Computational Materials Science 188 (2021) 110178

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

Optimal allocation of computational resources based on Gaussian process: Application to molecular dynamics simulations

John Chilleri [a], Yanyan He [b,*], Dmitry Bedrov [c], Robert M. Kirby [d]

[a] Department of Mathematics, New Mexico Tech, United States
[b] Departments of Mathematics, and of Computer Science and Engineering, University of North Texas, United States
[c] Department of Materials Science and Engineering, University of Utah, United States
[d] School of Computing and Scientific Computing and Imaging Institute, University of Utah, United States

ARTICLE INFO

Keywords:
Surrogate model
Gaussian process
Optimal time allocation
Uncertainty
Molecular dynamics simulations
Glass-forming system

ABSTRACT

Simulation models have been utilized in a wide range of real-world applications for behavior predictions of complex physical systems or material designs of large structures. While extensive simulation is mathematically preferable, external limitations such as available resources are often necessary considerations. With a fixed computational resource (i.e., total simulation time), we propose a Gaussian process-based numerical optimization framework for optimal time allocation over simulations at different locations, so that a surrogate model with uncertainty estimation can be constructed to approximate the full simulation. The proposed framework is demonstrated first via two synthetic problems, and later using a real test case of a glass-forming system with divergent dynamic relaxations where a Gaussian process is constructed to estimate the diffusivity and its uncertainty with respect to the temperature.

1. Introduction

Due to the increase in computational power, computer-based simulation models have been extensively utilized to predict behaviors or design materials for a wide range of real-world applications. However, their application can be limited if they are computationally expensive. For example, studying the statistics of a random process or optimizing the performance over a design space requires a large number of simulation evaluations, and consequently becomes computationally expensive (or even impossible) as a single simulation may require minutes, hours or even days to complete [1]. One way to circumvent this issue is to construct a computationally cheaper surrogate model based on a limited number of simulation evaluations to approximate the behavior of the full simulation model.

Extensive research work has been conducted on surrogate construction. For example, different surrogate models, such as polynomial response surfaces [2], radial basis functions [3], Gaussian processes (or kriging) [4] and neural networks [5], have been explored to approximate simulations in different application fields. Provided a surrogate model, sampling techniques for locations (where the simulation will be implemented and evaluated) are proposed so that the constructed surrogate is more accurate with limited samples [6–10]. Despite the significant contribution of the aforementioned work, the comprehensive research on the optimal allocation of the computational resources (or cost optimization) for a more accurate surrogate construction is missing. It would be of great interest to computational scientists to have guidance on the proper allocation of a fixed total computational time, including the number of simulation evaluations, the locations of simulations in a parameter space, and the computational time of each simulation (with error estimates). In this work, we will predefine the number of simulation evaluations and their parameter space locations, then focus on the optimal allocation of the fixed total computational time to the chosen simulations based on Gaussian process (GP).

Gaussian process (or kriging) has been widely used as a nonlinear regression technique to approximate simulations across various applications [4]. Tremendous research effort has been committed to the adaptive point (or sample) selection for Gaussian process so that a higher expected improvement can be obtained [8–10]. Our current work will fix the points (or samples) and focus on the error of simulations at the existing points.

The proposed GP-based optimal time allocation framework will be applied to molecular dynamics (MD) simulations. MD simulations have been extensively utilized to compute the estimates of ensemble averages of key quantities of interest and hence study the behavior of molecular

* Corresponding author.
E-mail addresses: John.Chilleri@student.nmt.edu (J. Chilleri), Yanyan.He@unt.edu (Y. He), d.bedrov@utah.edu (D. Bedrov), kirby@cs.utah.edu (R.M. Kirby).

https://doi.org/10.1016/j.commatsci.2020.110178
Received 12 August 2020; Received in revised form 7 November 2020; Accepted 9 November 2020
Available online 25 December 2020

systems and materials in various areas of science and engineering [11]. However, accurate sampling of the evolution of the system can be challenging due to the necessity of long-time MD simulations. To overcome this issue, accelerated molecular dynamics (AMD) methods have been proposed to capture the equilibrium states of the system in much less computational time using various techniques [11], such as modified potentials augmented by bias potentials to accelerate and extend the time scale in MD simulations [12,13], parallel replica dynamics based on parallel power to boost the time scale [14], hyperdynamics based on transition state theory and importance sampling [15], and temperature-accelerated dynamics by raising the temperature but allowing only events occurring at the original temperature [16]. However, these methods require in-depth understanding of the underlying MD processes and possible modification of the simulation code [17]. On the other hand, non-intrusive computational predictive models have been applied to MD simulation to achieve both accuracy and efficiency. For example, both deterministic polynomial chaos (PC) expansions based on nonintrusive spectral projection and non-deterministic PC expansions based on Bayesian inference are constructed to approximate the full MD simulations as surrogates [18,19]; Bayesian uncertainty quantification frameworks with parallelization have been used to deal with the uncertainty in the parameters of force fields employed in MD simulations, and adaptive Kriging models have been proposed to reduce the computational cost in [20,21]; function derivatives have been applied to quantify and correct uncertainties that originate from Lennard-Jones (LJ) two-body pair potential [22]; a multi-fidelity sampling approach has been proposed to enhance the convergence of properties predicted by MD simulations and hence serve as an accurate surrogate model to approximate the quantities of interest [17]. Despite the significant contribution to computational saving in MD simulations from the aforementioned work, the study on optimal time allocation (or cost distribution) for MD simulations over a temperature range for an accurate surrogate construction is missing. With fixed available computational resources (such as the total simulation time), better choices on where and how long to run MD simulations will help to produce a more accurate surrogate model. Although the Multi-fidelity sampling approach has explored where to run MD simulations to some extent, we will specifically focus on the optimal allocation (distribution) of total simulation time to MD simulations at fixed locations in our current work.

The developed algorithm will be applied to MD simulations of glass-forming liquids since very few reliable algorithms exist that perform well with the equilibration of glass-forming liquids at low temperatures [23]. We will consider a binary mixture of type A and B molecules that interact via a Lennard-Jones potential, which has been known to give a glassy system at low temperatures [24,25]; that is, it is not prone to crystallization as diffusion of species is diverging. Specifically, the composition studied in the current work includes 204 of type A and 820 of type B molecules, and the diameters of spheres and strength of interactions are slightly different. MD simulations are then conducted at a range of different temperatures to extract the diffusion coefficients. While extensive simulation is mathematically preferable, external limitations such as available resources are often necessary considerations. With a fixed total simulation time, we explore the optimal time allocation to MD simulations over a set of discrete temperatures so that an accurate Gaussian process can be constructed to serve as a surrogate to predict the temperature dependence of diffusion coefficients.

This paper is organized as follows. In Section 2, we provide the problem setup including the assumptions and the optimization problem for optimal time allocation. Following that, we introduce the basics of Gaussian process and propose a GP-based Optimization procedure for time allocation in Section 3. In Section 4, the proposed optimization procedure is demonstrated using synthetic numerical examples, and then applied to molecular dynamics simulations for a glassy system in Section 5. A summary is provided in Section 6.

2. Problem setup

Let the simulation model $\mathscr{M}$ map the parameter $\xi \in \Xi$ to the output $\widehat{u}$, which approximates the truth $u(\xi)$ (the simulation output normally deviates from the truth due to the uncertainty in the simulation process). The objective is to construct a cheaper surrogate $\widetilde{u}$ to approximate the truth u based on a finite number of implementations of the simulation model. For the purposes of the current work, a few assumptions are made for the simulation model $\mathscr{M}$.

1. The simulation model is computationally expensive and the total computational resource is limited. For example, the total available computational time (or the total cost) is bounded by a constant C.
2. The accuracy of the simulation model output $\widehat{u}$ depends on the simulation time. For example, numerical solvers with higher-order enrichment and/or finer mesh produce more accurate results but take a longer time. Let $\widehat{\epsilon}$ denote the error in $\widehat{u}$, $\widehat{\epsilon}$ depends on the simulation time (or cost) c.
3. The accuracy of the simulation model output $\widehat{u}$ depends on the input ξ, i.e., with the same computational time, the errors in $\widehat{u}(\xi_1)$ and $\widehat{u}(\xi_2)$ are different for $\xi_1 \neq \xi_2$. For example, $\widehat{u}$ may have a finer structure for ξ_1 and consequently require more computational power to reach the same accuracy as ξ_2. Under this assumption, $\widehat{\epsilon}$ depends on ξ as well.
4. The heuristic analytical form of the error in the numerical output $\widehat{u}$ can be known. Numerical analysis provides a rough estimation for the error in the numerical solution from certain numerical schemes. For example, the error in the numerical root based on bisection method can be estimated.

With the above assumptions, the simulation model $\mathscr{M}$ produces the model output $\widehat{u}(\xi, c)$ associated with error $\widehat{\epsilon}(\xi, c)$ for a given input ξ and a fixed simulation time (or cost) c. Then the objective becomes the optimal allocation of the total computational time C over the domain Ξ to construct a surrogate $\widetilde{u}(\xi)$ such that $\|\widetilde{u}(\xi) - u(\xi)\|_{L^2(\Xi)}$ is minimized subject to

$$\int_{\xi\in\Xi} c(\xi)d\xi = C, \qquad c(\xi)\geqslant 0, \tag{1}$$

where the surrogate $\widetilde{u}(\xi)$ is constructed using simulations $\widehat{u}(\xi, c)$.

In order to employ numerical computation, the problem needs to be discretized over the domain Ξ, which means the surrogate is constructed based on a finite number of implementations of the simulation model $\mathscr{M}$. Let N denote the number of simulations, $\{(\xi, c)_i\}_{i=1}^N$ be the N realizations of the parameter ξ where the simulations take place and the corresponding simulation times, $\{(\widehat{u}, \widehat{\epsilon})_i\}_{i=1}^N = \{\widehat{u}(\xi_i, c_i), \widehat{\epsilon}(\xi_i, c_i)\}_{i=1}^N$ be the output from N simulations and the corresponding simulation errors. The objective is then to optimize the number of simulations N, the location of simulations ξ_is, and the time allocation c_is among all simulations, i.e.,

$$\min_{N,\xi_i,c_i} \|\widetilde{u}(\widehat{u}(\xi_i, c_i), \widehat{\epsilon}(\xi_i, c_i), \xi) - u(\xi)\|_{L^2(\Xi)}, \tag{2}$$

subject to $\sum_{i=1}^N c_i = C$ and $c_i \geqslant 0$.

We start with a simplified problem: the number of implementations of the simulation N, and the locations of simulations $\{\xi_i\}_{i=1}^N$ are predefined. Then the objective function of the optimization problem becomes

$$\min_{c_i} \|\widetilde{u}(\widehat{u}(c_i), \widehat{\epsilon}(c_i), \xi) - u(\xi)\|_{L^2(\Xi)}, \tag{3}$$

$$\sum_{i=1}^N c_i = C, \qquad c_i \geqslant 0. \tag{4}$$

However, the true error in the constructed surrogate (i.e., the quantity

inside the 2-norm) is not available due to the lack of the truth. We propose to solve this time allocation optimization problem using the concept of Gaussian process regression. Specifically, we construct a Gaussian process as the surrogate $\widetilde{u}$, and use its standard deviation function (depending on ξ) to serve as the estimation for its error. The details of the proposed method will be provided in the following section.

3. Method

3.1. Gaussian process regression basics

A Gaussian process $f(x)$ is a collection of random variables, any finite number of which have a joint Gaussian distribution [4]. It can be completely determined by its mean $m(x)$ and covariance functions $k(x, x^{'})$, where

$$m(x) = E[f(x)], \tag{5}$$

$$k(x, x^{'}) = E[(f(x) - m(x))(f(x^{'}) - m(x^{'})]. \tag{6}$$

In Gaussian process regression models, the mean $m(x)$ is unknown and normally assumed to be 0 as a prior. Then for a number of input points $\boldsymbol{x}$, the corresponding output is a random Gaussian vector $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, K(\boldsymbol{x},\boldsymbol{x}))$, where the element $K(i,j) = k(x_i, x_j)$. Gaussian process regression models can be used for prediction given observations. Normally, one does not have access to the true function to obtain the exact function values as observations, instead, noisy versions of the function output are obtained $\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\tau}$. With the assumption of additive independent distributed Gaussian noise $\boldsymbol{\tau}$ with variance $\boldsymbol{\sigma}^2$, the prior covariance on noisy observations becomes

$$cov(\boldsymbol{y}) = K(\boldsymbol{x},\boldsymbol{x}) + I\boldsymbol{\sigma}^2. \tag{7}$$

Let $\boldsymbol{x}_*$ be the collection of test points for prediction. The joint distribution of the observations $\boldsymbol{y}$ and the function values $\boldsymbol{f}_*$ at the test locations under the prior is

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\boldsymbol{x},\boldsymbol{x}) + I\boldsymbol{\sigma}^2 & K(\boldsymbol{x},\boldsymbol{x}_*) \\ K(\boldsymbol{x}_*,\boldsymbol{x}) & K(\boldsymbol{x}_*,\boldsymbol{x}_*) \end{bmatrix}\right). \tag{8}$$

The derived posterior distribution of the function values $\boldsymbol{f}_*$ is

$$\boldsymbol{f}_* | \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}_* \sim \mathcal{N}(\bar{\boldsymbol{f}}_*, cov(\boldsymbol{f}_*)), \tag{9}$$

where

$$\bar{\boldsymbol{f}}_* = K(\boldsymbol{x}_*,\boldsymbol{x})[K(\boldsymbol{x},\boldsymbol{x}) + I\boldsymbol{\sigma}^2]^{-1}\boldsymbol{y}, \tag{10}$$

$$cov(\boldsymbol{f}_*) = K(\boldsymbol{x}_*,\boldsymbol{x}_*) - K(\boldsymbol{x}_*,\boldsymbol{x})[K(\boldsymbol{x},\boldsymbol{x}) + I\boldsymbol{\sigma}^2]^{-1}K(\boldsymbol{x},\boldsymbol{x}_*). \tag{11}$$

3.2. Optimization procedure of time allocation

Let the constructed surrogate $\widetilde{u}$ be a Gaussian process with prior distribution $\widetilde{u}(\boldsymbol{\xi}) \sim \mathcal{N}(\boldsymbol{0}, k(\boldsymbol{\xi}, \boldsymbol{\xi}^{'}))$. The covariance function kernel is chosen to be the squared exponential due to its popularity. There are other kernels in the literature and the impact of choosing different kernels on our optimization will be discovered in our future work. The formula for the squared exponential kernel is

$$k(\boldsymbol{\xi}, \boldsymbol{\xi}^{'}) = a\exp\left(\frac{-|\boldsymbol{\xi} - \boldsymbol{\xi}^{'}|^2}{2b}\right), \tag{12}$$

where the hyper-parameters a and b represent the prior standard deviation and the correlation length, respectively. To estimate the hyper-parameters, we run N simulations at the fixed locations $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_N\}$ with fixed small initial simulation time $c_0 \ll C/N$ for all simulations, which produce the initial observations $\widehat{\boldsymbol{u}}_0$, and error $\widehat{\in}_0$. The hyper-parameters can then be obtained by maximizing the likelihood based on the initial observations.

As mentioned in Section 2, the true error in the surrogate $\widetilde{u}$ is not available due to lack of the truth u, the posterior standard deviation of the Gaussian process surrogate $\widetilde{u}$ will be adopted to serve as the estimation of its error. To obtain the posterior distribution, we extract the observational data from N simulations $\widehat{\boldsymbol{u}} = \{\widehat{u}_i\} = \{\widehat{u}(\xi_i, c_i)\}$ at known fixed locations $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_N\}$ with unknown corresponding simulation time (cost) $\boldsymbol{c} = \{c_1, \ldots, c_N\}$ (to be decided in the optimization). The simulation errors are obtained from the error function $\widehat{\in} = \{\widehat{\in}_i\} = \{\widehat{\in}(\xi_i, c_i)\}$. Let the test points be M equally-distanced points $\boldsymbol{\xi}_* = \{\xi_{*,1}, \ldots, \xi_{*,M}\}$ in space Ξ, and $\widetilde{\boldsymbol{u}}_*$ be the corresponding surrogate output. With Eq. (11), the variance of the function output $\widetilde{\boldsymbol{u}}_*$ is obtained as the diagonal of matrix $cov(\widetilde{\boldsymbol{u}}_*)$, denoted as $\boldsymbol{e}^2$.

$$\boldsymbol{e}^2 = var(\widetilde{\boldsymbol{u}}_*) = \mathrm{diag}(cov(\widetilde{\boldsymbol{u}}_*)), \tag{13}$$

where

$$cov(\widetilde{\boldsymbol{u}}_*) = K(\boldsymbol{\xi}_*,\boldsymbol{\xi}_*) - K(\boldsymbol{\xi}_*,\boldsymbol{\xi})[K(\boldsymbol{\xi},\boldsymbol{\xi}) + I\widehat{\in}^2]^{-1}K(\boldsymbol{\xi},\boldsymbol{\xi}_*), \tag{14}$$

Our goal is to minimize the 2-norm of vector $\boldsymbol{e}^2$ with respect to the unknown simulation time allocations c_is as

$$\min_{c_i} \|\boldsymbol{e}^2(\boldsymbol{c})\|_2, \tag{15}$$

$$\sum_{i=1}^{N} c_i = C, \qquad c_i \geqslant 0. \tag{16}$$

The algorithm of the Gaussian process-based optimal time allocation procedure (we name it GP-based Optimization) is outlined below.

1. Specify the total simulation time (total cost) C, the number of simulations N, the locations of simulations $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_N\}$.
2. Obtain the heuristic analytical form of simulation error $\widehat{\in}$, which depends on simulation location ξ and simulation time c, i.e., the function $\widehat{\in}(\xi, c)$.
3. Determine the hyper-parameters a and b in the prior covariance matrix based on initial observations $\widehat{\boldsymbol{u}}_0, \widehat{\in}_0$, which is obtained from N base simulations at $\boldsymbol{\xi}$ with simulation time c_0 for each simulation.
4. Minimize the variance of the obtained Gaussian process $\boldsymbol{e}^2$ for the optimal time allocation $\boldsymbol{c}$.
5. Run simulations at locations $\boldsymbol{\xi}$ with obtained optimal simulation time $\boldsymbol{c}$ to obtain observational data $\widehat{\boldsymbol{u}}, \widehat{\in}$. Then the Gaussian process mean can be generated using Eq. (10).

The initial observations in step 2 are considered as extra information and $\boldsymbol{c}_0$ does not count as part of the total simulation time C. In the case that one has to choose $\boldsymbol{c}_0$ for initial observations, equal values can be assigned to each element (i.e., $\boldsymbol{c}_0 = [c_0, c_0, \ldots, c_0]$) for simplicity and the summation of $\boldsymbol{c}_0$ should be much smaller than the total computational resources C. The large error in initial observations will be considered in the optimization process through the observational error $\{\widehat{\in}(\xi_i, c_0)\}_{i=1}^{N}$.

4. Synthetic numerical examples

In this section, the proposed optimal time allocation procedure (based on a Gaussian process) will be demonstrated using two synthetic numerical examples with a polynomial function and an exponential function. For the purpose of comparison, we also perform the intuitive (or naive) way of optimizing time allocation (we name it Naive Optimization): minimizing the 2-norm of the discrete N simulation errors $\{\widehat{\in}(\xi_i, c_i)\}_{i=1}^{N}$ with respect to inputs $\boldsymbol{c} = \{c_i\}_{i=1}^{N}$. The optimization formulation is provided as follows.

$$\min_{c_i} \|\widehat{\in}(\boldsymbol{c})\|_2^2 = \min_{c_i} \{\widehat{\in}^2(c_1) + \cdots + \widehat{\in}^2(c_N)\}. \tag{17}$$

4.1. Example of polynomial function

Let the true function $u(x)$ be a third-order polynomial, and $\widehat{u}(x)$ be the simulation output with noise τ_i, which has zero mean and standard deviation $\widehat{\epsilon}(x_i, c_i)$.

$$u(x) = 1 + \frac{x}{2} + \frac{x^2}{4} + \frac{x^3}{8}, \qquad \widehat{u}(x_i, c_i) = u(x_i) + \tau_i, \tag{18}$$

$$\tau_i \sim \mathcal{N}(0, \widehat{\epsilon}^2(x_i, c_i)), \widehat{\epsilon}(x, c(x)) = \frac{1}{100}\left(\frac{1 + \frac{(x-1)^2}{5}}{\left(0.1 + \frac{(c(x)-0.1)}{5}\right)^2}\right). \tag{19}$$

From the formula $\widehat{\epsilon}$, one can observe that the simulation error is minimum for $x = 1$, and gets larger as x increases with fixed simulation time (or cost c); and the error is also larger for a smaller simulation time for a fixed x.

The fixed simulation locations and the fixed total simulation time are specified as:

$$\boldsymbol{x} = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10], \tag{20}$$

$$C = 8. \tag{21}$$

Our goal is to construct a Gaussian process surrogate $\widetilde{u}(x)$ to approximate the truth $u(x)$.

To have relatively meaningful prior information, we estimate the hyper-parameters in the covariance matrix based on initial observational data, which is produced by $N = 11$ simulations at $x = 0, \ldots, 10$ with the fixed initial simulation time $\boldsymbol{c}_0 = [0, \ldots, 0]$.

The proposed GP-based Optimization is implemented to find the optimal time allocation among the $N = 11$ simulations. From Fig. 1, one can observe that the simulation time is distributed to simulations at $x = 0, 1, 3, 5, 8, 10$, and the simulations at larger $x(> 1)$ are allocated longer times in general since they have larger errors compared to those at smaller $x(> 1)$ with the same simulation time. The allocated times to the boundaries $x = 0$ (which has more error than $x = 1$) and $x = 10$ (which has more error than $x = 8$) are reduced since their correlation with other locations in the Gaussian process is weaker. Compared to the proposed GP-based Optimization method, the Naive Optimization allocates the monotonically increasing simulation times to the simulations ordered based on their simulation errors (i.e., from smaller to larger x with the exception of the boundary $x = 0$). From the results, one can conclude that the Naive Optimization allocates the simulation time based only on the simulation error, while GP-based Optimization allocates the simulation time based on both simulation error and the correlation between x-locations.

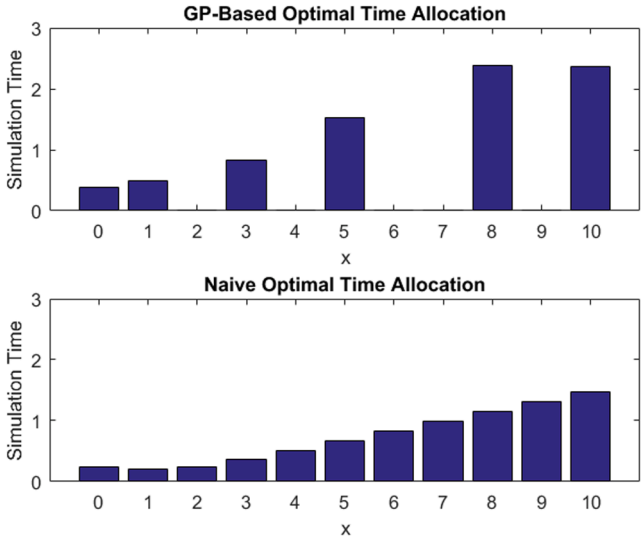

Fig. 1. The optimal time allocation for the example of polynomial function: (top) GP-based Optimization, and (bottom) Naive Optimization.

With the obtained optimal time allocations, the Gaussian processes $\widetilde{u}$ from the two methods are generated and plotted in Fig. 2(a). The red color is for Naive Optimization while the blue color is for GP-based optimization. The dots and stars denote the simulation output and the simulation error at $N = 11$ locations with the optimal simulation times, based on which, the Gaussian process surrogates are then constructed. The constructed GP surrogate means from both methods are compared to the true function depicted by the black solid curve. The comparison shows that the red curve (from Naive Optimization) is slightly deviated from the black curve (truth) while the blue curve (from GP-based Optimization) aligns with the black curve to a greater extent. Note: Based on the design principle of Naive Optimization (i.e., minimizing the L_2 norm of errors for 11 discrete simulations), one can expect that the overall 11 discrete simulations will have small errors as shown in Fig. 2(a). However, it does not necessarily result in a more accurate (continuous) mean curve comparing to GP-based Optimization, which takes into account the spacial information and produces less errors for the chosen locations at $x = 1, 3, 5, 8, 10$.

Since it is difficult to visually compare data due to the large span of the values in the y-axis, we focus on the region inside the black-dashed-box and the close-up figures for both mean curves and 95% confidence intervals (curves) are provided in Fig. 2(b,c). One can easily observe that the surrogate constructed from the GP-based Optimization is closer to the true mean, and that the GP-based Optimization produces a more precise surrogate with less variance.

To further compare the two methods, the L_2 norms of the error in the means of the Gaussian process surrogates and the variance are calculated for both methods. From Table 1, one can conclude that GP-based Optimization performs much better in this example of a polynomial function.

4.2. Example of exponential function

Let the true function $u(x)$ be an exponential function, and the simulation output $\widehat{u}(x_i)$ have the same noise τ_i as in the first synthetic example. Again, the simulation error is minimum at $x = 1$, and gets larger as x increases with fixed simulation time c, and the error is also larger for a smaller simulation time for a fixed x.

$$u(x) = \exp(x), \qquad \widehat{u}(x_i, c_i) = u(x_i) + \tau_i, \tag{22}$$

$$\tau_i \sim \mathcal{N}(0, \widehat{\epsilon}^2(x_i, c_i)), \widehat{\epsilon}(x, c(x)) = \frac{1}{100}\left(\frac{1 + \frac{(x-1)^2}{5}}{\left(0.1 + \frac{(c(x)-0.1)}{5}\right)^2}\right). \tag{23}$$

The fixed simulation locations and the fixed total simulation time are specified as:

$$\boldsymbol{x} = [0, 1, 2, 3, 4, 5], \tag{24}$$

$$C = 8. \tag{25}$$

Similarly, our goal is to construct a Gaussian process surrogate $\widetilde{u}(x)$ to approximate the truth $u(x)$.

We first generate observational data at $x = 0, \ldots, 5$ with specified initial simulation time $\boldsymbol{c}_0 = [0, \ldots, 0]$, based on which we estimate the hyper-parameters in the covariance matrix. Then we implement both GP-based Optimization and Naive Optimization for the optimal time allocations among the $N = 6$ simulations. From Fig. 3, one can observe that Naive Optimization allocates more simulation time to simulations that have more simulation error, while GP-based Optimization allocates

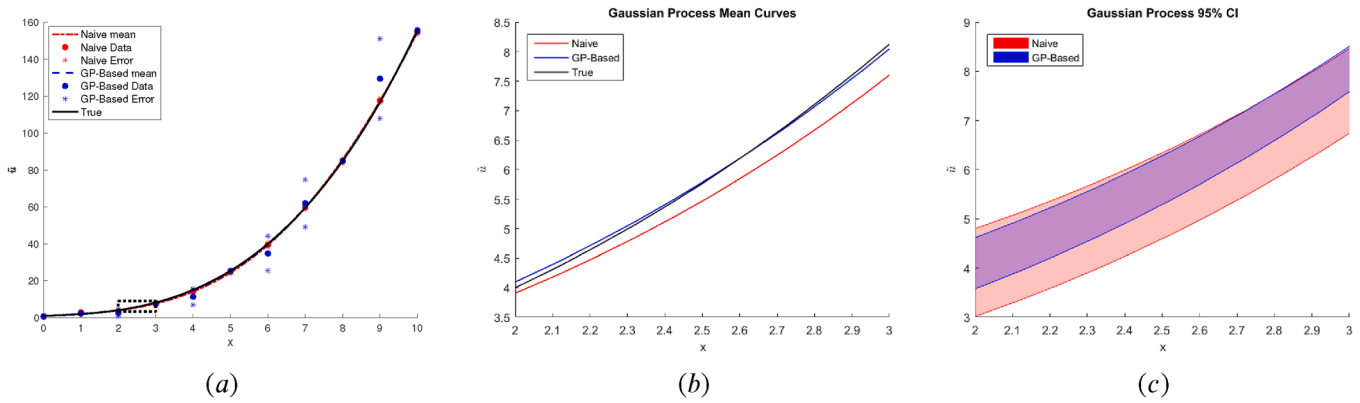

Fig. 2. The comparison of (a) the constructed Gaussian process surrogate; (b) close-up Gaussian processes mean curves with range $[2,3]$; and (c) the 95% confidence intervals with range $[2,3]$.

Table 1
The comparison of GP-based Optimization and Naive Optimization regarding the error in mean surrogate and the variance for the example of a polynomial function.

	GP-based Optimization	Naive Optimization
L_2 Error in $\widetilde{u}$	2.3064	5.9305
L_2 of Variance	0.9768	3.5601

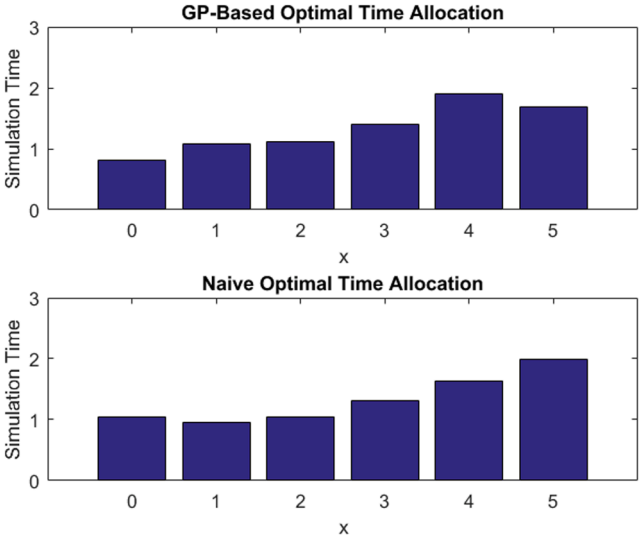

Fig. 3. The optimal time allocation for the example of exponential function: (top) GP-based Optimization, and (bottom) Naive Optimization..

more simulation time to x with a balance between larger simulation error and more influence on other locations.

With the obtained GP-based optimal time allocation, the simulation output and the error can be calculated for the $N = 6$ locations $\{x_i\}_{i=0}^{5}$. Based on which, the Gaussian process surrogate is then constructed (see Fig. 4(a) for the mean curve). To compare the two different optimization frameworks, the close-up (with range $[0.4, 0.95]$) mean curves and the 95% confidence interval curves are provided in Fig. 4(b), which shows that the GP-based Optimization provides slightly better results (more accurate mean and less variance) for this exponential example.

Similarly, the L_2 norms of the error in the mean of the Gaussian process surrogate and the variance are calculated for both methods (see Table 2). The quantitative comparison also verifies the better performance of the GP-based Optimization method.

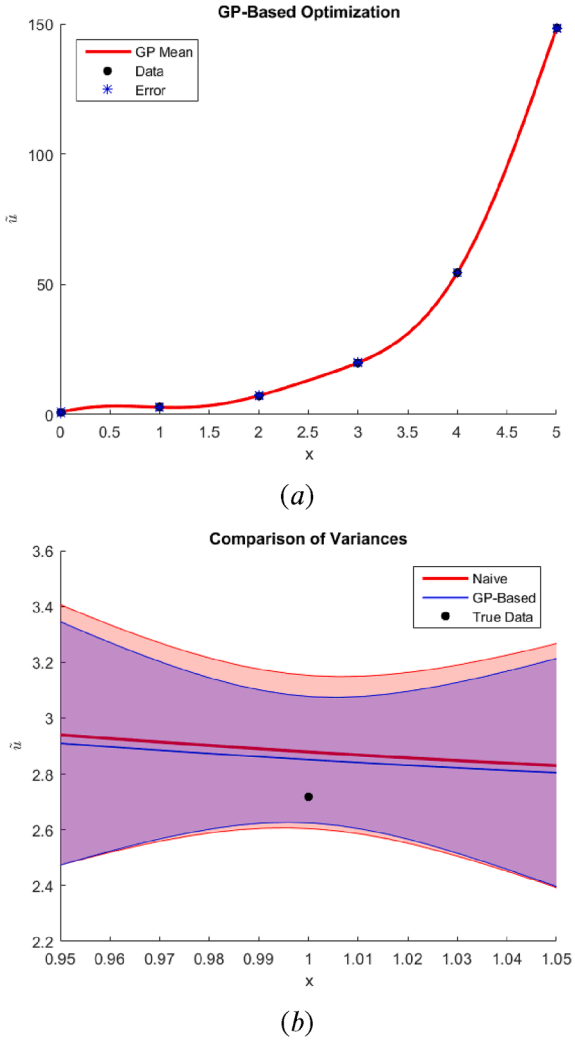

Fig. 4. The constructed Gaussian process for the example of exponential function: (a) the mean curve based on GP-based Optimization, and (b) the comparison of GP-based Optimization and Naive Optimization.

Table 2
The comparison of GP-based Optimization and Naive Optimization regarding the error in mean surrogate and the variance for the example of exponential function.

	GP-based Optimization	Naive Optimization
L_2 Error in $\tilde{u}$	20.3904	20.5355
L_2 of Variance	13.3907	13.5871

5. Molecular dynamics simulations

5.1. Molecular dynamics simulation model

As a real application example, in the current work, we consider a binary mixture of type A and B molecules that interact via a Lennard–Jones potential $U(r) = 4\epsilon[(\sigma/r)^{12} - (\sigma/r)^{6}]$ with parameters for self-interactions $\epsilon_{AA} = 0.5$ kcal/mol, $\epsilon_{BB} = 1.0$ kcal/mol, $\epsilon_{AB} = 1.5$ kcal/mol and $\sigma_{AA} = 3.4214$ Å, $\sigma_{BB} = 3.8880$ Å and $\sigma_{AB} = 3.1104$ Å. The systems were comprised of 204 of type A and 820 of type B molecules and simulated in a cubic box with periodic boundary conditions. This composition is known to not be prone to crystallization, instead it has the diffusion of species diverging, and gives an amorphous glassy system at low temperatures [26]. Therefore, it can be used to represent systems in which the characteristic dynamic relaxations/properties are diverging with lowering the temperature and hence requiring longer and longer simulations to get reliable statistics. In addition, the system may require a long time to reach equilibrium/stationary state especially at low temperatures, and consequently the MD simulations take a long time to produce meaningful statistical results. Therefore, it is important to construct fast predictive models based on a limited amount of MD simulations at a number of different temperatures to serve as a surrogate to approximate the full MD simulations. In this section, we employ the GP-based Optimization framework to explore the optimal allocation of the available computational resources to MD simulations at a set of discrete temperatures, and then construct an accurate Gaussian process surrogate based on the limited amount of initial MD simulations. To simplify the problem, we fixed the temperatures in the 180–250 K range (T = 180; 185; 190; 195; 200; 210; 220; 230; 250 K) at which the MD simulations will be launched to produce diffusion coefficients as model output.

5.2. Initial data

We first collect initial data over $t = 2ns$ (or cost $c = 2ns$) at each temperature for the estimation of hyper-parameters of the covariance matrix. At a high temperature ($T = 250\,K$), molecules move fast and hence we can reach equilibrium and sample properties of the system very quickly. Initial 2*ns* equilibration of the system was conducted in the NPT ensemble, energy and dimensions reached stationary values very quickly. To prepare initial systems for other temperatures in the 180–250 K range, we performed consecutive steps of dropping the temperature and equilibrating for 2*ns* (i.e., from 250 to 230 K with 2ns simulation at 230 K; then from 230 to 220 K with 2*ns* of simulation at 220 K), which corresponds to an effective cooling rate of 5 degrees per ns. After 2*ns* equilibration, we simulated each system for an additional 2*ns* to sample the mean squared displacements (MSD) which we use to acquire self-diffusion coefficients $D = \lim_{t\to\infty}(\mathrm{MSD}/6t)$. Note: Based on the diameter of type A particles (3.4214 Å) and type B particles (3.8880 Å), one can specify the minimum simulation time (denoted as t_0) that allows the molecules to move at least over their own dimensions, and consequently to be considered as an onset for the time required for the molecule to start a diffusive motion. The diffusion coefficients *D* may be extracted more accurately for higher temperatures with 2*ns* trajectory; however, they are extracted less accurately (if extractable) for lower temperatures. This is illustrated in Fig. 5, where we show MSD at $T =$ 220, 200, and 185 K. Note that MSD was fit from the time where MSD = 10 up until half of the trajectory length. For 220 K we show that MSD from the 2ns and 10ns trajectories are almost identical and hence the *D* obtained from the 2ns data is already well-converged. For 200 K we see a noticeable difference between the 2ns and 90ns runs and hence the *D* extracted from the 2ns data will be less accurate. For 185 K, we only show the 220ns trajectory length as the 2ns simulation would not result

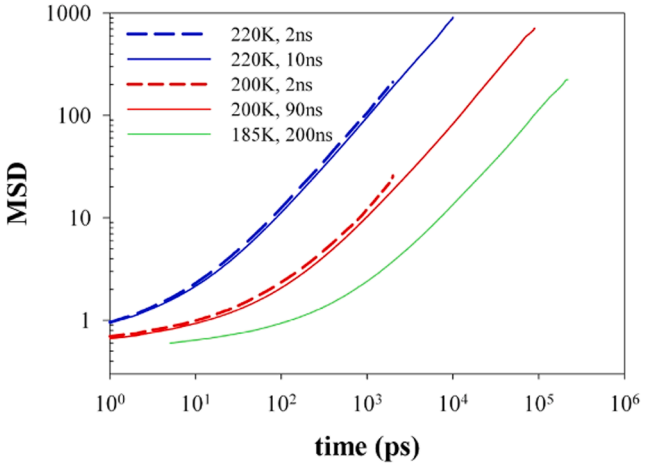

Fig. 5. MSD at $T = 220,\ 200,$ and 185 K.

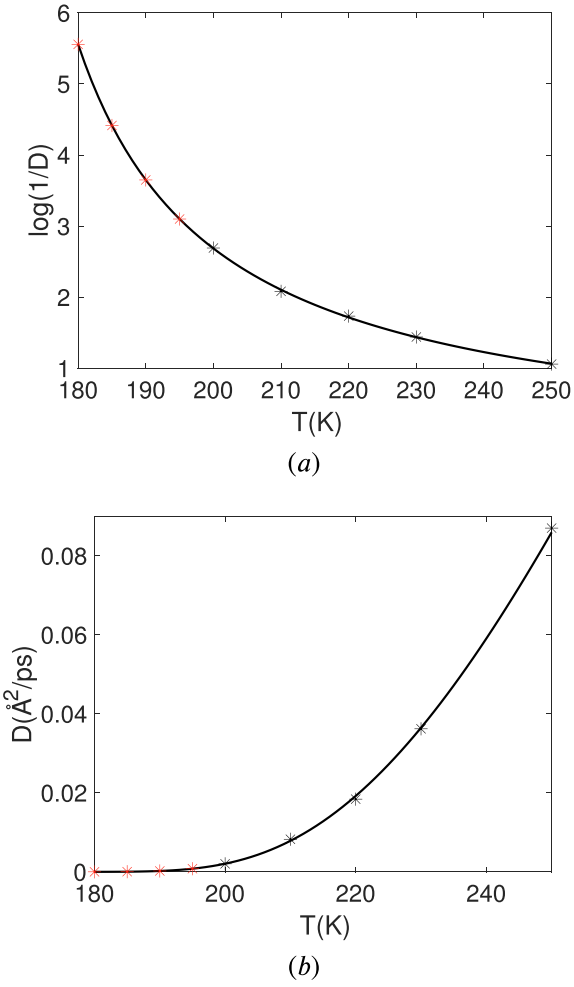

Fig. 6. (a) VF model ($log(1/D)$ versus T) fit to the higher temperature data (from 2*ns* of initial trajectory length); (b) Visualization of best fit VF model in *D* versus *T*.

Table 3
Diffusivities at different temperatures for 2ns simulation.

T (K)	250	230	220	210	200	195*	190*	185*	180*
D $\left(\frac{\text{Å}^2}{\text{ps}}\right)$	0.0870	0.0362	0.0184	0.00825	0.00203	7.94e−4	2.24e−4	3.84e−5	2.78e−6

in any meaningful MSD that could be used for the extraction of a diffusion coefficient.

Since type A particles move faster and make it possible to extract D for more temperatures under consideration, we will focus on the analysis of type A particles in the current work. With $2ns$ of initial trajectory length, we can extract the diffusion coefficients D at $T = 250, 230, 220, 210, 200\,K$. For the remaining temperatures, we utilize the Vogel-Fulcher (VF) model, which is known to describe the dependence of transport properties on the temperature in glass-forming liquids:

$$\log\left(\frac{1}{D}\right) = \alpha + \frac{\beta}{T - T_0}, \tag{26}$$

where T_0 is the temperature where diffusion is diverging and it is usually very close to the glass transition temperature [27–29]. Ultimately predicting this temperature is one of the important objectives of such simulations. Fitting the VF model to the extracted D values provides the estimation of the unknown parameters,

$$\alpha = -0.2445, \qquad \beta = 118.50, \qquad T_0 = 159.57. \tag{27}$$

The best fit VF model are plotted in Fig. 6 (in both log scale and normal scale). The black stars represent the higher temperature data (from $2ns$ of initial trajectory length) used to fit the model while the red stars represent the estimated data from the fit (VF model) at lower temperatures.

Then the complete initial diffusivity data D (associated with $t = 2ns$) is provided in Table 3, where the ones corresponding to temperature $T < 200\,K$ (indicated by *) are the estimation from the VF model rather than a simulation value.

As mentioned earlier, the initial preparation simulations (i.e. cooling $+2ns$ equilibration simulation steps) and the subsequent $2ns$ of sampling (at each of the nine temperatures) for the initial data set are considered as extra information and the dedicated computational time does not count as part of the total simulation time for optimal allocation.

5.3. Error estimation for MD simulations

The error in the time-dependent diffusion coefficient from molecular dynamics simulation has been studied by Kim et al. [30]. Their results showed that the error bars are proportional to the square root of the trajectory lengths or time (for example, increasing the trajectory length by a factor of 4 will reduce the error bar by a factor of 2). Define the simulation error to be $\hat{\epsilon}_0$ for the extracted D at $2ns$ simulation, then the error after $c + 2$ns simulation can be estimated as

$$\hat{\epsilon}(T, c) = \sqrt{\frac{2}{c+2}}\hat{\epsilon}_0(T). \tag{28}$$

The initial simulation error $\hat{\epsilon}_0$ is estimated based on the results in Fig. 7, which shows that trajectory length has to be on the order of $(50 - 100)t_0$ in order to converge. For this particular system, we also observe that it takes about $75t_0$ for diffusion to reach reasonable values (within 10% of the converged value).

We then define the initial error for $2ns$ simulation as

$$\hat{\epsilon}_0(T) = \lambda\sqrt{\frac{75t_0(T)}{2}}10\%. \tag{29}$$

Since the initial error is estimated based on the trajectories at higher temperatures, a scalar λ is introduced in the initial error formula for a range of temperatures from 180 K to 250 K. In our current work, we set $\lambda = 0.1$. Note: In general, it is difficult (if not impossible) to obtain the exact simulation error without knowing the true function, one needs to estimate the simulation error based on the available knowledge/information. In the current work, the formula for simulation error is only an estimation based on the prior analysis of the MD simulation. We will directly use this estimation in the process of optimizing the time allocation, leaving the sensitivity analysis of optimal time allocation with respect to the estimation of the error function as our future work.

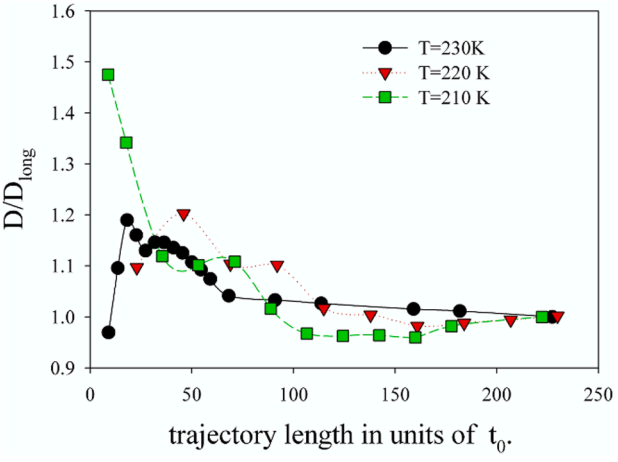

Fig. 7. Diffusion coefficients D normalized by the D_{long} extracted from the longest trajectory as a function of trajectory length (time) measured in units of t_0.

5.4. Data for validation

In order to validate our method, we also conduct MD simulations for a prolonged time to collect converged data (which can be considered as gold standard). Table 4 includes the simulation time, minimum time t_0 to reach MSD $= 10\,\text{Å}^2$, and the collected diffusion coefficients D.

Since 200ns of simulation time (or trajectory) does not yield $75t_0$ for $T = 180\,K, 185\,K$, the extracted D may not have converged for these temperatures. Therefore we use the diffusion coefficients corresponding to the first 7 temperatures to fit the Vogel–Fulcher model as

$$\log\left(\frac{1}{D}\right) = -0.9936 + \frac{212.5}{T - 144.5}, \tag{30}$$

which corresponds to the reduced temperature $T^* = \frac{k_B T}{\epsilon_{AA}} = 0.574$ andwill

Table 4
Diffusivities at Various Temperatures for Sufficient Simulation.

$T(K)$	Simulation time (ns)	t_0(ps)	D_A (Å^2/ps)
250	2	17	0.0870
230	2	44	0.0362
220	20	87	0.01553
210	40	225	0.00559
200	90	878	0.0014
195	70	2563	5.418e−4
190	180	5425	2.266e−4
185	220	7050	1.9467e−4
180	180	34985	3.808e−5

serve as the truth to validate our method.

5.5. Results and discussion

We assume the fixed total computational resource $C = 100ns$. The goal is to obtain an optimal time allocation over the 9 simulations at $T = 250, 230, 220, 210, 200, 195, 190, 185, 180\,K$ so that an accurate surrogate can be constructed based on the simulation results.

Using the initial data in Table 3 and the initial error (Eq. (29)), the prior information of the Gaussian process (i.e., the initial estimation of the hyper-parameters of the covariance matrix) is obtained. Then the proposed GP-based Optimization framework is employed to optimize the time allocation. To solve the optimization numerically, a set of 100 randomly generated distributions of $C = 100$ns over 9 locations are considered as initial guesses (see Fig. 8).

The optimization for all of the initial guesses converges to the same optimal time allocation (see Fig. 9(a)): $60ns$ for simulation at $T = 185\,K$ and $40ns$ for simulation at $T = 200\,K$. On the other hand, Naive Optimization assigns monotonically decreasing simulation time to the simulation with increasing temperatures as expected (see Fig. 9(b)). For the purpose of comparison, we also generate a time allocation randomly as in Fig. 9(c).

Based on the three sets of optimal time allocations, MD simulations at 9 different temperatures are performed up to the assigned simulation times (+2ns initial time). The diffusion coefficients are extracted and provided in Table 5, where the entries with "–" indicate that the diffusivity values from the 2ns simulation, read from Table 3, are used.

Using the collected diffusion coefficients from the MD simulations with the obtained optimal time allocations, the hyper-parameters of the covariance matrix are updated. Due to the nonnegativity constraint on the diffusion coefficient (i.e., $D \geqslant 0$), nonnegativity-enforced Gaussian process surrogates are constructed [31]. The Gaussian processes with 95% CI regions are provided in Fig. 9(d–f). To visualize the difference between Gaussian process mean curves obtained by three different approaches, we have plotted the curves in log-scale and compared to the VF model (Eq. (30)) in Fig. 10(a). From the figure, one can easily observe that GP-based Optimization outperforms the Naive Optimization and the Random Generation.

To further compare the two methods and the random generation quantitatively, the L_2 norms of the error in the mean of the Gaussian process surrogate (compared to Eq. (30)) and the variance are calculated (see Table 6). The quantitative comparison of the surrogate mean error also verifies the better performance of the GP-based Optimization method. Due to the enforcement of nonnegativity and update of hyper-parameters, the GP-based Optimization method produces a slightly larger L_2 norm of variance over the whole temperature range than Random Generation. However, focusing on the lower temperature range (see Fig. 10(b)) that is of more interest from practical applications point of view, the produced Gaussian process variance at [180,190]K is smaller for the GP-based Optimization method.

As mentioned earlier, predicting the temperature T_0 in the VF model is one of the important objectives of MD simulations. Therefore, we fit the VF model to the constructed Gaussian process mean values for the estimation of the unknown parameters α, β and T_0 in VF model. Table 7 shows that the GP-based Optimization method provides the closest estimation of T_0 to the one obtained from the true data-fit VF model.

All the results are obtained based on the assumption of a limited total simulation time $C = 100ns$. Due to the simulation error and surrogate approximation error, the best GP mean curve obtained from GP-based Optimization still slightly deviate from the truth especially towards

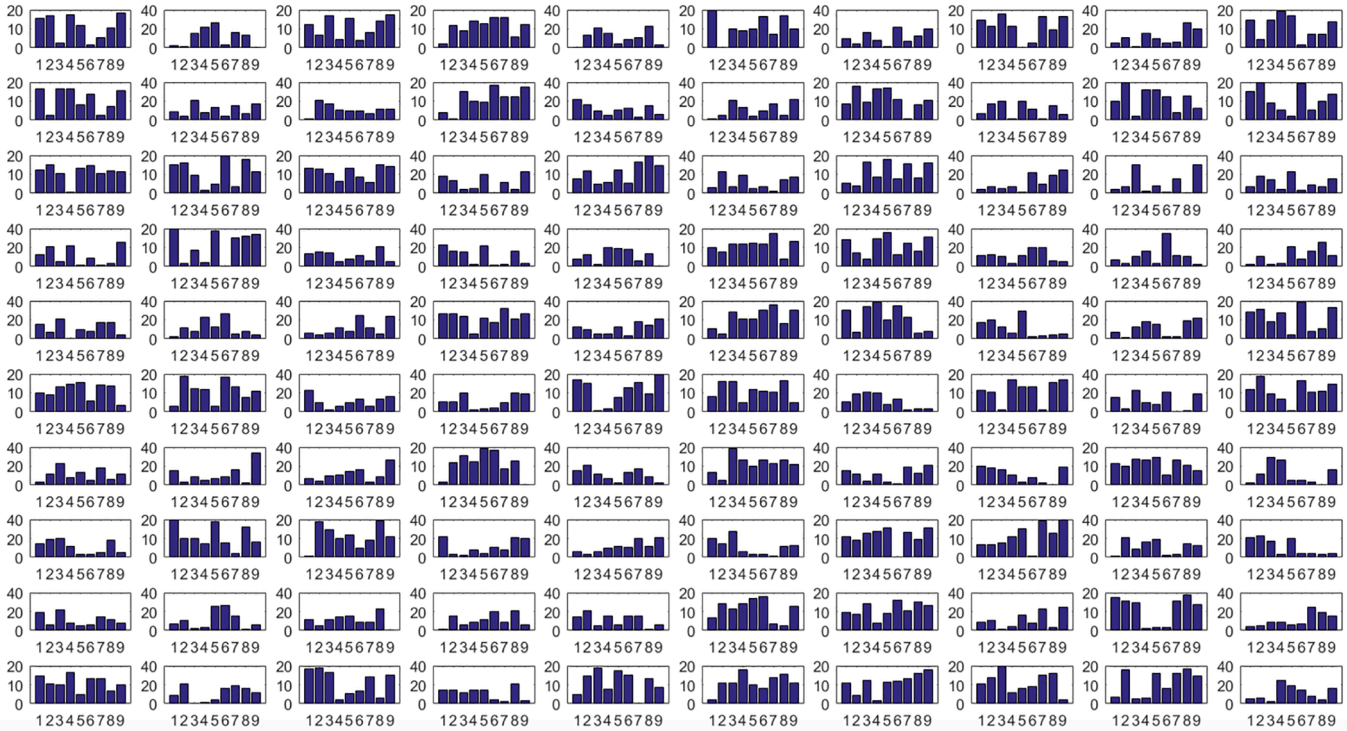

Fig. 8. The 100 randomly generated initial guesses for optimization.

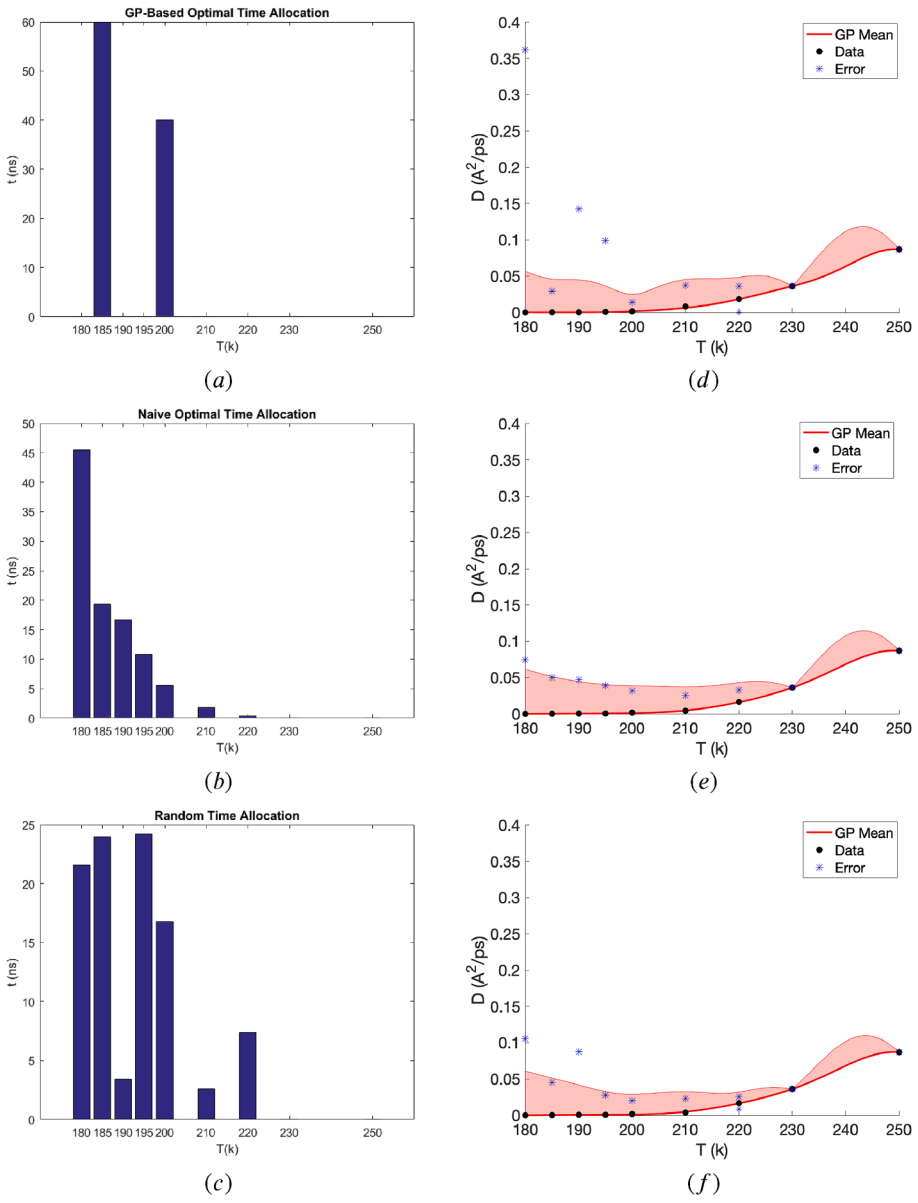

Fig. 9. The obtained time allocations from: (a) GP-based Optimization, (b) Naive Optimization, and (c) Random Generation; and the Gaussian process fits with 95% CI regions from: (d) GP-based Optimization, (e) Naive Optimization, and (f) Random Generation.

Table 5
Diffusivities (in $Å^2/ps$) at Various Temperatures with Optimally Allocated Simulation Time.

$T(K)$	GP-based	Naive	Random
250	–	–	–
230	–	–	–
220	–	1.64e−2	1.674e−2
210	–	4.26e−3	3.66e−3
200	1.519e−3	1.783e−3	1.654e−3
195	–	5.67e−4	6.4e−4
190	–	5.27e−4	6.35e−4
185	2.65e−4	2.96e−4	2.93e−4
180	–	5.253e−5	6.891e−5

the lower temperature region, and the closest estimations for VF parameters are also slightly different from the true data-fit VF parameters. To have a possible better approximation of temperature-diffusion curve and consequently more accurate estimation of VF model parameters, one may increase the total simulation time C to decrease the simulation error when additional computational resource becomes available.

6. Summary and conclusion

In this work, we propose a Gaussian process-based numerical optimization framework for optimal time allocation over simulations at different locations, so that a surrogate model with uncertainty estimation can be constructed to approximate the full simulation with a fixed total simulation time. Specifically, the L_2 norm of the (continuous)

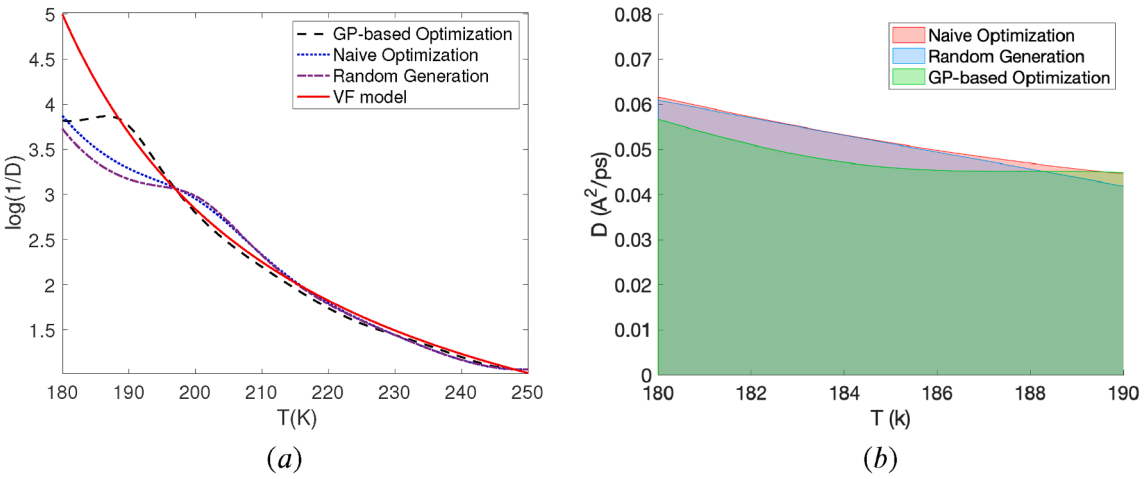

Fig. 10. (a) Comparison of Gaussian process mean curves to VF model in log-scale; (b) Comparison of the 95% CI regions at the range of lower temperatures.

Table 6
L_2 norms of errors in mean curves and L_2 norm of variance.

	L_2 of error in mean curve	L_2 of variance
GP-based Optimization	2.9e−02	3.6e−03
Naive Optimization	4.0e−02	3.7e−03
Random Generation	4.0e−02	3.1e−03

Table 7
VF parameters

Methods	α	β	T_0
GP-based Optimization	−1.149	250	135.3
Naive Optimization	−1.015	250	131.9
Random Generation	−0.9586	250	129.9
Data-fit VF	−0.9936	212.5	144.5

variance of the Gaussian process is minimized with respect to the cost (simulation time) distribution over the simulations at discrete locations.

The GP-based optimal time allocation framework is demonstrated using two synthetic numerical examples. Compared to a naive (intuitive) optimization setup, where the L_2 norm of the (discrete) errors defined at the fixed simulation locations is minimized, our proposed framework produces a more accurate mean function with less variance.

Despite its strong predictive power, MD simulations can be computationally expensive. With a fixed total simulation time, the GP-based optimal time allocation framework is applied to MD simulations for a glass-forming system with divergent dynamic relaxations to construct an accurate but cheaper surrogate model, which maps the temperature to diffusion coefficients. Specifically, the proposed framework provides guidance on how long to run MD simulations at predefined temperatures so that the variance of the obtained Gaussian process surrogate is minimized. Compared to both the Naive Optimization framework and a randomly assigned time allocation (or distribution), our GP-based optimal time allocation framework produces a mean function closest to the Vogel-Fulcher model fitted from converged data (considered as gold standard) and the best estimation of T_0.

7. Data availability

The raw data required to reproduce these findings are provided in Tables 3 and 4. The processed data required to reproduce these findings are provided in Table 5.

CRediT authorship contribution statement

John Chilleri: Software, Formal analysis, Validation, Writing - original draft. **Yanyan He:** Supervision, Methodology, Formal analysis, Writing - review & editing. **Dmitry Bedrov:** Investigation, Conceptualization, Writing - review & editing. **Robert M. Kirby:** Conceptualization, Methodology, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-12-2-0023. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Y. He, M. Razi, C. Forestiere, L.D. Negro, R. Kirby, Uncertainty quantification guided robust design for nanoparticles morphology, Comput. Methods Appl. Mech. Eng. 336 (2018) 578–593.
[2] J.P.C. Kleijnen, Regression and kriging metamodels with their experimental designs in simulation: A review, Eur. J. Oper. Res. 256 (1) (2017) 1–16.
[3] M.D. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, Cambridge, UK, 2003.
[4] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, The MIT Press, Cambridge, Massachusetts, 2006.

[5] T. Qin, K. Wu, D. Xiu, Data driven governing equations approximation using deep neural networks, J. Comput. Phys. 395 (2019) 620–635.
[6] A. Narayan, J.D. Jakeman, Adaptive leja sparse grid constructions for stochastic collocation and high-dimensional approximation, SIAM J. Sci. Comput. 36 (6) (2014) A2952–A2983.
[7] Y. Shin, D. Xiu, Nonadaptive quasi-optimal points selection for least squares linear regression, SIAM J. Sci. Comput. 38 (1) (2016) A385–A411.
[8] H. Liu, J. Cai, Y. Ong, An adaptive sampling approach for kriging metamodeling by maximizing expected prediction error, Comput. Chem. Eng. 106 (2017) 171–182.
[9] J.R. Dalbey, Efficient and robust gradient enhanced kriging emulators, Tech. rep., Sandia National Laboratories, Albuquerque, NM, 2013.
[10] D.R. Jones, A taxonomy of global optimization methods based on response surfaces, J. Global Optim. 21 (4) (2001) 345–383.
[11] D. Perez, B.P. Uberuaga, Y. Shim, J.G. Amar, A.F. Voter, Accelerated molecular dynamics methods: introduction and recent developments, Ann. Rep. Comput. Chem. 5 (2009) 79–98.
[12] A.F. Voter, A method for accelerating the molecular dynamics simulation of infrequent events, J. Chem. Phys. 106 (11) (1997) 4665–4677.
[13] D. Hamelberg, J. Mongan, J.A. McCammon, Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules, J. Chem. Phys. 120 (24) (2004) 79–98.
[14] A.F. Voter, Parallel replica method for dynamics of infrequent events, Phys. Rev. B 57 (1998) R13985–R13988.
[15] A.F. Voter, Hyperdynamics: accelerated molecular dynamics of infrequent events, Phys. Rev. Lett. 78 (1997) 3908–3911.
[16] M.R. Sorensen, A.F. Voter, Temperature-accelerated dynamics for simulation of infrequent events, J. Chem. Phys. 112 (2000) 9599–9606.
[17] M. Razi, A. Narayan, R. Kirby, D. Bedrov, Fast predictive models based on multi-fidelity sampling of properties in molecular dynamics simulations, Comput. Mater. Sci. 152 (2018) 125–133.
[18] F. Rizzi, H.N. Najm, B.J. Debusschere, K. Sargsyan, M. Salloum, H. Adalsteinsson, O.M. Knio, Uncertainty quantification in md simulations. part i: Forward propagation, Multiscale Model. Simul. 10 (4) (2012) 1428–1459.
[19] F. Rizzi, H.N. Najm, B.J. Debusschere, K. Sargsyan, M. Salloum, H. Adalsteinsson, O.M. Knio, Uncertainty quantification in md simulations. part ii: Bayesian inference of force-field parameters, Multiscale Model. Simul. 10 (4) (2012) 1460–1492.
[20] P. Angelikopoulos, C. Papadimitriou, P. Koumoutsakos, Pi4u, A high performance computing framework for bayesian uncertainty quantification of complex models, J. Comput. Phys. 284 (2012), 144103.
[21] P.E. Hadjidoukas, P. Angelikopoulos, C. Papadimitriou, P. Koumoutsakos, Pi4u: A high performance computing framework for bayesian uncertainty quantification of complex models, J. Comput. Phys. 284 (2015) 1–21.
[22] S.T. Reeve, A. Strachan, Error correction in multi-fidelity molecular dynamics simulations using functional uncertainty quantification, J. Comput. Phys. 334 (2017) 207–220.
[23] W. Kob, Computer Simulations of Supercooled Liquids, in: Vol. 704 of Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology. Lecture Notes in Physics, vol. 2, Springer, Berlin, Heidelberg, 2006.
[24] W. Kob, H.C. Andersen, Scaling behavior in the β-relaxation regime of a supercooled Lennard–Jones mixture, Phys. Rev. Lett. 73 (10) (1994) 1376–1379.
[25] Z. Chen, W. Qo, R.K. Bowles, Glass forming phase diagram and local structure of kob–andersen binary Lennard–Jones nanoparticles, J. Chem. Phys. 149 (2018), 094502.
[26] L.C. Valdes, F. Affouard, M. Descamps, J. Habasaki, Mixing effects in glass-forming Lennard-Jones mixtures, J. Chem. Phys. 130 (15) (2009), 154505.
[27] H. Vogel, The law of relation between the viscosity of liquids and the temperature, Phys. Z. 22 (1921) 645–646.
[28] G.S. Fulcher, Analysis of recent measurements of viscosity of glasses, J. Amer. Ceram. Soc. 8 (1925) 339–355.
[29] G. Tammann, W. Hesse, The dependence of viscosity upon the temperature of supercooled liquids, Z. Anorg. Allg. Chem. 156 (1926) 245–257.
[30] C. Kim, O. Borodin, G.E. Karniadakis, Nquantification of sampling uncertainty for molecular dynamics simulation: Time-dependent diffusion coefficient in simple fluids, J. Comput. Phys. 38 (1) (2015) 485–508.
[31] A. Pensoneault, X. Yang, X. Zhu, Nonnegativity-enforced gaussian process regression, Theor. Appl. Mech. Lett. 10 (2) (2020) 182–187.