Accelerating Optimal Experimental Design for Robust Synchronization of Uncertain Kuramoto Oscillator Model Using Machine Learning

Hyun-Myung Woo, Youngjoon Hong, Bongsuk Kwon, and Byung-Jun Yoon, *Senior Member, IEEE*

***Abstract*—Recent advances in objective-based uncertaintyquantification (objective-UQ) have shown that such a goal-driven approach for quantifying model uncertainty is extremely usefulin real-world problems that aim at achieving specific objectives based on complex uncertain systems. Central to this objective-UQ is the concept of mean objective cost of uncertainty (MOCU), which provides effective means of quantifying the impact of uncertainty on the operational goals at hand. MOCU is especially useful for optimal experimental design (OED) as the potential efficacy of an experimental (or data acquisition) campaign can be quantified by estimating the MOCU that is expected to remain after the campaign. However, MOCU-based OED tends to be computationally expensive, which limits its practical applicability. In this paper, we propose a novel machine learning (ML) scheme that can significantly accelerate MOCU computation and expedite MOCU-based experimental design. The main idea is to use an ML model to efficiently search for the optimal robust operator under model uncertainty, a necessary step for computing MOCU. We apply the proposed ML-based OED acceleration scheme to design experiments aimed at optimally enhancing the control performance of uncertain Kuramoto oscillator models. Our results show that the proposed scheme results in up to 154-fold speed improvement without any degradation of the OED performance.**

***Index Terms*—Approximate mean objective cost of uncertainty (MOCU), Kuramoto model, machine learning (ML), objective-based uncertainty quantification (objective-UQ), optimal experimental design (OED) acceleration.**

Manuscript received July 3, 2021; revised October 24, 2021 and November 20, 2021; accepted November 21, 2021. Date of publication November 26, 2021; date of current version December 17, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yao Xie. The work of Hyun-Myung Woo and Byung-Jun Yoon was supported in part by the NSF under Award 1835690 and the DOE under Award DE-SC0019303. The work of Youngjoon Hong was supported in part by Basic Science Research Program through the National Research Foundation of Korea, and in part by the Ministry of Education under Grant NRF-2021R1A2C1093579. The work of Bongsuk Kwon was in part by Basic Science Research Program through the National Research Foundation of Korea supported in part by the Ministry of Science and ICT and future planning under Grant NRF-2020R1A2C1A01009184. *(Corresponding author: Byung-Jun Yoon.)*

Hyun-Myung Woo is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: larcwind@tamu.edu).

Youngjoon Hong is with the Department of Mathematics, Sungkyunkwan University, Suwon 16419, Republic of Korea (e-mail: hongyj@skku.edu).

Bongsuk Kwon is with the Department of Mathematical Sciences, Ulsan National Institute of Science and Technology, Ulsan 44919, Republic of Korea (e-mail: bkwon@unist.ac.kr).

Byung-Jun Yoon is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA, and also with the Brookhaven National Laboratory, Upton, NY 11973 USA (e-mail: bjyoon@ece.tamu.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2021.3130967, provided by the authors.

Digital Object Identifier 10.1109/TSP.2021.3130967

I. Introduction

MANY real-world engineering applications involve mathematical modeling of complex systems, where the constructed models are used for designing operators–such as controllers, filters, classifiers, estimators–that can effectively achieve engineering goals of interest. For example, one may be interested in building a network model representing the transcription regulations in micro-organisms that regulate their metabolism [1]. The resulting model may be used to infer the potential impacts of modifications in the transcription regulatory network (TRN) on the metabolism of interest, for example, predicting the metabolic flux changes that result from the deletion of one or more transcription factors. In this example, the engineering goal may be predicting the optimal genetic modification in the TRN that will lead to maximizing the production of a metabolite of interest. In fact, designing optimized strains of micro-organisms for ethanol overproduction [2] is an active area of research due to its implications in efficient bio-energy production.

A fundamental challenge in the aforementioned application as well as many other real-world engineering problems involving complex systems is the difficulty of accurate model construction. While one may have ample training data for model inference, the data size may nevertheless pale in comparison to the complexity of the system being modeled. Prior knowledge, if available, may also aid in improving model construction, but the final model is likely to still have substantial uncertainties. Consequently, a critical question is how one may reliably and optimally achieve the given engineering goals in the presence of model uncertainty. Furthermore, when one has the experimental budget for the acquisition of additional data or relevant knowledge (e.g., via hypothesis testing), how should the experimental campaigns be designed to maximize the expected "return on investment"?

While these are fundamental problems in modern engineering with a long and rich history [3], [4], it has been recently shown that a novel Bayesian paradigm for objective-based uncertainty quantification (objective-UQ) based on the mean objective cost of uncertainty (MOCU) [5], [6] can effectively address the optimal design of operators and experiments for complex uncertain systems [7]–[14]. The core idea underlying the MOCU-based

optimal experimental design (OED) is that, when dealing with complex uncertain models, one should quantify the model uncertainty in an objective-based manner and design experiments that can reduce the uncertainty that impacts one's operational goals. By focusing on the uncertainty that matters to the operation to be performed, the experimental budget can be efficiently used for optimizing the operational performance. To date, the efficacy of MOCU-based OED has been demonstrated in various systems, including experimental design for robust intervention in gene regulatory networks (GRNs) [9], [10] and that for robust synchronization of inter-coupled Kuramoto oscillators [7].

One practical challenge that limits the potential applicability of the MOCU-based OED scheme is its high computational cost, as discussed in [7], [9]. The computation of MOCU involves identifying the optimal robust operator for an uncertainty class that consists of all possible models (e.g., models with different parameter values) as well as evaluating expectations based on high-dimensional prior (or posterior) probability distributions. Except for very simple cases, there is no closed-form expression for the optimal robust operator and the expectations have to be evaluated numerically [7]. As a result, the evaluation of MOCU involves costly optimization to find the optimal robust operator as well as extensive sampling of the uncertain model parameters from the uncertainty class to obtain reliable estimates, which may make the cost of MOCU computation formidably high in many applications.

In this paper, we tackle this issue by adopting a machine learning (ML) approach for an efficient design of the optimal robust operator, thereby significantly accelerating the computation of MOCU as well as the MOCU-based experimental design. To the best of our knowledge, this is the first study that investigates adopting ML to accelerate MOCU-based OED. In order to develop and validate this ML-based OED acceleration scheme, we focus on designing experiments that can enhance the robust control of uncertain Kuramoto models that was investigated recently in [7]. A Kuramoto model [15] consists of a network of interconnected oscillators, whose dynamics are described by coupled ordinary differential equations (ODEs). The Kuramoto oscillator model has been widely studied in various fields across engineering, physics, chemistry, and biology, due to its capability to model interesting collective behavior (e.g., global/partial synchronization) that emerge in complex networks [16]–[27]. For example, a *microgrid* system with droop-controlled inverters can be mathematically cast as a Kuramoto model, where the synchronization failure of the model corresponds to a power outage in the microgrid [22], [24], [27]–[31]. Another interesting example is the application of the Kuramoto model for studying brain dynamics [18], [19], [25], [26], where the synchronization phenomena may be associated with neurodegenerative diseases [25], [32]. We show that our proposed ML-based OED acceleration scheme can improve the speed of MOCU-based experimental design by $104 \sim 154$ times without degrading the OED performance.

The two major contributions of this paper are as follows. First, we propose an ML-based scheme for the acceleration of MOCU-based OED, which leads to significant speed improvement without performance degradation. Second, we present a comprehensive analysis of ML-based MOCU estimation and validate its performance in the context of OED.

The paper is organized as follows. In Section II, we provide a brief review of the Kuramoto model and the MOCU-based OED strategy for uncertain Kuramoto models. We propose the ML-based OED acceleration strategy in Section III. In Section IV and Section V, we evaluate the performance of the proposed scheme for approximate MOCU computation and experimental design, respectively. We conclude the paper in Section VI with further discussions and potential future research directions.

II. Overview of Optimal Experimental Design Strategy for the Uncertain Kuramoto Model

In this section, we provide a brief review of the OED strategy for uncertain Kuramoto oscillator models, which we originally proposed in our recent work [7]. We begin the section with an introduction to the Kuramoto model, followed by a brief description of the robust synchronization problem for uncertain Kuramoto models. Given an uncertain Kuramoto model, we describe how the MOCU can be used to quantify the impact of the model uncertainty on the control synchronization performance and how the MOCU-based OED strategy can be used to effectively reduce the uncertainty that matters to the objective at hand–*i.e.*, optimal robust synchronization of the Kuramoto model in the presence of uncertainty.

A. Uncertain Class of Kuramoto Models

Consider the Kuramoto model that consists of N interacting oscillators described by the following ODEs:

$$\dot{\theta}_i(t) = \omega_i + \sum_{j=1}^{N} a_{i,j} \sin\left(\theta_j(t) - \theta_i(t)\right), \tag{1}$$

for $i = 1, 2, \ldots, N$, where $\theta_i(t)$ is the instantaneous phase of the ith oscillator at time t, ω_i is the natural frequency of the ith oscillator, and $a_{i,j}$ is the coupling strength between the ith and jth oscillators. Kuramoto models have been widely studied to investigate the synchronization phenomena in various biological, chemical, or engineered oscillator systems, whose primary interest is whether the oscillators in a given Kuramoto model will get frequency synchronized as follows:

$$\lim_{t\to\infty} |\dot{\theta}_i(t) - \dot{\theta}_j(t)| = 0, \tag{2}$$

for $1 \le i, j \le N$. For example, it has been shown that modern smart grid networks referred to as microgrids can be modeled as a network of Kuramoto model oscillators, where the synchronization phenomena of the Kuramoto model are closely tied with the stability of the power grid network [28]–[31]. Furthermore, in neuroscience studies, brain network synchronization has been shown to be associated with various neurological disorders, where excessive neuronal activities can be represented as a global synchronization of the Kuramoto model [18], [25], [26], [32], [33]. While conditions for synchronization have been extensively studied for homogeneous Kuramoto models with uniform coupling strength [34]–[36], there is yet no closed-form

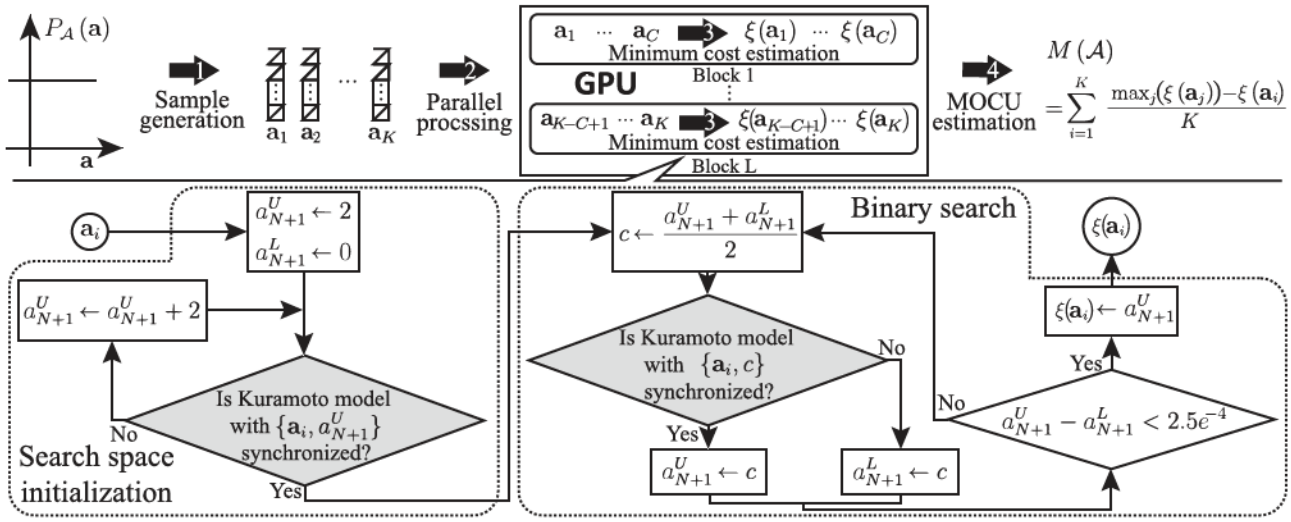

Fig. 1. Illustration of the original sampling-based mean objective cost of uncertainty (MOCU) computation scheme in [7]. For reliable MOCU estimation, a relatively large sample size K is needed (step 1). The sampling-based estimation scheme takes advantage of GPU programming for massive parallelization of the sampling operation. Specifically, we group the K sample points $\{\mathbf{a}_i\}$, $i = 1, 2, \ldots, K$, into L blocks (step 2), and the GPU processes L sample points in different processing blocks in parallel (step 3). Within block l, based on a sample point $\mathbf{a}_i$ that specifies a Kuramoto model (sampled from the uncertainty class), we find a valid search range $[a^L_{N+1}, a^U_{N+1}]$ that contains at least one valid solution that leads to global synchronization of the Kuramoto model (left bottom part). In the next phase (right bottom part), we find the solution with the smallest cost $\xi(\mathbf{a}_i)$ through a binary search, reducing the search range by half in every iteration. Finally, we compute the MOCU $M(\mathcal{A})$ of the uncertainty class $\mathcal{A}$ based on the K estimates $\xi(\mathbf{a}_i)$, $i = 1, 2, \ldots, K$, (step 4).

solution that can be used to predict the asymptotic synchronization of a general heterogeneous Kuramoto model based on its parameters.

In a real-world setting, the parameters of the Kuramoto model, which represents a complex network of oscillators, may not be completely known. For example, while it may be relatively easy to accurately estimate the natural frequency of each oscillator, in the absence of interactions with other oscillators, it will be practically challenging to accurately measure the coupling strengths between all oscillators in a large network. This uncertainty gives rise to an uncertainty class of Kuramoto models, which contains all possible Kuramoto models that are consistent with our prior knowledge regarding the true model and/or available observation data. Under this setting, our primary interest is how we can apply robust control to the uncertain Kuramoto model, comprised of a network of oscillators whose natural frequency ω_i is known but their coupling strength $a_{i,j}$ is only known up to a range $a_{i,j} \in [a^L_{i,j}, a^U_{i,j}]$. We denote the uncertainty class of all possible Kuramoto models as $\mathcal{A}$, which consists of all parameter vector $\mathbf{a} = [a_{1,2}, a_{1,3}, \ldots, a_{N-1,N}]^T \in \mathcal{A}$ that satisfies the given constraints. As in the previous study [7], we assume that prior distribution $P_{\mathcal{A}}(\mathbf{a})$ is uniformly distributed. However, this is not necessary. Non-uniform priors may be assumed, or custom priors may be constructed based on available prior domain knowledge [37], [38].

B. Robust Control of Uncertain Kuramoto Models

Suppose that we are interested in synchronizing an uncertain Kuramoto model that consists of N interacting oscillators, whose interaction strengths are only known up to a range, via external control. We adopt the synchronization method proposed in [7] that introduces an additional oscillator as a global "synchronizer" to the original model. Let the natural frequency of this $(N+1)$th oscillator be $\omega_{N+1} = \frac{1}{N}\Sigma_{i=1}^{N}\omega_i$, and we assume that this control oscillator interacts with all oscillators in the original model with a uniform coupling strength $a_{i,N+1} = a_{N+1}$, $\forall i$, which is a control parameter. The addition of the control oscillator augments the Kuramoto model as follows:

$$\dot{\theta}_i(t) = \omega_i + \sum_{j=1}^{N} a_{i,j}\sin(\theta_j(t) - \theta_i(t)) + a_{N+1}\sin(\theta_{N+1}(t) - \theta_i(t)), \tag{3}$$

for $i = 1, 2, \ldots, N+1$. As the increase of the coupling strength a_{N+1} will in practice lead to an increase of the control cost, our control objective is to find a minimum a_{N+1} that guarantees the asymptotic frequency synchronization of the Kuramoto model despite the uncertainty. If we had complete knowledge about the coupling strength $\mathbf{a}$, we would be able to find the optimal (minimum) coupling strength $a_{N+1} = \xi(\mathbf{a})$ that ensures synchronization by gradually increasing the value of a_{N+1} from 0 until synchronization is achieved. A more efficient approach will be to perform a binary search as illustrated in Fig. 1 (see the blow-up figure at the bottom). In the presence of uncertainty, we have to ensure that the control oscillator will be able to achieve synchronization for any $\mathbf{a} \in \mathcal{A}$. For this reason, we have chosen $a_{N+1} = \xi^*(\mathcal{A})$ as follows:

$$\xi^*(\mathcal{A}) = \max_{\mathbf{a}\in\mathcal{A}} \xi(\mathbf{a}), \tag{4}$$

which is the smallest a_{N+1} that guarantees global synchronization of the uncertain Kuramoto oscillators.

C. Objective-Based Quantification of the Uncertainty in the Kuramoto Model

Given an uncertain Kuramoto model, the expected impact of this model uncertainty on the operational goal–in this case, the global frequency synchronization of the Kuramoto oscillators–can be quantified by the MOCU [5]. For a given uncertainty class $\mathcal{A}$, MOCU $M(\mathcal{A})$ can be computed by:

$$M(\mathcal{A}) = E_{\mathcal{A}}\left[\xi^*(\mathcal{A}) - \xi(\mathrm{a})\right], \tag{5}$$

where $\xi^*(\mathcal{A})$ is the cost of the optimal robust control and $\xi(\mathrm{a})$ is the cost of the optimal control for the specific model configured with a specific parameter set a. As shown in (5), MOCU $M(\mathcal{A})$ quantifies the expected cost increase for applying the optimal robust control (which is inevitable to maintain robust control performance in the presence of uncertainty) instead of the model-specific optimal control (which cannot be applied in practice as the true model is unknown). In this study, the optimal robust interaction strength (cost) $\xi^*(\mathcal{A})$, ensuring that the uncertain Kuramoto model is synchronized by the added control oscillator while keeping the control cost minimum, is given by (4).

D. Numerical Computation of Mean Objective Cost of Uncertainty

In general, there is no closed-form expression of (5), as a result of which the MOCU $M(\mathcal{A})$ for uncertainty class $\mathcal{A}$ computation requires a numerical approximation. One practical way to compute the MOCU $M(\mathcal{A})$ is to take a sampling-based approach to approximate it through the empirical expectation of the differential cost based on samples drawn from the distribution $P_{\mathcal{A}}(\mathrm{a})$.

Figure 1 illustrates the sampling-based MOCU computation process. First, we draw K sample points $\{\mathrm{a}_i\}$, $i = 1, 2, \ldots, K$, from $P_{\mathcal{A}}(\mathrm{a})$. Then, for each sample point a_i, which is a potential true model parameter in the uncertainty class $\mathcal{A}$, we estimate the minimum coupling strength $\xi(\mathrm{a}_i)$ of the control oscillator that assures the asymptotic frequency synchronization of the Kuramoto model under control. To this aim, we consider a binary search to find the minimum coupling strength $\xi(\mathrm{a}_i)$ efficiently, as depicted in the dotted box at the bottom of Fig. 1. Specifically, we start with a broad search space that contains at least one coupling strength synchronizing the system. At each iteration, we solve the ODEs of the Kuramoto model augmented with the control oscillator whose coupling strength a_{N+1} is set to the median value c of the current search space: $a_{N+1} \leftarrow c = (a^U_{N+1} + a^L_{N+1})/2$. If the system under control is synchronized, we update the upper bound of the search space to the median value: $a^U_{N+1} \leftarrow c$. Otherwise, we set the lower bound of the search space to the median value: $a^L_{N+1} \leftarrow c$. The binary search continues until we find the minimum coupling strength $\xi(\mathrm{a}_i)$, for the given sample point a_i, which is within a specified tolerance level (set to 2.5×10^{-4} in this study). Based on the K sample points, we can obtain the MOCU $M(\mathcal{A})$ as follows:

$$M(\mathcal{A}) = \frac{1}{K}\sum_{i=1}^{K}\left(\max_j\left(\xi(\mathrm{a}_j)\right) - \xi(\mathrm{a}_i)\right). \tag{6}$$

Note that the accuracy of this numerical approximation of MOCU is dependent on the sample size K. In general, a larger K generally leads to a more accurate MOCU estimation. However, at the same time, the computational cost increases as the sample size increases. We can reduce the computational time for numerical MOCU computation by exploiting parallelism. For example, estimating the optimal cost $\xi(\mathrm{a}_i)$ of a sample point a_i is an independent process to those of the other samples a_j, $j \neq i$, which can be processed in a parallel manner with powerful parallel processors. In fact, the sampling-based MOCU computation in [7] takes advantage of GPU programming with compute unified device architecture (CUDA), in which 200 sample points are processed in parallel at a given time–*i.e.*, $L = 200$. However, for each sample point a_i, the estimation of the minimum cost $\xi(\mathrm{a}_i)$ via binary search (step 3 in Fig. 1) is a highly sequential process–which involves repeatedly solving the ODEs of the corresponding Kuramoto model and verifying whether or not the model is globally synchronized (*i.e.*, not amenable to parallelization). In Section III, we present a novel solution via ML that can effectively address this performance bottleneck, and thereby accelerate the numerical computational of MOCU by several orders of magnitude.

E. Designing Optimal Experiments for Effective Uncertainty Reduction

The significance of objective-UQ using MOCU is that it enables the design of experiments that focus on reducing the model uncertainty that matters. More specifically, as MOCU quantifies the expected cost increase (relevant to our operational goal) due to model uncertainty, it can be used to quantify the expected impact of a potential experiment on reducing the model uncertainty that affects the operational performance, hence how effective the experiment will be in reducing the operational cost.

The MOCU-based OED strategy for uncertain Kuramoto models has been recently proposed in [7]. In this study, a realistic experimental design space was considered, where an experiment corresponds to selecting a pair (i, j) of oscillators and observing whether they get spontaneously synchronized in isolation of other oscillators and in the absence of external control. The experimental outcome was a binary value–either synchronized or non-synchronized–based on which the uncertainty of the coupling strength $a_{i,j} \in [a^L_{i,j}, a^U_{i,j}]$ can be reduced. Theorem 1 in [7] reproduced below gives us the necessary and sufficient condition for an oscillator pair to be frequency synchronized:

Theorem 1: Consider the Kuramoto model of two-oscillators:

$$\begin{aligned}\dot{\theta}_1(t) &= \omega_1 + 0.5a\sin\left(\theta_2(t) - \theta_1(t)\right),\\ \dot{\theta}_2(t) &= \omega_2 + 0.5a\sin\left(\theta_1(t) - \theta_2(t)\right),\end{aligned} \tag{7}$$

with the initial angles $\theta_1(0), \theta_2(0) \in [0, 2\pi)$. Then, for any solutions $\theta_1(t)$ and $\theta_2(t)$ to (7), there holds $|\dot{\theta}_1(t) - \dot{\theta}_2(t)| \to 0$ as $t \to \infty$ if and only if $|\omega_1 - \omega_2| \leq a$. ■

According to Theorem 1, the Kuramoto oscillator pair (i, j) becomes frequency synchronized $\lim_{t\to\infty}|\dot{\theta}_i(t) - \dot{\theta}_j(t)| = 0$ if and only if $\frac{|\omega_i - \omega_j|}{2} \leq a_{i,j}$. As a result, if the two oscillators are observed to be synchronized, we can decrease the upper

bound $a^U_{i,j}$ to $|\omega_i - \omega_j|/2$. Otherwise, we can increase the lower bound $a^L_{i,j}$ to $|\omega_i - \omega_j|/2$. Since the experimental outcome is unknown in advance, we need to consider both possible outcomes to quantify the expected impact of a given experiment on reducing the objective uncertainty. To formalize this, let $O_{i,j}$ be a binary random variable representing the outcome of the pairwise synchronization experiment for the oscillator pair (i, j). Then, the expected remaining MOCU $R(i, j)$ is given by:

$$R(i,j) = E_{O_{i,j}}[M(\mathcal{A}|O_{i,j})] = \sum_{o\in\{0,1\}} P(O_{i,j}=o)\, M(\mathcal{A}|O_{i,j}=o), \quad (8)$$

where $M(\mathcal{A}|O_{i,j})$ is the conditional MOCU given $O_{i,j}$. The conditional MOCU $M(\mathcal{A}|O_{i,j}=o)$ given an experimental outcome $O_{i,j}=o$ can be computed by reducing the uncertainty class as previously described and numerically computing the MOCU of this reduced uncertainty class. The probability $P(O_{i,j}=o)$ can be derived in a straightforward manner, based on $P_{\mathcal{A}}(\mathbf{a})$ (see [7] for further details). The $R(i, j)$ in (8) quantifies the MOCU that is expected to remain after performing the pairwise synchronization experiment for the pair (i, j).

So, how should we prioritize the potential $\binom{N}{2}$ experiments? Naturally, the optimal choice will be to choose the experiment with the smallest $R(i, j)$:

$$(i^*, j^*) = \arg\min_{(i,j)\in\mathcal{E}} R(i,j), \quad (9)$$

as experiment (i^*, j^*) is expected to most effectively reduce the objective uncertainty among all potential experiments. In practice, rather than performing a single best experiment, we may perform a sequence of experiments prioritized by (9). In theory, $R(i, j)$ needs to be re-estimated after performing the predicted optimal experiment and observing its outcome, as it changes the uncertainty class, hence the expected remaining MOCU for the potential subsequent experiments. However, empirically, $R(i, j)$ computed based on the original uncertainty class $\mathcal{A}$ is a robust indicator of the efficacy of the potential experiments, which we will demonstrate in Section V.

F. Computational Complexity of Experimental Design

The overall computational complexity for predicting the optimal experiment is as follows:

$$O(TKN^4\, L^{-1} \log \epsilon), \quad (10)$$

where T is the time duration for solving the ODEs using the Runge-Kutta method (to check for asymptotic global frequency synchronization among the Kuramoto model oscillators), K is the sample size for numerical computation of MOCU, N is the number of oscillators in the Kuramoto model, L is the number of parallel processing blocks in GPU, and ϵ is the tolerance level for the binary search (set to $\epsilon = 2.5 \times 10^{-4}$ in this study). Note that the complexity for computing MOCU is $O(TKN^2\, L^{-1} \log \epsilon)$, where predicting the optimal experiment involves computing MOCU $2 \cdot \binom{N}{2}$ times to calculate $R(i, j)$ given by (8) for all oscillators pairs. As we can see in (10), the computational cost for OED sharply increases as the size N of the Kuramoto model increases, which limits the practical applicability of the OED scheme [7] for large models. For example, when $T = 5$, $K = 20,480$, and $L = 128$, respectively, identifying the optimal experiment (i^*, j^*) for the uncertain Kuramoto model operating on five oscillators required 650 seconds on average. However, it took 3,171 seconds to determine the optimal experiment (i^*, j^*) for the uncertain Kuramoto model with seven oscillators.

III. Accelerating Experimental Design via Machine Learning

We propose an ML approach for accelerating the quantification of the objective system uncertainty. As we discussed in the previous section, in real-world applications that typically involve the control consisting of highly non-linear sequential operations, the effective computational complexity is critically dependent on the computational complexity of the control rather than the number of samples. The proposed approach learns a surrogate model for (part of) the operations of the control for estimation of the control cost for a system, thereby reducing the effective computational complexity that cannot be further reduced by parallelism. Recently, there has been an increasing number of studies investigating the application of deep learning (DL) methods to scientific computation, including approximating and solving differential equations (DEs) (e.g., see [39]–[41] and references therein). However, it is worth noting that the primary focus of our current study does not lie in solving ODE systems via deep network models but in the accelerated design of optimal experiments based on the objective-based UQ via the concept of MOCU. Rather than aiming at a fast solution of DEs, our goal is to efficiently design experiments that can most effectively reduce model uncertainty, thereby optimally enhancing the control performance of uncertain Kuramoto oscillator models.

As discussed in Section II-D, estimating the MOCU of the uncertain Kuramoto model based on the sampling approach involves a binary search for each sample $\mathbf{a}_i$, where at each iteration solving the corresponding ODEs and determining if the system under control is synchronized or not. From a broad perspective, at each iteration, these operations, the gray box in Fig. 1, are nothing but a binary classification problem. Hence, if we collect enough samples to build an accurate classifier, we replace such a process with the binary classifier, which is computationally efficient. In this study, we considered a fully connected neural network (fcNN) with only one hidden layer, possibly the simplest ML structure that we can think of.

The proposed approach on the MOCU-based OED framework is realized by replacing part of the operations of control with the trained model for the estimation of the expected remaining MOCU $R(i, j)$ highlighted in gray in Fig. 2. Hence we focus on the difference in quantifying the expected remaining MOCU $R(i, j)$ between the proposed ML-based approach and the original approach that manually determines the synchronization of the Kuramoto model. To compute the expected remaining MOCU $R(i, j)$ we first need to estimate conditional MOCU $M(\mathcal{A}|O_{i,j}=o)$ given the experimental outcome $O_{i,j}=o \in \{0, 1\}$ (*i.e.*, synchronized or not) as derived in (8).

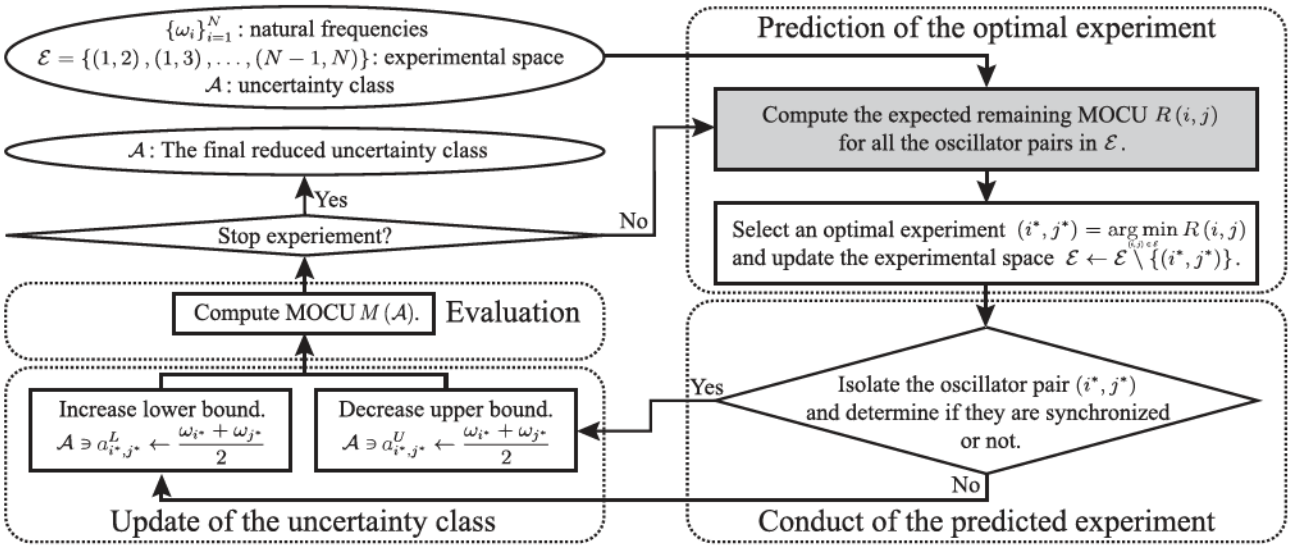

Fig. 2. Illustration of the MOCU-based optimal experimental design (OED) loop. First, we compute the expected remaining MOCU $R(i,j)$ for all possible experiments (i,j) in the experimental design space $\mathcal{E}$ based on the current uncertainty class $\mathcal{A}$. Next, we identify the optimal experiment (i^*,j^*) that has the smallest expected remaining MOCU such that $(i^*,j^*) = \arg\min_{(i,j)\in\mathcal{E}} R(i,j)$. In the second phase (right bottom), we conduct the selected experiment (i^*,j^*) and remove the performed experiment from the experimental space $\mathcal{E}$. Specifically, in this experiment, we isolate the selected oscillator pair (i^*,j^*) and determine whether or not they get synchronized without external control. Based on the experimental outcome, we update the uncertainty class accordingly [7]. Finally, we evaluate the actual efficacy of the conducted experiment by computing the MOCU of the updated uncertainty class $\mathcal{A}$. We iterate this experimental loop until the experimental space becomes empty (*i.e.*, there are no more experiments left to be performed).

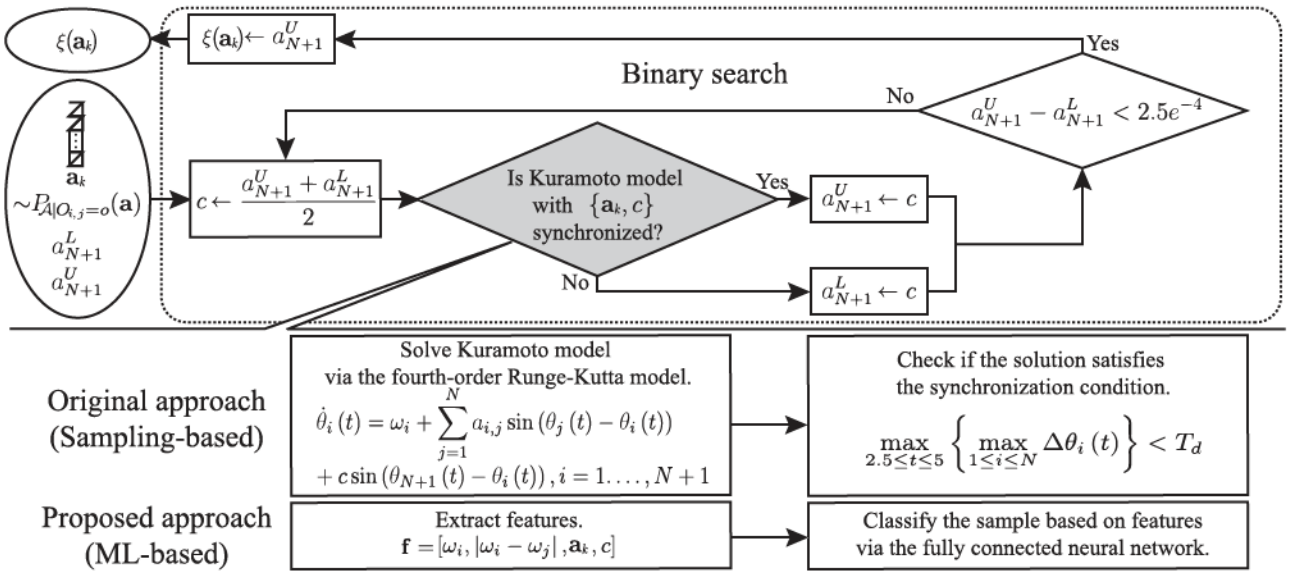

Fig. 3. Comparison between the original sampling-based estimation scheme adopted in [7] and the proposed machine learning-based (ML-based) estimation scheme. The proposed scheme obviates the need for repeatedly solving the coupled ordinary differential equations (ODEs) within the binary search routine to find the optimal robust coupling strength illustrated in Fig. 1. This significantly enhances the computational efficiency of MOCU estimation.

Specifically, as we described in Section II-D, we compute the control cost $\xi(\mathbf{a}_i)$ of all samples $\mathbf{a}_i, i = 1,2,\ldots,K$, drawn from the posterior uncertainty class distribution $P_{\mathcal{A}|O_{i,j}=o}(\mathbf{a})$ updated according to the experimental outcome $O_{i,j} = o$ as shown in Fig. 2. Figure 3 shows the difference in estimating the control cost $\xi(\mathbf{a}_i)$ of sample $\mathbf{a}_k$ between the proposed approach and the original approach. Both approaches find a numerical solution through the binary search that is a sequential process. At each iteration, the coupling strength a_{N+1} of the control oscillator is set to the midpoint $c \leftarrow (a^U_{N+1} + a^L_{N+1})/2$ of the search space. The original approach solves the Kuramoto model determined by the sample $\mathbf{a}_k$ and midpoint c and determines if the solutions are synchronized or not according to criterion (2). On the other hand, the proposed ML-based approach extracts features based on the natural frequencies ω_i, sample $\mathbf{a}_k$, and midpoint c and classifies the feature vector. The search space is then halved according

to the outcome. Note that the computational complexity of the original approach is critically dependent on the time precision and simulation time. Less time precision and shorter simulation time can reduce the overall computational complexity, but such parameters significantly affect the estimation accuracy of the MOCU. On the other hand, MOCU-based OED with the proposed approach is free from such a trade-off at the inference phase as the features are independent of the parameters.

IV. Evaluation of Machine Learning-Based Estimation of Approximate Mean Objective Cost of Uncertainty

A. Configuration of the Simulation

In this section, we demonstrate the efficacy of the proposed ML approach in accelerating the speed of objective-UQ, resulting in a very efficient OED. As described in the previous sections, we considered the OED for the Kuramoto model under uncertainty, where one's operational objective is to ensure synchronization of the model by adding an oscillator for control. For validation, we considered two experimental setups based on uncertain Kuramoto models with five and seven oscillators, respectively. As a reference ODE solver, we used the fourth-order Runge-Kutta method to solve the Kuramoto model sampled at the sampling frequency f_s of 160Hz for five seconds. To determine whether the Kuramoto model is synchronized or not, we used the following criterion:

$$\max_{2.5\le t\le 5}\left(\max_{1\le i\le N}\Delta\theta_i(t)\right) < T_d, \tag{11}$$

where $\Delta\theta_i(t) \triangleq \theta_i(t+(1/f_s)) - \theta_i(t)$, $\theta_i(t)$ is the instantaneous phase of the ith oscillator, and T_d is a threshold of tolerance. We set T_d to 0.001. To estimate the MOCU of a given uncertainty class, we randomly drew 20,480 sample points from the uncertainty class (*i.e.*, $K = 20,480$). We used a *Lambda workstation* equipped with *Intel i9-9960X*, 128 GB memory, and *GeForce RTX 2080 Ti* for the simulations.

At the core of the proposed method lies a binary classifier that accurately classifies the global frequency synchronization of the model when a control oscillator is introduced. To train an accurate classifier, we used an fcNN model with one hidden layer. In that regard, it is essential to extract representative features from the parameters that define the Kuramoto model, such as the number of oscillators, natural frequencies, initial phases, or coupling strength values between oscillators. Inspired by Theorem 1, which gives us the necessary and sufficient condition for pairwise frequency synchronization of Kuramoto oscillators, we used the natural frequencies, the absolute difference between the frequencies, and the corresponding coupling strength values as features. More specifically, given a parameter set that fully determines the Kuramoto model operating on $N+1$ oscillators, we first sort all the natural frequencies in descending order and rearrange the coupling strength accordingly. Then, we construct the corresponding feature set that consists of the sorted natural frequencies, the absolute difference of the natural frequencies of all oscillator pairs, and their coupling strengths. Note that this arrangement makes the feature set highly structured but does not affect the characteristics of the Kuramoto model. To accurately label a given sample point (the feature set of a given Kuramoto model), we used the fourth-order Runge-Kutta method with a much longer simulation time T of 400 seconds to determine whether the model reaches global frequency synchronization or not. Besides, we rigorously determined the synchronization of the Kuramoto model based on more stringent criteria. For the labeling purpose, we consider that a Kuramoto model is synchronized if both of the following two conditions are satisfied: First, frequencies of all oscillators rounded to the sixth decimal place are equal for the last 20 $(T * 0.95)$ seconds. Second, the sum of absolute change in the coherence value $r(t)$ of the order parameter $r(t)e^{j\psi(t)} = \frac{1}{N}\sum_{i=1}^{N} e^{j\theta_i(t)}$ is less than 10^{-6} for the last 20 seconds. Note that if the results for the two conditions differ, we excluded the sample point from the training dataset. Since the detailed structure and the training process of the classifier varies depending on the simulation scenario, we provide further details in the corresponding subsections. The source code used for the simulations whose results are presented in this study can be found at https://github.com/bjyoontamu/Kuramoto-Model-OED-acceleration.

B. Performance Evaluation

To assess the performance of the proposed approach and compare it to other existing approaches, we performed a wide range of evaluation experiments. First, we evaluated the efficacy of the shallow fcNN model, which is adopted in this study, in predicting the global synchronization of the Kuramoto model oscillators. For this purpose, we examined the asymptotic behavior of the fcNN model by estimating its prediction accuracy as a function of increasing training data size. Next, we computed the Pearson's correlation coefficient between the MOCU values computed by the proposed scheme and the original sampling-based scheme [7], respectively. Furthermore, the time complexity of each scheme was assessed to compare efficiency. Finally, we evaluated the efficacy of the proposed scheme in predicting experiments that can effectively reduce model uncertainty. To this aim, we compared the changes in the "objective uncertainty" (estimated by MOCU) after conducting the experiment selected by different experimental design schemes. We also examined how accurately each scheme is able to predict the true optimal experiment, and how it affects the overall experimental performance.

C. Comparison Between Machine Learning-Based and Sampling-Based Mean Objective Cost of Uncertainty Estimation

In order to validate the efficacy of the proposed method that incorporates ML-based predictions into MOCU estimation, we directly compared the MOCU values from the ML-based approach and the sampling-based approach that we considered in the previous work [7].

1) Mean Objective Cost of Uncertainty Estimation for Uncertain Kuramoto Model With Five Oscillators: As a first experimental scenario, we considered an uncertain Kuramoto model that consists of five oscillators that do not get spontaneously synchronized in the absence of external control. In this experiment,

we adopted the identical experimental setup in the previous work [7] for direct comparison. Specifically, we assumed that the five oscillators have the natural frequencies of -2.50, -0.6667, 1.1667, 2.0, and 5.8333, respectively. The natural frequency of the additional (*i.e.*, 6th) control oscillator was set to the average frequency of the five oscillators ($\omega_6 = 1.1667$). Besides, we set the initial phase of all the oscillators to zero. Finally, we used the uncertainty class defined as follows:

$$\mathbf{a}^U = \begin{bmatrix} 1.0541 & 0.6325 & 0.7762 & 1.4375 & 1.0542 \\ 0.6900 & 1.6819 & 0.4791 & 2.6833 & 2.2041 \end{bmatrix}^T, \quad (12)$$

$$\mathbf{a}^L = \begin{bmatrix} 0.7791 & 0.4675 & 0.5737 & 1.0625 & 0.7792 \\ 0.5100 & 1.2431 & 0.3541 & 1.9833 & 1.6291 \end{bmatrix}^T. \quad (13)$$

To train the classifier, we generated 40,000 sample points (a set of 20,000 parameter values that result in synchronization and another set of 20,000 parameter values that do not) from a multivariate uniform distribution whose support completely covers the range of the parameters in the uncertainty class at hand. Specifically, a parameter set has six real-values from the uniform distribution with a range of $(-2\pi, 2\pi)$ as natural frequencies of the six oscillators ω_i, $i = 1, 2, \ldots, 6$, and ten coupling strength values $a_{i,j}$, $1 \leq i < j \leq 6$, between oscillators ranging from $0.25|\omega_i - \omega_j|$ to $2.35|\omega_i - \omega_j|$. To build the classifier, we sorted the six natural frequencies in descending order and rearranged the coupling strength values accordingly. Then, we extracted the following features: the sorted natural frequencies, the absolute difference of the natural frequencies of all oscillator pairs, and their coupling strengths. Finally, we trained an fcNN model with a single hidden layer, whose width is three times the number of features, until the model is capable of classifying all the 40,000 sample points in the training dataset perfectly. We validated the trained model in terms of its asymptotic classification accuracy by assessing the accuracy as a function of the training data size. This result is shown in Fig. S1 in the supplemental material.

We started with the original uncertainty class defined in (12) and (13) and estimated the expected remaining MOCU of random oscillator pairs through both approaches one hundred times while randomly changing the true model (assumed to be unknown). Figure 4 is a scatter plot that shows the comparison between the expected remaining MOCU values computed by different methods. As shown in Fig. 4, the expected remaining MOCU values computed by the proposed ML-based method and the original sampling-based method display a strong linear relationship. The Pearson's correlation coefficient was 0.9849 with a p-value of 1.90×10^{-76}. This plot shows that the ML-based computational scheme has the potential to effectively replace the costly sampling-based scheme without affecting the MOCU-based OED performance, as it will likely not affect the ranking of potential experiments. In terms of computational cost, the ML-based approach was able to compute the expected remaining MOCU in 0.1110 seconds (on average) for a given uncertainty class, while it took 818.7 seconds (on average) for the sampling-based approach. These results clearly show the advantages of the proposed approach in efficiently quantifying the objective uncertainty.

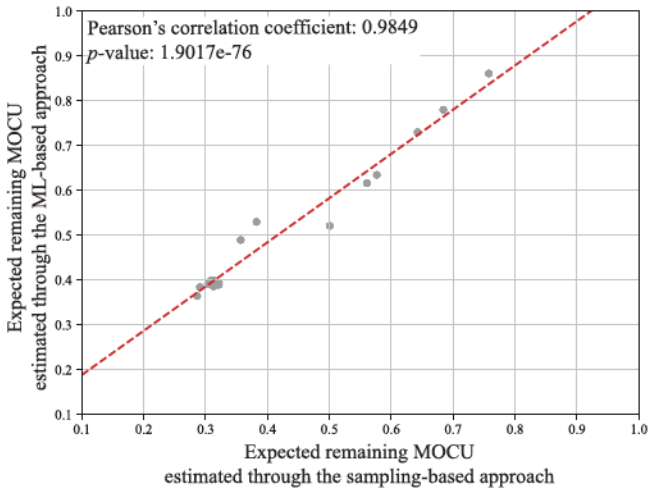

Fig. 4. The scatter plot shows the expected remaining MOCU values for the uncertain five-oscillator Kuramoto model estimated using the proposed ML-based approach and the original sampling-based approach in [7]. As shown, the estimated values are highly correlated to each other.

2) Mean Objective Cost of Uncertainty Estimation for Uncertain Kuramoto Model With Seven Oscillators: Some major disadvantages of the original sampling-based method are its high computational cost and the practical difficulty of scaling the method for large models with many oscillators. To examine the computational cost increase and scalability for larger models, we next considered an uncertainty class of Kuramoto models with seven oscillators. This increases the time for solving the ODEs and the number of possible experiments also increases from $\binom{5}{2} = 10$ to $\binom{7}{2} = 21$. Here we set the natural frequency of the oscillators to -3.4600, -1.9611, -0.6754, -0.3806, -0.3675, 6.1161, and 8.3287, respectively. We assumed that the natural frequency of the control oscillator (*i.e.*, 8th oscillator) is the average frequency of the seven oscillators ($\omega_8 = 1.0857$). We considered the uncertainty class shown below:

$$\mathbf{a}^U = \begin{bmatrix} 0.848 & 0.988 & 1.446 & 1.607 & 3.820 & 0.915 & 0.400 \\ 0.850 & 0.419 & 4.162 & 1.090 & 0.122 & 0.039 & 2.124 \\ 0.872 & 0.007 & 2.737 & 1.804 & 1.360 & 0.744 & 1.174 \end{bmatrix}^T, \quad (14)$$

$$\mathbf{a}^L = \begin{bmatrix} 0.073 & 0.172 & 0.153 & 0.054 & 0.501 & 0.463 & 0.043 \\ 0.015 & 0.096 & 0.501 & 0.103 & 0.007 & 0.009 & 0.139 \\ 0.408 & 0.000 & 0.131 & 0.119 & 0.300 & 0.286 & 0.131 \end{bmatrix}^T. \quad (15)$$

As in the previous experiment for the Kuramoto model with five oscillators, we set the initial phase of all oscillators to zero.

As the size of the parameter set is much greater for this Kuramoto model, we generated the training data in a more tailored way. Rather than generating the sample points (*i.e.*, Kuramoto model parameter sets) with random natural frequencies within a specific range as we did for the five oscillator model, we fixed the natural frequencies to -3.4600, -1.9611, -0.6754, -0.3806, -0.3675, 6.1161, 8.3287, and 1.0857 in this example. For the coupling strength values, we drew them from the uniform distribution for the uncertainty class, whose support is defined in (14) and (15). In this manner, we collected

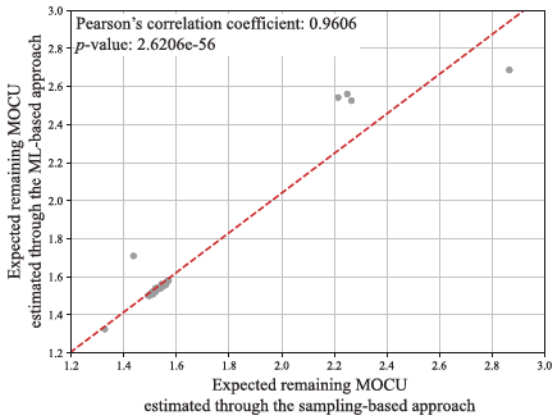

Fig. 5. The scatter plot shows the expected remaining MOCU values for the uncertain seven-oscillator Kuramoto model estimated using the proposed ML-based approach and the original sampling-based approach in [7]. As before, the estimated values show a high correlation.

50,000 sample points per label according to the same criteria we used for the five oscillator case. Then, we extracted the feature values as described previously for the five oscillator case and trained the classifier using an fcNN with a single hidden layer, whose width is four times the number of features. Figure S1 in the supplemental material shows that this model quickly learns the classification boundary, where the classification accuracy rapidly converges to 100% as the size of the training data increases.

As before, we started with the original uncertainty class defined in (14) and (15) and computed the expected remaining MOCU of random oscillator pairs using the ML-based method and the sampling-based method. We repeated this until we collected a hundred expected remaining MOCU values per method. Figure 5 shows the scatter plot that compares the expected remaining MOCU values computed by the two methods. Again, we can see that there is a strong linear relationship between the computed values. The Person's correlation coefficient was 0.9606 with a p-value of 2.62×10^{-56}. In terms of computational cost, it took 0.6953 seconds (on average) for the ML-based method to compute the expected remaining MOCU, which was still less than a second although the experimental design space has grown from $\frac{(5\times4)}{2} = 10$ experiments to $\frac{(7\times6)}{2} = 21$. It took the sampling-based approach $3,684.9$ seconds (on average) to compute the expected remaining MOCU values, which shows that our proposed method makes the computation 5,298 times faster at practically identical accuracy. These results clearly show the advantages of the proposed ML-based approach in quantifying the objective model uncertainty.

V. Performance of Experimental Design Using Machine Learning-Based Mean Objective Cost of Uncertainty Estimation

We compared the OED performance of the proposed ML-based method against three existing approaches:

- *Sampling-based approach*: the original approach proposed in [7] based on the MOCU framework, where a fourth-order Runge-Kutta method is to solve the Kuramoto model to determine synchronization.
- *Entropy-based approach*: the experiment is chosen for the oscillator pair whose coupling strength value has the largest entropy to reduce this uncertainty.
- *Random approach*: the experiment is randomly selected from the experimental design space.

As shown, the proposed ML-based estimation clearly outperforms the original sampling-based estimation [7] in terms of efficiency, where their costs differ by two orders of magnitude. For both ML-based/sampling-based schemes, iterative estimation requires further computations, as the uncertainty class is updated after each experiment, based on which the remaining expected MOCU values are assessed again.

For the MOCU-based OED schemes (*i.e.*, ML-based and sampling-based computations), we considered the following OED strategies. In the first approach (marked as *iterative* in the figures), we re-estimated the expected remaining MOCU for the remaining experiments in each iteration, after performing the predicted optimal experiment and updating the uncertainty class based on the observed experimental outcome. In the second approach, we estimated the expected remaining MOCU only based on the initial uncertainty class and prioritized all experiments based on this result. While this approach is theoretically suboptimal, it significantly reduced the overall computational cost and empirically showed comparable performance to the iterative scheme, as we will show in this section.

1) Optimal Experimental Design for Uncertain Kuramoto Models With Five Oscillators: First, we conducted OED simulations for the same five-oscillator Kuramoto model considered in the previous study [7] for direct comparison. We used identical model parameters described in Section IV-C1. The true (unknown) model a was assumed to be as follows:

$$\mathbf{a} = \begin{bmatrix} 0.9166 & 0.55 & 0.675 & 1.25 & 0.9167 \\ 0.6 & 1.4625 & 0.4166 & 2.3333 & 1.9166 \end{bmatrix}^T. \tag{16}$$

Figure 6 shows the experimental design performance of the different algorithms, where the objective uncertainty (quantified by MOCU) is shown as a function of the number of experimental updates (iterations). As shown in Fig. 6, the proposed ML-based approach with iterative re-estimation (red dotted line with asterisks) showed the nearly identical performance to sampling-based methods (both iterative and non-iterative schemes, shown in yellow lines). All three schemes reached the near minimum MOCU within only three experimental updates. The non-iterative ML-based scheme (red dashed line with squares) also identified the first optimal experiment accurately and showed comparable performance in the later updates with the other three MOCU-based OED schemes. All four MOCU-based OED schemes (both ML-based and sampling-based) significantly outperformed the entropy-based and random approaches, resulting in much sharper uncertainty reduction within fewer experimental updates.

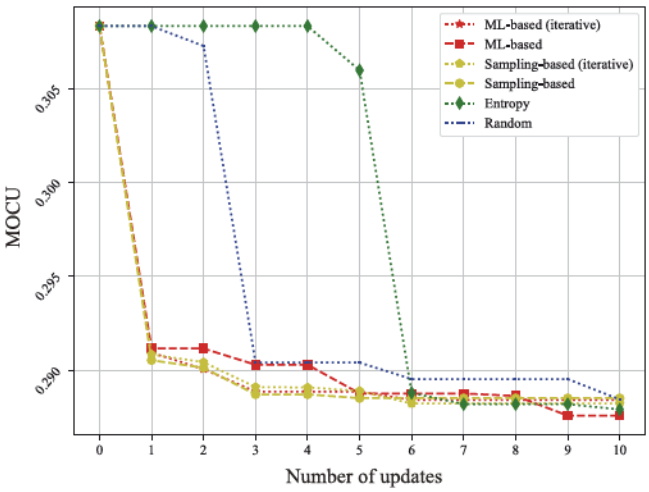

Fig. 6. Performance comparison of various experimental design strategies for the uncertain five-oscillator Kuramoto model considered in [7]. The results show that the three MOCU-based OED schemes perform similarly, regardless of how MOCU was estimated. The MOCU-based schemes clearly outperform other schemes as reported in [7].

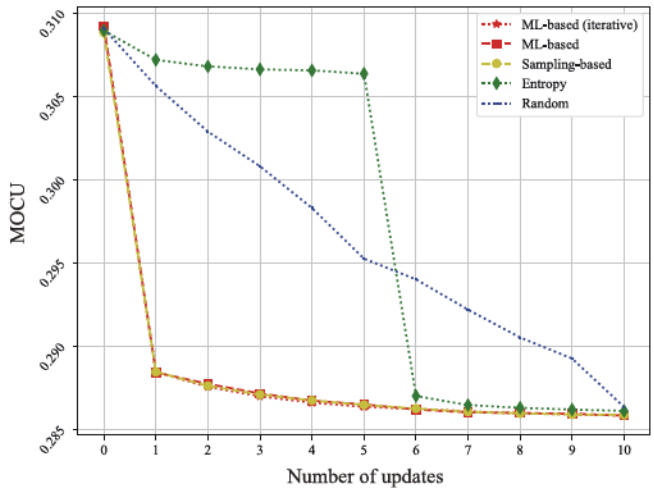

Fig. 8. Average performance of various experimental design strategies for uncertain Kuramoto models with five oscillators. The experiments have been repeated one hundred times by sampling potential true models from the uncertainty class. As shown, all three MOCU-based methods lead to the best performance. Random selection results in linear uncertainty reduction as expected.

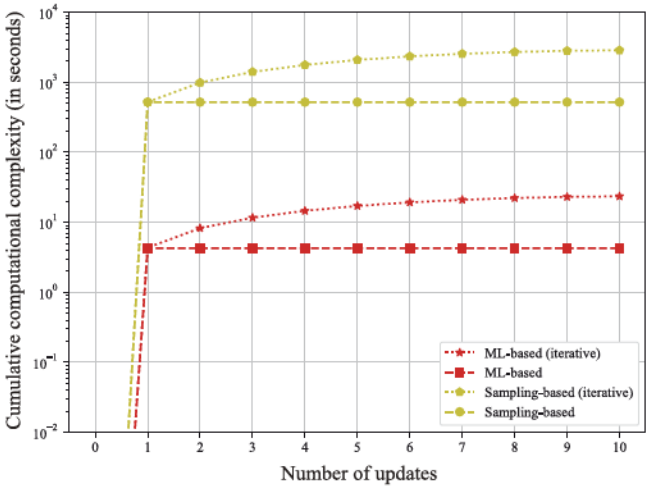

Fig. 7. Cumulative computational cost (in seconds) for identifying the optimal experiment.

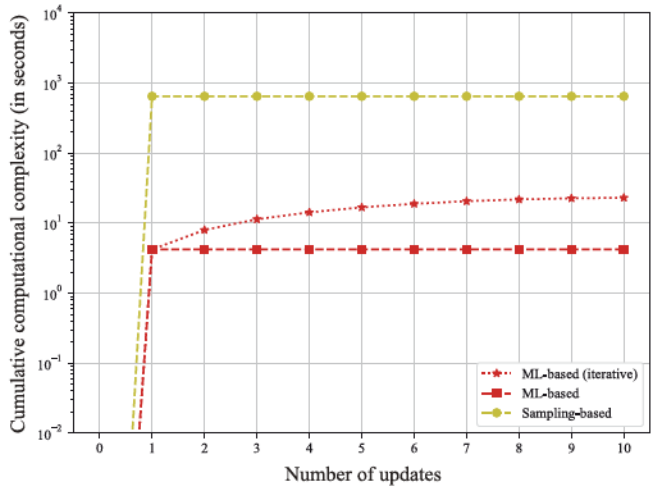

Fig. 9. Average cumulative computational cost (in seconds) for identifying the optimal experiment based on different experimental design strategies for uncertain Kuramoto models with five oscillators.

Figure 7 compares the overall computational cost between the ML-based OED schemes and the sampling-based OED schemes. The entropy-based approach and the random approach are not shown, as their computational cost is fixed and negligible. As we can see in Fig. 7, the proposed ML-based OED approaches, marked as red, showed significantly lower time complexity compared to the sampling-based OED approaches. Note that the ML-based methods (red dotted lines) were significantly faster compared to the sampling-based methods, despite maintaining equivalent OED performance. We did not include the time for training the fcNN model as the model training only needs to be performed once before the beginning of OED. In fact, the trained model can be reused for different uncertainty classes and *any* true model therein. Besides, the training time is negligible thanks to the shallow structure of the fcNN model considered in this study. Specifically, the model learned the training dataset for making a decision for the uncertain Kuramoto model with five oscillators within 90 seconds.

Next, we repeated the experiment based on one hundred different true models randomly drawn from the uncertainty class (*i.e.*, different coupling strength values were drawn from the prior distribution of the uncertainty class). The results of these large-scale experiments are shown in Fig. 8 and Fig. 9. Note that we excluded the iterative sampling-based OED method due to its excessive requirement of computational time. As shown in these figures, the proposed ML-based method without iterative re-estimation of the expected remaining MOCU showed identical performance to other best performers. Random experimental selection (blue dotted line) yielded a

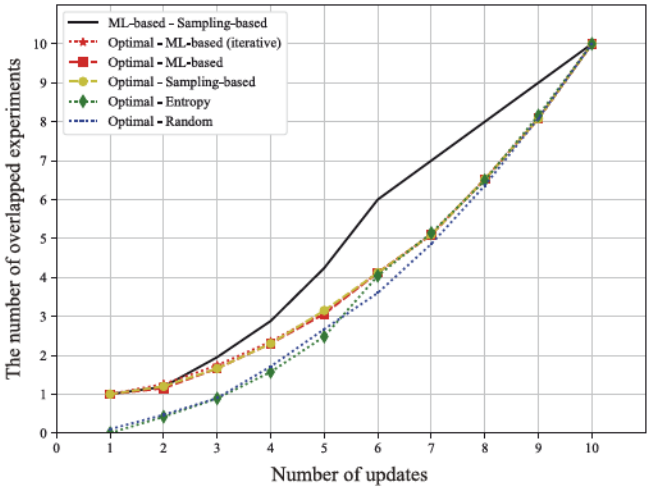

Fig. 10. Comparison between the optimal sequence of experiments predicted by different OED strategies for uncertain Kuramoto models with seven oscillators. The y-axis shows the number of common experiments within the first k experiments predicted by two different methods.

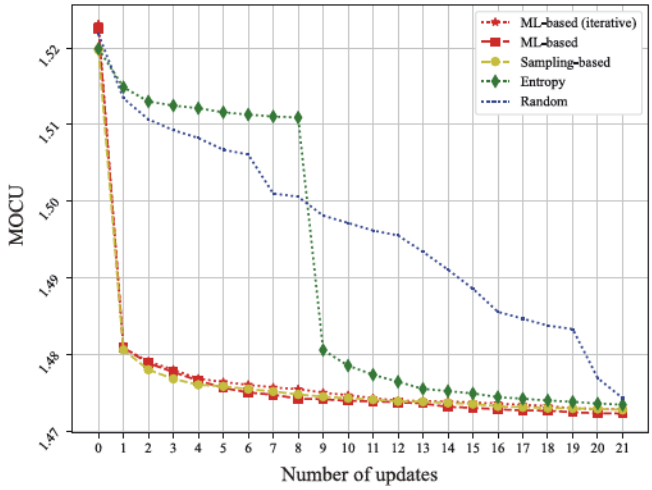

Fig. 11. Average performance of various experimental design strategies for uncertain Kuramoto models with seven oscillators based on one hundred experiments. All MOCU-based methods lead to the best performance, and random selection results in linear uncertainty reduction.

linearly decreasing MOCU curve, as we would expect on average. The entropy-based method showed similar performance as before (see Fig. 6). Computational cost in Fig. 9 shows a similar trend as before (see Fig. 7). As before, the time for training the ML model is not included in this plot. Furthermore, Fig. S2 in the supplemental material shows the RainCloud plot [42] that depicts the instantaneous performance of the different methods measured in terms of the remaining uncertainty (measured by MOCU) after performing the first experiment selected by the respective methods. As we can see from Fig. S2, all three MOCU-based OED schemes consistently yield the best overall performance.

Finally, we compared the experimental sequences determined by the ML-based method and the sampling-based methods, respectively, to further investigate if the proposed ML approach can practically replace the sampling-based method for prioritizing the experiments in the experimental design space. The vertical axis corresponds to the number of intersecting experiments in the first k experiments predicted by two different methods. If two methods predict the identical experimental sequence, the resulting curve will be a straight line (with unit slope). For example, the black line in Fig. 10 compares the ML-based method and the sampling-based method. From Fig. 10, we can see that the proposed ML-based method (without re-estimation) always identified the same first experiment as the sampling-based method in all one hundred evaluations. By comparing the true optimal experimental sequence (*i.e.*, predicted by an "oracle") and the sequences predicted by the ML-based method, we can see that the first optimal experiment was always accurately predicted. In fact, results in Fig. 8 show that the first experiment leads to the most significant drop in model uncertainty, and all MOCU-based OED schemes (both ML-based and sampling-based) accurately predict this critical experiment. We also note that the entropy-based/random approaches tended to mispredict the best first experiment, resulting in a substantial performance gap when compared to the MOCU-based approaches. Figure 10 also shows that the predicted experimental sequences diverge in later iterations. However, this did not impact the OED performance on average, as later experiments did not reduce the model uncertainty as significantly as the earlier experiments.

2) Optimal Experimental Design for Uncertain Kuramoto Models With Seven Oscillators: We also repeated the experiments for uncertain Kuramoto models that consist of seven oscillators. As before, true (unknown) models were randomly sampled from the uncertainty class one hundred times to evaluate average performance. We used the same parameters and model described in Section IV-C2.

Figure 11 shows the OED performance assessment results for the various experimental design methods based on the seven-oscillator Kuramoto model. As we can see from Fig. 11, the performance trends were very similar to those seen in Fig. 8 for the Kuramoto model with five oscillators. The proposed ML-based methods again accurately identified the first optimal experiment that maximally reduces MOCU on average. All four MOCU-based OED schemes (both ML-based and sampling-based), regardless of whether or not the remaining expected MOCU values were re-estimated after each experimental update, showed almost identical performance on average. Figure S3 compares the performance of different methods, where we measured the MOCU that remains after performing the first experiment selected by each method. The results are again shown for one hundred evaluations based on different true models. As shown in Fig. S3, the efficacy of the first experiment varies depending on the underlying true model, which is expected. As before, the results in Fig. S3 clearly show that the proposed ML-based OED scheme can effectively replicate the performance of the original sampling-based approach [7], the primary

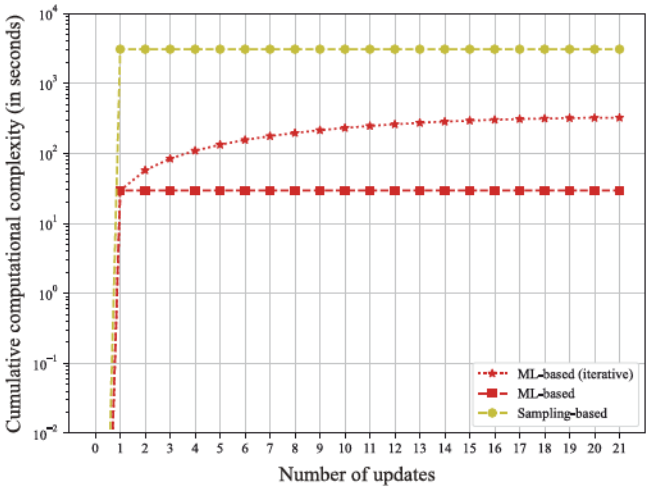

Fig. 12. Average cumulative computational cost (in seconds) for identifying the optimal experiment based on different experimental design strategies for uncertain Kuramoto models with seven oscillators.

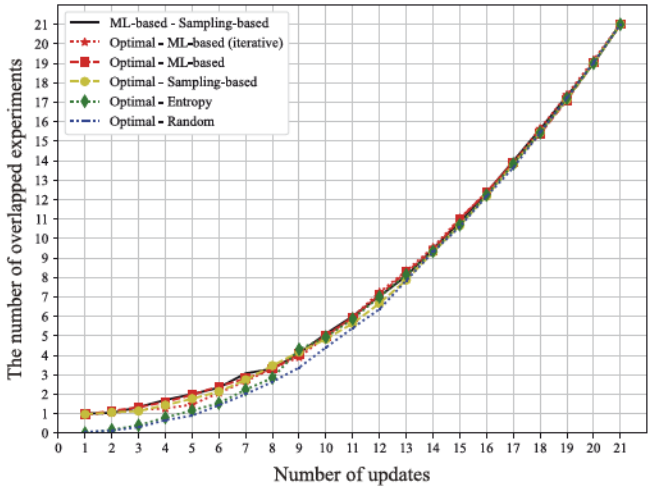

Fig. 13. Comparison between the optimal sequence of experiments predicted by different OED strategies for uncertain Kuramoto model with seven oscillators. The y-axis shows the number of common experiments within the first k experiments predicted by two different methods. The results are shown for uncertain Kuramoto models with seven oscillators.

goal of this study. The computational time is shown in Fig. 12, which clearly shows that the ML-based scheme (especially, the non-iterative scheme) is significantly faster compared to the original sampling-based approach. As before, the plot only shows the time for OED and does not include the training time for the ML model. Furthermore, even the ML-based method with the iterative update was considerably faster than the sampling-based that does not iteratively re-estimate the expected remaining MOCU.

Finally, we compared the experimental sequences identified by the different methods, including the true optimal experimental selection (*i.e.*, predicted by an "oracle"). Note that due to the excessive computational cost of the optimal experimental selection (as it requires exhaustive search), we identified the optimal experimental sequences only for the first thirty evaluations based on randomly sampled true models from the uncertainty class. For this reason, Fig. 13 shows the comparison results based on the first thirty experimental sequences (out of one hundred). As shown in the figure, both the ML-based and the sampling-based methods were able to accurately identify the first optimal experiment. The predicted sequences tended to diverge in later iterations. However, considering the simulation results shown in Fig. 11, it is likely that this was because many experiments in later updates did not significantly reduce the objective uncertainty, once the best experiment has been performed in the earlier iterations (especially, the first iteration). Also, we can see that the entropy-based and the random selection approaches tended to miss the best experiment, which resulted in a significant degradation in the overall experimental design performance. These comprehensive simulation results clearly showed that our proposed ML-based OED approach effectively quantifies the objective model uncertainty at a small fraction of the computational cost of the sampling-based method, thereby remarkably accelerating the OED process while maintaining excellent performance.

VI. Conclusion

In this paper, we proposed an ML approach that can significantly accelerate the objective-based quantification of model uncertainty via MOCU. A major bottleneck in applying MOCU for designing/prioritizing optimal experiments that can optimally reduce the uncertainty in models that represent real-world complex uncertain systems has been the high computational cost for accurately estimating MOCU. The proposed approach effectively addresses this issue in the context of OED for uncertain Kuramoto models by replacing the computational costly DE solver with an ML model, which remarkably speeds up the process of predicting the optimal controller (*i.e.*, the oscillator that guarantees global frequency synchronization at minimum cost). The trained ML model predicts the asymptotic behavior of a given Kuramoto model, namely, whether all oscillators in the model will be eventually frequency synchronized or not.

Note that there may be an outward similarity between the proposed ML-based approach in the context of the uncertain Kuramoto model with other earlier studies [39]–[41] whose main interest is to efficiently solve DEs via ML models, as the prediction result of our trained ML model includes the process of solving Kuramoto model equations, albeit implicitly. However, more precisely, the trained ML model in the current study makes a "decision" based on the scientific knowledge regarding the underlying Kuramoto model for the purpose of identifying the optimal experiment that is expected to be most effective in reducing model uncertainty, which pertains to a broader issue rather than solving DEs as in other previous studies.

The results shown in Section IV and Section V clearly demonstrate that the ML-based MOCU calculations are highly correlated with those computed by the sampling-based scheme originally proposed in [7]. Furthermore, the OED performance

of the ML-based scheme is practically equivalent to that of the original sampling-based OED scheme. However, despite achieving equivalent OED performance, our proposed ML-based OED scheme accelerates the experimental design process by at least two orders of magnitude, resulting in significant computational gains. The remarkably enhanced computational efficiency enables more reliable MOCU calculation by further increasing the sample size (*i.e.*, K) as needed. Furthermore, it allows us to iteratively recompute the remaining MOCU $R(i, j)$ after performing the predicted optimal experiment at each experimental update (see Fig. 9 and Fig. 12), which can–in theory–lead to a more accurate prediction of the optimal experiment, although the actual gain will depend on the underlying model uncertainty. Such iterative update is practically infeasible for the original sampling-based OED scheme without resorting to HPC (high-performance computing).

As shown in Section IV and Section V, our ML-based MOCU estimation and OED approach remarkably enhance the computational efficiency by refraining from repeatedly solving the DEs for the uncertain Kuramoto models for the sake of finding the optimal robust operator (which is required in the original sampling-based approach) but instead adopting ML for decision making. However, as training the ML model requires the generation of sufficient training data, which also requires solving the coupled ODEs for different Kuramoto models in the uncertainty class, it will be interesting to compare the proposed ML-based approach with the sampling-based approach from the perspective of "data efficiency". For this purpose, we quantitatively compare the proposed approach with the sampling-based approach in terms of data requirements. For the uncertain Kuramoto model with five oscillators, we trained the ML model (an fcNN with a single hidden layer) with 40,000 labeled sample points. Each sample point corresponds to the Kuramoto model with a different parameter, and labeling the sample point (*i.e.*, synchronized vs non-synchronized) requires solving the corresponding ODEs. The trained model is used throughout the entire experimental design process without the need for generating additional sample points. On the other hand, the sampling-based method requires generating approximately 2.2×10^7 labeled sample points (*i.e.*, by solving the DEs for different Kuramoto model parameters). Similarly, for the uncertain Kuramoto model with seven oscillators, we trained an fcNN model based on 100,000 labeled sample points, and the trained model is used throughout the experimental design process. In comparison, the sampling-based approach requires the generation of around 9.4×10^7 labeled sample points. These comparisons clearly show that our proposed ML-based OED acceleration scheme not only improves the computational efficiency but also drastically improves the data efficiency.

It is worth noting that the proposed approach, applying ML models to the estimation of the (remaining expected) MOCU can be generalized, extended, and applied to other MOCU-based OED problems concerning real-world applications that do not possess closed-form (remaining expected) MOCU. For example, in the case of OED for robust intervention in gene regulatory networks (GRN) [9], [10], an ML model may be used to predict the efficacy of an intervention scheme in reducing the steady-state mass of the GRN in undesirable (e.g., pathological) states. This will obviate the need for estimating the steady-state probabilities of the underlying Markov chain, thereby significantly accelerate the estimation of MOCU and enhance the scalability of the MOCU-based OED scheme.

In such cases, based on the applications and the data types, one may consider different ML models including convolutional neural networks (CNN) [43], recurrent neural networks (RNN) [44], long short term memory networks (LSTM) [45], and graph convolutional networks (GCN) [46]. While we have focused on accelerating the experimental design for uncertain Kuramoto models using ML, the proposed ML-based OED acceleration scheme is fairly general and its applicability goes beyond the Kuramoto models. For example, the proposed scheme may be used to accelerate the design of effective experiments to reduce uncertainty and improve the control performance of various other engineering models in the presence of uncertainty [47]–[50].

An interesting direction for future research is to utilize ML models to learn scientific knowledge from data. In this paper, we considered pairwise synchronization experiments, whose result can be used to reduce model uncertainty using Theorem 1 that gives us the necessary and sufficient condition for frequency synchronization of an oscillator pair. In the absence of such knowledge, one cannot design experiments for effective uncertainty reduction. As mentioned before, similar theorems do not exist in general for non-homogeneous Kuramoto models that consist of more than two oscillators. Discovering useful relational knowledge regarding the model parameters via ML can lead to the design of more effective experiments as well as a significant expansion of the potential experimental design space. We are currently investigating the potential utilization of deep neural network (DNN) models for knowledge discovery in non-homogeneous Kuramoto models with multiple oscillators.

REFERENCES

[1] P. Niu *et al.*, "TRIMER: Transcription regulation integrated with metabolic regulation," *iScience*, vol. 24, no. 11, Nov. 2021, Art. no. 103218.
[2] F. Shen *et al.*, "OptRAM: In-silico strain design via integrative regulatory-metabolic network modeling," *PLoS Comput. Biol.*, vol. 15, no. 3, 2019, Art. no. e1006835.
[3] E. R. Dougherty, "Scientific epistemology in the context of uncertainty," in *Berechenbarkeit der Welt*? Wiesbaden, Germany: Springer, 2017, pp. 129–154.
[4] E. R. Dougherty, L. A. Dalton, and R. Dehghannasiri, *Objective Uncertainty Quantification*. Cham, Switzerland: Springer International Publishing, 2019.
[5] B.-J. Yoon, X. Qian, and E. R. Dougherty, "Quantifying the objective cost of uncertainty in complex dynamical systems," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2256–2266, May 2013.
[6] B.-J. Yoon, X. Qian, and E. R. Dougherty, "Quantifying the multi-objective cost of uncertainty," *IEEE Access*, vol. 9, pp. 80351–80359, 2021.
[7] Y. Hong, B. Kwon, and B.-J. Yoon, "Optimal experimental design for uncertain systems based on coupled differential equations," *IEEE Access*, vol. 9, pp. 53804–53810, 2021.
[8] G. Zhao, X. Qian, B.-J. Yoon, F. J. Alexander, and E. R. Dougherty, "Model-based robust filtering and experimental design for stochastic differential equation systems," *IEEE Trans. Signal Process.*, vol. 68, pp. 3849–3859, Jun. 2020.
[9] R. Dehghannasiri, B.-J. Yoon, and E. R. Dougherty, "Efficient experimental design for uncertainty reduction in gene regulatory networks," *BMC Bioinf.*, vol. 16, no. 13, pp. 1–18, 2015.

[10] R. Dehghannasiri, B.-J. Yoon, and E. R. Dougherty, "Optimal experimental design for gene regulatory networks in the presence of uncertainty," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 4, pp. 938–950, Jul./Aug. 2015.
[11] A. Broumand, M. S. Esfahani, B.-J. Yoon, and E. R. Dougherty, "Discrete optimal Bayesian classification with error-conditioned sequential sampling," *Pattern Recognit.*, vol. 48, no. 11, pp. 3766–3782, 2015.
[12] G. Zhao, E. Dougherty, B.-J. Yoon, F. Alexander, and X. Qian, "Bayesian active learning by soft mean objective cost of uncertainty," in *Proc. 24th Int. Conf. Artif. Intell. Statist.*, 2021, pp. 3970–3978.
[13] G. Zhao, E. Dougherty, B.-J. Yoon, F. Alexander, and X. Qian, "Uncertainty-aware active learning for optimal Bayesian classifier," in *Proc. 9th Int. Conf. Learn. Representations*, 2021.
[14] G. Zhao, E. Dougherty, B.-J. Yoon, F. J. Alexander, and X. Qian, "Efficient active learning for Gaussian process classification by error reduction," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021.
[15] Y. Kuramoto, "Self-entrainment of a population of coupled non-linear oscillators," in *International Symposium on Mathematical Problems in Theoretical Physics*. Berlin, Heidelberg, Germany: Springer, 1975, pp. 420–422.
[16] K. Wiesenfeld, P. Colet, and S. Strogatz, "Frequency locking in Josephson arrays: Connection with the Kuramoto model," *Phys. Rev. E*, vol. 57, no. 2, pp. 1563–1569, Feb. 1998.
[17] Z. Néda, E. Ravasz, T. Vicsek, Y. Brechet, and A. L. Barabási, "Physics of the rhythmic applause," *Phys. Rev. E*, vol. 61, no. 6, pp. 6987–6992, Jun. 2000.
[18] C. Hammond, H. Bergman, and P. Brown, "Pathological synchronization in Parkinson's disease: Networks, models and treatments," *Trends Neurosciences*, vol. 30, no. 7, pp. 357–364, 2007.
[19] M. G. Kitzbichler, M. L. Smith, S. R. Christensen, and E. Bullmore, "Broadband criticality of human brain network synchronization," *PLoS Comput. Biol.*, vol. 5, no. 3, 2009, Art. no. e1000314.
[20] M. Breakspear, S. Heitmann, and A. Daffertshofer, "Generative models of cortical oscillations: Neurobiological implications of the kuramoto model," *Front. Hum. Neurosci.*, vol. 4, 2010, Art. no. 190.
[21] D. Bhowmik and M. Shanahan, "How well do oscillator models capture the behaviour of biological neurons?," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2012, pp. 1–8.
[22] J. W. Simpson-Porco, F. Dörfler, and F. Bullo, "Droop-controlled inverters are kuramoto oscillators," *IFAC Proc. Volumes*, vol. 45, no. 26, pp. 264–269, 2012.
[23] B. Fernandez, D. Gérard-Varet, and G. Giacomin, "Landau damping in the Kuramoto model," *Annales Henri Poincaré*, vol. 17, no. 7, pp. 1793–1823, Dec. 2015.
[24] P. S. Skardal and A. Arenas, "Control of coupled oscillator networks with application to microgrid technologies," *Sci. Adv.*, vol. 1, no. 7, 2015, Art. no. e1500339.
[25] A. Mohseni, S. Gharibzadeh, and F. Bakouie, "The role of driver nodes in managing epileptic seizures: Application of Kuramoto model," *J. Theor. Biol.*, vol. 419, pp. 108–115, 2017.
[26] H. Choi and S. Mihalas, "Synchronization dependent on spatial structures of a mesoscopic whole-brain network," *PLoS Comput. Biol.*, vol. 15, no. 4, 2019, Art. no. e1006978.
[27] Y. Guo, D. Zhang, Z. Li, Q. Wang, and D. Yu, "Overviews on the applications of the Kuramoto model in modern power system analysis," *Int. J. Elect. Power Energy Syst.*, vol. 129, 2021, Art. no. 106804.
[28] M. Rohden, A. Sorge, M. Timme, and D. Witthaut, "Self-organized synchronization in decentralized power grids," *Phys. Rev. Lett.*, vol. 109, Aug. 2012, Art. no. 064101.
[29] A. Motter, S. Myers, M. Anghel, and T. Nishikawa, "Spontaneous synchrony in power-grid networks," *Nature Phys.*, vol. 9, no. 3, pp. 191–197, Mar. 2013.
[30] J. W. Simpson-Porco, F. Dorfler, and F. Bullo, "Synchronization and power sharing for droop-controlled inverters in islanded microgrids," *Automatica*, vol. 49, no. 9, pp. 2603–2611, 2013.
[31] F. Dörfler, M. Chertkov, and F. Bullo, "Synchronization in complex oscillator networks and smart grids," *Proc. Nat. Acad. Sci.*, vol. 110, no. 6, pp. 2005–2010, 2013.
[32] K. Lehnertz *et al.*, "Synchronization phenomena in human epileptic brain networks," *J. Neurosci. Methods*, vol. 183, no. 1, pp. 42–48, 2009.
[33] J. C. Pang, L. L. Gollo, and J. A. Roberts, "Stochastic synchronization of dynamics on the human connectome," *NeuroImage*, vol. 229, 2021, Art. no. 117738.
[34] A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, and C. Zhouu, "Synchronization in complex networks," *Phys. Rep.*, vol. 469, no. 3, pp. 93–153, 2008.
[35] F. A. Rodrigues, T. K. D. Peron, P. Ji, and J. Kurths, "The Kuramoto model in complex networks," *Phys. Rep.*, vol. 610, pp. 1–98, 2016.
[36] J. A. Acebrón, L. L. Bonilla, C. J. Pérez Vicente, F. Ritort, and R. Spigler, "The Kuramoto model: A simple paradigm for synchronization phenomena," *Rev. Mod. Phys.*, vol. 77, pp. 137–185, Apr. 2005.
[37] S. Boluki, M. S. Esfahani, X. Qian, and E. R. Dougherty, "Incorporating biological prior knowledge for Bayesian learning via maximal knowledge-driven information priors," *BMC Bioinf.*, vol. 18, no. 14, pp. 61–80, 2017.
[38] S. Boluki, M. S. Esfahani, X. Qian, and E. R. Dougherty, "Constructing pathway-based priors within a Gaussian mixture model for Bayesian regression and classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 2, pp. 524–537, Mar./Apr. 2019.
[39] I. Lagaris, A. Likas, and D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 987–1000, Sep. 1998.
[40] J. Han, A. Jentzen, and W. E., "Solving high-dimensional partial differential equations using deep learning," *Proc. Nat. Acad. Sci.*, vol. 115, no. 34, pp. 8505–8510, 2018.
[41] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.
[42] M. Allen, D. Poggiali, K. Whitaker, T. R. Marshall, and R. A. Kievit, "Raincloud plots: A multi-platform tool for robust data visualization," *Wellcome Open Res.*, vol. 4, 2019, Art. no. 63.
[43] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition" in *Competition and Cooperation in Neural Nets*. Berlin, Heidelberg, Germany: Springer, 1982, pp. 267–285.
[44] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[46] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[47] H. Tao, J. Li, Y. Chen, V. Stojanovic, and H. Yang, "Robust point-to-point iterative learning control with trial-varying initial conditions," *IET Control Theory Appl.*, vol. 14, no. 19, pp. 3344–3350, 2020.
[48] H. Tao, X. Li, W. Paszke, V. Stojanovic, and H. Yang, "Robust PD-type iterative learning control for discrete systems with multiple time-delays subjected to polytopic uncertainty and restricted frequency-domain," *Multidimensional Syst. Signal Process.*, vol. 32, no. 2, pp. 671–692, 2021.
[49] X. Zhang *et al.*, "Asynchronous fault detection for interval type-2 fuzzy nonhomogeneous higher-level Markov jump systems with uncertain transition probabilities," *IEEE Trans. Fuzzy Syst.*, to be published, 10.1109/TFUZZ.2021.3086224.
[50] X. Xin *et al.*, "Online reinforcement learning multiplayer non-zero sum games of continuous-time Markov jump linear systems," *Appl. Math. Comput.*, vol. 412, 2022, Art. no. 126537.

Hyun-Myung Woo was born in Ulsan, South Korea. He received the B.S. degree in information & communication engineering and the M.S. degree in computer science in 2008 and 2010, respectively, from Yonsei University. He worked with IDIS and LG Electronics CTO as a Researcher for four years. He is currently working toward the Ph.D. degree with Texas A&M University. His main research interests include optimal experimental design, objective-based uncertainty quantification, and scientific machine learning for a wide range of applications, including bioinformatics and computational biology.

Youngjoon Hong received the Ph.D. degree in mathematics from Indiana University, Bloomington, IN, USA, in 2015. From 2015 to 2018, he was a Research Assistant Professor with University of Illinois, Chicago, IL, USA. From 2018 to 2021, before he joined the Sungkyunkwan University, he was an Assistant Professor with the Department of Mathematics and Statistics, San Diego State University . His research interest include the area of applied analysis and computational mathematics. Much of his works are motivated by connections between practical applications such as computational fluid dynamics, geophysics, electromagnetism, nonlinear waves, data science, and deep learning.

Bongsuk Kwon was born in Busan, South Korea. He received the B.S. degree from Korea University, Seoul, South Korea, the M.S. degree from the Pohang University of Science and Technology, Pohang, South Korea, and the Ph.D. degree from Indiana University, Bloomington, IN, USA, in 2009, all in mathematics. From 2009 to 2012, he was a Visiting Assistant Professor with the Department of Mathematics, Texas A&M University, College Station. In 2012, he joined Ulsan National Institute of Science and Technology, Ulsan, South Korea, where he is currently an Associate Professor of Mathematics. Dr. Kwon's main research interests include the theory of partial differential equations arising from continuum mechanics, which include stability and the asymptotic behavior of nonlinear waves.

Byung-Jun Yoon (Senior Member, IEEE) received the B.S.E. (*summa cum laude*) degree from Seoul National University, Seoul, South Korea, in 1998, and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, CA, USA, in 2002 and 2007, respectively, all in electrical engineering. He joined the Department of Electrical and Computer Engineering, Texas A&M University, College Station, in 2008, where he is currently an Associate Professor. Dr. Yoon also holds a joint appointment at Brookhaven National Laboratory, Upton, NY, where he is a Scientist in Computational Science Initiative, Applied Mathematics group. His honors include the National Science Foundation CAREER Award, the Best Paper Award at the 9th Asia Pacific Bioinformatics Conference, the Best Paper Award at the 12th Annual MCBIOS Conference, and the SLATE Teaching Excellence Award from the Texas A&M University System. His main research interests include optimal experimental design, objective-based uncertainty quantification, and scientific machine learning for a wide range of applications, including bioinformatics, computational network biology, materials science, and drug discovery.