Efficient neutrino oscillation parameter inference using Gaussian processes

Lingge Li, Nitish Nayak, Jianming Bian, and Pierre Baldi
University of California, Irvine, California, USA

(Received 11 June 2019; published 2 January 2020)

The unified approach of Feldman and Cousins allows for exact statistical inference of small signals that commonly arise in high energy physics. It has gained widespread use, for instance, in measurements of neutrino oscillation parameters in long-baseline experiments. However, the approach relies on the Neyman construction of the classical confidence interval and is computationally intensive as it is typically done in a grid-based fashion over the entire parameter space. In this article, we propose an efficient algorithm for the Feldman-Cousins approach using Gaussian processes to construct confidence intervals iteratively. We show that in the neutrino oscillation context, one can obtain confidence intervals fives times faster in one dimension and ten times faster in two dimensions, while maintaining an accuracy above 99.5%.

DOI: 10.1103/PhysRevD.101.012001

I. INTRODUCTION

Constructing classical confidence intervals for physical parameters with boundary conditions is challenging when dealing with small signals. The challenge is especially evident when studying neutrino oscillations because of the low event counts and multiple competing effects on the energy spectrum. The low event counts are primarily caused by the extremely low interaction cross section of neutrinos, arising from the fact that they interact via the weak nuclear force. In order to extract meaningful statistical conclusions, one has to resort to means other than the asymptotic properties of Poisson data. The gold standard is the so-called unified approach outlined by Feldman and Cousins [1]. It builds upon the Neyman construction of classical confidence intervals by specifying an ordering principle based on likelihood ratios and is known for providing correct coverage.

The Feldman-Cousins approach is firmly grounded in statistical theory and widely used in neutrino experiments, e.g., Refs. [2–4]. However, it comes at a heavy computational cost, which in some cases such as Ref. [2] renders it infeasible for multidimensional confidence intervals. For the $1-\alpha$ confidence interval, the Feldman-Cousins approach includes all the values in the parameter space where the likelihood ratio test fails to reject at α level. However, it does not provide a prescription for how to sample that parameter space. Therefore, one is forced to sample it in its entirety in a grid-based fashion. Moreover, at each point one has to perform a large number of Monte Carlo simulations in order to calculate the p-value for the likelihood ratio test.

To accelerate the Feldman-Cousins approach, we propose approximating the function of p-values over the parameter space with Gaussian processes. Instead of performing a large number of Monte Carlo simulations, we start with just a small number of them at several parameter values to get noisy estimates of the p-values. We then train a Gaussian process model to interpolate over these estimates. Iteratively, we perform more Monte Carlo simulations to refine the Gaussian process approximation. We can control the p-value approximation error so that it does not change the likelihood ratio test decisions and the confidence interval. Meanwhile, the Monte Carlo simulations can be allocated intelligently in the parameter space to achieve substantial savings in computation.

The proposed algorithm is rooted in the framework of Bayesian optimization [5]. It was originally designed to find the extremal points of an objective function that is unknown *a priori*. In the Feldman-Cousins approach, the function of p-values over the parameter space is unknown. We adapt Bayesian optimization to locate a set of points in the parameter space that lie on the boundary of desired confidence intervals. By side-stepping points that are estimated to be either inside or outside the confidence interval with high probability, we can thus reduce the computational cost while producing the same result. We show that in the context of neutrino oscillation experiments, one can accelerate the construction of one-dimensional and two-dimensional confidence intervals by a factor of 5 and 10, respectively, without sacrificing the accuracy of the Feldman-Cousins approach.

II. STATISTICAL INFERENCE FOR NEUTRINO OSCILLATIONS

A. Neutrino oscillations

Neutrino oscillations demonstrate that neutrinos have mass and that the neutrino mass eigenstates are different from their flavor eigenstates. In the three flavor framework,

© 2020 American Physical Society

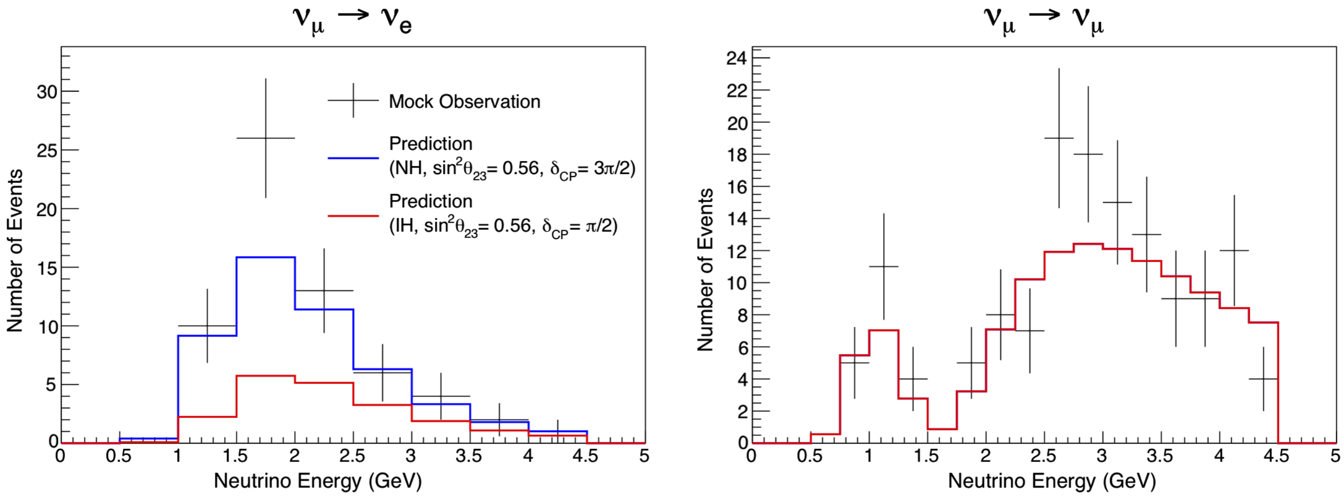

FIG. 1. An illustration of a toy neutrino oscillation experiment setup with the $\nu_\mu \to \nu_e$ channel on the left and the $\nu_\mu \to \nu_\mu$ on the right. Expectations for different oscillation parameters are compared to mock observations in order to find maximum likelihood estimates. The likelihood of observed data is maximized using the extended likelihood function. The fit is performed in both channels simultaneously.

the transformation of the mass eigenstates (ν_1, ν_2, ν_3) into the flavor eigenstates (ν_e, ν_μ, ν_τ) is described by the 3×3 unitary matrix U_{PMNS} [6], which is parametrized by three mixing angles θ_{12}, θ_{23} and θ_{13}, and a CP violation phase δ_{CP}. The probability of oscillations between different neutrino flavor states of given energy E_ν over a propagation distance (baseline) L depends on the U_{PMNS} parameters and the difference of the squared masses of the eigenstates, Δm^2_{32} and Δm^2_{21}.

The mixing angles θ_{12} and θ_{13} along with the squared-mass splitting Δm^2_{12} have been measured to relatively high accuracy by several experiments, e.g., Refs. [7–9]. One can then infer the remaining parameters, θ_{23}, δ_{CP}, and Δm^2_{32}, by measuring the probabilities $P(\nu_\mu \to \nu_\mu)$ and $P(\nu_\mu \to \nu_e)$. Of particular interest are (1) the sign of Δm^2_{32}, positive indicating a "normal hierarchy" (NH) and negative indicating an "inverted hierarchy" (IH) of neutrino mass states, (2) whether $\delta_{CP} \neq 0, \pi$, indicating charge-parity (CP) violation in the lepton sector, and (3) whether the mixing angle is in fact maximal, i.e., $\theta_{23} = 45°$. The neutrino mass hierarchy has important implications for current and future neutrino experiments [10] involved in measuring the absolute neutrino mass and investigating the possible Majorana nature of the neutrino. Leptonic CP violation could be important to deduce the origin of the predominance of matter in the Universe.

To infer neutrino oscillation parameters θ, a typical long-baseline neutrino oscillation experiment sends a beam of ν_μ neutrinos into a detector and observes a handful of oscillated ν_e neutrinos along with ν_μ neutrinos that survive over the baseline. As the oscillation probability is a function of neutrino energy, the observed neutrinos are binned by their energy. The neutrino oscillation parameters are inferred by comparing the observed neutrino energy spectra with the expected spectra for different oscillation parameters as shown in Fig. 1.

B. Feldman-Cousins approach

Denote the random variable for the neutrino count in the ith energy bin by X_i. Further, assume that each X_i follows an independent Poisson distribution with mean λ_i. For a given θ, the expectations $\vec{\lambda}$ are also influenced by systematic uncertainties in the beam configuration and the interaction model among others, which we parametrize by δ. For given oscillation and nuisance parameters (θ, δ), the expectations $\vec{\lambda}$ given (θ, δ) are obtained through simulations as they are analytically intractable. Denote the implicit mapping between $\vec{\lambda}$ and (θ, δ) by v. The extended log likelihood of (θ, δ) is given by

$$\log L(\theta,\delta) = \sum_{i\in I} \log Pois(x_i; v(\theta,\delta)_i) + \log Pois\left(\sum_{i\in I} x_i; \sum_{i\in I} v(\theta,\delta)_i\right) - \frac{1}{2}\delta^2,$$

where $-\frac{1}{2}\delta^2$ is a penalty term for systematic error [11].

For a unified treatment of constructing classical confidence intervals for both null and non-null observations, an ordering principle based on likelihood ratios was introduced by Feldman and Cousins in 1997. The unified approach provides correct coverage even at parameter boundaries and has the highest statistical power as a result of the Neyman-Pearson lemma. In essence, a particular parameter value θ_0 is included in the $1 - \alpha$ confidence interval if the likelihood ratio test fails to reject the null hypothesis $\theta = \theta_0$ at the α level. The likelihood ratio test statistic is given by

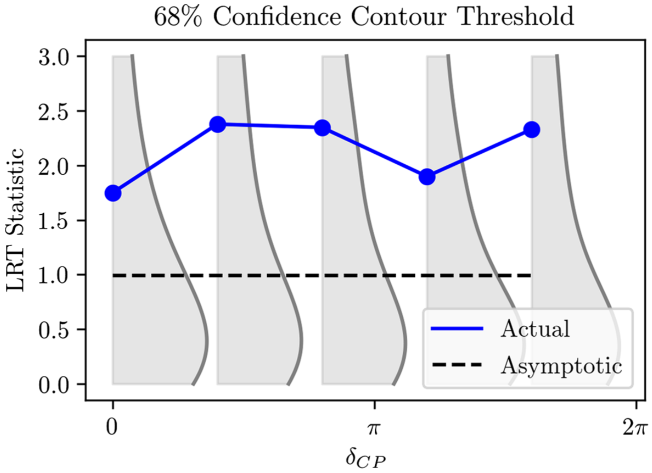

FIG. 2. In the context of neutrino oscillations, the likelihood ratio test statistic distribution changes in the parameter space. Here the parameter is δ_{CP} and ranges from 0 to 2π. The solid blue line indicates the 68th percentile of Monte Carlo simulated distributions while the dashed black line is the 68th percentile of the asymptotic χ^2_1 distribution.

$$-2\log\frac{L(\theta_0)}{\mathrm{argmax}_\theta L(\theta)}$$

and has an asymptotic χ^2 distribution by Wilks' theorem.

In the context of neutrino oscillations, the asymptotic distribution is unreliable because of the small sample size in neutrino data and physical boundaries on the oscillation parameters. The reference distribution of the likelihood ratio test statistic can vary drastically as a function of θ; Fig. 2 shows several distributions at different θ values and comparisons of their critical values, in particular. Therefore, for any given θ, Monte Carlo experiments are used to simulate the reference distribution and calculate the p-value for the likelihood ratio test. Since the parameter space is bounded, the simulations are performed on a grid for a large number of θ values and the computational cost adds up quickly.

III. GAUSSIAN PROCESS ALGORITHM

A. Gaussian process regression

A Gaussian process ($\mathcal{GP}$) is a stochastic process where any finite collection of points are jointly Gaussian with mean μ and covariance Σ. An interpretation of the $\mathcal{GP}$ is an infinite extension of multivariate Gaussian; a $\mathcal{GP}$ can be thought of as a distribution in the function space where each draw from the distribution is a curve. Typically, the zero mean $\mathcal{GP}$ is used for modeling but it is still impossible to specify an infinite-dimensional covariance matrix explicitly. Instead, we can parametrize a zero mean $\mathcal{GP}$ with a kernel function κ that defines the pairwise covariance. Let $f\sim\mathcal{GP}(0,\kappa(\cdot,\cdot))$. Then for any pair x and x' we have

$$\begin{pmatrix} f(x)\\ f(x')\end{pmatrix}\sim\mathcal{N}\left(0,\begin{bmatrix}\kappa(x,x) & \kappa(x,x')\\ \kappa(x,x') & \kappa(x',x')\end{bmatrix}\right).$$

Given a finite set of observed data $\vec{x}_{\mathrm{obs}}$, we can write down the multivariate Gaussian likelihood in this fashion and maximize it through kernel parameters ω. Conveniently, at a new point x^* we can obtain the closed form predictive distribution,

$$f(x^*)|f(\vec{x}_{\mathrm{obs}})\sim N(\kappa(x^*,\vec{x}_{\mathrm{obs}})(\kappa(\vec{x}_{\mathrm{obs}},\vec{x}_{\mathrm{obs}}))^{-1}f(\vec{x}_{\mathrm{obs}}),\\ \kappa(x^*,x^*)-\kappa(x^*,\vec{x}_{\mathrm{obs}})(\kappa(\vec{x}_{\mathrm{obs}},\vec{x}_{\mathrm{obs}}))^{-1}\kappa(\vec{x}_{\mathrm{obs}},x^*)).$$

Since a $\mathcal{GP}$ is uniquely characterized by the kernel, different kernels produce distinct behaviors. A commonly used kernel is squared exponential $\kappa(x_1,x_2)=\exp(-(x_1-x_2)^2/l^2)$, where l is called the length scale. Intuitively, the length scale determines the distance over which the $\mathcal{GP}$ interpolates between points. The squared exponential kernel is infinitely differentiable and functions drawn from such a $\mathcal{GP}$ would be smooth. However, this smoothness assumption might not be appropriate for some applications; a more general kernel is the Matérn kernel. The Matérn kernel has an additional parameter ν that controls the smoothness and the squared exponential kernel is a special case where $\nu\to\infty$. Figure 3 shows some $\mathcal{GP}$ examples and please refer to Ref. [12] for more details on Gaussian processes.

Different kernels can be combined to compose a $\mathcal{GP}$ as long as the new kernel covariance matrix is still positive semidefinite. With the squared exponential kernel alone, the covariance implies that the observed data have no error. To account for error in the data, a diagonal matrix $\sigma^2 I$ is usually added to model constant variance across observations. In many situations such as ours, there exists heteroskedasticity, which means that different observations have different errors. When we iteratively perform Monte Carlo simulations to calculate p-values, the errors in the estimates also vary based on the number of simulations. We can actually model the p-value error as a diagonal matrix and add it to the $\mathcal{GP}$ covariance.

B. Monte Carlo error estimation

In the Feldman-Cousins approach, a large number of Monte Carlo simulations are required in order to make the error in p-value calculation negligible. When the Monte Carlo error in p-value calculation is not negligible, we should try to quantify it. Since the p value is the quantile of the observed likelihood ratio statistic under the reference distribution, we can use a binomial proportion confidence interval as the p-value error estimate as outlined below [13]. As shown in Fig. 4, the Monte Carlo error only slowly approaches 0 when the number of simulations increases to 10,000.

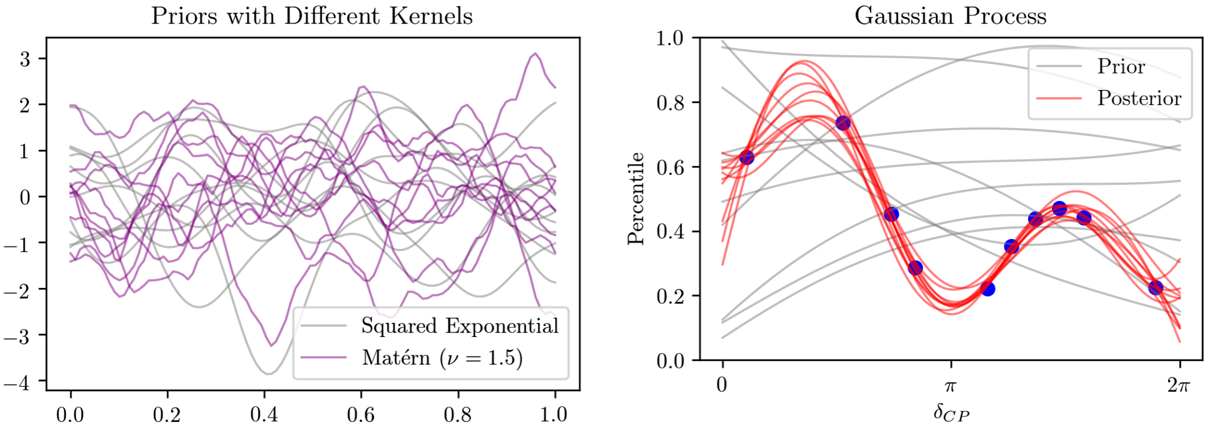

FIG. 3. (Left) Sampled paths from Gaussian processes with different kernels. With $\nu = 1.5$, the Matérn kernel produces functions that are only once differentiable. (Right) Sampled paths from Gaussian process prior and posterior with squared exponential kernel. The posterior paths, representing the curves drawn from the predictive distribution, are better aligned with the observed data points in solid blue.

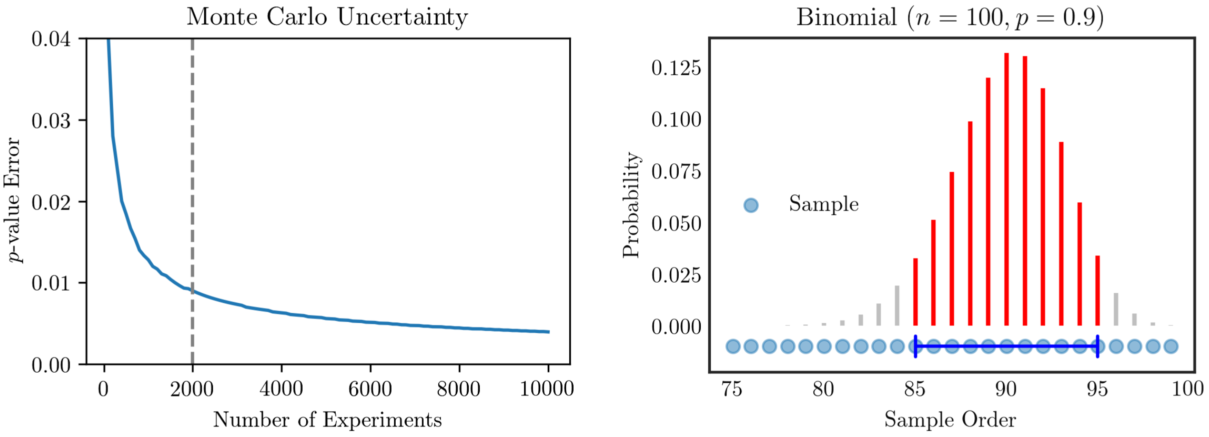

FIG. 4. (Left) Monte Carlo error in terms of p-value as a function of the number of experiments. (Right) Example of nonparametric quantile interval construction using binomial distribution. In a sample with 100 draws, the 85th and 95th order statistics form a 95% confidence interval for the 90th quantile of the unknown distribution.

Suppose $X_1, \ldots, X_n$ are independent draws from an unknown distribution F whose qth quantile is denoted by $F^{-1}(q)$. Each draw X_i is either below or above $F^{-1}(q)$ with probability q. Consequently, M, the number of X_i's less than or equal to $F^{-1}(q)$, has a binomial (n, q) distribution. We can obtain a confidence interval for $F^{-1}(q)$ with sample statistics $X_{(l)}, X_{(u)}$ (the lth and uth ordered draws) with $1 \leq l \leq u \leq n$ such that

$$B(u-1; n, q) - B(l-1; n, q) \geq 1 - \alpha.$$

$B(u-1; n, q) - B(l-1; n, q)$ is the probability that M is between l and $u-1$. Thus, (l, u) would form a confidence interval for M. Correspondingly, $(X_{(l)}, X_{(u)})$ would form a confidence interval for $F^{-1}(q)$. Our goal, however, is to estimate $F(x^*)$ for an arbitrary x^* given sample X, where, in our context, x^* is the observed likelihood ratio test statistic and $F(x^*)$ is the p value. This can be done by inverting the quantile confidence interval until the confidence intervals for $F^{-1}(q_l)$ and $F^{-1}(q_u)$ no longer contain x^*. Then (q_l, q_u) would form a confidence interval for $F(x^*)$.

C. Proposed algorithm

Bayesian optimization can be used to find the extremum of a black-box function h when h is expensive to evaluate such that a grid search is too computationally intensive. Bayesian optimization is an iterative procedure; in each iteration, h is evaluated at a number of points to update an approximation of h. The approximation usually starts from a zero-mean Gaussian process prior $\mathcal{GP}(0, \kappa(\cdot,\cdot))$. After each iteration, the $\mathcal{GP}$ model yields a posterior distribution, hence Bayesian. Based on the approximation posterior, the points in the next iteration are proposed by an acquisition function a. The acquisition function a aims to balance

Algorithm 1. $\mathcal{GP}$ iterative confidence interval construction.

```
for each iteration $t = 1, 2, \ldots$ do
    Propose points in parameter space $\mathrm{argmax}_\theta a(\theta)$
    for each point $\theta'$ do
        Simulate likelihood ratio statistic distribution
        for $k = 1, 2, \ldots$ do
            Perform a pseudoexperiment
            Maximize the likelihood with respect to $(\theta, \delta)$
            Maximize the likelihood with constraint $\theta = \theta'$
            Calculate likelihood ratio statistic
        end for
        Calculate $p$ value based on the simulated distribution
    end for
    Train $\mathcal{GP}$ approximation $f(\theta)$ for the $p$ values
    Update confidence intervals
end for
```

between "exploration," reducing approximation uncertainty, and "exploitation," reaching the extremum.

In our context, the expensive black-box function is the function of p values over the parameter space. Denote the grid points in the parameter space, where Monte Carlo simulations are performed, by $\vec{\theta}_o$, the simulated p values at these points by $y(\vec{\theta}_o)$, and the independent simulation errors by $\sigma(\vec{\theta}_o)$. The $\mathcal{GP}$ predictive posterior distribution of the unobserved p values at $\vec{\theta}_u$ conditional on obtained p values $y(\vec{\theta}_o)$ at points $\vec{\theta}_o$ is then given by

$$f(\vec{\theta}_u)|y(\vec{\theta}_o) \sim \mathcal{N}(K_{uo}(K_{oo} + \mathrm{diag}(\sigma^2(\vec{\theta}_o)))^{-1}y(\vec{\theta}_o),\\ K_{uu} - K_{uo}(K_{oo} + \sigma^2(\vec{\theta}_o)I)^{-1}K_{ou})$$

where $K_{oo}, K_{ou}, K_{uo}, K_{uu}$ denote the covariance matrices between points $\vec{\theta}_o$ and $\vec{\theta}_u$.

Different from typical Bayesian optimization, we do not simply wish to find the minimum or maximum p value. Instead, we want to find the points where the p value is equal to α so that they enclose the confidence interval. Moreover, we want to be able to find multiple intervals at different confidence levels. Therefore, we choose our acquisition function to be

$$a(\theta) = \sum_{\alpha_i} \left| \frac{f(\theta) - \alpha_i}{\sigma_{f(\theta)}} \right|^{-1},$$

where $f(\theta)$ is the $\mathcal{GP}$ approximated p value (posterior mean) at θ and $\sigma_{f(\theta)}$ is the $\mathcal{GP}$ posterior standard deviation at θ.

Iteratively, the $\mathcal{GP}$ algorithm seeks points on the boundary of confidence intervals, for which it is unsure about. Points far from the boundary, which have p values much greater or less than α_i, are probabilistically "ruled out." At these points, we end up performing fewer Monte Carlo experiments or skipping them altogether. Every point on the grid would be either included or rejected with some uncertainty based on the $\mathcal{GP}$ posterior. With more iterations, the uncertainty diminishes so that the approximated confidence intervals converge to the ones produced by a full grid search. Figure 5 illustrates the proposed algorithm on a one-dimensional example.

Here we use the squared exponential kernel with the Monte Carlo errors added to the covariance diagonal and a small amount of white noise as often done in a regression setting [12]. Point estimates of the $\mathcal{GP}$ kernel parameters by optimizing the log marginal likelihood

$$-\frac{1}{2}y(\vec{\theta}_o)^T(K_{oo} + \mathrm{diag}(\sigma^2(\vec{\theta}_o)))^{-1}y(\vec{\theta}_o)\\ -\frac{1}{2}\log|K_{oo} + \mathrm{diag}(\sigma^2(\vec{\theta}_o))| - \frac{n}{2}\log 2\pi.$$

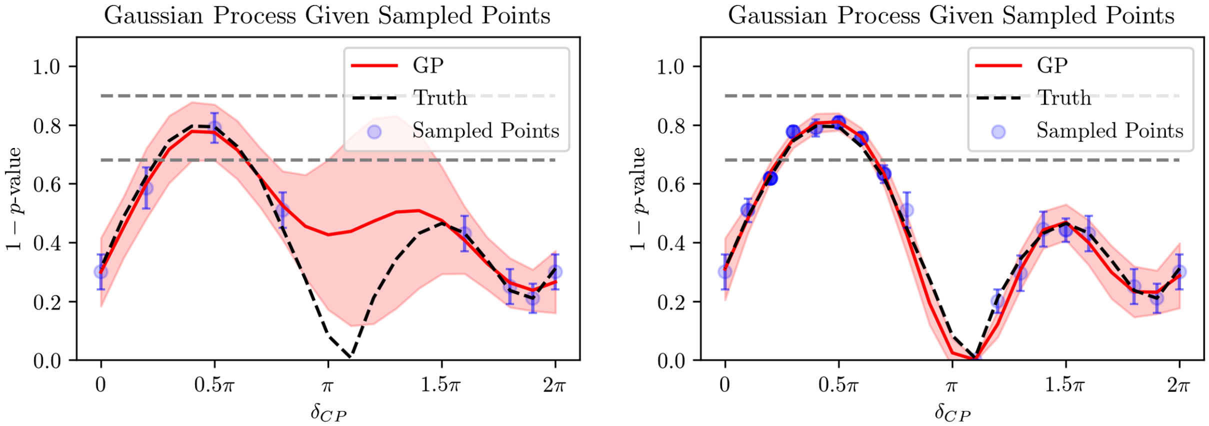

FIG. 5. An illustration of our construction for the 68% and 90% confidence intervals for δ_{CP}, which consist of points lying underneath the dashed horizontal lines. From a few initial points with high variance, the $\mathcal{GP}$ learns a rough approximation of the true curve (left). Based on the approximation, more points are proposed around the interval boundary, shown in dark blue, and the $\mathcal{GP}$ improves itself (right). The shade of blue represents the number of simulations used to calculate the p value and the error bars are for the p value.

There are constraints on the kernel parameters that should be incorporated. For instance, the length scale l should be greater than the grid resolution and less than the grid range.

IV. NUMERICAL STUDIES

By way of illustration, we set up a toy long-baseline neutrino oscillation experiment in order to construct confidence intervals for the oscillation parameters. A flux distribution of ν_μs is modeled as a Landau function over neutrino energies, $E_\nu \in (0.5, 4.5)$ GeV, with the location parameter at 2 GeV as shown in Fig 6. The normalization uncertainty is taken to be 10% and is applied as a nuisance parameter. The ν_μ distribution is then oscillated into ν_es using the PMNS model for a toy baseline of 810 km through the Earth. Corrections from matter interactions [14] are applied assuming a constant matter density of $2.84\ \mathrm{g/cm^3}$. The setup is similar to NOvA [3], an accelerator-based long-baseline experiment at Fermilab. The oscillated ν_es are then "observed" with a toy interaction cross-section distribution, similar in shape to Ref. [15]; the cross section increases as a function of neutrino energy from 0 GeV up to 1 GeV and decreases slowly until a maximum neutrino energy of 4.5 GeV as shown in Fig 6. A 10% normalization uncertainty is applied on the cross section as another nuisance parameter. Finally, we scale up the ν_e distribution to get an energy spectrum expectation, in energy bins of 0.5 GeV between the flux range, similar to observations from NOvA [3]. The expected spectrum is computed from scratch for each set of oscillation and nuisance parameters in the toy experiment as shown in Fig. 1. A similar setup is used for the $\nu_\mu \to \nu_\mu$ channel. However, in order to expedite the computation, the two-flavor oscillation probability approximation is used. The reactor mixing angle θ_{13} and the solar parameters θ_{12} and Δm^2_{12} are fixed at the values given in Ref. [16]. A mock data set is obtained by applying Poisson variations on the expected spectrum at oscillation parameter values given by NOvA.

We then use this setup to construct one-dimensional confidence intervals for δ_{CP} and two-dimensional confidence intervals for $\sin^2\theta_{23}$ vs δ_{CP} by the two algorithms, a standard grid-search implementation of Feldman-Cousins and the $\mathcal{GP}$-based algorithm. $|\Delta m^2_{32}|$ is treated as a nuisance parameter while $\sin^2\theta_{23}$ is treated as another in the case of the one-dimensional interval for δ_{CP}. The likelihood function is integrated over the nuisance parameters assuming a flat prior in the range $(2,3)\times 10^{-3}\ \mathrm{eV}^2$ for $|\Delta m^2_{32}|$ and (0.3,0.7) for $\sin^2\theta_{23}$, similar to Ref. [2]. The prior on the nuisance parameters for the systematic uncertainties in the toy model is assumed to be a standard normal distribution. The toy model and parameter fitting routine are implemented in ROOT [17] while the Gaussian process algorithm is implemented with scikit learn [18].

A. One-dimensional confidence intervals

To make an inference on δ_{CP}, a significance curve is usually drawn under different assumptions of mass hierarchy as shown in Fig. 7. The portion of the significance curve below a certain value gives us the confidence interval at that level. We can observe that the NH curves by both the standard Feldman-Cousins (FC) and $\mathcal{GP}$ algorithms have the same intersections with the 1σ horizontal line, which implies that the 1σ confidence intervals are the same. Though there are slight discrepancies, the shape of the $\mathcal{GP}$ significance curve is mostly correct.

To evaluate the performance of the $\mathcal{GP}$ algorithm, we perform the same inference procedure on 200 different data sets to find the 68% and 90% confidence intervals. First, with standard FC results as ground truth, we consider the accuracy of the $\mathcal{GP}$ algorithm for classifying whether or not each grid point is included in the confidence intervals.

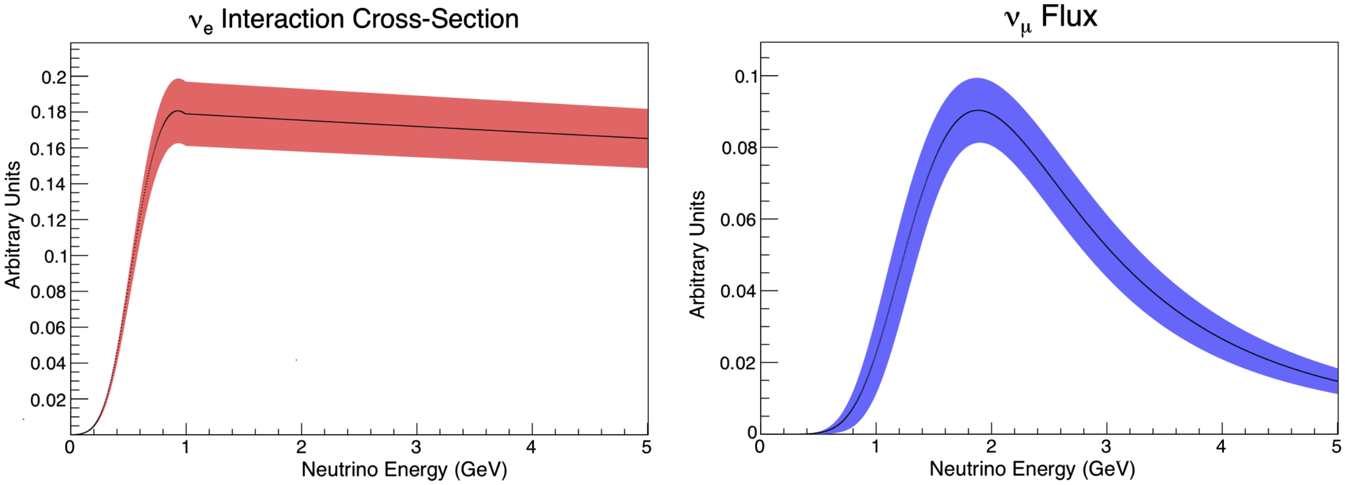

FIG. 6. The distributions for the ν_e interaction cross section (left) and ν_μ flux (right) are shown along with a normalization systematic error of 10%.

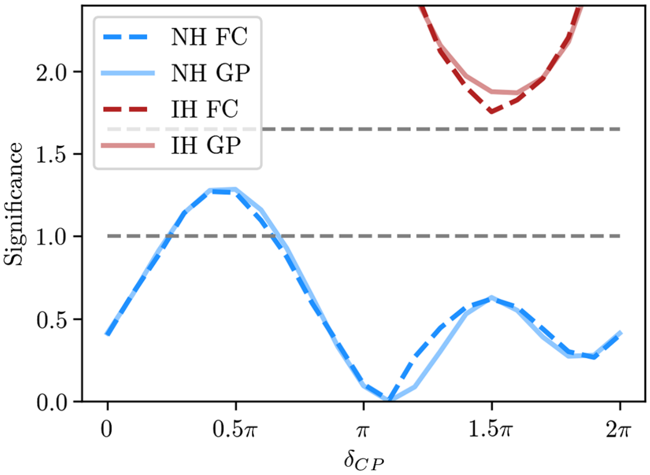

FIG. 7. Example significance curves obtained with the standard Feldman-Cousins and Gaussian process algorithms mostly overlap, especially when the significance is close to 1σ and 1.6σ as desired. In this case, the IH is rejected at 1.6σ level and the NH has the same 1σ confidence interval.

As the $\mathcal{GP}$ algorithm is iterative, we can calculate the accuracy at the end of each iteration with fixed computation. When the computation reaches 20% of that required by standard FC, we stop the algorithm and calculate the absolute error as the difference in confidence interval end points. Figure 8 shows that the median accuracy reaches 1 with less than 20% of computation and the error is no more than 0.1π for most data sets. As δ_{CP} ranges from 0 to 2π and there are only 20 grid points, an error of 0.1π is just one grid point. With a finer grid, we expect the performance of the $\mathcal{GP}$ algorithm to improve.

B. Two-dimensional confidence contours

To find the two-dimensional confidence contours under hierarchy constraints, the $\mathcal{GP}$ algorithm approximates the p-value surface on the parameter grid as shown in Fig. 9 and specifically prioritizes points on the contour boundaries. Grid points below a certain value are included in the confidence contour at that level. To make the final smooth contours in Fig. 10, we use Fourier smoothing to draw the closest elliptical curves. We can observe that the FC and $\mathcal{GP}$ contours overlap in the same areas. In fact, the area difference between the contours is on the same order of magnitude with Fourier smoothing.

Similarly, we use both algorithms on 200 different data sets to find the 68% and 90% confidence contours and calculate the grid point classification accuracy after each iteration up to 10% of the standard FC computation. A concern is that contours with larger area could require more computation to achieve the same accuracy as there are more points along the boundary. We address this concern by stratifying contours by area quartile and plotting median accuracy as a function of computation. Figure 11 shows that the median accuracy reaches 1 with less than 10% of computation and contour area does not have an effect. The reason is that while larger contours have more points on the boundary, smaller contours are more difficult to locate precisely. Overall, it takes roughly the same computation to probe the p-value surface accurately, so the $\mathcal{GP}$ algorithm should have similar performance regardless of the contour size.

Lastly, we are interested in where the computational savings come from. We keep track of the number of grid points explored by the $\mathcal{GP}$ algorithm and the number of simulations at each point for the 200 data sets. Figure 12 shows that the algorithm explores about half of the total grid points, and on average only about 300 Monte Carlo simulations are done instead of 2000 in standard FC. We conclude that most of the computational savings come from performing fewer Monte Carlo simulations; skipping grid points nearly doubles the computational savings. As mentioned earlier, the advantage of the $\mathcal{GP}$ algorithm could be greater on a finer grid.

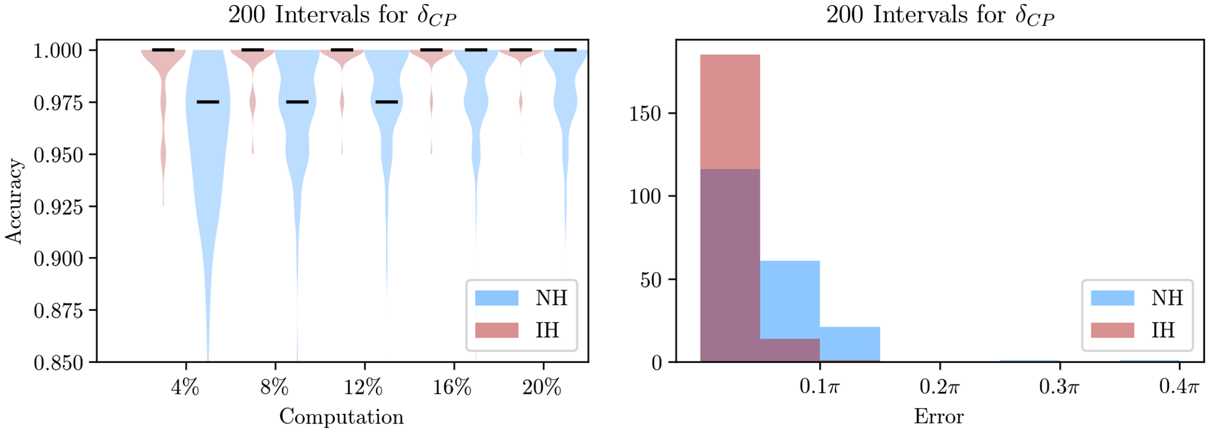

FIG. 8. Relative accuracy of the confidence intervals in terms of correctly included grid points as a function of computation (left) and the distribution of absolute errors for both normal and inverted hierarchies (right).

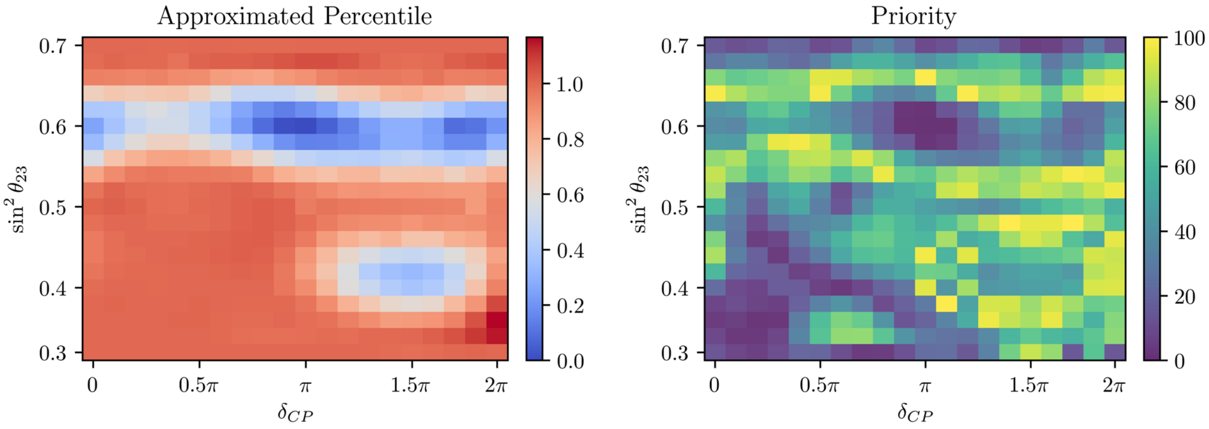

FIG. 9. $\mathcal{GP}$ approximated percentile ($1 - p$-value) on the 20×20 grid for $\sin^2\theta_{23}$ vs δ_{CP} (left) and the priority to sample points from the grid (right). Notice that the points near 68% and 90% have the highest priority.

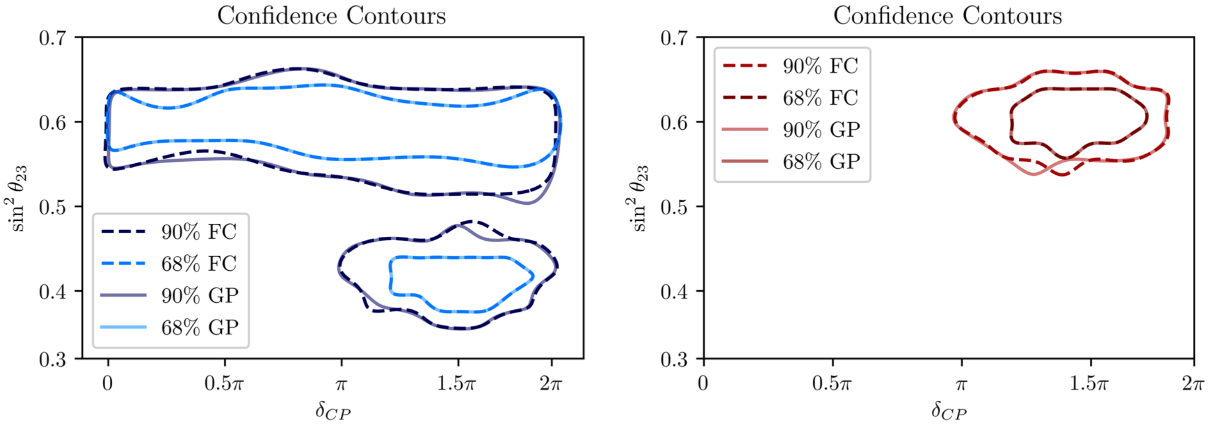

FIG. 10. Confidence contours for the same data constrained to normal (left) and inverted hierarchies (right). The true (dashed) and approximated (transparent) contours are almost indistinguishable.

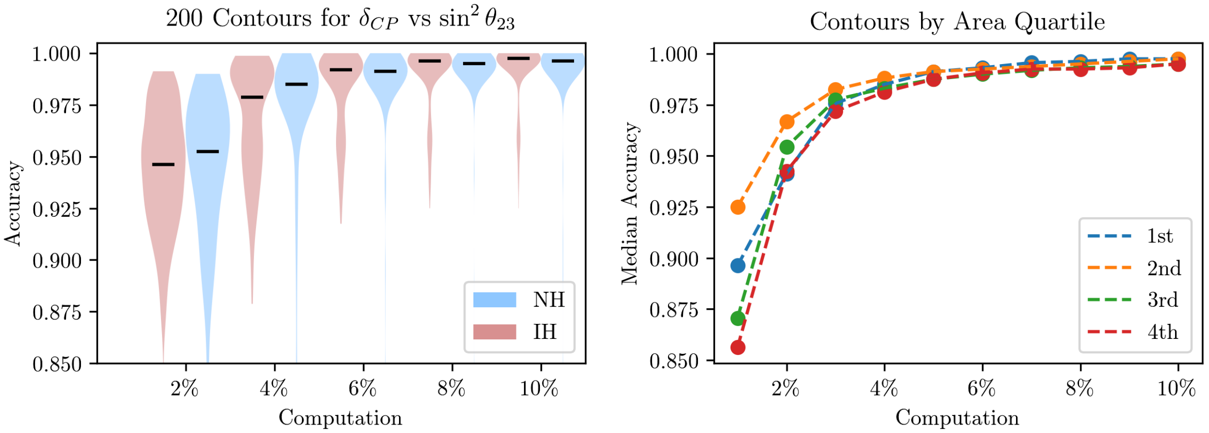

FIG. 11. Relative accuracy of the confidence contours as a function of computation (left) and median accuracy stratified by area as a function of computation (right).

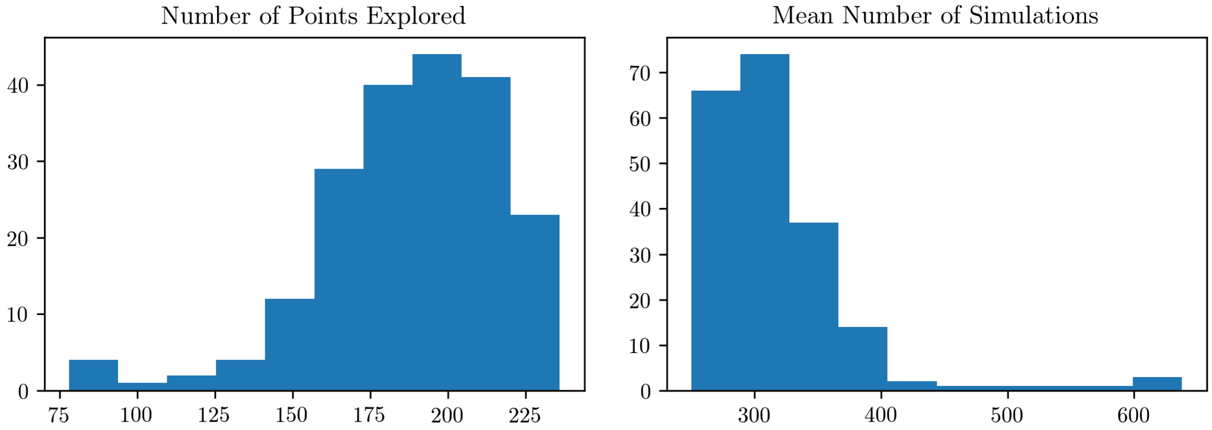

FIG. 12. Distribution of the number of points explored on the grid (left) and distribution of the average number of Monte Carlo experiments simulated at a point (right).

V. DISCUSSION

The proposed algorithm significantly accelerates the Feldman-Cousins approach wherein experiments have to devote enormous computational resources in order to estimate uncertainties in neutrino oscillation parameters [19]. This could also prove useful in estimating confidence intervals from a combined fit of neutrino oscillation results from different experiments when the respective likelihood functions are available. While we design the $\mathcal{GP}$ based construction in the neutrino oscillation context, the $\mathcal{GP}$ approximation does not have a particular parametric form. The same idea can therefore be applied to many other scenarios where the confidence interval construction for a continuous parameter over a bounded region normally proceeds via the unified approach.

ACKNOWLEDGMENTS

The work of L. L. and P. B. is in part supported by NSF Grant No. 1839429 to P. B. The work of J. B. and N. N. is in part supported by DOE Grants No. DE-SC0019077 and No. DE-SC0009920-TaskB to J. B.

[1] G. J. Feldman and R. D. Cousins, Unified approach to the classical statistical analysis of small signals, Phys. Rev. D **57**, 3873 (1998).

[2] K. Abe *et al.*, Measurement of neutrino and antineutrino oscillations by the T2K experiment including a new additional sample of ν_e interactions at the far detector, Phys. Rev. D **96**, 092006 (2017).

[3] M. A. Acero *et al.*, New constraints on oscillation parameters from ν_e appearance and ν_μ disappearance in the NOvA experiment, Phys. Rev. D **98**, 032012 (2018).

[4] P. Adamson *et al.*, Search for Sterile Neutrinos in MINOS and MINOS+Using a Two-Detector Fit, Phys. Rev. Lett. **122**, 091803 (2019).

[5] J. Močkus, On Bayesian methods for seeking the extremum, in *Optimization Techniques IFIP Technical Conference* (Springer, 1975), pp. 400–404.

[6] B. Pontecorvo, Mesonium and antimesonium, Zh. Eksp. Teor. Fiz. 33 (1957).

[7] Q. R. Ahmad *et al.*, Direct Evidence for Neutrino Flavor Transformation from Neutral-Current Interactions in the Sudbury Neutrino Observatory, Phys. Rev. Lett. **89**, 011301 (2002).

[8] Y. Fukuda *et al.*, Evidence for Oscillation of Atmospheric Neutrinos, Phys. Rev. Lett. **81**, 1562 (1998).

[9] F. P. An *et al.*, Measurement of electron antineutrino oscillation based on 1230 days of operation of the Daya Bay experiment, Phys. Rev. D **95**, 072006 (2017).

[10] G. Drexlin, V. Hannen, S. Mertens, and C. Weinheimer, Current direct neutrino mass experiments, Adv. High Energy Phys. 2013 (2013).

[11] R. Barlow, Extended maximum likelihood, Nucl. Instrum. Methods Phys. 293 (1990).

[12] C. E. Rasmussen, Gaussian processes in machine learning, in *Advanced Lectures on Machine Learning* (Springer, 2004), pp. 63–71.

[13] G. J. Hahn and W. Q. Meeker, *Statistical Intervals: A Guide for Practitioners* (John Wiley & Sons, 2011), Vol. 92.

[14] Y. Smirnov, The MSW effect and matter effects in neutrino oscillations, Phys. Scr. T **2005**, 57 (2005).

[15] J. A. Formaggio and G. P. Zeller, From eV to EeV: Neutrino cross sections across energy scales, Rev. Mod. Phys. **84**, 1307 (2012).

[16] Particle Data Group, Neutrino masses, mixing and oscillations (2017).

[17] R. Brun and F. Rademakers, ROOT: An object oriented data analysis framework, Nucl. Instrum. Methods Phys. Res., Sect. A **389**, 81 (1997).

[18] F. Pedregosa *et al.*, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. **12**, 2825 (2011).

[19] A. Sousa, N. Buchanan, S. Calvez, P. Ding, D. Doyle, A. Himmel, B. Holzman, J. Kowalkowski, A. Norman, and T. Peterka, Implementation of Feldman-Cousins corrections and oscillation calculations in the HPC environment for the NOvA Experiment, in *Proceedings of the 23rd International Conference on Computing in High-Energy and Nuclear Physics (CHEP 2018), Sofia, Bulgaria, 2018* (to be published).