsymbol{x} - \mu_i)^T, \tag{18}$$

where μ_i is the mean of lens i's posterior and $\Sigma_{\rm data}$ is the empirical covariance matrix for the 400,000 training set samples. This distance metric defines concentric ellipses, and we can get the probability volume contained within an ellipse associated to distance $d_{\rm elip}$ by probing the number of posterior samples with $d(\boldsymbol{x})_i < d_{\rm elip}$:

$$\Delta V_{\rm prob} = \frac{N_{d<d_{\rm elip}}}{N_{\rm samps}}. \tag{19}$$

This gives us the first piece of our calibration metric: a region with $x\%$ of the probability volume. The second piece comes from calculating the same distance metric on the true sample value. We then know, for a specific lens i, how much probability value we need before we have encompassed the true value. By averaging over all the lenses in the validation set, we can test if $x\%$ of the probability volume contains the true value $x\%$ of the time.

The distance metric we use in Equation (18) is not unique because there are infinitely many choices of volume that contain $x\%$ of the probability mass for any posterior. A model can even fulfill our metric of calibration without proposing a posterior that is identical to the true posterior.[11] Unfortunately, this is a limitation of only having one sample from the true posterior. However, our calibration metric is particularly sensitive to both multimodal distributions and covariances; this makes it a good choice for the posteriors we expect based on the forward modeling of individual lenses.

The quantile–quantile plot results for our 11 models on the validation set can be found in Figure 1. For each of our three aleatoric parameterizations, we show a comparison between the three dropout rates explored in this work. For the full and GMM parameterizations, we also show the no-dropout case. The simplest of the three aleatoric models—the diagonal posterior—requires a large amount of dropout before it begins to return results that perform well on our calibration metric. At 30% dropout, the model becomes underconfident in the inner regions of the posterior, but only just returns a good calibration for the outer 20% of the probability volume. Underconfidence like that exhibited by the 30% dropout model can lead to poorer constraints, but the overconfidence shown by the 10% and 5% dropout models is much more concerning. For example, 97% of the probability volume for the 10% dropout model contains just over 80% of the true values. If we were to use the 10% model for inference, we could expect to be catastrophically biased on nearly a fifth of our lenses. For that reason, only the 30% dropout model passes the initial calibration cut.

The full posterior model requires nearly no dropout and achieves a better calibration than the diagonal posterior model. The 1% dropout and 0.5% dropout models are slightly underconfident, while the 0.1% dropout model returns a nearly perfect calibration. For the full posterior model, we also plot the no-dropout case and find that it returns performance in line with 0.1% dropout. To better understand why the full posterior model prefers small to no dropout, we can compare the median aleatoric and total[12] covariance predicted by the diagonal and full posterior models. In Figure 2, we present the comparison, narrowing our discussion to the 0.1% dropout full posterior model and the 30% dropout diagonal posterior model. The median aleatoric uncertainties are fairly mundane: the diagonal posterior has a diagonal aleatoric covariance matrix, while the full posterior has a covariance matrix with meaningful correlations between ellipticity and shear. What is surprising is that the median total covariance matrix, which includes the epistemic uncertainty, has essentially the same form for both models. The weight marginalization being learned by the diagonal posterior model appears to be capturing the same covariances that are explicitly parameterized in the full posterior model. The fact that the diagonal posterior needs such large dropout rates seems to be a direct consequence of the fact that the aleatoric parameterization being used is not sufficiently flexible. The real total uncertainty of the model is fixed, and because our choice of posterior has imposed an artificial constraint on what the aleatoric uncertainty can account for, the epistemic uncertainty has to fill the gap.

This conclusion is further supported by the quantile–quantile plots of the GMM posterior model. Much like the full posterior model, it appears to prefer little to no dropout. Surprisingly, the performance of the full and GMM models is almost identical on our calibration metric. Figure 1 also includes a comparison of all three models with their "optimal" dropout value on the test set. The full and GMM posteriors clearly outperform the diagonal posterior, although all three posteriors avoid catastrophic overconfidence in their predictions. Up to the limited sensitivity of our calibration metric, the full and GMM posteriors with 0.1% dropout appear to have near-perfect calibration.

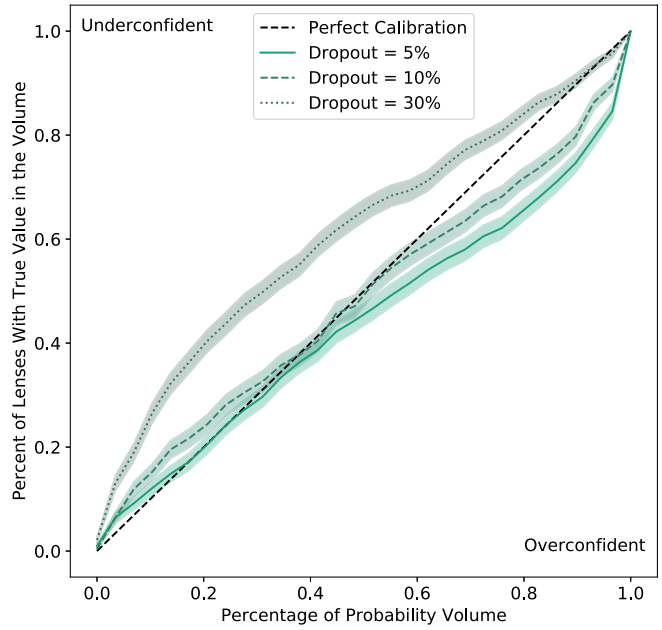
4.1.2. Prediction Accuracy

The cuts we impose in Section 4.1.1 narrow down our models to those that are well calibrated, but they do not tell us how constraining the posteriors are. A well-calibrated model is not necessarily informative. For example, if the BNN returned the interim prior for every lens in our validation set, we would report a perfect calibration; by construction, $x\%$ of the true values fall within $x\%$ of the interim prior's probability volume. As a measure of the information content of our posteriors, we use the MAE between the mean value of the posterior samples and the true value for each lens. The MAE per parameter for each one of our eleven models can be found in Table 5. For each posterior type (diagonal, full, and GMM), we bold the row that corresponds to the well-calibrated model.

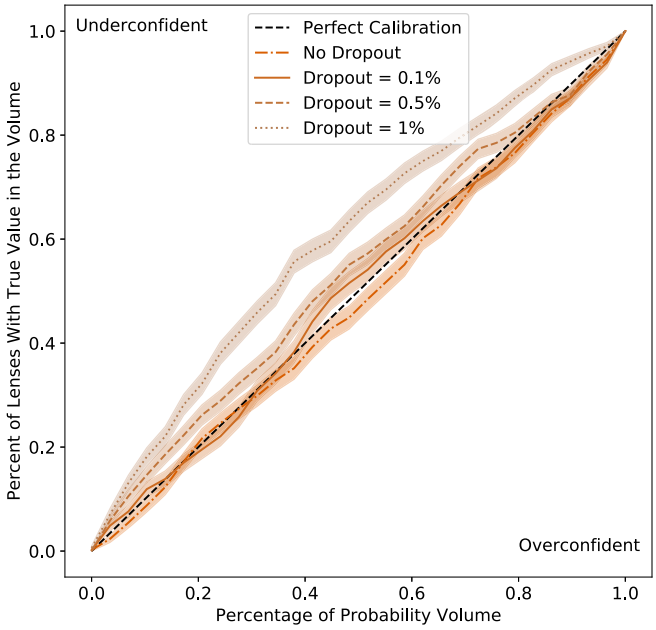
With the exclusion of the no-dropout models, all of the full and GMM posterior models have lower MAE values than the diagonal models. Within the diagonal models, increasing the dropout to achieve better calibration appears to also increase the MAE in the parameter predictions. The opposite trend seemed to hold for the full and GMM models: going from 1% to 0.1% dropout appears to slightly increase the MAE, but the

[11] See Appendix A for a more detailed discussion of this issue.

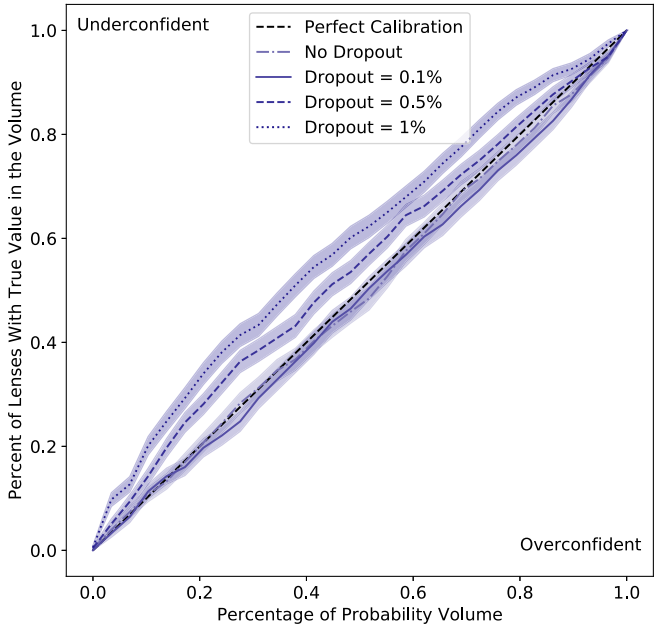
[12] Because the epistemic uncertainty does not have a parameterized form, the total covariance must be empirically measured by drawing samples.



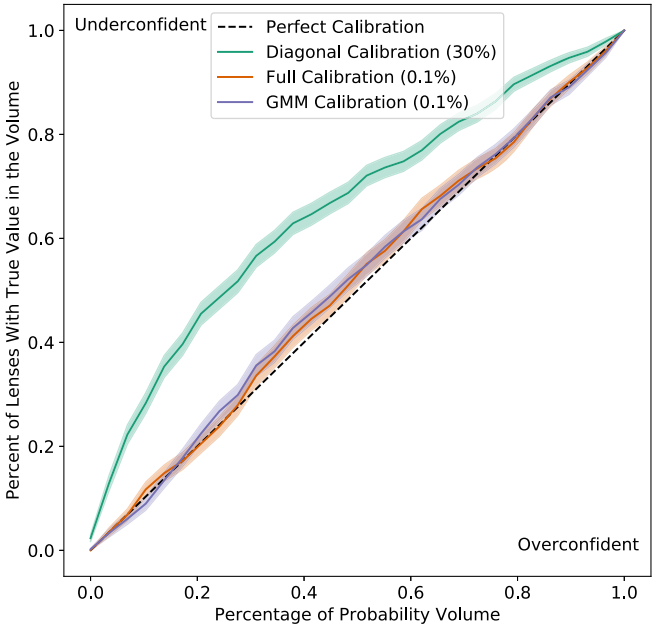
(a) Calibration comparison for diagonal models (**Validation**)



(b) Calibration comparison for full models (**Validation**)



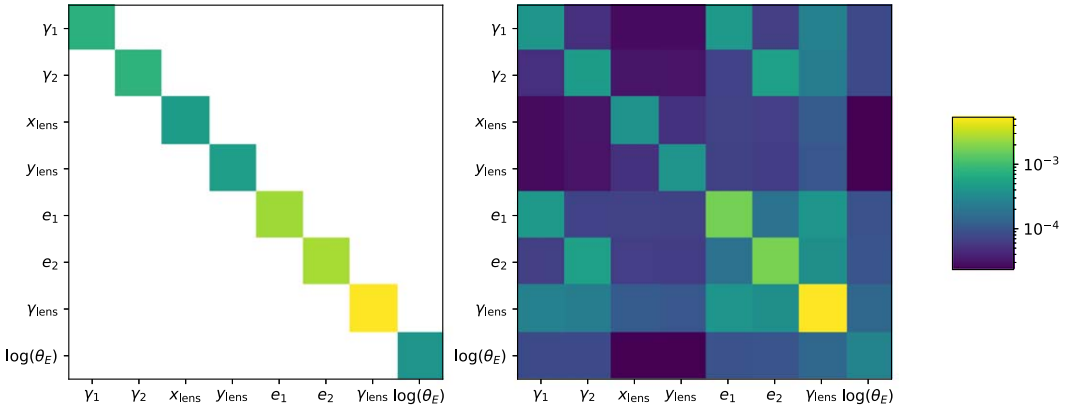
(c) Calibration comparison for GMM models (**Validation**)



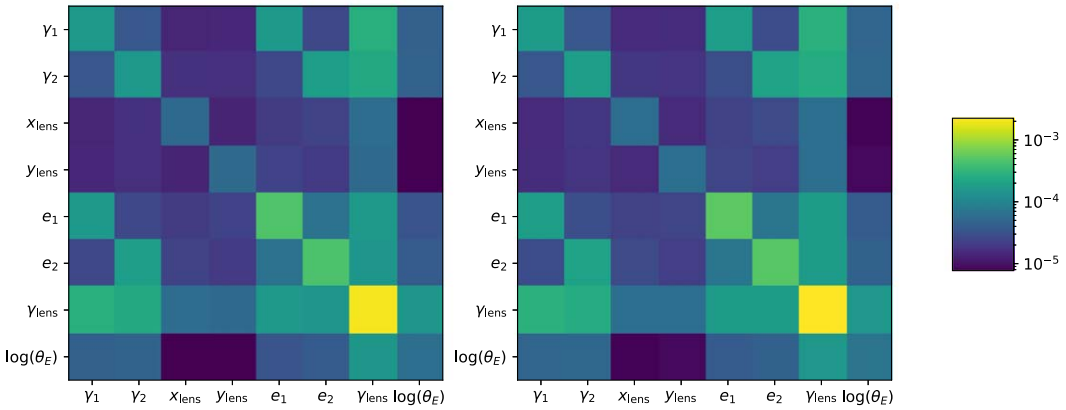
(d) Calibration comparison for three BNN types (**Test**)

Figure 1. The calibration plots (also known as quantile–quantile plots) for (a) the diagonal posterior models, (b) the full posterior models, (c) the GMM posterior models, and (d) all three posterior models using the best dropout rate for each. The comparison between BNN hyperparameters in (a)–(c) is done on the validation set, while the final comparison of the three models in (d) is carried out on the test set. The shaded region around each calibration line represents the 1σ uncertainty obtained from jackknife resampling. As we go from the most restrictive (diagonal) posterior to the most flexible (GMM) posterior, the models require less dropout to achieve a good calibration and return better overall calibration. The final comparison of the three models in (d) shows that the GMM and full posterior models can return near-ideal calibration and that all three posteriors, given sufficient dropout, can avoid overconfidence.

jump between 0.1% dropout and no dropout has a significant impact on the MAE. These two diverging trends are likely caused by the order-of-magnitude difference in the dropout being applied to both model types. In the large dropout regime, the dropout rate significantly impacts the variance of our models' predictions, therefore giving a larger MAE for larger dropout. In the small dropout regime, we have shown in Section 4.1 that the dropout has a much smaller effect on the variance. Instead, it is likely that its main impact on MAE performance comes from its ability to mitigate overfitting. In the machine-learning literature, a small dropout rate is often used as a regularizer to help reduce the gap in performance



(a) Diagonal posterior model 30% dropout median aleatoric covariance (left) and median total covariance (right)



(b) Full posterior model 0.1% dropout median aleatoric covariance (left) and median total covariance (right)

Figure 2. A comparison of the median covariances for (a) the diagonal posterior model with 30% dropout and (b) the full posterior model with 0.1% dropout. As expected, the diagonal posterior has a diagonal aleatoric covariance matrix while the full posterior has a covariance matrix with sizable correlations. However, the total covariance matrix has nearly the same form for both models. The weight marginalization being learned by the diagonal posterior model appears to be capturing the same covariances that are explicitly parameterized in the full posterior model.

Table 5
Median Absolute Error on Parameter for All Nine Models

Model	γ_1	γ_2	$x_{\rm lens}$	$y_{\rm lens}$	e_1	e_2	$\gamma_{\rm lens}$	θ_E
Diagonal 5%	0.009	0.008	0.005	0.005	0.014	0.013	0.030	0.005
Diagonal 10%	0.009	0.009	0.005	0.006	0.016	0.015	0.034	0.006
Diagonal 30%	**0.012**	**0.013**	**0.007**	**0.007**	**0.021**	**0.021**	**0.039**	**0.007**
Full 0%	0.011	0.010	0.007	0.006	0.019	0.018	0.036	0.006
Full 0.1%	**0.009**	**0.009**	**0.006**	**0.006**	**0.015**	**0.017**	**0.029**	**0.005**
Full 0.5%	0.008	0.008	0.005	0.005	0.014	0.013	0.026	0.005
Full 1%	0.008	0.008	0.006	0.006	0.014	0.013	0.027	0.005
GMM 0%	0.010	0.011	0.007	0.007	0.018	0.018	0.036	0.006
GMM 0.1%	**0.009**	**0.010**	**0.005**	**0.006**	**0.015**	**0.017**	**0.028**	**0.006**
GMM 0.5%	0.007	0.008	0.005	0.005	0.014	0.013	0.026	0.005
GMM 1%	0.008	0.008	0.004	0.005	0.013	0.011	0.028	0.005

Note. All MAE calculations were done on the validation set. The model rows that are bolded correspond to the models that passed that calibration cut from Section 4.1.1. For definitions of the parameter, see Section 3.2.

between training and validation. Even with our large training set, the efforts made to add noise on the fly, and our validation loss criteria for halting training, a small amount of dropout still appears to be beneficial.

Overall, it is clear that the improved calibration of the GMM and full models over the diagonal models does not come at the cost of prediction accuracy. However, while the 0% and 0.1% dropout models perform equivalently on our calibration metric,

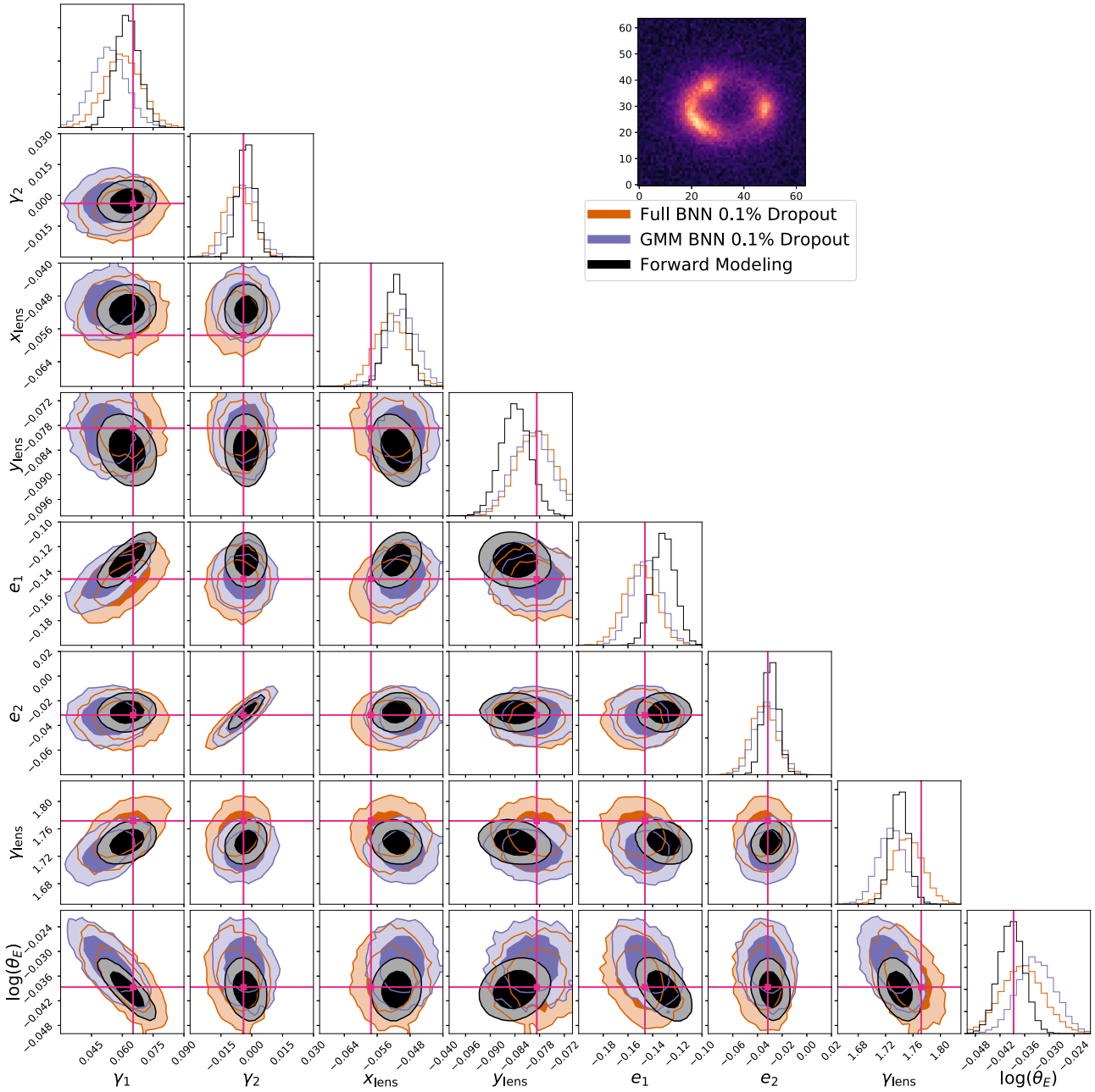


Figure 3. A comparison of the 0.1% dropout full BNN posterior (orange), 0.1% dropout GMM BNN posterior (purple), and forward-model posterior (black) for the lens image shown in the figure. The darker and lighter contours correspond to the 68% and 95% confidence interval, respectively. All three posteriors are statistically consistent with each other and the truth. The forward model, which uses the same model to predict the data likelihood as was used to generate the image, has the smallest uncertainties. However, both the full and GMM posteriors capture the same parameter covariances as the forward model.

their MAE performance is substantially different. Therefore, throughout the remainder of this work, we will focus our attention on the full 0.1% model and the GMM 0.1% model rather than their no-dropout counterparts.

4.1.3. Comparison to Forward Modeling

Although the calibration metric and MAE metric are useful population statistics, it is also interesting to better understand the performance of our different models on a lens-by-lens basis. To do this, we compare the output of the three models that pass our calibration cut to the posterior produced by forward modeling. The likelihoods in our forward-modeling posterior are calculated with the LENSTRONOMY package, and the sampling is conducted using the EMCEE package.[13] While our BNN posteriors are only required to predict the lens and shear parameters, the source parameters of images in our training set vary. Therefore, we are training our BNN to predict posteriors that are marginalized over possible source parameter configurations. To place the comparison on equal footing, the forward-modeling approach also marginalizes over the source parameters.

In Figure 3, we show the two-dimensional corner plots of the GMM 0.1% dropout posterior, the full 0.1% dropout posterior, and

[13] https://emcee.readthedocs.io/en/stable/

Table 6
True/Test Sky Distributions

Component	True/Test Sky Distributions		
	Centered Narrow	Shifted Narrow	Empirical
Number of training points in the test set	577 of 400,000—0.144%	22 of 400,000—0.006%	73,088 of 400,000—18.272%
Lens: PEMD			
x-coordinate lens center ($''$)	$x_{\rm lens} \sim \mathcal{N}(\mu: 0, \sigma: 0.05)$	$x_{\rm lens} \sim \mathcal{N}(\mu: 0.102, \sigma: 0.05)$	$x_{\rm lens} \sim \mathcal{N}(\mu: 0, \sigma: 0.05)$
y-coordinate lens center ($''$)	$y_{\rm lens} \sim \mathcal{N}(\mu: 0, \sigma: 0.05)$	$y_{\rm lens} \sim \mathcal{N}(\mu: -0.102, \sigma: 0.05)$	$y_{\rm lens} \sim \mathcal{N}(\mu: 0, \sigma: 0.05)$
Einstein radius ($''$)	$\theta_E \sim \mathcal{N}_{\log}(\mu: 0.0, \sigma: 0.01)$	$\theta_E \sim \mathcal{N}_{\log}(\mu: 0.1, \sigma: 0.01)$	$\begin{bmatrix} \theta_E \\ q_{\rm lens} \\ \gamma_{\rm lens} \end{bmatrix} \sim \mathcal{N}_{\log}\left(\begin{bmatrix} 0.24 \\ -0.41 \\ 0.70 \end{bmatrix}, \begin{bmatrix} 0.01 & -0.01 & -0.004 \\ -0.01 & 0.13 & 0.01 \\ -0.004 & 0.01 & 0.004 \end{bmatrix}\right)$
Power-law slope	$\gamma_{\rm lens} \sim \mathcal{N}_{\log}(\mu: 0.7, \sigma: 0.01)$	$\gamma_{\rm lens} \sim \mathcal{N}_{\log}(\mu: 0.8, \sigma: 0.01)$	
x-direction ellipticity eccentricity	$e_1 \sim \mathcal{N}(\mu: 0, \sigma: 0.03)$	$e_1 \sim \mathcal{N}(\mu: 0.2, \sigma: 0.03)$	
xy-direction ellipticity eccentricity	$e_2 \sim \mathcal{N}(\mu: 0, \sigma: 0.03)$	$e_2 \sim \mathcal{N}(\mu: -0.2, \sigma: 0.03)$	$\phi_{\rm lens} \sim \text{Unif}(-\frac{\pi}{2}, \frac{\pi}{2})$
External shear			
Shear modulus	$\gamma_{\rm ext} \sim \mathcal{N}_{\log}(\mu: -2.73, \sigma: 0.1)$	$\gamma_{\rm ext} \sim \mathcal{N}_{\log}(\mu: -1.3, \sigma: 0.1)$	$\gamma_{\rm ext} \sim \mathcal{N}_{\log}(\mu: -2.73, \sigma: 0.1)$
Orientation angle	$\phi_{\rm ext} \sim \text{Unif}\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$	$\phi_{\rm ext} \sim \text{Unif}\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$	$\phi_{\rm ext} \sim \text{Unif}\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$

Note. $\mathcal{N}$ is the normal distribution and $\mathcal{N}_{\log}$ is the log-normal distribution. The source parameters are not specified because they are identical to those presented in Table 3. Note that, for the empirical distribution, we draw from the axis ratio $q_{\rm lens}$ and the angle $\phi_{\rm lens}$ as specified in Section 3.2.

the forward-modeling posterior for a specific lens. To avoid any bias, the lens was selected at random from the test set. From a visual comparison, we can see that the three distributions are all statistically consistent with each other. No distribution exhibits any obvious bias from the true value, and the covariances between parameters in the GMM, full, and forward-modeling case match closely. Both BNN posteriors give larger uncertainties across the board, but this is to be expected: the forward-modeling approach samples the true likelihood of the data using the same model that generated the data. It therefore has access to the maximum information and generates contours that represent the limits on the constraining power of the lens image. The forward-modeling uncertainty represents the theoretical minimum on the uncertainty our BNN could achieve. In Appendix B, we produce the same plots for the diagonal model. As we would expect from the results of our previous sections, all three posteriors appear to be unbiased. But as we move from our diagonal posterior to the full and GMM posterior, the BNN predictions tighten and exhibit covariances that more closely correspond to those of the forward model.

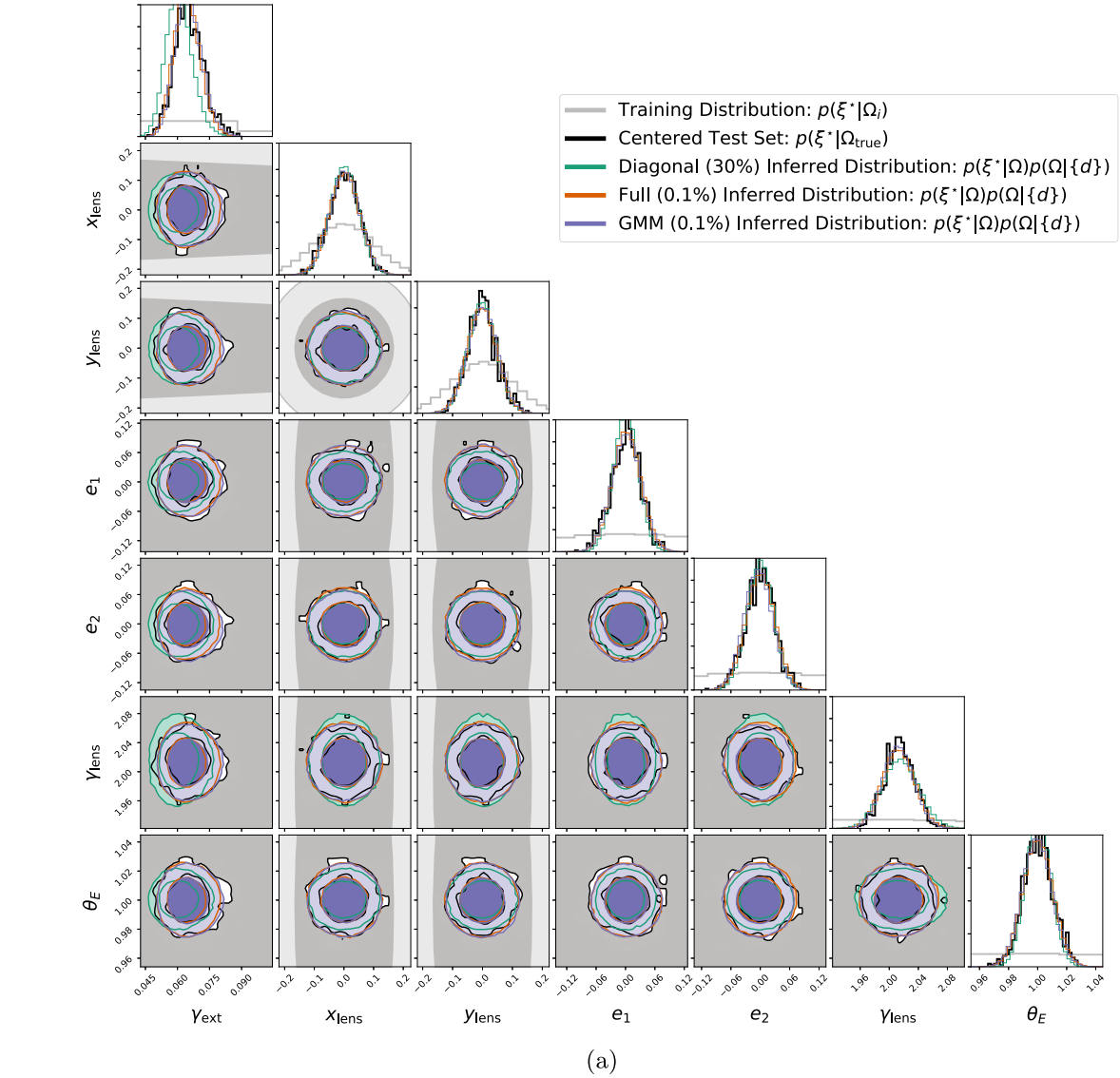
4.2. Tests on "True" Sky Distributions

So far, we have established that the full and GMM BNN models produce well-calibrated posteriors, capture the desired covariances, and are accurate and precise. However, we also want to demonstrate the value and limitations of our networks in producing scientific constraints. We will focus on the ability of our BNN to infer the population hyperparameters used to generate a set of "true" sky test sets. Using these test sets highlights one potential limitation of BNNs. If the training samples are drawn from a different distribution than the test sky —as is almost guaranteed to be the case for real-world applications—then the interim prior will produce biased posteriors. In Section 2.2, we introduce a hierarchical inference framework that achieves two goals: first, it allows us to reconstruct the population-level distributions of the lens parameters; second, it allows us to reweight our inference and correct for the assumption of the interim prior. To test the viability of this framework, we introduce three "true" skies that exhibit some of the systematic biases we would expect between our training set and a sample of real data:

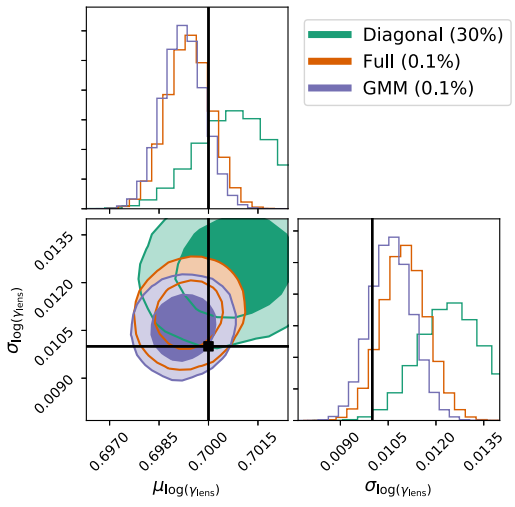
1. Centered Narrow Distribution: the distributions used to draw the "true" sky lens parameters have the same means as the training set but are much narrower.
2. Shifted Narrow Distribution: the distributions used to draw the "true" sky lens parameters are much narrower than the training set and have their means shifted by $\pm\sigma_{\rm train}$—the standard deviation of the training set distributions. Note that, because each parameter is shifted by $\pm\sigma_{\rm train}$, the shift in the full eight-dimensional space is much larger.
3. Empirical Distribution: the distributions used to draw the "true" sky lens parameters are narrower than the training set and include covariances between the parameters $\gamma_{\rm lens}$, θ_E, and $q_{\rm lens}$. The correlation coefficients have been matched to empirical estimates from the Strong Lensing Legacy Survey (SL2S) and the Sloan ACS Lens Survey (SLACS; Sonnenfeld et al. 2013, 2015). The means of the parameters $\gamma_{\rm lens}$ and $q_{\rm lens}$ are also set to agree with the SL2S and SLACS lens samples.

The specific parameters of each of these three distributions can be found in Table 6. For all three of our test distributions, we have drawn 1024 lens samples obeying the same instrumental and noise specifications used on our training set (see Section 3.2 for more details). We do not specify the source parameters because they follow the same distribution as the training set. Because our BNN does not predict posteriors on the source parameters, we felt varying the source distribution could be better explored in future work.

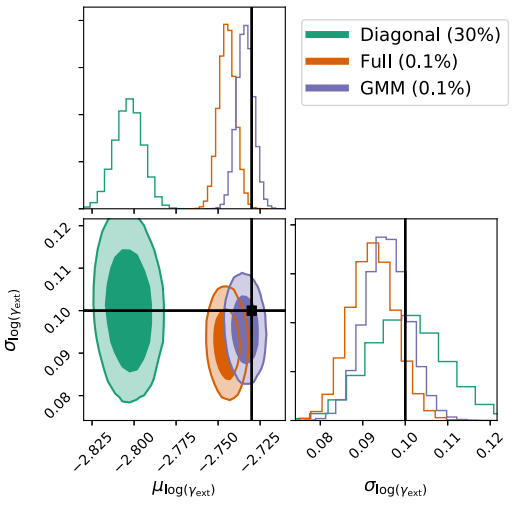
Each of the three distributions introduces an additional element of complexity that increases both the realism of the lens sample and the potential bias introduced by the interim prior. The centered narrow distribution addresses our framework's ability to reconstruct tight distributions. As we discuss in Section 2.2, our hierarchical inference framework imposes the explicit requirement that our training distribution be broader



(a)



(b)



(c)

Figure 4. A set of figures demonstrating the performance of our three types of BNNs on the centered narrow test set. In (a), we plot a comparison of the training set distribution, the centered test set samples, and the parameter distributions inferred by our three BNNs after hierarchical inference. Overall, all three BNNs reconstruct the population distribution of the centered test set with a high level of precision. The only exception is the BNN reconstruction of the $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$ distributions where the diagonal BNN appears to show some bias. The posterior on the population hyperparameters for $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$ are shown in (b) and (c), respectively.

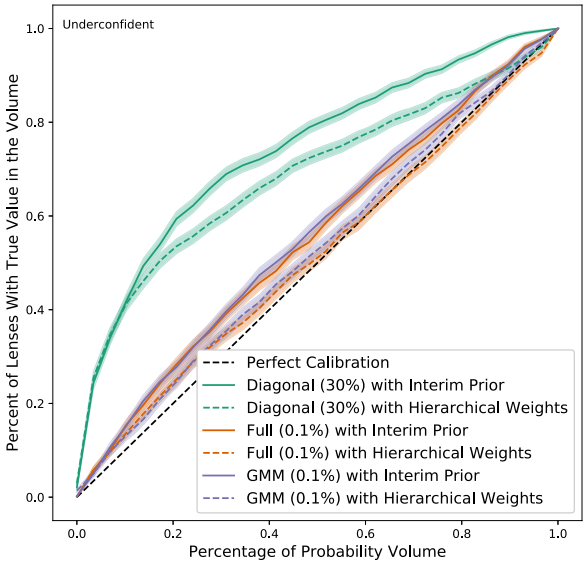


Figure 5. The quantile–quantile plot for the BNN posteriors before and after hierarchical reweighting on the centered narrow test set. The underconfidence introduced by transitioning to the centered narrow test set is effectively mitigated by the use of our hierarchical inference framework.

than our expected test distribution, so this type of bias is almost guaranteed. The shifted narrow distribution adds a 1σ shift in the mean of each individual parameter, testing the framework's ability to reweight correctly in an asymmetric and heavily undersampled region of the interim prior. Finally, the empirical distribution introduces means and covariances that better agree with our current lens sample. The addition of population-level covariances is important not only because it introduces a significant bias between the training and test set but also because these covariances probe loosely understood properties of galaxy formation.

Our goal is to sample the posterior given by Equation (10), reproduced below:

$$p(\Omega|\{d\}) = \underbrace{p(\Omega)}_{\Omega \text{ prior}} \times \underbrace{\prod_k \frac{p(d_k|\Omega_{\rm int})}{p(\{d\})}}_{\text{normalizing factor}} \times \underbrace{\prod_k \frac{1}{N} \sum_{\xi_k \sim p(\xi_k|d_k,\Omega_{\rm int})} \frac{p(\xi_k|\Omega)}{p(\xi_k|\Omega_{\rm int})}}_{\text{MC with re-weighting}}. \quad (20)$$

As a reminder, k is a product over our 1024 lenses, $\xi_k \sim p(\xi_k|d_k, \Omega_{\rm int})$ are draws from our BNN posterior, and $\Omega_{\rm int}$ is our interim prior. We sample 1000 points from our BNN posterior for each lens. To construct our posterior, we use an ensemble sampler with affine invariance (Goodman & Weare 2010) implemented using the EMCEE package[14] (Foreman-Mackey et al. 2013). The distributions Ω we are sampling over are restricted to having the same functional form as the distributions used to generate the true/test skies. We use broad uniform priors $p(\Omega)$ for all our hyperparameters.[15]

[14] https://emcee.readthedocs.io

[15] For conciseness, we do not reproduce the exact bounds of all our priors here, but they can be found in the repo (https://github.com/swagnercarena/ovejero/tree/master/configs/baobab_configs).

4.2.1. Centered Narrow Distribution

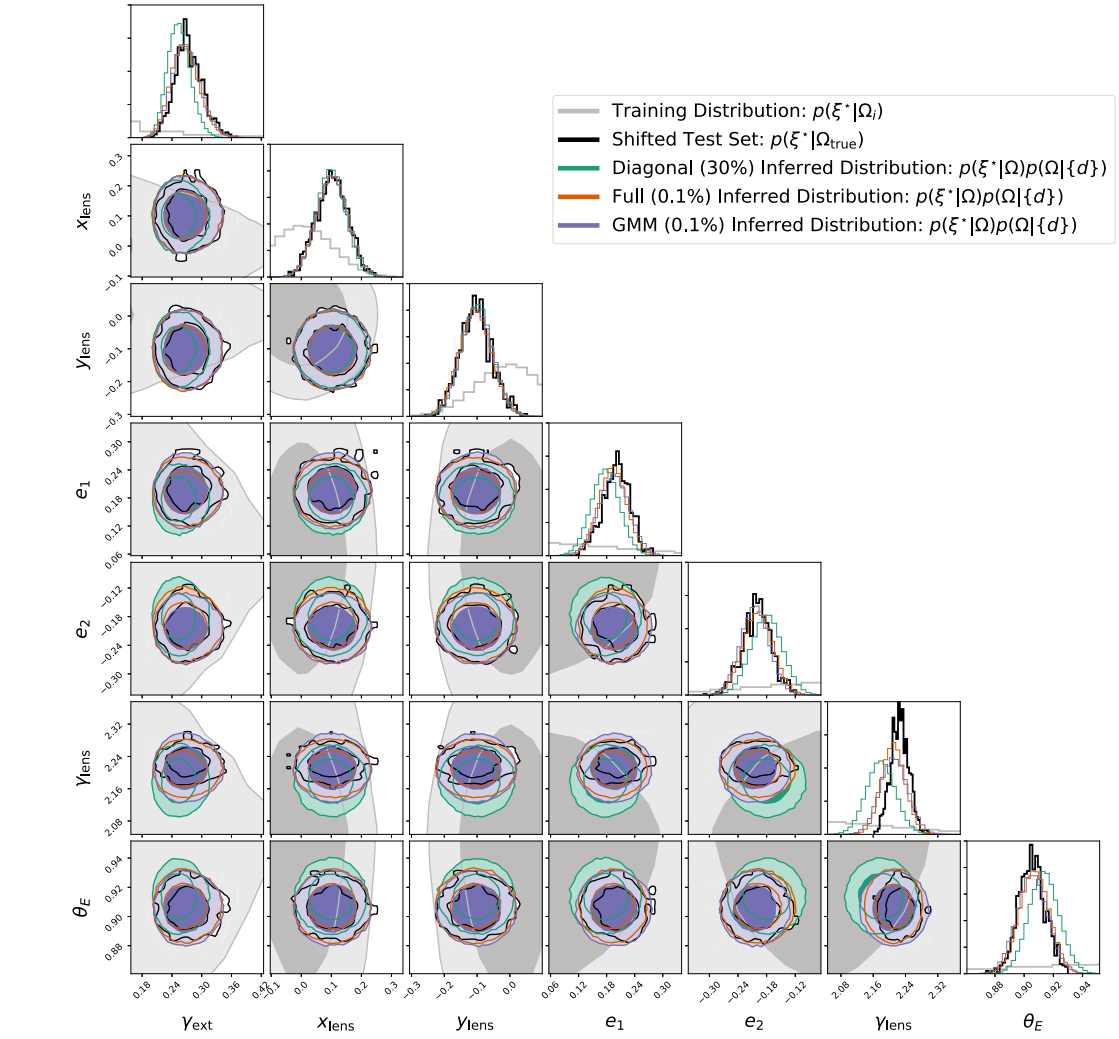
In Figure 4(a), we show a comparison of the training set, centered narrow test set, and the inferred population distribution for each of our three BNN models. For the GMM and full posterior models, the inferred distributions match the test distribution well across all the parameters. The diagonal model also does a good job of matching the distributions with the notable exception of $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$, where it displays some bias in the width and mean. $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$ are two of the most important parameters for population studies of strong lenses because they connect directly to the large- and small-scale distribution of dark matter. In Figures 4(b) and (c), we show the inferred population parameter for $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$, respectively. The bias of the diagonal distribution is most pronounced in $\gamma_{\rm ext}$, where the range of inferred means is substantially offset from the true value. The GMM model returns posteriors for the population hyperparameters that are both tightly constrained and consistent with the truth. The full posterior also offers tight, unbiased constraints with the exception of the mean in the shear. There, the constraints exhibit a slight downward bias of approximately 0.5% in the mean.

We can also return the calibration metric we introduced in Section 4.1 to understand how our hierarchical inference affects the calibration of our posteriors. In Figure 5, we show the quantile–quantile plot for our three BNN models before and after the hierarchical reweighting. If we do no reweighting, and therefore assume the interim prior, all three models return posteriors that are underconfident compared to what we had gotten on the validation set. This follows our intuition of what should happen on a centered narrow test set: the narrower distribution of lenses should allow for tighter constraints than the interim prior, while the shared means should allow us to assume the interim prior without introducing bias. When we use Equation (11) to factor in the hierarchical weights we find that the full and GMM models once again return a near-perfect calibration. The diagonal model is still underconfident, although this is consistent with its original performance in Section 4.1.

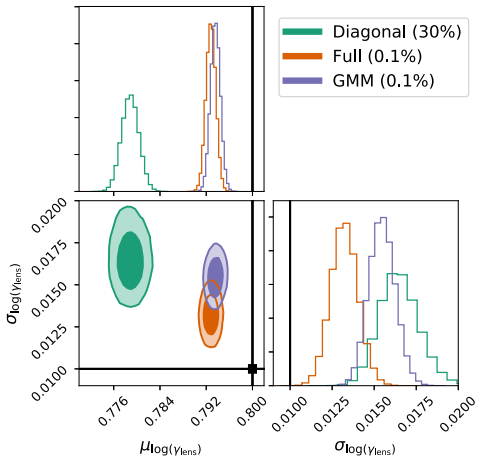
4.2.2. Shifted Narrow Distribution

In Figure 6(a), we show a comparison of the training set, shifted narrow test set, and the inferred population distribution for each of our three BNN models. Unlike the centered narrow distribution, all three BNN models show bias toward the training set in the inferred population hyperparameters. For the full and GMM models, the only pronounced shift is in the distribution of $\gamma_{\rm lens}$; the hierarchical inference on both models returns a distribution that is slightly too broad and shifted toward smaller values of $\gamma_{\rm lens}$. The diagonal model shows a similar bias across multiple parameters, including $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$. Figures 6(b) and (c) show the posteriors for the population hyperparameters of $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$. None of the three models contain the truth within their 95% confidence interval, although the full and GMM models are much closer than the diagonal model. For the full and GMM models, the error in the means for $\gamma_{\rm lens}$ is roughly 8% of the shift from the training distribution whereas for $\gamma_{\rm ext}$ it is around 2%.

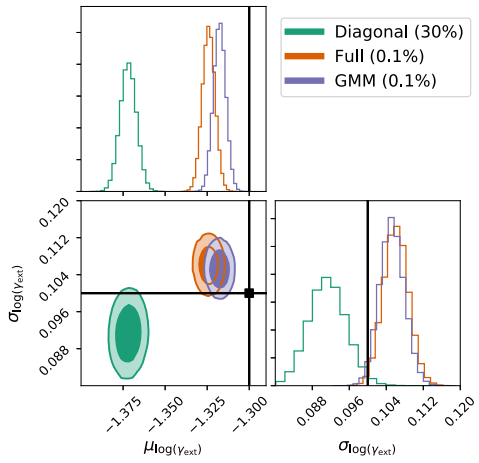
The calibration results in Figure 7 show that all three models are overconfident in their predictions when the interim prior is assumed. The hierarchical reweighting helps correct for this overconfidence, but both the full and GMM models still do not



(a)



(b)



(c)

Figure 6. A set of figures demonstrating the performance of our three types of BNNs on the shifted narrow test set. In (a), we plot a comparison of the training set distribution, the shifted test set samples, and the inferred parameter distributions by our three BNNs after hierarchical inference. Unlike the centered narrow test set, the BNNs have mixed success reconstructing the population hyperparameters. This is especially true for the diagonal BNN, which shows a consistent bias toward the training set in its inferred distribution. The posteriors on the population hyperparameters for $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$ are shown in (b) and (c), respectively. While the bias in the means for the GMM and full model is small (on the order of 1%–2%), none of the BNNs return constraints statistically consistent with the truth.

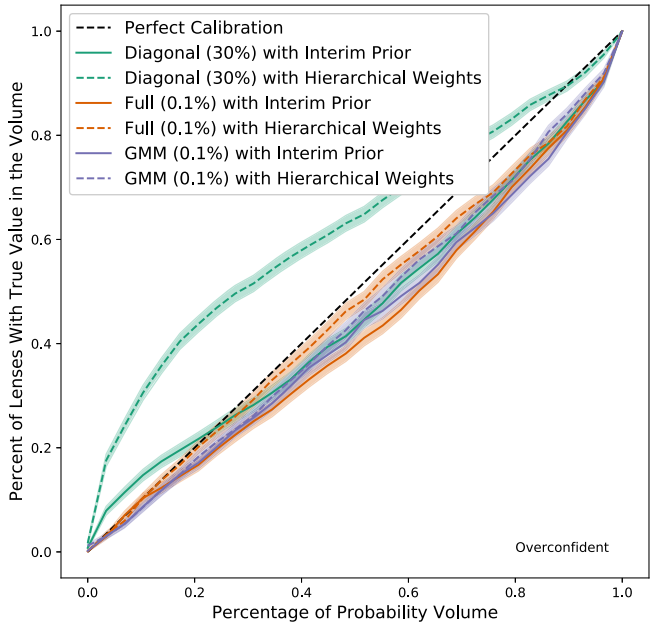


Figure 7. The quantile–quantile plot for the BNN posteriors before and after hierarchical reweighting on the shifted narrow test set. The overconfidence introduced by transitioning to the shifted narrow test set is partially but not fully corrected by the hierarchical weights. This is consistent with the bias in the estimate of the population hyperparameters shown in Figure 6.

return the near-perfect calibration they achieved on the centered narrow test set. This is most prominent in the tails of the posteriors. Given the bias in the inferred population hyperparameters, it is not surprising that the reweighted posteriors still exhibit overconfidence. The calibration metric on the reweighted posteriors of the diagonal model does not demonstrate overconfidence, but it does show a significant underconfidence. While far from optimal, this is in line with the diagonal model's performance in Figure 1.

The shifted narrow test set performance demonstrates that if the training set is sufficiently offset from the test distribution (or true sky), the hierarchical inference procedure will show bias. A 1σ shift in each of our eight parameters means that only 0.006% of the training examples fall within the test distribution. However, even in this undersampled space, the full and GMM models still return inferred distributions that largely overlap with the test set distribution. If these inferred distributions were returned on a real strong-lensing data set, it would strongly indicate the need for some form of retraining.[16]

4.2.3. Empirical Distribution

Figure 8 shows a comparison of the training set, empirical test set, and the inferred population distribution for each of our three BNN models. Recall that, unlike the centered narrow and shifted narrow test distributions, the empirical distribution has a different functional form than the training distribution. All three models capture the correct population-level covariance between θ_E and $\gamma_{\rm lens}$, but struggle on the covariance matrix parameters tied to $q_{\rm lens}$. In particular, the inferred distributions for our diagonal, full, and GMM models all underrepresent values of $q_{\rm lens}$ near 1. This bias seems to stem from the limitations of the training set: there are very few training examples with $q_{\rm lens} > 0.9$. This is a natural consequence of defining our training distribution in terms of Gaussian samples of the ellipticities e_1 and e_2 instead of the axis ratio $q_{\rm lens}$. A Gaussian sample of axis ratio values corresponds to an exponentially peaked sample of ellipticities. In Figures 9(a) and (b), we show the posteriors on the population hyperparameters for the multivariate Gaussian component of the empirical distribution. The full and GMM models capture the truth for all parameters that do not involve $q_{\rm lens}$. For $q_{\rm lens}$, the estimate of the mean and variance is biased low. There is also some bias in the covariance parameters associated with $q_{\rm lens}$, but the bias can be fully explained by the underestimate of the variance in $q_{\rm lens}$. While the diagonal model infers a distribution close to that of the full and GMM models, it also shows bias for the $\gamma_{\rm lens}$ population hyperparameters.

The diagonal model's biggest failure is on the population hyperparameters for $\gamma_{\rm ext}$. In Figure 8, we can see that the inferred distribution for the diagonal model significantly underestimates the scatter and the mean. The calibration results presented in Figure 10 suggest that part of the cause may be the diagonal model's significant underconfidence. An overestimate of the observational uncertainties could lead to the significant underestimate of the intrinsic scatter seen here. We will explore this possibility further in Section 4.2.5.

Figure 10 also shows that the full and GMM models already return well-calibrated posteriors without hierarchical reweighting. Including the hierarchical reweighting improves the posterior calibration for both models, but does not quite reach the same performance seen in Figure 1. Both results agree with the intuition we have built thus far: the BNNs return good calibration results without reweighting because the overlap between the empirical distribution and training distribution is substantial. Similarly, the weights should not fully correct the underconfidence in the posterior because the inferred distributions have some bias.

The empirical test set serves both as a good demonstration of the strengths and limitations of our combined BNN and hierarchical inference approach. Although our training set is drawn from a distribution with no population-level covariance between parameters, our method is capable of accurately reconstructing the covariances present in the test set. At the same time, as we saw with the shifted narrow test distribution, our hierarchical pipeline returns poorer results when our inference is pushed to the tails of our training set.

4.2.4. Varying the Number of Lenses

So far, all of our inference has used the full 1024 lenses in each test distribution. In Figure 11, we show how the posteriors on the population hyperparameters for $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$ change as we reduce the number of lenses. We focus on the centered narrow distribution. The scaling between our constraining power and the number of lenses seems to roughly follow a $\sqrt{N_{\rm lenses}}$ relation. As we go from 64 to 1024 lenses, the posteriors remain statistically consistent with one another.

4.2.5. Varying the BNN Dropout Rate

In Section 4.1, we argued for the importance of well-calibrated posteriors. Here, we seek to demonstrate how errors in the calibration can affect our ability to constrain the population hyperparameters. We focus on our 0.1%, 0.5%, and

[16] In this regime, one could turn to the extensive work on iterative retraining in machine learning. For an example of the recent advances in this field, see Greenberg et al. (2019).

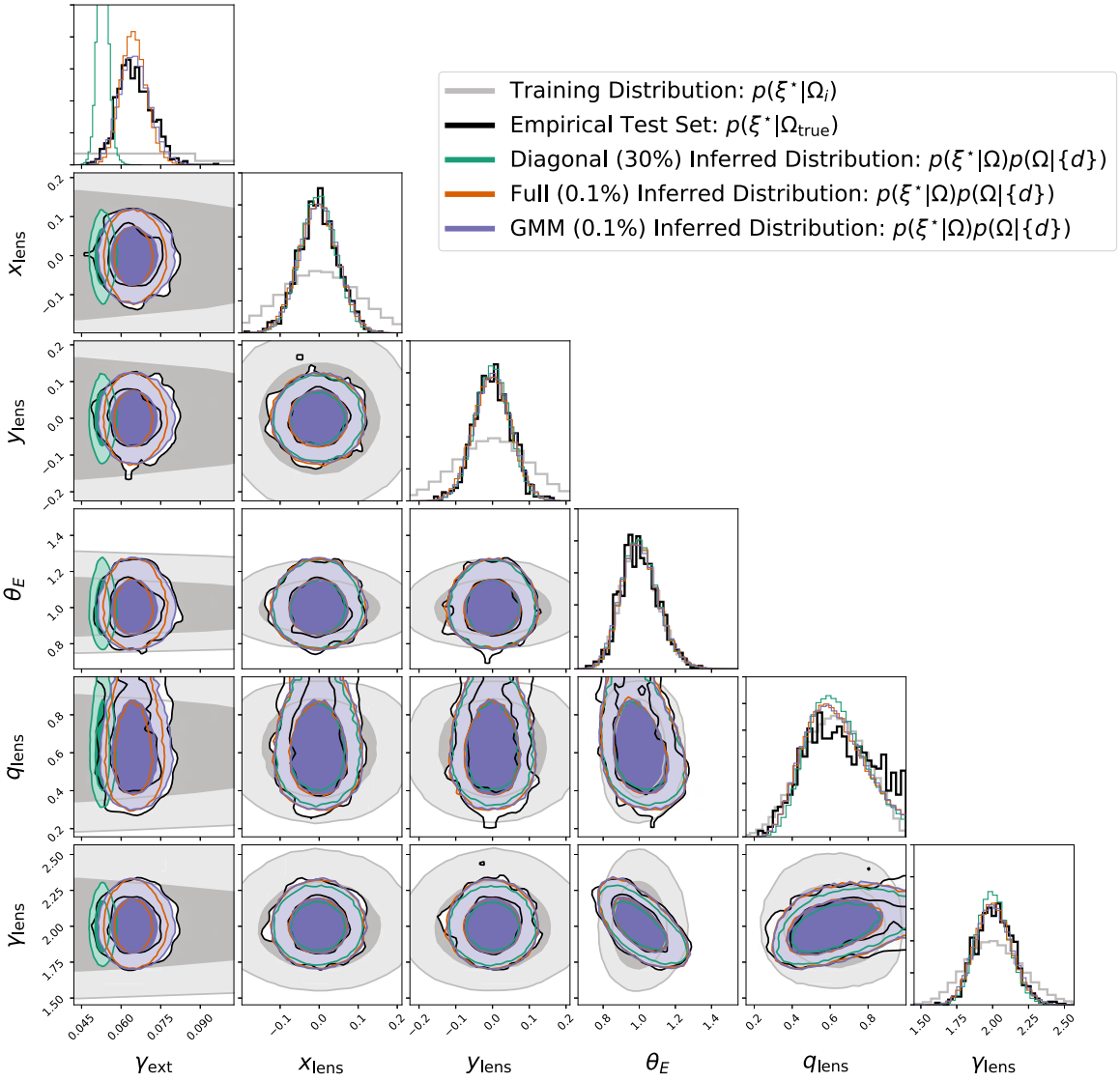


Figure 8. A comparison of the training set distribution, the empirical test set samples, and the inferred parameter distributions by our three BNNs after hierarchical inference. All three models appear to pick up on the covariances in the test set. The inferred distribution from the full and GMM models matches the empirical distribution closely, with the only exception being a bias against values of $q_{\rm lens}$ near 1. The inferred distribution of $\gamma_{\rm ext}$ by the diagonal BNN model also shows significant bias.

1% GMM models and apply them to hierarchical inference on the centered narrow data set. As we show in Figure 1, going from 0.1% to 1% dropout introduces progressively more underconfidence into our posteriors. Figure 12(a) compares the posteriors on the hyperparameters for $\gamma_{\rm lens}$ and $\gamma_{\rm ext}$. The increasing underconfidence of the models appears to map perfectly to a smaller inferred variance. Recast into more traditional astrophysics language, Figure 12 shows that overestimates of the observational uncertainties lead to underestimates of the intrinsic scatter.

Table 5 shows that the 0.5% and 1% GMM models are marginally more accurate in their parameter estimates than the 0.1% GMM model. Despite this, only the 0.1% model returns unbiased posteriors for all four hyperparameters shown here. The volume of the contours also seems unaffected by the difference in MAE between the three GMM models. This reinforces our assertion that calibration is a more important metric for assessing model performance than raw prediction accuracy.

5. Conclusion

We have presented a combined BNN and hierarchical modeling framework that is capable of producing rapid, unbiased samples of lens parameter posteriors. Utilizing our publicly available package OVEJERO, we have extended a previous implementation of BNNs in the strong-lensing literature (Perreault Levasseur et al. 2017) to include more flexible posteriors and calibration metrics on the eight-dimensional parameter space of a PEMD model with external shear. We show that a mixture of two Gaussians is capable of returning posteriors that are both precise and statistically consistent. When applied to "true" skies drawn from different distributions than the training set, our models begin to show systematic biases associated with the interim prior learned in training. To address this shortcoming, we have developed a hierarchical analysis framework for our lenses that allows us to deconvolve the interim prior in the posteriors of individual lenses. Our final ensemble approach produces unbiased posterior samples and gives us access to the underlying

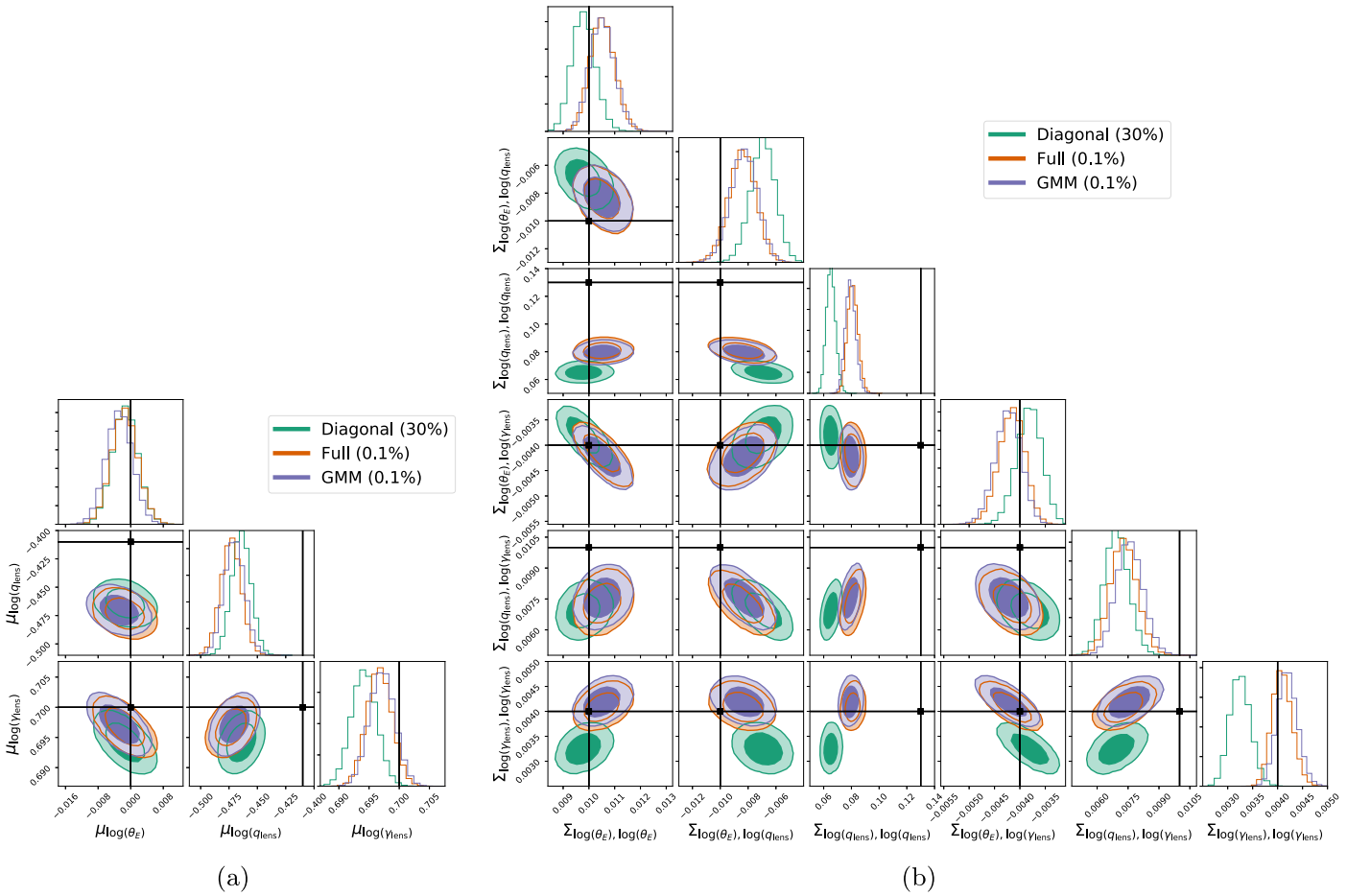


Figure 9. The posteriors on the population hyperparameters for θ_E, $q_{\rm ext}$, and $\gamma_{\rm lens}$ on the empirical test set. Because these three lens parameters are governed by a multivariate Gaussian distribution, we group the hyperparameters by the mean (a) and the covariance matrix (b). The posteriors for the full and GMM models capture the truth for the population mean and covariances governing $\gamma_{\rm lens}$ and θ_E. However, there is a clear bias for the mean and covariance values associated with $q_{\rm lens}$. The diagonal model posteriors exhibit this same bias along with a bias on the $\gamma_{\rm lens}$ parameters.

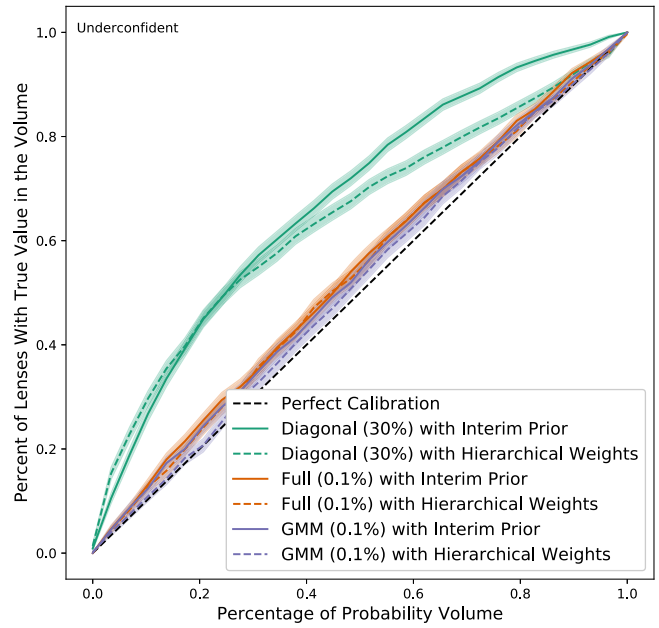


Figure 10. The quantile–quantile plot for the BNN posteriors before and after hierarchical reweighting on the empirical test set. The underconfidence introduced on the empirical test set is small and mitigated by the hierarchical reweighting. However, only the GMM model approaches the near-perfect calibration from Figure 1.

distribution of lenses in the sky. Notably, nothing in the approach we outline here is limited to strong-lensing science. Our framework offers a general methodology for mitigating training set bias in BNN inference.

Returning to the questions we introduced at the beginning of this work, we conclude with the following thoughts:

1. By default, BNN predictions are not robust to test sets drawn from distributions different from the training set. However, so long as the test set is well contained within the training distribution, adding a hierarchical inference framework allows for statistically accurate posteriors on lens populations that do not match the training distribution. This is true even if the population hyperparameters include higher-order statistics (i.e., covariances) not present in the training set.
2. The same hierarchical inference methodology that extends the BNN to new test distributions can also return posteriors on the population hyperparameters themselves. As with the posterior calibration results, the best performance is achieved when the test distribution is well encompassed by the training set. However, even when the test distribution is on the edges of the training distribution, the bias in the inferred population hyperparameters is small. Reconstructing population hyperparameters also requires BNN models that are very well

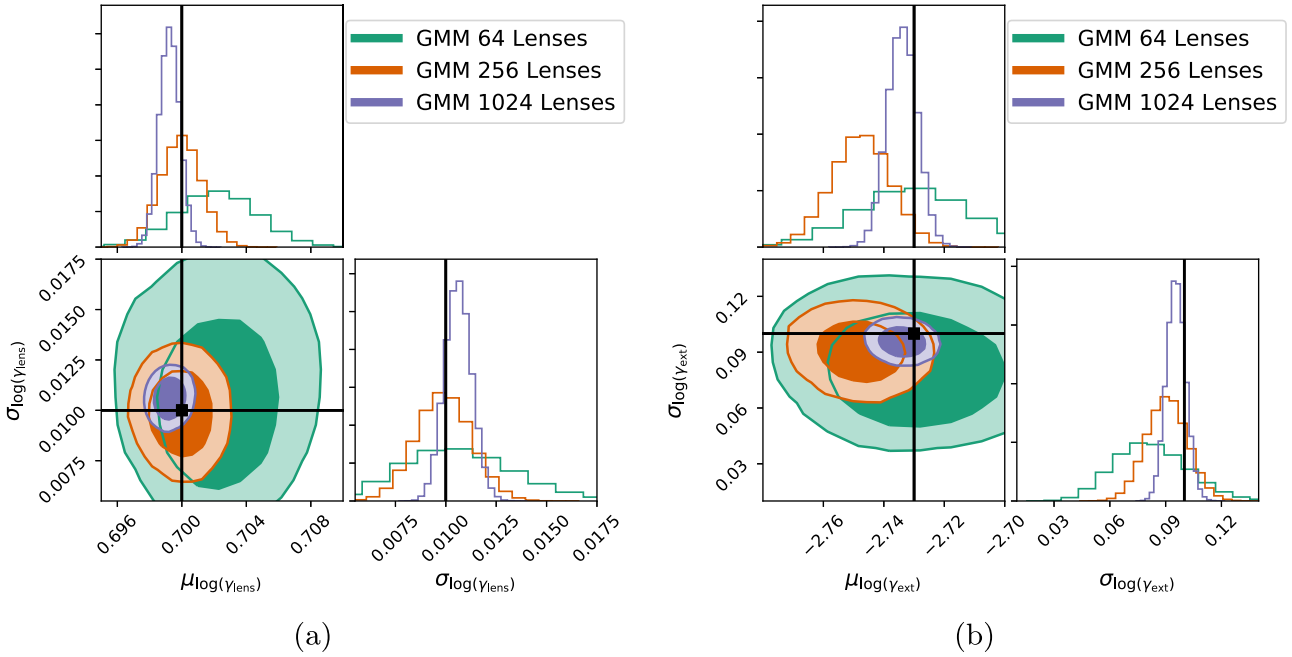


Figure 11. The posterior on the population hyperparameters for $\gamma_{\rm lens}$—(a) and $\gamma_{\rm ext}$—(b) as a function of the number of lenses drawn from the centered narrow distribution. The constraining power scales roughly as $\sqrt{N_{\rm lenses}}$.

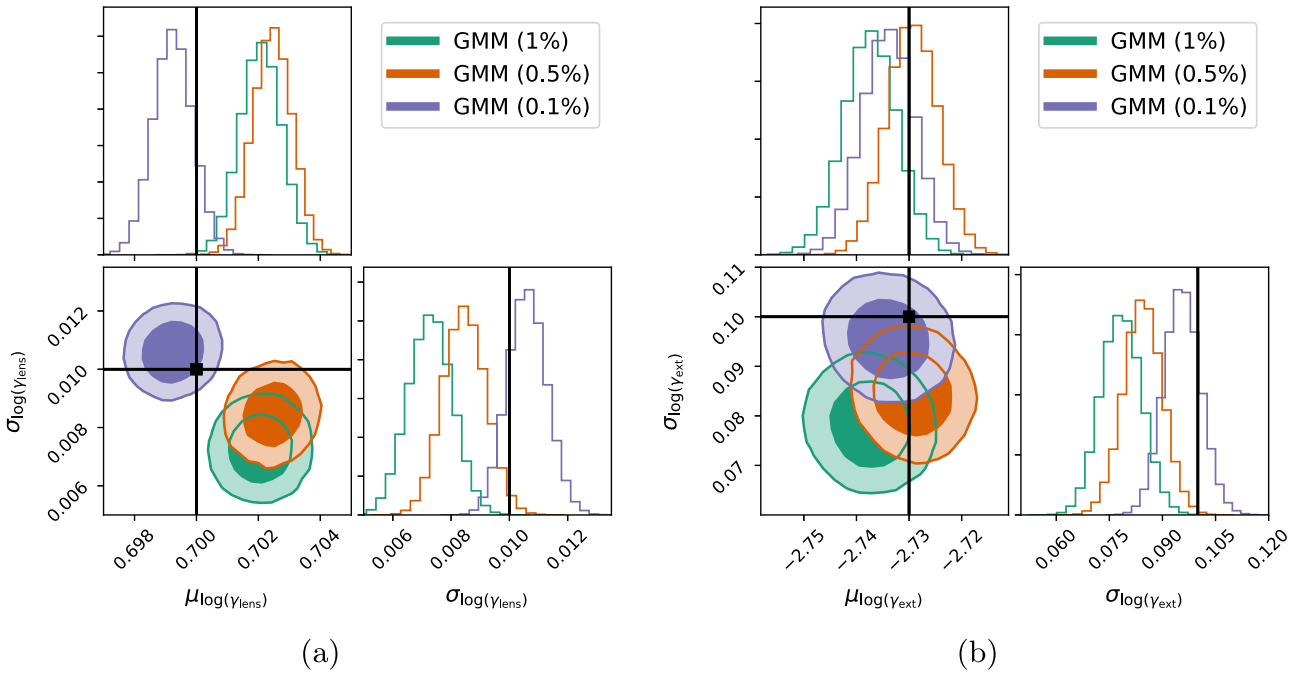


Figure 12. The posterior on the population hyperparameters for $\gamma_{\rm lens}$ (a) and $\gamma_{\rm ext}$ (b) on the centered test set as a function of BNN dropout. As seen in Figure 1, larger dropouts correspond to increasingly underconfident posteriors for the GMM model. This underconfidence (which can also be thought of as an overestimate of the observational uncertainty) maps directly to an underestimate of the intrinsic scatter in the population.

calibrated. Models that are precise in their mean parameter estimates but misquantify their uncertainties will give biased hyperparameters. Notably, being "conservative" by overestimating uncertainties can lead to disastrous underestimates in the intrinsic population scatter.

3. BNNs are capable of returning posteriors on PEMD parameters that are statistically consistent and constraining. When compared directly to forward modeling, the BNN results are consistent and capture the same underlying parameter covariances. However, the overall constraining power of the forward-modeling approach is higher than any of the BNNs we explore here.
4. For simulated PEMD lenses, it is sufficient for our BNN to predict a single multivariate Gaussian with full flexibility in its covariance matrix. For models without this flexibility, the dropout rate can be tuned to capture some of the missing covariances. However, these large

dropout models do not perform well when reconstructing the population hyperparameters.
5. The pipeline we present here trains a BNN, predicts parameter values for lenses, and conducts hierarchical reweighting on a 1000 lens data set in a day.

The industrial-sized samples produced by upcoming surveys will pose a host of new challenges for astronomers. We are confident that the analysis techniques and insights presented here provide the tools necessary to extract the full scientific constraining power these data sets will offer.

This paper has undergone internal review in the LSST Dark Energy Science Collaboration. We would like to thank Laurence Perreault Levasseur, Alessandro Sonnenfeld, and Francois Lanusse for their thoughtful comments on our work. Additionally, we also thank the LSST Dark Energy Science Collaboration Publication Board for their time and feedback.

S.W.C. developed and applied the OVEJERO package described in this work; ran the analysis on the training, validation, and test data sets; wrote the main text; and produced the figures. S.W.C. was supported by the KIPAC-Chabolla fellowship and NSF Award DGE-1656518.

J.W.P. developed the BAOBAB package used to generate all the data sets, provided input on the BNN optimization and posterior inference, and contributed to the text.

S.B. provided support with the training set generation; advised on scope, analysis, and context; and contributed to the text.

P.J.M. provided the initial project design and advised on the probability theory used, the realism of the simulations, and the numerical experimental set-up.

A.R. advised on the scope and context of the project.

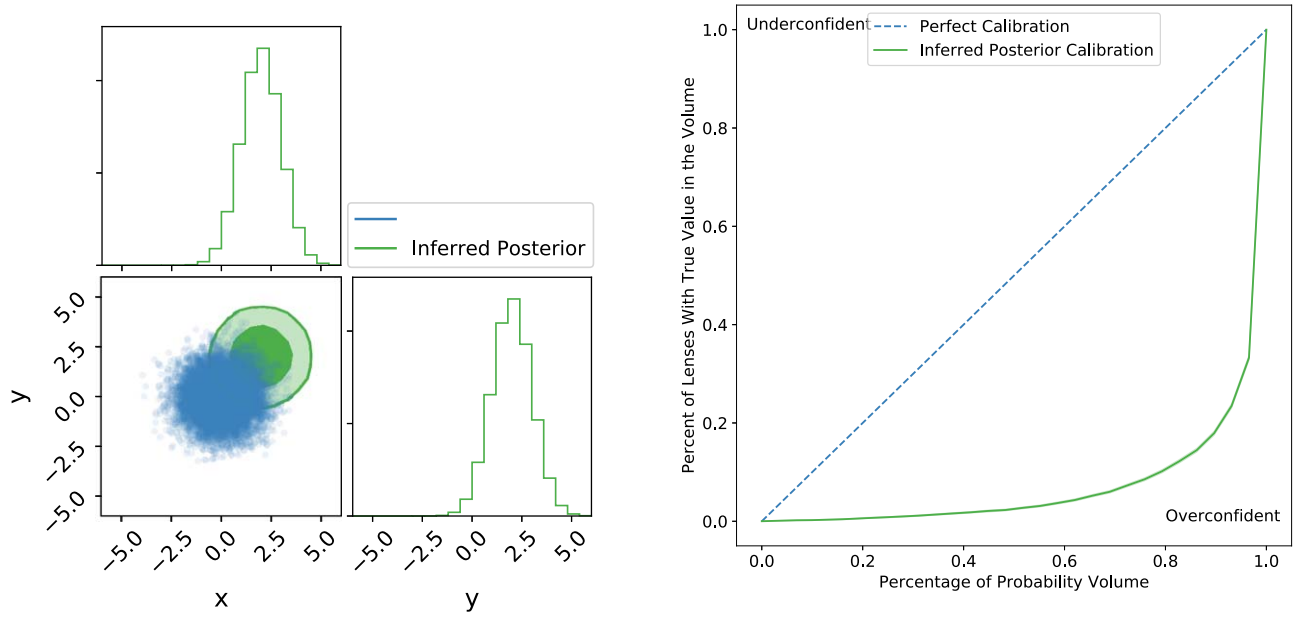
R.H.W. advised on the scope and context of the project and contributed to the text.

The DESC acknowledges ongoing support from the Institut National de Physique Nucléaire et de Physique des Particules in France; the Science & Technology Facilities Council in the United Kingdom; and the Department of Energy, the National Science Foundation, and the LSST Corporation in the United States. DESC uses resources of the IN2P3 Computing Center (CC-IN2P3–Lyon/Villeurbanne—France) funded by the Centre National de la Recherche Scientifique; the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under contract No. DE-AC02-05CH11231; STFC DiRAC HPC Facilities, funded by UK BIS National E-infrastructure capital grants; and the UK particle physics grid, supported by the GridPP Collaboration. This work was performed in part under DOE contract DE-AC02-76SF00515 and NSF award AST-1716527.

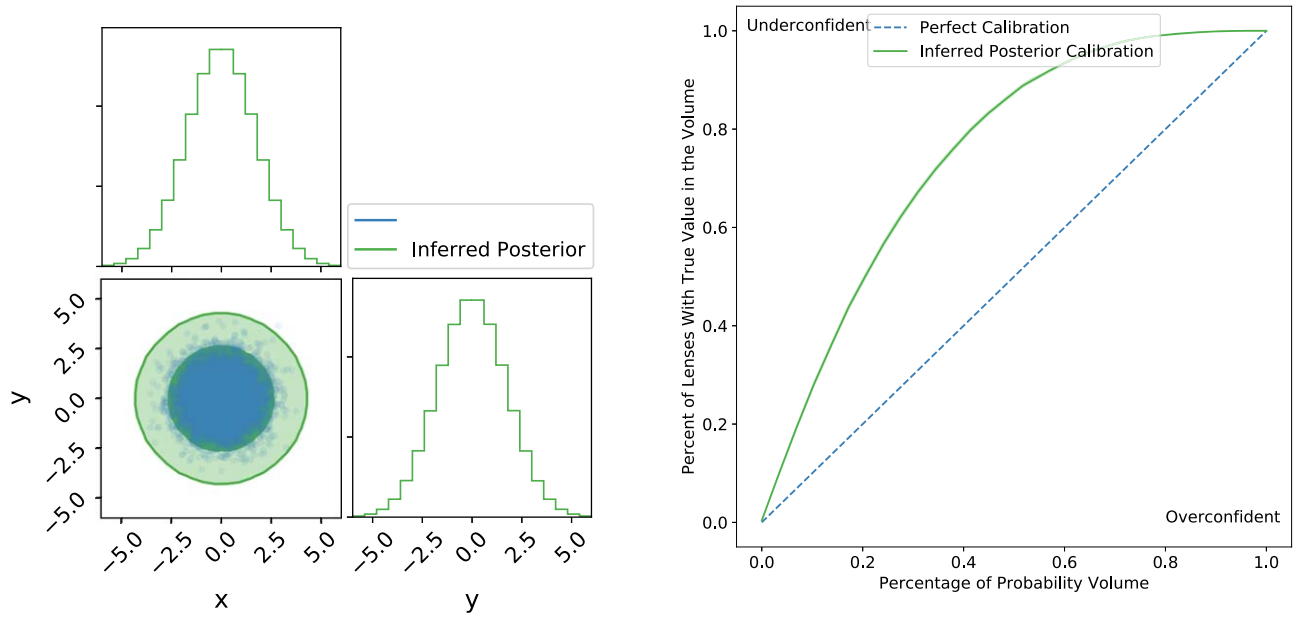
This work was also made possible by the Google Cloud Platform research credits program.

Appendix A
Calibration Toy Models

In Figures 13 and 14, we present a few toy true and inferred posteriors to better build intuition for the quantile–quantile plot. Figure 13 focuses on univariate comparisons and shows that the quantile–quantile plot reflects our intuition of what it means for a model to be under- or overconfident. When the inferred distribution is offset and does not properly account for that offset with the size of its uncertainties, it gives a strong signal of overconfidence. On the opposite end, when the inferred posterior is well calibrated but involves large uncertainties, the quantile–quantile plot gives a strong underconfidence signal. The final example in Figure 14 shows a more realistic scenario: here the inferred posterior is univariate but the true posterior is bivariate. Unlike our more simplistic toy models, the calibration error is neither consistently under- nor overconfident. Instead, in the interior region of our inferred posterior, we find more true posterior samples than we would expect. This is because our single inferred posterior is stretched to try and assign some probability weight to the second mode of the true posterior. This lack of true posterior samples comes through as an underconfidence signal. In the tails of our inferred posterior, we find fewer true posterior samples than we expect because most of the inferred posterior weight is being wasted on the space between the two modes of our true posterior. In our quantile–quantile plot this comes through as an overconfidence signal for x-axis values above 0.8.



(a) Calibration of an offset inferred posterior



(b) Calibration of an inferred posterior with large uncertanties

Figure 13. A visualization of the quantile–quantile plot for two toy scenarios. In (a) the inferred posterior has the correct spread but is offset from the correct mean, leading to significant overconfidence. In (b), the inferred posterior is correctly centered but has too large a spread, leading to significant underconfidence.

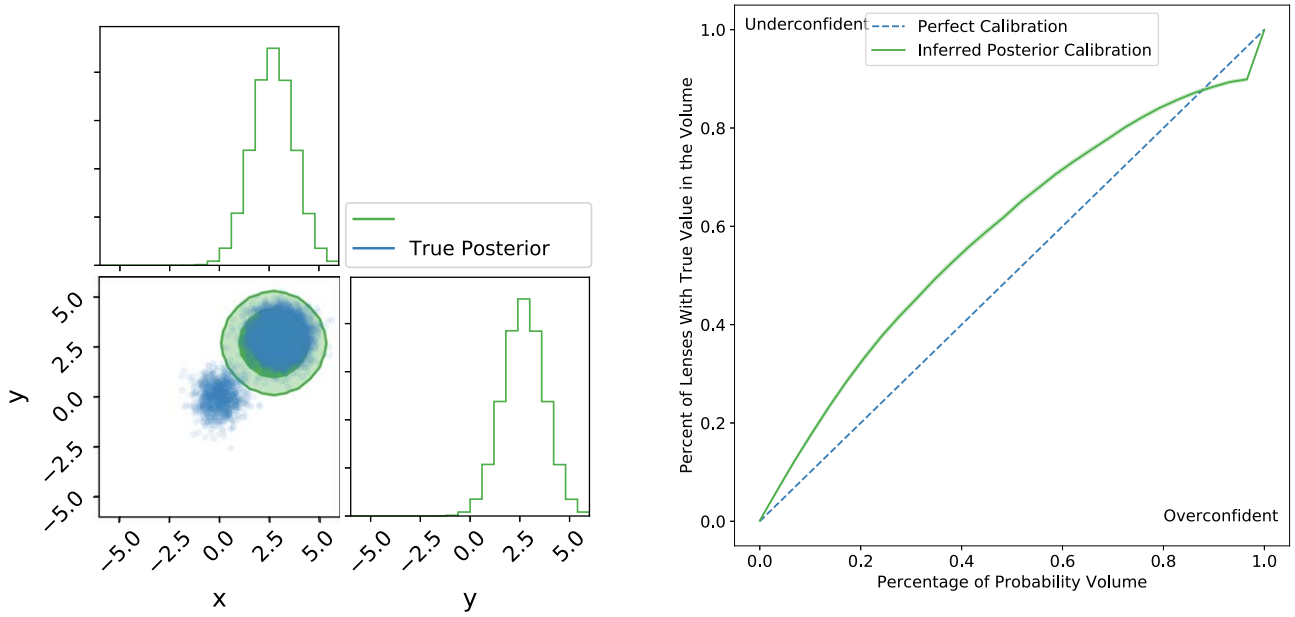


Figure 14. A visualization of the quantile–quantile plot for a toy model where the inferred posterior is univariate but the true posterior is bivariate. Note that the signal falls neither cleanly in the over- or underconfident regions, but rather crosses from one to the other.

Appendix B
Forward-modeling Comparison for Diagonal GMM

In Figure 15, we show a comparison of the forward-modeling posterior to the posterior for the diagonal BNN model on a specific lensing image. As we expect from the results in Section 4, the diagonal BNN gives much larger uncertainties across the board than those seen in Figure 3. Unlike the GMM or full BNN models, the diagonal model does not qualitatively display all of the covariances shown in the forward-model posterior. However, it does include a number of parameter covariances that cannot be explained by the diagonal aleatoric uncertainty. As we discuss in the paper, the preference for a large epistemic uncertainty appears to be partially caused by the epistemic uncertainty's ability to provide posterior flexibility not inherent in the aleatoric model.

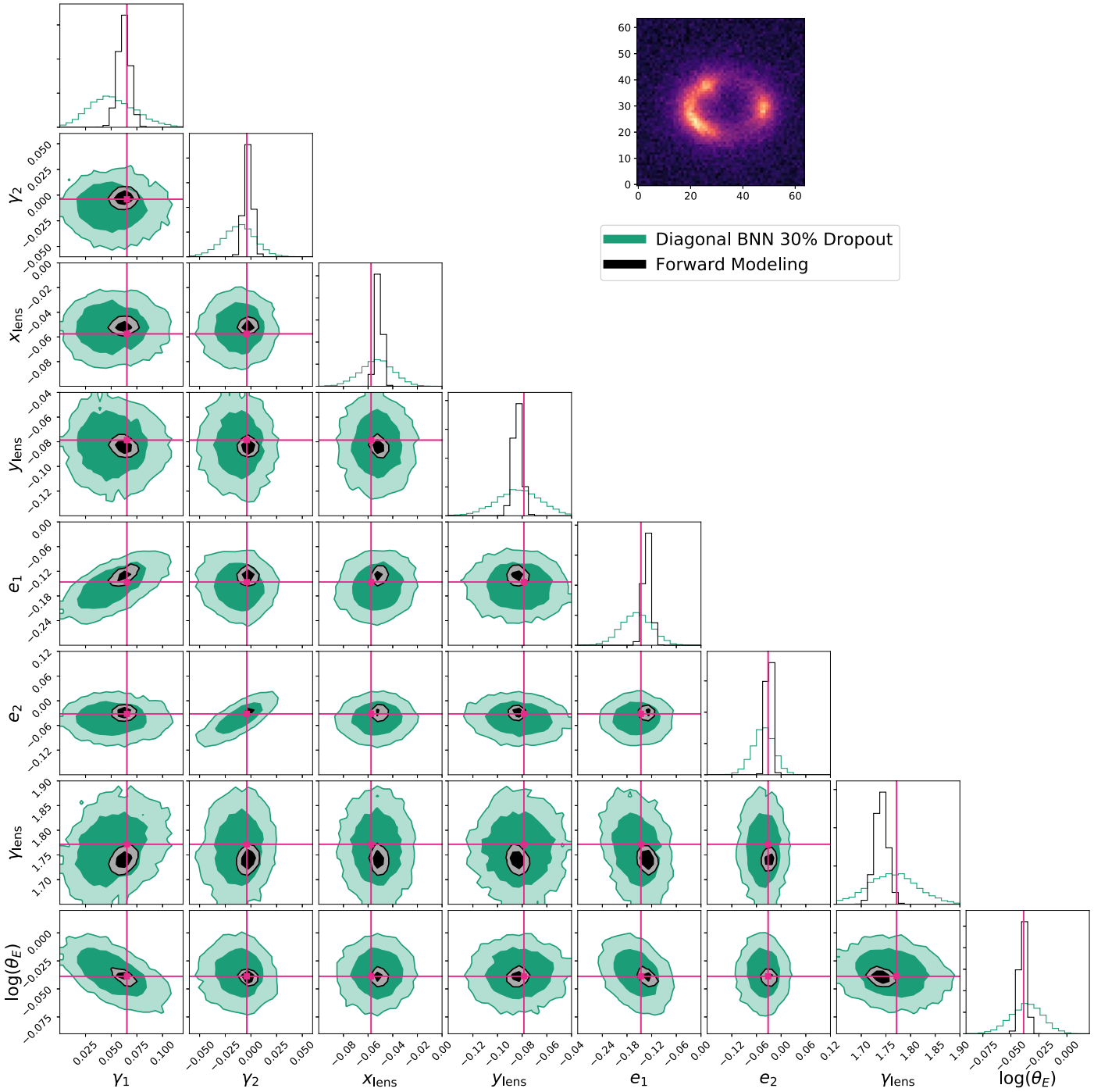


Figure 15. A comparison of the 30% dropout diagonal BNN posterior (green) and forward-model posterior (black) for the lens image shown in the figure. The darker and lighter contours correspond to the 68% and 95% confidence intervals, respectively. Both posteriors are statistically consistent with each other and the truth. The forward model, which uses the same model to predict the data likelihood as was used to generate the image, has much smaller uncertainties than the diagonal covariance model. The diagonal BNN does capture some of the parameter covariances exhibited by the forward model, reinforcing the conclusion that the large epistemic uncertainty is being used to supplement the lack of flexibility in the aleatoric model.

Appendix C
Hierarchical Inference Derivation

Here we derive the equations given in Section 2.2. We start by calculating the probability of a specific test set distribution given the test images $\{d\}$:

$$p(\Omega|\{d\}) = \int \{d\xi\}\ p(\Omega, \{\xi\}|\{d\}) \tag{C1}$$

$$= \int \{d\xi\}\ \frac{p(\{d\}|\{\xi\}, \Omega)p(\{\xi\}|\Omega)}{p(\{d\})} p(\Omega). \tag{C2}$$

So far we have only exploited Bayes' theorem. We assume that each lens is an independent draw from the true distribution Ω. Therefore, given Ω, the parameters of each lens, ξ_k, should be conditionally independent of the other lens parameters. Similarly, given ξ_k, the data generated by that lens should be

conditionally independent both from Ω (since ξ_k is fixed) and from the data for the other lenses. This allows us to simplify the previous equation to

$$p(\Omega|\{d\}) = p(\Omega)\prod_k \int d\xi_k \frac{p(d_k|\xi_k)p(\xi_k|\Omega)}{p(\{d\})}. \quad \text{(C3)}$$

As we discussed in Section 2.1, the distribution we have access to after training on lenses drawn from the interim prior is $p(\xi_k|d_k, \Omega_{\rm int})$, so we will manipulate our expression to give us this term:

$$p(\Omega|\{d\}) = p(\Omega)\prod_k \int d\xi_k \times \frac{p(d_k|\xi_k)p(\xi_k|\Omega_{\rm int})}{p(d_k|\Omega_{\rm int})}\frac{p(d_k|\Omega_{\rm int})}{p(\{d\})}\frac{p(\xi_k|\Omega)}{p(\xi_k|\Omega_{\rm int})} \quad \text{(C4)}$$

$$= p(\Omega)\prod_k \int d\xi_k\, p(\xi_k|d_k, \Omega_{\rm int})\frac{p(d_k|\Omega_{\rm int})}{p(\{d\})}\frac{p(\xi_k|\Omega)}{p(\xi_k|\Omega_{\rm int})}. \quad \text{(C5)}$$

Our BNN allows us to efficiently sample from $p(\xi_k|d_k, \Omega_{\rm int})$, so we can compute our integral through importance sampling:

$$p(\Omega|\{d\}) = \underbrace{p(\Omega)}_{\Omega\text{ prior}} \times \underbrace{\prod_k \frac{p(d_k|\Omega_{\rm int})}{p(\{d\})}}_{\text{normalizing factor}} \times \underbrace{\prod_k \frac{1}{N_{\rm imp}} \sum_{\xi_k\sim p(\xi_k|d_k,\Omega_{\rm int})} \frac{p(\xi_k|\Omega)}{p(\xi_k|\Omega_{\rm int})}}_{\text{MCMC with re-weighting}}, \quad \text{(C6)}$$

where $N_{\rm imp}$ is the number of samples drawn from $p(\xi_k|d_k, \Omega_{\rm int})$. For a detailed discussion of Equation (C6), see Section 2.2. We can now turn our attention to our final goal: calculating the unbiased posterior of a single lens given the full data set, $p(\xi_k|\{d\})$:

$$p(\xi_k|\{d\}) = \int d\Omega p(\xi_k, \Omega|\{d\}) \quad \text{(C7)}$$

$$= \int d\Omega\, \frac{p(\{d\}|\xi_k, \Omega)p(\xi_k|\Omega)p(\Omega)}{p(\{d\})}. \quad \text{(C8)}$$

With the hopes of extracting terms similar to what we find in Equation (C6), we can introduce an integral over the full set of lens parameters $\{\xi\}$:

$$p(\xi_k|\{d\}) = \int d\Omega \int \{d\xi\}\, \frac{p(\{d\}, \{\xi\}|\xi_k, \Omega)p(\xi_k|\Omega)p(\Omega)}{p(\{d\})} \quad \text{(C9)}$$

$$= \int d\Omega \int \{d\xi\}\, \frac{p(\{d\}|\{\xi\}, \xi_k, \Omega)p(\{\xi\}|\xi_k, \Omega)}{p(\{d\})} p(\xi_k|\Omega)p(\Omega). \quad \text{(C10)}$$

As with our previous calculation, $\{d\}$ is independent of Ω given the lens parameters $\{\xi\}$. Similarly, we can take advantage of the conditional independence of d_i on d_j for $i \neq j$ given ξ_i.

$$p(\xi_k|\{d\}) = \int d\Omega \int \{d\xi\}\, \frac{p(\{d\}|\{\xi\})p(\{\xi\}|\xi_k, \Omega)}{p(\{d\})} p(\xi_k|\Omega)p(\Omega) \quad \text{(C11)}$$

$$= \int d\Omega\, p(\Omega)\prod_j \int d\xi_j\, \frac{p(d_j|\xi_j)p(\xi_j|\xi_k, \Omega)}{p(\{d\})}p(\xi_k|\Omega) \quad \text{(C12)}$$

$$= \int d\Omega\, p(\xi_k|\Omega)p(d_k|\xi_k)p(\Omega)\prod_{j\neq k} \int d\xi_j\, \frac{p(d_j|\xi_j)p(\xi_j|\Omega)}{p(\{d\})}. \quad \text{(C13)}$$

In the last step, we have taken advantage of the fact that $p(\xi_j|\xi_k, \Omega)$ for $j = k$ is just a delta function. Now we can introduce our interim prior back into our equation:

$$p(\xi_k|\{d\}) = \int d\Omega\, p(\xi_k|\Omega)p(d_k|\xi_k)p(\Omega)\prod_{j\neq k} \times \int d\xi_j\, \frac{p(d_j|\xi_j)p(\xi_j|\Omega_{\rm int})}{p(d_j|\Omega_{\rm int})}\frac{p(\xi_j|\Omega)}{p(\xi_j|\Omega_{\rm int})}\frac{p(d_j|\Omega_{\rm int})}{p(\{d\})}. \quad \text{(C14)}$$

We can once again introduce the sampling distribution for our BNN $p(\xi_j|d_j, \Omega_{\rm int})$:

$$p(\xi_k|\{d\}) = \int d\Omega\, p(\xi_k|\Omega)p(d_k|\xi_k)\prod_{j\neq k} \times \int d\xi_j\, p(\xi_j|d_j, \Omega_{\rm int})p(\Omega)\frac{p(\xi_j|\Omega)}{p(\xi_j|\Omega_{\rm int})} \times \frac{p(d_j|\Omega_{\rm int})}{p(\{d\})}. \quad \text{(C15)}$$

The term in the product is identical to Equation (C6), except we are not including the lens k in our product. In the limit of many lenses, we can assume that the additional information from one lens to $p(\Omega|\{d\})$ is negligible and rewrite this as[17]

$$p(\xi_k|\{d\}) \approx \int d\Omega\, p(\xi_k|\Omega)p(d_k|\xi_k)p(\Omega|\{d\}). \quad \text{(C16)}$$

We can do one final reintroduction of $\Omega_{\rm int}$:

$$p(\xi_k|\{d\}) \approx \int d\Omega\, \frac{p(\xi_k|\Omega_{\rm int})p(d_k|\xi_k)}{p(d_k|\Omega_{\rm int})} \times p(d_k|\Omega_{\rm int})\frac{p(\xi_k|\Omega)}{p(\xi_k|\Omega_{\rm int})}p(\Omega|\{d\}) \quad \text{(C17)}$$

$$= \int d\Omega\, p(\xi_k|d_k, \Omega_{\rm int})p(d_k|\Omega_{\rm int})\frac{p(\xi_k|\Omega)}{p(\xi_k|\Omega_{\rm int})}p(\Omega|\{d\}) \quad \text{(C18)}$$

$$\propto p(\xi_k|\, d_k, \Omega_{\rm int})\frac{1}{N}\sum_{\Omega\sim p(\Omega|\{d\})} \frac{p(\xi_k|\Omega)}{p(\xi_k|\Omega_{\rm int})}, \quad \text{(C19)}$$

where in the final step we have drawn samples over $p(\Omega|\{d\})$ N times and dropped the normalizing constant $p(d_k|\Omega_{\rm int})$.

ORCID iDs

Sebastian Wagner-Carena https://orcid.org/0000-0001-5039-1685
Ji Won Park https://orcid.org/0000-0002-0692-1092
Simon Birrer https://orcid.org/0000-0003-3195-5507
Aaron Roodman https://orcid.org/0000-0001-5326-3486
Risa H. Wechsler https://orcid.org/0000-0003-2229-011X

[17] To avoid this approximation, we would have to recalculate $p(\Omega|\{d\}_{\neq k})$ for each lens. In the case where we have few lenses, the approximation begins to break down, but $p(\Omega|\{d\}_{\neq k})$ becomes much easier to calculate.

References

Abadi, M., Agarwal, A., Barham, P., et al. 2016, arXiv:1603.04467
Barkana, R. 1998, ApJ, 502, 531
Birrer, S., & Amara, A. 2018, PDU, 22, 189
Birrer, S., Refregier, A., & Amara, A. 2018, ApJL, 852, L14
Birrer, S., Shajib, A., Galan, A., et al. 2020, A&A, 643, A165
Birrer, S., & Treu, T. 2020, arXiv:2008.06157
Blandford, R., & Narayan, R. 1992, ARA&A, 30, 311
Bolton, A. S., Burles, S., Koopmans, L. V., Treu, T., & Moustakas, L. A. 2006, ApJ, 638, 703
Bolton, A. S., Treu, T., Koopmans, L. V., et al. 2008, ApJ, 684, 248
Brehmer, J., Mishra-Sharma, S., Hermans, J., Louppe, G., & Cranmer, K. 2019, ApJ, 886, 49
Charnock, T., Perreault-Levasseur, L., & Lanusse, F. 2020, arXiv:2006.01490
Chianese, M., Coogan, A., Hofma, P., Otten, S., & Weniger, C. 2020, MNRAS, 496, 381
Collett, T. E. 2015, ApJ, 811, 20
Diaz Rivero, A., & Dvorkin, C. 2020, PhRvD, 101, 023515
Ding, X., Treu, T., Shajib, A. J., et al. 2018, arXiv:1801.01506
Dressel, L. 2019, Wide Field Camera 3 Instrument Handbook, Version 12.0 (Baltimore, MD: STScI)
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
Freedman, W. L., Madore, B. F., Hatt, D., et al. 2019, ApJ, 882, 34
Freedman, W. L., Madore, B. F., Hoyt, T., et al. 2020, ApJ, 891, 57
Gal, Y., & Ghahramani, Z. 2015, arXiv:1506.02158
Gal, Y., & Ghahramani, Z. 2016, in ICML 33, Int. Conf. on Machine Learning, ed. M. Balcan & K. Q. Weinberger (New York: JMLR), 1050
Gal, Y., Hron, J., & Kendall, A. 2017a, in Advances in Neural Information Processing Systems, ed. I. Guyon et al., Vol. 30 (Red Hook, NY: Curran Associates, Inc.), 3581
Gal, Y., Islam, R., & Ghahramani, Z. 2017b, arXiv:1703.02910
Giavalisco, M., Sahu, K., & Bohlin, R. C. 2002, New Estimates of the Sky Background for the HST Exposure Time Calculator (Baltimore, MD: STScI)
Goodman, J., & Weare, J. 2010, Comm. App. Math. Comp. Sci., 5, 65
Greenberg, D. S., Nonnenmacher, M., & Macke, J. H. 2019, arXiv:1905.07488
Hogg, D. W., Myers, A. D., & Bovy, J. 2010, ApJ, 725, 2166
Hortúa, H. J., Malagò, L., & Volpi, R. 2020a, Mach. Learn.: Sci. Technol., 1, 035014
Hortúa, H. J., Volpi, R., Marinelli, D., & Malagò, L. 2020b, PhRvD, 102, 103509
Kampffmeyer, M., Salberg, A.-B., & Jenssen, R. 2016, in IEEE Conf. on Computer Vision and Pattern Recognition Workshops, ed. S. Lazebnik & A. Davison (Las Vegas, NV: CVPRW), 1
Kendall, A., & Gal, Y. 2017, in Advances in Neural Information Processing Systems, Vol. 30, ed. I. Guyon et al. (Red Hook, NY: Curran Associates, Inc.), 5574
Koopmans, L. V., Treu, T., Bolton, A. S., Burles, S., & Moustakas, L. A. 2006, ApJ, 649, 599
Kormann, R., Schneider, P., & Bartelmann, M. 1994, A&A, 284, 285
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems, Vol. 25, ed. F. Pereira et al. (Red Hook, NY: Curran Associates, Inc.), 1097
Leibig, C., Allken, V., Ayhan, M. S., Berens, P., & Wahl, S. 2017, NatSR, 7, 1
Oguri, M., & Marshall, P. J. 2010, MNRAS, 405, 2579
Ostdiek, B., Rivero, A. D., & Dvorkin, C. 2020a, arXiv:2009.06663
Ostdiek, B., Rivero, A. D., & Dvorkin, C. 2020b, arXiv:2009.06639
Park, J. W., Wagner-Carena, S., Marshall, P. J., Birrer, S., & Roodman, A. 2020, arXiv:2012.00042
Perreault Levasseur, L., Hezaveh, Y. D., & Wechsler, R. H. 2017, ApJL, 850, L7
Planck Collaboration 2020, A&A, 641, A6
Riess, A. G., Casertano, S., Yuan, W., Macri, L. M., & Scolnic, D. 2019, ApJ, 876, 85
Schuldt, S., Suyu, S., Meinhardt, T., et al. 2021, A&A, 646, A126
Shajib, A., Birrer, S., Treu, T., et al. 2019, MNRAS, 483, 5649
Shajib, A., Birrer, S., Treu, T., et al. 2020, MNRAS, 494, 6072
Shajib, A. J., Treu, T., & Agnello, A. 2018, MNRAS, 473, 210
Sonnenfeld, A., Gavazzi, R., Suyu, S. H., Treu, T., & Marshall, P. J. 2013, ApJ, 777, 97
Sonnenfeld, A., Treu, T., Marshall, P. J., et al. 2015, ApJ, 800, 94
Suyu, S., Auger, M., Hilbert, S., et al. 2013, ApJ, 766, 70
Treu, T. 2010, ARA&A, 48, 87
Treu, T., & Koopmans, L. V. 2004, ApJ, 611, 739
Verde, L., Treu, T., & Riess, A. 2019, NatAs, 3, 891
Wong, K. C., Suyu, S. H., Chen, G. C.-F., et al. 2019, MNRAS, 498, 1420