Machine learning-assisted approximation of symmetrized quantum time correlation functions

Arif Ullah*

State Key Laboratory of Physical Chemistry of Solid Surfaces, Fujian Provincial Key Laboratory of Theoretical and Computational Chemistry, Department of Chemistry, and College of Chemistry and Chemical Engineering, Xiamen University, Xiamen 361005, China

E-mail: ua2024@xmu.edu.cn

Abstract

Open-chain imaginary-time path-integral sampling approach known with the acronym OPSCF (J. Chem. Phys. 148, 102340 (2018)) is an approach to the calculation of approximate symmetrized quantum time correlation functions. In OPSCF approach, the real time \( t \) is treated as a parameter, and therefore for each real time \( t \), a separate simulation on the imaginary time axis is needed to be run, which makes the OPSCF approach quite expensive and as a result, the approach loses the advantage of being a standard path-integral sampling approach. In this study, I propose that the use of OPSCF approach in combination with machine learning can reduce the computational cost by 80% to 90% (depending on the problem at hand). Combining OPSCF approach with ML is very straightforward which gives an upper hand to OPSCF approach over the trajectory-based methods such as the centroid molecular dynamics (CMD) and the ring-polymer molecular dynamics (RPMD).
Introduction

The study of quantum time correlation functions (TCFs) is important because it connects experimental measurements with quantum theory. For instance, velocity correlation function can be used to calculate the diffusion coefficient of a molecule in a liquid. Similarly, reaction rates and infrared absorption spectrum of a liquid can be calculated from the correlation functions of flux and dipole moment, respectively. The most famous approach to calculate exact quantum TCFs is the path integral formulation but the highly oscillatory phase factor (in other words the sign problem) in the Heisenberg picture makes it extremely hard to converge (see Eq. (14)), hence diverging the efforts to the development of methods approximating the TCFs. From these methods, the centroid molecular dynamics (CMD) and the ring-polymer molecular dynamics (RPMD) are well-known. CMD and PIMD both are imaginary time methods based on path integral formalism approximating Kubo-transformed correlation function. The main difference between CMD and RPMD is the choice of fictitious dynamical masses. In the CMD approach, in contrast to RPMD, much smaller bead masses are used except for the centroid. In addition, when it comes to calculating spectra, CMD is preferred over RPMD, however, RPMD is computationally less demanding than CMD.

In addition to CMD and RPMD methods, there are semi-classical approaches that approximate the problematic phase factor via linearization techniques. In most of these methods, the phase factor is approximated by using a transformation to sum and difference variables, and then linearization is carried out with respect to the difference variables. Inspired by the same idea, recently Cendagorta et al. presented an open-chain imaginary-time path-integral sampling approach (with the acronym OPSCF, which stands for open path symmetrized correlation function) for the calculation of symmetrized quantum time correlation functions. The OPSCF approach utilizes the same idea of transformation to the sum and difference variables (see Eq. (25)) along the discretized complex time paths. With this transformation, a ring polymer of 2P beads transferred is into an open chain of P+1.
beads. After transformation, the potential is expanded as a power series in the difference variables, and the resulting integrals are then performed analytically. In OPSCF approach, phase information is included up to second order and this approximation can be improved up to the desired order. In contrast to the expensive trajectory-based methods (such as CMD and RPMD) OPSCF is an imaginary time path integral approach and can be directly evaluated with the typical sampling schemes in particular with Monte Carlo, hybrid Monte Carlo, or any other path integral sampling scheme. However, a drawback of OPSCF approach is that the real time $t$ is treated as a parameter, and therefore for each real time $t$, a separate simulation is needed to be run regardless of the sampling scheme, which makes the OPSCF approach quite expensive and the approach loses the advantage of being a standard path-integral sampling approach.

In the past decade, with the increase in computational resources and the availability of large data, machine learning (ML) has revolutionized the field of science. In specific, ML has been used to construct the potential energy surfaces (PESs), to predict molecules based on properties, to learn coarse-grained force fields, atomic partial charges, dielectric constants in crystals, absorption cross-sections, and do excited state dynamics. In addition, ML is employed to study quantum dissipative dynamics of open quantum systems including my recent work on spin-boson model and natural harvesting systems. The use of ML is not new to molecular dynamics and has been utilized to construct the PESs while calculating the thermal rate coefficients. Here in this study, I borrow the idea of constructing PES with ML and use it to minimize the computational cost of OPSCF method. As in OPSCF method, for each real time $t$, a separate simulation is needed, I propose that with the use of ML, the whole symmetrized quantum time correlation function $\tilde{c}_{AB}(t)$ can be approximated by calculating $\tilde{c}_{AB}(t)$ only for a few time-steps, hence avoiding the expensive calculation of $\tilde{c}_{AB}(t)$ for each time-step. In this study, I utilize ML based non-parametric kernel ridge regression (KRR) method, saving computational cost from 75% to 95%, depending on the problem at hand. The use of OPSCF approach in combination with ML
gives an upper hand to OPSCF over the trajectory-based methods.

The remainder of this paper is organized as follows. In the Theory section, I briefly revise the theory of quantum TCFs, the Feynman path integral approach, and the OPSCF method. In addition, I also present the theory of kernel ridge regression model which is later used for speedy calculation of quantum TCFs. In the Computational details section, all the details regarding the preparation of input data and training architecture are given. The next section is Results and Discussion which is followed by Concluding Remarks.

Theory

Time correlation function

A general quantum correlation function for two operators $\hat{A}$ and $\hat{B}$ can be written as

$$c_{AB}(t) = \frac{1}{Z} \text{Tr} \left[ e^{-\beta \hat{H}} \hat{A}(0) \hat{B}(t) \right], \tag{1}$$

where $Z$ is the canonical partition function, $\hat{H}$ is Hamiltonian, $\beta = 1/k_B T$ and

$$\hat{B}(t) = e^{i\hat{H}t/\hbar} \hat{B}(0) e^{-i\hat{H}t/\hbar}. \tag{2}$$

$\hat{A}$ and $\hat{B}$ are operators of interest such as velocity operator $\hat{v}$ or dipole moment operator $\hat{\mu}$.

As Eq. (1) is a complex function of time, the direct evaluation with path integral formalism results in no positive definite weights which is because of real-time propagators with purely imaginary phases, and as a result importance sampling techniques cannot be used to evaluate the path integrals. An alternative to Eq. (1) is the symmetrized correlation function

$$\tilde{c}_{AB}(t) = \frac{1}{Z} \text{Tr} \left[ \hat{A}(0) e^{\frac{i}{\hbar} \hat{H} \tau_c} \hat{B}(0) e^{-\frac{i}{\hbar} \hat{H} \tau_c} \right], \tag{3}$$

where $\tau_c$ is the correlation time.
where $t_c = t - i\beta \hbar/2$ is a complex time. In contrast to Eq. (1), Eq. (3) is a real function as the two elements are complex conjugate of each other. The practical advantage of Eq. (3) is that in the path integral formulation, now we do have positive definite weights and importance sampling techniques can be used to evaluate the path integrals.\textsuperscript{52} Another symmetric and the most classical alternative to Eq. (1) is the Kubo-transformed correlation function\textsuperscript{12}

$$
\tilde{c}_{AB}(t) = \frac{1}{\beta Z} \int_0^\beta d\lambda \text{Tr} \left[ e^{-(\beta-\lambda)\hat{H}} \hat{A}(0) e^{-\lambda \hat{H}} \hat{B}(t) \right].
$$

(4)

Kubo-transformed correlation function has many advantages. It possesses the same symmetry properties as a classical correlation function.\textsuperscript{10,54} In addition, it plays the same role in quantum linear response theory as classical correlation function plays in classical linear response theory\textsuperscript{11} and if we Kubo-transform the position and velocity autocorrelation functions, they become identical to the corresponding classical autocorrelation functions for a harmonic oscillator.\textsuperscript{55} Because of these advantages, in most cases, the symmetrized correlation function Eq. (3) or Kubo-transformed correlation function Eq. (4) is evaluated which later can be transformed into Eq. (1) via the following relations in Fourier space

$$
c_{AB}(\omega) = e^{\beta \hbar \omega/2} \tilde{c}_{AB}(\omega), \quad \bar{c}_{AB}(\omega) = \frac{\beta \hbar \omega}{1 - e^{-\beta \hbar \omega}} \bar{c}_{AB}(\omega),
$$

(5)

where

$$
c_{AB}(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} c_{AB}(t) dt,
\tilde{c}_{AB}(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} \tilde{c}_{AB}(t) dt,
\bar{c}_{AB}(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} \bar{c}_{AB}(t) dt.
$$

(6)

For $\omega = 0$, all of them are identical. Using Eq. (5), knowing one of the correlation functions is the same to know all of them. As a result, we have the freedom to evaluate the numerically
feasible correlation functions (Eq. (3) and Eq. (4)) than the standard one Eq. (1). Methods like CMD and RPMD approximates the Kubo-transformed correlation function while standard path-integral sampling approaches such as OPSCF approximates symmetrized correlation function which is the concern of this article.

The Feynman path integral approach

The Feynman path integral approach is a classical description of quantum mechanics. In the Feynman path approach, all possible paths are considered between an initial point \( a \) and a final point \( b \). The sum of all paths gives the quantum mechanical amplitude \( K(b, t_b; a, t_a) \) (called as kernel or propagator), and the square of \( K(b, t_b; a, t_a) \) gives us the probability of propagation from point \( a \) to point \( b \) in time \( t = t_b - t_a \). For one dimensional case, \( K(b, t_b; a, t_a) \) can be expressed as

\[
K(b, t_b; a, t_a) = \int_a^b e^{i S(b, t_b; a, t_a)/\hbar} Dx(t),
\]

where \( D \) shows that all the paths are considered. The \( S(b, t_b; a, t_a) \) is the classical action

\[
S(b, t_b; a, t_a) = \int_{t_a}^{t_b} L(x, \dot{x}, t) dt,
\]

where \( L \) is Lagrangian and \( \dot{x} \) shows derivative with respect to time. In theory, the sum in the Feynman path integral approach is over an infinite number of paths, albeit only a subset of paths should be contributing the most to the sum. Still, analytically or numerically, the exact evaluation of the path integral is rarely possible. However, the interesting aspect of the path integral approach is the classical limit which can be obtained by letting \( \hbar \to 0 \). In the classical limit, only paths obeying classical mechanics survive. This nice feature makes the path integral approach suitable for finding approximate ways of performing quantum dynamics. In position eigen-state basis, the \( K(b, t_b; a, t_a) \) is written as

\[
K(x, x'; t) = \langle x | \hat{U}(t) | x' \rangle,
\]
where \( \hat{U}(t) \) is the real-time evolution operator expressed as

\[
U(t) = e^{-i\hat{H}t/\hbar}.
\]  \hspace{1cm} (10)

We can split \( \hat{U}(t) \) into \( P \) equal time slices using Suzuki–Trotter expansion

\[
\hat{U}(t) = \prod_{i=1}^{P} \hat{U}(\Delta t), \quad \Delta t = \frac{t}{P};
\]  \hspace{1cm} (11)

where for short time interval \( \Delta t \), the \( \hat{U}(t) \) in Eq. (10) is approximated as

\[
\hat{U}(\Delta t) \approx e^{-i\hat{T}t/\hbar} + e^{-i\hat{V}t/\hbar},
\]  \hspace{1cm} (12)

where \( \hat{T} \) and \( \hat{V} \) are the kinetic and potential operators, respectively. The above splitting in Eq. (12) is an approximation as generally \( \hat{T} \) and \( \hat{V} \) do not commute, however, this approximation becomes more accurate for very large \( P \). Manipulating only the kinetic part of Eq. (12), we get

\[
\langle x_{j+1} | \hat{U}(\Delta t) | x_j \rangle = \frac{1}{2\pi \hbar} \int dp_j \exp \left\{ \frac{i}{\hbar} \left[ -\Delta t \left( \frac{p_j^2}{2m} + V(x_j) \right) + p_j (x_{j+1} - x_j) \right] \right\},
\]  \hspace{1cm} (13)

where \( V(x_j) \) is the eigen value of operator \( \hat{V} \) at \( x_j \). Above expression has been obtained by writing the kinetic part as \( \langle x_{j+1} | \exp \{ -i\hat{p}^2 t/(2m\hbar) \} | x_j \rangle \), and by inserting the following identity twice \( I = \int \frac{dp}{2\pi \hbar} |p\rangle \langle p| \). In addition, we also utilize the following relations \( \langle x | p \rangle = \exp \{ ipx/\hbar \} \) and \( \langle p | p' \rangle = 2\pi \hbar \delta(p - p') \). Writing kinetic part for each time slice in Eq. (11) same as Eq. (13), we get

\[
\langle x_{P+1} | \hat{U}(t) | x_1 \rangle = \langle x_{i+1} | \prod_{i=1}^{P} \hat{U}(\Delta t) | x_i \rangle = \left( \frac{1}{2\pi \hbar} \right)^P \int dp_1 \int dx_2 dp_2 \ldots \int dx_P dp_P
\]

\[
\times \exp \left\{ \frac{i}{\hbar} \sum_{j=1}^{P} \left[ -\Delta t \left( \frac{p_j^2}{2m} + V(x_j) \right) + p_j (x_{j+1} - x_j) \right] \right\}.
\]  \hspace{1cm} (14)
In the above equation, the exponential term is purely imaginary and the highly oscillatory phase makes the evaluation of the integral very difficult, leading to the famous sign problem as have already been mentioned in the Introduction section. In real applications, the sign problem is avoided by using imaginary time. Let’s consider the partition function

\[
Z = \text{Tr} \left[ \exp\{-\beta \hat{H}\} \right] = \langle x | \exp\{-\beta \hat{H}\} | x \rangle. \tag{15}
\]

By setting \( it/\hbar = \beta \) in Eq. (10) which leads to Eq. (14) in imaginary time

\[
\langle x_{P+1}|\hat{U}(t)|x_1 \rangle = \langle x_{i+1} \left| \prod_{i=1}^{P} \hat{U} (\Delta t) \right| x_i \rangle = \left( \frac{1}{2\pi \hbar} \right)^P \int dp_1 \int dx_2 dp_2 \ldots \int dx_P dp_P \times \exp \left\{ \sum_{j=1}^{P} \left[ -\frac{\beta}{P} \left( \frac{p_j^2}{2m} + V(x_j) \right) + p_j (x_{j+1} - x_j) \right] \right\}. \tag{16}
\]

Integrating out the momentum variables leads to

\[
Z_P = \left( \frac{mP}{2\pi \hbar^2 \beta} \right)^{P/2} \int dx_1 \ldots dx_P e^{-\beta V_{\text{eff}}(x_1 \ldots x_P)}, \tag{17}
\]

where \( q_t = q(t\beta \hbar/P) \) and the effective potential \( V_{\text{eff}} \) is

\[
V_{\text{eff}}(x_1 \ldots x_P) = \sum_{j=1}^{P} \left[ \frac{mP}{2\hbar^2 \beta^2} (x_{j+1} - x_j)^2 + \frac{V(x_j)}{P} \right], \tag{18}
\]

which converges to the exact result for \( P \to \infty \). However for a finite \( P \), Eq. (17) is isomorphic to the partition function of a classical system of \( P \) beads.\(^{58}\) The coordinates \( x_1, x_2, \ldots, x_P \) represent the position of the beads where \( x_P = x_1 \) form a ring polymer or necklace. Every bead in the ring polymer is connected to only its nearest neighbors through a harmonic potential with the force constant \( \frac{mP}{\hbar^2 \beta^2} \). For a position-dependent operator \( \hat{A} \), the average
value can be obtained as

\[ \langle \hat{A} \rangle = \frac{\int dx_1 \int dx_2 \ldots \int dx_P A(x_1) \exp(-\beta V_{\text{eff}})}{\int dx_1 \int dx_2 \ldots \int dx_P \exp(-\beta V_{\text{eff}})} . \]  

(19)

To evaluate the multidimensional integrals in Eq. (19), the path integral Monte Carlo (PIMC) method\textsuperscript{59–61} can be used. An alternative is the path integral molecular dynamics (PIMD) method.\textsuperscript{62–64} In PIMD, momenta are added to the effective potential which allows performing molecular dynamics. In one dimension, for \( P \) beads, the PIMD Hamiltonian can be written as

\[ H_P(p, x) = \sum_{j=1}^{P} \frac{p_j^2}{2m} + V_{\text{eff}} = \sum_{j=1}^{P} \left[ \frac{p_j^2}{2m} + \frac{mP}{2\hbar^2 \beta^2} (x_{j+1} - x_j)^2 + \frac{V(x_j)}{P} \right] . \]  

(20)

The masses are fictitious and can have arbitrary values as the static properties do not depend on them. Usually, in PIMD calculations, a thermostat is added to control the temperature and to ensure proper canonical distribution of the path integral beads.\textsuperscript{65}

**Open-chain imaginary-time path integral approach**

The theory of open-chain imaginary-time path integral approach for symmetrized correlation function with acronym OPSCF is well-explained in Ref.\textsuperscript{28} but to self-contained this article, I outline it here. I begin with the symmetrized correlation function Eq. (3) with the consideration that both operators \( \hat{A} \) and \( \hat{B} \) are only position-dependent. In the position basis, Eq. (3) is expressed as

\[ \bar{c}_{AB}(t) = \frac{1}{Z} \int dx dx' \langle x \left| e^{i\hat{H}^c t / \hbar} \right| x' \rangle b(x') \langle x' \left| e^{-i\hat{H}^c t / \hbar} \right| x \rangle a(x) , \]  

(21)

where \( \hat{H} = \hat{p}^2 / 2m + V(\hat{x}) \) and \( a(x) \) and \( b(x') \) are the eigenvalues of the corresponding operators at \( x \) and \( x' \). Using the same strategy as for Eq. (15) which leads to Eq. (17), I get
the following path-integral representation for $\bar{c}_{AB}(t)$

$$
\bar{c}_{AB,P}(t) = \frac{1}{Z} \int d\mathbf{x}_1 \ldots d\mathbf{x}_{2P} a(\mathbf{x}_1) b(\mathbf{x}_{P+1}) \rho(\mathbf{x}_1 \ldots \mathbf{x}_{2P}) e^{iV_{\text{eff}}(\mathbf{x}_1 \ldots \mathbf{x}_{2P})},
$$

(22)

with $\rho(\mathbf{x}_1 \ldots \mathbf{x}_{2P})$ as a positive-definite distribution

$$
\rho(\mathbf{x}_1 \ldots \mathbf{x}_{2P}) = \left(\frac{mP}{2\pi|\tau_c|\hbar}\right)^P \exp\left\{-\sum_{k=1}^{2P} \left[\frac{mP\beta}{4|\tau_c|^2} (x_{k+1} - x_k)^2 + \frac{\beta}{2P} V(x_k)\right]\right\},
$$

(23)

and the $V_{\text{eff}}(\mathbf{x}_1 \ldots \mathbf{x}_{2P})$ as a phase factor

$$
V_{\text{eff}}(\mathbf{x}_1 \ldots \mathbf{x}_{2P}) = \frac{mPt}{2\hbar|\tau_c|^2} \left[ \sum_{k=1}^{P} (x_{k+1} - x_k)^2 - \sum_{k=P+1}^{2P} (x_{k+1} - x_k)^2 \right] - \frac{t}{P\hbar} \left[ \sum_{k=2}^{P} V(x_k) - \sum_{k=P+2}^{2P} V(x_k) \right].
$$

(24)

As Eq. (21) contains two propagators, thus we have $2P$ beads in the ring polymer chain ($2P+1 = \mathbf{x}_1$). Eq. (22) suffers from the same sign problem because of the imaginary phase. As positive-definite distribution and phase factor carry the same set of parameters, in OPSCF approach, the phase factor is suppressed by introducing the following transformation of variables

$$
u_1 = x_1
$$

$$
u_{P+1} = x_{P+1},
$$

$$
u_i = \frac{x_i + x_{2(P+1) - i}}{2},
$$

$$
u_i = x_i - x_{2(P+1) - i},
$$

(25)

where $i = 2, \ldots, P$ and $u_i$ and $v_i$ are the so-called path average and difference variables, respectively. With the above transformation, the ring of $2P$ is transformed into an open chain of $P+1$ beads. In terms of transformed variables, the positive-definite distribution
and the phase factor can be written as

\[
\rho(u_1 \ldots u_{P+1}, v_2 \ldots v_P) = \left(\frac{mP}{2\pi |\tau_c| \hbar}\right)^P \exp \left\{- \left[ \frac{mP\beta}{4 |\tau_c|^2} \left( \sum_{i=1}^{P} (u_{i+1} - u_i)^2 \right) + \frac{1}{2} \sum_{i=2}^{P-1} (v_{i+1} - v_i)^2 + \frac{1}{2} \left( v_2^2 + v_P^2 \right) \right] \right\} \\
\times \exp \left\{ - \frac{\beta}{2P} \sum_{i=1}^{P} \left( \sum_{i=2}^{P} \left[ 2V(u_i) + \frac{1}{4} V''(u_i) v_i^2 \right] \right) + V(u_1) + V(u_{P+1}) \right\},
\]

(26)

\[
V_{\text{eff}}(u_1 \ldots u_{P+1}, v_2 \ldots v_P) = \frac{mPt}{2|\tau_c| \hbar^2} \left[ \sum_{i=2}^{P-1} (u_{i+1} - u_i)(v_{i+1} - v_i) \right. \\
+ \left. 2v_2 (u_2 - u_1) - 2v_P (u_{P+1} - u_P) - \frac{t}{P\hbar} \sum_{i=2}^{P} V'(u_i) v_i \right],
\]

(27)

where the \(V'(u_i)\) and \(V''(u_i)\) denote the single and double derivatives with respect to average variable \(u_i\). For rigorous derivation of Eq. (26) and Eq. (27), readers are referred to Ref. 28.

Eq. (26) can be separated into two terms on the basis of variables

\[
\rho(u_1 \ldots u_{P+1}, v_2 \ldots v_P) = \rho_1(u_1 \ldots u_{P+1}) \rho_2(u_2 \ldots u_P, v_2 \ldots v_P),
\]

(28)

where

\[
\rho_1(u_1 \ldots u_{P+1}) = \left(\frac{mP}{2\pi |\tau_c| \hbar}\right)^P \exp \left\{- \left[ \frac{mP\beta}{4 |\tau_c|^2} \left( \sum_{i=1}^{P} (u_{i+1} - u_i)^2 \right) \right] \right\} \\
\times \exp \left\{ - \frac{\beta}{2P} \sum_{i=2}^{P} \left[ 2V(u_i) + V(u_1) + V(u_{P+1}) \right] \right\},
\]

(29)
\[ \rho_2(u_2 \ldots u_P, v_2 \ldots v_P) = \exp \left\{ - \left[ \frac{mP\beta}{4|\tau_c|^2} \left( \frac{1}{2} \sum_{i=2}^{P-1} (v_{i+1} - v_i)^2 + \frac{1}{2} (v_2^2 + v_P^2) \right) \right] \right\} \times \exp \left\{ - \frac{\beta}{2P} \sum_{i=2}^{P} \left( \frac{1}{4} V''(u_i) v_i^2 \right) \right\} \]  

(30)

Eq. (27) can be rewritten in the matrix form

\[ V_{\text{eff}}(u_1 \ldots u_{P+1}, v_2 \ldots v_P) = K(u') S, \]  

(31)

where \( K \) is a \((P - 1)\)-dimensional vector defined as

\[ K_i(u') = \gamma (2u_i - u_{i-1} - u_{i+1}) - \frac{t}{P\hbar} V'(u_i), \quad i = 2, \ldots, P \]  

(32)

where \( u' \equiv u_1, \ldots, u_{P+1} \), and

\[ \gamma = \frac{mPt}{|\tau_c|^2 \hbar}. \]  

(33)

In Eq. (31), \( S = (v_2, \ldots, v_P) \) which is a \((P - 1)\)-dimensional vector. Similar to Eq. (27), Eq. (30) can also be expressed in the matrix form

\[ \rho_2(u_2 \ldots u_P, v_2 \ldots v_P) = e^{-\frac{1}{2} S^T M(u) S}, \]  

(34)

where \( M \) is a square tridiagonal \((P - 1) \times (P - 1)\) matrix

\[ M_{ij}(r) = \left[ 2Q + \frac{\beta}{4P} V''(u_i) \right] \delta_{ij} - A\delta_{i,j+1} - Q\delta_{i+1,j}, \]  

(35)

with \( u \equiv u_2, \ldots, u_P \) and

\[ Q = \frac{mP\beta}{4|\tau_c|^2}. \]  

(36)
The overall expression for $\bar{c}_{AB}(t)$ becomes

$$\bar{c}_{AB}(t) = \frac{1}{Z} \int du_1 \ldots du_{P+1} \rho_1 (u_1 \ldots u_{P+1}) a(u_1) b(u_{P+1}) \times \int dv_2 \ldots dv_P \exp \left\{ -\frac{1}{2} S^T M(u) S + i K(u') S \right\}. \tag{37}$$

The last integral in the above equation is Gaussian integral which can be solved analytically

$$\int dv_2 \ldots dv_P \exp \left\{ -\frac{1}{2} S^T M(u) S + i K(u') S \right\} = \left[ \frac{(2\pi)^{P-1}}{\det[M(u)]} \right]^{1/2} \exp \left\{ -\frac{1}{2} K(u')^T M^{-1}(u) K(u') \right\}. \tag{38}$$

Using above equation in Eq. (37), I get

$$\bar{c}_{AB}(t) = \frac{2\pi^{(P-1)/2}}{Z} \int du_1 \ldots du_{P+1} \rho_1 (u_1 \ldots u_{P+1}) a(u_1) b(u_{P+1}) \times \exp \left\{ -\frac{1}{2} \left( K^T(u') M^{-1}(u) K(u') + \ln \det[M(u)] \right) \right\}. \tag{39}$$

Eq. (39) is the final result of OPSCF approach which is just an imaginary path integral. The real $t$ is taken as an initial parameter, thus for each value of $t$, we need to run a separate simulation regardless of the sampling scheme.\textsuperscript{28} The typical sampling schemes are molecular dynamics,\textsuperscript{66} Monte Carlo,\textsuperscript{29} hybrid Monte Carlo,\textsuperscript{30} or any path integral sampling scheme.\textsuperscript{31,32}

**Kernel ridge regression**

The theory of kernel ridge regression (KRR) is well documented by many authors\textsuperscript{67–71} but for the sake of completeness, I outline it here briefly. For a vector of input values $\mathbf{r}$, KRR
approximates a function $f(r)$ which is defined as:

$$f(r) = \sum_{i=1}^{N_{tr}} \alpha_i k(r, r_i),$$

where $N_{tr}$ is the number of training points and $\alpha = \{\alpha_i\}$ is a vector of regression coefficients $\alpha_i$. The kernel function $k(r, r_i)$ takes two vectors $r$ and $r_i$ from the input space and measures the distance between them. The most common kernel function is the Matérn kernel:

$$k(r_i, r_j) = \exp \left( -\frac{\|r_i - r_j\|_2}{\sigma} \right) \sum_{k=0}^{n} \frac{(n+k)!}{(2n)!} \left( \frac{n}{k} \right) \left( \frac{2\|r_i - r_j\|_2}{\sigma} \right)^{n-k},$$

where $\sigma$ is the width of the kernel which is always positive and $n$ is an integer ($n \geq 0$). In Matérn kernel, the Euclidean distance between vectors $r$ and $r_i$ is taken as $L^2$ norm, i.e., $\|r - r_i\|_2$. For $n = 0$, the Matérn kernel function reduces to exponential kernel function:

$$k(r, r_i) = \exp \left( -\frac{\|r - r_i\|_2}{\sigma} \right).$$

Another property of Matérn kernel function is that it behaves very similar to Gaussian kernel function in the limit of $n \to \infty$:

$$k(r, r_i) = \exp \left( -\frac{\|r - r_i\|_2^2}{2\sigma^2} \right).$$

In both Gaussian and exponential kernel functions, we have only one hyperparameter $\sigma$ defining the length scale. Another kernel that uses $L^1$ or Manhattan norm $\|r - r_i\|_1$ is the Laplacian kernel function:

$$k(r, r_i) = \exp \left( -\frac{\|r - r_i\|_1}{\sigma} \right).$$
In my previous work I have used Gaussian kernel as implemented in the MLatom package and per my tests, it has a good performance for my application, hence in my current study, I also stick to the Gaussian kernel.

Intuitively, the Gaussian kernel measures similarity between the vectors \( \mathbf{r} \) and \( \mathbf{r}_i \). The output of these kernels increases as \( \| \mathbf{r} - \mathbf{r}_i \| \to 0 \) and becomes unity at \( \mathbf{r} = \mathbf{r}_i \) while for large distance \( \| \mathbf{r} - \mathbf{r}_i \| \to \infty \), the kernel \( k(\mathbf{r}, \mathbf{r}_i) \) tends to zero. After choosing the kernel function, we need to find the regression coefficients \( \alpha \) in Eq. (40). It is done by minimizing a squared error loss function:

\[
\min_\alpha \sum_{i} (f(\mathbf{r}_i) - y_i)^2 + \lambda \alpha^T \mathbf{K} \alpha,\quad (45)
\]

where \( y = \{y_i\} \) is the target output vector, \( \mathbf{K} \) is the kernel matrix and \( \lambda \) denotes a non-negative regularization hyperparameter. In Eq. (45), the second term is usually added to stop KRR model from giving too much weight to a single point. After simple algebra, Eq. (45) leads to:

\[
\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} y,\quad (46)
\]

where \( \mathbf{I} \) is the identity matrix and \( y \) is the vector of target values. Matrix inversion in Eq. (46) scale as \( O(N^3) \), therefore instead of solving Eq. (46), it is more efficient to solve a system of linear equations \((\mathbf{K} + \lambda \mathbf{I}) \alpha = y\).

**Computational details**

I consider two correlation functions as was done in Ref. 28; the position symmetrized autocorrelation function \( \bar{c}_{xx}(t) \) and the position-squared symmetrized autocorrelation function \( \bar{c}_{x^2x^2}(t) \). I consider four different potentials, e.g., harmonic potential, a mildly anharmonic potential, a quartic potential, and a symmetric double-well potential, which are respectively
expressed as

\[
V(x) = \frac{1}{2}x^2, \quad (47)
\]
\[
V(x) = \frac{1}{100}x^4 + \frac{1}{10}x^3 + \frac{1}{2}x^2, \quad (48)
\]
\[
V(x) = \frac{1}{4}x^4, \quad (49)
\]
\[
V(x) = \frac{1}{10} \left( x^2 - \frac{5}{2} \right). \quad (50)
\]

These potentials have been wildly used to benchmark various approximation methods for computing quantum time correlation functions.\textsuperscript{6,10,14,21,28,77,78} Just to emphasize, the applicability of the approach is not limited to these potentials and is equally applicable to any shape of potential such as the Silvera–Goldman pair potential\textsuperscript{79} used for the static equilibrium properties of liquid para-hydrogen.\textsuperscript{80} For my study, I take OPSCF data from Ref. 28. For training, I prepare my input data as a function of time \( f(t) \) and take \( \bar{c}_{AB}(t) \) as target values (see Fig. [1]). I train separate ML models for each case, therefore apart from the time feature, I do not need to include any other features such as temperature or number of beads in the input. As I just have a single feature, I use \( f(t) = t \). In the case of multi-features
input such as in Ref. 48, it is suggested to use logistic functions such as

\[ f(t) = \frac{a}{1 + b \cdot \exp\left\{-\frac{t-c}{d}\right\}}, \]  

(51)

where \( a, b, c, \) and \( d \) are constants. Logistic functions are helpful to normalize the input features and keep their values in the same range. Keeping all features in the same range improves the convergence of the ML model. Here in this study, I train KRR model for each potential using Gaussian kernel with \( \sigma = 1.0 \) and \( \lambda = 0.001 \). For ML calculations, I used the publicly available MLatom package.

Results and Discussion

Harmonic potential: As my first case study, I demonstrate the applicability of ML approach for the position symmetrized autocorrelation function \( \bar{c}_{xx}(t) \) with harmonic potential Eq. (47). I take OPSCF data from Ref. 28 for high temperature case \( \beta = 1 \). In Ref. 28 for each real time step, the OPSCF simulations were respectively run for minimum \( 10^7 \) steps. To get \( \bar{c}_{xx}(t) \) with a smooth behavior for the time-range \( t = 0-10 \) [a.u.], the OPSCF calculations should be carried out with the time-step not larger than \( dt = 0.1 \) making it 101 time-steps in total. However by combining ML, calculating \( \bar{c}_{xx}(t) \) values at \( t = 0, 1, 2, \ldots, 10 \) with OPSCF and then training a KRR model saves ca. 90% computational cost. The training of KRR model takes around two to three seconds. After training, the KRR model successfully interpolates between the training points and predicts \( \bar{c}_{xx}(t) \) for all the unknown time-steps as shown in Fig. 2(a). On a single Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz, the prediction takes ca. 2mins which is substantially fast keeping in mind the time-cost of OPSCF for each time-step.

Mildly harmonic potential: For mildly anharmonic potential Eq. (48), I train a KRR model to approximate the position-squared symmetrized autocorrelation function \( \bar{c}_{x^2 x^2}(t) \) for high temperature case \( \beta = 1 \). The position symmetrized autocorrelation function \( \bar{c}_{xx}(t) \)
Figure 2: (a) Symmetrized autocorrelation function for position $\bar{c}_{xx}(t)$ with harmonic potential; (b) position-squared symmetrized autocorrelation function $\bar{c}_{x^2x^2}(t)$ for mildly anharmonic potential. The OPSCF data was taken from Ref. 28.

Table 1: Mean absolute error (MAE), root mean square error (RMSE) and coefficient of determination $R^2$ for all potentials.

<table>
<thead>
<tr>
<th>Error type</th>
<th>harmonic</th>
<th>mildly anharmonic</th>
<th>quartic</th>
<th>symmetric double-well</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>0.008</td>
<td>0.045</td>
<td>0.005</td>
<td>0.010</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.010</td>
<td>0.070</td>
<td>0.008</td>
<td>0.013</td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.999</td>
<td>0.997</td>
<td>0.998</td>
<td>0.999</td>
</tr>
</tbody>
</table>

For mildly anharmonic potential is similar in behavior to Fig. 2(a), therefore I don’t consider it here. The $\bar{c}_{x^2x^2}(t)$ which has small and large peaks as shown in Fig. 2(b), is a challenging case for ML. By calculating $\bar{c}_{x^2x^2}(t)$ for only 20 time-steps $t = 0, 1, 2, \ldots, 20$ and then training KRR model on it, save ca. 90% computational cost. KRR model predicts $\bar{c}_{x^2x^2}(t)$ for the remaining time-steps by interpolation, and on a single Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz, the prediction of the whole time-range takes ca. 3 min only. Surprisingly, the KRR model was able to learn the small and large peaks very well as shown in Fig. 2(b).

**Quartic and symmetric double-well potentials:** For quartic and symmetric double-well potentials, I respectively calculate $\bar{c}_{xx}(t)$ and $\bar{c}_{x^2x^2}(t)$. For quartic potential, calculating $\bar{c}_{xx}(t)$ for time-steps $t = 0, 1, 2, \ldots, 20$ is enough for KRR model to capture the behavior of the $\bar{c}_{xx}(t)$ for the remaining time-steps (see Fig. 3(a)), and as a results saves ca. 90%
computational cost. For symmetric double-well potential case, calculation of \( \bar{c}_{x^2x^2}(t) \) for only seven time-steps \( t = 0, 0.5, 1, 1.5, \ldots, 3 \) is enough, and using KRR model to predict \( \bar{c}_{x^2x^2}(t) \) for the remaining time-steps saves ca. 75% computational cost (see Fig. 3(b)).

Table 1 shows mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination \( R^2 \) for all the considered potentials. As already have mentioned, \( \bar{c}_{x^2x^2}(t) \) for mildly anharmonic potential (see Fig. 2(b)) was slightly challenging for ML and therefore we see comparatively more error as compared to other potentials. The parity plots in Fig. 4 show that the deviation of the predicted values from the OPSCF values is very negligible. The goodness-of-fit is measured with the coefficient of determination \( R^2 \). The close to one \( R^2 \) values shown in Table 1 indicate a good fit of the KRR model to the OPSCF data.

### Concluding Remarks

In this paper, I have shown that the computational cost of the costly imaginary time sampling-based methods such as OPSCF can be significantly reduced by using ML. In imaginary time sampling-based methods where real time \( t \) is taken just as a parameter, for each
real time-step, a costly simulation is run which makes these methods quite expensive. In addition, the computational cost increases with the increase in number of beads, particularly at low temperatures. The use of these methods in combination with ML substantially scale down the computational cost by 75% to 95%. With the ability that we can directly combine OPSCF with ML, give superiority to OPSCF approach over the trajectory-based methods and encourage to develop the standard path-integral sampling approaches. Though here I have demonstrated the applicability of my approach just for OPSCF approach, but in reality, it is not limited to OPSCF, and can be extended to any sampling-based approach.

Figure 4: Parity plots present the distribution of OPSCF values vs. predicted values by KRR model for (a) $\tilde{c}_{xx}(t)$ with harmonic potential; (b) $\tilde{c}_{x^2x^2}(t)$ with mildly anharmonic potential; (c) $\tilde{c}_{xx}(t)$ with quartic potential and (d) $\tilde{c}_{x^2x^2}(t)$ with symmetric double-well potential.
Acknowledgement

AU acknowledge funding by the National Natural Science Foundation of China (No. 22003051) and via the Lab project of the State Key Laboratory of Physical Chemistry of Solid Surfaces.

Data and Code Availability

The data and code for this study are available from the author upon reasonable request.

References


(15) Hone, T. D.; Rossky, P. J.; Voth, G. A. A comparative study of imaginary time path
integral based methods for quantum dynamics. *The Journal of chemical physics* 2006,
124, 154103.

(16) Wang, H.; Sun, X.; Miller, W. H. Semiclassical approximations for the calculation
of thermal rate constants for chemical reactions in complex molecular systems. *The

(17) Shi, Q.; Geva, E. Nonradiative Electronic Relaxation Rate Constants from Approxima-
tions Based on Linearizing the Path-Integral Forward-Backward Action. *The Journal

(18) Liu, J.; Miller, W. H. Using the thermal Gaussian approximation for the Boltzmann op-
erator in semiclassical initial value time correlation functions. *The Journal of chemical
physics* 2006, 125, 224104.

(19) Liu, J.; Miller, W. H. Test of the consistency of various linearized semiclassical initial
value time correlation functions in application to inelastic neutron scattering from liquid

(20) Bonella, S.; Monteferrante, M.; Pierleoni, C.; Ciccotti, G. Path integral based calcu-
lations of symmetrized time correlation functions. II. *The Journal of chemical physics*
2010, 133, 164105.

(21) Monteferrante, M.; Bonella, S.; Ciccotti, G. Linearized symmetrized quantum time
correlation functions calculation via phase pre-averaging. *Molecular Physics* 2011, 109,
3015–3027.

(22) Huo, P.; Coker, D. F. Communication: Partial linearized density matrix dynamics for
dissipative, non-adiabatic quantum evolution. *The Journal of chemical physics* 2011,
135, 201101.


(48) Ullah, A.; Dral, P. O. Predicting the future of excitation energy transfer in light-harvesting complex with artificial intelligence-based quantum dynamics. [https://doi.org/10.26434/chemrxiv-2021-d2ksx](https://doi.org/10.26434/chemrxiv-2021-d2ksx).


