

# Normalizing flows for likelihood-free inference with fusion simulations

C. S. Furia<sup>1</sup>, R. M. Churchill<sup>2</sup>

<sup>1</sup> Rutgers University

<sup>2</sup> Princeton Plasma Physics Laboratory

E-mail: [chirag.furia@rutgers.edu](mailto:chirag.furia@rutgers.edu)

February 2022

**Abstract.** Fluid based scrape-off layer transport codes such as UEDGE are heavily utilized in tokamak analysis and design, but typically require user-specified anomalous transport coefficients to match experiment. Determining uniqueness of these parameters and the uncertainties in them to match experiments can provide valuable insights to fusion scientists. We leverage recent work in the area of likelihood-free inference (“simulation-based inference”) to train a neural network which enables accurate statistical inference of the anomalous transport coefficients given experimental plasma profile input. UEDGE is treated as a black-box simulator and run multiple times with anomalous transport coefficients sampled from priors, and the neural network is trained on these simulations to emulate the posterior. The neural network is trained as a normalizing flow model for density estimation, allowing it to accurately represent complicated, high-dimensional distribution functions. With a fixed simulation budget, we compare a single-round procedure to a multi-round approach that guides the training simulations toward a specific target observation. We discuss the future possibilities for use of amortized models which train on a wide range of simulations and enable fast statistical inference for results during experiments.

Submitted to: *Plasma Phys. Control. Fusion*

## 1. Introduction

The scrape-off layer (SOL) is the outermost region of a confined plasma in a tokamak, serving as a buffer between the fusion-harboring main plasma and the solid device surface. Modeling the SOL is not a straightforward process, as it requires multi-physics processes such as turbulence, neutral transport, and material interactions as well as analysis of the Debye sheath that lies just above divertor targets or limiters. Despite being relatively thin (on the order of  $\approx 1$  cm), it is central to many outstanding challenges in tokamak design, such as limiting damaging heat flux and erosion of material walls, and preventing contamination of the main plasma with neutrals and impurities.

Therefore, understanding the SOL and delving deeper into its underlying physics is key to sustaining progress towards viable tokamak-driven fusion energy sources.

2D fluid codes such as UEDGE [1] are commonly used to simulate edge transport in tokamaks, encapsulating the SOL and also parts of the confined plasma inside the Last Closed Flux Surface (LCFS). Plasma state variables such as electron temperature ( $T_e$ ) and ion temperature ( $T_i$ ) are first initialized over a mesh of poloidal and radial grid points, then evolved based on equations for particle, momentum, ion energy, and electron energy balance until the system converges to equilibrium. In addition to being dependent on boundary conditions and magnetic geometries, these simulations are also influenced by user-defined anomalous, radial transport coefficients in the fluid equations that mimic the effects of turbulent transport. Currently, these coefficients are determined by ad-hoc tuning and algorithmic searches to yield simulations that match experiment, but this process can be inefficient and also ignores the possibility of multiple solutions existing over the space of the anomalous transport coefficients.

Recent efforts have been made to reduce the uncertainty associated with finding anomalous transport coefficients, with notable examples being the autoUEDGE package [2] which iteratively adjusts the transport coefficients to match experimental plasma profiles, and also work by Carli et al. [3] using Bayesian inference and maximum-a-posteriori (MAP) estimation on profiles from SOLPS. However, there are still improvements to be made, such as mapping the full probability distributions of anomalous transport coefficients for a specific experimental profile. We introduce likelihood-free inference (also known as simulation-based inference) [4] with normalizing flows as a new approach to quantifying anomalous transport coefficients that merges machine learning and Bayesian principles. Future work in this area may lead to amortized models capable of quickly performing inference for multiple shots on a single tokamak, or potentially shots across a range of tokamaks.

Likelihood-free inference methods represent probability distributions using mathematical structures that can be trained using neural networks on datasets of simulations. In this way, they are able to learn the relationship between the inputs and outputs of natural processes modelled by complex, stochastic, and intractable simulations. Likelihood-free inference has already been successfully implemented in a variety of fields, including neuroscience and cosmology [5, 6, 7]. In this paper, we apply sequential neural posterior estimation (SNPE) in conjunction with neural spline flows (NSF) to perform statistical inference of anomalous transport coefficients for UEDGE simulations from synthetic experimental data, based on the DIII-D tokamak.

## 2. Likelihood-free inference

For a simulation that takes a set of inputs,  $\boldsymbol{\theta} \in \mathbb{R}^n$ , and returns a set of outputs,  $\mathbf{x} \in \mathbb{R}^m$ , Bayes' Theorem holds that  $p(\boldsymbol{\theta}|\mathbf{x}) \propto p(\mathbf{x}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta})$ . In this framework,  $p(\boldsymbol{\theta})$  is the prior while  $p(\mathbf{x}|\boldsymbol{\theta})$  is the likelihood and  $p(\boldsymbol{\theta}|\mathbf{x})$  is the posterior. In many cases, determining the posterior distribution  $p(\boldsymbol{\theta}|\mathbf{x}_0)$  for a target observation  $\mathbf{x}_0$  taken from

experiment is desirable and can provide insights into the uniqueness and sensitivity of results given specific input parameters into the simulator. However, the complexity of many simulations means that calculating the likelihood  $p(\mathbf{x}|\boldsymbol{\theta})$  is expensive or intractable to calculate directly, or by traditional approximate methods [8].

In this paper we use a likelihood-free inference method known as (Sequential) Neural Posterior Estimation, which does efficient probability density estimation (i.e. learns to predict  $p(\boldsymbol{\theta}|\mathbf{x})$ ) by learning from pairs of  $(\boldsymbol{\theta}, \mathbf{x})$  generated by the simulation, with input parameters  $\boldsymbol{\theta}$  drawn from the prior. The learning of a surrogate model  $q_{F(\mathbf{x},\phi)}$  for the complicated posterior distribution from simulation is made possible using normalizing flows implemented as a deep neural network.

### 2.1. Normalizing Flows

Normalizing flows allow for mapping a simple probability distribution to a more complicated one, as shown in Figure 1. More detailed information on normalizing flows can be found in various reviews [9, 10]. Formally, normalizing flows are diffeomorphisms, defining a transform  $T$  which is invertible and for which both its forward and inverse transforms  $T$  and  $T^{-1}$  are differentiable. Mathematically, the transform  $T$  operates on a  $D$ -dimensional vector  $\mathbf{u}$  sampled from a base probability distribution  $p_u$  to produce a  $D$ -dimensional vector  $\mathbf{z}$ :

$$\mathbf{z} = T(\mathbf{u}) \quad \text{with } \mathbf{u} \sim p_u(\mathbf{u}) \quad (1)$$

Due to the invertible and differentiable properties, along with both  $\mathbf{u}$  and  $\mathbf{z} \in \mathbb{R}^D$ , the probability distribution  $p_z(\mathbf{z})$  can then be calculated as a change of variable (using  $\mathbf{u} = T^{-1}(\mathbf{z})$ ):

$$p_z(\mathbf{z}) = p_u(T^{-1}(\mathbf{z})) |\det J_{T^{-1}}(\mathbf{z})| \quad (2)$$

with  $J_{T^{-1}}$  the Jacobian of the inverse transform  $T^{-1}$  with respect to  $\mathbf{z}$ . These transforms  $T$  are also composable, so that many such transforms can be chained together and maintain the necessary properties of invertibility and differentiability. Typically, the overall transform  $T$  is parameterized by a neural network (of various flavors), and  $p_u$  is chosen to be a multi-variate normal distribution. For our application of neural density estimation, we parameterize the inverse transform  $T^{-1}$  with the neural network, and sample from  $p_u(\mathbf{u})$ . As we are interested in conditional density estimation, the applicable form of Equation 2 becomes:

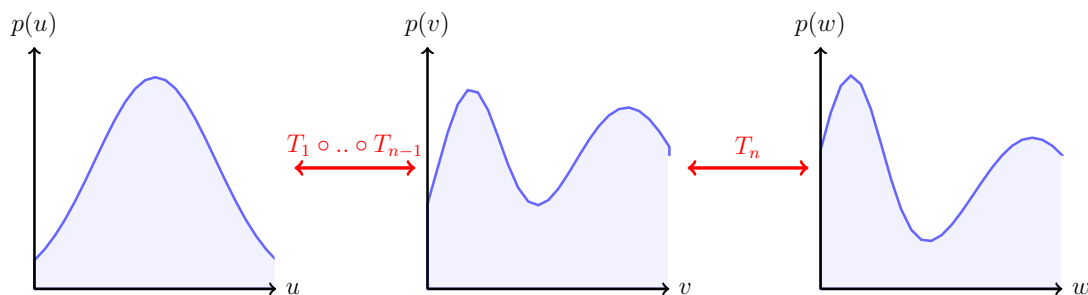
$$p(\boldsymbol{\theta}|\mathbf{x}) \approx q_{F(\mathbf{x},\phi)}(\boldsymbol{\theta}) = p_u(T_{\mathbf{x},\phi}^{-1}(\boldsymbol{\theta})) \left| \det J_{T_{\mathbf{x},\phi}^{-1}}(\boldsymbol{\theta}) \right| \quad (3)$$

To train the neural network, we minimize a metric of the statistical distance between the true distribution  $p(\boldsymbol{\theta}|\mathbf{x})$  and the neural network-based surrogate  $q_{F(\mathbf{x},\phi)}$  known as the Kullback-Leibler (KL) divergence, given by  $D_{KL}(p||q) = -\int p(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}$  for distributions  $p(\mathbf{x})$ ,  $q(\mathbf{x})$  [11, 12]. This results in minimizing the expectation over the

negative  $\log q_{F(\mathbf{x}, \phi)}$ , which given a set of samples  $(\boldsymbol{\theta}_j, \mathbf{x}_j)$  can be approximated using Monte Carlo with the estimation value denoted by  $\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{x})}$ :

$$\begin{aligned} \mathcal{L}(\phi) &= D_{KL} [p(\boldsymbol{\theta}|\mathbf{x}) || q_{F(\mathbf{x}, \phi)}(\boldsymbol{\theta})] \\ &= -\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{x})} [\log q_{F(\mathbf{x}, \phi)}(\boldsymbol{\theta})] + \text{const} \\ &\approx -\sum_{j=1}^N \left[ \log p_u(T_{\mathbf{x}_j, \phi}^{-1}(\boldsymbol{\theta}_j)) + \log \left| \det J_{T_{\mathbf{x}_j, \phi}^{-1}}(\boldsymbol{\theta}_j) \right| \right] + \text{const} \end{aligned} \quad (4)$$

Importantly, normalizing flows must not only be differentiable transforms, but also have a tractable Jacobian, allowing them to be computed quickly in training. In this paper, we will use a normalizing flow known as Neural Spline Flows (NSF) with coupling layers [13], which uses monotonic rational-quadratic spline transforms for increased flexibility in function representation. This combination allows for neural spline flows to maintain high sampling speeds without losing the breadth associated with autoregressive flows.



**Figure 1.** A visual representation of normalizing flows. Starting with the latent distribution,  $p(u)$ , a number of intermediary distributions are created via the transformations  $T_i$  until the final multi-modal distribution  $p(w)$  is reached. In likelihood-free inference, this entire flow is trained each round.

## 2.2. (Sequential) Neural Posterior Estimation

In neural posterior estimation (NPE),  $q_{F(\mathbf{x}, \phi)}$  is trained to approximate the posterior. In the first round of training,  $N$  pairs of data  $(\boldsymbol{\theta}_j, \mathbf{x}_j)$  are generated by sampling from the prior and running the corresponding simulations, then the neural network weights  $\phi$  are optimized using the loss given in Equation 4. A benefit to NPE is that the training is amortized, so the model can be trained once but used many times to do inference with different experimental observations  $\mathbf{x}$ . However, the drawback to pure NPE is that it may require a large number of simulations to cover the necessary simulation input parameter space. For this reason multi-round sequential neural posterior estimation (SNPE) was introduced, which can reduce the number of simulations needed but is non-amortized, i.e. capable of performing inference only for a single observation  $\mathbf{x}_0$ . The theory behind this approach is that, with a good initial guess for  $p(\boldsymbol{\theta}|\mathbf{x}_0)$  in the first round, the algorithm will selectively target  $\boldsymbol{\theta}$  samples that yield “useful” simulations in the vicinity of  $\mathbf{x}_0$ , increasing efficiency at the expense of being

unable to perform reliable inference for multiple target observations. SNPE accomplishes this by replacing the prior after an initial round of training on  $P$  simulation samples with a proposal distribution  $\tilde{p}(\boldsymbol{\theta})$  set to the current estimate of the posterior  $p(\boldsymbol{\theta}|\mathbf{x}_0)$ , which is  $q_{F(\mathbf{x}_0, \phi)}(\boldsymbol{\theta})$ . This can be repeated for multiple rounds, generating  $P$  more pairs of data each round and further refining the neural density estimator.

However, beyond the first round the density estimator no longer estimates  $p(\boldsymbol{\theta}|\mathbf{x})$ , but rather becomes  $\tilde{q}_{F(\mathbf{x}, \phi)}(\boldsymbol{\theta})$  and estimates the proposal posterior:

$$\tilde{p}(\boldsymbol{\theta}|\mathbf{x}) = p(\boldsymbol{\theta}|\mathbf{x}) \cdot \frac{\tilde{p}(\boldsymbol{\theta})p(\mathbf{x})}{p(\boldsymbol{\theta})\tilde{p}(\mathbf{x})} \quad \text{for} \quad \tilde{p}(\mathbf{x}) = \int_{\boldsymbol{\theta}} \tilde{p}(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})d\boldsymbol{\theta} \quad (5)$$

Multiple approaches have been developed to deal with this correction, the most recent and versatile being Automatic Posterior Transformation or SNPE-C by Greenberg et al [14]. SNPE-C first minimizes the loss for  $\tilde{q}_{F(\mathbf{x}, \phi)}(\boldsymbol{\theta})$ :

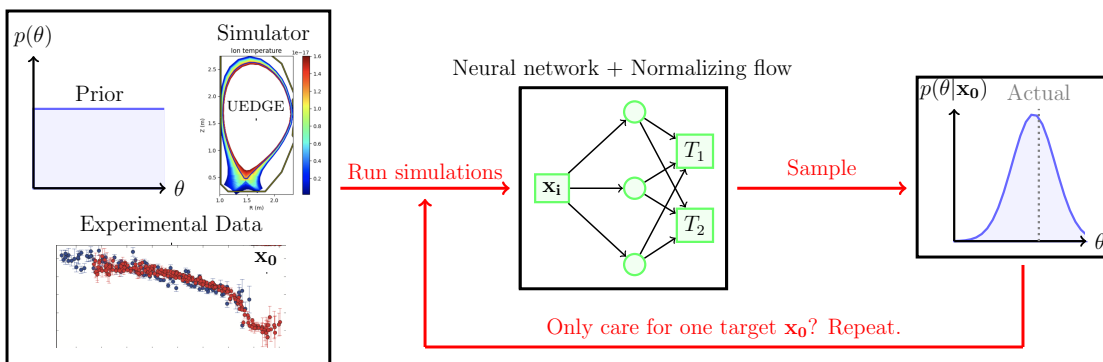
$$\tilde{\mathcal{L}}(\phi) = \sum_{j=1}^N -\log \tilde{q}_{F(\mathbf{x}_j, \phi)}(\boldsymbol{\theta}_j) \quad (6)$$

and later recovers  $q_{F(\mathbf{x}, \phi)}(\boldsymbol{\theta})$  by introducing a conversion factor. In this way, it enables simulations from previous rounds to be used in training despite having different proposals, as the ideal solution  $q_{F(\mathbf{x}, \phi)}(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{x})$  will minimize the negative log-probability loss for any proposal distribution.

In order to accommodate the complex distributions represented by normalizing flows and simplify the necessary integrals, for each pair  $(\boldsymbol{\theta}_j, \mathbf{x}_j)$  SNPE-C discretizes the space of  $\boldsymbol{\theta}$  into a distinct set of atoms  $\Theta_j = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_m\}$  when calculating probabilities. This resolves the training to a process that parallels contrastive learning, where the algorithm selects the “best” choice of  $\boldsymbol{\theta}$  for a particular  $\mathbf{x}_j$  from a discrete set of choices, and ensures that  $q_{F(\mathbf{x}_0, \phi)}(\boldsymbol{\theta})$  will approach  $p(\boldsymbol{\theta}|\mathbf{x}_0) \cdot K$  for some factor  $K \in \mathbb{R}$ . While this means that atomic loss can be used to correctly infer the shape of  $p(\boldsymbol{\theta}|\mathbf{x}_0)$ , there is no guarantee that the amplitude will also be accurately reproduced. This means that leakage issues can occur where  $q$  will have substantial densities in regions where they should be zero, either because the original prior is bounded or because  $\boldsymbol{\theta}$  samples from those regions would yield unreasonable simulation results. Figure 2 illustrates NPE and SNPE.

### 3. UEDGE

UEDGE is a 2D fluid simulation code developed at the Lawrence Livermore National Laboratory for modeling edge-plasma transport in tokamaks, including but not limited to the DIII-D device operated by General Atomics. It implements the Braginskii fluid equations on a curvilinear grid with radial and poloidal cells, providing additional functionality to specify anomalous transport coefficients, model multiple ion species, quantify  $\mathbf{E} \times \mathbf{B}$  and  $\nabla B$  drifts, incorporate neutrals, and account for other processes. Fluid neutral models are default in the code, though UEDGE can be coupled to kinetic



**Figure 2.** A visualization of (sequential) neural posterior estimation, based on a graphic created by Goncalves et al [5]. For a given prior, simulator, and target observation,  $N$  simulations will be run based on samples from the prior. The corresponding  $(\theta, \mathbf{x})$  pairs will be used to train a normalizing flow model based on deep neural networks. This flow can then be sampled from for a particular  $\mathbf{x}_0$  in order to plot the pdf corresponding to  $p(\theta|\mathbf{x}_0)$ . For the sake of simplicity,  $\theta$  is restricted to one dimension in this diagram but in reality will often be multidimensional. If non-amortized SNPE inference is desired, the process of running simulations and training can be repeated by denoting  $p(\theta|\mathbf{x}_0)$  as the proposal distribution to be sampled from instead of the prior.

neutral codes, e.g. EIRENE. The primary plasma variables used in UEDGE are ion density ( $n_i$ ), parallel ion flow velocity ( $u_{\parallel}$ ), electron temperature ( $T_e$ ), ion temperature ( $T_i$ ), neutral gas density ( $n_g$ ), and electrostatic potential ( $\Phi$ ). Based on the initial state of the plasma, values for the anomalous transport coefficients, and boundary conditions, UEDGE updates the plasma variables using the fully-implicit method of Newton-Krylov iterations until steady-state is reached or convergence is deemed impossible. These iterations can be executed in both time-dependent and time-independent modes to obtain steady-state solutions.

An example of one of the fluid equations used, the ion continuity equation, is:

$$\frac{\partial}{\partial t} n_i + \frac{1}{V} \frac{\partial}{\partial x} \left( \frac{V}{h_x} n_i u_{ix} \right) + \frac{1}{V} \frac{\partial}{\partial y} \left( \frac{V}{h_y} n_i u_{iy} \right) = \langle \sigma_i v_{te} \rangle n_e n_g - \langle \sigma_r v_{te} \rangle n_i n_g \quad (7)$$

where  $x, y$  are the poloidal, radial directions respectively,  $h_x, h_y$  are metric coefficients,  $V$  is a toroidal volume element, and the  $u$  are velocities, with  $u_{ix}$  taken as classical parallel streaming with poloidal drifts, and the radial velocity  $u_{iy} = \frac{D_a}{n_i} \frac{\partial n_i}{\partial y} + V_a + v_{y, drifts}$ . Here  $D_a$  and  $V_a$  are the anomalous particle transport coefficients, simulating in an ad-hoc way turbulent transport, which has been shown to be significant in the SOL (note that this ad-hoc manner is a simplification of the true transport fluxes that can arise using a proper turbulence code). In UEDGE, the anomalous transport coefficients offered are density diffusion ( $D_a$ ), convective particle pinch velocity ( $V_a$ ), radial parallel momentum diffusion ( $\Upsilon_{a\parallel}$ ), and electron, ion energy diffusion ( $\chi_e, \chi_i$ ). The values for these coefficients must be defined for each cell, although UEDGE affords the user flexibility when it comes to how they are defined. They can be set constant

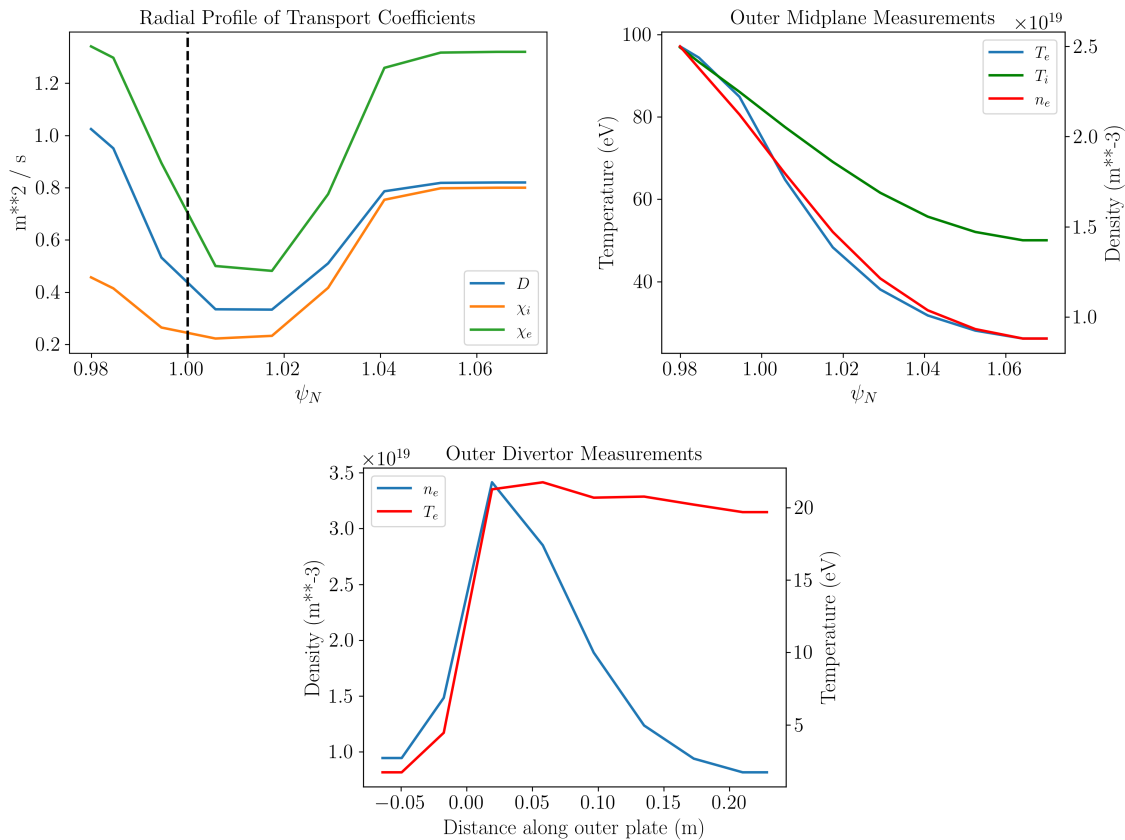
across the grid, user-specified across the entire grid, or user-specified in the radial direction at the outer midplane and varied poloidally using magnetic field ratios or Bohm-like diffusion (e.g.  $D_a(\vartheta) \propto 1/B$ , where  $\vartheta$  is the poloidal angle).

#### 4. Likelihood-free inference with UEDGE workflow

The simulations used are set up and run using the Python implementation of UEDGE, known as pyUEDGE. An initial setup including a magnetic geometry mesh ( $N_x = 18$  radial and  $N_y = 10$  poloidal zones) based on the DIII-D tokamak was taken from the LLNL/UEdge/pyexamples/d3dHsm GitHub directory<sup>‡</sup> and converged in the time-dependent mode for  $D = \chi_e = \chi_i = 1 \frac{\text{m}^2}{\text{s}}$  at the outer midplane with Bohm-like poloidal variations. Notable boundary conditions and configuration details include  $P_{\text{core}, e} \approx P_{\text{core}, i} \approx 1.4$  MW and zero energy flux at the wall and in the private plasma. The simplified diffusive neutral model was used [15]. The converged solution was saved to an HDF5 file and later used as the starting point for future simulations, which will vary the anomalous transport coefficients. For all simulations, the recycling coefficient was held constant at 0.8 as well as  $\Upsilon_{i,e,\parallel}$ , a transport coefficient typically adjusted in UEDGE simulations, which was maintained at  $1 \frac{\text{m}^2}{\text{s}}$  for the outer midplane. It is important to note that  $V_a$  also remained zero for all simulations, meaning that the diffusion coefficients represent effective diffusion,  $\Gamma_a = -D_{a,eff} \partial n_a / \partial y$ . Under these specifications, UEDGE is deterministic but still intractable, meaning that for the same transport coefficients and magnetic geometry the plasma variables of the converged solution will always be the same (assuming it exists).

For the purposes of conducting likelihood-free inference, it is essential to specify the output observables of interest  $\mathbf{x}$  and the prior distribution  $p(\boldsymbol{\theta})$  over the input parameters  $\boldsymbol{\theta}$  (here the anomalous transport coefficient profiles). The following arrays, with entries at each of the 10 values of  $\psi_N$  along the outer midplane corresponding to the 10 radial indices defined for the plasma mesh, are extracted from UEDGE results to form  $\mathbf{x}$  and represent each converged simulation: electron density ( $n_e$ ) at the outer midplane, electron temperature ( $T_e$ ) at the outer midplane, ion temperature ( $T_i$ ) at the outer midplane, electron density at the outer divertor plate, and electron temperature at the outer divertor plate (here  $\psi_N$  is the normalized poloidal flux,  $\psi_N = (\psi - \psi_0) / (\psi_x - \psi_0)$ , where  $\psi$  is the poloidal magnetic flux, and  $\psi_0, \psi_x$  the value at the magnetic axis, X-point). This leads to a dimensionality of  $5 \cdot 10 = 50$  for  $\mathbf{x}$ . This choice aligns with the diagnostic measurements that are typically made using Thomson Scattering systems and Langmuir probes to observe magnetically-confined plasmas during tokamak shots.  $p(\boldsymbol{\theta})$  is taken to be a uniform distribution on the range (0.1, 1.5) for each of  $D, \chi_e$ , and  $\chi_i$  at each radial index along the outer midplane, leading to a dimensionality of  $3 \cdot 10 = 30$  for  $\boldsymbol{\theta}$ . The bounds on the prior reflect traditional choices for the ranges that the transport coefficients are typically given. The poloidal variation of the anomalous transport coefficients is Bohm-like, using UEDGE to determine. Note that many comparisons

<sup>‡</sup> <https://github.com/LLNL/UEdge/tree/master/pyexamples/d3dHsm>



**Figure 3.** The  $\theta_0$  profile of the transport coefficients and corresponding observables  $\mathbf{x}_0$  used to evaluate the (S)NPE algorithm. While the signals at the outer midplane are plotted against  $\psi_N$ , the signals at the outer divertor plate are plotted against a normalized distance metric.

of fluid codes to experiment in the SOL have required additional poloidal variation in the anomalous transport coefficients beyond Bohm-like in order to get a good match to experiment [16, 17]. Future work will look at including the poloidal variation of the anomalous transport coefficients in the physics parameters of interest  $\theta$ . Note also that in general any physics parameter affecting UEDGE simulations can be included in  $\theta$ , such as  $V_a$  and the recycling coefficient, and a goal of future work will be to test the limits on the dimensionality of  $\theta$  for which inference is possible.

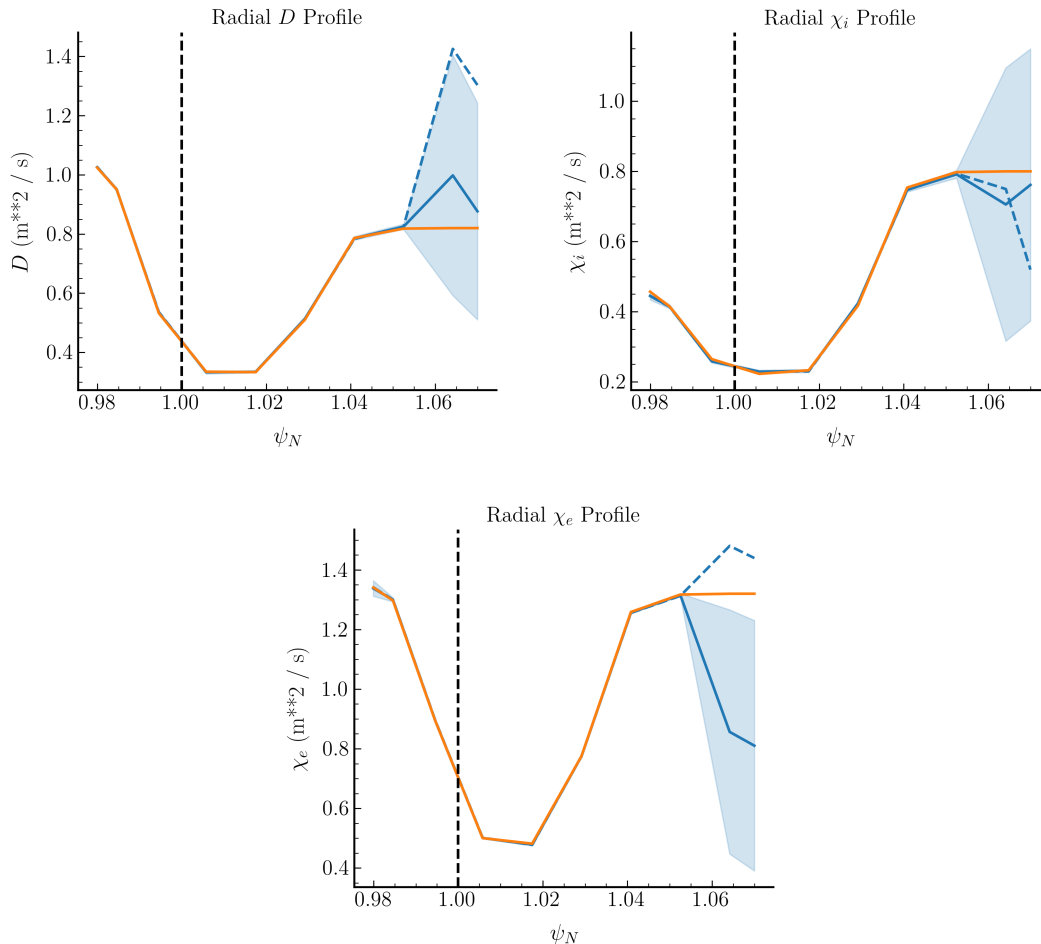
As an initial first step, in this paper we use synthetic experimental signals  $\mathbf{x}_0$ , which are generated from a ground truth  $\theta_0$ , in order to determine how well the (S)NPE algorithm recovers the ground truth (see Section 6 for a discussion on including experimental error bars).  $\theta_0$  was created using piecewise tanh functions to resemble realistic anomalous transport profiles, with a dip in values in the pedestal region, and higher values towards the core and in the SOL. Figure 3 shows both  $\theta_0$  and  $\mathbf{x}_0$ .

SBI [18] is a Python package for likelihood-free inference developed by mackelab from the University of Tübingen, providing frameworks to perform multi-round or amortized inference for a variety of methods including SNPE-C. Beyond choices of the

method and amortization, SBI also allows a variety of hyperparameters to be adjusted, such as the type of density estimator used (neural spline flows, masked autoregressive flows, or MDNs), the number of transforms in the normalizing flow, the number of hidden features in the neural network training the flow, and the number of atoms per  $\Theta_j$  for SNPE-C. Neural spline flows were chosen over masked autoregressive flows, while the number of transforms, hidden features, validation fraction, and atoms per  $\Theta_j$  were set to 10, 120, 0.20, and 20 within the SBI architecture, respectively. More conventional neural network hyperparameters such as the learning rate and training batch size were left at their default values of 0.005 and 50, and the default hyperparameters for the Adam optimizer were used. For SBI, which is built on PyTorch, this translates to  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 10^{-8}$ . The deep neural network was not retrained from scratch each round, with the last round’s network being used as the starting point for the next round’s training. Combined loss, which incorporates maximum likelihood estimation on previous training samples in addition to the negative log-probability, was also used to minimize leakage.

In order to analyze the posterior models created each round, SBI provides a neural network-based method to approximate the maximum-a-posteriori (MAP) estimate, calculate normalized log-probabilities, and sample from  $q_{F(\mathbf{x}_0, \phi)}(\boldsymbol{\theta})$  as a means of evaluating expected values and confidence intervals. Other error metrics such as median distance, maximum mean discrepancy, and classifier 2-sample tests have been applied to likelihood-free inference methods, but they are most helpful in cases where the posterior  $p(\boldsymbol{\theta}|\mathbf{x}_0)$  is already known [19]. This is far from the case for the complex, professional-grade fusion simulations on which SNPE-C will eventually be performing inference, therefore they are not used.

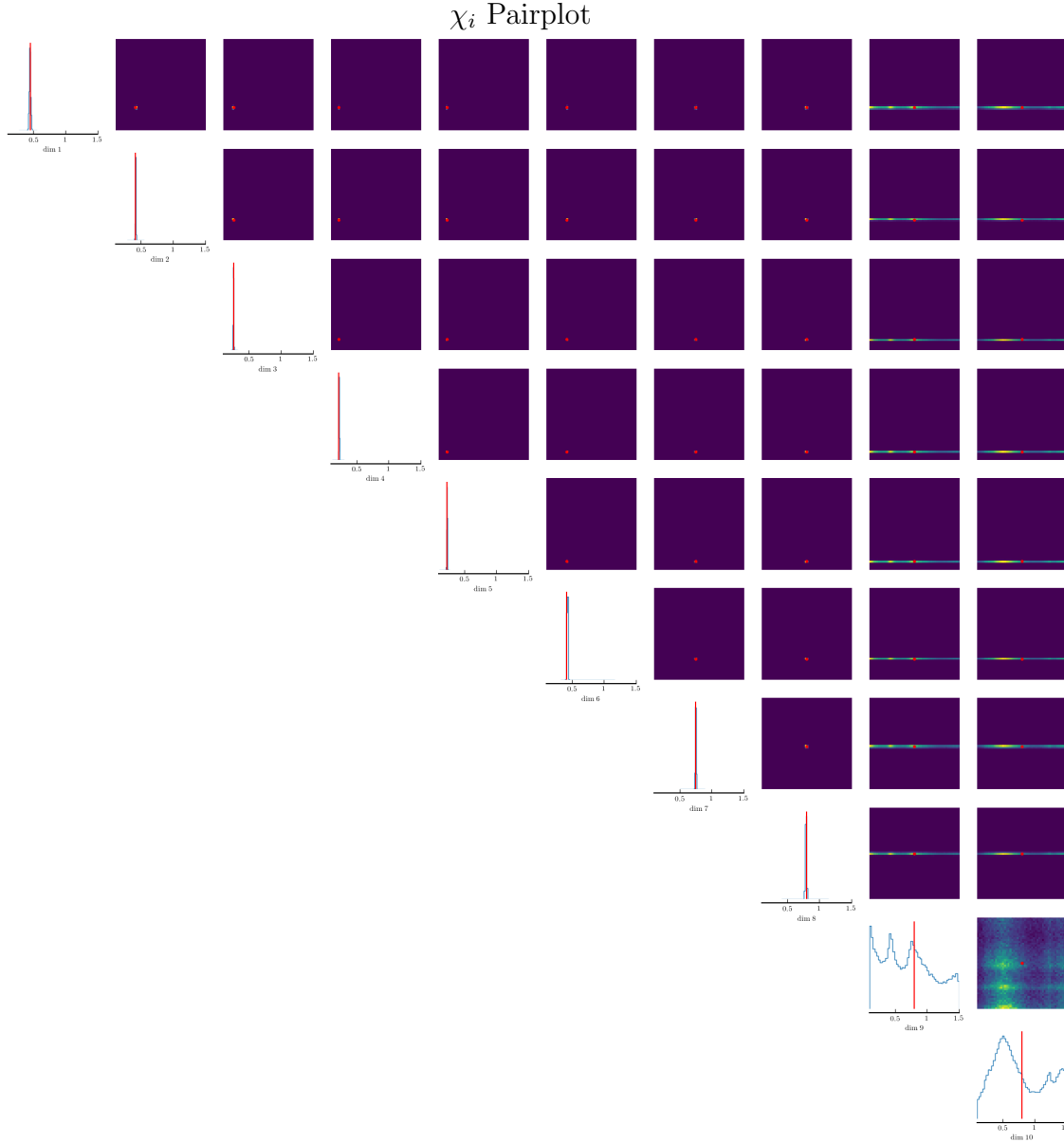
Two trials, both with a total of 10,000 simulations, were run at varying levels of amortization in order to compare SNPE with single-round NPE and provide a benchmark for how best to allocate limited computational resources for simulations. Trial One consisted of a single round with 10,000 simulations, while Trial Two consisted of 10 rounds with 1,000 simulations in each round. On average, the simulations took  $\sim 10$  seconds to converge and the neural network training  $\sim 30$  minutes. The simulations in each round can be run in parallel, as they are independent from one another, but in this work were run in serial due to complications arising from parallel SBI with the pyUEDGE code that are still being worked out. This led to total run times of between 25 and 35 hours. The simulations and network training were performed on Princeton Plasma Physics Laboratory (PPPL) clusters. Progress has already been made towards parallelizing individual UEDGE simulations, and implementing this will be a key step towards scaling up these methods in the future, as preliminary experiments with  $N_x = 34$ ,  $N_y = 18$  grids have yielded convergence times of  $\sim 45$  seconds per simulation. Further discussion on this topic is included in Section 6.



**Figure 4.** The radial  $D$ ,  $\chi_e$ , and  $\chi_i$  profiles plotted against  $\psi_N$  at the outer midplane for Trial One with  $N = 10,000$  simulations. The true values,  $\theta_0$ , are depicted in orange, while the MAP estimate (dashed), expected value (solid), and confidence intervals are in blue. The separatrix at  $\psi_N = 1$  is denoted by the dashed vertical lines.

## 5. Results

Figure 4 shows the inferred radial profiles for the transport coefficients in Trial One, which consisted of amortized NPE inference with a single round of 10,000 simulations. The true profile,  $\theta_0$ , is represented by the orange curves, the maximum-a-posteriori (MAP) estimate is represented by the dashed blue curves, the expected value (based on  $10^5$  posterior samples) is represented by the solid blue curves, and the 95% confidence intervals corresponding to the expected value are represented by the light blue error bars. For all three transport coefficients, the MAP estimate and expected value of the normalizing flow surrogate model,  $q$ , inferred by the algorithm fit the true values very strongly, with the blue curves and confidence intervals being hidden behind the orange. However, for values of  $\psi_N$  greater than 1.06 that approach the wall, the algorithm's estimates are far more uncertain, with wide error bars and inaccurate MAP estimates.



**Figure 5.** The  $\chi_i$  pairplot at the outer midplane for Trial One with  $N = 10,000$  simulations. The true values,  $\theta_0$ , are depicted as solid red lines in the 1D distributions and red dots for the marginal distributions. From left to right, the dimensions correspond to the radial indices of the UEDGE grid in increasing order of  $\psi_N$ . While the 1D distributions are plotted on  $[0.1, 1.5]$  and the marginal distributions on  $[0.1, 1.5] \times [0.1, 1.5]$ , there may be nonzero probability densities beyond these ranges. Pairplots for  $D$  and  $\chi_e$  are not shown, but mirror the trends of the  $\chi_i$  pairplot.

For both  $D$  and  $\chi_i$ , the true values at the outer boundary remain firmly within the confidence intervals, but this is not the case for  $\chi_e$ , although the MAP estimate is far more accurate. Especially for  $D$ , this uncertainty at the edge is expected and is reflective of the added complexity that exists near the wall due to atomic processes and boundary conditions.

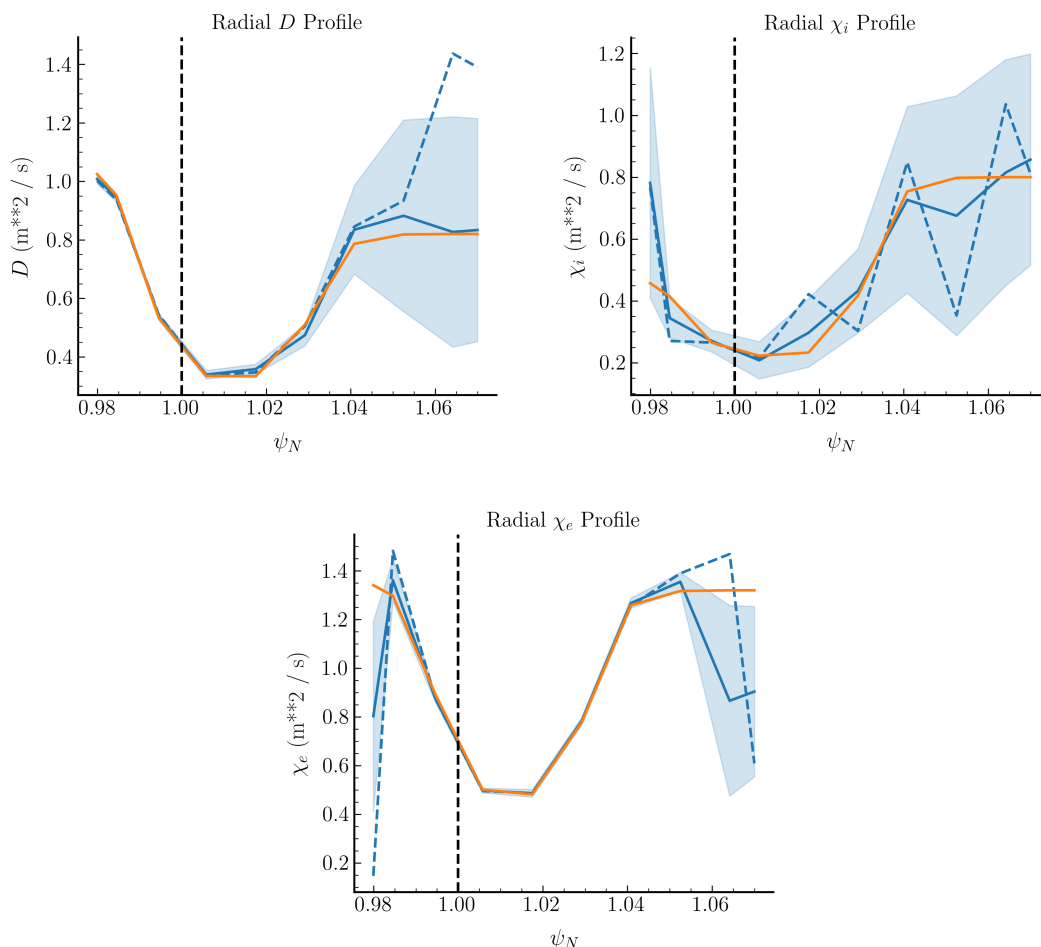
Figure 5, which shows a pairplot consisting of marginal 2D probability distributions

as well as individual distributions based on  $10^5$  posterior samples for  $\chi_i$  at each radial ( $\psi_N$ ) index, reflects many of the same trends found in Figure 4. The 1D, individual distributions are depicted along the diagonals, while the marginal distributions occupy the other grid points. For the first eight radial points, with  $\psi_N < 1.06$ , the distributions generated by the surrogate model are hidden by the red lines and dots representative of the  $\theta_0$  values at those indices, similar to the radial profiles. The uncertainty in the outer boundary estimates is more clear in the pairplot, as the distributions at the radial points with  $\psi_N > 1.06$  are far more spread out. While not displayed, these trends are mirrored in the pairplots for  $\chi_e$  and  $D$  as well. This suggests that the MAP estimates aren't as useful in this region, since the density estimator has not zeroed in on a particular estimate and may settle on a different value in later rounds. The multi-modal, asymmetric distributions at these points appear to be caused by the splines within neural spline flows, which is a common feature when they are unable to zero in on a singular estimate.

Earlier attempts often produced estimated profiles that matched the general shape of the  $\theta_0$  distributions, but were shifted either up or down by some constant factor. This issue is commonly faced when trying to fit transport coefficient profiles to experimental results, and in our particular case was likely attributed to the fact that the UEDGE simulations were being run using fixed electron and ion temperatures,  $T_e$  and  $T_i$ , at both the inner and outer boundaries. The simulation configurations were adjusted to set constant power fluxes across the core and zero energy fluxes at the wall and private plasma instead of the fixed temperatures, which allowed for the shifts to be minimized. Both Trial One and Trial Two used these constant flux boundary conditions.

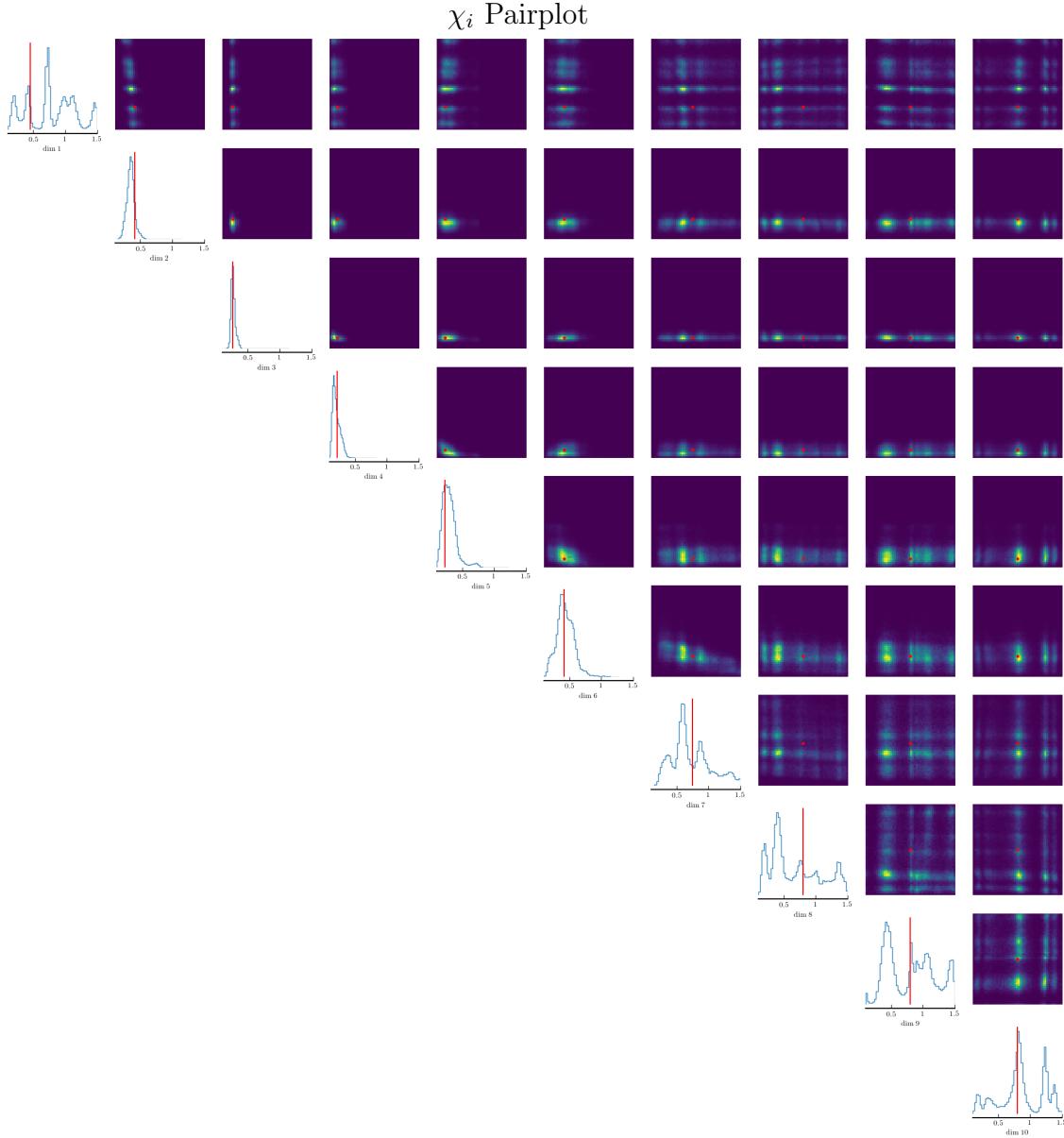
Trial Two, which was the non-amortized SNPE trial with 10 rounds of inference (Round One through Round Ten) and 1,000 simulations per round, yielded weaker results than the NPE inference of Trial One regarding the accuracy of the MAP estimates, width of the confidence intervals, and validation loss. For Round One, whose radial profiles are depicted in Figure 6, the surrogate model is far less accurate and more uncertain for  $\chi_i$ , as well as at the inner boundary for  $\chi_e$ . However, for both  $D$  and  $\chi_i$  the true values are well within the error bounds. This is also seen in the representative  $\chi_i$  pairplot of Figure 7, as the marginal and 1D distributions tend to be more spread out and multi-modal at a significant number of radial indices. This contrasts with the results obtained from the earlier attempts with fixed boundary temperatures as opposed to fixed fluxes, where the difference between using 1,000 and 10,000 simulations per round was far less noticeable. This suggests that using fixed fluxes may lead to simulations that are more sensitive to the transport coefficients, and thus require a greater number of simulations to capture the converged plasma variables that can be generated across the full space of the prior.

Because of the fact that for each round of inference, the neural network encoding the normalizing flows is trained to completion (non-decreasing validation loss), the loss measured using negative log-probability fluctuates and can sometimes increase between rounds due to the changing validation set. Additionally, because SNPE only guarantees



**Figure 6.** The radial  $D$ ,  $\chi_e$ , and  $\chi_i$  profiles at the outer midplane for Trial Two, Round One based on a total of  $N = 1,000$  simulations.

that the trained posterior,  $q_{F(\mathbf{x}, \phi)}$ , will match the true posterior up to a constant factor, changes in the loss may be more so determined by changes to that factor and in turn the level of leakage rather than improvements to the shape of the trained distribution. The lowest validation loss, given by Equation 4, was recorded in Round Nine at  $-52$ , compared to  $-36$  for Round One. Note that despite both values being representative of probabilities, they are derived from different validation sets since the validation set is reconstructed and guided towards  $\mathbf{x}_0$  as the total size of the dataset increases each round. Although there is a significant improvement over rounds, Round Nine’s loss is still far greater than the loss recorded for Trial One, which was close to  $-100$ . While these comparisons are relative due to differences in the validation sets, this suggests that in addition to being more accurate, the normalizing flow in Trial One likely had less leakage as well. Figure 9, which shows the radial profiles for Round Nine, shows noticeable improvements in the surrogate model’s predictions for  $\chi_i$ . Although the boundary points are still somewhat uncertain, the error bars are much tighter and both the expected value and MAP estimates follow the true profile for the radial indices where

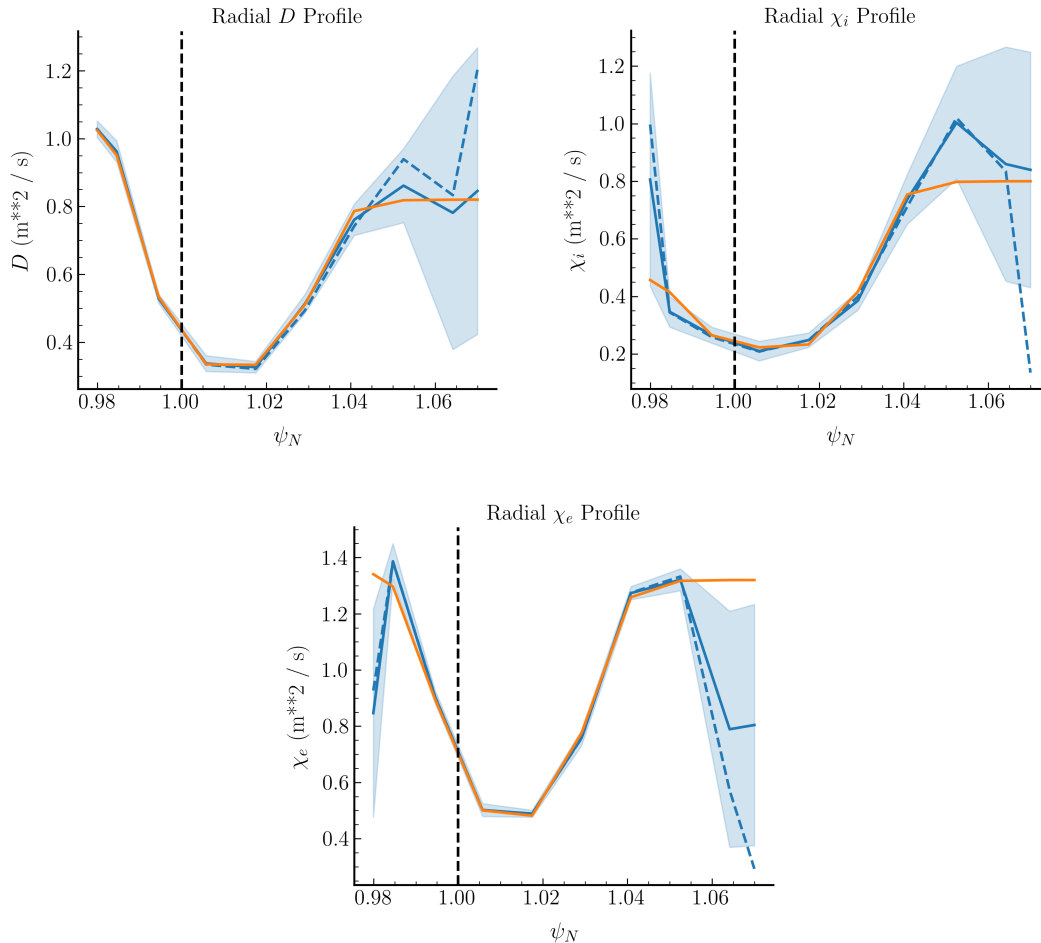


**Figure 7.** The  $\chi_i$  pairplot for Trial Two, Round One based on a total of  $N = 1,000$  simulations. As with Trial One, this pairplot is representative of the pairplots for  $D$  and  $\chi_e$  which displayed similar trends.

Trial	Boundary Conditions
Earlier Attempts*	Fixed $T_e, T_i$ at core and wall
Trial One	$P_{\text{core}, e} \approx P_{\text{core}, i} \approx 1.4$ MW and zero energy flux at wall
Trial Two	$P_{\text{core}, e} \approx P_{\text{core}, i} \approx 1.4$ MW and zero energy flux at wall

**Figure 8.** A table explaining the boundary conditions used for Trials One and Two, as well as the earlier attempts that are referenced in Section 5. \*Graphs and plots for the earlier attempts are not included.

$0.98 < \psi_N < 1.05$ . The estimates for  $D$  and  $\chi_e$  have remained roughly unchanged across rounds. It is unclear whether this set of profiles corresponds to a local minimum, or whether the SNPE algorithm would be able to make further improvements, especially at the boundaries, with additional rounds. The representative  $\chi_i$  pairplot in Figure 10 suggests that the probability distributions may be beginning to smooth and converge to specific values at points with large uncertainties.

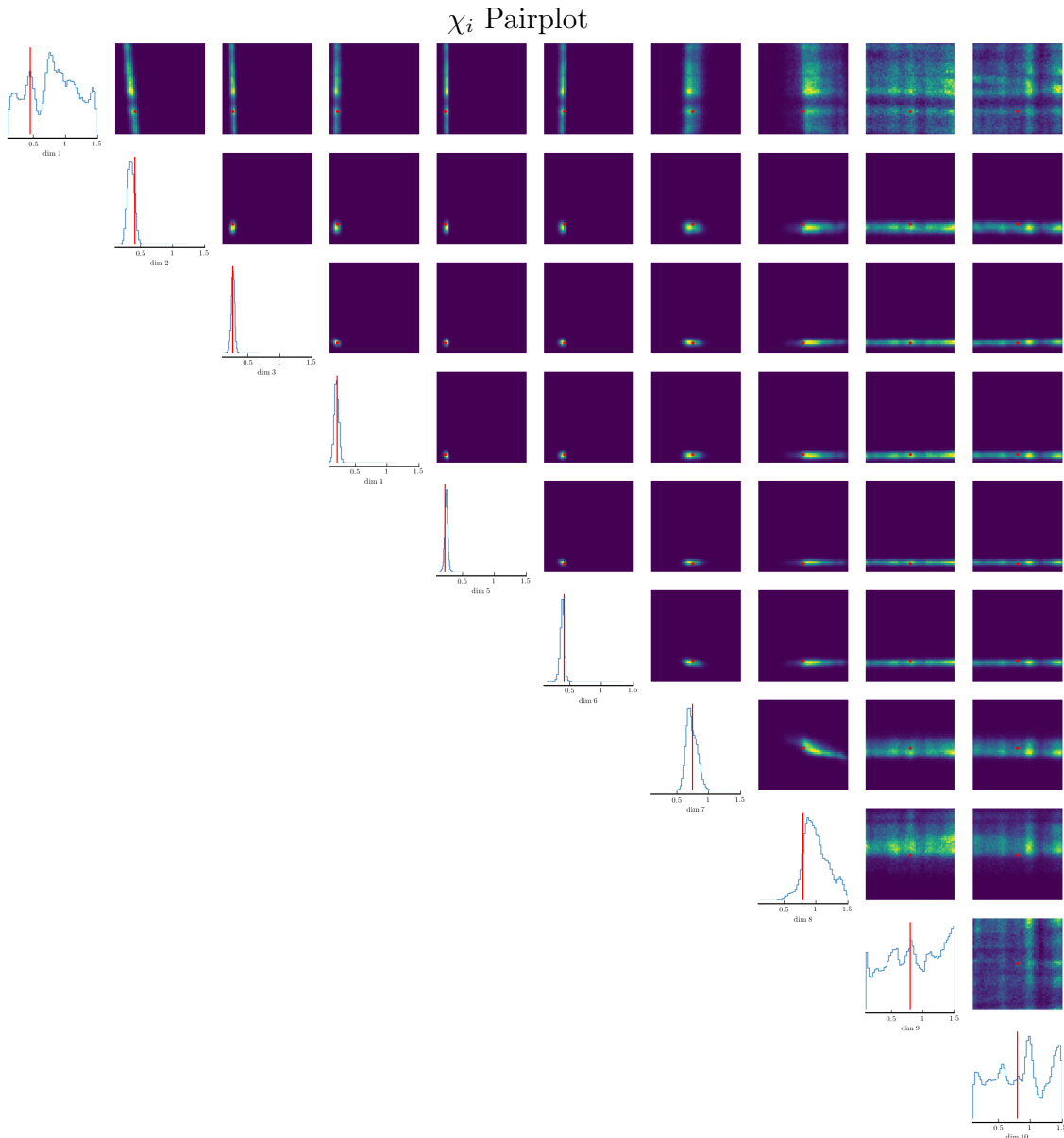


**Figure 9.** The radial  $D$ ,  $\chi_e$ , and  $\chi_i$  profiles at the outer midplane for Trial Two, Round Nine based on a total of  $N = 9,000$  simulations.

## 6. Future Work

Despite the progress that has been made, there remain some questions that need to be answered before this method can successfully be scaled up to the larger grid sizes that are used in the more detailed analysis of tokamaks.

As was shown in Trial Two, because the deep neural network trains to completion every round, the first round is by far the most important with later rounds providing adjustments and fine-tuning. Leakage also remains a concern, both in terms of how it



**Figure 10.** The representative  $\chi_i$  pairplot for Trial Two, Round Nine based on a total of  $N = 9,000$  simulations.

impacts the loss and its role in enabling certain training configurations. In order to avoid the sampling issues that arise when a significant proportion of the probability density in the inferred posterior lies beyond the bounds of the prior, the neural network could not be retrained from scratch or forgo the combined loss. It is likely that removing these constraints would lead to a greater level of training beyond the first round, and potentially lead to results such as those in Greenberg et al [14] where the accuracy of the surrogate models increases more gradually and significantly over rounds. The number of simulations may have had a role in the leakage as well, since we were limited to a total of roughly 10,000 simulations that could be distributed across rounds. Based on the stronger performance of the single-round model, it is possible that we were using

too few simulations in the non-amortized case to sufficiently capture the prior space in the first few rounds and prevent leakage issues upon enabling the “retrain from scratch” and “no combined loss” features.

Going forward, it will be crucial to parallelize the UEDGE simulations, especially for larger grids where it can take upwards of 60 seconds for a converged profile to be obtained for a single set of transport coefficients (if starting from an unconverged solution, may take many hours to converge depending on physics included, like drifts, etc.). The dataset generation of UEDGE simulations is embarrassingly parallel, and implementing this will be key to generating the larger datasets that will likely be needed for more complex simulations. We also acknowledge the relative simplicity of the current model. Multiple commonly-adjusted transport coefficients, such as the parallel momentum diffusion and  $V_a$ , were held constant on the side of the inference, while nuisance parameters such as the magnetic geometry, recycling coefficient, and other atomic physics assumptions that vary from shot to shot in a tokamak were not varied nor included within  $\mathbf{x}_0$ . In order to create an amortized master algorithm that is capable of performing inference for any experimental results from a tokamak, these factors would have to be incorporated.

Scaling up these methods will undoubtedly lead to higher dimensionalities, both in the prior space and the corresponding simulation outputs. This will increase the simulation budgets necessary to cover the prior space, which in turn corresponds to increased memory and run time costs. Parameterization methods, such as the piecewise tanh function approach used to generate  $\boldsymbol{\theta}_0$  or the more complex interchange turbulence model that solves an additional PDE of turbulent kinetic energy transport [3],  $\kappa$ , can be implemented to reduce the dimensionality of the transport coefficients that need to be inferred. Further, embedding neural networks and other methods such as principal component analysis (PCA) can be used to reduce the dimensionality of the outputs. Additionally, methods such as nuisance hardened data compression are being developed to perform marginal inference with nuisance parameters without having to explicitly include them within the prior space [20]. This would allow for the dimensionality of the prior to be significantly reduced without compromising on the range of boundary conditions, magnetic geometries, transport coefficients, and other parameters that can be captured, making it an ideal candidate for large-scale amortized algorithms.

For comparisons to actual experiment, the uncertainties in the diagnostic measurement have to be included to derive a proper posterior over the anomalous transport coefficients. To properly include, we plan to follow the method in Ref. [21], by first creating a noise model distribution (e.g. a simple Normal distribution,  $p(\eta_x) = \mathcal{N}(0, \sigma_x)$ ) for the diagnostic measurements, which is then sampled from and added to the simulations outputs,  $\hat{\mathbf{x}} = \mathbf{x} + \eta_x$ . Note that a single simulation can potentially be reused, sampling different noise levels and using as separate training example for the normalizing flow. We then include  $\sigma_x$  as a conditional variable in the trained posterior, e.g.  $p_\phi(\boldsymbol{\theta}|\mathbf{x}, \sigma_x)$ , so that at inference time (with the trained model), we pass in the experimental measurements  $\mathbf{x}$  and their uncertainties ( $\sigma_x$ ), which should

then give an accurate posterior for the physical quantities  $\theta$ .

## 7. Conclusions

We have shown that (sequential) neural posterior estimation with neural spline flows holds promise for inferring from experimental data physics parameters which are consistent with simulation models. This method allows us to have estimates and confidence intervals (beyond a sole maximum-a-posteriori estimate) for anomalous transport coefficients consistent with experimental results and the UEDGE code. When applied to 18x10 UEDGE grids with a total simulation budget of 10,000 simulations, the amortized NPE inference in Trial One was shown to be stronger than the non-amortized SNPE inference in Trial Two with 1,000 simulations per round, despite the latter’s theoretical increase in efficiency due to solely focusing in on a single observation  $\mathbf{x}_0$ . However, noticeable improvements were measured over the rounds in Trial Two, especially for predictions of  $\chi_i$ .

In the previous Section, we discussed several future work items to related to using these techniques in true experimental scenarios. For example, this will require using more realistic (and more expensive) UEDGE simulations, which take longer due to e.g. the inclusion of drifts, finer mesh resolution, etc. While individual simulations will have longer run times, for amortized models (e.g. NPE) the many simulations can be run in parallel. Large-scale HPC resources can be leveraged to run these many simulations in parallel, and train the corresponding normalizing flow surrogate model. These amortized models can then be used many times for varying experimental conditions.

These likelihood-free inference techniques using neural density estimators such as normalizing flows can find broader application to many of the codes used in fusion energy research. This can include, for example, MHD codes or gyrokinetic turbulence codes, but with the understanding that the inferred parameter space be tractable, and the simulation sufficiently fast (or the compute resources sufficiently large) to generate a significant number of simulations. These techniques can bring more rigor to the uncertainty quantification task of comparing simulation to experiment, and due to the neural network basis, be flexible surrogate models of the simulators for more routine use in experimental comparisons.

## 8. Acknowledgements

We gratefully acknowledge the UEDGE team for the published code, and acknowledge Tom Rognlien, Jerome Guterl, and Andrew Xing for very helpful discussions about UEDGE. This work was made possible by funding from the Department of Energy for the High School Internship program. This work is supported by the US DOE Contract No. DE-AC02-09CH11466.

## 9. References

- [1] Rognlien T D, Ryutov D D, Mattor N and Porter G D 1999 *Physics of Plasmas* **6** 1851–1857 ISSN 1070-664X, 1089-7674 URL <http://aip.scitation.org/doi/10.1063/1.873488>
- [2] Izacard O 2018 Automatic UEDGE simulations of a large series of time-slices for tokamak discharges *Bulletin of the American Physical Society* vol Volume 63, Number 11 (American Physical Society) URL <https://meetings.aps.org/Meeting/DPP18/Event/337398>
- [3] Carli S, Dekeyser W, Blommaert M, Coosemans R, Van Uytven W and Baelmans M 2021 *Contributions to Plasma Physics* n/a e202100184 ISSN 1521-3986 .eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/ctpp.202100184> URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/ctpp.202100184>
- [4] Cranmer K, Brehmer J and Louppe G 2020 *Proc Natl Acad Sci USA* **117** 30055–30062 ISSN 0027-8424, 1091-6490 URL <http://www.pnas.org/lookup/doi/10.1073/pnas.1912789117>
- [5] Gonçalves P J, Lueckmann J M, Deistler M, Nonnenmacher M, Öcal K, Bassetto G, Chintaluri C, Podlaski W F, Haddad S A, Vogels T P, Greenberg D S and Macke J H 2020 *eLife* **9** e56261 ISSN 2050-084X publisher: eLife Sciences Publications, Ltd URL <https://doi.org/10.7554/eLife.56261>
- [6] Cranmer M D, Galvez R, Anderson L, Spergel D N and Ho S 2019 *arXiv:1908.08045 [astro-ph, stat]* ArXiv: 1908.08045 URL <http://arxiv.org/abs/1908.08045>
- [7] Green S R and Gair J 2021 *Mach. Learn.: Sci. Technol.* **2** 03LT01 ISSN 2632-2153 URL <https://iopscience.iop.org/article/10.1088/2632-2153/abfaed>
- [8] Beaumont M A 2019 *Annual Review of Statistics and Its Application* **6** 379–403 ISSN 2326-8298 publisher: Annual Reviews URL <https://www.annualreviews.org/doi/10.1146/annurev-statistics-030718-105212>
- [9] Papamakarios G, Nalisnick E, Rezende D J, Mohamed S and Lakshminarayanan B 2021 *arXiv:1912.02762 [cs, stat]* ArXiv: 1912.02762 URL <http://arxiv.org/abs/1912.02762>
- [10] Kobzyev I, Prince S J D and Brubaker M A 2021 *IEEE Trans. Pattern Anal. Mach. Intell.* **43** 3964–3979 ISSN 0162-8828, 2160-9292, 1939-3539 arXiv: 1908.09257 URL <http://arxiv.org/abs/1908.09257>
- [11] Bishop C M 2006 *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Berlin, Heidelberg: Springer-Verlag) ISBN 0387310738
- [12] Kullback S and Leibler R 1951 *The Annals of Mathematical Statistics* **22** 79–86 URL <https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694.full>
- [13] Durkan C, Bekasov A, Murray I and Papamakarios G 2019 Neural Spline Flows *Advances in Neural Information Processing Systems* vol 32 (Curran Associates, Inc.) URL <https://proceedings.neurips.cc/paper/2019/hash/7ac71d433f282034e088473244df8c02-Abstract.html>
- [14] Greenberg D, Nonnenmacher M and Macke J 2019 Automatic Posterior Transformation for Likelihood-Free Inference *Proceedings of the 36th International Conference on Machine Learning* (PMLR) pp 2404–2414 iSSN: 2640-3498 URL <https://proceedings.mlr.press/v97/greenberg19a.html>
- [15] Rognlien T D, Brown P N, Campbell R B, Kaiser T B, Knoll D A, McHugh P R, Porter G D, Rensink M E and Smith G R 1994 *Contributions to Plasma Physics* **34** 362–367 ISSN 1521-3986 .eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/ctpp.2150340241> URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/ctpp.2150340241>
- [16] Wu H, Subba F, Wischmeier M, Cavedon M and Zanino R 2021 *Plasma Phys. Control. Fusion* **63** 105005 ISSN 0741-3335 publisher: IOP Publishing URL <https://doi.org/10.1088/1361-6587/ac1568>
- [17] Reimold F, Wischmeier M, Bernert M, Potzel S, Coster D, Bonnin X, Reiter D, Meisl G, Kallenbach A, Aho-Mantila L and Stroth U 2015 *Journal of Nuclear Materials* **463** 128–134 ISSN 00223115 URL <https://linkinghub.elsevier.com/retrieve/pii/S002231151400960X>

- [18] Tejero-Cantero A, Boelts J, Deistler M, Lueckmann J M, Durkan C, Gonçalves P J, Greenberg D S and Macke J H 2020 *Journal of Open Source Software* **5** 2505 ISSN 2475-9066 URL <https://joss.theoj.org/papers/10.21105/joss.02505>
- [19] Lueckmann J M, Boelts J, Greenberg D S, Gonçalves P J and Macke J H 2021 *arXiv:2101.04653 [cs, stat]* ArXiv: 2101.04653 URL <http://arxiv.org/abs/2101.04653>
- [20] Alsing J and Wandelt B 2019 *Monthly Notices of the Royal Astronomical Society* **488** 5093–5103 ISSN 0035-8711, 1365-2966 arXiv: 1903.01473 URL <http://arxiv.org/abs/1903.01473>
- [21] Hahn C and Melchior P 2022 Accelerated Bayesian SED Modeling using Amortized Neural Posterior Estimation Tech. Rep. arXiv:2203.07391 arXiv arXiv:2203.07391 [astro-ph, stat] type: article URL <http://arxiv.org/abs/2203.07391>