

First estimation of model parameters for neutrino-induced nucleon knockout using simulation-based inference

Karla Tame-Narvaez¹, Steven Gardiner¹, Aleksandra Ćiprijanović^{1,2,3} and Giuseppe Cerati¹

¹*Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA*

²*Department of Astronomy and Astrophysics, University of Chicago, Chicago, Illinois 60637, USA*

³*NSF-Simons AI Institute for the Sky (SkAI), 172 E. Chestnut St., Chicago, Illinois 60611, USA*



(Received 18 March 2026; revised 14 August 2026; accepted 18 August 2026; published 23 September 2026)

To enable an accurate determination of oscillation parameters, accelerator-based neutrino experiments require detailed simulations of nuclear interaction physics in the GeV regime. While substantial effort from both theory and experiment is currently being invested to improve the fidelity of these simulations, their present deficiencies typically oblige experimental collaborations to resort to empirical tuning of simulation model parameters. As the precision requirements of the field continue to become more stringent, machine learning techniques may provide a powerful means of handling corresponding growth in the complexity of future neutrino interaction model tuning exercises. To study the suitability of simulation-based inference (SBI) for this physics application, in this paper we revisit a tuned configuration of the GENIE neutrino configuration generator that was originally developed by the MicroBooNE collaboration. Despite closely reproducing the adopted values of four physics parameters when confronted with the tuned cross-section predictions as input, we find that our trained SBI algorithm prefers modestly different values (within MicroBooNE's assigned uncertainties) and achieves slightly better goodness-of-fit when inference is run on the experimental data set originally used by MicroBooNE. We also find that our trained algorithm can create a fair approximation of an alternative neutrino scattering simulation, NuWro, that shares only a subset of its physics model parameters with GENIE.

DOI: [10.1103/tzhn-7tcq](https://doi.org/10.1103/tzhn-7tcq)

I. INTRODUCTION

Obtaining a precise and comprehensive understanding of neutrino oscillations is a goal at the forefront of worldwide investments in high-energy physics. Next-generation neutrino experiments based at accelerator facilities, including Hyper-Kamiokande [1] and the Deep Underground Neutrino Experiment [2,3], will require percent-level control of systematic uncertainties in order to deliver their flagship oscillation results [4]. Among the sources of uncertainty for these efforts, some of the most challenging are those arising from imperfect modeling of neutrino-nucleus interactions at the relevant GeV energy scale. Due to the broad range of energies generated by present neutrino beam facilities, as well as the complexity of inferring the incident neutrino energy from each detected configuration, oscillation measurements can be sensitive to various details of the interaction model used in the analysis. While considerable research effort is being directed toward

improving the interaction simulations used by the neutrino community, modeling deficiencies remain problematically large even in state-of-the-art codes. As a result, experimental neutrino collaborations typically rely on data-driven mitigation strategies in order to achieve the needed measurement precision.

A common approach for addressing the mismodeling of neutrino scattering has been empirical tuning of simulation parameters to cross-section data. Using predictions from the popular GENIE configuration generator [5,6] as a starting point, the MINERvA [7], NOvA [8], and MicroBooNE [9] collaborations have all developed their own tuned neutrino interaction models in recent years. This experiment-specific work has been pursued in parallel with more generic tuning studies carried out by the GENIE collaboration itself [10–13]. Although such tuning must be executed carefully, both to ensure robust oscillation results [14] and to avoid obscuring potential signatures of exotic physics [15], this general technique has provided a successful foundation for numerous modern results in neutrino physics. In light of its established track record and the notable difficulty of comprehensive, *a priori* modeling of nuclear effects at GeV energies, empirical tuning of neutrino interaction simulations is likely to continue as standard practice in the field. However, as

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

the precision goals of future oscillation analyses become increasingly stringent, the complexity of related tuning campaigns will necessarily grow in response. More detailed input data and a larger number of tuned parameters may both be required in the future to achieve satisfactory simulation performance, and these constraints will in turn lead to greater computational challenges.

To address these anticipated future needs, artificial intelligence and machine learning techniques may provide especially powerful solutions. As an example particularly well-suited to this use case, we consider simulation-based inference (SBI), which has gained popularity across scientific disciplines over the last several years. SBI includes a broad group of methods that leverage simulators as a part of the statistical inference procedure, where the goal is to infer the likelihood or posterior distribution of underlying parameters for a given experiment or observation [16,17]. SBI methods are valuable in situations where forward simulations are available, but the likelihood is unknown or intractable (e.g., when using a complex model with hundreds or even thousands of parameters), and hence explicit likelihood calculations are not feasible.

Traditional SBI methods have been present for some time, and include for example, approximate Bayesian computation [18] and approximate frequentist computation [19], which are closely related to the traditional template histogram and kernel density estimation approaches. More recently, SBI algorithms suitable for high-dimensional parameter spaces have been developed, which utilize deep neural networks as surrogates for modeling the conditional probability densities of the likelihood or posterior distribution [16].

Deep learning-based SBI models have already been successfully used in different areas of physics. In astrophysics and cosmology, these methods have been used for the dark matter substructure inference in galaxy-galaxy strong lenses [20–22], inference of strong lensing parameters [23–25], inference of galaxy properties from spectra [26], cosmology inference from galaxy cluster abundance [27], inference of the Hubble constant from binary neutron star mergers [28], etc. In collider physics [29], SBI has been used for constraining the Higgs potential for di-Higgs production [30], searching for CP violation in associate W and Higgs boson production [31], measuring QCD splittings [32], etc.

Building on these precedents from other areas of physics, as well as recent work on SBI for modeling the hardware response of the JUNO neutrino detector [33], in this article we examine the suitability of SBI as a tool for neutrino interaction model tuning. For this application, SBI provides an efficient alternative to conventional methods, as it offers amortized inference, i.e., after the upfront cost of training the model is paid, inference can be performed in seconds. To provide a realistic context for our exploration of SBI-based tuning, we revisit the tuned GENIE model adopted by the MicroBooNE collaboration [9]. Using a software framework called NUISANCE [34], MicroBooNE

collaborators adjusted four GENIE simulation parameters in comparison to an external neutrino cross-section measurement reported by the T2K experiment [35]. Our choice of this *MicroBooNE Tune* is motivated by its relative simplicity among other representative examples in the field, its continuing use in the latest MicroBooNE analyses¹ [36,37], and recent scrutiny from the community into the robustness of its predictions and underlying tuning methodology [38]. In particular, the nontrivial fitting procedure MicroBooNE had to use to obtain a satisfactory fit result makes this tune an interesting case-study for new fitting methods, so that it can be verified whether they are affected by the same pathologies and whether they converge to similar results or not.

In the remainder of this article, we demonstrate that our SBI algorithm can (1) infer the correct parameter values when confronted with mock data generated using the MicroBooNE Tune configuration of GENIE,² (2) describe the T2K data with equal or better precision than the MicroBooNE Tune while overcoming issues with Peelle’s Pertinent Puzzle [40,41] encountered in the original fitting procedure, and (3) adjust the input GENIE model to approximate the predictions of NuWro [42], an alternative neutrino configuration generator that seeks to model the same physics processes as GENIE. These results pioneer the use of SBI to obtain configuration generator parameter values from actual experimental data sets, thus establishing a novel approach, potentially more powerful and scalable, for tuning neutrino interaction simulations. In Sec. II we describe our training data, and in Sec. III we document the SBI algorithm, its neural network architecture, and our training procedure. In Sec. IV, we benchmark the performance of our SBI algorithm, and in Sec. V we show the results of our fits to T2K data and a NuWro prediction. Finally, in Sec. VI, we provide a summary and conclusions from our work.

II. TRAINING DATA

We simulate neutrino–nucleus interactions using version 3.4.2 of the GENIE framework³ [5,6]. Within this interaction model, we vary the four parameters that were adjusted in the MicroBooNE Tune, namely,

- (i) θ_1 (MaCCQE). This parameter is used in the calculation of the charged-current quasielastic (CCQE) cross section. It represents the mass that appears in a

¹While the MicroBooNE Tune interaction model remains unaltered, additional data-driven, analysis-specific model constraints are applied in some cases.

²See also our earlier demonstration of this capability in Ref. [39].

³In Ref. [9], MicroBooNE used version 3.0.6 of GENIE. Comparisons of the cross-section predictions relevant to this study revealed good agreement across code versions. Exact reproduction of the prior MicroBooNE results is not a goal of the present work.

dipole parameterization of the axial-vector form factor of the nucleon. It has an important impact on the normalization CCQE configurations, as well as the dependence of the cross section on the negative square of the four-momentum transfer Q^2 .

- (ii) θ_2 (NormCCMEC). This parameter scales the overall normalization of the charged-current multi-nucleon interaction cross-section arising from meson exchange currents (CCMEC).
- (iii) θ_3 (XSecShape_CCMEC). This parameter controls the shape of the CCMEC cross section within the two-dimensional phase space of leptonic energy transfer and three-momentum transfer. As the parameter value moves between zero and one, GENIE interpolates between the shape predictions from two different cross-section models. Parameter values outside of this range are invalid.
- (iv) θ_4 (RPA_CCQE). This parameter controls the strength of the Random Phase Approximation (RPA) correction applied to CCQE processes. It has an important impact on both the normalization and the shape of the cross section in Q^2 .

Further information on the implementation and physical meaning of these parameters is given in the original MicroBooNE Tune publication [9]. To generate training data for our SBI model, we created an ensemble of configurations in which the four parameters were independently sampled from uniform distributions with the following ranges⁴:

$$\begin{aligned}\theta_1 &\in [0.528, 1.39] \text{ GeV}, & \theta_2 &\in [0, 3.0], \\ \theta_3 &\in [0.0, 1.0], & \theta_4 &\in [0.0, 1.5].\end{aligned}$$

These ranges were chosen to respect physical validity, provide coverage of the parameter space in the vicinity of the best-fit values from the MicroBooNE Tune, and allow for significant variations around them. We made the following specific choices for each parameter:

- (i) θ_1 (MaCCQE): The upper bound was chosen as a whole number in GENIE's internal units that covers the central values of historical fits to MiniBooNE data [43]. These fits neglected the multinucleon contribution entirely, so we regard them as a fairly conservative upper bound. Downward variations have a symmetrical cutoff about the nominal value from the GENIE base model (G18_10a_02_11a) of $M_A = 0.961242$ GeV [9]. However, we consider the low end of the range implausible based on virtually all evaluations of this parameter in the literature.

⁴Values of these parameters are consistently reported in this article in physical units. Different units are used internally by the GENIE implementation.

- (ii) θ_2 (NormCCMEC): Allow nearly 3σ coverage around the MicroBooNE result while forbidding unphysical negative values.
- (iii) θ_3 (XSecShape_CCMEC): Allowed to vary over its full physical range.
- (iv) θ_4 (RPA_CCQE). A substantially stronger RPA correction would likely be disfavored by other neutrino data, so we cap the upper range at $\theta_4 = 1.5$, which covers a $+1.5\sigma$ variation of the MicroBooNE Tune. Since some competing models do not include this effect, we allow low values down to the physical bound of 0 (no RPA correction at all).

For each configuration of the parameters, we processed the output of GENIE with NUISANCE [34] to produce a 58-bin histogram of the joint muon energy and muon angle distribution, representing a theoretical prediction that can be directly compared to the “Analysis I” T2K neutrino interaction data set reported in Ref. [35]. For the original MicroBooNE Tune, this procedure was used together with a subsequent likelihood fit to the measured histogram to obtain best-fit parameter values. In this study, we created a training set of 171,747 configurations and a test set of 1,000 independent configurations to test our SBI workflow. Throughout this paper, we use the term *configuration* to denote a single $(\theta, \mathbf{x})$ pair, where θ represents one choice of the four GENIE parameters and $\mathbf{x}$ the corresponding predicted T2K-binned histogram. Data used in this project is publicly available on Zenodo.⁵

III. METHODS

In traditional explicit Bayesian inference, the posterior $p(\theta|\mathbf{x})$ is given by Bayes's theorem as,

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta) * p(\theta)}{p(\mathbf{x})}, \quad (1)$$

where $\mathbf{x}$ denotes the data vector, θ is the set of model parameters to be inferred, $p(\mathbf{x}|\theta)$ is the likelihood, $p(\theta)$ is the prior on the parameters, and $p(\mathbf{x})$ is the evidence (or marginal likelihood). However, in the case of many complex scientific problems, the likelihood function is unknown or computationally intractable (due to high model complexity), making explicit likelihood calculation impossible. To address these challenges, SBI (also referred to in some papers as implicit likelihood inference or Likelihood-Free Inference) methods have gained popularity in recent years. The most recent and advanced SBI methods utilize deep learning models to directly create surrogates for the likelihood [44], likelihood ratio [45] or posterior distribution [46,47] in a given inference

⁵<https://zenodo.org/uploads/18747090>.

problem. These models are trained using pairs of input model parameters θ and model outputs, i.e., simulated data x . Due to high model capacity and amortized inference, deep-learning-based SBI methods opened doors for inference with uncertainty quantification even in high-dimensional parameter spaces [16].

A. Neural posterior estimation

In this work, we focus on direct inference of parameter posteriors from the data $p(\theta|x)$, i.e., neural posterior estimation (NPE). We perform one round of the NPE training based on the NPE-A method from [46]. Note that there are two extensions of this method (NPE-B [48] and NPE-C [47]), which utilize sequential training. These sequential NPE approaches utilize multiple rounds of training, where the posteriors inferred in one round are used as proposal priors for the next round. Unfortunately, in Ref. [49] it was shown that sequential approaches often produce overconfident posteriors. To utilize the full benefits of amortized inference and to avoid potential issues with our inferred posteriors, we limit our initial analysis to only one round of NPE training.

In NPE, a neural network with weights ϕ , is trained on M parameter-data pairs i.e., configurations $\{\theta_j, x_j\}_{j=1}^M$ from the simulator, to approximate the posterior $p(\theta|x)$. Again, note that in our case each θ is actually a set of four parameters $\theta = \{\theta_i\}_{i=1}^4$ described in Sec. II. If we denote this neural network-based approximate posterior with $q_\phi(\theta|x)$, we can then train the neural network by maximizing the average log-likelihood with respect to ϕ ,

$$\mathcal{L}(\phi) = \frac{1}{M} \sum_j \log q_\phi(\theta_j|x_j). \quad (2)$$

If $q_\phi(\theta|x)$ is a sufficiently flexible model, and $M \rightarrow \infty$, this approximate posterior should converge to the true posterior $p(\theta|x)$ when the average log-likelihood is maximized. In other words, by using the negative average log-likelihood as a loss function, we can train the neural network on the training parameter-data pairs to approximate the true posterior.

To build our NPE, we utilize masked autoregressive flows (MAFs) [50], due their state-of-the-art flexibility in modeling complex density distributions. MAFs represent a fusion between two families of deep learning density estimators—autoregressive models [51] and normalizing flows [52]. Autoregressive density estimators utilize the fact that any n -dimensional probability density $p(\mathbf{v})$, where $\mathbf{v} \equiv [v_1, \dots, v_n]$, can be factorized into a product of 1-dimensional conditional probability distributions by using the chain rule. On the other hand, normalizing flows focus on transforming samples from a simple base density, like a standard normal distribution, into samples from a more complex target density by using invertible

transformations. Hence, a particular type of autoregressive models, which are parametrized as Gaussian conditional probabilities, can also be thought of as belonging to the normalizing flow family of models. In this case, one autoregressive block will contain conditional probabilities which can be written as,

$$p(v_k|\mathbf{v}_{1:k-1}) = \mathcal{N}(v_k|\mu_k, (\exp \sigma_k)^2), \quad (3)$$

where $p(v_k|\mathbf{v}_{1:k-1})$ is the k th conditional, conditioned on the subset of the vector containing data points with indices less than k , i.e., $\mathbf{v}_{1:k-1}$. Additionally, μ_k and σ_k are the mean and log standard deviation of the Gaussian that v_k is sampled from, and are themselves scalar functions of previous data points $\mathbf{v}_{1:k-1}$,

$$\mu_k = f_{\mu_k}(\mathbf{v}_{1:k-1}), \quad (4)$$

$$\sigma_k = f_{\sigma_k}(\mathbf{v}_{1:k-1}). \quad (5)$$

To allow for more flexibility, we can then create stacks of these autoregressive model blocks, such that samples of the output density of the last block are used as the input for the next block. In this case, the initial input density (the standard normal distribution) is progressively transformed into samples from increasingly more complex distributions, which are then used as input densities for the following autoregressive model blocks to produce even more complex output densities, until we reach the final target data distribution $\mathbf{v}$. This stacking greatly increases the flexibility of the density model, allowing application to complex scientific problems.

Finally, MAFs use a masked autoencoder for distribution estimation (MADE) [53] to encode the autoregressive model blocks described previously. The main advantage of MADE over other autoregressive model architectures is that it avoids recursive computation. Each MADE block represents a neural network which utilizes binary masks applied to some of the network weights, such that in each of the autoregressive blocks, the functions f_{μ_k} and f_{σ_k} for all k -s can be calculated efficiently in a single forward pass. For more technical details about MADE, we refer the reader to Ref. [53], and about MAFs to Ref. [50].

B. Model architecture and training

The aim of this work is to infer the underlying set of parameters used within GENIE from a histogram generated according to the approach described in Sec. II. We utilize the SBI framework developed by mackelab,⁶ implemented in Python. Our SBI model takes as input the four physics parameters discussed in Sec. II, θ_i , along with their associated histograms, $\mathbf{x}$, to learn the

⁶<https://github.com/sbi-dev/sbi>.

inverse mapping from histograms to parameters, i.e., we aim to directly learn the posteriors of the θ_i parameters $p(\theta_i|x)$.

To improve efficiency in learning the posterior distribution, we employ a three-layer fully connected embedding network with ReLU activations after the first two layers. The hidden layers contain 45 and 32 neurons, respectively, progressively reducing the input dimensionality from 58 to a 20-dimensional output embedding.⁷ These embeddings are then used as inputs to the NPE, for which we use the MAF architecture with three MADE blocks and 55 hidden features in each. Both the embedding network and the MAF are trained together to allow the embedding network to learn the most informative summaries, which will lead to the best posterior predictions. The code used in this project can be found in our GitHub repository.⁸

During SBI model training, we use minibatches of size 512, a learning rate of 10^{-2} , and a training/validation split of 90%/10% to monitor and mitigate overfitting. We also use an early stopping criterion, with patience of 50 epochs. The training converges in an average of 300 epochs, running approximately 7 minutes in a regular CPU environment. We use a fixed random seed set to 555 for reproducibility, but confirm that changing the seed does not present significant changes to the SBI performance, calibration, or the final fits.

IV. MODEL PERFORMANCE AND CALIBRATION

The performance of the NPE for an individual randomly chosen test configuration—i.e., one (θ, x) pair drawn from the 1,000-configuration test set defined in Sec. III—is illustrated in Fig. 1. The diagonal panels show the one-dimensional marginal posterior distributions for each parameter, and the off-diagonal panels display the corresponding two-dimensional joint posteriors. The contours represent the 68% and 95% confidence levels, enclosing the highest posterior density intervals under the assumption of a well-behaved posterior distribution. The vertical and horizontal dashed lines denote the corresponding true parameter values used to generate the plotted test data point. Overall, the posteriors are centered near the target values within the 68% interval. The shape and orientation of the contours provide insight into the potential correlation among the parameters (elongated and tilted contours indicate correlations, while circular contours suggest independence). We do not see any

⁷We note that when the full 58-dimensional histograms are used as input, the model tends to become overconfident; however, reducing the input dimensionality to fewer than 16 parameters leads to a loss of precision.

⁸https://github.com/karlaTame/SBI_parameter_tuning.

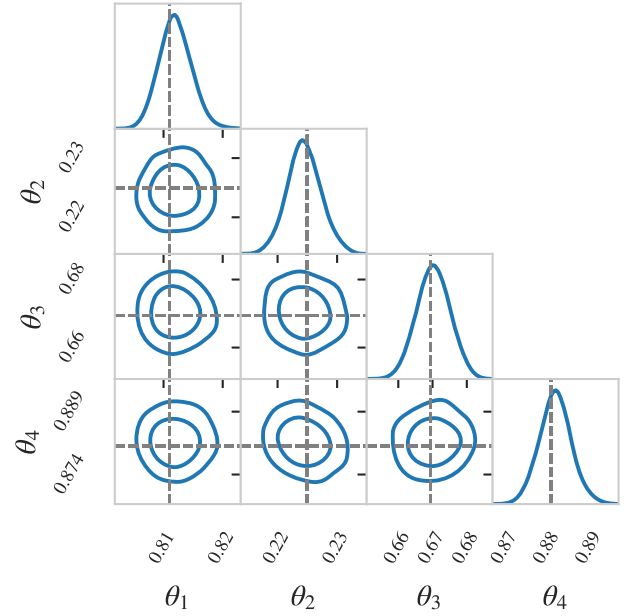


FIG. 1. Neural posterior estimate of the four parameters for a single test configuration. The gray dashed lines indicate the true values while the blue contours represent the estimates at 68% and 95% confidence levels.

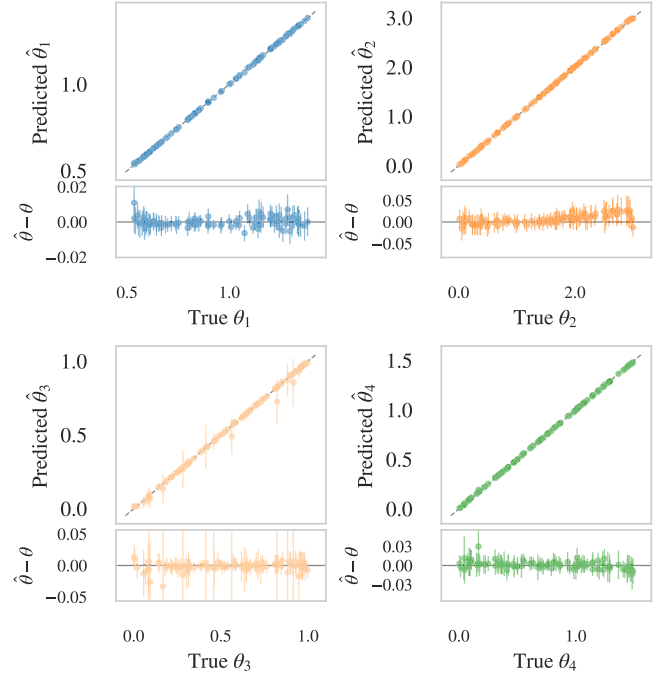


FIG. 2. Comparison of true and predicted values for each parameter, shown for a subset of 200 test configurations with the predicted 1σ error bars. Top subpanels: predicted vs true with the dashed line indicating perfect recovery. Bottom subpanels: residual predicted minus true vs true, enlarged to make individual error bars visible. The residuals are consistent with zero within 1σ for all four parameters.

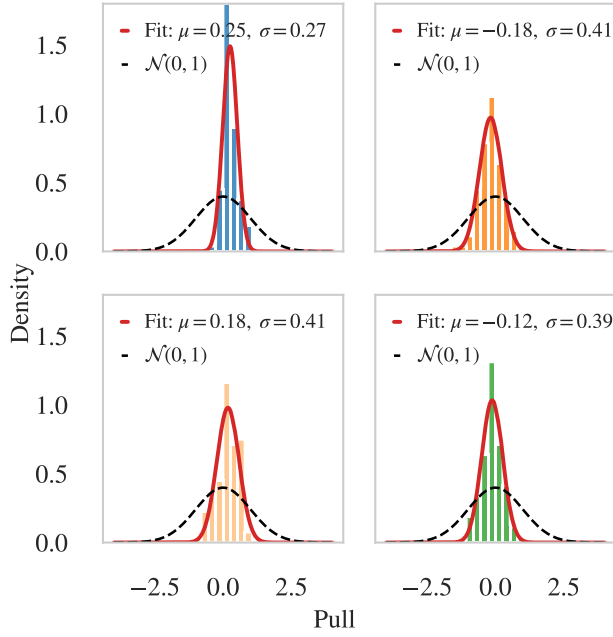


FIG. 3. Histograms of pulls for 1000 test configurations for each parameter. All four parameter distributions (red solid line) are centered close to zero and somewhat narrower than the standard normal distribution (black dashed line), suggesting that the uncertainties may be overestimated.

correlation in our inferred posteriors, which is also evident on this randomly chosen test configuration.

To assess the overall performance of the model, in Fig. 2 we show the comparison between the true θ_i^{true} and predicted best fit values for each parameter $\hat{\theta}_i$ (where $i = 1, 2, 3, 4$), on a sample of 200 configurations randomly chosen from our test set. Each point corresponds to an individual configuration prediction, with its associated 1σ error bars for each parameter (too small to be visible on the plot for all parameters except θ_3), shown in different colors. All parameters are accurately reconstructed and lie close to the true values, with θ_3 exhibiting somewhat larger deviations; nevertheless, in all cases, the true value remains within the predicted 1σ uncertainties.

We also utilize pull distributions to more closely examine how the NPE predicted uncertainties behave. In Figure 3 we present the pull distributions for each parameter, computed using the full test set of 1000 configurations. The pulls are defined as

$$\text{pull}_j = \frac{\hat{\theta}_{i,j} - \theta_{i,j}^{\text{true}}}{\sigma_{i,j}}, \quad (6)$$

where $\sigma_{i,j}$ is the predicted NPE one-sigma error for each element of the test set, $j = 0 \dots 1000$, and $i = 1, 2, 3, 4$ corresponds to each of the four GENIE parameters. The color bars are the histograms for the pull for each

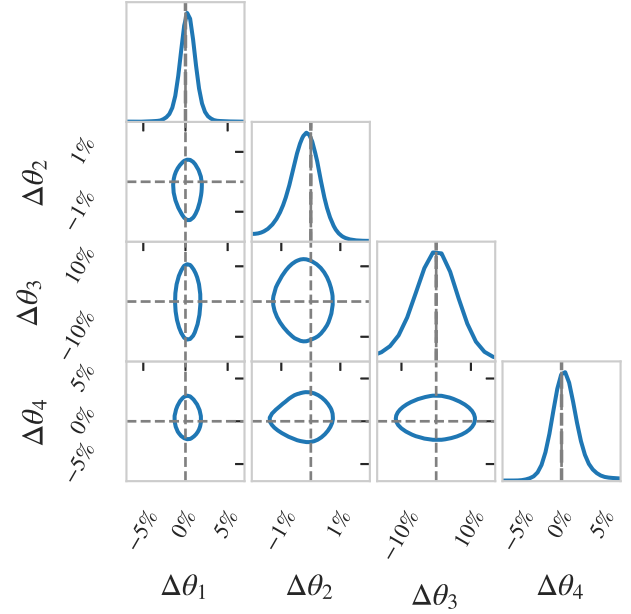


FIG. 4. Residual distribution (in percentages) of four parameters for 1000 test configurations. The gray dashed lines indicate the zero value (perfect prediction). The distributions of residuals are very narrow and show no visible biases, i.e., they are all centered around zero. Additionally, all deviations are within a few percent. The blue contours represent the 68% confidence regions.

parameter, the red solid curves represent Gaussian fits to the pulls, with the extracted mean μ and width σ indicated in the legend of each subplot, while the black dashed curves show the standard normal distribution $N(0,1)$ expected for an unbiased and well-calibrated inference. We observe that all four distributions are centered close to zero but somewhat narrower than the standard distribution width, suggesting that the uncertainties may be overestimated.

To further assess the overall performance of the model, in Fig. 4 we show the distribution of the residuals (in percentages) between model's best-fit estimate $\hat{\theta}_i$, and true parameter value θ_i^{true} i.e., $\Delta\theta_i = (\hat{\theta}_i - \theta_i^{\text{true}})/\theta_i^{\text{true}}$ for $i = 1, 2, 3, 4$, computed over the same sample of 1000 independent test configurations. Each diagonal panel displays the one-dimensional distribution of residuals for an individual parameter, while the off-diagonal panels illustrate the joint distributions between pairs of residuals. The dashed lines indicate zero bias. We observe that all residuals are centered around zero with narrow widths (up to a few percent, with only $\Delta\theta_3$ extending up to around 10%), indicating that the model yields unbiased estimates with low variance.

In Fig. 5, we present the posterior coverage for each parameter θ_i , evaluated again over the same 1000 test configurations. The solid colored lines represent the empirical coverage for each parameter, indicating the fraction of configurations where the true value lies within

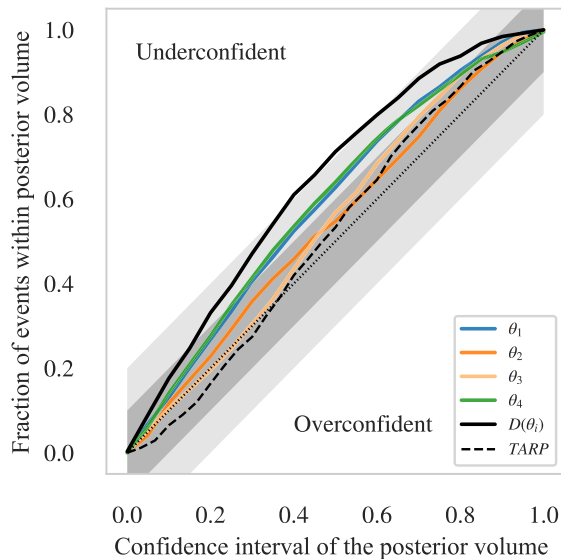


FIG. 5. Posterior coverage of the θ_i parameters for 1000 test configurations (colored solid lines). $D(\theta_i)$ and TARP curves show combined model performance over all parameters, and are plotted with solid and dashed black lines, respectively. The diagonal black dotted line indicates perfect uncertainty calibration. The gray regions indicate thresholds of 10% (dark gray) and 20% (light gray) uncertainty miscalibration. All curves are within the 20% band and are showing slight underconfidence, which is preferred compared to an overconfident model with too narrow error bars.

a given posterior confidence interval. The black solid curve represents the combined performance of the model on all parameters, for which we utilize a distance metric $D(\theta_i)$ from [25,54]. $D(\theta_i)$ combines parameter values from posterior samples into a single objective function that takes into account the covariance between different θ_i parameters. The black dashed line shows another common global diagnostic called the truncated acceptance ratio posterior (TARP) [55]. Given the posterior samples predicted by the model, the TARP diagnostic compares them to the corresponding ground-truth parameters by evaluating whether the true values fall within posterior credible regions at different confidence intervals. For each level, the method computes the fraction of true parameters contained within the inferred regions. We use the TARP implementation provided in the `sbi` diagnostics framework [56]. The black dotted diagonal corresponds to perfect uncertainty calibration, where the predicted confidence intervals exactly match the empirical coverage. The shaded bands around the diagonal (dark and light gray) denote 10% and 20% tolerance regions, respectively, for quantifying miscalibration. All of our inferred parameters are within the 10%, with θ_1 and θ_4 leaking to the 20% tolerance band for high posterior volumes. Note that all our parameters are underconfident, which is generally preferable to overconfidence, and it allows room for further improvement. More importantly,

overconfidence would indicate unwarranted certainty in potentially incorrect results, thereby reducing the reliability of the model's predictions. These conclusions are in line with the conclusions from the pulls (Fig. 3), which also show that the predicted errors are wider than what is necessary. Furthermore, even though the predicted posteriors of the individual parameter are wider than they need to be, this is not actually a problem for the analysis, as the largest uncertainties in the measurement come from the experimental errors (more details in Sec. V).

V. METHOD VALIDATION AND PARAMETER MEASUREMENT IN DATA

Our goal is to apply our SBI algorithm to determine optimal values of the four GENIE simulation parameters originally tuned by MicroBooNE [9]. These parameters control portions of GENIE's cross-section calculation for nucleon knockout via the CCQE (θ_1 , θ_4) and CCMEC (θ_2 , θ_3) channels. Their best-fit values were determined using a double-differential measurement from T2K of charged-current interactions of muon neutrinos without pions in the final state [35].⁹ This T2K result is based on data collected by the ND280 near detector, which used a plastic scintillator target and was exposed to a beam of neutrinos with an average energy of 0.6 GeV at a direction offset by 2.5° from the beam axis.

Before performing the actual data fit, we carried out two validation steps. First, we verified that our NPE model is able to reproduce the correct parameters when presented with the nominal histogram obtained from the MicroBooNE Tune configuration of GENIE. This test is equivalent to what is shown in Fig. 1, but choosing a configuration that was obtained by fitting our target data set from T2K. As expected, our NPE model reproduces the MicroBooNE Tune parameters almost perfectly, which can be seen in the first two rows of Table I. To compare these numbers, we also plot the original MicroBooNE Tune values from [9] and the NPE inferred MicroBooNE values in Fig. 8, as red and violet points, respectively.

As a next step, we consider as input a cross-section prediction for the T2K measurement calculated using the default configuration of version 25.03.2 of the NuWro generator [57]. Goals of this test are, on one side, to check our workflow with input not produced by GENIE before presenting it with the actual T2K data and, on the other side, to verify how well SBI-based modifications to our four GENIE parameters could perform as a surrogate model for NuWro. The latter goal is particularly relevant since, due to the limited capability of current experimental workflows to support large-scale simulations with

⁹Two cross-section measurements are reported in Ref. [35]. Following MicroBooNE, we use data from “Analysis I” for this study.

TABLE I. Inferred parameters, their uncertainties, and the corresponding χ^2 values obtained when comparing with the T2K measurement. Some values are also presented in Fig. 8: MicroBooNE Tune (red), Inferred MicroBooNE Tune (violet), and Inferred T2K (blue). The agreement between the original and NPE fits is excellent, both for the MicroBooNE Tune and NuWro. The NPE fit to the T2K data provides the overall best χ^2 . Not all θ parameters are shared between the NuWro and GENIE physics models, so we omit their values in the NuWro row of the table.

Model	θ_1	θ_2	θ_3	θ_4	σ_{θ_1}	σ_{θ_2}	σ_{θ_3}	σ_{θ_4}	χ^2 (T2K data, 58 bins) ^c
MicroBooNE Tune ^a	1.10	1.66	1.00	0.85	± 0.10	± 0.50	$+1.00(-0.00)$	± 0.40	115.10
Inferred MicroBooNE Tune ^b	1.09	1.67	0.94	0.85	± 0.003	± 0.01	± 0.01	± 0.007	113.07
NuWro	...	...	...	...	...	...	...	...	126.73
Inferred NuWro ^b	1.10	0.70	0.09	0.57	± 0.003	± 0.015	± 0.013	± 0.007	126.93
Inferred T2K ^a	1.04	1.85	0.65	1.07	$+0.11(-0.12)$	$+0.55(-0.24)$	$+0.24(-0.18)$	$+0.27(-0.30)$	107.29
Traditional χ^2 fit ^d	0.75	0.66	1.00	1.00	± 0.07	± 0.19	± 0.09	± 0.0	63.82

^aThe σ errors correspond to experimental errors.

^bThe σ errors correspond to one standard deviation of the NPE predicted posteriors.

^c χ^2 values reported here include all measurement bins shown in Figs. 6 and 10. The full data covariance matrix (including off-diagonal elements) is used in the χ^2 calculation.

^dThe σ errors in this row are taken directly from the numerical estimates provided in the output of NUISANCE. Despite being included in the fit, θ_4 remained unchanged from its nominal value and had an estimated error of zero.

multiple neutrino configuration generators, the prospect of a computationally cheap method for approximating the predictions of one generator with another is worthy of study. A first solution to this problem, proposed by MINERvA, is to utilize an approach based on boosted decision trees (BDTs) to derive event weights [58]. However, BDT scores lack interpretability, so it is worth exploring approaches that can provide insight into underlying differences in terms of physics dynamics. The values of the GENIE model parameters inferred from an input cross-section histogram calculated with NuWro are shown in the fourth row of Table I. Although the inferred GENIE approximation to NuWro is not a perfect

match, we note that the overall χ^2 metric for goodness-of-fit with the T2K data is nearly identical between NuWro and its GENIE surrogate model. Bin-by-bin visual agreement can be qualitatively assessed between the original NuWro prediction and the “Inferred NuWro” surrogate model in Fig. 6. Each plot in the figure shows the muon momentum distribution measured by T2K (black points) within a particular range of $\cos\theta$, the cosine of the angle between the incoming neutrino and outgoing muon. The two angular bins shown represent a portion of the full data set used in our study, with the remainder plotted in Appendix. The prediction of the original MicroBooNE Tune model is

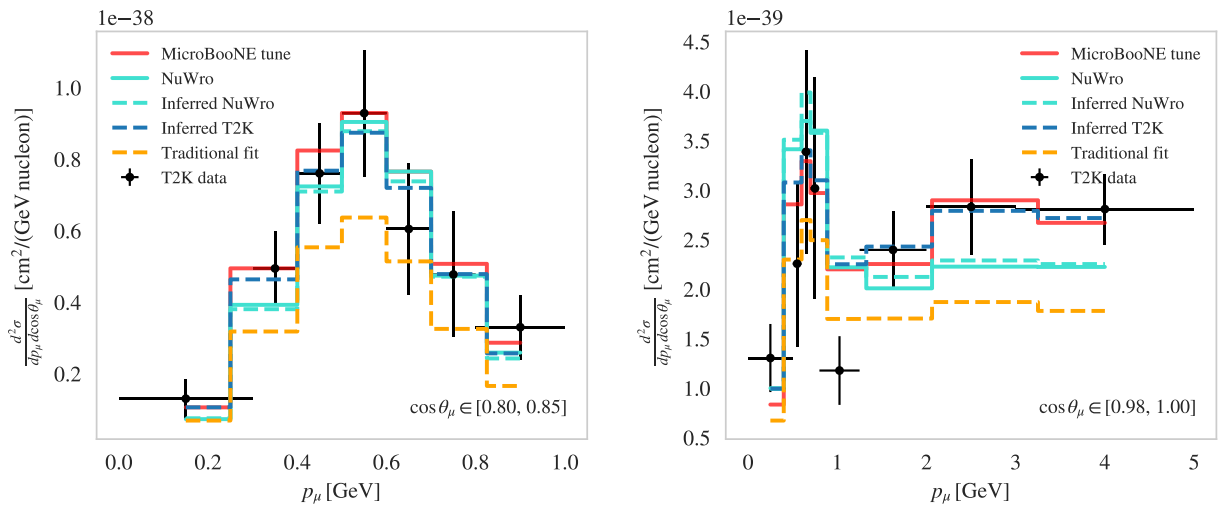


FIG. 6. Comparison of the T2K cross-section data (black points) to the NuWro [57] (solid cyan), NPE-inferred NuWro (dashed cyan), MicroBooNE Tune [9] (solid red), NPE-inferred T2K (dashed blue), and traditional χ^2 fit (dashed orange) model predictions in two different slices of $\cos\theta$.

plotted as the solid red line. The NuWro and Inferred NuWro model predictions are drawn as solid and dashed cyan lines, respectively. While the quality of the agreement between these last two models varies across bins of the measurement, in general, we find that the surrogate model mimics the original NuWro prediction reasonably well. We attribute residual differences between the original and inferred NuWro results to two sources. First, the GENIE parameters used in the MicroBooNE Tune may imperfectly represent relevant degrees of freedom considered in the NuWro model. In particular, it seems unlikely that any allowed value of θ_3 would be able to fully reproduce the differential shape of the NuWro cross-section for the two-nucleon knockout process. Second, there are known differences between other aspects of the GENIE and NuWro physics models that are unaffected by the parameters studied here. An important example for the T2K data set of interest is the different approach taken by each code to intranuclear absorption of pions produced in inelastic neutrino-nucleon interaction channels. Despite these caveats, the level of agreement achieved here between the NuWro and Inferred NuWro predictions clearly motivates a thorough pursuit of SBI-based surrogate models for neutrino event generators.

In Fig. 6 and the accompanying plots in Appendix, we also plot in dashed blue the cross-section prediction obtained with our SBI algorithm using the actual T2K measurement as input. The inferred parameter values and χ^2 metric for this SBI-based GENIE prediction are also shown in the “Inferred T2K” row of Table I. While the dashed blue prediction shows the result of running inference directly on the central-value data points reported by T2K, a full covariance matrix describing the measurement uncertainties is also provided in the collaboration’s data release. In the original MicroBooNE Tune publication [9], the authors reported difficulty in using the full covariance matrix within their tuning algorithm, which was based on numerical minimization of χ^2 in repeated comparisons to the data. Ultimately, a common pathology of similar procedures, known as Peele’s Pertinent Puzzle (PPP) [41], led MicroBooNE to neglect the off-diagonal covariance matrix elements in their final fit, effectively ignoring the significant measurement correlations reported by T2K. While this choice was cross-checked in the original analysis using a different technique involving a “norm-shape” transformation of the covariance matrix [59,60] (which yielded a compatible fit), MicroBooNE’s approach has been shown to yield poor results when applied to other datasets [38].

Details of the initial attempt by MicroBooNE to perform a fit to the T2K data using the full, unaltered

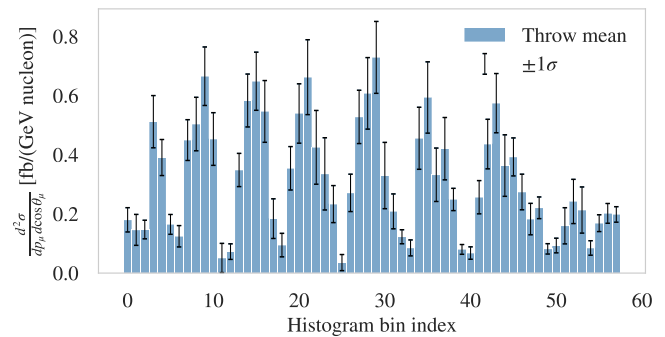


FIG. 7. Variation of the T2K histogram over one thousand samples drawn from the covariance matrix around the central T2K value. The bars show the mean per bin and the vertical error bars indicate the corresponding 1σ spread.

experimental covariance matrix were not reported in Ref. [9]. For comparison to our SBI-based analysis, the final row of Table I shows our own fit results obtained using such a procedure. The corresponding differential cross section is plotted in Fig. 6 and in Appendix using a dashed orange line. Although this “traditional” fit yields a χ^2 value that is much lower than all other models studied (unsurprising since this metric is directly minimized in repeated comparisons to the T2K data), the normalization of the resulting prediction (and the fitted value of θ_1 in particular) is implausibly low. This is a clear example of PPP: because the four model parameters of interest do not provide sufficient freedom to fit the shape of the T2K data well, the best χ^2 score is obtained by pathologically suppressing the overall normalization.

The general problem of PPP for interaction model tuning is widely recognized in the neutrino community, and discussions continue on optimal methods for addressing it [40]. While other solutions to PPP are possible [60,61], a strength of our approach is that the issue is sidestepped entirely; inference is performed each time using a single histogram without associated uncertainties, so the problematic correlations do not enter into the calculation. It is true that the SBI procedure cannot overcome the inadequate shape freedom in the four-parameter variations of the GENIE model studied (thus yielding a χ^2 that is too high for a satisfactory fit). However, it does not suffer from a catastrophic normalization bias; the parameter values obtained with SBI are seen to align far better with prior expectations than those from the traditional χ^2 fit.

To propagate the measurement uncertainties into our SBI results while preserving correlations, we first generated an ensemble of one thousand variations of the T2K measurement using random throws from a multivariate zero-mean Gaussian distribution using the

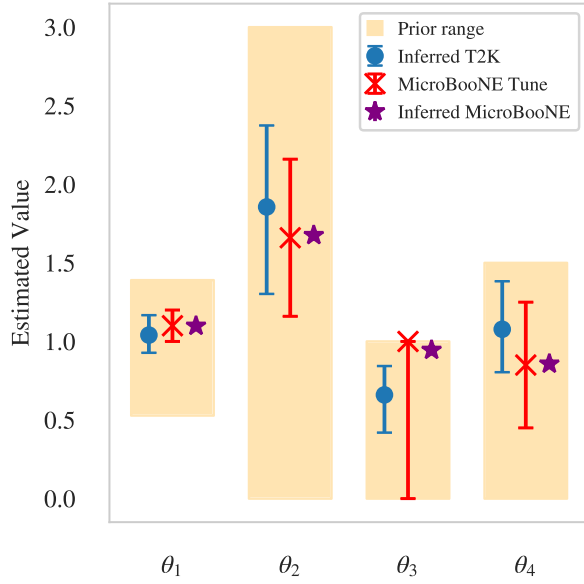


FIG. 8. Red points correspond to the MicroBooNE fit parameters and uncertainties reported in Ref. [9]. In blue, we display the four parameters θ_i inferred by our network using T2K data as input, together with their corresponding error bars, computed using throws from the data covariance matrix as described on the main text. The orange bands indicate the prior ranges employed to train the SBI, as defined in Sec. II. As a validation, the violet points show the parameters inferred using the exact MicroBooNE histogram, exhibiting excellent agreement (with very small error bars, which are not visible on the figure) and thus corroborating the validity of our analysis. Overall, we observe a very good correspondence between the T2K-inferred parameters and the true MicroBooNE values.

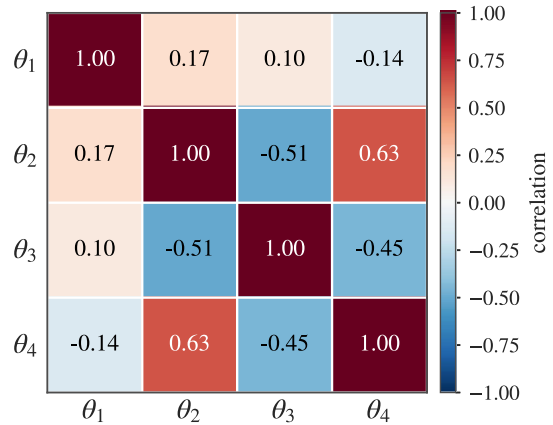
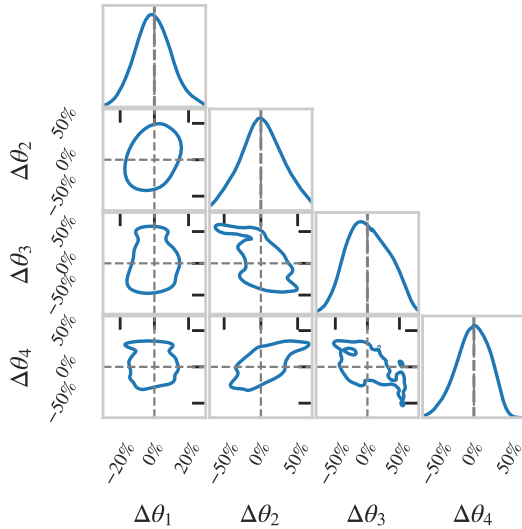


FIG. 9. LEFT: Residual distribution (in percentages) of four parameters for 1000 input T2K data histograms generated via throws of the full covariance matrix (including all correlations). The gray dashed lines indicate the zero value (exact agreement with the result of inference on the central-value T2K data histogram). The two-dimensional contours represent 68% confidence levels. RIGHT: Parameter correlation matrix constructed from the residual distribution.

data covariance matrix. Figure 7 shows the mean (solid line) and the corresponding standard deviation (shaded band) obtained from this procedure. To plot the entire two-dimensional measurement on a single axis, we show the cross-section as a function of histogram bin index, which ranges from 0 to 58.

In Fig. 8, the inferred parameter values obtained using the T2K central values are shown as blue dots. Here we plot the corresponding 1σ error bars, also in blue, which are computed as follows. First, we compute the distribution of deviations between each prediction obtained from the covariance matrix histogram throws, $\hat{\theta}_{i,j}^{\text{T2K}}$ and the T2K central value $\hat{\theta}_i^{\text{T2K}}$. For each sample i , we define

$$\Delta\hat{\theta}_{i,j}^{\text{T2K}} = \hat{\theta}_{i,j}^{\text{T2K}} - \hat{\theta}_i^{\text{T2K}}, \quad (7)$$

where again $j = 0, \dots, 1000$ and $i = 1, 2, 3, 4$. Then, the full set of deviations is stacked into an array, from which the median and the 16th and 84th percentiles are calculated along each parameter axis. These percentiles are then used to define asymmetric 68% confidence levels corresponding to the 1σ error bars. As expected, these error bars are much wider than errors coming from a single predicted posterior from the NPE model (e.g., for a single test configuration as in Fig. 1). Numerical values of the blue 1σ uncertainties are reported in the final row of Table I. Figure 8 also shows that the inferred values (blue points with error bars) are in close agreement with the MicroBooNE Tune [9], indicated in

red, with all our inferred values being within MicroBooNE error bars. We also show the prior range (orange bands) for all θ_i parameters used to generate the training data for our NPE model. Finally, the violet points show the parameters inferred using the exact MicroBooNE histogram, which are in excellent agreement with true MicroBooNE parameters.

In the left-hand plot of Fig. 9, we show the NPE posterior distributions for the deviations $\Delta\hat{\theta}_{i,j}^{\text{T2K}}$ computed according to Eq. (7) for the T2K cross-section measurement. The distributions are all maximum near zero, revealing the expected preference for the inference result from the central-value histogram. The two-dimensional contours represent 68% confidence levels and are markedly larger than those seen in Fig. 4 (note the difference in axis scales). This reflects the dominance of the experimental uncertainties (from throws of the data covariance matrix) over the intrinsic uncertainties on our inference technique (quantified in the prior validation tests). Propagation of the experimental uncertainties into our results also introduces significant correlations between the inferred values of the parameters, as shown by the more complicated shapes of the two-dimensional contours in Fig. 9 compared to Fig. 4. For our analysis of the T2K data, we find that θ_1 has relatively weak correlations with the other parameters, while the remaining three are strongly (anti)correlated with each other. The parameter correlation matrix for the T2K dataset is shown in the right-hand plot of Fig. 9.

In summary, in Table I we report the parameter values for all of our tests and the corresponding χ^2 values obtained when comparing with the T2K measurement. The agreement between the original and NPE fits is excellent, both for the MicroBooNE Tune and NuWro. A traditional fit to the T2K measurement (based on direct minimization of χ^2) yields the best quantitative agreement with the data but shows symptoms of PPP (low normalization, unlikely parameter values). The NPE fit avoids these deficiencies at the price of a higher χ^2 score, which is nevertheless comparable to or slightly better than those achieved by the NuWro and MicroBooNE Tune models.

VI. SUMMARY AND CONCLUSION

Shortcomings in neutrino scattering models are commonly addressed by tuning simulation parameters to match interaction data, a process that is growing ever more demanding. For the first time, here we show that simulation-based inference with neural posterior estimation can meet this challenge by efficiently inferring neutrino interaction model parameters from data.

As a first application, we train a model that infers the four GENIE parameters that control the quasielastic and MEC processes in charged-current neutrino-nucleus interactions. We set up the model to replicate the procedure followed by MicroBooNE to extract those parameters from T2K data. After validating the model on different GENIE configurations, including the MicroBooNE Tune, we derived a GENIE configuration that approximates the predictions of the NuWro generator for the T2K dataset. This effectively defines a procedure for deriving surrogate models of other generators, potentially allowing experiments to save computing resources and avoid producing a full simulation sample for each generator of interest. This procedure can be particularly useful for comparing predictions of different generators to data before unfolding. Finally, we inferred the parameters on the same T2K measurement used for the MicroBooNE Tune. We find that the parameters extracted by our model are within the uncertainties of the MicroBooNE result and provide a slightly improved fit to the data when evaluated in terms of χ^2 .

The method we have developed in this article can be scaled up in future work. Classical χ^2 -based fits become increasingly costly as the number of free parameters grows, since the optimization must be repeated whenever the input data, the covariance, or the parameter set changes. SBI, by contrast, has a one-time training cost and amortized inference, which we expect to make it well suited to fits with substantially more parameters than the four considered here. Such an increase in complexity will likely be required to achieve a fit to the present T2K data set (or many other neutrino cross-section measurements) in which both a low χ^2 score and plausible posterior values of the model parameters are obtained. It is expected that NPE using discrete normalizing flows, like MAFs, should scale well up to around 30 dimensions [25,62]. For more complex problems, the simulation budget and flow expressiveness scales unfavorably with the number of parameter-data pairs needed to jointly model them with discrete flows. In case the model includes both physics parameters of interest and NUISANCE parameters, our future work could utilize factorized normalizing flows [63], which decompose the problem into nominal density and per NUISANCE parameter deformation instead of brute-forcing a single high-dimensional joint flow. Another approach we could explore for modeling data from upcoming high-statistics experiments is using flow-matching or continuous normalizing flow estimators, which have been shown to work well with much larger or even variable number of parameters [64].

Another area for continued exploration in which SBI may prove advantageous is in interpreting neutrino cross-section data releases with non-Gaussian measurement uncertainties. While reporting an approximate data covariance matrix is the *de facto* standard for the community at present, other representations are possible and may achieve better accuracy. A recent attempt by the MicroBooNE Collaboration to provide more detailed uncertainties may be found in Supplemental Material published with Ref. [65]. Although the “basic data release” therein includes a standard covariance matrix describing Gaussian uncertainties on the cross-section measurement, the “extended data release” tabulates the full set of variations to the simulated MicroBooNE event rates used to estimate the systematic uncertainties. By applying SBI to an ensemble of cross-section results reextracted from the uncorrected data using each of the simulated predictions, MicroBooNE’s original non-Gaussian systematic uncertainty treatment may be directly propagated into corresponding uncertainties on the inferred parameter values.

Future studies may also focus on other topics related to uncertainty quantification, improving the model coverage, and evaluating strategies for including experimental uncertainties as part of the model inputs. Overall, we expect that our approach will be generalizable to similar problems in neutrino interaction modeling, potentially leading to more powerful parameter fits that are currently limited by personnel effort and computing resources.

ACKNOWLEDGMENTS

This work was produced by Fermi Forward Discovery Group, LLC under Contract No. 89243024CSC000002 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. The work of K. T. is supported by DOE Grants No. DE-SCL0000156 and

FNAL 23-32. Tame-Narvaez: Conceptualization, methodology, investigation, formal analysis, software, validation, data curation, resources, visualization, writing—original draft, writing—review and editing. Gardiner: Conceptualization, methodology, investigation, formal analysis, software, validation, data curation, resources, visualization, writing—original draft, writing—review and editing, supervision. Ćiprijanović: Conceptualization, methodology, writing—original draft, writing—review and editing, supervision. Cerati: Conceptualization, methodology, writing—original draft, writing—review and editing, supervision. The U.S. Government retains, and the publisher, by accepting the work for publication, acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan [66].

DATA AVAILABILITY

The data that support the findings of this article are openly available at [67,68].

APPENDIX: ADDITIONAL BINS FROM THE T2K DATA SET

Figure 10 presents the remaining bins of the two-dimensional “Analysis I” T2K cross-section measurement [35] that are included in our analysis but not shown in Fig. 6. Each panel shows a single bin of scattering cosine $\cos \theta$, which is subdivided into bins of muon momentum p_μ . The plotting format is identical to Fig. 6. In these portions of the dataset, the inferred predictions follow trends similar to those already discussed in the main text.

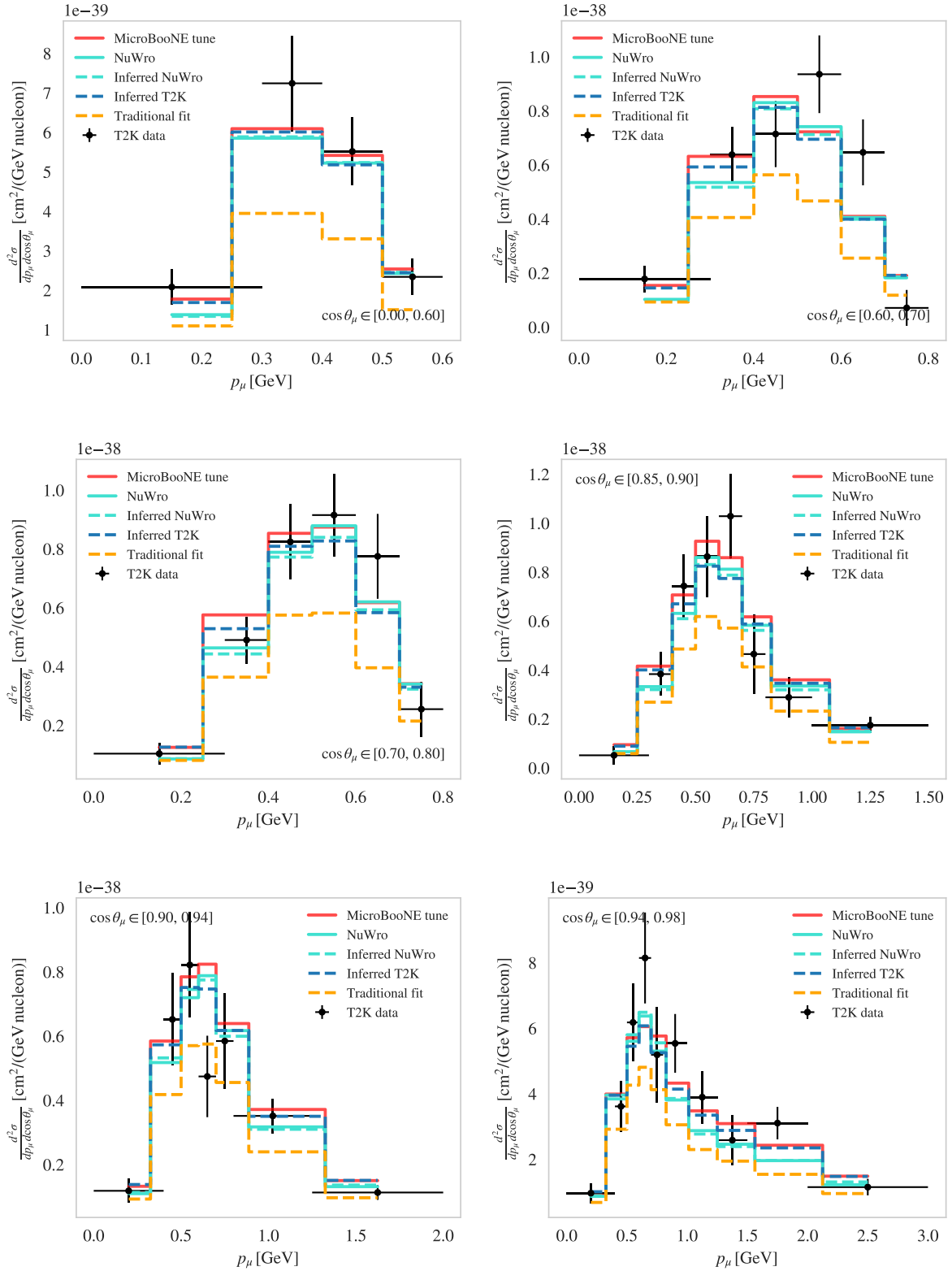


FIG. 10. Comparison of the T2K cross-section data (black points) to the NuWro [57] (solid cyan), NPE-inferred NuWro (dashed cyan), MicroBooNE Tune [9] (solid red), NPE-inferred T2K (dashed blue), and traditional χ^2 fit (dashed orange) model predictions in different slices of $\cos\theta$.

- [1] K. Abe *et al.* (Hyper-Kamiokande Collaboration), [arXiv:1805.04163](#).
- [2] R. Acciarri *et al.* (DUNE Collaboration), [arXiv:1512.06148](#).
- [3] B. Abi *et al.* (DUNE Collaboration), [arXiv:2002.03005](#).
- [4] L. Alvarez-Ruso *et al.* (NuSTEC Collaboration), *Prog. Part. Nucl. Phys.* **100**, 1 (2018).
- [5] C. Andreopoulos *et al.*, *Nucl. Instrum. Methods Phys. Res., Sect. A* **614**, 87 (2010).
- [6] L. Alvarez-Ruso *et al.* (GENIE Collaboration), *Eur. Phys. J. Special Topics* **230**, 4449 (2021).
- [7] P. Stowell *et al.* (MINERvA Collaboration), *Phys. Rev. D* **100**, 072005 (2019).
- [8] M. A. Acero *et al.* (NOvA, R. Group Collaborations), *Eur. Phys. J. C* **80**, 1119 (2020).
- [9] P. Abratenko *et al.* (MicroBooNE Collaboration), *Phys. Rev. D* **105**, 072001 (2022).
- [10] J. Tena-Vidal *et al.* (GENIE Collaboration), *Phys. Rev. D* **104**, 072009 (2021).
- [11] J. Tena-Vidal *et al.* (GENIE Collaboration), *Phys. Rev. D* **105**, 012009 (2022).
- [12] J. Tena-Vidal *et al.* (GENIE Collaboration), *Phys. Rev. D* **106**, 112001 (2022).
- [13] W. Li *et al.* (GENIE Collaboration), *Phys. Rev. D* **110**, 072016 (2024).
- [14] N. M. Coyle, S. W. Li, and P. A. N. Machado, *Phys. Rev. D* **111**, 093010 (2025).
- [15] N. M. Coyle, S. W. Li, and P. A. N. Machado, *J. High Energy Phys.* **12** (2022) 166.
- [16] K. Cranmer, J. Brehmer, and G. Louppe, *Proc. Natl. Acad. Sci. U.S.A.* **117**, 30055 (2020).
- [17] M. Deistler *et al.*, [arXiv:2508.12939](#).
- [18] D. B. Rubin, *Ann. Stat.* **12**, 1151 (1984), <https://www.jstor.org/stable/2240995>.
- [19] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, *Phys. Rev. D* **98**, 052004 (2018).
- [20] J. Brehmer, S. Mishra-Sharma, J. Hermans, G. Louppe, and K. Cranmer, *Astrophys. J.* **886**, 49 (2019).
- [21] A. Coogan, N. Anau Montel, K. Karchev, M. W. Grootes, F. Nattino, and C. Weniger, *Mon. Not. R. Astron. Soc.* **527**, 66 (2024).
- [22] N. Anau Montel, A. Coogan, C. Correa, K. Karchev, and C. Weniger, *Mon. Not. R. Astron. Soc.* **518**, 2746 (2023).
- [23] R. Legin, Y. Hezaveh, L. Perreault Levasseur, and B. Wandelt, [arXiv:2112.05278](#).
- [24] S. Wagner-Carena, J. Aalbers, S. Birrer, E. O. Nadler, E. Darragh-Ford, P. J. Marshall, and R. H. Wechsler, *Astrophys. J.* **942**, 75 (2023).
- [25] J. Poh, A. Samudre, A. Ćiprijanović, J. Frieman, G. Khullar, and B. D. Nord, *J. Cosmol. Astropart. Phys.* **05** (2025) 053.
- [26] G. Khullar, B. Nord, A. Ćiprijanović, J. Poh, and F. Xu, *Mach. Learn.* **3**, 04LT04 (2022).
- [27] M. Reza, Y. Zhang, B. Nord, J. Poh, A. Ciprijanovic, and L. Strigari, in *Machine Learning for Astrophysics* (2022), p. 20, [arXiv:2208.00134](#).
- [28] F. Gerardi, S. M. Feeney, and J. Alsing, *Phys. Rev. D* **104**, 083531 (2021).
- [29] J. Brehmer, *Nat. Rev. Phys.* **3**, 305 (2021).
- [30] R. Mastandrea, B. Nachman, and T. Plehn, *Phys. Rev. D* **110**, 056004 (2024).
- [31] R. Barrué, P. C. Muñio, V. Dao, and R. Santos, *J. High Energy Phys.* **04** (2024) 014.
- [32] S. Bieringer, A. Butter, T. Heimel, S. Höche, U. Köthe, T. Plehn, and S. T. Radev, *SciPost Phys.* **10**, 126 (2021).
- [33] A. Gavrikov *et al.*, *Commun. Phys.* **9**, 63 (2026).
- [34] P. Stowell *et al.*, *J. Instrum.* **12**, P01016 (2017).
- [35] K. Abe *et al.* (T2K Collaboration), *Phys. Rev. D* **93**, 112012 (2016).
- [36] P. Abratenko *et al.* (MicroBooNE Collaboration), *Nature (London)* **648**, 64 (2025).
- [37] P. Abratenko *et al.* (MicroBooNE Collaboration), [arXiv:2511.17342](#).
- [38] J. Wolfs and C. M. Marshall, *Phys. Rev. D* **112**, 092018 (2025).
- [39] K. Tame-Narvaez, A. Ćiprijanović, S. Gardiner, and G. Cerati, in *39th Annual Conference on Neural Information Processing Systems: Includes Machine Learning and the Physical Sciences (MLAPS)* (2025), [arXiv:2510.07454](#).
- [40] S. Abe *et al.*, *Phys. Rev. D* **113**, 012011 (2026).
- [41] R. Frühwirth, D. Neudecker, and H. Leeb, *EPJ Web Conf.* **27**, 00008 (2012).
- [42] T. Golan, J. Sobczyk, and J. Żmuda, *Nucl. Phys. B, Proc. Suppl.* **229–232**, 499 (2012).
- [43] C. Juszczak, J. T. Sobczyk, and J. Żmuda, *Phys. Rev. C* **82**, 045502 (2010).
- [44] G. Papamakarios, D. C. Sterratt, and I. Murray, in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2018), <https://proceedings.mlr.press/v89/papamakarios19a>.
- [45] J. Hermans, V. Begy, and G. Louppe, in *International Conference on Machine Learning* (PMLR, 2019), <https://proceedings.mlr.press/v119/hermans20a>.
- [46] G. Papamakarios and I. Murray, [arXiv:1605.06376](#).
- [47] D. S. Greenberg, M. Nonnenmacher, and J. H. Macke, in *International Conference on Machine Learning* (PMLR, 2019), <https://proceedings.mlr.press/v97/greenberg19a.html>.
- [48] J.-M. Lueckmann, G. Bassetto, T. Karaletsos, and J. H. Macke, [arXiv:1805.09294](#).
- [49] J. Hermans, A. Delaunoy, F. Rozet, A. Wehenkel, V. Begy, and G. Louppe, [arXiv:2110.06581](#).
- [50] G. Papamakarios, T. Pavlakou, and I. Murray, [arXiv:1705.07057](#).
- [51] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle, *J. Mach. Learn. Res.* **17**, 205:1 (2016).
- [52] D. J. Rezende and S. Mohamed, [arXiv:1505.05770](#).
- [53] M. Germain, K. Gregor, I. Murray, and H. Larochelle, in *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 37, edited by F. Bach and D. Blei (PMLR, Lille, France, 2015), pp. 881–889.
- [54] S. Wagner-Carena, J. W. Park, S. Birrer, P. J. Marshall, A. Roodman, and R. Wechsler, *Astrophys. J.* **909** (2020).
- [55] P. Lemos, A. Coogan, Y. Hezaveh, and L. Perreault-Levasseur, [arXiv:2302.03026](#).
- [56] P. Lemos, A. Coogan, Y. Hezaveh, and L. Perreault-Levasseur, in *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 202, edited by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J.

- Scarlett (PMLR, 2023) pp. 19256–19273, <https://proceedings.mlr.press/v202/lemos23a.html>.
- [57] C. Juszczak, Acta Phys. Pol. B **40**, 2507 (2009), <https://ui.adsabs.harvard.edu/abs/2009AcPPB..40.2507J/abstract>.
- [58] Z. Lin *et al.* (MINERvA Collaboration), Phys. Rev. D **114**, 012015 (2026).
- [59] J. Chakrani *et al.*, Phys. Rev. D **109**, 072006 (2024).
- [60] L. Koch, Phys. Rev. D **103**, 113008 (2021).
- [61] V. G. Pronyaev, S. A. Badikov, C. Zhenpeng, A. D. Carlson, E. V. Gai, G. M. Hale, F. Hambsch, H. M. Hofmann, N. M. Larson, D. L. Smith, S. Oh, S. Tagesen, and H. Vonach, AIP Conf. Proc. **769**, 808 (2005).
- [62] P. J. Gonçalves, J.-M. Lueckmann, M. Deistler, M. Nonnenmacher, K. Öcal, G. Bassetto, C. Chintaluri, W. F. Podlaski, S. A. Haddad, T. P. Vogels, D. S. Greenberg, and J. H. Macke, eLife **9**, e56261 (2020).
- [63] D. Valsecchi, M. Donegà, and R. Wallny, arXiv:2602.13184.
- [64] M. Gloeckler, M. Deistler, C. Weilbach, F. Wood, and J. H. Macke, arXiv:2404.09636.
- [65] P. Abratenko *et al.* (MicroBooNE Collaboration), Phys. Rev. D **112**, 112004 (2025).
- [66] <http://energy.gov/downloads/doe-public-access-plan>.
- [67] <https://zenodo.org/uploads/18747090>.
- [68] https://github.com/karlaTame/SBI_parameter_tuning.