Supporting Information: Efficient discovery of multiple minimum action pathways using Gaussian process

JaeHwan Shim*[1], Juyong Lee†[2] and Jaejun Yu‡[1]

[1] *Center for Theoretical Physics, Department of Physics and Astronomy, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea*
[2] *College of Pharmacy, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea*

S1. Discretized formula for the derivatives of modified action at the time step n

Discretized formula for the classical action and energy restraint terms are

$$\frac{\partial S_{\mathrm{cls}}}{\partial \mathbf{x}^{(n)}} = dt \sum_{a=0}^{A-1} \left(-\frac{\partial}{\partial \mathbf{x}_a^{(n)}} V\left(\mathbf{x}^{(n)}\right) + m_a \mathbf{a}_a^{(n)} \right) \tag{S1}$$

$$\begin{aligned} &\frac{\partial}{\partial \mathbf{x}^{(n)}} \left(\mu_{\mathrm{E}} \sum_{n=0}^{N-1} \left(E^{(n)} - E_{\mathrm{t}} \right)^2 \right) \\ &\quad = 2\mu_{\mathrm{E}} dt \sum_{a=0}^{A-1} \left[\left(\frac{\partial}{\partial \mathbf{x}_a^{(m)}} V\left(\mathbf{x}^{(n)}\right) + \frac{1}{dt}\mathbf{p}_a^{(m)} \right) \left(-E_{\mathrm{t}} + V\left(\mathbf{x}^{(n)}\right) + \frac{1}{2} m_a \mathbf{v}_a^{(n)T} \cdot \mathbf{v}_a^{(n)} \right) \right. \\ &\quad \left. + \frac{1}{dt}\mathbf{p}_a^{(n-1)} \left(-E_{\mathrm{t}} + V\left(\mathbf{x}^{(n-1)}\right) + \frac{1}{2} m_a \mathbf{v}_a^{(n-1)T} \cdot \mathbf{v}_a^{(n-1)} \right) \right], \end{aligned} \tag{S2}$$

where $\mathbf{v}_a^{(n)} = (\mathbf{x}_a^{(n+1)} - \mathbf{x}_a^{(n)})/dt$, $\mathbf{a}_a^{(n)} = (2\mathbf{x}_a^{(n)} - \mathbf{x}_a^{(n-1)} - \mathbf{x}_a^{(n+1)})/dt^2$ and $\mathbf{p}_a^{(n)} = m_a \mathbf{v}_a^{(n)}$. Our parameter set for the classical action on the MB potential is $dt = 0.02$, $m_a = 1$, $\mu_{\mathrm{E}} = 0.1$, $E_{\mathrm{t}} = -0.45$ and $N = 300$.

*schinavro@snu.ac.kr
†nicole23@snu.ac.kr
‡jyu@snu.ac.kr

Discretized formula for the derivatives of OM action is

$$\begin{aligned}\frac{\partial S_{\mathrm{OM}}}{\partial \mathbf{x}^{(n)}} &= \frac{\gamma dt}{2}\mathbf{a}^{(n)} \\ &+ \frac{\partial \nabla V\left(\mathbf{x}^{(n)}\right)}{\partial \mathbf{x}^{(n)}}\left(\frac{dt}{2\gamma}\nabla V\left(\mathbf{x}^{(n)}\right) - \frac{dt^2}{4}\mathbf{a}^{(n)}\right) \\ &+ \frac{1}{4}\nabla V\left(\mathbf{x}^{(n-1)}\right) - \frac{1}{2}\nabla V\left(\mathbf{x}^{(n)}\right) + \frac{1}{4}\nabla V\left(\mathbf{x}^{(n+1)}\right), \end{aligned} \tag{S3}$$

where $\partial\nabla V\left(\mathbf{x}^{(n)}\right)/\partial\mathbf{x}^{(n)}$ is the Hessian.

S2. Details on Gaussian Process Regression

For an atomic system, the vector representation for the input coordinates of data is

$$\begin{aligned}\mathbf{x}^{(m)} = \Big(& x_1^{(m)}, x_2^{(m)}, \cdots, x_a^{(m)}, \cdots, x_A^{(m)}, \\ & y_1^{(m)}, y_2^{(m)}, \cdots, y_a^{(m)}, \cdots, y_A^{(m)}, \\ & z_1^{(m)}, z_2^{(m)}, \cdots, z_a^{(m)}, \cdots, z_A^{(m)}\Big), \end{aligned} \tag{S4}$$

where $x_a^{(m)}, y_a^{(m)}, z_a^{(m)}$ are the x, y, z coordinates of the a-th atom of the m-th image. For an arbitrary coordinate system with D dimensions, the representation vector for the system is simply

$$\mathbf{x}^{(m)} = \left(x_1^{(m)},\ x_2^{(m)}, \cdots, x_d^{(m)}, \cdots,\ x_D^{(m)}\right), \tag{S5}$$

where $x_d^{(m)}$ is the element of d-th coordinate at m-th image. For example, an atomic system (Eq. (S4)) has $D = 3A$ and Müller-Brown potential has $D = 2$.

The representation for the entire dataset of GP or pathways can be expressed in a matrix form by combining all the data into one where each datum vector is concatenated laterally. With the M number of evaluated data points and the N number of images, representation for data points and pathway $\mathbf{X}$ and $\mathbf{X}_*$ have the size $(D \times M)$, $(D \times N)$, respectively:

$$\mathbf{X} = \sum_{m=1}^{M} \mathbf{e}^{(m)} \otimes \mathbf{x}^{(m)} = \left(\mathbf{x}^{(1)}\ \ \mathbf{x}^{(2)} \cdots \mathbf{x}^{(M)}\right) \tag{S6}$$

$$\mathbf{X}_* = \sum_{n=1}^{N} \mathbf{e}^{(n)} \otimes \mathbf{x}^{(n)} = \left(\mathbf{x}^{(1)}\ \ \mathbf{x}^{(2)} \cdots \mathbf{x}^{(N)}\right). \tag{S7}$$

The representation for the output $\mathbf{Y}$ has the $(D+1)\,M$ dimensional vector with the following sequence:

$$\mathbf{Y} = \Bigg(V^{(1)}, V^{(2)}, \cdots, V^{(M)}, \frac{\partial V^{(1)}}{\partial x_1^{(1)}}, \frac{\partial V^{(2)}}{\partial x_1^{(2)}}, \cdots, \frac{\partial V^{(M)}}{\partial x_1^{(M)}}, \\ \frac{\partial V^{(1)}}{\partial x_2^{(1)}}, \frac{\partial V^{(2)}}{\partial x_2^{(2)}}, \cdots, \frac{\partial V^{(M)}}{\partial x_2^{(M)}}, \frac{\partial V^{(1)}}{\partial x_3^{(1)}}, \cdots, \frac{\partial V^{(M)}}{\partial x_D^{(M)}} \Bigg), \tag{S8}$$

where $V^{(m)}$ is the potential energy of the m-th image $\mathbf{x}^{(m)}$ and $\partial V^{(m)}/\partial x_d^{(m)}$ is the gradient of the potential with respect to $x_d^{(m)}$.

The covariance or kernel function, $k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right)$, quantifies how two data $\mathbf{x}^{(m)}$ and $\mathbf{x}^{(n)}$ are related. In the case of the MB potential example, we use the standard (squared exponential) kernel. For the alanine dipeptide example, which has a periodic potential energy surface (PES), we use a periodic kernel. Our method also incorporates gradient information to exploit both potential energies and the gradient of system energies to construct a more accurate Gaussian PES. The elements of a simple kernel matrix $\boldsymbol{K}$ can be written as

$$(\boldsymbol{K})_{mn} = k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right), \tag{S9}$$

where $(m,\ n)\ \in (M \times N)$ are the indices of the matrix. $\partial_n \boldsymbol{K}$, $\partial_m \boldsymbol{K}$ and $\partial_{mn} \boldsymbol{K}$ are the tensors derived from the derivatives of the simple kernel matrix with their shapes: $(M \times D \times N)$, $(D \times M \times N)$, and $(D \times M \times D \times N)$, respectively. The components for these tensors are expressed as follows:

$$(\partial_n \boldsymbol{K})_{m,d,n} = \frac{\partial}{\partial x_d^{(n)}} k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \tag{S10}$$

$$(\partial_m \boldsymbol{K})_{d,m,n} = \frac{\partial}{\partial x_d^{(m)}} k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \tag{S11}$$

$$(\partial_{mn} \boldsymbol{K})_{d,m,d',n} = \frac{\partial}{\partial x_d^{(m)}} \left(\frac{\partial}{\partial x_{d'}^{(n)}} k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \right). \tag{S12}$$

An extended kernel matrix $\boldsymbol{K}_{\text{ext}}$ can be constructed via reshaping its tensors into a matrix form. We reshape the tensors $\partial_n \boldsymbol{K}$, $\partial_m \boldsymbol{K}$ and $\partial_{mn} \boldsymbol{K}$ of the form $(M \times D \times N)$, $(D \times M \times N)$, $(D \times M \times D \times N)$ into $(M \times DN)$, $(DM \times N)$, $(DM \times DN)$. The explicit expression can be written as follows:

$$\boldsymbol{K}_{\text{ext}} = \begin{pmatrix} \boldsymbol{K} & \partial_n \boldsymbol{K} \\ \partial_m \boldsymbol{K} & \partial_{mn} \boldsymbol{K} \end{pmatrix} = \begin{pmatrix} k\left(\mathbf{X}, \mathbf{X}'\right) & \frac{\partial k\left(\mathbf{X}, \mathbf{x}'^{(1)}\right)}{\partial \mathbf{x}'^{(1)}} & \frac{\partial k\left(\mathbf{X}, \mathbf{x}'^{(2)}\right)}{\partial \mathbf{x}'^{(2)}} & \cdots & \frac{\partial k\left(\mathbf{X}, \mathbf{x}'^{(\mathrm{N})}\right)}{\partial \mathbf{x}'^{(\mathrm{N})}} \\ \frac{\partial k\left(\mathbf{x}^{(1)}, \mathbf{X}'\right)}{\partial \mathbf{x}^{(1)}} & \frac{\partial^2 k\left(\mathbf{x}^{(1)}, \mathbf{x}'^{(1)}\right)}{\partial \mathbf{x}^{(1)} \partial \mathbf{x}'^{(1)}} & \frac{\partial^2 k\left(\mathbf{x}^{(1)}, \mathbf{x}'^{(2)}\right)}{\partial \mathbf{x}^{(1)} \partial \mathbf{x}'^{(2)}} & \cdots & \frac{\partial^2 k\left(\mathbf{x}^{(1)}, \mathbf{x}'^{(\mathrm{N})}\right)}{\partial \mathbf{x}^{(1)} \partial \mathbf{x}'^{(\mathrm{N})}} \\ \frac{\partial k\left(\mathbf{x}^{(2)}, \mathbf{X}'\right)}{\partial \mathbf{x}^{(2)}} & \frac{\partial^2 k\left(\mathbf{x}^{(2)}, \mathbf{x}'^{(1)}\right)}{\partial \mathbf{x}^{(2)} \partial \mathbf{x}'^{(1)}} & \frac{\partial^2 k\left(\mathbf{x}^{(2)}, \mathbf{x}'^{(2)}\right)}{\partial \mathbf{x}^{(2)} \partial \mathbf{x}'^{(2)}} & \cdots & \frac{\partial^2 k\left(\mathbf{x}^{(2)}, \mathbf{x}'^{(\mathrm{N})}\right)}{\partial \mathbf{x}^{(2)} \partial \mathbf{x}'^{(\mathrm{N})}} \\ \vdots & \vdots & \vdots & \ddots & \\ \frac{\partial k\left(\mathbf{x}^{(\mathrm{M})}, \mathbf{X}'\right)}{\partial \mathbf{x}^{(\mathrm{M})}} & \frac{\partial^2 k\left(\mathbf{x}^{(\mathrm{M})}, \mathbf{x}'^{(1)}\right)}{\partial \mathbf{x}^{(\mathrm{M})} \partial \mathbf{x}'^{(1)}} & \frac{\partial^2 k\left(\mathbf{x}^{(\mathrm{M})}, \mathbf{x}'^{(2)}\right)}{\partial \mathbf{x}^{(\mathrm{M})} \partial \mathbf{x}'^{(2)}} & & \frac{\partial^2 k\left(\mathbf{x}^{(\mathrm{M})}, \mathbf{x}'^{(\mathrm{N})}\right)}{\partial \mathbf{x}^{(\mathrm{M})} \partial \mathbf{x}'^{(\mathrm{N})}} \end{pmatrix},$$

where the shape of the matrix $k\left(\mathbf{X}, \mathbf{X}'\right) = \boldsymbol{K}$ is $(M \times N)$, $\partial k\left(\mathbf{X}, \mathbf{x}'^{(\mathrm{n})}\right)/\partial \mathbf{x}'^{(n)}$ is $(M \times D)$, $\partial k\left(\mathbf{x}^{(\mathrm{m})}, \mathbf{X}'\right)/\partial \mathbf{x}^{(m)}$ is $(D \times N)$, and $\partial^2 k\left(\mathbf{x}^{(\mathrm{m})}, \mathbf{x}'^{(\mathrm{n})}\right)/\partial \mathbf{x}^{(\mathrm{m})}\partial \mathbf{x}'^{(\mathrm{n})}$is $(D \times D)$ matrix. In total, the shape of the matrix $\boldsymbol{K}_{\mathrm{ext}}$ is $(D(M+1) \times D(N+1))$.

Bayesian inference comes handy if one wants to estimate the energies of unknown pathway $\mathbf{X}_*$ when the observed energy $\mathbf{Y}$ of a structure $\mathbf{X}$ is given. Specifically, the probability of $\mathbf{X}_*$ to have $\mathbf{Y}_*$, for given $\mathbf{X}$ and $\mathbf{Y}$ can be calculated by integrating throughout possible posterior. Since the integration of Gaussian is also Gaussian, we obtain a Gaussian function with posterior predictive mean $\boldsymbol{\mu}_*$ and covariance $\boldsymbol{\Sigma}_*$ as follows:

$$\begin{aligned} p\left(\mathbf{Y}_* | \mathbf{X}_*, \mathbf{X}, \mathbf{Y}\right) = & \int p\left(\mathbf{Y}_* | \mathbf{X}_*, \mathbf{w}\right) p\left(\mathbf{w} | \mathbf{X}, \mathbf{Y}\right) d\mathbf{w} \\ = & \mathcal{N}\left(\mathbf{X}_* | \boldsymbol{\mu}_*,\ \boldsymbol{\Sigma}_*\right), \end{aligned} \tag{S13}$$

where $\boldsymbol{\mu}_*$ and $\boldsymbol{\Sigma}_*$ are defined as follows:

$$\begin{gathered} \boldsymbol{\mu}_* = \mathbf{m}_* + \mathbf{K}_*^T \left(\mathbf{K}_y + \sigma_n \mathbf{I}\right)^{-1} \left(\mathbf{Y} - \mathbf{m}_y\right) \\ \boldsymbol{\Sigma}_* = \mathbf{K}_{**} + \mathbf{K}_*^T \left(\mathbf{K}_y + \sigma_n \mathbf{I}\right)^{-1} \mathbf{K}_*, \end{gathered} \tag{S14}$$

where $\mathbf{K}_y = k\left(\mathbf{X},\ \mathbf{X}\right),\ \mathbf{K}_{**} = k\left(\mathbf{X}_*, \mathbf{X}_*\right)$. $\sigma_n \mathbf{I}$ is the error matrix having zero for all off-diagonal terms with the size $(M(D+1) \times N(D+1))$. For simplicity, we denote $\mathbf{X}^{(m)} \equiv \mathbf{x}^{(m)}$ and $(\boldsymbol{K})_{ij} = \left(k\left(\mathbf{X}, \mathbf{X}'\right)\right)_{ij} = k\left(\mathbf{X}^{(i)}, \mathbf{X}'^{(j)}\right) = k\left(\mathbf{x}^{(i)}, \mathbf{x}'^{(j)}\right)$. The kernel matrix can be simply represented as $\boldsymbol{K} = k\left(\mathbf{X}_*, \mathbf{X}\right)$. $\mathbf{m}_y$ and $\mathbf{m}_*$ are mean vectors at each site $\mathbf{X}$ and $\mathbf{X}_*$, respectively. We set the elements of the mean vector corresponding to potential data to the average of the gathered data. We set the elements of the mean vector corresponding to force to zero. The predictive Hessian $\boldsymbol{H}_*$ of Gaussian PES is

$$\boldsymbol{H}_* = \mathbf{K}_H^T \left(\mathbf{K}_y + \sigma_n \mathbf{I}\right)^{-1} \left(\mathbf{Y} - \mathbf{m}_y\right), \tag{S15}$$

where $\mathbf{K}_H = \begin{pmatrix} \partial_{nn}\boldsymbol{K} & \partial_{mnn}\boldsymbol{K} \end{pmatrix}$ is a Hessian kernel, having the second and third derivatives of covariance function as its components. $\partial_{nn}\boldsymbol{K}, \partial_{mnn}\boldsymbol{K}$ are tensors with shape $(M \times D \times D \times N)$ and $(D \times M \times D \times D \times N)$. The tensors are reshaped as the matrices of the sizes of $(M \times DDN)$ and $(DM \times DDN)$, respectively. Each component of tensors can be computed by

$$\begin{aligned} \left(\partial_{nn}\boldsymbol{K}\right)_{m,d',d,n} = & \frac{\partial}{\partial x_{d'}^{(n)}} \left(\frac{\partial}{\partial x_d^{(n)}} k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \right) && \text{(S16)} \\ \left(\partial_{mnn}\boldsymbol{K}\right)_{d,m,d'',d',n} = & \frac{\partial}{\partial x_{d''}^{(n)}} \left(\frac{\partial}{\partial x_{d'}^{(n)}} \left(\frac{\partial}{\partial x_d^{(m)}} k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \right) \right), && \text{(S17)} \end{aligned}$$

Most kernels used in GP are associated with hyperparameters. Hyperparameters are selected to maximize the log-likelihood function:

$$
\begin{aligned}
\ln p\left(\mathbf{Y}|\mathbf{X}\right) =& \ln \mathcal{N}\left(\mathbf{Y}|\boldsymbol{m}_Y, \mathbf{K}_y\right) \\
=& -\frac{1}{2}\left(\mathbf{Y}-\boldsymbol{m}_y\right)^T \left(\mathbf{K}_y+\sigma_n\mathbf{I}\right)^{-1}\left(\mathbf{Y}-\boldsymbol{m}_y\right) \\
& -\frac{1}{2}\ln\left|\mathbf{K}_y+\sigma_n\mathbf{I}\right| - \frac{N}{2}\ln\left(2\pi\right). \qquad \text{(S18)}
\end{aligned}
$$

However, adding gradient information often breaks the positive definiteness of a kernel matrix. If it happens, the determinant is not guaranteed to be positive, which may make $\ln|\mathbf{K}_y+\sigma_n\mathbf{I}|$ impossible. In cases when we get a negative determinant, we generate pseudo-kernel and output vector, $k\left(\mathbf{X}+\mathbf{dX},\ \mathbf{X}+\mathbf{dX}\right), \mathbf{Y}+\mathbf{dY}$ where all the force information is replaced with the interpolated potential value near $\mathbf{dX}$. With this positive-definite guaranteed kernel, we can find hyperparameters that minimize the log-likelihood function, Eq. (S18).

S3. Squared Exponential Kernel

The elements of the squared exponential kernel matrices used for the MB potential are as follows:

$$
\left(\partial_n\boldsymbol{K}\right)_{mn} = \frac{\partial}{\partial\mathbf{x}^{(n)}} k\left(\mathbf{x}^{(m)},\mathbf{x}^{(n)}\right) = \frac{\sigma_f}{l^2}\left(\mathbf{x}^{(m)}-\mathbf{x}^{(n)}\right)\times k\left(\mathbf{x}^{(m)},\mathbf{x}^{(n)}\right) \qquad \text{(S19)}
$$

$$
\left(\partial_m\boldsymbol{K}\right)_{mn} = \frac{\partial}{\partial\mathbf{x}^{(m)}} k\left(\mathbf{x}^{(m)},\mathbf{x}^{(n)}\right) = -\frac{\sigma_f}{l^2}\left(\mathbf{x}^{(m)}-\mathbf{x}^{(n)}\right)\times k\left(\mathbf{x}^{(m)},\mathbf{x}^{(n)}\right) \qquad \text{(S20)}
$$

$$
\left(\partial_{mn}\boldsymbol{K}\right)_{d,m,d',n} = \frac{\sigma_f}{l^2}\left(\delta_{dd'} - \frac{1}{l^2}\left(\mathbf{x}_d^{(m)}-\mathbf{x}_d^{(n)}\right)\left(\mathbf{x}_{d'}^{(m)}-\mathbf{x}_{d'}^{(n)}\right)\right)\times k\left(\mathbf{x}^{(m)},\mathbf{x}^{(n)}\right), \qquad \text{(S21)}
$$

where the shape of $\partial_n\boldsymbol{K}$ is $(M\times 2N)$. The matrix is constructed from a $(M\times 2\times N)$ tensor and the number 2 comes from the system dimension, i.e. $D=2$. That is, in the tensor representation, for each $m\in M$, there exists a $(2\times N)$ matrix. We can reshape the $(2\times N)$ matrix to a $2N$ dimensional array by flattening it. Resulting $\partial_n\boldsymbol{K}$ term forms the $(M\times 2N)$ matrices. The $\partial_m\boldsymbol{K}$ term is constructed similarly. A $(2\times M\times N)$ tensor is converted to a $(2M\times N)$ matrix meaning that there exist a $(2\times M)$ matrix for each $n\in N$. Similarly, $\partial_{mn}\boldsymbol{K}$ is a $(2M\times 2N)$ matrix constructed from a $(2\times M\times 2\times N)$ tensor. $\partial_{mn}\boldsymbol{K}$ consists of two parts, the global and diagonal parts. The global part is the derivative of the kernel and the diagonal part is the derivative of a coefficient. $\delta_{dd'}$ is a Kronecker delta function having nonzero only when the terms of the second derivative are conducted in the same dimension as the first one.

We use a zero mean Gaussian function in the MB potential model, meaning that the expectation value is $\boldsymbol{\mu}_* = \mathbf{K}_*^T\left(\mathbf{K}_y+\sigma_n\mathbf{I}\right)^{-1}\mathbf{Y}$. The Hessian of the Gaussian PES is calculated separately by using Hessian kernel, $\mathbf{K}_H = \big(\partial_{nn}\boldsymbol{K} \quad \partial_{mnn}\boldsymbol{K}\big)$. The tensor representations of $\partial_{nn}\boldsymbol{K}$ and $\partial_{mnn}\boldsymbol{K}$ are given by

$$
\begin{aligned}
\left(\partial_{nn}\boldsymbol{K}\right)_{m,d',d'',n} =& \frac{\sigma_f}{l^2}\left\{\delta_{d'd''} + \frac{1}{l^2}\left(\mathbf{x}_{d''}^{(m)} - \mathbf{x}_{d''}^{(n)}\right)\left(\mathbf{x}_{d'}^{(m)} \ - \mathbf{x}_{d'}^{(n)}\right)\right\} \times k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \qquad \text{(S22)} \\
\left(\partial_{mnn}\boldsymbol{K}\right)_{d,m,d',d'',n} =& \frac{\sigma_f}{l^2}\left\{\frac{\delta_{d'd''}}{l^2}\left(\mathbf{x}_{d}^{(m)} - \mathbf{x}_{d}^{(n)}\right) - \frac{1}{l^4}\left(\mathbf{x}_{d}^{(m)} - \mathbf{x}_{d}^{(n)}\right)\left(\mathbf{x}_{d''}^{(m)} - \mathbf{x}_{d''}^{(n)}\right)\left(\mathbf{x}_{d'}^{(m)} \ - \mathbf{x}_{d'}^{(n)}\right)\right. \\
&\left. + \frac{1}{l^2}\left(\delta_{d''d} + \delta_{d'd}\right)\left(\mathbf{x}_{d}^{(m)} - \mathbf{x}_{d}^{(n)}\right)\right\} \times k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right). \qquad \text{(S23)}
\end{aligned}
$$

S4. Periodic Kernel

Suppose that $\sin(\mathbf{x})$ is a vector that maps every element x_i to $\sin(x_i)$. The kernel for AD model is

$$
k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) = \sigma_f \exp\left(-\frac{2}{l^2}\sum_{d}^{D}\sin^2\frac{1}{2}\left(\mathbf{x}_{d}^{(m)} - \mathbf{x}_{d}^{(n)}\right)\right).
$$

Tensors for each kernel matrix $\partial_n\boldsymbol{K}, \partial_m\boldsymbol{K}, \partial_{mn}\boldsymbol{K},\ \partial_{nn}\boldsymbol{K}, \text{and}\, \partial_{mnn}\boldsymbol{K}$ are

$$
\left(\partial_{n}\boldsymbol{K}\right)_{m,d',n} = \frac{\sigma_f}{l^2}\sin\left(\mathbf{x}_{d'}^{(m)} - \mathbf{x}_{d'}^{(n)}\right)k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \qquad \text{(S24)}
$$

$$
\left(\partial_{m}\boldsymbol{K}\right)_{d,m,n} = -\frac{\sigma_f}{l^2}\sin\left(\mathbf{x}_{d}^{(m)} - \mathbf{x}_{d}^{(n)}\right)k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \qquad \text{(S25)}
$$

$$
\begin{aligned}
\left(\partial_{mn}\boldsymbol{K}\right)_{d,m,d',\ n} =& \frac{\sigma_f}{l^2}\left(\frac{1}{2}\cos 2\left(\mathbf{x}_{d}^{(m)} - \mathbf{x}_{d}^{(n)}\right)\delta_{dd'}\right. \\
&\left. - \frac{1}{l^2}\sin\left(\mathbf{x}_{d'}^{(m)} - \mathbf{x}_{d'}^{(n)}\right)\sin\left(\mathbf{x}_{d}^{(m)} - \mathbf{x}_{d}^{(n)}\right)\right)k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \qquad \text{(S26)}
\end{aligned}
$$

$$
\begin{aligned}
\left(\partial_{nn}\boldsymbol{K}\right)_{m,d',d'',\ n} =& \frac{\sigma_f}{l^2}\left(\frac{1}{2}\cos 2\left(\mathbf{x}_{d''}^{(m)} - \mathbf{x}_{d''}^{(n)}\right)\delta_{d''d'}\right. \\
&\left. + \frac{1}{l^2}\sin\left(\mathbf{x}_{d''}^{(m)} - \mathbf{x}_{d''}^{(n)}\right)\sin\left(\mathbf{x}_{d'}^{(m)} - \mathbf{x}_{d'}^{(n)}\right)\right)k\left(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}\right) \qquad \text{(S27)}
\end{aligned}
$$

$$
\begin{aligned}
(\partial_{mnn}\boldsymbol{K})_{d,\ m,d'',d',\ n} &= \\
&\frac{\sigma_f}{l^2}\Bigg\{-\sin 2\left(\mathbf{x}^{(m)}_{d''}-\mathbf{x}^{(n)}_{d''}\right)\delta_{d''d'}\delta_{d''d}+\frac{\delta_{d''d}}{l^2}\cos\left(\mathbf{x}^{(m)}_{d''}-\mathbf{x}^{(n)}_{d''}\right)\sin\left(\mathbf{x}^{(m)}_{d'}-\mathbf{x}^{(n)}_{d'}\right)\\
&+\frac{\delta_{d'd}}{l^2}\cos\left(\mathbf{x}^{(m)}_{d'}-\mathbf{x}^{(n)}_{d'}\right)\sin\left(\mathbf{x}^{(m)}_{d''}-\mathbf{x}^{(n)}_{d''}\right)\\
&+\frac{1}{2l^2}\cos 2\left(\mathbf{x}^{(m)}_{d''}-\mathbf{x}^{(n)}_{d''}\right)\sin\left(\mathbf{x}^{(m)}_{d}-\mathbf{x}^{(n)}_{d}\right)\delta_{d''d}\delta_{d''d'}\\
&+\frac{1}{l^4}\sin\left(\mathbf{x}^{(m)}_{d''}-\mathbf{x}^{(n)}_{d''}\right)\sin\left(\mathbf{x}^{(m)}_{d'}-\mathbf{x}^{(n)}_{d'}\right)\sin\left(\mathbf{x}^{(m)}_{d}-\mathbf{x}^{(n)}_{d}\right)\Bigg\}\,k\left(\mathbf{x}^{(m)},\mathbf{x}^{(n)}\right).
\end{aligned}
\tag{S28}
$$

S5. Alanine dipeptide GPAO example

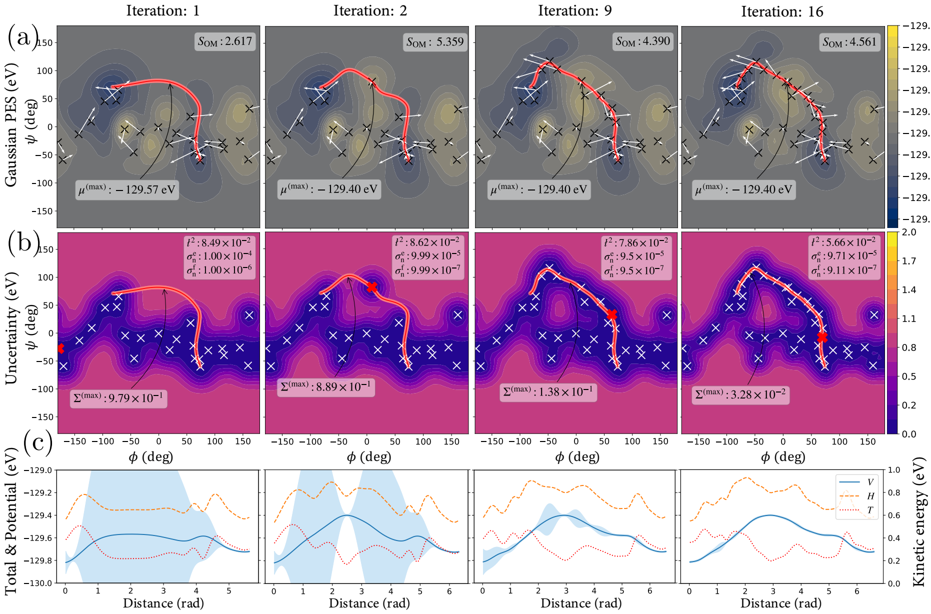

Figure S1: The progress of GP-CSA process applied to alanine dipeptide. (a) Gaussian PES, (b) uncertainty map and (c) energy profile along the pathway are displayed. Gaussian PES and uncertainty map are generated by the data marked with black (or white) crosses. White arrows indicate the directions and relative strengths of the data. The converged pathway (Iteration 16) corresponds to the pathway obtained with GP-CSA under 2 ps time step (Figure 4b-F).

S6. Efficient global action optimization using Action-CSA

The Action-CSA is set to start with randomly generated 40 initial pathways, followed by local optimizations using GPAO. Afterward, we proceed with the following steps:

1. Pick two pathways from the candidates randomly.
2. Generate a new pathway from the two pathways using a crossover operator.
3. Apply random mutations/perturbations to the new pathway with a certain probability
4. Locally minimize the new pathway.
5. Pick the most *similar* pathway to the new pathway from the candidates. If the distance between the new pathway and a *similar* pathway is lower than a cutoff distance, compare the OM action between them. Otherwise, compare the highest OM pathway among the candidates.
6. If the new one has lower action than its counterpart, replace it with the new one.
7. Reduce the cutoff distance. If the cutoff distance is lower than a certain threshold, stop the iteration. Otherwise, go back to step 1.

With two randomly selected pathways, we choose a random image along the pathway. With the selected position, we slice both trajectories at the selected image and merge them. After a new pathway is generated, a random mutation/perturbation is performed with a probability of 30%. For the pathway's mutation, the pathway is modified by adding another random number at the sine components of a pathway. After the local minimization of the pathway using the GPAO method, we search the nearest neighbor by measuring the Fréchet distance between the new pathway and the current candidates. The nearest neighbor is considered *similar*, readily accessible to each other, if the Fréchet distance is shorter than the current cutoff distance d_{cut}. The initial cutoff distance is set to half of the average distance between the initial pathways. After an iteration, we reduce d_{cut} by 2% and repeat the whole procedure from step 1 until d_{cut} becomes below 0.05.