GraphVAMPnets for Uncovering Slow Collective Variables of Self-Assembly Dynamics

Bojun Liu[1], Mingyi Xue[1], Yunrui Qiu[1], Kirill A. Konovalov[1], Michael S. O'Connor[2], Xuhui Huang[1,2*]

[1]Department of Chemistry, Theoretical Chemistry Institute, University of Wisconsin-Madison, Madison, WI, 53706, USA

[2]Biophysics Graduate Program, University of Wisconsin-Madison, Madison, WI, 53706, USA

[*]To whom correspondence should be addressed. E-mail: xhuang@chem.wisc.edu

Abstract

Uncovering slow collective variables (CVs) of self-assembly dynamics is important to elucidate its numerous kinetic assembly pathways and drive the design of novel structures for advanced materials through the bottom-up approach. However, identifying the CVs for self-assembly presents several challenges. First, self-assembly systems often consist of identical monomers and the feature representations should be invariant to permutations and rotational symmetries. Physical coordinates, such as aggregate size, lack the high-resolution detail, while common geometric coordinates like pairwise distances are hindered by the permutation and rotational symmetry challenge. Second, self-assembly is usually a downhill process, and the trajectories often suffer from insufficient sampling of backward transitions that correspond to the dissociation of self-assembled structures. Popular dimensionality reduction methods, like tICA, impose detailed balance constraints, potentially obscuring the true dynamics of self-assembly. In this work, we employ GraphVAMPnets which combines graph neural networks with variational approach for Markovian process (VAMP) theory to identify the slow CVs of the self-assembly processes. First, GraphVAMPnets bears the advantages of graph neural networks, in which the graph embeddings can represent self-assembly structures in a high-resolution while being invariant to permutations and rotational symmetries. Second, it is built upon VAMP theory that studies Markov processes without forcing detailed balance constraint, which addresses the out-of-equilibrium challenge in self-assembly process. We demonstrate GraphVAMPnets for identifying slow CVs of self-assembly kinetics in two systems: aggregation of two hydrophobic molecules and self-assembly of patchy particles. We expect that our GraphVAMPnets can be widely to applied to molecular self-assembly.

I. Introduction

Self-assembly, the process by which monomers assemble themselves into highly ordered structures, has been studied over decades.(*1-8*) Traditionally, self-assembly can be studied thermodynamically(*9-15*) by comparing the free energies of monomeric and assembled structures, however, this may fail to reveal the true nature of those complex self-assembly processes which are governed via kinetic control. Therefore, it is important to study the kinetics of self-assembly, which provides the opportunity to understand the underlying mechanism of self-assembly process and further realize the rational design of advanced materials with novel structures through the bottom-up approach.(*16-19*)

Molecular dynamic (MD) simulations(*20-22*) can serve as a powerful tool to elucidate the detailed dynamics of self-assembly at a molecular level, while it's usually difficult for experiments to capture the transient intermediate states. However, solely looking at MD trajectories is still insufficient as there is a lack of statistical description of major kinetic pathways of self-assembly, which may lead to missing or biased interpretations of the relevant events in dynamics. Kinetic network models (KNMs)(*16*) exhibit the great potential to study the self-assembly dynamics by statistically combining many MD simulation trajectories. The Markovian KNM is also known as Markov state model (MSM)(*23-30*) and has been widely applied for the studies of biomolecular dynamics. For self-assembly, KNMs can be used to estimate both thermodynamic and kinetic properties, extract parallel kinetic pathways and predict long time-scale dynamics by integrating many short trajectories of assembly processes sampled by MD simulations.

Several key steps are involved in constructing KNM for self-assembly dynamics. The first step is to identify collective variables (CVs) that can sufficiently describe the relevant dynamics of the system. In general, one can first select the representative features, e.g., physical coordinates (such as size of aggregates, asphericity parameter that represents the shape of aggregates etc.) or geometric coordinates (such as pairwise distances via permutation-invariant modifications). Then, several dimensionality reduction approaches (e.g., time-structure independent component analysis (tICA)(*31, 32*), VAMPnets(*33, 34*), diffusion maps(*35, 36*)) can be further applied to obtain a reduced kinetic space constructed by the slow collective variables. The second step involves discretizing the reduced continuous kinetic space into microstates using clustering algorithms (e.g., k-centers(*37, 38*), k-means(*39*), density-based clustering(*40*)), and further coarse-graining microstates into macrostates (also called metastable states) based on the separations of fast and slow dynamics using kinetic lumping algorithms (e.g., PCCA(*41, 42*), PCCA+(*43*)). The third step is to count the number of transitions between states with a certain lag time to obtain the transition count matrix (TCM) and further normalize TCM to obtain the Transition probability matrix (TPM). If the model is Markovian, TPM can be used to predict the long time-scale dynamics based on master equation: $p^T(t + n\tau) = p^T(t)T^n(\tau)$, where T is the TPM, $p(t)$ and $p(t + n\tau)$ are the distributions of the ensemble at time t and $t + n\tau$, respectively. In addition, the first left eigenvector (eigenvalue equals to one) of

TPM represents the stationary distribution, and the other correspond to the population-weighted slowest dynamic modes of the system. Finally, transition path theory (TPT)(*44-46*) can be applied to extract the dominant kinetic pathways from the source state (monomeric structure) to the sink state (assembled structure). And mean first passage time (MFPT), defined as the average time for the first transition between metastable states, can also be computed along these kinetic pathways.

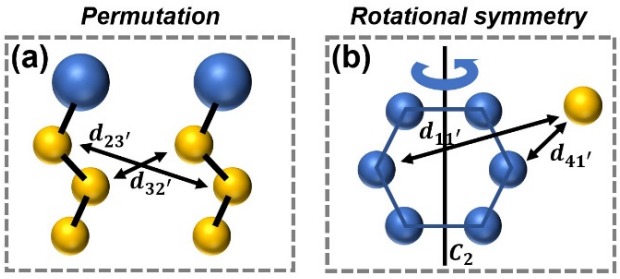

Figure 1. Illustration of challenges in feature selection for self-assembly: (a) Permutation-invariance: the monomers in assembly system are usually identical. If two monomers exchange their positions, the structure is the same but the defined pairwise distances $d_{23'}$ and $d_{32'}$ will change; (b) Rotational symmetry-invariance: the monomers in assembly system could be rotationally symmetric. If the hexagonal ring rotates along its C_2 axis, the structure is identical but the defined pairwise distances $d_{11'}$ and $d_{41'}$ will change.

In recent years, KNM has become increasingly popular in material science and been adopted to study the dynamics of self-assembly, such as ice nucleation(*47*), colloidal particles(*48-50*), and amphiphiles assembly(*51-54*). However, there still remain two major challenges for its applications in self-assembly(*16*). Firstly, the selected representative features should be able to capture the relevant dynamics but invariant to permutations and rotational symmetries. Usually, the selected physical coordinates are invariant to permutations and rotational symmetries but requires additional physical insights and may not be able to capture the complex dynamics. And for those non-physical coordinates, such as pairwise distances, they can describe the assembly structures in more detail but are hindered by permutation and rotational symmetry-invariance nature in self-assembly dynamics (see Fig 1). Secondly, self-assembly is typically an energetically downhill process, which means that the simulated trajectories could be out of equilibrium with the insufficient sampling of the backward disassociation processes. However, for the method in traditional KNMs construction pipeline such as tICA, it requires the additional detailed balance constraint by symmetrizing the time-lagged correlation matrix with its transpose, which may result in a highly biased estimation of true dynamics.

To address the above challenges, we employ GraphVAMPnets(*55*) (also known as GDyNet(*56*)) which combines graph neural networks with variational approach for Markovian process (VAMP)(*57*) theory to identify the slowest CVs of self-assembly dynamics. GraphVAMPnets has been applied to study lithium-ion transport in amorphous polymer electrolytes and successfully revealed four distinct solvation environments with implications for ion conductivity(*56*). GraphVAMPnets has also been applied to study protein folding and displayed

a faster convergence rate than the original VAMPnets(*55*). To apply GraphVAMPnets for studying self-assembly, we have introduced several methodological advancements in this work. Firstly, unlike the previous implementations of GraphVAMPnets that use a fixed number of nearest neighbors as edges for each node, we introduce a variable number of edges for each node in self-assembly systems, depending on the self-assembled structures. Moreover, we define edges beyond the simple distance-based metrics by also considering the orientation of particles such as in the patch particle self-assembly. In addition, we initialize particles that are identical, subject to permutation or rotational symmetries, with the same node embeddings. These node embeddings are constrained to have the same values during the training. To demonstrate the power of our method, we apply it to two systems: the aggregation of two hydrophobic molecules (9f9f) in water and the self-assembly of patchy particles. For both systems, we show that our GraphVAMPnets approach can effectively addresses the two aforementioned key challenges and surpasses other similar methods for identifying slow CVs for self-assembly.

II. Methods

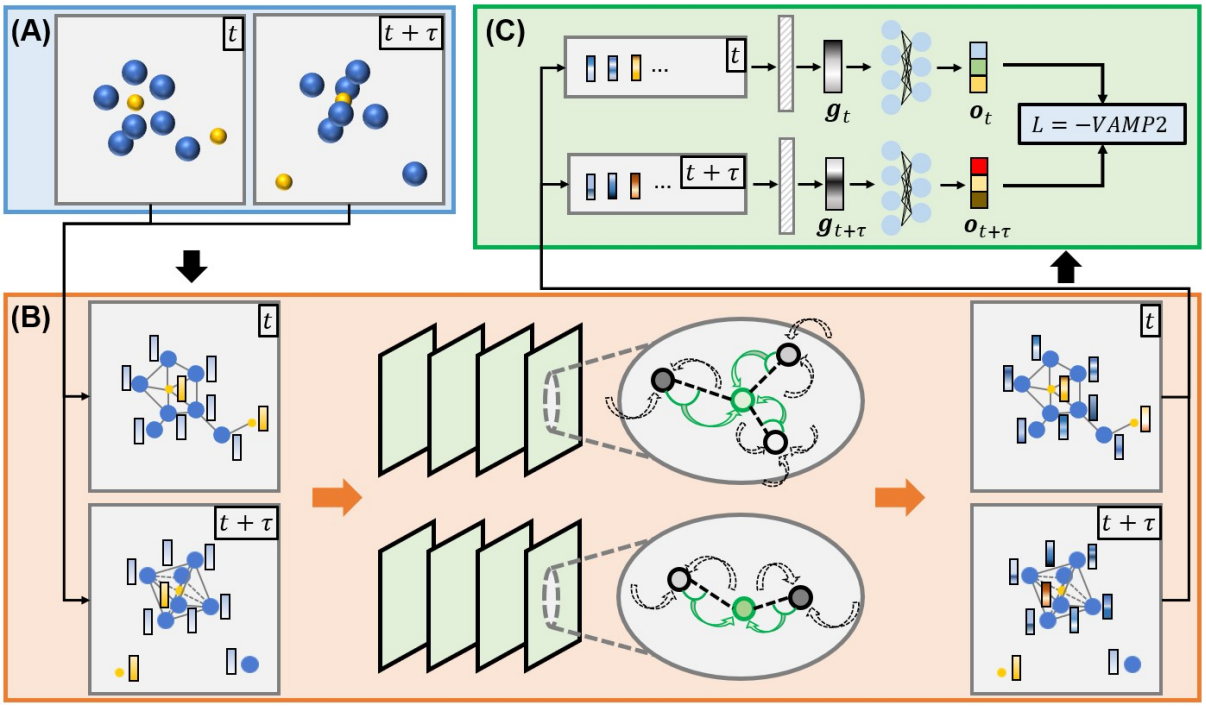

Figure 2. A schematic architecture of GraphVAMPnets for identifying the slow CVs of self-assembly dynamics. The method can be divided into three parts: (A) Pairs of structures separated by a lag time τ are extracted from the dataset. (B) Each pair of structures is represented as graphs and fed into two parallel graph convolutional neural networks. (C) Graph embeddings obtained by average pooling are fed into a two-lobe deep neural network trained under a VAMP-2 based objective function .

A. Graph representations

We first define particles/atoms of the self-assembly system as nodes in the graph and embed

them using node vectors $\{v_i\}_{i=1,\ldots,n}$. The particles that are identical or rotational symmetric should be initialized with the same node embeddings. Then, we identify all the edges between nodes based on one of the following rules: (a) consider the edges that only exist between the particles and a certain number of their nearest neighbors; (b) define a cut-off distance and only the pairwise particles that have distances within the cut-off are considered as having edges; (c) define the edges between particles manually based on the existing bonds/interactions. Then, we embed these edges as vectors by using Gaussian expanded pairwise distances:

$$e_{ij}^{k} = \exp\left(-\frac{\left(d_{ij} - c_k\right)^2}{\sigma^2}\right) \tag{1}$$

where e_{ij}^{k} is the k^{th} element of edge vector e_{ij}, d_{ij} is the pairwise distance between node i and node j, and c_k is the center of the expansion function.

B. Graph neural network structure and algorithms

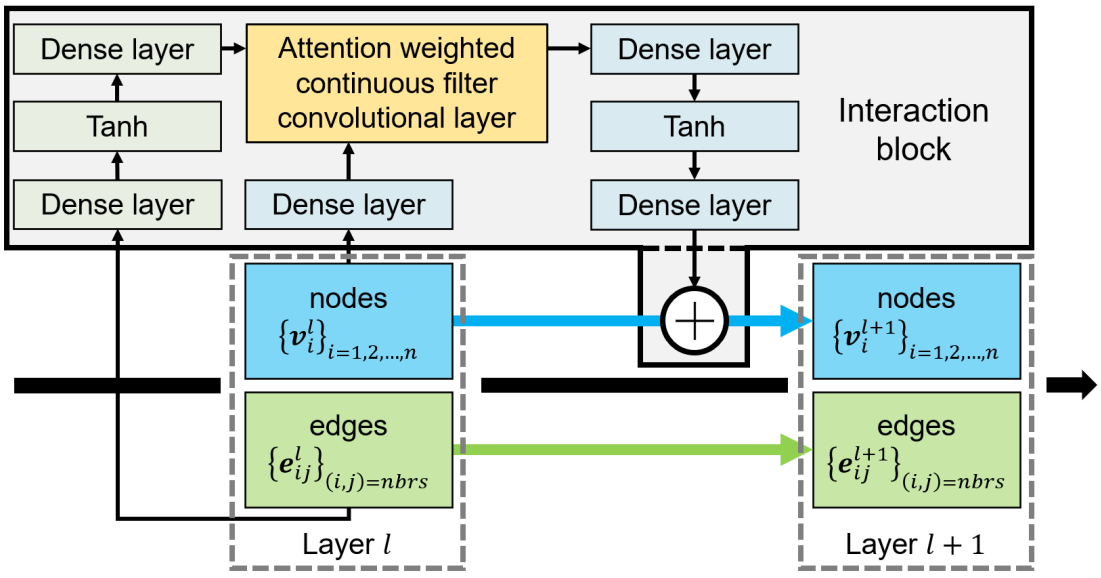

Figure 3. Illustration of layers l to $l+1$ in GraphVAMPnets for studying self-assembly. The nodes (in blue) with edges (in green) are updated through an interaction block (in grey) before being passed to the next layer.

The graph neural network structure that we employed in GraphVAMPnets is based on SchNet framework(*58*) as shown in Fig 3. In the following section, we show how node vectors at layer l are updated to layer $l+1$ according to the algorithms given in each interaction block:

(a). All the node vectors and edge vectors in the graph at layer l are mapped into the same dimensions by using dense layers:

$$\begin{cases} \left(v_i^l\right)' = W_v^l v_i^l + b_v^l \\ \left(e_{ij}^l\right)' = W_{2_e}^{\ l}\left(\mathrm{Tanh}\left(W_{1_e}^{\ l} e_{ij}^l + b_{1_e}^{\ l}\right)\right) + b_{2_e}^{\ l} \end{cases} \tag{2}$$

where v_i^l and e_{ij}^l represent the original node and edge vectors at layer l, while $\left(v_i^l\right)'$ and $\left(e_{ij}^l\right)'$ correspond to the mapped node and edge vectors with the same dimensions.

(b). We feed mapped node and edge vectors into the attention weighted continuous filter convolutional layer in which the local interaction of each particle i from its surroundings is obtained:

$$\begin{aligned} \left(z_i^l\right)' &= \sum_j \alpha_{ij}^l \left(v_j^l\right)' \circ \left(e_{ij}^l\right)' \\ \alpha_{ij}^l &= \frac{\exp\left(\left(\left(v_j^l\right)' \circ \left(e_{ij}^l\right)'\right)^T W_\alpha^l\right)}{\sum_j \exp\left(\left(\left(v_j^l\right)' \circ \left(e_{ij}^l\right)'\right)^T W_\alpha^l\right)} \end{aligned} \tag{3}$$

where $\circ$ represents the element-wise multiplication, and α_{ij}^l is the attention weight, which indicates the relative importance of all adjacent particles j to their central particle i in the convolutions.

(c). The local interactions on particles obtained from convolutional layer are projected back and added to the original node vectors via residue connections, where the edge vectors remain the same:

$$\begin{cases} z_i^l = W_{2_z}^{\ l}\left(\mathrm{Tanh}\left(W_{1_z}^{\ l}\left(z_i^l\right)' + b_{1_z}^{\ l}\right)\right) + b_{2_z}^{\ l} \\ v_i^{l+1} = v_i^l + z_i^l \end{cases} \tag{4}$$

With the deepening of graph neural networks, the information considering local environments of particles are propagated through edges and the representative feature embedding of each particle can be successfully obtained with capturing the global multi-body interactions within the system.

C. The VAMP theory

Variational approach for Markov process (or VAMP) was developed by Wu and Noe(*57*). The theoretical foundation of VAMP relies on Koopman operator theory which originates from fluid mechanics field and can be applied to investigate the dynamics of an arbitrary time-homogeneous Markov process without time-reversible constraints. The Koopman operator $\mathcal{K}_\tau$ is defined as follows:

$$\begin{aligned}\mathcal{K}_\tau g(x) &= E_t[g(x_{t+\tau})|x_t = x] \\ &= \int g(x_{t+\tau})\, p(x_{t+\tau}|x_t = x) dx_{t+\tau} \\ &= \int g(y) p_\tau(x, y) dy\end{aligned} \tag{5}$$

where $g(x)$ is an arbitrary observable defined on the full configuration space, $p_\tau(x, y)$ is the transition probability from configuration x to configuration y with a certain lag time τ. The Koopman operator is defined to compute the conditional expected value of the observable $g(x)$ for a given lag time τ. The Koopman operator is also known as backward operator, which can be used to backward propagate the ensemble distribution in the configuration space with lag time τ.

With the above definitions, the goal is to obtain the leading singular functions of Koopman operator, which can serve as the slowest dynamic modes of the kinetic system and be propagated on average by a linear model:

$$\begin{gathered}\langle \psi_i, \mathcal{K}_\tau \phi_i \rangle_{\rho_0} = \sigma_i, \;\; 1 = \sigma_1 > \sigma_2 \geq \sigma_3 \geq \cdots \\ \mathbb{E}_t[\boldsymbol{\phi}(x_{t+\tau})] = K^T \mathbb{E}_t[\boldsymbol{\psi}(x_t)], \quad K^T = diag(\sigma_1, \sigma_2, \dots)\end{gathered} \tag{6}$$

where ϕ_i, ψ_i and σ_i are the ith left, right singular functions, and singular value of the Koopman operator, respectively.

VAMP utilizes the variational approach (VA) to approximate the leading left and right singular functions of Koopman operator with two sets of feature functions $\mathbf{f} = (f_1, f_2, \dots, f_k)^T$, $\mathbf{g} = (g_1, g_2, \dots, g_k)^T$. It defines the VAMP-2 score (denoted as $\hat{R}_2$) as the summation of the square of approximate singular values and aims to find the feature functions which provide the largest VAMP-2 score:

$$\begin{cases} \langle f_i, f_j \rangle_{\rho_0} = \delta_{ij}, \quad \langle g_i, g_i \rangle_{\rho_1} = \delta_{ij} \\ \hat{R}_2 = \sum_{i=1}^{k} \hat{\sigma}_i^2 = \sum_{i=1}^{k} \langle f_i, K_\tau g_i \rangle_{\rho_0}^2 \end{cases} \tag{7}$$

In practical, the feature functions can be further decomposed as the linear combinations $\mathbf{f} = U^T \boldsymbol{\mathcal{X}}_\mathbf{0}$, $\mathbf{g} = V^T \boldsymbol{\mathcal{X}}_\mathbf{1}$ of two set of basis functions $\boldsymbol{\mathcal{X}}_\mathbf{0} = (\mathcal{X}_{01}, \mathcal{X}_{02}, \dots, \mathcal{X}_{0m})^T$, $\boldsymbol{\mathcal{X}}_\mathbf{1} = (\mathcal{X}_{11}, \mathcal{X}_{12}, \dots, \mathcal{X}_{1n})^T$, and the problem can be converted as following:

$$\begin{cases} U^T C_{00} U = I, \quad V^T C_{11} V = I \\ \hat{R}_2 = \sum_{i=1}^{k} (u_i^T C_{01} v_i)^2 \end{cases} \tag{8}$$

where the time-instantaneous and time-lagged covariance matrices are defined as:

$$\begin{cases} C_{00} = \mathbb{E}_t[\boldsymbol{\chi_0}(x_t)\boldsymbol{\chi_0}(x_t)^T] \\ C_{11} = \mathbb{E}_t[\boldsymbol{\chi_1}(x_{t+\tau})\boldsymbol{\chi_1}(x_{t+\tau})^T] \\ C_{01} = \mathbb{E}_t[\boldsymbol{\chi_0}(x_t)\boldsymbol{\chi_1}(x_{t+\tau})^T] \end{cases} \tag{9}$$

To solve the above optimization problem, C_{00} and C_{11} can be further split as the product of their square roots $C_{00}^{-\frac{1}{2}}$, $C_{11}^{-\frac{1}{2}}$, and the singular values of matrix $K_s = C_{00}^{-\frac{1}{2}} C_{01} C_{11}^{-\frac{1}{2}}$ can be selected as equivalent to the approximate singular values of Koopman operator. As a result, the VAMP-2 score equals the square of Frobenius norm of K_s:

$$\hat{R}_2 = \left\| C_{00}^{-\frac{1}{2}} C_{01} C_{11}^{-\frac{1}{2}} \right\|_F^2 = \left\| \bar{C}_{00}^{-\frac{1}{2}} \bar{C}_{01} \bar{C}_{11}^{-\frac{1}{2}} \right\|_F^2 + 1 \tag{10}$$

where the matrices with bars represent the remove-mean covariance matrices. The time evolution equation of the average observable functions can be rewritten with respect to the arbitrary basis functions:

$$E_t[\boldsymbol{\chi_1}(x_{t+\tau})] \approx K^T E_t[\boldsymbol{\chi_0}(x_t)] \tag{11}$$

where K is known as Koopman matrix and is equal to $C_{00}^{-1} C_{01}$. The Koopman matrix can also be interpreted as the reduced Koopman operator which governs the dynamics of latent variables $\boldsymbol{\chi_0}, \boldsymbol{\chi_1}$ by a finite-dimensional linear model with the minimum least square error $E_t[\|\boldsymbol{\chi_1}(x_{t+\tau}) - K^T \boldsymbol{\chi_0}(x_t)\|^2]$.

Based on VAMP theory, Noe et al. further developed VAMPnets(*33, 59*), in which the basis functions $\boldsymbol{\chi_0}$ and $\boldsymbol{\chi_1}$ are parameterized by two parallel deep neural networks to introduce non-linearity and the loss function is designed as the negative VAMP-2 score. By minimizing the loss function of VAMPnets, VAMP-2 score is maximized, and the trained neural networks can thus serve as the optimal basis functions to reconstruct the feature functions $\mathbf{f}$ and $\mathbf{g}$ which best approximate true singular functions of the Koopman operator. These features functions are also called the slow CVs, capturing the relevant dynamics of the system.

In our GraphVAMPnets method, the graph neural networks with average pooling layer can serve as the basis functions in VAMP theory to embed each MD data frame, where $\boldsymbol{o}_t = (\mathcal{X}_{01}(x_t), \dots, \mathcal{X}_{0m}(x_t))^T$, $\boldsymbol{o}_{t+\tau} = (\mathcal{X}_{11}(x_{t+\tau}), \dots, \mathcal{X}_{1m}(x_{t+\tau}))^T$ (see Fig. 2C). As a result, the empirical remove-mean covariance matrices $\bar{C}_{00}, \bar{C}_{01}, \bar{C}_{11}$ can be estimated. From these matrices we can compute the VAMP-2 score. The parameters in the graph neural networks are then updated by minimizing the negative VAMP-2 score and the optimal approximated leading singular functions of Koopman operator ($\mathbf{f} = U^T \boldsymbol{\chi_0}$, $\mathbf{g} = V^T \boldsymbol{\chi_1}$) can be obtained. These singular functions can then serve as the slow CVs to construct the reduced kinetic space for projection.

D. Permutation and rotational symmetry-invariance of GraphVAMPnets

One of the key properties of graph neural networks is that the learned representation of each node only depends on its local environment instead of the node index, which is called index equivariance of node embeddings. After the average pooling of all node vectors in the graph, the obtained graph representation is therefore invariant to the node indexing. In self-assembly systems, if two particles are identical with respect to permutations or rotational symmetries, they should be initialized with the same node embeddings and constrained to have the same values during the training. Thus, if two of these identical particles exchange positions through permutations or rotations, it is equivalent to the exchange of their node indices. As a result, the same graph embedding can be obtained corresponding to the same conformations in the self-assembly system.

III. Results and Discussions

A. Aggregation of two hydrophobic molecules in water

To demonstrate the power of the GraphVAMPnets in identifying the slow CVs that are invariant to permutation and rotational symmetry, we investigated an aggregation process consisting of two indistinguishable hydrophobic molecules (i.e., (9-(Diphenylmethylene)-9H-fluorene) or 9d9f) in water (see Fig. 4A). Our MD dataset consists of 40 MD trajectories, with an aggregated simulation time of $4\mu s$. In order to establish the permutation and rotational symmetry-invariant nature of the system, we measured (θ_1, θ_2) as the angles between the normal vectors and the center of mass displacement vectors of 9d9f molecules for each structure (Fig. 4B left panel). As shown in Fig. 4B right panels, the conformation with the angle of θ is identical with the conformation with the angle of $(180° - \theta)$, as the two conformations can be perfectly overlaid via the permutation and rotation of 9d9f molecules. Subsequently, we projected all MD conformations onto the (θ_1, θ_2) constructed space, where four identical regions with the same conformations can be observed (see Fig. 6A).

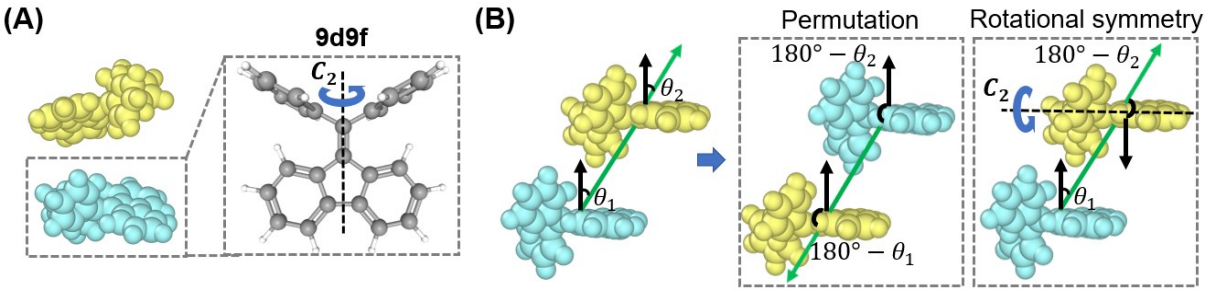

Figure 4. Illustrations of two aggregation-prone molecules (9d9f) and the rotational symmetry-invariant features of the system. (a) The chemical structure of 9-(Diphenylmethylene)-9H-fluorene (9d9f) molecule. (b) θ_1 and θ_2 are defined as the angles between the normal vectors and the center of mass displacement vectors of 9d9f molecules. When two 9d9f molecules are exchanged, the angles change from (θ_1, θ_2) to $(180 - \theta_1, 180 - \theta_2)$, while the structure is identical. When one of the 9d9f molecules (yellow) rotates along its C_2 axis, θ_2 becomes $180 - \theta_2$, while the structure of the system remains the same.

We then compare the performance of tICA, VAMPnets, and GraphVAMPnets in identifying the slow CVs for the aggregation of two 9d9f molecules. For tICA and VAMPnets, we selected 14 representative carbon atoms of each 9d9f molecule, and subsequently extracted 196 inter-molecular pairwise distances between pairs of these atoms on different molecules as the input features (see Fig. 5A). For the GraphVAMPnets approach, we represented each conformation using a graph. In particular, we selected the same 14 carbon atoms of each 9d9f molecule and embedded these atoms with node vectors, where permutable or rotationally symmetric atoms are embedded with the same node vector (denoted as of the same color as shown in Fig. 5B). We then considered the edges existing between each atom and its 10 nearest neighbors and identified all the edges for each conformation accordingly. In addition, we created a control test of GraphVAMPnets (so-called GraphVAMPnets-rand), in which node embeddings are not constrained based on their symmetries. All the methods were conducted using the same hyperparameters (see SI Table. S1) and two collective variables were successfully obtained from each method. The training curves of GraphVAMPnets and the implied time scales (ITS) plots of two collective variables from different methods are shown in SI Figure S1 and S2.

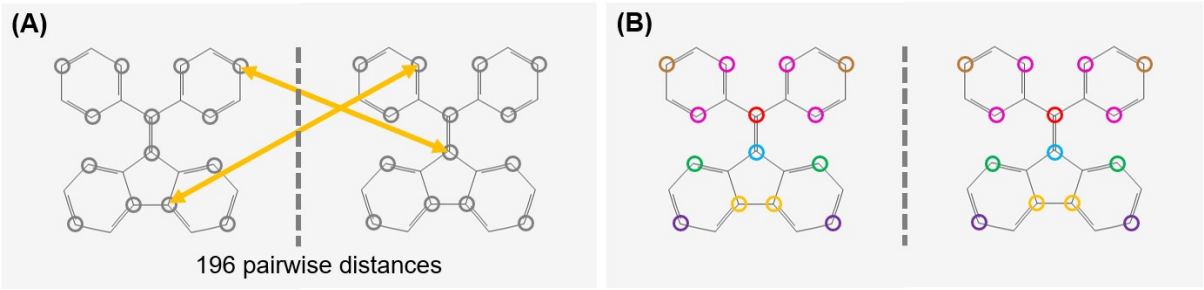

Figure 5. Feature embedding of 9d9f system using different methods. (A) For tICA and VAMPnets, 14 carbon atoms (in grey circle) are selected in each of the 9d9f molecules, forming a set of 196 inter-molecular pairwise distances (represented by orange arrows). (B) For GraphVAMPnets method, the same 28 carbon atoms are embedded with node vectors. The atoms circled with the same color are identical with respect to permutations and rotational symmetries and hence constrained to have the same node embeddings.

In Fig 6, we visualized the two slowest CVs obtained from different methods (i.e., tICA, VAMPnets, GraphVAMPnets-rand, and GraphVAMPnets) in the (θ_1, θ_2) space. Each CV represents a specific dynamic mode of the system with respect to the transitions between its positive and negative regions. As shown in Fig 6C (top row), the first CV obtained from each method is consistent. Specifically, there exist four symmetric positive regions of the CV1 (see red regions in the top row of Fig 6C), and aggregate conformations in each of these regions are identical and only subject to a rotation (see example conformations in Fig. 6B). However, the second CV (bottom row in Fig 6B) shows different dynamic modes of transitions obtained from different methods. Specifically, the second mode from tICA, VAMPnets and VAMPnets-rand corresponds to the dynamic transition between structurally identical states (they differ by permutations or rotations along C_2 axis of 9d9f molecules instead of the actual conformational change). In contrast, the second mode obtained from our GraphVAMPnets exhibits a

symmetrical pattern consistent with four identical regions in the (θ_1, θ_2) space. These results demonstrate that GraphVAMPnets can yield correct CVs of the aggregation dynamic that are invariant to permutation and rotational symmetries.

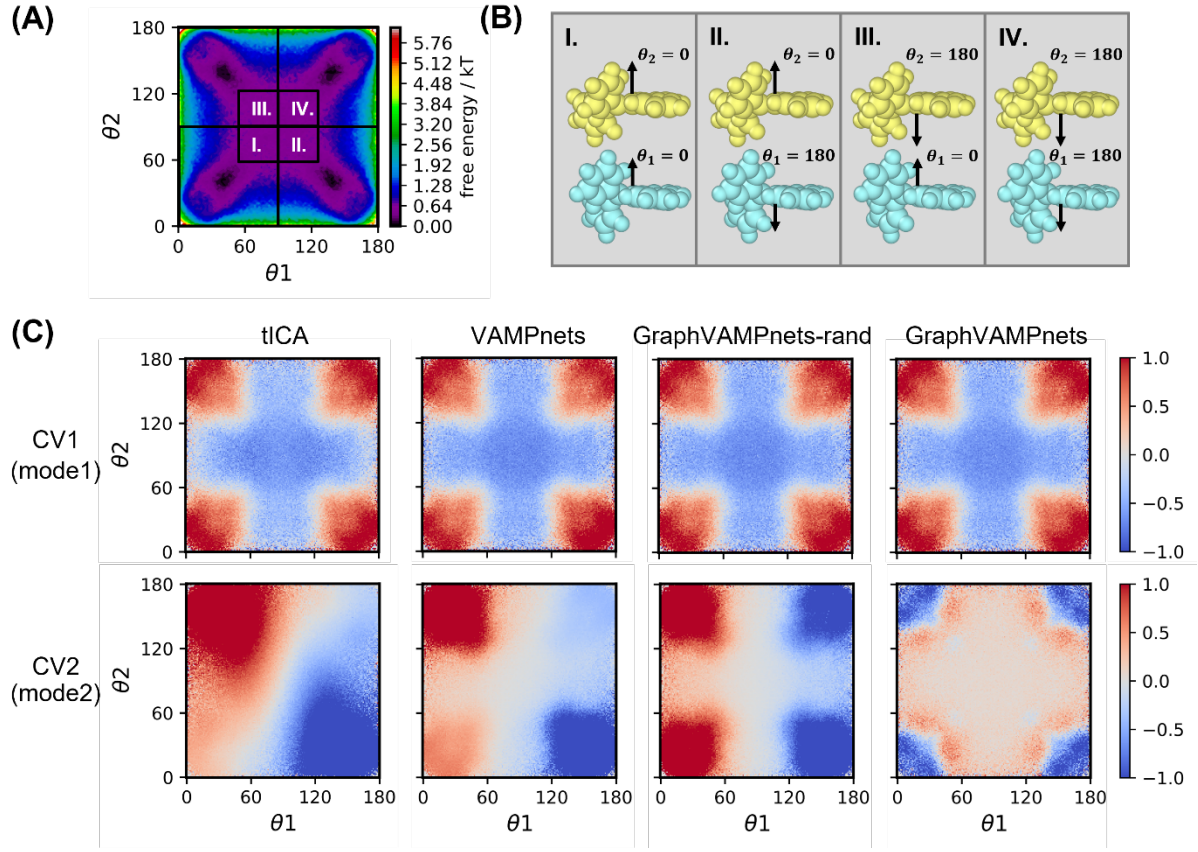

Figure 6. Visualization of the identified CVs from different methods in (θ_1, θ_2) space. (A) Free energy landscape of all MD conformations projected on the (θ_1, θ_2) space. (B) Identical representative conformations from the four regions labeled I through IV are illustrated . (C) Projection of CV values onto the (θ_1, θ_2) space. All tested methods identified the same first CV, however, only GraphVAMPnets identified the second CV that is symmetric with respect to permutation and rotation of molecules in the system.

To further analyze the dynamics of aggregation, we projected all the MD conformations onto the reduced space constructed by the two CVs obtained from GraphVAMPnets (see Fig. 7B). To understand the physical insights of each CV, solvent-accessible surface area (SASA) and ϕ angle (representing the relative orientation of two 9d9f molecules) are reported in Fig. 7B. Interestingly, we found that the SASA feature keeps decreasing along CV1, indicating the aggregation process from the left to the right in the projected 2D free energy map. Furthermore, we noticed that the ϕ angle spans from nearly 0 to 180 degrees along CV2, which reveals a structural rearrangement of 9d9f aggregates. Specifically, the free energy basin at the left side of the map represents a dispersed state (left insert panel of Fig. 7B), where two 9d9f molecules are separated and the rotation angle ϕ between two molecules equals to ~90 degrees. The free energy basins at the right represent various aggregation states. Specifically, the top one contains

conformations in which two 9d9f molecules stack with each other along the same directions, while the bottom one corresponds to conformations in which two 9d9f molecules stack in an opposite way. In addition, the asymmetric behavior along the CV2 of the map also corresponds to slightly different free energies of the two aggregated conformations (i.e., stacking along the same and opposite directions). These observations suggest that the conformation of the aggregate state with two 9d9f molecules stacked in the opposite way is slightly more compact and displays a smaller SASA. As a results, this aggregate state (bottom basin with negative values of CV2 in Fig. 7B) is slightly more stable than the other aggregate state with two 9d9f molecules stacked in the same direction (top basin with positive values of CV2 in Fig. 7B). This result is consistent with the chemical intuitions, as two phenyl groups have a steric hindrance effect if the two 9d9f molecules approach at the same direction, resulting in the less $\pi - \pi$ stack interactions between fluorenyl-rings of 9d9f molecules and an increase in free energy.

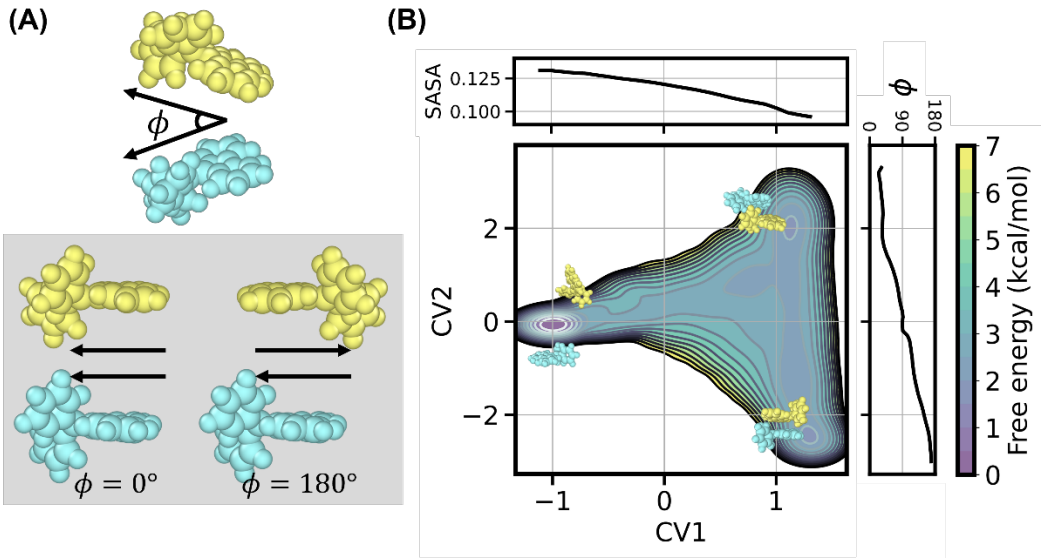

Figure 7. Conformations of 9d9f aggregation-system in the space of the two slowest collective variables identified by the GraphVAMPnets approach. (A) ϕ is defined as the angle between the two vectors pointing from fluorenyl rings to phenyl rings in 9D9F molecules, which represents the relative orientation of two 9d9f molecules. (B) The free energy landscape of all MD conformations in CV1 and CV2 space. Solvent-accessible surface area (SASA) decreases along CV1. ϕ is decreasing from 180 to 0 degrees along CV2. Three representative conformations, i.e., a dispersed structure (left) and stacked structures (right) with the same and opposite directions correspond to the three energy basins in this projection.

B. Self-assembly of patchy particles

In this section, we applied GraphVAMPnets to find the slow CVs and build a KNM to investigate the dynamics for the patchy particle self-assembly system.(*48*) In this system, our dataset taken from previous studies(*48, 60, 61*) consists of 60 coarse-grained MD simulation

trajectories of the self-assembly based on the soft patchy particle models (SPPMs). Each MD trajectory contains 2,000 frames where the saving interval for each frame is 1,000 MD steps and the assembled dodecahedral cages are formed in these MD simulations (see Fig. 8B). For each MD frame, there are 960 patchy particles in the simulation box, and each patchy particle consists of one sphere particle at the center and three patches with the half opening angle $\theta_m^k = 55°$ along different directions, and the angles between patches are $108°$, $108°$, and $120°$ respectively (see Fig. 8A). To build GraphVAMPnets, we first identified all existing interactions (bonds) as edges between patchy particles in a MD frame. Specifically, if the distance between the centers of particles is smaller than 0.75 and each particle contains a patch with the angle from r_{ij} smaller than the half opening angle ($\theta_i^k \leq 55°, \theta_j^\lambda \leq 55°$), we consider that there exists a bond between these two patchy particles (see Fig. 8C). Then, we embedded all the conformations using graphs, in which each patchy particle corresponds to one node vector in the graph. We selected 100 frames (i.e., 100,000 MD steps) as the training lag time and set two output nodes at the last layer for obtaining two slowest dynamic modes (or CVs). And the GraphVAMPnets was trained for 50 epochs and repeated for 10 times, where the validation scores had shown perfect convergence (see SI Fig. S3 and the values of hyperparameters are listed in SI Table S2).

After the training, we successfully obtained two CVs, and we further projected all the conformations onto these two CVs (see Fig. 8D for the 2D free energy map). The ITS of these two CVs are also reported in Fig. 8E, and they exhibit good separations of time scales and reach convergence within 50 frames. Furthermore, we explored the underlying physical meanings of these identified CVs and found that the first CV has the strong correlation to the average aggregate size (with the Pearson correlation coefficient of 0.94) while the second CV corresponds to the average asphericity parameter (or Ap) of the aggregates (with the Pearson correlation coefficient of 0.58). We note that Ap is the morphology parameter where 0 represents perfect sphere and 1 represents perfect rod for aggregates. To facilitate the visualization, we also projected the MD conformations onto the CV versus the average aggregate size (see Fig. 8F) and onto the second CV versus the average Ap (see Fig. 8G). These two projected free energy maps demonstrate that the majority of MD conformations lie along the diagonal of the plots, indicating their strong correlations. Moreover, when we plot the first CV and the average aggregate size (see Fig. 8H) or the second CV and the average Ap (see Fig. 8I) as a function of simulation time, we observe similar correlations.

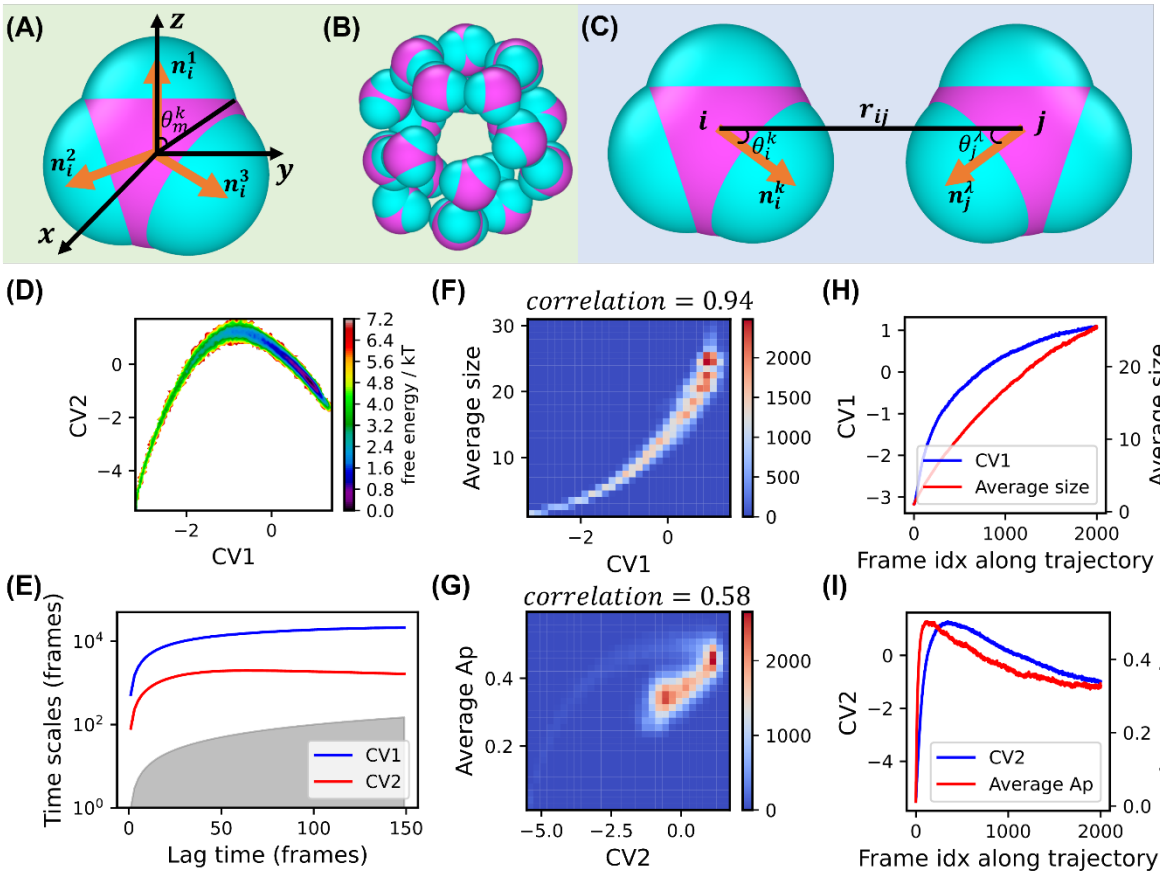

Figure 8. Illustration of the patchy particle system and the two slowest collective variables obtained from GraphVAMPnets. (A) The structure of a single patchy particle, where directions of three patches are $n_i^1 = (0,0,1)^T, n_i^2 = (0,-0.9531,-0.309)^T, n_i^3 = (0.5257,0.6935,-0.5038)^T$. The opening angle θ_m^k of each patch equals to 55°. (B) The self-assembled structure of patchy particles, i.e., dodecahedral cage. (C) Two patchy particles are considered as connected if $r_{ij} \leq 0.75$ and $\theta_i^k, \theta_j^\lambda$ are smaller than 55°. (D) The free energy landscape of all MD conformations projected onto CV1 and CV2 identified by GraphVAMPnets. (E) The implied time scales (or ITS) of the two identified CVs. (F) Histogram of all conformations in the CV1 vs average aggregate size space. The Pearson correlation coefficient equals 0.95. (G) Histogram of all conformations in the CV2 vs average asphericity parameter (Ap) space. The Pearson correlation coefficient between CV2 and average Ap equals to 0.56. (H) Time evolution of CV1 and average size along an MD trajectory. (I) Time evolution of CV2 and average Ap along an MD trajectory.

After obtaining slow CVs, we constructed a KNM to analyze the dynamics of the patchy particle self-assembly process. Following the key steps of KNM construction outlined in the introduction, we first discretized the two-dimensional continuous space into one hundred microstates (see Fig. 9A) using the k-means clustering algorithm and then lumped these microstates into five larger macrostates (see Fig. 9B) by the PCCA+ method. The corresponding ITS plots for microstate-model and macrostate-model are shown in SI Figure

S4, respectively. We then examined the physical properties of five metastable macrostate. As shown in Fig. 9C, the average aggregate size increases from State 4 to 0, indicating an overall aggregation process across states $4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$. At the first stage ($4 \rightarrow 3$), the concentration of monomers rapidly decreases and approaches zero at State 3 (see Fig. 9D). At the second stage ($3 \rightarrow 2$), aggregates containing 12 pentacycles begin to appear, and these aggregates already have a similar shape to the final self-assembled structures, namely dodecahedral cages formed by 12 pentagonal rings. In the third stage ($2 \rightarrow 1$), small aggregates collide and form larger aggregates, including those composed of 12 pentacycles (Fig. 9E and F). At the final stage ($1 \rightarrow 0$), the number of aggregates and the number of 12-pentacycle aggregates do not change significantly. However, we observed that in State 0, nearly all aggregates with 12 pentacycles form additional hexagonal rings, while those dodecahedral cages (i.e., 12-pentacyle aggregates containing zero hexagonal rings) mostly disappear (see Fig. 9G). This result indicates that the kinetics from State 1 to State 0 is related to a structural rearrangement that breaks the dodecahedral cage to form larger aggregate containing more hexagonal rings.

We next estimated the MFPTs between each pair of states along the aggregation pathway via maximum likelihood estimation (MLE) (see Fig. 9H). The results revealed a downhill dynamic process of patchy particle self-assembly from the monomeric state (State 4) to the dodecahedral cage state (State 1), because the forward aggregation processes are significantly faster than their backward transitions. In addition, our results show that the dodecahedral cage is the thermodynamically most-favorable structure. Moreover, an off-pathway intermediate state (State 0) was also identified where the dodecahedral cage undergoes further structural transitions to form larger aggregates containing additional hexagonal rings, resulting in an increase in free energy (estimated from the ratio of forward and backward MFPTs between State 1 and 0).

Overall, our KNM reveals a downhill process of patchy particle self-assembly from the monomeric structures (State 4) to the dodecahedral cages (State 1), while aggregates with hexagonal rings (State 0) are kinetically reachable, but energetically unfavorable. Our results agree well with the physical insights provided by a previous study of the same system(*48*). Specifically, one patchy particle contains three patches with angles between them as $108°, 108°, 120°$, respectively. As the formation of the pentagonal rings are favored by patches with $108°$ angles, the dodecahedral cage consisted of pentagonal rings is the thermodynamically favorable assembled structure (i.e., State 1). However, the additional $120°$ angle in each patchy particle allows the formation of the hexagonal rings, leading to the metastable state of State 0. Our results open up a window of opportunities for the rational design of self-assembly structures with varying yields of dodecahedral cages (containing only pentagonal rings) versus those containing hexagonal rings, achieved by tuning the angles between patches in patchy particles.

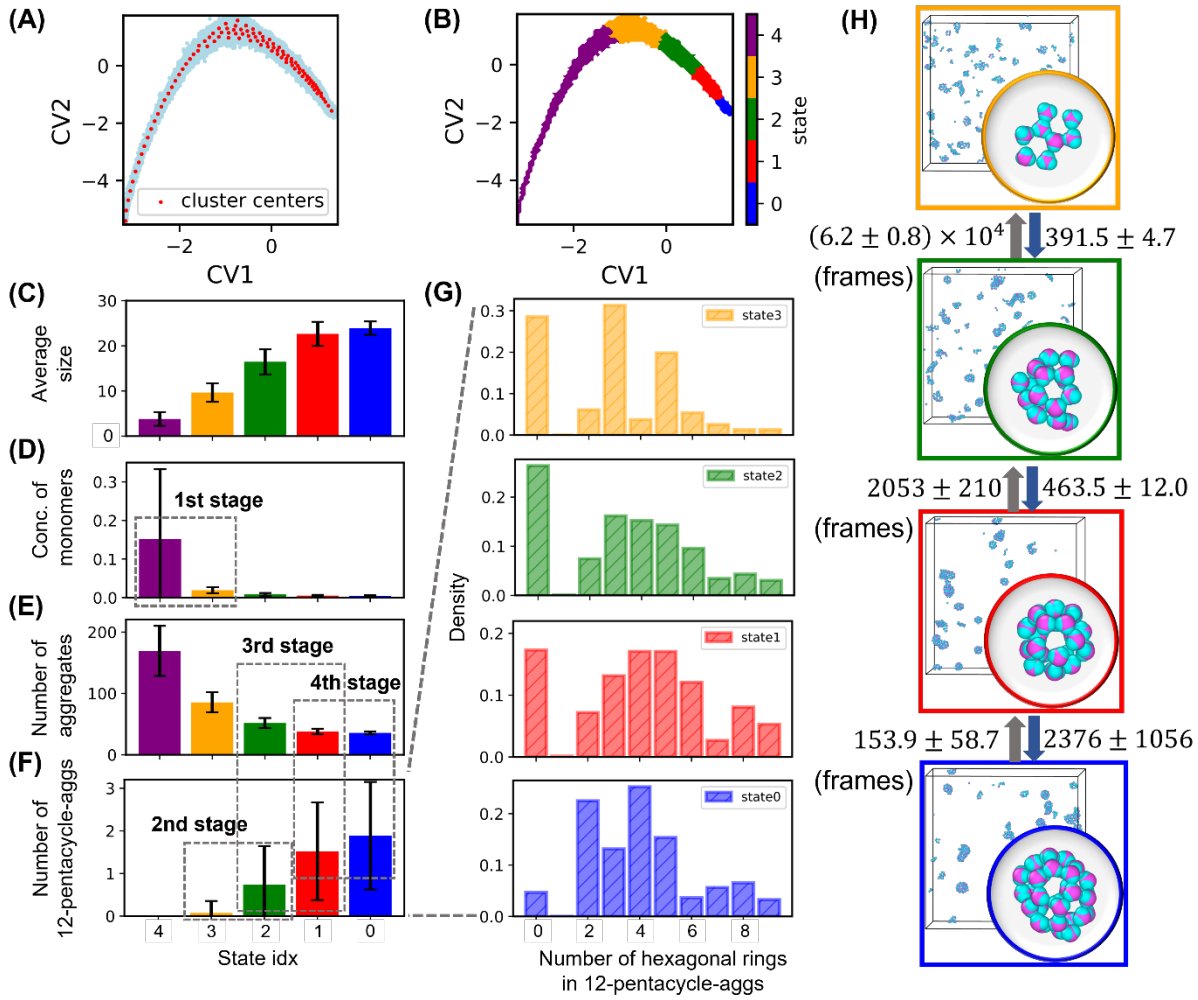

Figure 9. Analysis of patchy particle self-assembly dynamics. (a) Projection of all the MD conformations onto the CV1, CV2 space. The 100 microstates obtained by the k-means clustering method are shown with red dots. (b) Projection of all the MD conformations onto the CV1, CV2 space colored according to their macrostate labels. (c) Average size of aggregates including monomers for the conformations in each state. (d) Concentration of monomers in each state. (e) Number of aggregates in each state. (f) Number of aggregates with 12 pentagonal rings (12-pentacycle-aggregate) in each state. (g) Distributions of the number of hexagonal rings in each 12-pentacycle-aggregate for states excluding State 4. (h) Mean first passage time (MFPT) between states along the assembly process from state 3 (orange) to state 0 (blue). In panels (c)-(f) the height of bars represents the mean value and the error bar represent standard deviation.

In this study, we tackled the inherent challenges in kinetics of self-assembly systems by further developing GraphVAMPnets. In the original GraphVAMPnets(*55*), each node is connected to a fixed number of nearest neighbors. This approach may work for systems with a relatively stable structure (e.g., a protein in the native conformation), but is ill-suited for self-assembly systems where the number of neighbors can change dramatically. By using a variable number of edges, our model adapts to changes in local particle density, allowing a more accurate description of the dynamics. Additionally, in self-assembly systems, particles are defined as identical if they are symmetric to permutations or rotations. By initializing these particles with

the same node embeddings, our model can obtain the slow CVs that invariant to these symmetries. Finally, our approach allows us to embed the whole simulation box, providing a wholistic model of the dynamics crucial in many-body systems. With these enhancements, we demonstrated that our method is superior to other methods traditionally used to construct KNMs for self-assembly. While the implementation of our GraphVAMPnets is general to self-assembly, we anticipate that multi-component systems might require a hierarchical approach to achieve adequate resolution. In the future, our model may also adopt the iVAMPnets framework(*62*) to decompose the system into separate Markovian domains or conduct the cluster-level pooling to obtain the appropriate feature embeddings for individual aggregates. We foresee that our GraphVAMPnets holds potential to be applied to study other systems, for example, phase separation of intrinsically disordered proteins.

IV. Conclusion

In this work, we developed GraphVAMPnets for studying dynamics of self-assembly processes. GraphVAMPnets takes the advantages of graph neural networks (i.e., SchNet), in which the complex and diverse structures that arise in self-assembly processes are represented as graphs. This representation captures the system's local environment and global many-body interactions while being invariant to permutations and rotational symmetries. By optimizing the graph neural networks using the VAMP-2 score, slow CVs can be obtained for the self-assembly process. To apply the GraphVAMPnets approach to self-assembly systems, we have introduced several methodological advancements. These include allowing for a variable number of edges for each node, considering particle orientation when defining edges, and initializing particles that are identical subject to permutation or rotational symmetries with the same embeddings. To demonstrate the effectiveness of our method, we applied it to two different systems: the aggregation of two hydrophobic molecules in water and the self-assembly of patchy particles. For hydrophobic aggregation, our GraphVAMPnets approach successfully distinguished the permutation and rotational symmetry-invariant aggregation pathways, while other commonly used methods (such as tICA) failed to achieve this. For the self-assembly of patchy particles, our approach identified two slow CVs which well described the average size and shape of the patchy particle conformations and allowed us to distinguish metastable states in patchy particle aggregation with subtle differences in the ring structure of aggregates, which is infeasible with traditional approaches such as tICA. We anticipate that the GraphVAMPnets approach shows great promise for widespread application in the study of self-assembly dynamics and for facilitating the rational design of advanced materials with novel self-assembled structures.

V. Data Availability Statement

The source code and simulation datasets are available from the following website: https://github.com/xuhuihuang/graphvampnets.

VI. Supplementary Material

Supplementary material consists of supplementary tables: Table. S1 & S2 and supplementary figures: Fig. S1 to S4.

VII. Acknowledgements

XH acknowledges the support from the Office of the Vice-Chancellor for Research and Graduate Education at the University of Wisconsin-Madison with funding from the Wisconsin Alumni Research Foundation. XH also acknowledges the support from the Hirschfelder Professorship Fund. MO acknowledges the support by the NIH/NIGMS under award number T32 GM130550.

VIII. References:

1. D. Philp, J. Fraser Stoddart, Self-Assembly in Natural and Unnatural Systems. *Angewandte Chemie International Edition in English* **35**, 1154-1196 (1996).
2. G. M. Whitesides, B. Grzybowski, Self-assembly at all scales. *Science* **295**, 2418-2421 (2002).
3. S. Sacanna, W. T. M. Irvine, P. M. Chaikin, D. J. Pine, Lock and key colloids. *Nature 2010 464:7288* **464**, 575-578 (2010).
4. C. Sanchez *et al.*, Design and elaboration of colloidal molecules: an overview. *Chemical Society Reviews* **40**, 941-960 (2011).
5. Y. Wang *et al.*, Colloids with valence and specific directional bonding. *Nature 2012 491:7422* **491**, 51-55 (2012).
6. A. H. Gröschel *et al.*, Guided hierarchical co-assembly of soft patchy nanoparticles. *Nature 2013 503:7475* **503**, 247-251 (2013).
7. S. Whitelam *et al.*, Common physical framework explains phase behavior and dynamics of atomic, molecular, and polymeric network formers. *Physical Review X* **4**, 011044-011044 (2014).
8. V. W. W. Yam, V. K. M. Au, S. Y. L. Leung, Light-Emitting Self-Assembled Materials Based on d8 and d10 Transition Metal Complexes. *Chemical Reviews* **115**, 7589-7728 (2015).
9. S. Conti, M. Cecchini, Predicting molecular self-assembly at surfaces: a statistical thermodynamics and modeling approach. *Physical chemistry chemical physics : PCCP* **18**, 31480-31493 (2016).
10. S. Conti *et al.*, Perchlorination of Coronene Enhances its Propensity for Self-Assembly on Graphene. *ChemPhysChem* **17**, 352-357 (2016).
11. C. A. Palma, M. Cecchini, P. Samorí, Predicting self-assembly : from empirism to determinism. *Chemical Society Reviews* **41**, 3713-3730 (2012).
12. S. Haar *et al.*, A Supramolecular Strategy to Leverage the Liquid-Phase Exfoliation of Graphene in the Presence of Surfactants: Unraveling the Role of the Length of Fatty Acids. *Small* **11**, 1691-1702 (2015).
13. S. Bonacchi *et al.*, Surface-Induced Selection During In Situ Photoswitching at the Solid/Liquid Interface. *Angewandte Chemie International Edition* **54**, 4865-4869 (2015).
14. D. M. Packwood, P. Han, T. Hitosugi, Chemical and entropic control on the molecular

self-assembly process. *Nature Communications 2017 8:1* **8**, 1-8 (2017).
15. S. Conti, M. Cecchini, Accurate and Efficient Calculation of the Desorption Energy of Small Molecules from Graphene. *The Journal of Physical Chemistry C* **119**, 1867-1879 (2015).
16. B. Liu, Y. Qiu, E. C. Goonetilleke, X. Huang, Kinetic network models to study molecular self-assembly in the wake of machine learning. *MRS Bulletin*, 1-9 (2022).
17. X. Zeng, L. Zhu, X. Zheng, M. Cecchini, X. Huang, Harnessing complexity in molecular self-assembly using computer simulations. *Physical Chemistry Chemical Physics* **20**, 6767-6776 (2018).
18. J. Wang, K. Liu, R. Xing, X. Yan, Peptide self-assembly: thermodynamics and kinetics. *Chemical Society Reviews* **45**, 5589-5604 (2016).
19. Y. Mai, A. Eisenberg, Self-assembly of block copolymers. *Chemical Society Reviews* **41**, 5969-5985 (2012).
20. X. Zheng *et al.*, Elucidation of the key role of Pt⋯ Pt interactions in the directional self-assembly of platinum (II) complexes. *Proceedings of the National Academy of Sciences* **119**, e2116543119 (2022).
21. S. O. Nielsen, C. F. Lopez, G. Srinivas, M. L. Klein, Coarse grain models and the computer simulation of soft materials. *Journal of Physics: Condensed Matter* **16**, R481 (2004).
22. D. T. Allen, C. D. Lorenz, Molecular Scale Simulations of the Self-Assembly of Amphiphilic Molecules: Current State-of-the-Art and Future Directions. *Journal of Self-Assembly and Molecular Electronics* **3**, 1-38 (2016).
23. J. H. Prinz *et al.*, Markov models of molecular kinetics: Generation and validation. *The Journal of Chemical Physics* **134**, 174105-174105 (2011).
24. R. D. Malmstrom, C. T. Lee, A. T. Van Wart, R. E. Amaro, Application of molecular-dynamics based markov state models to functional proteins. *Journal of chemical theory and computation* **10**, 2648-2657 (2014).
25. K. A. Konovalov, I. C. Unarta, S. Cao, E. C. Goonetilleke, X. Huang, Markov state models to study the functional dynamics of proteins in the wake of machine learning. *JACS Au* **1**, 1330-1341 (2021).
26. J. D. Chodera, N. Singhal, V. S. Pande, K. A. Dill, W. C. Swope, Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics. *The Journal of Chemical Physics* **126**, 155101-155101 (2007).
27. A. C. Pan, B. Roux, Building Markov state models along pathways to determine free energies and rates of transitions. *The Journal of chemical physics* **129**, 064107 (2008).
28. N.-V. Buchete, G. Hummer, Coarse master equations for peptide folding dynamics. *The Journal of Physical Chemistry B* **112**, 6057-6069 (2008).
29. L. Zhang *et al.*, Elucidation of the Dynamics of Transcription Elongation by RNA Polymerase II using Kinetic Network Models. *Accounts of Chemical Research* **49**, 687-694 (2016).
30. H. Jiang *et al.*, Markov state models reveal a two-step mechanism of miRNA loading into the human argonaute protein: selective binding followed by structural re-arrangement. *PLoS computational biology* **11**, e1004404 (2015).

31. C. R. Schwantes, V. S. Pande, Improvements in Markov State Model Construction Reveal Many Non-Native Interactions in the Folding of NTL9. *Journal of chemical theory and computation* **9**, 2000-2009 (2013).
32. G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, F. Noé, Identification of slow molecular order parameters for Markov model construction. *The Journal of Chemical Physics* **139**, 015102-015102 (2013).
33. A. Mardt, L. Pasquali, H. Wu, F. Noé, VAMPnets for deep learning of molecular kinetics. *Nature Communications 2018 9:1* **9**, 1-11 (2018).
34. W. Chen, H. Sidky, A. L. Ferguson, Nonlinear discovery of slow molecular modes using state-free reversible VAMPnets. *The Journal of Chemical Physics* **150**, 214114-214114 (2019).
35. R. R. Coifman *et al.*, Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the national academy of sciences* **102**, 7426-7431 (2005).
36. A. L. Ferguson, A. Z. Panagiotopoulos, I. G. Kevrekidis, P. G. Debenedetti, Nonlinear dimensionality reduction in molecular simulation: The diffusion map approach. *Chemical Physics Letters* **509**, 1-11 (2011).
37. Y. Zhao, F. K. Sheong, J. Sun, P. Sander, X. Huang, A fast parallel clustering algorithm for molecular simulation trajectories. *Journal of Computational Chemistry* **34**, 95-104 (2013).
38. T. F. Gonzalez, Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science* **38**, 293-306 (1985).
39. D. Sculley, Web-scale k-means clustering. *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, 1177-1178 (2010).
40. A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks. *Science* **344**, 1492-1496 (2014).
41. P. Deuflhard, M. Weber, Robust Perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications* **398**, 161-184 (2005).
42. P. Deuflhard, W. Huisinga, A. Fischer, C. Schütte, Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications* **315**, 39-59 (2000).
43. S. Röblitz, M. Weber, Fuzzy spectral clustering by PCCA+: Application to Markov state models and data classification. *Advances in Data Analysis and Classification* **7**, 147-179 (2013).
44. F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, T. R. Weikl, Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences of the United States of America* **106**, 19011-19016 (2009).
45. W. E, E. Vanden-Eijnden, Towards a Theory of Transition Paths. *Journal of Statistical Physics 2006 123:3* **123**, 503-523 (2006).
46. P. Metzner, C. Sch¨utte, S. Sch¨utte, E. Vanden-Eijnden, Transition Path Theory for Markov Jump Processes. *Multiscale Modeling & Simulation* **7**, 1192-1219 (2009).
47. C. Li, Z. Liu, E. C. Goonetilleke, X. Huang, Temperature-dependent kinetic pathways of heterogeneous ice nucleation competing between classical and non-classical

nucleation. *Nature Communications* **12**, (2021).
48. X. Zeng *et al.*, Improving the productivity of monodisperse polyhedral cages by the rational design of kinetic self-assembly pathways. *Physical Chemistry Chemical Physics* **20**, 10030-10037 (2018).
49. X. Tang *et al.*, Optimal Feedback Controlled Assembly of Perfect Crystals. *ACS Nano* **10**, 6791-6798 (2016).
50. X. Tang, M. A. Bevan, M. A. Grover, The construction and application of Markov state models for colloidal self-assembly process control. *Molecular Systems Design & Engineering* **2**, 78-88 (2017).
51. X. Zeng *et al.*, Elucidating dominant pathways of the nano-particle self-assembly process. *Physical Chemistry Chemical Physics* **18**, 23494-23499 (2016).
52. X. Zheng *et al.*, Kinetics-Controlled Amphiphile Self-Assembly Processes. *Journal of Physical Chemistry Letters* **8**, 1798-1803 (2017).
53. J. Weng, M. Yang, W. Wang, X. Xu, Z. Tian, Revealing Thermodynamics and Kinetics of Lipid Self-Assembly by Markov State Model Analysis. *Journal of the American Chemical Society* **142**, 21344-21352 (2020).
54. K. Imamura, T. Yamamoto, H. Sato, Coarse-grained modeling of nanocube self-assembly system and transition network analyses. *Chemical Physics Letters* **742**, 137135-137135 (2020).
55. M. Ghorbani, S. Prasad, J. B. Klauda, B. R. Brooks, GraphVAMPNet, using graph neural networks and variational approach to Markov processes for dynamical modeling of biomolecules. *The Journal of Chemical Physics* **156**, 184103 (2022).
56. T. Xie, A. France-Lanord, Y. Wang, Y. Shao-Horn, J. C. Grossman, Graph dynamical networks for unsupervised learning of atomic scale dynamics in materials. *Nature Communications* **10**, (2019).
57. H. Wu, F. Noé, Variational Approach for Learning Markov Processes from Time Series Data. *Journal of Nonlinear Science* **30**, 23-66 (2020).
58. K. T. Schütt, H. E. Sauceda, P. J. Kindermans, A. Tkatchenko, K. R. Müller, Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics* **148**, 241722 (2018).
59. A. Mardt, L. Pasquali, F. Noé, H. Wu, Deep learning Markov and Koopman models with physical constraints. *Proceedings of Machine Learning Research* **107**, 451-475 (2020).
60. Z.-W. Li, Y.-L. Zhu, Z.-Y. Lu, Z.-Y. Sun, Supracolloidal fullerene-like cages: design principles and formation mechanisms. *Physical Chemistry Chemical Physics* **18**, 32534-32540 (2016).
61. Z.-W. Li, Y.-L. Zhu, Z.-Y. Lu, Z.-Y. Sun, A versatile model for soft patchy particles with various patch arrangements. *Soft Matter* **12**, 741-749 (2016).
62. A. Mardt, T. Hempel, C. Clementi, F. Noé, Deep learning to decompose macromolecules into independent Markovian domains. *Nature Communications* **13**, 7101 (2022).