Fast Estimation of Physical Galaxy Properties using Simulation-Based Inference

Maxime Robeyns [*1] **Sotiria Fotopolou** [2] **Mike Walmsley** [3] **Laurence Aitchison** [1]

Abstract

Astrophysical surveys present the challenge of scaling up accurate simulation based inference to billions of different examples. We develop a method to train fast, accurate and amortised approximate posteriors that avoids the biases of e.g. variational inference. To train our approximate posterior, we first sample from it, conditioned on an observation. We then do a few steps of an MCMC method (we use HMC) to improve the sample, and we update the approximate posterior parameters to maximize the probability of the resulting MCMC samples. This allows us to amortise the posterior implied by any MCMC procedure. On our astrophysical samples, the amortised approximate posterior is very close to the true MCMC posterior, yet is approximately five orders of magnitude faster.

We additionally provide a library to facilitate the use of this method for upcoming surveys: https://github.com/MaximeRobeyns/SPItorch.

1. Introduction

Inferring galaxy parameters is an important task for extragalactic surveys. For example, physical parameter estimation has led to significant discoveries in the domain of galaxy evolution such as the cosmic star-formation history (Madau et al., 1998; Madau & Dickinson, 2014; Elbaz et al., 2007).

As astronomical surveys become larger, the need for computationally-efficient methods for inferring physical properties underlying these observations becomes pressing. For instance the Dark Energy Spectroscopic Instrument survey (Aghamousa, 2016) will produce tens of millions of observations, the Euclid survey (Racca, 2016) is projected to capture about 10 billion sources, while the Vera C. Rubin Observatory is projected to capture tens of billions of sources annually (Ivezić, 2018).

Current approaches to inferring physical galaxy parameters often involve creating a forward model for galaxy emissions, and using Markov-Chain Monte Carlo (MCMC) methods (MAGPHYS, CIGALE, PROSPECTOR, AGNFitter, Fortesfit) to invert that forward model. In this approach, the model parameter space is probed on the fly during sampling, and incur a computational cost that renders them infeasible for use in the large-scale surveys described above.

Recently, the use of simulation-based inference (SBI) has enabled drastically faster parameter inference over current MCMC approaches (Zhang et al., 2021; Hahn & Melchior, 2022). Existing forward-modelling libraries and software for creating spectral energy distribution (SED) models can readily be applied to SBI as an *implicit* statistical model; that is, a model from which we may draw samples but cannot evaluate probability densities.

We take inspiration from both synthetic likelihood and posterior density estimation methods to develop a drop-in replacement for current photometry MCMC approaches which is thousands of times faster, while providing equal or better posterior estimates for normal galaxies and AGN.

1.1. Method

We use PROSPECTOR to define an SED forward model describing the relation between physical galaxy parameters and the photometric observations from a given survey. This mapping from physical parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ to Gaussian-noise corrupted photometric observations $\mathbf{x} \in \mathbb{R}^n$ results in our implicit likelihood $p(\mathbf{x}|\boldsymbol{\theta})$ (or *simulator*).

Using a suitable parameter prior $p(\boldsymbol{\theta})$, we simulate a training dataset $\{(\boldsymbol{\theta}_i, \mathbf{x}_i)\}_{i=1}^N$ of samples from the joint $p(\boldsymbol{\theta}, \mathbf{x})$; first sampling from the prior $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta})$ and then running the forward model $\mathbf{x}_i \sim p(\mathbf{x}|\boldsymbol{\theta}_i)$. This is readily parallelisable, runs in an offline manner, and even for $N = 1e7$ represents a small computational cost; completing in under an hour.

We use this dataset to learn two densities, using conditional

[*]Equal contribution [1]Department of Computer Science, University of Bristol, UK [2]HH Wills Physics Laboratory, University of Bristol, Bristol, UK [3]Jodrell Bank Centre for Astrophysics, Department of Physics & Astronomy, University of Manchester, Oxford Road, Manchester M13 9PL, UK. Correspondence to: Maxime Robeyns <ez18285@bristol.ac.uk>.

ICML 2022 Workshop on Machine Learning for Astrophysics, Baltimore, Maryland, USA, 2022.

neural density estimation. The neural density estimator used for both of these models is a *sequential autoregressive network*, which we introduce in the next section.

The first density is a crude approximation to the posterior, $q_\phi(\boldsymbol{\theta}|\mathbf{x}) \approx p(\boldsymbol{\theta}|\mathbf{x})$, where the parameters of the density estimator ϕ are optimised by maximising the likelihood of physical parameters drawn from the prior under simulated observations:

$$\phi \doteq \arg\max_{\phi} \sum_{i=1}^{N} \log q_\phi(\boldsymbol{\theta}_i|\mathbf{x}_i), \tag{1}$$

using the ADAM optimiser. Direct posterior estimation is notoriously difficult since there may only be a sparse set of simulated training points $(\boldsymbol{\theta}_i, \mathbf{x}_i)$ in the vicinity of a given real observation $\mathbf{x}_o$ (Papamakarios et al., 2019). The aim of this step is to simply initialise the parameters ϕ of this approximate posterior to a sensible value, which will help during a subsequent HMC training step.

Drawing inspiration from Wiqvist et al. (2021); Glöckler et al. (2022), we additionally use this simulated dataset to obtain an approximation to the intractable likelihood $\ell_\varphi(\mathbf{x}|\boldsymbol{\theta}) \approx p(\mathbf{x}|\boldsymbol{\theta})$, which we also train using maximum likelihood

$$\varphi \doteq \arg\max_{\varphi} \sum_{i=1}^{N} \log \ell_\varphi(\mathbf{x}_i|\boldsymbol{\theta}_i). \tag{2}$$

Approximating the likelihood with a neural density estimator affords us a fast, differentiable and reasonably accurate simulator (see Figure 2 for an evaluation of accuracy), which can moreover exploit GPU parallelism, making it suitable for use in the inner loop of a training algorithm. Note that the likelihood is often much simpler to estimate than the posterior, since the conditioning information $\boldsymbol{\theta}$ is noise-free and lies on a bounded domain (as defined by $p(\boldsymbol{\theta})$), in contrast to the noise-corrupted observations $\mathbf{x}$ used in posterior estimation.

Training both these density estimators for 10 epochs took under 2 hours using commodity hardware (`NVIDIA RTX 3090`). This is comparable to running MCMC inference on tens of galaxies.

A common challenge in simulation-based inference is the use of an inaccurate or misspecified model. To account for possible discrepancies between the SED model and the observational data, as well as the potential sparsity of training examples in the vicinity of observations, we introduce an additional training step for the parameters ϕ of the neural posterior. In this procedure, we perform further maximum-likelihood updates to ϕ on new $(\hat{\boldsymbol{\theta}}_{\text{HMC}}, \mathbf{x}_o)$ pairs generated on-the-fly from a subset of the photometric observations $\mathbf{x}_o$ and parameter estimates $\hat{\boldsymbol{\theta}}$ which are obtained via HMC. The target density for HMC uses the neural likelihood $p(\hat{\boldsymbol{\theta}}|\hat{\mathbf{x}}) \propto \ell_\varphi(\hat{\mathbf{x}}|\hat{\boldsymbol{\theta}})p(\hat{\boldsymbol{\theta}})$, and we initialise the HMC chains at an initial prediction $\hat{\boldsymbol{\theta}}_{\text{q}} \sim q_\phi(\boldsymbol{\theta}|\mathbf{x}_o)$, reducing the need for burn-in steps. The target density is fully differentiable with respect to $\boldsymbol{\theta}$, and using deep learning tools (PyTorch) we parallelise the HMC procedure to work on large $\boldsymbol{\theta}$ batches, further benefiting from GPU compute.

The full method is given in Algorithm 1, returning a neural density estimator q_ϕ to perform amortised posterior inference over the physical parameters for a specific survey.

Algorithm 1 Training procedure.

Input: PROSPECTOR forward model $p(\mathbf{x}|\boldsymbol{\theta})$, parameter prior $p(\boldsymbol{\theta})$, observed data $\mathbf{X}_{\text{o}}$.
Output: Approximate posterior $q_\phi(\boldsymbol{\theta}|\mathbf{x})$, for $\mathbf{X}_{\text{o}}$.

Simulate $\{(\boldsymbol{\theta}_i, \mathbf{x}_i)\}_{i=1}^N$, where $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta})$, $\mathbf{x}_i \sim p(\mathbf{x}|\boldsymbol{\theta}_i)$.
Train approximate posterior using maximum-likelihood: $\arg\max_{\boldsymbol{\phi}} \sum_{i=1}^N \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}_i|\mathbf{x}_i)$.
Train a neural likelihood using maximum-likelihood: $\arg\max_{\varphi} \sum_{i=1}^N \log \ell_\varphi(\mathbf{x}_i|\boldsymbol{\theta}_i)$.
for each *mini-batch of* $\mathbf{x}_o$ *in* $\mathbf{X}_o$ **do**
  Draw $\hat{\boldsymbol{\theta}}_{\text{q}} \sim q_\phi(\boldsymbol{\theta}|\mathbf{x}_o)$
  $\hat{\boldsymbol{\theta}}_{\text{HMC}} = \text{HMC}(\text{init} = \hat{\boldsymbol{\theta}}_{\text{q}}, \text{target dist} = \ell_\varphi(\mathbf{x}_o|\boldsymbol{\theta})p(\boldsymbol{\theta}))$
  Train q_ϕ on $\hat{\boldsymbol{\theta}}_{\text{HMC}}$; $\phi \leftarrow \phi + \alpha\nabla_\phi q_\phi(\hat{\boldsymbol{\theta}}_{\text{HMC}}|\mathbf{x}_o)$
return q_ϕ

1.2. Sequential Autoregressive Network

We draw inspiration from both mixture density networks (Bishop, 1994) and autoregressive models (Germain et al., 2015) to obtain a simple conditional neural density estimator which is both fast and expressive.

Autoregressive networks for multivariate density estimation work by factorising the distribution as a product of nested conditionals,

$$\begin{aligned} q(\boldsymbol{\theta}|\mathbf{x}) &= \prod_{d=1}^{D} q(\theta_d|\boldsymbol{\theta}_{<d}, \mathbf{x}) \\ &= q(\theta_1|\mathbf{x})q(\theta_2|\theta_1, \mathbf{x}) \cdots q(\theta_D|\theta_{d-1}, \ldots, \theta_1, \mathbf{x}), \end{aligned}$$

thus avoiding the need to compute a potentially expensive (or even intractable) normalisation constant.

In this work we treat the distribution of each dimension of $\boldsymbol{\theta}$, $q(\theta_d|\theta_{<d}, \mathbf{x})$ as a mixture density network. We additionally condition each dimension of the posterior on latent variables $\boldsymbol{z}_d$ (termed '*sequence features*') which govern the relationship between subsequent dimensions of $\boldsymbol{\theta}$. The posterior model factorises as

$$q_\phi(\boldsymbol{\theta}|\mathbf{x}) = \prod_{d=1}^{D} q_\phi^d(\theta_d|\boldsymbol{\theta}_{<d}, \boldsymbol{z}_{d-1}, \mathbf{x}), \tag{3}$$

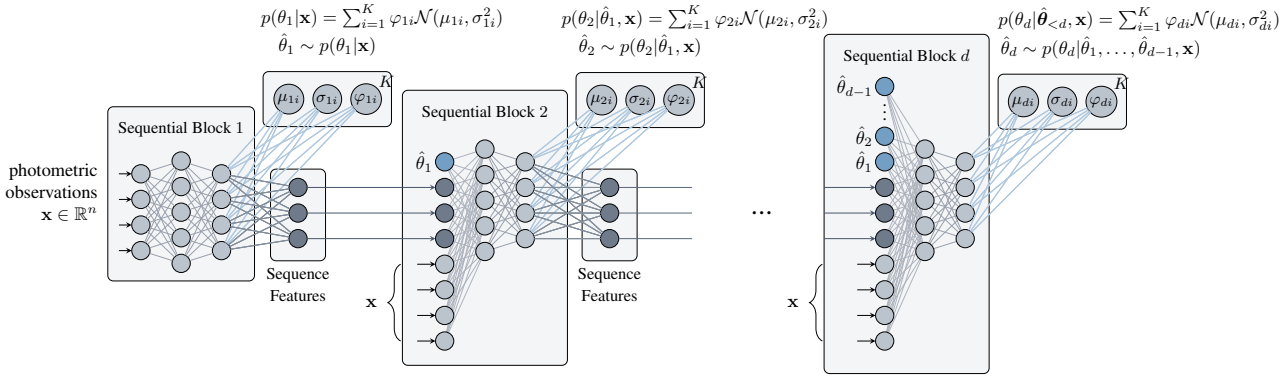

Figure 1. Sequential autoregressive network architecture, with n-dimensional conditioning information $\mathbf{x}$, and d-dimensional outputs $\boldsymbol{\theta}$. Each output dimension is modelled by a (truncated) mixture of K Gaussians.

and can be seen as an autoregressive sequence of mixture density networks with auxiliary latent variables. This architecture is visualised in Figure 1.2.

1.2.1. Alternative Approaches

Recent papers using conditional generative models for accelerated posterior estimation have primarily used masked autoregressive flows (MAFs) as the neural density estimator (Zhang et al., 2021; Hahn & Melchior, 2022).

However, MAFs trade-off fast likelihood evaluations for slow sampling time. While evaluations only require a single forward pass through the MAF owing to the use of MADE blocks (Germain et al., 2015), the autoregressive structure means that drawing parameter samples from the posterior at inference time $\boldsymbol{\theta} \sim q_\phi(\boldsymbol{\theta}|\mathbf{x}), \boldsymbol{\theta} \in \mathbb{R}^d$, requires d forward passes through the network.

While the SAN architecture also uses autoregressive sampling (unrolling the d forward passes into a single forward pass), the cost of autoregressive sampling in MAFs is exacerbated due to the constraints of normalising flows. Here, the Jacobian log-determinant of each layer must remain easy to compute, limiting each layer's flexibility and resulting in deeper architectures to retain the same expressiveness as an unconstrained network.

Hence, sampling from a SAN requires fewer layers, less memory and is empirically faster (see Table 1.2.1), making it better suited to performing inference on billions of galaxies. We also found MAFs to be highly sensitive to the choice of hyperparameters, while the SAN was far more robust.

2. Results

To evaluate our approach, we use a subset of the Dark Energy Survey (DES, 2005) as the real dataset $\mathbf{X}_o$. We use

Method	Device	Amortised Single Sample Time (s)
EMCEE	CPU	4.82×10^{-1}
MAF	CPU	2.52×10^{-4}
MAF	GPU	7.31×10^{-5}
SAN	CPU	5.73×10^{-5}
SAN	GPU	$\mathbf{1.52 \times 10^{-6}}$

Table 1. Time to draw a single posterior sample: EMCEE is a popular sample-based (MCMC) method, MAF is a masked autoregressive flow, and SAN is our proposed neural density estimator.

PROSPECTOR for our forward model with realistic magnitude distributions and uncertainties, using both stellar and AGN components. Using a dataset of 10 million simulated observations, $\mathcal{D} = \{(\boldsymbol{\theta}_i, \mathbf{x}_i)\}_{i=1}^{1e7}$, we follow the training procedure described in Algorithm 1. We describe our full model architecture and hyperparameters in the supplementary material (Appendix A).

The success of our method is contingent on having accurate likelihood model. The top-left panel of Figure 2 shows a probability-probability plot for the neural likelihood $\ell_\varphi(\boldsymbol{x}|\boldsymbol{\theta})$ for all 7 filter bands, showing good agreement between the true forward model and the learned likelihood.

In the top right pane we show a probability-probability plot for the neural posterior, using 10,000 simulated points. There is a reasonable agreement between the true CDF and that of the approximate posterior for all parameters.

We also show SED reconstructions for simulated ($\mathbf{x}$) and real ($\mathbf{x}_o$) observations in the bottom two panes of Figure 2. We find that our posteriors are estimated as well or better than EMCEE, in 5 orders of magnitude less time (reducing posterior sampling times for 10,000 samples from hundreds of seconds to thousandths of a second).

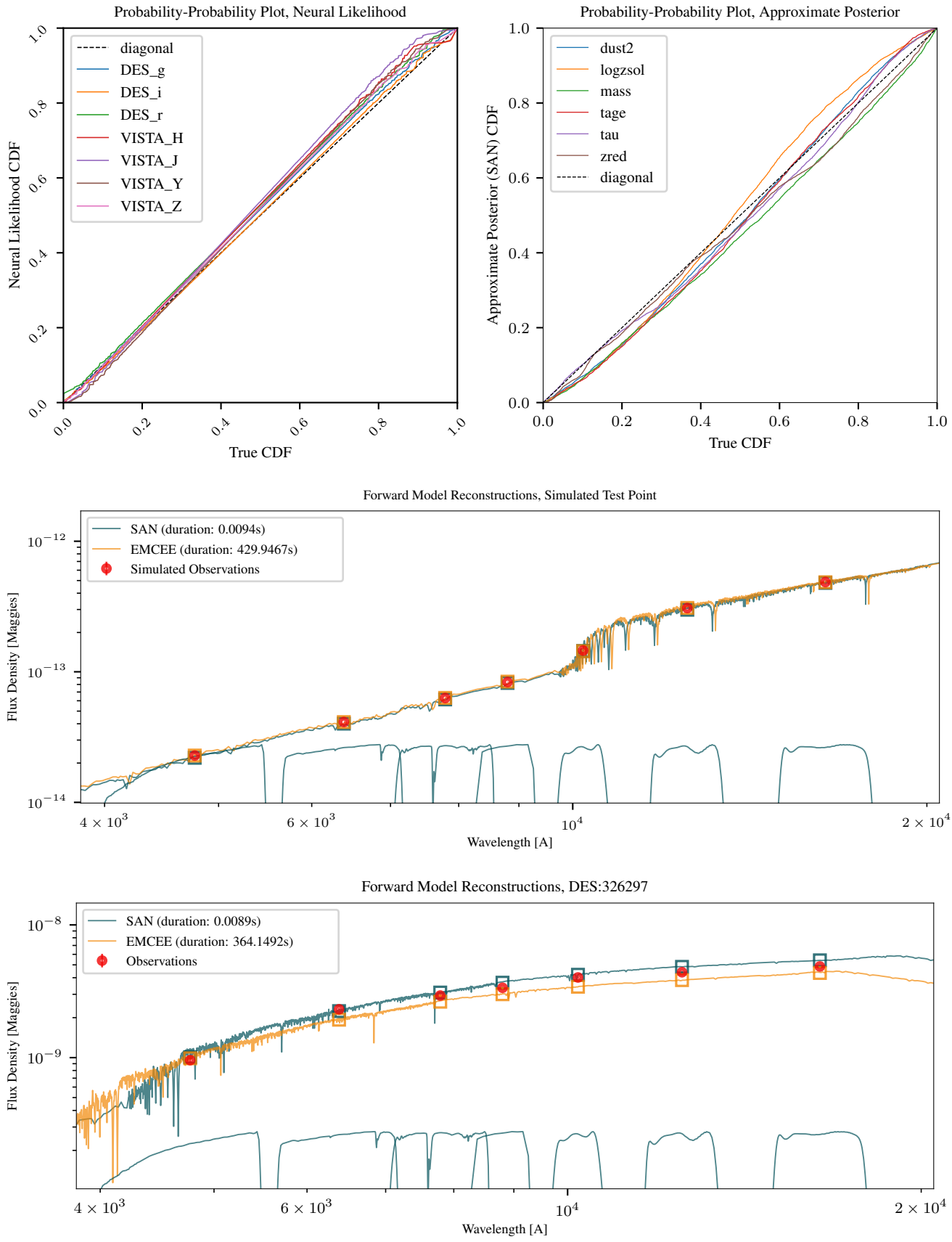

Figure 2. Evaluation Plots. Top: likelihood and posterior evaluation, mid: forward model reconstruction for an in-distribution, simulated test point, bottom: forward model reconstructions for a real, out-of-distribution test point.

Acknowledgements

We thank the ACRC at the University of Bristol for computing resources.

References

Aghamousa. The DESI Experiment Part I: Science,Targeting, and Survey Design. *arXiv:1611.00036 [astro-ph]*, December 2016. URL http://arxiv.org/abs/1611.00036. arXiv: 1611.00036.

Bishop, C. M. Mixture density networks. *Mixture density networks*, 1994. ISSN NCRG/94/004. Place: Birmingham Publisher: Aston University.

Collaboration, T. D. E. S. The Dark Energy Survey. *arXiv:astro-ph/0510346*, October 2005. URL http://arxiv.org/abs/astro-ph/0510346. arXiv: astro-ph/0510346.

Elbaz, D., Daddi, E., Borgne, D. L., Dickinson, M., Alexander, D. M., Chary, R.-R., Starck, J.-L., Brandt, W. N., Kitzbichler, M., MacDonald, E., Nonino, M., Popesso, P., Stern, D., and Vanzella, E. The reversal of the star formation-density relation in the distant universe. *Astronomy & Astrophysics*, 468(1):33–48, June 2007. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361:20077525. URL https://www.aanda.org/articles/aa/abs/2007/22/aa7525-07/aa7525-07.html. Number: 1 Publisher: EDP Sciences.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. MADE: Masked Autoencoder for Distribution Estimation. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 881–889. PMLR, June 2015. URL https://proceedings.mlr.press/v37/germain15.html. ISSN: 1938-7228.

Glöckler, M., Deistler, M., and Macke, J. H. Variational methods for simulation-based inference. In *The Tenth International Conference on Learning Representations*, March 2022. URL https://openreview.net/pdf?id=kZ0UYdhqkNY. arXiv: 2203.04176.

Hahn, C. and Melchior, P. Accelerated Bayesian SED Modeling using Amortized Neural Posterior Estimation. *arXiv:2203.07391 [astro-ph, stat]*, March 2022. URL http://arxiv.org/abs/2203.07391. arXiv: 2203.07391.

Ivezić. LSST: from Science Drivers to Reference Design and Anticipated Data Products. *arXiv:0805.2366 [astro-ph]*, May 2018. doi: 10.3847/1538-4357/ab042c. URL http://arxiv.org/abs/0805.2366. arXiv: 0805.2366.

Madau, P. and Dickinson, M. Cosmic star-formation history. *Annual Review of Astronomy and Astrophysics*, 52(1):415–486, 2014. doi: 10.1146/annurev-astro-081811-125615. URL https://doi.org/10.1146/annurev-astro-081811-125615.

Madau, P., Pozzetti, L., and Dickinson, M. The Star Formation History of Field Galaxies. *The Astrophysical Journal*, 498:106–116, May 1998. ISSN 0004-637X. doi: 10.1086/305523. URL https://ui.adsabs.harvard.edu/abs/1998ApJ...498..106M. ADS Bibcode: 1998ApJ...498..106M.

Papamakarios, G., Sterratt, D., and Murray, I. Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 837–848. PMLR, April 2019. URL https://proceedings.mlr.press/v89/papamakarios19a.html. ISSN: 2640-3498.

Racca. The Euclid mission design. *arXiv:1610.05508 [astro-ph]*, pp. 99040O, July 2016. doi: 10.1117/12.2230762. URL http://arxiv.org/abs/1610.05508. arXiv: 1610.05508.

Wiqvist, S., Frellsen, J., and Picchini, U. Sequential Neural Posterior and Likelihood Approximation. *arXiv:2102.06522 [cs, stat]*, June 2021. URL http://arxiv.org/abs/2102.06522. arXiv: 2102.06522.

Zhang, K., Bloom, J. S., Gaudi, B. S., Lanusse, F., Lam, C., and Lu, J. R. Real-time Likelihood-free Inference of Roman Binary Microlensing Events with Amortized Neural Posterior Estimation. *The Astronomical Journal*, 161(6):262, May 2021. ISSN 1538-3881. doi: 10.3847/1538-3881/abf42e. URL https://doi.org/10.3847/1538-3881/abf42e. Publisher: American Astronomical Society.

A. Model and Hyperparameters

Here we describe the network architecture and model hyperparameters used in our experiments.

For the approximate posterior $q_\phi(\boldsymbol{\theta}|\mathbf{x})$, we use sequential blocks with 2 layers, each with 1024 neurons and ReLU activations. These sequential blocks parametrise a mixture of 10 truncated Gaussians, the boundaries of which ensure that any $\hat{\boldsymbol{\theta}}$ proposals will lie within the support of the prior, which is important for the HMC step[1]. Latent variables ('sequence features') with 16 dimensions are passed between blocks.

The nerual likelihood $\ell_\varphi(\mathbf{x}|\boldsymbol{\theta})$ is both simpler to estimate and will also be evaluated far more often in the highly parallelised HMC step. We therefore choose a smaller architecture with sequential blocks of width 256 to the benefit of both speed and memory requirements. We also apply layer normalisation and use ReLU activations. The marginals are modelled as a mixture of just 3 (non-truncated) Gaussians, and we pass just 8 sequence features between blocks.

We use a simple normalisation scheme where filter values $\mathbf{x}$ are prepared by merely taking their logarithm—any features which are useful for inference, such as colours, may be inferred by the flexible SAN network architecture. We normalise the parameter values $\boldsymbol{\theta}$ to lie uniformly in the $[0, 1]$ range.

[1] We also tried using a mixture of Beta distribution to enforce boundary conditions. While the quality of the posterior distributions was good, evaluating a Beta distribution has a greater computational cost and hence yielded longer running times than the truncated mixture of Gaussians.

B. Posterior Plot

Figure 3 shows 10,000 samples drawn from the approximate posterior $\hat{\boldsymbol{\theta}} \sim q_\phi(\boldsymbol{\theta}|\boldsymbol{x}_\mathrm{o})$, for a randomly selected observation $\mathbf{x}_\mathrm{o}$ drawn from the simulated dataset.

The true physical parameters $\boldsymbol{\theta}$ used to produce the simulated observation $\mathbf{x}_\mathrm{o}$ are plotted in orange.

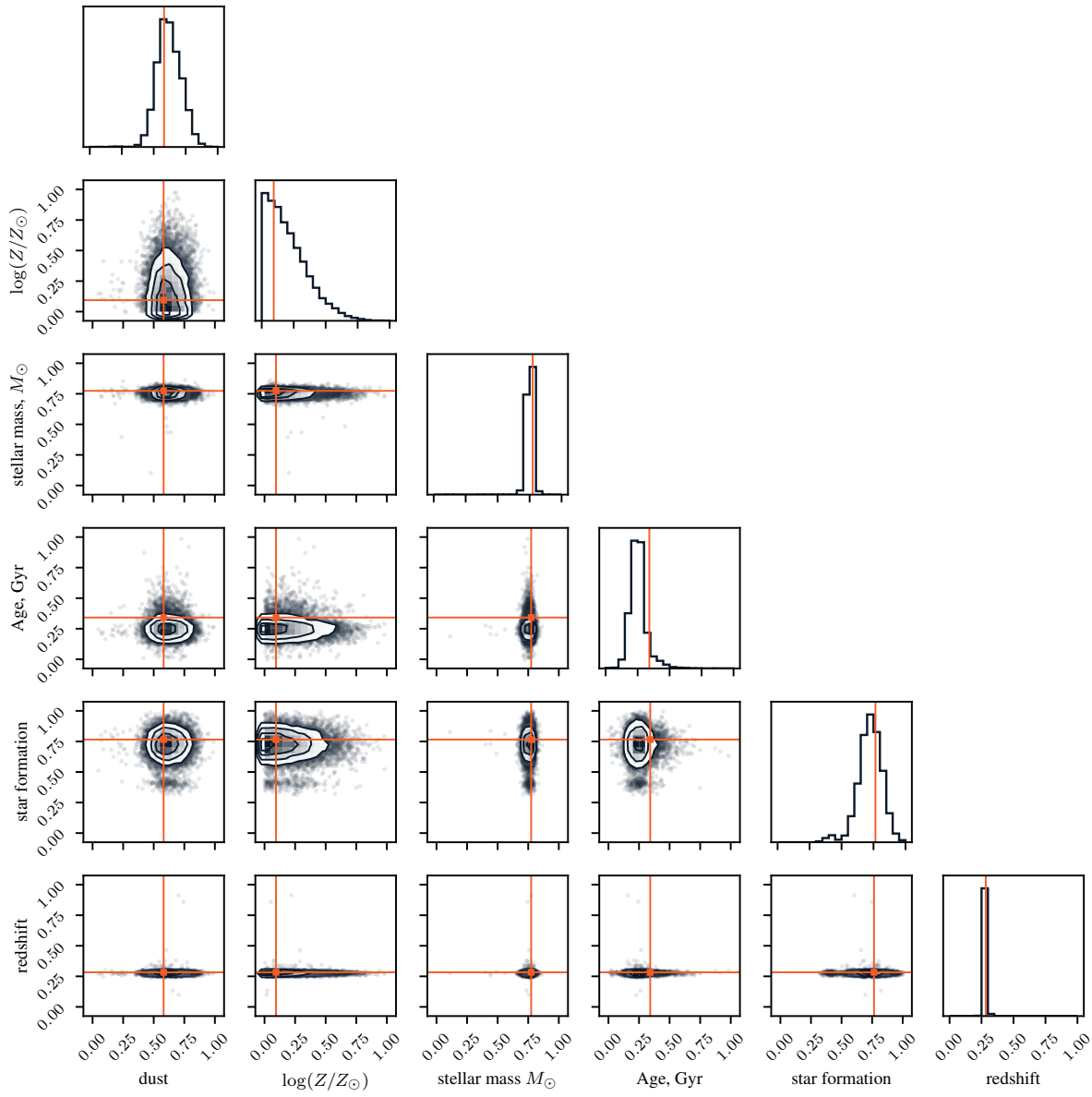

Figure 3. Corner plot showing samples from the (normalised) SAN approximate posteriors for a simulated test point. The true parameter values are indicated in orange.

C. Updated Model

Following acceptance to this ICML workshop, we modified our neural density estimator (SAN) to make more efficient use of computation while reducing the number of parameters, thus improving computation time.

In the original SAN, the later dimensions would benefit from a deeper network and thus potentially better representations,

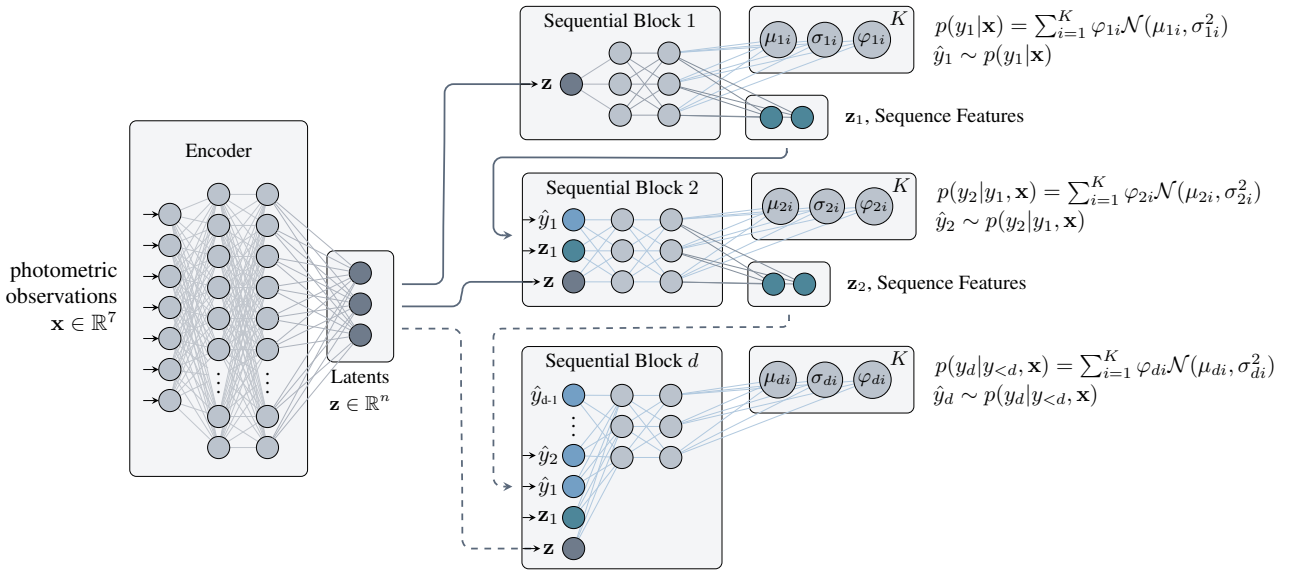

Figure 4. Updated network architecture for our neural density estimator.

while the first few marginals would be occasioned far less computation.

To address this, we introduced a single large encoder block E, which deterministically maps photometric observations to a low-dimensional latent feature vector $E : \mathbf{x} \to \mathbf{z}$. This feature vector is then provided as the conditioning information to a (now smaller) SAN; thus allowing us to re-use computation when generating each marginal. This has the additional benefit that the order of the marginals is now a far less important consideration.

The modified network architecture is visualised in Figure 4, which we use with the following model parameters:

parameter	likelihood	posterior
encoder shape	`[512,]`	`[7000, 7000]`
SAN module shape	`[128,]`	`[2000,]`
activation	`ReLU`	`ReLU`
latent features	25	100
sequence features	2	8
batch size	2048	5000
mixture components, K	3	5
Adam learning rate	`3e-3`	`7e-4`
Adam decay	`1e-4`	`1e-4`
normalisation	`layer norm`	`layer norm`

This updated architecture allows us to acheive better approximations to the likelihood and posterior. The result of the full training procedure (including the HMC update step) is shown in Figure 5, and represents a tangible improvement over the original SAN architecture. This model is available in the accompanying codebase.

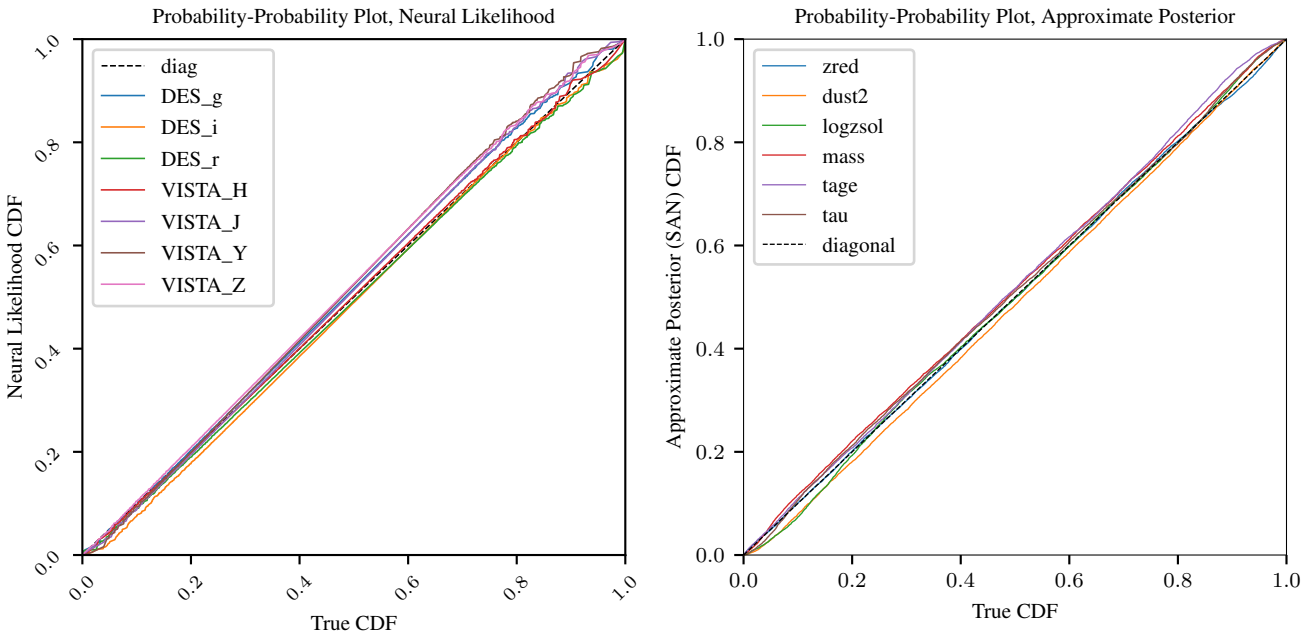

Figure 5. Evaluation Plots for the updated SAN architecture.