Markov State Model Approach to Simulate Self-Assembly

Anthony Trubiano* and Michael F. Hagan†
Martin Fisher School of Physics, Brandeis University, Waltham, Massachusetts 02454, USA
(Dated: October 6, 2024)

Computational modeling of assembly is challenging for many systems because their timescales can vastly exceed those accessible to simulations. This article describes the MultiMSM, which is a general framework that uses Markov state models (MSMs) to enable simulating self-assembly and self-organization of finite-sized structures on timescales that are orders of magnitude longer than those accessible to brute force dynamics simulations. As with traditional MSM approaches, the method efficiently overcomes free energy barriers and other dynamical bottlenecks. In contrast to previous MSM approaches to simulating assembly, the framework describes simultaneous assembly of many clusters and the consequent depletion of free subunits or other small oligomers. The algorithm accounts for changes in transition rates as concentrations of monomers and intermediates evolve over the course of the reaction. Using two model systems, we show that the MultiMSM accurately predicts the concentrations of the full ensemble of intermediates on timescales required to reach equilibrium. Importantly, after constructing a MultiMSM for one system concentration, yields at other concentrations can be approximately calculated without any further sampling. This capability allows for orders of magnitude additional speed up. In addition, the method enables highly efficient calculation of quantities such as free energy profiles, nucleation timescales, flux along the ensemble of assembly pathways, and entropy production rates. Identifying contributions of individual transitions to entropy production rates reveals sources of kinetic traps. The method is broadly applicable to systems with equilibrium or nonequilibrium dynamics, and is trivially parallelizable and thus highly scalable.

I. INTRODUCTION

The self-assembly of basic subunits into larger, more complex structures is fundamental to life. Critical functions of cells and pathogens are performed by assembled structures with a well-defined finite-size and architecture such as the outer shells (capsids) of viruses [1–8] or bacterial microcompartments [9–17], cytoskeletal filaments [18–20], and ordered protein layers on bacteria exteriors [21]. Self-assembly is also transforming nanotechnology, where designing synthetic building blocks that are preprogrammed to form particular structures is enabling scalable bottom-up synthesis of materials with desirable properties [22–44].

Since it is an inherently out-of-equilibrium process, understanding or designing self-assembly requires detailed knowledge of assembly intermediates and the dynamical transitions among them. Computational modeling is an essential tool for revealing such assembly pathways, since most intermediates are too transient to characterize in experiments. In particular, molecular dynamics simulations of tractable models for subunits have revealed numerous insights about the principles controlling assembly (e.g. [7, 45–61], and references in [62]). However, simulating assembly dynamics at experimentally relevant conditions is intractable for many models, since assembled structures are much larger than their components and form on timescales that are orders of magnitude beyond computational limitations. One approach to overcome this limitation is to coarse-grain over length and/or timescales; e.g. by treating subunit association at the level of reaction diffusion equations [63–65] or kinetic Monte Carlo approaches [66–75]. However, these approaches require alternative assumptions and restrictions on validity, and can also become intractable for systems with sufficiently large sizes or free energy barriers.

This article describes a Markov state model (MSM) framework that can overcome limitations on accessible timescales for a broad array of self-assembly and self-organization systems. The algorithm reduces computational times by orders of magnitude while describing the time-dependent concentrations of subunits and the complete ensemble of assembly intermediates and products. This capability enables dynamical particle-based simulations of systems with unprecedented size and complexity, at experimentally relevant conditions. The framework can also be applied to reaction-diffusion [76] and kinetic Monte Carlo approaches. Further, MSMs enable highly efficient analysis of the resulting simulation data. We focus on self-limited assembly examples that preferentially terminate at a finite-size [8], but note that the method can also be applied to unlimited assembly examples such as crystallization or extended ribbons or sheets [75, 77–79] if the structures are limited to a maximum size.

MSMs are a powerful approach to simulate dynamics on long timescales; by performing short simulations to estimate transition rates among system configurations, one can construct an MSM that accurately describes dynamics on timescales that are orders of magnitude longer than the individual simulations [80–99]. Importantly, because only short trajectories are required to estimate transition rates, MSMs efficiently harvest trajectories that involve barrier crossings or other dynamical bottlenecks [80–99]. In contrast to many other non-Boltzmann sampling techniques for rare events (e.g. [100–117]), MSMs can be used to study reactions with multiple barriers and relevant transition pathways, and are applicable to nonequilibrium systems. MSMs also provide a means to coarse-grain complex dynamical processes into reduced-order forms that facilitate identifying key slow degrees of freedom

* trubiano@brandeis.edu
† hagan@brandeis.edu

and corresponding mechanisms. Furthermore, MSMs can enable designing non-equilibrium assembly protocols that can accelerate assembly and increase selectivity of a specific target state by orders of magnitude in comparison to equilibrium processes [118–122].

In contrast to previous MSM approaches to self-assembly that pre-assume the state space and transition rates [123–134], we seek a framework in which the state space and transition rates are computed directly from dynamical simulations, and the accuracy of the resulting MSM (including the validity of the Markov assumption) can be directly tested against microscopic dynamics. While several approaches have been developed to construct MSMs from particle-based assembly simulations [122, 135–140], these algorithms are designed to track individual assembling clusters evolving under constant conditions such as the concentration of free subunits. Thus, they cannot describe a typical experiment in which a fixed total number of subunits assemble into many structures. In this case, the concentrations of intermediates and free subunits, and thus the transition rates, continuously evolve over time. Moreover, some transitions involve association or dissociation of oligomers or larger intermediates. Therefore, although sophisticated approaches have been recently developed to design optimal assemblies and compute free energy landscapes [141–152], to our knowledge, there is no existing enhanced sampling method that can comprehensively model such self-assembly experiments.

In this article we present the MultiMSM approach, which provides a complete description of assembly reactions, accounting for changes in transition rates as concentrations evolve, as well as association between intermediates. Using two model systems, we show that that the MultiMSM algorithm accurately predicts the depletion of monomers and the ensemble of resulting intermediate and target assembly species, on the long timescales required for reactions to reach equilibrium. While brute-force dynamics simulations with related models have been limited to restricted parameter ranges, such as high subunit concentrations [48, 56, 153–155], the MultiMSM approach enables simulation over a broad range of experimentally relevant parameter values. In particular, the algorithm reduces simulation times by orders of magnitude for systems with large nucleation barriers, which are typically required for productive assembly at experimental conditions [8, 156–160].

Crucially, once the MultiMSM has been constructed for one value of the total subunit concentration, assembly dynamics can be simulated over a wide range of lower concentrations without any additional sampling. This enables representing a typical experiment in which assembly is performed over a range of subunit concentrations, but with the computational cost of a single subunit concentration. Further, the method provides a detailed analysis of assembly mechanisms by computing quantities such as the free energy landscape, nucleation timescales, committor probabilities and flux along different assembly pathways, and entropy production rates. The latter quantify the extent to which a reaction is out of equilibrium and identify sources of kinetic trapping that impede productive assembly. These capabilities allow analyzing data from particle-based assembly simulations in unprecedented ways.

We provide an open-source Python library [161, 162] which performs all calculations required to construct MultiMSMs and the analysis described in this work, from simulations performed with the open-source molecular dynamics simulation package HOOMD-blue [163, 164]. The library can be readily generalized to other software packages.

II. MODEL SYSTEMS

We first describe the two model self-assembly systems that we use to test and demonstrate our MultiMSM approach. Our first example, dodecahedral capsid assembly from pentagonal subunits, is sufficiently tractable that brute-force dynamics simulations can be performed on relatively long-time scales to stringently test the MultiMSM results. Our second example, $T = 3$ capsid assembly from triangular subunits, is more complex and computationally expensive, and shows that the MultiMSM method scales well for complicated problems with a large state space.

For all models and results presented in this work, we give energies in units of the thermal energy $k_{\mathrm{B}}T$ and lengths and concentrations in units of l_0 and l_0^{-3} respectively, where l_0 is related to the subunit size for each model (see Appendix A).

A. Dodecahedron Assembly

Our model subunit (Fig. 1a) is adapted from previous studies of dodecahedral capsids, including assembly of empty capsids and assembly around RNA and synthetic polyelectrolytes [53, 154, 167–169]. The subunit is a rigid body consisting of five attractor sites, placed at the vertices of a regular pentagon, that have attractive interactions through a Morse potential with well-depth ϵ_{11}. Each subunit also has a 'Top' and 'Bottom' pseudoatom; Top-Top and Top-Bottom pairs on nearby subunits each have repulsive Weeks-Chandler-Anderson (WCA) interactions [170]. Top-Top interactions drive subunit-subunit binding angles consistent with a dodecahedron ($116.57°$), while Top-Bottom interactions suppress subunit-subunit binding in inverted orientations [53, 154, 167]. We perform simulations in three distinct assembly regimes by setting $\epsilon_{11} \in \{5.0, 5.5, 6.0\}$ with total subunit concentration $c_0 = 0.0156$, which spans from almost no assembly to rapid assembly.

B. T=3 Capsid Assembly

Our second example model was previously developed as a simplified representation of an experimental system in which DNA origami forms rigid triangular subunits that assemble into $T = 3$ icosahedral capsids [165, 166]. The model builds on extensive previous simulations of capsid assembly [7, 48–60, 171, 172]. The model subunit excluded volume shape is represented by three layers of 'excluder' atoms arranged so

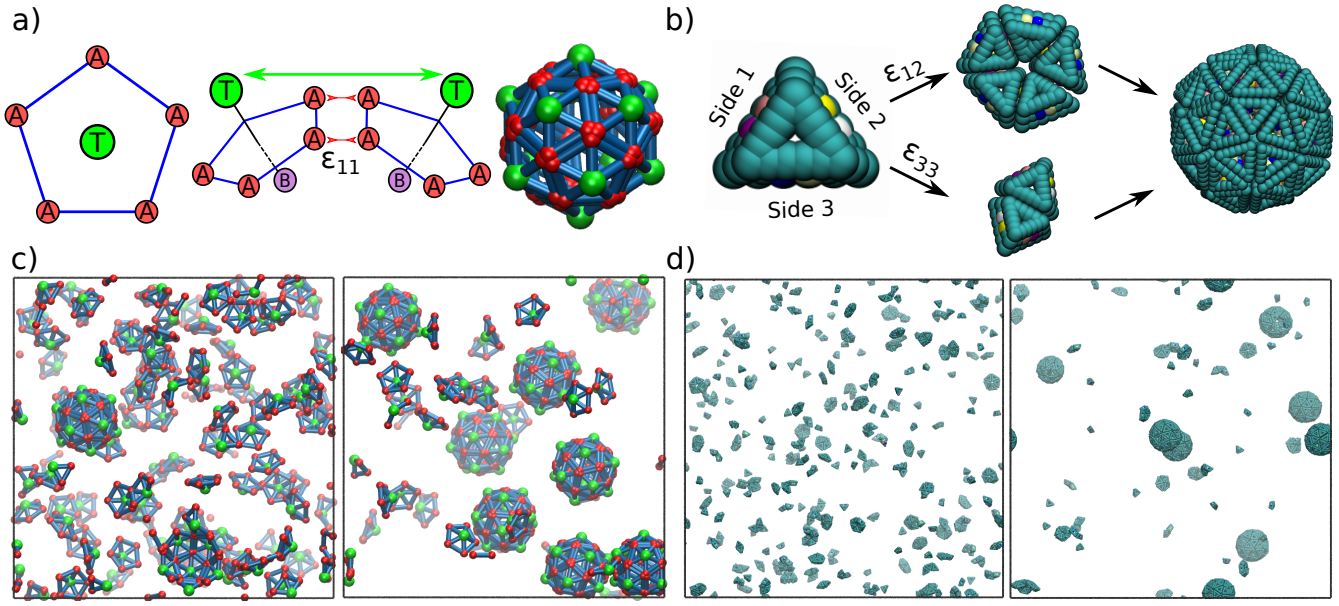

FIG. 1. Schematics of the subunits and their interactions for the two model self-assembly systems. **(a)** (Left) The subunit is a rigid body with Attractors at the vertices of a pentagon and a 'Top' and 'Bottom' pseudoatom above and below the vertex plane. (Middle) Subunit-subunit interactions. Attractor-Attractor interactions drive subunit association, with binding affinity parameter ϵ_{11}. Top-Top repulsions result in a preferred subunit-subunit binding angle of $116.57°$, making a dodecahedral capsid the ground state. (Right) A snapshot of an assembled capsid from a simulation. **(b)** (Left) The subunit is a rigid triangular body, motivated by recent DNA origami experiments [165, 166]. The cyan pseudoatoms enforce excluded volume, and complementary pairs of attractor beads (other colors) on each side drive association between Side 1 – Side 2 and Side 3 – Side 3 pairs. (Middle) These interactions, with binding affinity parameters ϵ_{12} and ϵ_{33}, respectively stabilize pentamers and dimers. (Right) A snapshot of a complete $T = 3$ capsid from a simulation. **(c)** Snapshots of the simulation box during dodecahedron assembly with $\epsilon_{11} = 5.5$ at early times (left) and late times (right). **(d)** Same as (c) for $T = 3$ capsid assembly with $\epsilon_{12} = 11$ and $\epsilon_{33} = 8$.

that the edges have a bevel angle consistent with an icosahedron ($156.72°$). Each excluder interacts with all pseudoatoms through a WCA potential. Subunit-subunit attractions in the experimental system are driven by DNA blunt-end stacking and hybridization of single-stranded DNA molecules on complementary subunit edges. In the computational model, these short-ranged interactions are represented by placing two 'attractor' atoms on each subunit edge, on the middle layer of excluders. Complementary pairs of attractors interact through a Lennard-Jones potential. To match the experimental $T = 3$ system, attractors on Side 1 and Side 2 of two interacting subunits are complementary with binding energy (Lennard-Jones potential well-depth) ϵ_{12}, and attractors on Side 3 are complementary with binding energy ϵ_{33}. Attractors that are not complementary interact through a repulsive WCA potential. Fig. 1b shows a representation of the triangular subunit, the preferred intermediate for each interaction, and an example of the fully assembled capsid in simulation. Despite the simplicity of the model, Wei et al. 2024 [166] found that the simulation results semi-quantitatively match experimental observations of capsid assembly dynamics.

Simulations and experiments in Wei 2024 [166] showed that sufficiently imbalanced values of ϵ_{12} and ϵ_{33} lead to hierarchical assembly pathways, since ϵ_{12} and ϵ_{33} respectively stabilize intra-pentamer and intra-dimer interactions. Stronger ϵ_{12} leads to pentamer-biased assembly pathways, in which subunits rapidly form pentamers, which in turn undergo assembly into capsids; whereas stronger ϵ_{33} leads to dimer-biased pathways, with rapid formation of dimers and their subsequent assembly of capsids. In this work, we focus on parameters that lead to pentamer-biased assembly pathways, $\epsilon_{12} = 11$ and $\epsilon_{33} = 8$, with total subunit concentration $c_0 = 1.7 \times 10^{-4}$.

All dynamics simulations described in this work were performed with HOOMD-blue [163]. Full simulation details can be found in Appendix A.

III. METHODS

Here we review how a traditional MSM is constructed and then describe the protocol to construct and use the MultiMSM. The procedure is separated into four steps: selection of discrete states, data processing and transition counting, monomer fraction discretization and transition matrix construction, and model evaluation and prediction.

Our python libraries to construct MultiMSMs from HOOMD simulations and perform all the calculations described below are available on Github [161, 162].

A. Constructing an MSM

We first review how a traditional MSM is constructed from relatively short, unbiased simulations. Configurations from these simulations are partitioned into discrete states, which are defined so that conformations that inter-convert rapidly are within the same state. Transitions between configurations in different states occur on longer timescales, ensuring that the model will behave Markovian on timescales longer than a 'lag time' τ. The state probability vector, $\vec{p}^n$, a row vector giving the probability distribution over the discrete states at timepoint n, is then given by the forward Kolmogorov equation

$$\vec{p}^{n+1} = \vec{p}^n \boldsymbol{P}, \tag{1}$$

with $\boldsymbol{P}$ as the transition matrix, which gives the probability of a transition between each pair of states after a lag time τ. The transition matrix is estimated by counting the number of transitions separated by a lag time of τ in the simulation data and normalizing the rows into a probability distribution.

Several statistical tests exist for selecting an appropriate lag time to ensure Markovianity [173–175]. The most commonly used approach is to compute the implied timescales of the transition matrix as a function of τ, given by $t_i(\tau) = -\frac{\tau}{\log \lambda_i(\tau)}$, where λ_i are the eigenvalues of the transition matrix $\boldsymbol{P}(\tau)$. For the Markov assumption to hold, these timescales should be approximately independent of τ. Note that this condition is not always sufficient to guarantee Markovianity, which further requires that the eigenvectors of the transition matrix be independent of τ [85, 173, 174]. A more comprehensive and reliable approach for assembly systems is to test that the MSM prediction for the assembly time distribution, which depends on all of the implied timescales, becomes independent of lag time above a threshold value of τ [136].

B. Selection of Discrete States

A crucial aspect of constructing an accurate MSM is choosing a mapping of configurations into discrete states that ensures sufficient separation of timescales to justify the Markovian approximation. That is, pairs of configurations within a state inter-convert much more rapidly than pairs of configurations in different states. In this section we focus on characterizing the state of an individual assemblage (cluster); we address multiple clusters in section III C. We have previously shown that a general state decomposition for assembly is enabled by mapping an assemblage to an undirected graph, with nodes and edges respectively corresponding to subunits and 'bonds' (subunit-subunit interactions) [136, 140]. Alternative approaches based on pairwise distances between subunits and other structural properties have also been used [96, 98, 136, 139, 176–182]. However, these descriptions can be simplified, and the size of the state space significantly decreased, with a simplified state definition that characterizes the number of subunits and bonds within an assemblage [122, 136, 154]. We use the latter approach for both examples in this article; we define a state as $\mathcal{S} = (\mathcal{N}, \mathcal{B})$ where $\mathcal{N}$ is the number of subunits in the configuration and $\mathcal{B}$ is a count of the number of each type of bond present in the configuration. For the examples we consider in this work, $\mathcal{B}$ is a scalar for the dodecahedron assembly since there is only one type of bond, but is a vector with two components for the $T = 3$ capsid assembly, since there are two interaction types (the Side1-Side2 bond and the Side3-Side3 bond, see Fig. 1b). We find that these coordinates are sufficient to accurately characterize the dynamics of both systems studied here. However, for systems in which such simple descriptions cannot uniquely define clusters, the more complex discretizations mentioned above, or data-driven discretization approaches [183, 184] should be used. Further, the choice of discretization should be tested as described in Appendix C 1.

Since a bond refers to a pair of sufficiently strongly interacting subunits, it must be defined based on a threshold. In this work, we use cutoff distances between corresponding particle types to define a bond. See Appendix A 3 for details on the bond definition for each system.

C. Processing Simulation Data and Counting Transitions

We seek to model self-assembly dynamics in the canonical (NVT) ensemble. Since there may be many clusters undergoing different stages of nucleation and growth at the same time, we must compute the time evolution of the joint probability distribution of all cluster types j. A complete state decomposition would classify the assembly configuration of every cluster at a given time point. However, for a large system with many clusters the number of such states would be intractable. Therefore, we use the independent Markov decomposition (IMD) method [185], in which each cluster is considered as a quasi-independent local subsystem. However, note that the clusters are not strictly independent since pairs of clusters can merge or split during assembly. In this framework, the state probability distribution at frame i is given by [185]

$$\vec{p}_i = \vec{p}_i^{\,1} \otimes \vec{p}_i^{\,2} \otimes \ldots \vec{p}_i^{\,n_{\text{types}}} \tag{2}$$

where $\vec{p}_i^{\,j}$ is the probability distribution for the state of cluster j at frame i, n_{types} is the total number of cluster-types, and $\otimes$ is the Kronecker product [185, 186]. Here, each cluster is defined according to the state decomposition described in section III B.

For self-assembly, it is useful to cast the cluster probabilities as concentrations

$$c_i^j = \mathcal{N}^j n_i^j / V \tag{3}$$

where $\mathcal{N}^j$ is the number of subunits in cluster-type j, n_i^j is the number of such clusters at a given frame, and V is the volume. It is important that we use mass-weighted concentrations to maintain the constant total subunit concentration

$$c_0 = \sum_j c_i^j \quad \forall i. \tag{4}$$

Using number-weighted concentrations would result in a probability that is normalized to the cluster distribution, which depends on time and interaction parameters, whereas the mass-weighted concentrations maintain a normalization that depends only on the control parameter c_0.

The transition matrix elements can be estimated from the ensemble of short simulations by recording the number of each cluster type j at each simulation window, and then computing the number of transitions between all cluster types, ranging from monomer to dimer transitions to association/dissociation of larger intermediates and complete capsids, as a function of lag time. Importantly, to maintain the mass-weighted cluster distribution, the transition counts need to be weighted by the number of subunits involved in each transition. For example, if a cluster j with $\mathcal{N}^j$ subunits transitions to a cluster l with $\mathcal{N}^l > \mathcal{N}^j$ subunits, then all $\mathcal{N}^j$ of those subunits undergo the transition. That is, the transition $j \to k$ occurs $\mathcal{N}^j$ times. Intuitively, this can be thought of as viewing transitions from the perspective of individual subunits rather than clusters (see Fig. 2).

Procedure for counting transitions. For each frame in the simulation, we group the subunits into clusters, with a cluster defined as any collection of more than one subunit that is bonded together. On the first frame, all clusters are given an ID. On subsequent frames, we check if any newly found clusters are derived from existing clusters, either through merging or splitting of sub-clusters, and update any matching existing cluster with the new configuration. In the case of splitting, we record what the parent cluster was and form a new cluster. Any cluster that was not derived from an existing one is assigned a new ID.

We treat monomers separately, tracking their addition and removal as a separate time series. For example, if a 10-mer loses two subunits, but they do not form a dimer after dissociating, we record two transitions to monomers. If they do form a dimer, we record nothing in the monomer time series, since no monomers are involved in this transition, but record a transition to the dimer. To track monomer-to-monomer transitions, we store the IDs of all subunits that are not bonded in frame i in a list M_i. To determine the number of monomer-to-monomer transitions after a lag time k, we take the cardinality of the intersection of these lists, $|M_i \cap M_{i+k}|$. Finally, we also save the monomer fraction, $f_1(i) = |M_i|/N_{\text{tot}}$, where N_{tot} is the total number of subunits, at every frame, and augment any transitions that occur in that frame with this value.

We then construct the transition count matrix as follows. We loop over every cluster's time series of configurations and extract every pair of configurations separated by a lag time k, incrementing the corresponding entry of the count matrix by 1. Then, to ensure mass-weighting, we multiply that transition count by $\min(\mathcal{N}_t, \mathcal{N}_{t+k})$.

D. Monomer Fraction Discretization and Construction of MSMs

In this section we describe how to construct the transition matrices for the different monomer concentrations that arise as subunits are depleted during an assembly reaction.

The first step is to discretize the monomer fraction in the interval $[0, 1]$, where the monomer fraction is defined as $f_1 = c^1/c_0$ with c^1 the concentration of monomers (free subunits). The discretization contains $N + 1$ intervals, $D_N = (0, d_1, d_2, \cdots, d_N, 1)$. We then estimate the transition matrix for each of these intervals, following the approach described in Section III C. Note that, by construction, the state space is the same for each concentration. Although we will describe a smoothing procedure in Section III E 1 to interpolate transition matrices between interval edges, the accuracy of the MultiMSM depends strongly on the choice of discretization. While increasing the resolution leads to higher accuracy, it also increases the amount of total sampling required for convergence of the MSMs. Thus, the size of each interval must be chosen such that a sufficiently large number of relevant transitions are sampled. Sampling efficiency can be improved using adaptive sampling [187–194].

We have employed a heuristic, but adaptive discretization procedure, in which intervals are defined to contain monomer fraction values that give rise to similar dynamics. For example, in most cases monomer fractions between 0.95 and 1 correspond to the initial stage of a reaction when most transitions correspond to monomer-dimer association or dissociation. In contrast, when the monomer fraction approaches its infinite-time limit (e.g. the equilibrium monomer concentration for reversible assembly), most transitions will involve large intermediates. Ensuring that each interval separates different types of dynamics improves the accuracy of the MultiMSM. Appendix B describes a systematic method for refining the monomer fraction discretization in cases where there is a ground truth to compare to. Additionally, we propose an error-based refinement in Appendix C 1 that can be used even when a ground truth is infeasible to compute, which helps refine the monomer fraction discretization as well as identify bins that can benefit the most from additional sampling.

We compute the MSM in each interval as follows. We initialize a sparse count matrix to store the number of observed transitions between each pair of states on each of these intervals. Using the output of our cluster analysis at a given lag time (see section III C), we identify the monomer fraction at which each transition occurred and increment the count for the corresponding transition and monomer fraction interval. Once all transitions have been recorded, we compute the probability transition matrix for each interval by normalizing the rows of the count matrices to sum to 1.

Choosing an appropriate lag time for the MultiMSM requires testing that the choice is appropriate for the transition matrix in each monomer fraction bin of the discretization. The same statistical tests used for traditional MSMs (see section III A) can be used for each component MSM here. We ensured that, given the true starting distribution of states on a discretization interval, the correct final distribution on that interval was predicted by the corresponding transition matrix. In the absence of a ground truth distribution, one can verify that the predicted assembly distributions become independent of lag time. For simplicity, we use the same lag time for all discretization intervals, but it would be straightforward to ex-

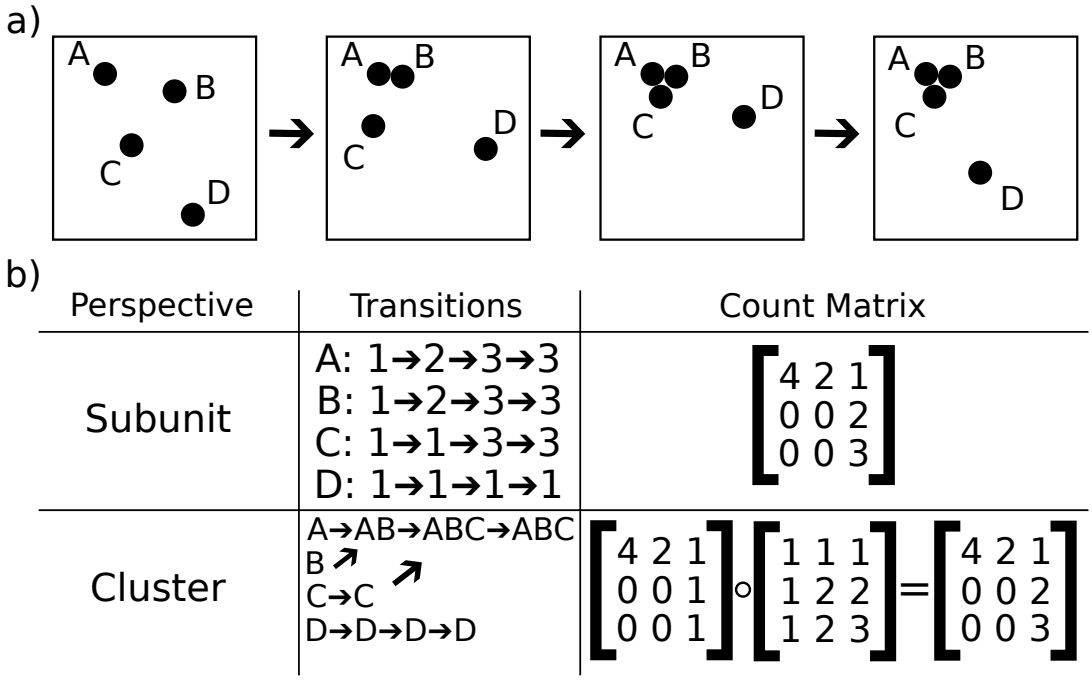

FIG. 2. A simple assembly example showing two equivalent approaches to count transitions. **(a)** A time series of four disk shaped particles with attractive interactions undergoing clusterization. **(b)** A table detailing how to count transitions from both a subunit and cluster perspective for this example. In the subunit perspective, a time series of the cluster size containing each subunit is computed, and a count matrix is built from the number of times each transition is observed. In the cluster perspective, only merging and splitting of clusters is tracked. The count matrix is built by tallying the number of times a cluster of size i transitions to a cluster of size j, and then multiplied elementwise by $W_{ij} = \min(i, j)$ to get the mass-weighted counts (i.e in the subunit perspective).

tend the implementation to nonuniform lag times if needed.

E. Calculating the yield as a function of time

The forward equation to compute yields for the MultiMSM is a straightforward extension of the forward equation for a standard MSM (Equation (1)). For the MultiMSM, the equation has the same form, but involves a collection of transition matrices. At each timepoint n we use the transition matrix corresponding to the current monomer fraction. By construction, we store the monomer fraction in the first component of the probability distribution, p_0^n, and the mass-weighted fraction for each intermediate is given by $p_j^n = c_j^n / c_0$, with c_j and c_0 given by Eqs. (3) and (4). Formally, we write

$$\vec{p}^{n+1} = \vec{p}^{n} \boldsymbol{P}_{m^n}, \quad m^n = \text{index}(p_0^n), \tag{5}$$

where the index operator converts a value in $[0, 1]$ to its corresponding interval in the discretization. We perform this calculation for each monomer increment, which gives the full distribution of intermediate concentrations as a function of time for each monomer interval.

Note that some of the methods typically used to efficiently solve the forward equation, such as computing the spectral decomposition or pre-computing large powers of the transition matrix, cannot be directly used here since we do not know a priori at what timepoints the monomer fraction will cross the discretization boundaries and change the transition matrix. The most straightforward approach is to solve Eq. (5) iteratively via vector-matrix multiplication for each timepoint. However, it is also possible to pre-assume the monomer concentration as a function of time, and then iteratively apply an efficient approach such as spectral decomposition, in which the computed monomer concentration is updated at each iteration.

1. *Smoothing Solutions*

Solving Eq. (5) following the above approach typically leads to solutions that are well behaved within each interval of the discretization, but have 'jumps' (abrupt changes in slope) at time steps when a discretization boundary is crossed. These jumps reflect the abrupt change in the transition dynamics due to changing the transition matrix. The jumps introduce small errors in the solution that can accumulate and reduce the accuracy of the MultiMSM prediction at long times. To solve this problem, we describe a smoothing procedure to continuously interpolate between the two transition matrices across a discretization boundary.

Consider the two intervals closest to 1, $I_1 = (d_{N-1}, d_N)$ and $I_2 = (d_N, 1)$. Let $L_1 = d_N - d_{N-1}$ and $L_2 = 1 - d_N$ be the length of each interval and let $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ be the transition matrices on intervals 1 and 2, respectively. Since neighboring intervals can have significantly different lengths, we

define a smoothing region that is agnostic of absolute interval sizes. Let χ be the fraction of each interval that is used to smoothly interpolate between them. The transition region will then begin at $a = d_N - \chi L_1$ and end at $b = d_N + \chi L_2$. If the monomer fraction falls within $[a, b]$, we construct a linear combination of each interval's transition matrix to use at that value. We choose the weights proportional to where in the region the monomer fraction falls, with an even split if the monomer fraction is precisely d_N. In general, we compute

$$\alpha(f_1) = \begin{cases} \frac{1}{2}\frac{f_1 - a}{d_N - a} & \text{if } a \leq f_1 \leq d_N, \\ \frac{1}{2} + \frac{1}{2}\frac{f_1 - d_N}{b - d_N} & \text{if } d_N \leq f_1 \leq b, \end{cases} \tag{6}$$

where f_1 is the current monomer fraction. We then construct the final transition matrix as

$$\boldsymbol{P}_m = (1 - \alpha(f_1))\boldsymbol{P}_1 + \alpha(f_1)\boldsymbol{P}_2. \tag{7}$$

Since each individual transition matrix is normalized and their coefficients sum to 1, $\boldsymbol{P}_m$ is also normalized and thus a valid transition matrix.

This smoothing procedure works remarkably well for $\chi \in [0.2, 0.3]$. We include χ as an optional parameter to our solvers, with a default value of 0.25. Setting a value of 0 turns off all smoothing and solves Eq. (5) as stated.

IV. RESULTS AND TESTING OF THE MULTIMSM

We constructed a MultiMSM using simulation data for each of the model systems described in Section II. For each example, we solved Eq. (5) to predict the time-dependent yields of each discrete state. See Section A 5 for a detailed description of how many trajectories were used to build each MSM, adaptive sampling strategies, and the values used for all parameters to the models, such as the monomer fraction discretization.

A. Dodecahedron Capsids

Fig. 3 shows example results of assembly dynamics predicted by the MultiMSM for the dodecahedron system at several values of the binding energy ϵ_{11}, compared against brute-force dynamics. Fig. 3a shows results for the strongest interactions, $\epsilon_{11} = 6$, for which assembly is rapid and thus the MultiMSM results can be directly compared against brute-force dynamics simulations on all relevant timescales. Note that for this relatively strong binding energy, it is common for the 12-th subunit to bind to an 11-mer in the wrong orientation, and then become trapped for long times. We refer to this off-target, metastable configuration as a 'dangler'. The smooth lines (blue, green, purple) in Fig. 3a show the MultiMSM prediction of the mass-weighted yields of the monomer, dodecahedra, and the size-12 structures (dodecahedron and dangler), respectively, while the noisy lines show estimates from brute-force dynamics. The agreement is excellent, with the largest differences being only a few percent at intermediate times, while the short- and long-time behaviors show even closer agreement.

At early times about half of the 12-mers are danglers. These off-pathway intermediates gradually anneal into the target dodecahedron structure.

Figs. 3b,c show the MultiMSM predictions for lower binding energy values $\epsilon_{11} = 5.5$ and 5.0 respectively. Notably, the weaker subunit-subunit attractions result in significantly longer assembly timescales (about $20\times$ and $500\times$ respectively). Note that the dangler intermediates do not occur for these weaker binding energies, and thus we focus on the most common structures (monomers and dodecahedrons). Fig. 3b shows that the MultiMSM predictions closely match the brute-force dynamics predictions, even on the long timescales required for this system to approach equilibrium. We provide further evidence that this model is accurate over longer timescales in Section V A and Fig. 7b. In Fig. 3c we cannot directly compare the MultiMSM predictions against brute-force dynamics across all timescales, as the simulations would take about 2 GPU-months per trajectory. We do perform a comparison over accessible timescales ($T_\mathrm{f} = 2.5 \times 10^5 t_0$) and see good agreement (see supplement Fig. S7) [51, 53, 80, 195–204].

To further test the accuracy of the MultiMSM prediction, we note that the MultiMSM predictions (see Fig. 4), brute-force dynamics simulations, and previous modeling results (e.g. [153, 202, 205, 206]) show that intermediates are present at extremely low concentrations for weak binding energies such as $\epsilon_{11} = 5.0$. Thus, Fig. 3c also compares $1 - f_1$ with the dodecahedron fraction, showing that these two results are within a few percent for all times as expected for small intermediate concentrations. Additionally, Fig. 4 shows the MultiMSM results for the full cluster-size distribution as a function of time for $\epsilon_{11} = 6$ and $\epsilon_{11} = 5$. In the case of stronger binding, there is a broad distribution of detectable intermediate sizes during the rapid assembly phase. In contrast, weak binding results in approximately two-state kinetics — only dodecahedra and monomer occur at high concentrations, with low concentrations of dimers and trimers, and trace amounts of other transient intermediates. The snapshots in Fig. 4c,d show representative system configurations during the rapid assembly phase for each case. For $\epsilon_{11} = 6$ (Fig. 4c) we see dodecahedra and monomers coexisting with intermediates of various sizes, while for $\epsilon_{11} = 5$ (Fig. 4d) we observe two dodecahedra and monomers along with a few transient dimers.

These results demonstrate a powerful aspect of MSMs that is preserved in our MultiMSM approach; the simulations used to construct the model are all of length $0.2 \times 10^5 t_0$, which is orders of magnitude smaller than the relevant assembly timescales.

B. T=3 Capsids

Fig. 5a shows MultiMSM predictions (smooth curves) for $T = 3$ capsid assembly dynamics, with results shown for monomers, complete capsids, and 'near-capsids' which include any structure with 56 or more subunits (including complete capsids with 60 subunits). These predictions are

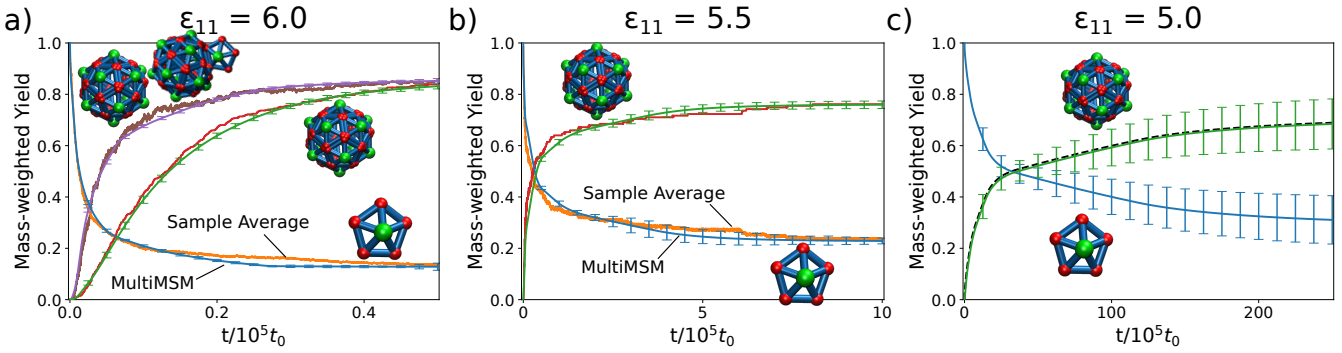

FIG. 3. MultiMSM predictions for dodecahedron capsid assembly dynamics at three values of the binding affinity parameter ϵ_{11}, compared against results from brute-force dynamics on accessible timescales. Note the increasing timescale on the x-axis as binding affinity decreases. **(a)** Strong affinity $\epsilon_{11} = 6$. The blue, green, and purple curves denote the MultiMSM predictions for mass-fraction of monomer, dodecahedron, and all size 12 structures (capsids and 'danglers'), respectively, with representative structures from simulations labeling each curve. The noisy curves (orange, red, brown) show the same mass-fractions estimated from 50 independent brute-force dynamics trajectories. **(b)** Moderate affinity $\epsilon_{11} = 5.5$. MultiMSM predictions for the monomer (blue) and dodecahedron (green) mass fractions; danglers do not form at this binding affinity. The noisy curves (orange, red) show the same mass-fractions estimated from 20 independent brute-force dynamics trajectories. **(c)** Weaker affinity $\epsilon_{11} = 5.0$. MultiMSM predictions are shown for monomers and capsids. The dashed line is $1 - f_1$ with f_1 the monomer fraction. The total subunit concentration for (a)-(c) is $c_0 = 0.0156$. Error bars are estimated for the MultiMSM by bootstrapping with 1000 resamplings (see Section IV C). In this work, all energies are given in units of the thermal energy $k_B T$ and all length scales in units of l_0 (see text).

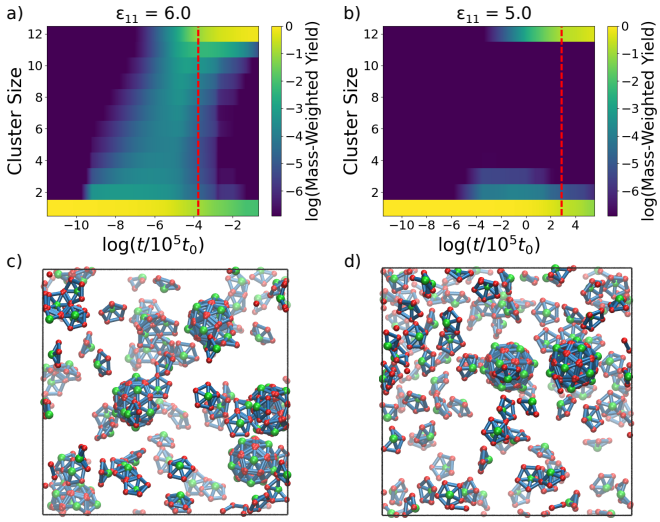

FIG. 4. Comparison of MultiMSM intermediate size distributions as a function of time for dodecahedron assembly with **(a)** $\epsilon_{11} = 6.0$ and **(b)** $\epsilon_{11} = 5.0$. Times and yields are plotted on a log-log scale, over the same time intervals for the corresponding plot in Fig. 3. **(c)**, **(d)** Snapshots of representative configurations at times corresponding to the red dashed lines in (a), (b).

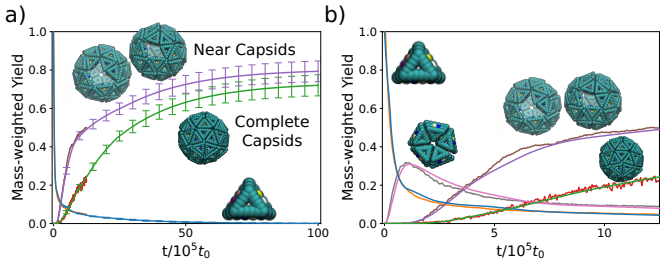

FIG. 5. MultiMSM predictions of $T = 3$ capsid assembly dynamics with $(\epsilon_{12}, \epsilon_{33}) = (11, 8)$ and $c_0 = 1.7 \times 10^{-4}$, compared against results from brute-force dynamics on accessible timescales. **(a)** MultiMSM predictions of mass-fractions (smooth curves) are shown for monomers (blue), capsids (green), and 'near-capsids' (purple, $56 - 60$ subunits), with example snapshots labeling each curve. These predictions are compared against mass fractions estimated from 40 brute-force dynamics trajectories performed up to $12.5 \times 10^5 t_0$ (noisy curves). Error bars are estimated for the MultiMSM by bootstrapping with 1000 resamplings (see Section IV C) **(b)** A more detailed comparison of MultiMSM and brute-force results, also showing mass fractions of pentamers (pink).

compared against results from brute-force dynamics (noisy curves) on accessible timescales (up to $12.5 \times 10^5 t_0$). This comparison is shown in more detail in Fig. 5b, where we also include pentamers. We observe extremely close agreement, within the statistical error of the brute-force simulations. In particular, the MultiMSM captures the rapid conversion of monomers into pentamers at early times, slow monomer depletion at late times, and the tendency of assembly pathways to become trapped in near-capsid intermediates.

These behaviors arise because these simulations are performed at binding affinity values $\epsilon_{12} = 11$ and $\epsilon_{33} = 8$ for which intra-pentamer interactions are strong and intra-dimer interactions are relatively weak [166]. This imbalance leads to hierarchical assembly pathways in which many subunits first form pentamers, which in turn assemble into nearly complete capsids. However, due to a combination of steric effects, monomer binding, and pentamer depletion, many assembly pathways become trapped in near-capsid structures with $56 - 59$ subunits. While some of these structures are

converted into complete capsids by monomer additions, they persist at $\approx 10\%$ mass fraction even at very long times.

C. Error and Efficiency

	D12 $\epsilon_{11} = 6.0$	D12 $\epsilon_{11} = 5.5$	D12 $\epsilon_{11} = 5.0$	T3 $(\epsilon_{12}, \epsilon_{33}) = (11, 8)$
Dynamics	0.85(6)	0.75(5)	0.11(7)	0.21(8)
MultiMSM	0.826(9)	0.74(2)	0.13(3)	0.25(2)

TABLE I. Comparing the estimated capsid yield and error at T_f for each of the four examples with brute-force dynamics and the MultiMSM. Yields and errors are estimated from brute-force dynamics by sample averages (see Fig. 3 and 5), and from the MultiMSM by bootstrapping with 1000 resamplings. Final simulation times for the dodecahedron examples are $T_\mathrm{f} = 0.5 \times 10^5 t_0$ for $\epsilon_{11} = 6$, $T_\mathrm{f} = 10 \times 10^5 t_0$ for $\epsilon_{11} = 5.5$, $T_\mathrm{f} = 2.5 \times 10^5 t_0$ for $\epsilon_{11} = 5$; for the $T = 3$ capsid $T_\mathrm{f} = 12.5 \times 10^5 t_0$.

Error. For a standard MSM, the uncertainty in the equilibrium distribution and other quantities can be directly propagated from uncertainty estimates in transition matrix entries [207–210]. However, in the MultiMSM such propagation is complicated by the unknown switching times between the component MSMs and the smoothing procedure. Therefore, we quantify errors using bootstrapping [211, 212] (see Appendix C for further details).

Table I shows a comparison of the estimated means and standard errors of the capsid yield for each of the examples from the MultiMSM by bootstrapping with 1000 resamples. These results are compared against sample averages from the brute-force dynamics simulations. In each example the comparison is shown for the final simulation time point T_f. We see that the MultiMSM yields are within the statistical error of the estimates from brute-force dynamics, and that the statistical error for the MultiMSM is consistently smaller than that from the dynamics.

We have used the same bootstrapping approach to compute the statistical error of the MultiMSM yield predictions as a function of time for each of the assembly examples (Fig. 3 and 5a). Where available, the sample-averaged yields are typically within the error bars of the MultiMSM prediction. An exception is dodecahedron assembly with $\epsilon_{11} = 6$ (Fig. 3a), for which the computed error bars are quite small at some times and sample averages lay outside them. This is likely because this example used the fewest sample trajectories to build the MultiMSM and we did not perform adaptive sampling, so the sampling with replacement step performed for the bootstrap results in very similar models. For the examples with longer assembly timescales (Fig. 3c and 5a), the errors generally grow in time due to accumulated error from each monomer fraction interval of the MultiMSM. While the error for dodecahedron assembly with $\epsilon_{11} = 5$ is particularly large at long time scales, our analysis shows that this is because we had very limited sampling at the small monomer fraction values that occur at long timescales. Importantly though, this error could be significantly reduced with further sampling. The error estimates provide a guide to refining the discretization of the monomer fractions and performing additional sampling (see Appendix A 4 and Appendix C 1).

As for a traditional MSM, the accuracy of the MultiMSM at long times is limited by the sampling of the most relevant slow transitions. For example, in the triangles system which involves strong intra-pentamer interactions, we sampled pentamer-to-monomer transitions only 25 times and dissociation from complete capsids only ~ 500 times. As noted above, adaptive sampling techniques focus on such transitions to improve statistics. However, estimates of transition matrix elements that involve such rare events can be improved much more efficiently by incorporating non-Boltzmann techniques [213–216].

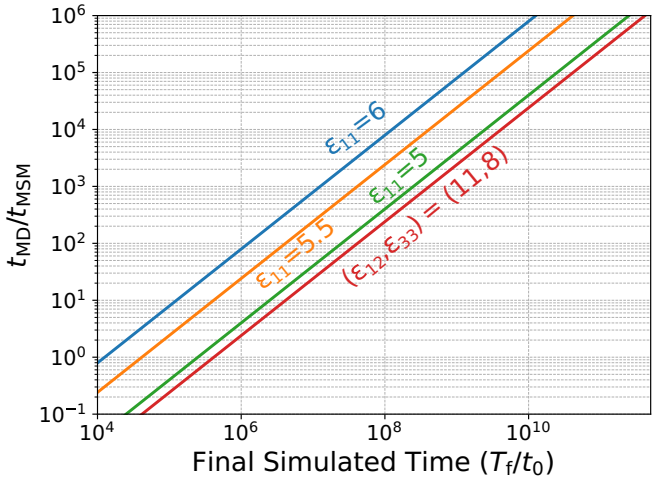

FIG. 6. Estimate of the computational speedup provided by the MultiMSM. The plot shows the efficiency, defined as the ratio of the total computational time required to simulate a given observation time T_f by constructing a MultiMSM or by brute-force dynamics, $t_\mathrm{MSM}/t_\mathrm{MD}$. Estimates are shown for the four example systems considered in this work: dodecahedron assembly with $\epsilon_{11} = 6$ (blue), $\epsilon_{11} = 5.5$ (orange), $\epsilon_{11} = 5$ (green), and $T = 3$ capsid assembly (red) with $\epsilon_{12} = 11$ and $\epsilon_{33} = 8$. As noted in the text, the ratio is independent of the standard error of the estimate of the capsid yield.

Efficiency. Next, we compare the efficiency of constructing a MultiMSM compared to performing an ensemble of straight-forward dynamics simulations. We find that the standard error of the estimated capsid yield for both the brute-force dynamics and the MultiMSM (without adaptive or enhanced sampling) scale with the number of statistically independent samples, consistent with the central limit theorem. Thus, defining the total computational time for brute-force dynamics and the time required to build a converged MultiMSM respectively as t_MD and t_MSM, the standard error scales as $\sim t_\mathrm{MD}^{-1/2}$ and $\sim t_\mathrm{MSM}^{-1/2}$. The time required to perform the bootstrapping with the MultiMSM is negligible compared to the simulation time, so we exclude this time from the calculation. The efficiency of the MultiMSM can be improved by adaptive sampling and enhanced sampling as noted in the pre-

vious paragraph.

As a measure of speedup of the MultiMSM, we define the *efficiency* as the ratio of sampling time $t_{\text{MD}}/t_{\text{MSM}}$ required to simulate to a given observation time T_{f}. Since the simulation time for both approaches has the same scaling with error (not accounting for adaptive/enhanced sampling) the efficiency is independent of error tolerance. However, the efficiency scales linearly with observation time, since the brute-force dynamics simulation time is $\sim T_{\text{f}}$, while once the MultiMSM is constructed, it can be simulated to any timescale with a cost that is negligible compared to sampling time. The proportionality coefficient and relevant timescales are system dependent. Thus, as with any enhanced sampling method, the MultiMSM speedup depends on the separation of timescales. For self-assembly, the speedup will increase exponentially with the height of nucleation barriers; i.e., with decreasing binding affinity or subunit concentration.

Fig. 6 shows the efficiency values as a function of observation time for each example system studied in this work. We see that the efficiency or speedup of the MultiMSM can be quite significant, as large as 10^6 for considered timescales for the dodecahedron system with $\epsilon_{11} = 5$. The speedup is significantly lower for the $T = 3$ system, for which we used a very strong binding affinity with correspondingly small nucleation barriers and rapid assembly. For reference, Table II in Appendix A 5 shows the total computational times required for the estimates of capsid yields by the MultiMSM and brute-force dynamics (Figs. 3 and 5). Here, the speedup of the MultiMSM is modest because parameters were chosen to enable direct comparison with brute-force dynamics as much as possible. Thus, we were restricted to small T_{f} and parameters that lead to relatively small barrier heights and consequently a small proportionality coefficient. As evident from Fig. 6, a longer observation time and lower binding affinities or subunit concentrations would lead to significantly higher efficiency of the MultiMSM.

V. APPLICATIONS

In this section we describe additional dynamics calculations and analysis that can be performed from the MultiMSM without any additional sampling of short trajectories. First, we describe how the assembly dynamics can be computed for total subunit concentrations below that of the original calculation. Then we show how free energy profiles and entropy production rates can be computed over the course of the assembly reaction. Finally, in the supplement section S2 D, we show how previous applications of transition path theory to compute assembly pathways and committor probabilities can be extended to the MultiMSM.

A. Concentration Sweeps

A powerful consequence of how the MultiMSM is constructed is that assembly dynamics at total subunit concentrations c_0 below that used in the simulations to build the MultiMSM can be computed from the same set of transition matrices, requiring no additional sampling. While the calculation is not rigorous, the procedure is an excellent approximation provided that concentrations of intermediates remain small, which is typically the case for productive assembly reactions.

Consider a discretization of the monomer fraction, $D = [0, d_1, ..., d_N, 1]$, such that a monomer fraction of 1 corresponds to a maximum total subunit concentration of c_0^{max}. For our primary analysis, we initialize the system using the MSM defined on the interval $[d_N, 1]$ with a starting distribution of all monomers, $p_i^0 = \delta_{i0}$, and the solution to Eq. (5) gives the system dynamics with concentration c_0^{max}.

We can then approximate the dynamics for lower subunit concentrations $c_0 = d_k c_0^{\text{max}}$ without any additional sampling, by initializing the MSM defined on one of the inner intervals, $[d_{k-1}, d_k]$ for $k < N$, with the same initial distribution of all monomers. We define a new monomer fraction discretization $D_k = [0, d_1/d_k, d_2/d_k, ..., d_{k-1}/d_k, 1]$, and assign corresponding transition matrices to the same intervals. For example, if $\boldsymbol{P}_1$ originally was the transition matrix on the interval $[0, d_1]$, in the reduced system it will be the transition matrix on the interval $[0, d_1/d_k]$. A limitation is that we are constrained to the finite set of concentrations corresponding to the bins of our discretization, $d_k c_0$, and depending on the quality and amount of sampling initially performed in the bins closer to zero, the error may increase as smaller concentrations are probed. However, if needed, additional sampling can be performed to refine the discretization based on error analysis as described in Section IV C and Appendix C 1.

Fig. 7a shows the results of a concentration sweep performed using this method for the dodecahedron system with $\epsilon_{11} = 5.5$. The capsid yield curves are computed for the initial concentration c_0^{max} and smaller values corresponding to the monomer fraction discretization bin cutoffs, and then interpolated between these discrete values. The results show that the final capsid yields decrease while assembly timescales increase as the total subunit concentration is reduced, consistent with previous theory and experiment [2–5, 7, 49, 50, 57–61, 67, 78, 127, 156, 169, 171, 172, 195, 202–205, 217–232]. We can also infer the critical assembly concentration, as the MultiMSM initialized at concentration $0.25 c_0^{\text{max}}$ predicts a capsid yield of zero.

This procedure provides an excellent approximation to the dynamics when there is a large separation of timescales between nucleation and growth timescales, so that monomers are depleted slowly in comparison to the timescale for transitions among larger intermediates. Such a separation of timescales is consistent with the usual criteria for MSMs to provide effective computational speed up. The dodecahedron assembly examples with lower binding energies ($\epsilon_{11} = 5.0, 5.5$) are good examples of this scenario. Cluster nucleation is a rare event, and entire transition pathways from monomer to capsid are frequently sampled in each of the monomer fraction discretization bins. Consequently the approximation is highly accurate in these cases, as shown in Fig. 7b, which compares the MultiMSM predictions made by extrapolating to lower total concentrations against brute-force dynamics trajectories. We performed simulations until $10 \times 10^5 t_0$, when

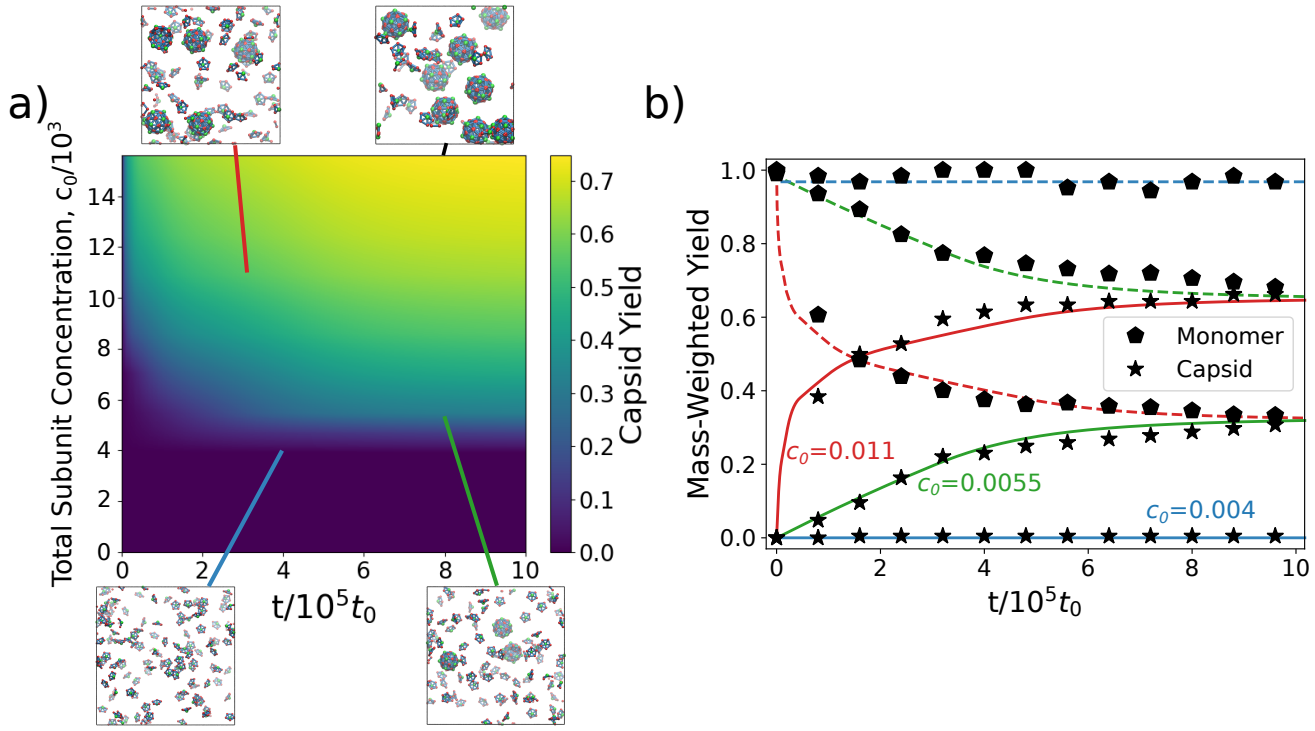

FIG. 7. Using the MultiMSM to estimate how assembly dynamics depends on total subunit concentration, c_0, without requiring additional simulations. **(a)** The MultiMSM prediction for the capsid mass fraction is shown as a function of time and c_0 for the dodecahedron assembly with $\epsilon_{11} = 5.5$. Representative snapshots of the simulation box are shown for four choices of time and total subunit concentration. **(b)** Comparison of the MultiMSM predictions of monomer (dashed lines) and capsid (solid lines) mass fractions against brute-force dynamics results (symbols) for total subunit concentrations $0.7c_0^{\max}$ (red), $0.35c_0^{\max}$ (green), and $0.25c_0^{\max}$ (blue). For the brute-force dynamics simulations, the total number of subunits was fixed at $N_{\text{tot}} = 125$, the box size was increased to $L_0 = 22.525l_0$, $28.380l_0$, and $31.750l_0$ respectively, and 20 independent trajectories were performed for each system. Results from the brute-force dynamics simulations at lower concentrations were not used in construction of the MultiMSM.

the capsid yield reaches $\approx 99\%$ of its equilibrium value according to the MultiMSM for $c_0 = 0.7c_0^{\max} = 0.011$ and $c_0 = 0.35c_0^{\max} = 0.0055$. For $c_0 = 0.25c_0^{\max} = 0.004$, the MultiMSM predicts no assembly, and we observe only a single capsid out of a possible 200 in the brute-force dynamics. These comparisons provide further evidence that the transition matrix estimation is accurate in the lower monomer fraction bins of the original MultiMSM. This reinforces the accuracy of the long-time predictions at the original concentration $c_0^{\max}$ in Fig. 3b, which depends on the accuracy of the transition matrix on each interval.

For the example in Fig. 3c, approaching equilibrium with brute-force simulations is computationally intractable, and thus we cannot directly test the MultiMSM predictions. Importantly though, performing this concentration sweep out to such timescales using the MultiMSM takes only a few minutes on a CPU, demonstrating many orders of magnitude of computational speed-up.

The approximate concentration results are less accurate in cases with poor separation of timescales. The triangle system at the simulated binding affinity provides an example of this scenario, in which monomers deplete quickly, but larger intermediates form over a longer timescale. Fig. 8 shows the result of performing the concentration sweep for the $T = 3$ capsid assembly, comparing to brute-force dynamics simulations over the computationally accessible timescale of $T_{\text{f}} = 12.5 \times 10^5 t_0$. The blue curves used the initial total subunit concentration, $c_0 = c_0^{\max}$, as a reference, while the green and purple curves used concentrations $c_0 = 0.63c_0^{\max} = 1.0 \times 10^{-4}$ and $c_0 = 0.1c_0^{\max} = 1.7 \times 10^{-5}$, respectively. We see that for the intermediate concentration (green), $c_0 = 1.0 \times 10^{-4}$, the MultiMSM predictions match well with the brute-force estimates except for a slight overestimate of capsid yields at early times. Monomer depletion is still accurately characterized by the MultiMSM in this case. However, for lower concentrations such as $c_0 = 1.7 \times 10^{-5}$ (purple), the method breaks down; the MultiMSM predicts too rapid monomer depletion and a nonzero capsid yield, even though the largest intermediate size is 10 in brute-force simulations on this timescale. This breakdown can be attributed to the relatively high populations of intermediates; once the monomer fraction becomes small enough ($< 10\%$) in the initial system, most of the sampled transitions are between monomers and larger intermediates. When we then use these transition matrices to predict the dynamics of a system with all monomers in the approximate concentration sweep, the prediction overestimates the rate of monomers forming larger intermediates. This breakdown of the concentration sweep approximation can be identified by

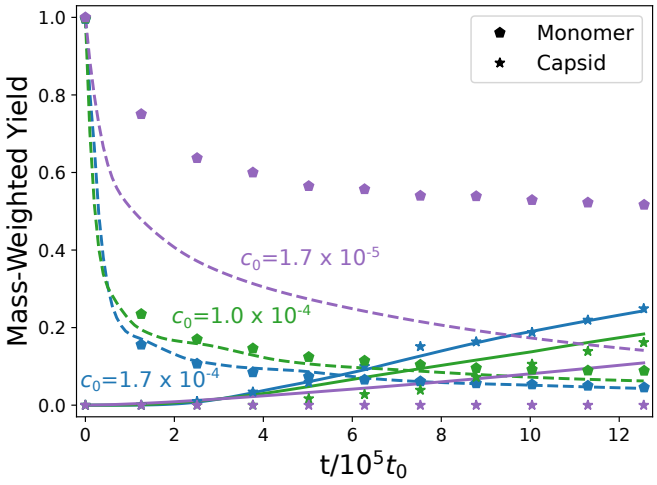

FIG. 8. The $T = 3$ capsid assembly example shows that the concentration sweep approximation can become inaccurate for insufficient separation of timescales. Comparison of the MultiMSM predictions of monomer (dashed lines) and capsid (solid lines) mass fractions against brute-force dynamics results (symbols) for total subunit concentrations $c_0^{\max}$ (blue), $0.63c_0^{\max}$ (green), and $0.1c_0^{\max}$ (purple). For the brute-force dynamics simulations, the total number of subunits was fixed at $N_{\text{tot}} = 600$, the box size was set to $L_0 = 153l_0, 179.4l_0$, and $331.4l_0$ respectively, and 20 independent trajectories were performed for each system. Results from the brute-force dynamics simulations at lower concentrations were not used in construction of the MultiMSM.

comparing the yield of intermediates to the total subunit concentration. When the triangle monomer fraction reaches 0.63, intermediates with sizes between 5 and 55 subunits account for $\sim 10\%$ of the yield, and the concentration sweep remains accurate when initialized at this total subunit concentration. When the triangle monomer fraction reaches 0.1, the same set of intermediates account for over 55% of the yield and the concentration sweep approximation breaks down. In our examples, an intermediate yield of about 20% seems to be an upper bound for getting reasonable results from the MultiMSM concentration sweep.

B. Entropy Production Rates

The entropy production rate provides a means to quantitatively measure how far from equilibrium a process is, thus elucidating the irreversibility of a process (which produces entropy), the heat produced or work done by the system, or the efficiency of a process [233–236]. Much recent work has developed optimal control algorithms for non-equilibrium systems to minimize entropy production [155, 237–250]. However, entropy production is frequently difficult to measure in experiments or simulations of complex models, due to the large amount of data needed to reliably estimate probability distributions and currents, although approaches based on machine learning [251–257] and automatic-differentiation [258] can help. Fortunately, the MSMs enable computationally efficient computation of entropy production.

The time-dependent entropy production rate for a Markov chain at step n is given by the expression

$$\epsilon_{\text{p}}^n = \frac{1}{2}\sum_{i,j}\left(p_i^n P_{ij} - p_j^n P_{ji}\right)\log\frac{p_i^n P_{ij}}{p_j^n P_{ji}}, \tag{8}$$

where p_i^n is the probability of being in state i at time n, and P_{ij} is the transition probability from state i to state j. This extends naturally to the MultiMSM framework, with the transition matrix replaced by the time-dependent transition matrix across the monomer fraction bins

$$\epsilon_{\text{p}}^n = \frac{1}{2}\sum_{i,j}\left(p_i^n P_{ij}(n) - p_j^n P_{ji}(n)\right)\log\frac{p_i^n P_{ij}(n)}{p_j^n P_{ji}(n)}, \tag{9}$$

where $\boldsymbol{P}(n)$ is the transition matrix used to update the system at time step n. This quantity can be computed while solving the forward Kolmogorov equation as described in Section III E.

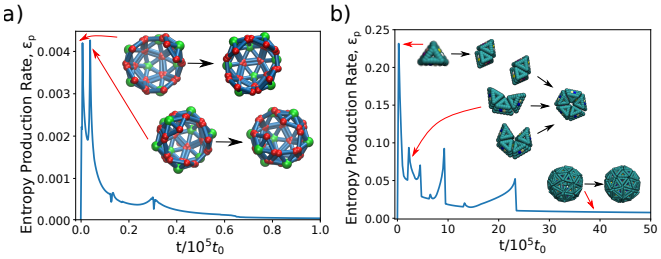

FIG. 9. Entropy production rates as a function of time as computed by the MultiMSM for **(a)** dodecahedron assembly with $\epsilon_{11} = 5.5$ and $c_0 = 0.0156$ and **(b)** T=3 capsid assembly with $(\epsilon_{12}, \epsilon_{33}) = (11, 8)$ and $c_0 = 1.7 \times 10^{-4}$. Selected peaks and other areas of interest are labeled with the transition or transitions that contribute most to the total entropy at that time.

Figs. 9a,b show the result of computing the entropy production rate as a function of time using the MultiMSM for dodecahedron assembly at the intermediate binding energy, $\epsilon_{11} = 5.5$, and the T=3 model. For the dodecahedron case, we observe an initial spike in the entropy production rate at early times, corresponding to a non-equilibrium flux of monomers into larger intermediates. The entropy production rate then decreases rapidly and remains small throughout the remainder of the simulation. Importantly, note the difference in timescales for the entropy production rate decreasing toward zero and the yields reaching a steady state in Fig. 4b. This suggests the assembly is near-equilibrium once the monomer concentration reaches roughly half its starting value. This is consistent with the accepted notion that productive assembly occurs when the system is near equilibrium [7, 203, 259].

We can also track the dominant contributions to the sum in Eq. (9) to better understand the assembly process. The two initial peaks are labeled with a representation of the transition contributing the most to the entropy at that time. The first peak corresponds to the transition between $(\mathcal{N}, \mathcal{B}) = (7, 12)$ to $(\mathcal{N}, \mathcal{B}) = (8, 15)$, accounting for approximately 7% of the

entropy production rate, while the second peak corresponds to the transition between $(\mathcal{N}, \mathcal{B}) = (9, 18)$ to $(\mathcal{N}, \mathcal{B}) = (10, 21)$, accounting for approximately 11% of the entropy production rate. These two transitions are dominant considering that the next largest contributions account for less than 2%. This observation suggests these are the dominant nearly irreversible transitions in the assembly process, and indeed we can see these transitions occur on the most likely assembly pathway identified in supplement Fig. S5b. The enhanced lack of reversibility for these transitions can be understood because both transitions add three bonds to the configuration, compared to just one or two for most of the other early transitions along the pathway. This result demonstrates that the entropy production can provide important insights into a reaction pathway by identifying the key transitions that stabilize intermediates or products.

The $T = 3$ model (Fig. 9b) exhibits a similar large spike in entropy production at early times due to rapid formation of dimers and small intermediates, followed by a decay over time. The initial peak corresponds to the monomer to dimer transition, accounting for approximately half the entropy production at this time. The next peak is dominated by transitions from a dimer, trimer, or tetramer to a pentamer, with each transition contributing approximately 2.5% for a total of 7.5%. These are the expected dominant contributions, as the strong intra-pentamer interactions make pentamerization a nearly irreversible process under these parameter values. There are no intermediate peaks that have a dominant contribution to the total entropy production, but rather many transitions with roughly the same contribution. An interesting observation for this example is that the entropy production rate does not tend to zero at large times, but rather decays to a small positive number, approximately 0.006. It does not decay further even if we increase the final time by up to two orders of magnitude. This entropy production is almost entirely ($> 99\%$) due to configurations with 59 subunits transitioning to the T=3 capsid, as well as configurations with 60 subunits but the wrong bond configuration transitioning to the T=3 capsid. At this point in the assembly, the free monomer concentration has already decayed to less than 1% of its initial value, which indicates that the system has become trapped in metastable states. This result demonstrates that entropy production provides a useful measure of the extent to which a system is kinetically trapped.

C. Free Energy

Another useful application of MSMs is calculating the equilibrium free energy of each microstate, which can then be projected onto reaction coordinate(s) for mechanistic insight. Notably, the MultiMSM allows computing the *equilibrium* free energy profile independent of free monomer concentration, from nonequilibrium simulations at arbitrary monomer concentration.

While a typical MSM enables computing the Helmholtz free energy from the equilibrium distribution of microstates $\vec{\pi}$ as $F_i = -k_\mathrm{B}T \log(\pi_i)$, in the MultiMSM framework the transition matrix depends on the free subunit concentration and thus the corresponding chemical potential. Therefore, the MultiMSM gives the grand free energy for the j-th component MSM as

$$\Omega_i^j = -k_\mathrm{B}T \log(\pi_i^j) = F_i - \mu_j(n_i - 1), \tag{10}$$

where Ω_i^j is the grand free energy for state i in the j-th MSM, $\mu_j = k_\mathrm{B}T \log(c_j/c_\mathrm{ss})$ is the chemical potential for the j-th MSM with c_j as the average monomer concentration within the bin and c_ss as the standard state concentration, and n_i is the number of subunits in microstate i.

Importantly, since the Helmholtz free energy F_i in Eq. (10) is independent of the chemical potential (and thus the free monomer concentration), it should be the same for each component MSM of the MultiMSM. Therefore, its statistics can be improved by averaging over each of the component MSMs.

To simplify the following presentation, we project the free energy onto a reaction coordinate where there is one state per intermediate size n, but the approach readily generalizes to multiple states per size. To compute F_i within a particular MSM, we compute the equilibrium distribution from the transition matrix and then the set of equilibrium constants for the formation of each n-mer,

$$K_n = [n] \,/\, [1]^n, \tag{11}$$

where $[n]$ denotes the equilibrium concentration of n-mers. Note that any absorbing states (states or groups of states for which there were an insufficient number of exit transitions to estimate an outward transition rate) should be eliminated from the transition matrix before computing equilibrium quantities. As noted above, outward transition rates could be computed

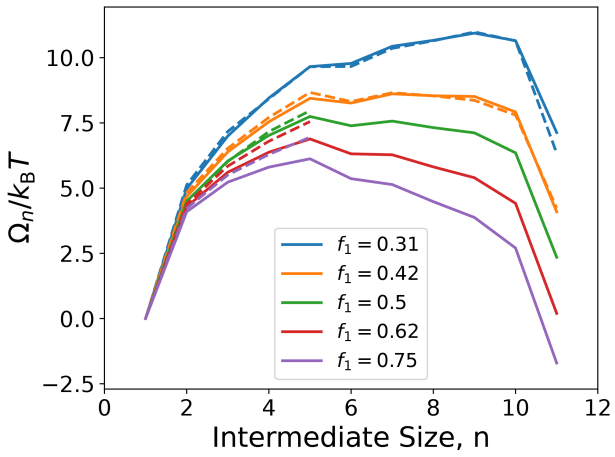

FIG. 10. The grand free energy for dodecahedron assembly at $\epsilon_{11} = 5.5$, computed analytically from the Helmholtz free energy using Eq. (10) and evaluated at the average monomer fraction within each discretization bin (solid lines). Dashed lines show the grand free energy computed directly from the MSM equilibrium distribution, for monomer fractions and intermediate sizes that are sufficiently close to equilibrium to enable comparison.

for such states by combining free energy calculations with the dynamical simulations used to estimate transition rates [213], but we have not implemented this approach for the present work. Then, the concentration of subunits in absorbing states should be subtracted from the total subunit concentration to ensure proper normalization of the grand free energy (see Appendices D 1 and D 2 for details).

The Helmholtz free energy can be computed as

$$F_n = -k_{\mathrm{B}}T \log(c_{\mathrm{ss}}^{n-1} K_n). \tag{12}$$

While the equilibrium constant corresponds to the true observable and is thus independent of standard state, the free energy values necessarily depend on c_{ss}. For the results presented here, we choose c_{ss} in simulation units such that there is one subunit per circumscribed volume occupied by the subunit: $c_{\mathrm{ss}} = 0.66/l_0^3$ for the pentagonal subunit and $c_{\mathrm{ss}} = 0.17/l_0^3$ for the triangular subunits. See supplement Section S2 C 1 and Fig. S2 for plots of the Helmholtz free energy for both systems, verifying the collapse with respect to monomer fraction. We also independently test the MSM free energy calculations against direct calculations from the Brownian dynamics simulations in supplement Fig. S4, showing excellent agreement.

The Helmholtz free energy enables computing the grand free energy at any free monomer concentration by adding the chemical potential term as in Eq. (10). For example, Fig. 10 shows the grand free energy for dodecahedra with $\epsilon_{11} = 5.5$ at several representative subunit concentrations. We evaluate the grand free energy at monomer fractions corresponding to the average monomer fraction within each discretization window (solid lines). This allows for a direct comparison of our computed grand free energy with the result of computing it directly from the component MSM equilibrium distribution (dashed lines). Note that the nucleation barrier increases, and the stability of intermediates decrease, as free monomer concentrations decrease due to the increased monomer chemical potential. Note that these curves should coincide only when the assembly is at or near equilibrium. Fig. 9a shows that the entropy production vanishes once the monomer fraction is below roughly 0.5, and we see excellent agreement between the two calculations at lower monomer fractions. For larger monomer fractions, we may expect structures smaller than the critical nucleus to be in quasi-equilibrium [203], so we make comparisons up to an intermediate size of 5. In this case, we find loose agreement between the curves that improves as the monomer fraction decreases.

We show free energy profiles for a range of monomer fractions for $T = 3$ capsids in supplement Section S2 C 1 and Fig. S3. However, since we have shown above that this system is kinetically trapped and thus not in equilibrium for our chosen parameters, we do not make comparisons with the equilibrium MSM for this case.

VI. CONCLUSIONS

We have described the MultiMSM, a general framework to construct MSMs for systems in which many clusters assemble simultaneously and concentrations (and potentially other parameters) change over the course of the reaction. Using two model systems, we show that MultiMSMs can accurately describe assembly dynamics over the long timescales required to approach equilibrium, even when constructed from trajectories that are orders of magnitude shorter. This capability enables particle-based simulations with complex models to be simulated at experimentally relevant concentrations. The degree of speed up enabled by the MultiMSM increases exponentially for systems with large nucleation barriers, and the method is well-suited for systems that assemble by diverse pathways. Moreover, the method is trivially parallelizable and thus highly scalable.

In addition to extending on previous work by allowing for multiple clusters and the depletion of free subunits, the MultiMSM approach allows for a number of further applications. Notably, a MultiMSM constructed at one total subunit concentration can be used to perform an approximate parameter sweep over a wide range of concentrations with no additional sampling. This capability corresponds to orders of magnitude additional speed up in comparison to brute-force simulations. The results are highly accurate for conditions leading to productive assembly, under which concentrations of intermediates remain relatively low, and qualitatively accurate for more aggressive assembly conditions that lead to a buildup of intermediates. The MultiMSM framework computes transition path theory quantities, such as the committor probability describing the extent of progress along a reaction coordinate and the relative flux along different assembly pathways. It can also be used to estimate the Helmholtz free energy of a system, as well as compute the grand free energy as a function of the monomer concentration. Further, the method allows for efficient calculation of entropy production rates, a quantity which has been difficult to compute from particle-based simulation trajectories. We find that the entropy production rate provides a useful quantification of how far an assembly reaction is from equilibrium and whether it is susceptible to kinetic traps. More interestingly, by analyzing which transitions contribute most to the entropy production rate, one can identify factors that stabilize critical nuclei or other key intermediates, and whether these transitions lead to productive assembly or engender kinetic traps. These insights can form the basis for rational design of synthetic assembly systems, or, in the case of biomedically relevant assembly systems such as viruses, help to identify targets for antiviral molecules that interfere with assembly [68, 260–264].

ACKNOWLEDGMENTS

This work was supported by the NSF through DMR 2309635 and the Brandeis Center for Bioinspired Soft Materials, an NSF MRSEC (DMR-2011846). We also acknowledge computational support from NSF XSEDE computing resources allocation TG-MCB090163 (NCSA Delta GPU) and the Brandeis HPCC which is partially supported by the NSF through DMR-MRSEC 2011846 and OAC-1920147.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Our Python library to post-process the results of a HOOMD simulation [161] and our Python library to construct and perform calculations on MultiMSMs [161, 162] are freely available on Github. Our simulation scripts, a subset of post-processed trajectory data, analysis scripts, and figure generating scripts are hosted on the Open Science Framework OSFHome (https://osf.io/ak49n/).

Appendix A: Simulation and MultiMSM Construction Details

1. Dodecahedron system

The subunits are rigid bodies composed of several kinds of pseudoatoms. There are attractor pseudoatoms ('A') at the vertices of a regular pentagon, which facilitate subunit assembly via an attractive Morse potential with an equilibrium length of $L_0 = 0.2$, a range parameter $\alpha = 2.5/L_0$, a cutoff distance of 2, and a well-depth (subunit-subunit binding strength) ϵ_{11} that can be varied. In this work, we use values $\epsilon_{11} \in \{5.0, 5.5, 6.0\}$. We also include a top pseudoatom ('T') and a bottom pseudoatom ('B'), located at positions $z = \pm 0.5$ with respect to the center of the regular pentagon in the xy plane. The 'T' pseudoatoms interact with other 'T' pseudoatoms via a repulsive Lennard-Jones potential with $\sigma_{\mathrm{TT}} = 2.1$, cutoff distance equal to σ_{TT}, and well-depth $\epsilon_{\mathrm{TT}} = \epsilon_{11}/4$. These values favor a subunit-subunit binding angle consistent with that of a dodecahedron. The 'B' pseudoatoms have a similar repulsive interaction with 'T' pseudoatoms, with $\sigma_{\mathrm{TB}} = 1.8$, a cutoff distance equal to σ_{TB}, and well-depth $\epsilon_{\mathrm{TB}} = \epsilon_{11}/4$. This interaction helps to prevent upside down assembly, i.e. ensuring that the top atom is in the direction of the outward normal vector. Finally, we add edge ('E') pseudoatoms at the midpoint between each adjacent vertex. These have no interactions and thus do not affect the simulation, but they are used to track assembly progress more easily compared to vertex pseudoatoms.

Results are reported in units for which the unit length l_0 corresponds to the edge length of the pentagonal subunit and energies are measured in units of $k_{\mathrm{B}}T$. Simulations are initialized with 125 pentagonal subunits, enough to form 10 dodecahedral capsids (12 subunits each), on an equally spaced lattice. Subunit positions and orientations are then equilibrated with a purely repulsive potential for 2×10^5 time steps before writing any output. The simulation box is a cube with periodic boundary conditions and side lengths $20l_0$, giving a total subunit concentration of $c_0 = 0.0156/l_0^3$. The time step is $0.001t_0$, and the base simulations to construct MSMs are run for 5×10^6 time steps unless otherwise specified. The simulations use the HOOMD-blue [163] version 3.9.0 Langevin integrator, with an inverse temperature $\beta = 1$. The configurations are recorded every $\Delta t = t_0$ units of simulation time.

2. $T = 3$ capsid system

The subunits are rigid triangles with each edge consisting of three stacked layers of six overlapping 'excluder' pseudoatoms at a specified bevel angle. These excluder pseudoatoms interact with all other pseudoatoms in the simulation through a Weeks-Chandler-Anderson (WCA) potential to enforce excluded volume. Embedded in the middle row of each edge of the triangle are two attractor pseudoatoms, which have attractive Lennard-Jones interactions with the pseudoatoms on complementary edges to facilitate edge-edge binding of the subunits. Each pseudoatom has the same diameter, σ, which we set as the unit distance l_0. To match the dimensions of the experimental subunit [166], we can set $\sigma = 18$nm in real units.

As described so far, the triangular subunits can be designed to form a broad variety of target structures by tuning the interaction strengths, side lengths, and bevel angles. For a $T = 3$ capsid target, we make two sides of the triangle equivalent. Sides 1 and 2 have an edge length of $3\sigma = 54$nm and have complementary pseudoatoms (Side 1 has pseudoatoms '4' and '5', which bind with pseudoatoms '7' and '6' on Side 2, respectively) that attract with a binding energy of ϵ_{12}. These two sides do not interact at all with Side 3, which has a slightly longer edge length of $3.35\sigma = 60.3$nm, with attractive pseudoatoms (pseudoatoms '2' and '3') that are self-complementary with a binding well-depth of ϵ_{33}. The bevel angle of each side is the same, approximately $11.64°$. This design can produce hierarchical assembly [166]. When $\epsilon_{33} \gg \epsilon_{12}$, dimers form rapidly via the Side 3 – Side 3 interaction, and then the dimers more slowly assemble into larger structures via the Side 1 – Side 2 interaction. Conversely, when $\epsilon_{33} \ll \epsilon_{12}$, pentamers form rapidly and then subsequently assemble into larger structures. In this work, we consider an example of pentamer-biased assembly, with $\epsilon_{12} = 11$ and $\epsilon_{33} = 8$.

The simulations contain $N = 600$ triangular subunits, enough to form 10 $T = 3$ capsids (60 subunits each). The simulation domain is an $L \times L \times L$ box with periodic boundary conditions, whose side lengths determine the total subunit concentration, $c_0 = N/(N_{\mathrm{A}}L^3)$, where N_{A} is Avogadro's number. We set $c_0 = 50$nM to be on the order of experimental conditions, for which the corresponding box side length is $L = 2.71$ microns. In simulation units, the box side lengths are $153l_0$, giving a total subunit concentration of $c_0 = 1.7 \times 10^{-4}/l_0^3$.

We initialize subunit positions on an equally spaced, truncated lattice. Subunit positions and orientation are then equilibrated with a purely repulsive potential for 8×10^4 time steps, after which attractive interactions are turned on. We use a time step of $0.0025t_0$, and the base simulations to construct MSMs are run for 5×10^8 time steps unless stated otherwise.

3. Bond Definitions and Discrete States

For the dodecahedron system, the pentagonal subunits have edge ('E') pseudoatoms that align when the adjacent vertex attractors bind with another subunit. We use a cutoff distance of $0.3l_0$ between these edge pseudoatoms to define a bond. For the $T = 3$ capsid assembly, there are a pair of complementary pseudoatoms on Sides 1 and 2 ('4' and '7' pseudoatoms), and Side 3 is self-complementary ('2' and '3' pseudoatoms). We use a cutoff distance of $1.3l_0$ for both pairs of complementary pseudoatoms in this case. These values were chosen by selecting the minimum distance that would correctly identify fully formed capsids across many different realizations of the assembly. Our results are insensitive to increasing these cutoffs up to the next-nearest neighbor bond distances, which are approximately $0.8l_0$ for the pentagonal subunit and $2.4l_0$ for the triangular subunits. Decreasing these cutoffs results in frequent oscillations in the number of bonds, even for stable configurations like the fully formed dodecahedron. Such behavior is undesirable when building a Markov state model, as it overestimates the rate of bond breakage in the model, so we ensure our choice of cutoff correctly identifies the target structure over long timescales at the strongest binding energies.

Using this definition of a bond, we construct our discrete state space as all observed combinations of $(\mathcal{N}, \mathcal{B})$, where $\mathcal{N}$ is the number of subunits in a cluster and $\mathcal{B}$ is the distribution of the number of bonds in a cluster. Defined in this way, the number of unique discrete states we observe is system dependent. There are 180 states for dodecahedral capsid assembly with binding energies $\epsilon_{11} = 5$ and $\epsilon_{11} = 5.5$. For $\epsilon_{11} = 6$, the interactions are strong enough to result in transient intermediates that are larger than 12 subunits, resulting in a state space size of 447. Despite the enhanced size, most of these states are extremely low probability and could likely be excluded without affecting the model predictions, though we have not tested this. For $T = 3$ capsid assembly, the number of states is much larger due to a larger capsid size and having two distinguishable bond types. For the pentamer-biased assembly set that we consider, the state space size is 1662.

4. Additional Enhanced Sampling

A powerful feature of MSMs is the ability to simulate dynamics on timescales that are much longer than those of the brute force dynamics simulations that are used to estimate the transition matrix elements. Adaptive sampling, in which sampling is focused on important, potentially under-sampled transitions, increase the accuracy of such predictions.

Ensuring the accuracy of longtime predictions is more challenging for the MultiMSM than a standard MSM because the transition matrices change over time in the MultiMSM. Straightforward brute-force sampling may not result in good statistics for the long time behaviors. For example, in the pentamer-biased simulations of the $T = 3$ capsid, the average monomer fraction at the final simulation time is about 5%. About $1/4$ of the base trajectories sampled lower values than this, but there are only $\sim$ 20000 transitions sampled for $0\% < f_1 < 3.5\%$ while there are $\sim$ 160000 transitions sampled for the smaller interval $3.5\% < f_1 < 6\%$. Thus, MultiMSM dynamics on timescales that lead to such low monomer fractions will have limited accuracy without additional sampling at low monomer fractions. We have developed techniques to more efficiently generate such data, particularly in typical challenging cases.

The first case, mentioned above, occurs when the monomer fraction at the final simulation time has not yet reached its equilibrium value, but is close to it. To generate additional sampling at lower monomer fractions, we identify which of the existing trajectories ended with the lowest monomer fraction and initialize a new simulation in the final frame of the existing one. Since transition rates will depend on the precise distribution of intermediates present, we try to perform this for as many different starting frames as possible, with multiple random seeds for each. We refer to this as 'continued' sampling.

A particularly challenging sub-case occurs when the interaction strengths are relatively weak. In this case, the nucleation of a cluster is a rare event, and reaching equilibrium may take orders of magnitude longer than available computational times. Furthermore, as the reaction proceeds and the monomer concentration is depleted, the nucleation barrier grows. In this case, we artificially push the system closer to equilibrium, in a way that does not bias the equilibrium distribution of intermediates. Since the equilibrium distribution in assembly systems with a large nucleation barrier can be well approximated by a coexistence of just full capsids and monomers, we can construct a starting frame from an existing frame by randomly selecting monomers and manually assembling them into a capsid, placing them in the simulation box in such a way that there is no overlap. We refer to this as 'fraction' sampling, since we can target a particular monomer fraction range to sample. Alternatively, one can also remove the monomers forming full capsids from the simulation box. As long as the simulation box size is reduced to account for the volume of the capsid, this should not bias the dynamics of the remaining monomers, and could significantly speed up the simulations depending on the system size. We refer to this as 'reduced' sampling.

Another issue can arise in cases where the interaction strengths are relatively strong. In this case, monomers will very quickly deplete, resulting in very few sample transitions being used to construct MSMs in the discretization bins corresponding to larger monomer fractions. In this case, we perform many short simulations, anywhere from 5 to 10 percent of the original simulation time, to gather more samples for the larger monomer fraction values. We refer to this as 'short' sampling. Fortunately, we have found that in cases of fast depletion, we can choose the monomer fraction discretization such that the entire fast depletion region is contained within a single discretization bin, which reduces the need for this additional sampling.

5. MultiMSM Parameter Overview

Here we summarize the parameters used to construct our MultiMSMs for each example in the main text. Relevant parameters include the amount of sampling data and what type of enhanced sampling simulation it came from, the monomer fraction discretization, the minimum number of observations of a transition, and the smoothing parameter used to solve the forward equation for the yields.

First, we list the monomer fraction discretization for each example. These were

$$\begin{aligned}
\text{Dodecahedron, } \epsilon_{11} = 6.0\text{: } & [0, 0.05, 0.12, 0.13, 0.23, \\
& 0.27, 0.52, 0.59, 0.76, 1.0] \\
\text{Dodecahedron, } \epsilon_{11} = 5.5\text{: } & [0, 0.15, 0.25, 0.35, 0.45, \\
& 0.55, 0.71, 0.81, 1.0] \\
\text{Dodecahedron, } \epsilon_{11} = 5.0\text{: } & [0, 0.08, 0.16, 0.31, 0.45, \\
& 0.61, 0.75, 0.89, 0.98, 1.0] \\
\text{T=3 Capsid: } & [0, 0.035, 0.06, 0.1, 0.2, \\
& 0.57, 0.63, 1.0]
\end{aligned}$$

In general, these were the result of applying the optimization scheme described next in Appendix B, using test data to generate sample estimates to optimize over computationally accessible timescales. In the cases where no simulation data is available to compare against, such as taking the dodecahedron estimates out to equilibrium for the smaller binding energies, the bins closer to one were optimized as before, and subsequent bins selected manually by performing the error-based refinement procedure detailed in Appendix C 1. We find the resulting solutions in these cases to be insensitive to small perturbations in the bin locations (± 0.01).

The two scalar parameters to the MultiMSM are the smoothing parameter to the forward solver, χ, and the minimum number of observations to keep for a particular transition, which we call the prune tolerance. The latter removes transitions that are rare and unlikely to be important to the dynamics. The default values for these parameters are $\chi = 0.25$ and a prune tolerance of 1, which means no pruning. The only exception for the smoothing parameter is the dodecahedron assembly with $\epsilon_{11} = 6$, which used $\chi = 0.5$. The only exception for the prune tolerance is the dodecahedron assembly with $\epsilon_{11} = 5.5$, which used a prune tolerance of 2.

Finally, we report the amount of sampling (including enhanced sampling) performed to build each MultiMSM. The base simulations for each system are described above, with dodecahedron and $T = 3$ simulations run for 5×10^6 and 5×10^8 time steps respectively. For 'fraction' and 'reduced' sampling, a variable number of these simulations were performed per 0.1 monomer fraction bin, so we report the number of trajectories per bin. Note that bin i in this case refers to simulations initialized with monomer fraction $0.1(i + 1)$. These simulations were the same length as the base simulations for each system.

For the dodecahedron assembly with $\epsilon_{11} = 6$, we performed 120 base simulations and 35 'continued' simulations that were twice the length of the base simulations, for a total simulation time of 9.5×10^8 time steps. For $\epsilon_{11} = 5.5$, we performed 80 base simulations. The number of 'fraction' simulations was 200 in bin 3, 300 in bin 4, 100 in bin 5, and 40 in bin 6. The number of 'reduced' simulations was 200 in bin 2, 200 in bin 3, and 50 in bins 4, 5, and 6. The total simulation time is 6.35×10^9 time steps. For $\epsilon_{11} = 5$, we performed 100 base simulations. In addition to this, we performed 200 'reduced' simulations in each bin 1 through 9. The total simulation time is 9.5×10^9 time steps. For the $T = 3$ assembly, we performed 35 base simulations, as well as 10 'continued' simulations that were half the length of the base simulations. The total simulation time is 10^{10} time steps. The total computational time required for the MSM estimates of capsid yields shown in Figs. 3 and 5 are given in Table II under t_{MSM}. For comparison, we also show the corresponding simulation time that would be required to make these estimates with the same level of error using only brute-force dynamics simulations as t_{MD}.

Example	t_{MSM}	t_{MD}
$\epsilon_{11} = 6.0$	110	520
$\epsilon_{11} = 5.5$	107	2580
$\epsilon_{11} = 5.0$	420	42050
$(\epsilon_{12}, \epsilon_{33}) = (11, 8)$	1120	24200

TABLE II. Total simulation time required for the MultiMSM (t_{MSM}) and brute-force dynamics (t_{MD}) for the comparison of capsid yields predicted by both approaches in Figs. 3 and 5.

Appendix B: Choosing a Good Monomer Fraction Discretization

In this section we describe a protocol that aims to select an optimal monomer fraction discretization, and a metric to gauge the quality of the discretization.

Our cluster analysis software, in addition to returning all transitions a particular cluster makes along its lifetime, can also be used to track the yield of any of the discrete states, s, from any set of simulations. For each system, we perform a set of simulations that are initialized with all subunits as monomers with thermalized positions and orientations. We average the mass-weighted yields of each discrete state over each of these simulations to get an estimate of the true probability of observing that structure, $\{\hat{p}_s^n\}_n$. In particular, we can do this for 'important' intermediate states, such as monomers, dimers, pentamers, the full capsid, other intermediates that are relatively high probability or are involved in important transitions, or any combination thereof. We can then compare these curves to the corresponding entries of the time-dependent probability distribution we get from solving Eq. (5) from the MultiMSM with a discretization D, $\{p_s^n(D)\}_n$. Our metric for the quality of the discretization is then a normed difference of these two quantities,

$$C(D) = \|\hat{p}_s - p_s(D)\|^p = \left(\sum_n |\hat{p}_s^n - p_s^n(D)|^p \right)^{1/p}, \quad \text{(B1)}$$

where we typically use the 2-norm, but leave this as a tunable parameter for the optimization.

This metric can then be used as the cost function to minimize for some optimization procedure in which the discretization points d_i are varied. We experimented with both Monte Carlo and gradient descent optimization schemes, but both tended to be both slow and prone to getting stuck in local minima. Instead, we apply a sequential optimization procedure that works as follows.

Consider the discretization $D_N = (0, d_1, d_2, \cdots, d_N, 1)$ as an initial guess. Since all our examples begin with a monomer fraction of 1 and deplete monomers over time, the bins closer to 1 have a larger effect on the accuracy of the MultiMSM yield curves. This is because any error made in the bin closest to 1 will propagate to all future bins as the monomers deplete. Therefore, our sequential optimizer works by fixing d_1 through d_{N-1} and choosing the optimal value for d_N, that minimizes the cost function. We then vary d_{N-1} while keeping all of the other bins fixed, then d_{N-2}, and so on, continuing until we reach d_1. It is possible that modifying the other bin locations has shifted the optimal value for d_N, so we perform another sequential optimization cycle. We repeat the procedure until we reach a cutoff number of cycles, or until a cycle terminates without changing any of the d_i.

Every time a new discretization is tested the MultiMSM must be reconstructed since many of the transitions may now lie in a different bin. While we have implemented a caching system to reduce the time it takes to reconstruct the MultiMSM with a new discretization, the construction time and time to solve the forward equation is non-negligible. Therefore, we want to minimize the number of discretizations we test while performing this optimization. When optimizing d_i, we construct the interval between neighboring discretization points, $[d_{i-1}, d_{i+1}]$, and place M equally spaced points within this interval. We keep M relatively small, typically $M = 4$, and check if using those value for d_i reduces the cost function. If not, we keep the same d_i and move to the next point. This keeps the optimization time to a reasonable level; typically we observe convergence in about $5-10$ minutes, using $N = 6-8$, $M = 4$, and 5 cycles on a single 3.5 GHz CPU.

Appendix C: Bootstrapping Procedure

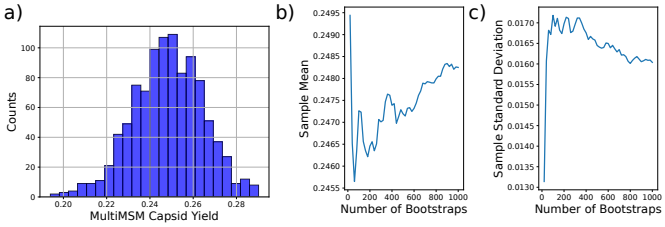

FIG. 11. Error estimates by bootstrapping for the MultiMSM capsid yield estimate at the final simulation time for the $T = 3$ system. **(a)** Histogram of MultiMSM capsid yield estimates using 1000 bootstraps. **(b)** and **(c)** show the sample mean and standard deviation, respectively, as a function of the number of bootstraps.

To estimate errors for the MultiMSM yield predictions, particularly in cases where comparison against brute-force simulation is intractable, we perform a bootstrapping procedure. Bootstrapping is a resampling technique in which multiple random samples, known as bootstrap samples, are drawn with replacement from an observed dataset. This method allows for the estimation of the sampling distribution of a statistic, to estimate uncertainties when the underlying distribution is unknown or performing error propagation is not straightforward [211, 212].

The typical bootstrapping procedure begins with a dataset with N measurements. From this set, we construct a resampling that consists of N samples from the original dataset, drawn uniformly at random with replacement. The quantity of interest is evaluated for the resampled dataset and becomes a bootstrap sample. After collecting M bootstrap samples, a histogram can be constructed to show the full distribution, and the bootstrap sample mean and standard deviation can be computed. We perform this procedure for the MultiMSM yields, using the training trajectories as our dataset, with a small modification; we keep the number of each type of trajectory (see Sections A 4 and A 5) fixed during the resampling, instead of just the total number of trajectories. This is particularly important for the examples with slow monomer depletion, as a resampling that does not include enough trajectories in the lower monomer fraction regimes will give nonsensical results. We generate $M = 1000$ bootstrap samples for each yield measurement for which we wish to estimate errors, which are reported in the main text. Fig. 11 shows an example histogram of bootstrapping samples for the $T = 3$ capsid yield at the final simulation time, as well as the sample mean and standard deviation as a function of number of bootstrap samples. We can see the histogram is approximately normally distributed, and that the sampling estimates do not vary by much over the last 200 samples, indicating we have performed enough bootstraps.

1. Error-Based Refinement

Bootstrapping can be performed on the MultiMSM at various time points during the assembly to estimate the model error as a function of time. These errors can then be used as a guide for refining the MultiMSM model; by identifying which monomer fraction bin the MultiMSM is using at the time the error becomes large, this gives information about where the monomer fraction discretization should be refined or more sampling should be performed.

For example, the solid lines and error bars in Fig. 12 were computed by bootstrapping for the dodecahedron example with binding energy $\epsilon_{11} = 5.5$, but with a poor choice for the monomer fraction discretization, $D_{\text{test}} = [0, 0.1, 0.35, 0.45, 0.55, 0.71, 0.81, 1.0]$. This model differs from the converged model from the main text (dashed lines), $D_{\text{main}} = [0, 0.15, 0.25, 0.35, 0.45, 0.55, 0.71, 0.81, 1.0]$ only in the first two monomer fraction bins. We see a large spike in the error when the monomer fraction goes below 0.35 for D_{test}, indicating an issue with the model in this bin, despite

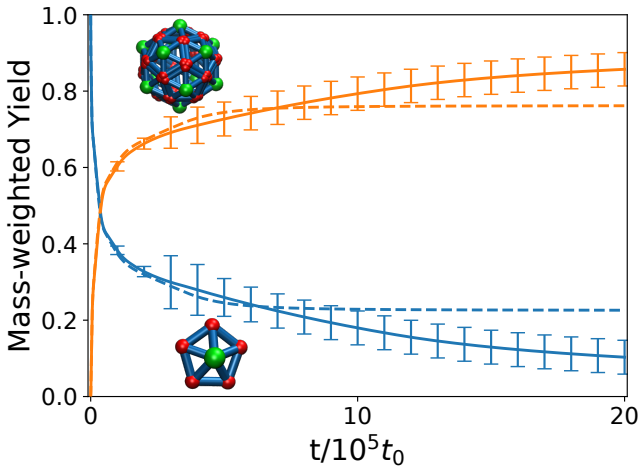

FIG. 12. Example error refinement using bootstrapping error estimates. The dashed lines show the MultiMSM yield predictions for monomers (blue) and dodecahedral capsids (orange) using the converged model from the main text for $\epsilon_{11} = 5.5$. Solid lines and error bars are generated using 1000 bootstrap samples with a poor choice of monomer fraction discretization, $D_{\text{test}} = [0, 0.1, 0.35, 0.45, 0.55, 0.71, 0.81, 1.0]$. Refinement of the discretization then gives the converged MultiMSM that produced the dashed line results.

the model means being quite close at this time, which is all the optimization based refinement in Appendix B addresses. Over longer timescales, the test model performs poorly; overestimating capsid formation and monomer depletion as well as the timescale to reach equilibrium.

Since we have a converged and validated model with D_{main}, we can directly assess why the test model fails. We have noted in the main text that the critical aggregation concentration (CAC) for this example corresponds to a monomer fraction of approximately 0.25. Slightly above this value, the system assembles slowly, while below it the system does not assemble at all. Both of these possibilities are captured in the discretization window $[0.1, 0.35]$ of the test model. This means that transition data above the CAC is being used to approximate the dynamics below the CAC, resulting in poor model performance. The resampling process for the bootstrap will construct models that are biased toward either side of the CAC, resulting in estimates with a large spread of possible values and therefore large error estimates. By breaking up this large window with the intermediate monomer fraction of 0.25, we get the model reported in the main text, which presents far smaller error estimates and matches well with sample averages from brute-force dynamics simulations.

While this example may seem a bit contrived, this error analysis is precisely how we arrived at our final monomer discretization for this model. Before identifying the CAC or verifying the MultiMSM predictions with simulations, the test model above was our best working model for this example, obtained partly through trial-and-error and partly by running the optimization in Appendix B using trajectories up until a final time of $2.5 \times 10^5 t_0$. Since there were no estimates of the late time yields at that point, we could not evaluate this test model without additional error analysis. By performing this error analysis, we were able to identify a place to refine the discretization and construct a model that accurately predicts the equilibrium yields without knowing them a priori. While we have not done this, an interesting possibility is to add these bootstrap error bars as a metric to the refinement objective function in Appendix B, which would allow for model refinement in the absence of sample trajectories, the main weakness of the refinement approach we used there.

Appendix D: Free Energy Calculation Details

1. Removal of Absorbing States

The free energy computations detailed in the main text all involve computing the equilibrium distribution for each component transition matrix of the MultiMSM. The equilibrium distribution is given by the left eigenvector of the transition matrix corresponding to an eigenvalue of 1. If the transition matrix is ergodic, this distribution is guaranteed to be unique and can be computed using standard numerical linear algebra techniques. This is often not the case in practice; target structures are typically designed to be stable which means they may not be sampled reversibly, particularly in cases with strong interactions between subunits. States (or sets of states) that can be entered but cannot be left are reffered to as *absorbing* states, which must be identified and removed from the transition matrix before computing the equilibrium distribution.

We use a depth-first search algorithm to determine the strongly connected component of the transition matrix; the maximal sub-graph for which every state has a non-zero probability of reaching any other state in a finite number of steps. Each state that is not a member of the strongly connected component is classified as an absorbing state. For each absorbing state, we remove its corresponding row and column in the transition matrix to form a reduced transition matrix. The rows of this reduced transition matrix need to be renormalized to sum to one, and the resulting ergodic matrix is used to compute the equilibrium distribution over the remaining states.

2. Equilibrium Constants

Another subtlety arises when computing the equilibrium constants as a function of intermediate size. The equilibrium constants are a function of the equilibrium *concentrations* of each species, not the equilibrium probability as we have computed thus far. For an n-mer, we can write the equilibrium concentration (in simulation units) as $[n] = \gamma c_0 \pi_n$, where c_0 is the total subunit concentration, π_n is the total equilibrium probability for observing an n-mer, and γ is a scaling factor to account for the lost concentration due to removal of absorbing states. This scaling factor is necessary because removing the absorbing states from the equilibrium calculation effectively

reduces the total subunit concentration, while the probability distribution remains normalized to sum to one. Its value depends on the fraction of absorbing states, which generally increases in time, and thus depends on the monomer fraction discretization bin. For a discretization bin between monomer fractions d_i and d_{i+1}, and corresponding times t_i and t_{i+1} for which $f_1 = d_i$ and $f_1 = d_{i+1}$, respectively, we compute γ_i as the average yield of non-absorbing states over this time interval,

$$\gamma_i = 1 - \frac{1}{t_i - t_{i+1}} \sum_{j=t_{i+1}}^{t_i} p_A^j, \tag{D1}$$

where p_A^j is the total yield of absorbing states at time j, computed from the original transition matrix.

[1] D. L. D. Caspar and A. Klug, Physical principles in the construction of regular viruses., Cold Spring Harbor symposia on quantitative biology **27**, 1 (1962).

[2] M. G. Mateu, Assembly, stability and dynamics of virus capsids, Archives of Biochemistry and Biophysics **531**, 65 (2013).

[3] R. F. Bruinsma and W. S. Klug, Physics of viral shells, Annual Review of Condensed Matter Physics **6**, 245 (2015).

[4] J. D. Perlmutter and M. F. Hagan, Mechanisms of virus assembly, Annual Review of Physical Chemistry, Vol 62 **66**, 217 (2015).

[5] M. F. Hagan and R. Zandi, Recent advances in coarse-grained modeling of virus assembly, **18**, 36 (2016).

[6] R. Twarock, R. J. Bingham, E. C. Dykeman, and P. G. Stockley, A modelling paradigm for RNA virus assembly, **31**, 74 (2018).

[7] R. Zandi, B. Dragnea, A. Travesset, and R. Podgornik, On virus growth and form, Physics Reports **847**, 1 (2020).

[8] M. F. Hagan and G. M. Grason, Equilibrium mechanisms of self-limiting assembly, Reviews of Modern Physics **93**, 025008 (2021).

[9] C. A. Kerfeld, S. Heinhorst, and G. C. Cannon, Bacterial microcompartments, Microbiology+ **64**, 391 (2010).

[10] S. Tanaka, C. A. Kerfeld, M. R. Sawaya, F. Cai, S. Heinhorst, G. C. Cannon, and T. O. Yeates, Atomic-level models of the bacterial carboxysome shell, Science **319**, 1083 (2008).

[11] B. D. Rae, B. M. Long, M. R. Badger, and G. D. Price, Functions, compositions, and evolution of the two types of carboxysomes: Polyhedral microcompartments that facilitate CO2 fixation in cyanobacteria and some proteobacteria, Microbiology and Molecular Biology Reviews **77**, 357 (2013).

[12] T. A. Bobik, B. P. Lehman, and T. O. Yeates, Bacterial microcompartments: Widespread prokaryotic organelles for isolation and optimization of metabolic pathways, Molecular Microbiology **98**, 193 (2015).

[13] C. Chowdhury, S. Sinha, S. Chun, T. O. Yeates, and T. A. Bobik, Diverse bacterial microcompartment organelles, Microbiology and Molecular Biology Reviews **78**, 438 (2014).

[14] C. A. Kerfeld and M. R. Melnicki, Assembly, function and evolution of cyanobacterial carboxysomes, Current Opinion in Plant Biology **31**, 66 (2016).

[15] J. K. Polka, S. G. Hays, and P. A. Silver, Building spatial synthetic biology with compartments, scaffolds, and communities, Cold Spring Harbor perspectives in biology **8**, 10.1101/cshperspect.a024018 (2016).

[16] C. A. Kerfeld, C. Aussignargues, J. Zarzycki, F. Cai, and M. Sutter, Bacterial microcompartments, Nature Reviews Microbiology **16**, 277 (2018).

[17] M. Slininger Lee and D. Tullman-Ercek, Practical considerations for the encapsulation of multi-enzyme cargos within the bacterial microcompartment for metabolic engineering, Current Opinion in Systems Biology 10.1016/j.coisb.2017.05.017 (2017).

[18] S. Banerjee, M. L. Gardel, and U. S. Schwarz, The actin cytoskeleton as an active adaptive material, Annual Review of Condensed Matter Physics **11**, 421 (2020).

[19] M. T. Cabeen and C. Jacobs-Wagner, The bacterial cytoskeleton, Annual Review of Genetics **44**, 365 (2010).

[20] M. Pilhofer and G. J. Jensen, The bacterial cytoskeleton: More than twisted filaments, Current Opinion in Cell Biology **25**, 125 (2013).

[21] S. Whitelam, Control of pathways and yields of protein crystallization through the interplay of nonspecific and specific attractions, Physical Review Letters **105**, 088102 (2010).

[22] T. Garg, G. Rath, and A. K. Goyal, Colloidal Drug Delivery Systems: Current Status and Future Directions, Critical Reviews in Therapeutic Drug Carrier Systems **32**, 89 (2015).

[23] M. Beija, R. Salvayre, N. Lauth-de Viguerie, and J.-D. Marty, Colloidal systems for drug delivery: From design to therapy, Trends in Biotechnology **30**, 485 (2012).

[24] S. Ebbens, Active colloids: Progress and challenges towards realising autonomous applications, Current Opinion in Colloid & Interface Science **21**, 14 (2016).

[25] S. A. Mallory, C. Valeriani, and A. Cacciuto, An active approach to colloidal self-assembly, Annual Review of Physical Chemistry **69**, 59 (2018).

[26] J. A. Fan, Y. He, K. Bao, C. Wu, J. Bao, N. B. Schade, V. N. Manoharan, G. Shvets, P. Nordlander, D. R. Liu, and F. Capasso, DNA-enabled self-assembly of plasmonic nanoclusters, Nano Letters **11**, 4859 (2011).

[27] J.-H. Huh, K. Kim, E. Im, J. Lee, Y. Cho, and S. Lee, Exploiting colloidal metamaterials for achieving unnatural optical refractions, Advanced Materials **32**, 2001806 (2020).

[28] Y. Ke, L. L. Ong, W. M. Shih, and P. Yin, Three-dimensional structures self-assembled from DNA bricks, Science **338**, 1177 (2012).

[29] D. J. Kraft, J. Groenewold, and W. K. Kegel, Colloidal molecules with well-controlled bond angles, Soft Matter **5**, 3823 (2009).

[30] S. Sacanna, W. T. M. Irvine, P. M. Chaikin, and D. J. Pine, Lock and key colloids, Nature **464**, 575 (2010).

[31] S. Sacanna, M. Korpics, K. Rodriguez, L. Colón-Meléndez, S.-H. Kim, D. J. Pine, and G.-R. Yi, Shaping colloids for self-assembly, Nature Communications **4**, 1 (2013).

[32] G.-R. Yi, D. J. Pine, and S. Sacanna, Recent progress on patchy colloids and their self-assembly, Journal of Physics: Condensed Matter **25**, 193101 (2013).

[33] Y. Wang, Y. Wang, X. Zheng, G.-R. Yi, S. Sacanna, D. J. Pine, and M. Weck, Three-dimensional lock and key colloids, Journal of the American Chemical Society **136**, 6866 (2014).

[34] M. He, J. P. Gales, É. Ducrot, Z. Gong, G.-R. Yi, S. Sacanna, and D. J. Pine, Colloidal diamond, Nature **585**, 524 (2020).

[35] M. He, J. P. Gales, X. Shen, M. J. Kim, and D. J. Pine, Colloidal Particles with Triangular Patches, Langmuir : the ACS journal of surfaces and colloids (2021).

[36] Q. Chen, J. K. Whitmer, S. Jiang, S. C. Bae, E. Luijten, and S. Granick, Supracolloidal reaction kinetics of Janus spheres, Science **331**, 199 (2011).

[37] Q. Chen, S. C. Bae, and S. Granick, Directed self-assembly of a colloidal kagome lattice, Nature **469**, 381 (2011).

[38] D. Zerrouki, J. Baudry, D. Pine, P. Chaikin, and J. Bibette, Chiral colloidal clusters, Nature **455**, 380 (2008).

[39] J. Yan, K. Chaudhary, S. C. Bae, J. A. Lewis, and S. Granick, Colloidal ribbons and rings from Janus magnetic rods, Nature Communications **4**, 1 (2013).

[40] J. R. Wolters, G. Avvisati, F. Hagemans, T. Vissers, D. J. Kraft, M. Dijkstra, and W. K. Kegel, Self-assembly of "Mickey Mouse" shaped colloids into tube-like structures: Experiments and simulations, Soft Matter **11**, 1067 (2015).

[41] G. Tikhomirov, P. Petersen, and L. Qian, Triangular DNA Origami Tilings, Journal of the American Chemical Society **140**, 17361 (2018).

[42] J. S. Oh, S. Lee, S. C. Glotzer, G.-R. Yi, and D. J. Pine, Colloidal fibers and rings by cooperative assembly, Nature Communications **10**, 1 (2019).

[43] A. Ben-Ari, L. Ben-Ari, and G. Bisker, Nonequilibrium self-assembly of multiple stored targets in a dimer-based system, The Journal of Chemical Physics **155**, 234113 (2021).

[44] J. S. Kahn, B. Minevich, A. Michelson, H. Emamy, K. Kisslinger, S. Xiang, S. K. Kumar, and O. Gang, Encoding hierarchical 3D architecture through inverse design of programmable bonds, ChemRxiv : the preprint server for chemistry **https://doi.org/10.26434/chemrxiv-2022-xwbst** (2022).

[45] I. Palaia and A. Šarić, Controlling cluster size in 2D phase-separating binary mixtures with specific interactions, The Journal of Chemical Physics **156**, 194902 (2022).

[46] A. E. Hafner, N. G. Gyori, C. A. Bench, L. K. Davis, and A. Šarić, Modeling Fibrillogenesis of Collagen-Mimetic Molecules, Biophysical Journal **119**, 1791 (2020).

[47] L. Y. Rivera-Rivera, T. C. Moore, and S. C. Glotzer, Inverse design of triblock Janus spheres for self-assembly of complex structures in the crystallization slot *via* digital alchemy, Soft Matter **19**, 2726 (2023).

[48] D. C. Rapaport, Molecular dynamics study of T = 3 capsid assembly, Journal of Biological Physics **44**, 147 (2018).

[49] H. D. Nguyen, V. S. Reddy, and C. L. Brooks, Deciphering the kinetic mechanism of spontaneous self-assembly of icosahedral capsids, Nano Letters **7**, 338 (2007).

[50] H. D. Nguyen and C. L. I. Brooks, Generalized structural polymorphism in self-assembled viral particles, Nano Letters **8**, 4574 (2008).

[51] O. M. Elrad and M. F. Hagan, Encapsulation of a polymer by an icosahedral virus, Physical Biology **7**, 045003 (2010).

[52] A. J. Williamson, A. W. Wilber, J. P. K. Doye, and A. A. Louis, Templated self-assembly of patchy particles, Soft Matter **7**, 3423 (2011).

[53] J. D. Perlmutter, C. Qiao, and M. F. Hagan, Viral genome structures are optimal for capsid assembly, eLife **2**, e00632 (2013).

[54] R. Zhang and P. Linse, Icosahedral capsid formation by capsomers and short polyions, The Journal of Chemical Physics **138**, 154901 (2013).

[55] R. Zhang and P. Linse, Topological effects on capsomer-polyion co-assembly, The Journal of Chemical Physics **140**, 244903 (2014).

[56] J. E. Baschek, H. C. R. Klein, and U. S. Schwarz, Stochastic dynamics of virus capsid formation: Direct versus hierarchical self-assembly, Bmc Biophysics **5**, 10.1186/2046-1682-5-22 (2012).

[57] M. Castelnovo, T. Verdier, and L. Foret, Comparing open and closed molecular self-assembly, Epl-europhys Lett **105**, 10.1209/0295-5075/105/28006 (2014).

[58] M. Castelnovo, D. Muriaux, and C. Faivre-Moskalenko, Entropic control of particle sizes during viral self-assembly, New Journal of Physics **15**, 10.1088/1367-2630/15/3/035028 (2013).

[59] M. A. Boettcher, H. C. R. Klein, and U. S. Schwarz, Role of dynamic capsomere supply for viral capsid self-assembly, Physical Biology **12** (2015).

[60] C. I. Mendoza and D. Reguera, Shape selection and misassembly in viral capsid formation by elastic frustration, eLife **9**, e52525 (2020).

[61] R. Schwartz, P. W. Shor, P. E. Prevelige, and B. Berger, Local Rules Simulation of the Kinetics of Virus Capsid Self-Assembly, Biophysical Journal **75**, 2626 (1998).

[62] A. E. Hafner, J. Krausser, and A. Šarić, Minimal coarse-grained models for molecular self-organisation in biology, Current Opinion in Structural Biology **58**, 43 (2019).

[63] Y. Qian, D. Evans, B. Mishra, Y. Fu, Z. H. Liu, S. Guo, and M. E. Johnson, Temporal control by cofactors prevents kinetic trapping in retroviral Gag lattice assembly, Biophysical Journal **122**, 3173 (2023).

[64] S.-K. Guo, A. J. Sodt, and M. E. Johnson, Large self-assembled clathrin lattices spontaneously disassemble without sufficient adaptor proteins, PLOS Computational Biology **18**, e1009969 (2022).

[65] M. E. Johnson and G. Hummer, Free-Propagator Reweighting Integrator for Single-Particle Dynamics in Reaction-Diffusion Models of Heterogeneous Protein-Protein Interaction Systems, Physical Review X **4**, 031037 (2014).

[66] G. M. Rotskoff and P. L. Geissler, Robust nonequilibrium pathways to microcompartment assembly, Proceedings of the National Academy of Sciences **115**, 6341 (2018).

[67] S. Panahandeh, S. Li, L. Marichal, R. Leite Rubim, G. Tresset, and R. Zandi, How a Virus Circumvents Energy Barriers to Form Symmetric Shells, ACS Nano **14**, 3170 (2020).

[68] K. Kra, S. Li, L. Gargowitsch, J. Degrouard, J. Pérez, R. Zandi, S. Bressanelli, and G. Tresset, Energetics and Kinetic Assembly Pathways of Hepatitis B Virus Capsids in the Presence of Antivirals, ACS Nano **17**, 12723 (2023).

[69] D. L. Lynch, A. Pavlova, Z. Fan, and J. C. Gumbart, Understanding Virus Structure and Dynamics through Molecular Simulations, Journal of Chemical Theory and Computation **19**, 3025 (2023).

[70] S. Panahandeh, S. Li, and R. Zandi, The equilibrium structure of self-assembled protein nano-cages, Nanoscale **10**, 22802 (2018).

[71] Zhang, A. S. Keys, T. Chen, and S. C. Glotzer, Self-Assembly of Patchy Particles into Diamond Structures through Molecular Mimicry, Langmuir **21**, 11547 (2005).

[72] T. Chen, Z. Zhang, and S. C. Glotzer, A precise packing sequence for self-assembled convex structures, Proceedings of the National Academy of Sciences **104**, 717 (2007).

[73] F. Mohajerani, B. Tyukodi, C. J. Schlicksup, J. A. Hadden-Perilla, A. Zlotnick, and M. F. Hagan, Multiscale Modeling of Hepatitis B Virus Capsid Assembly and Its Dimorphism, ACS Nano **16**, 13845 (2022).

[74] B. Tyukodi, F. Mohajerani, D. M. Hall, G. M. Grason, and M. F. Hagan, Thermodynamic Size Control in Curvature-Frustrated Tubules: Self-Limitation with Open Boundaries, ACS Nano **16**, 9077 (2022).

[75] H. Fang, B. Tyukodi, W. B. Rogers, and M. F. Hagan, Polymorphic self-assembly of helical tubules is kinetically controlled, Soft Matter **18**, 6716 (2022).

[76] M. J. Del Razo, M. Dibak, C. Schütte, and F. Noé, Multiscale molecular kinetics by coupling Markov state models and reaction-diffusion dynamics, The Journal of Chemical Physics **155**, 10.1063/5.0060314 (2021).

[77] Y. Yang, R. B. Meyer, and M. F. Hagan, Self-Limited Self-Assembly of Chiral Filaments, Physical Review Letters **104**, 10.1103/physrevlett.104.258102 (2010).

[78] S. Whitelam, C. Rogers, A. Pasqua, C. Paavola, J. Trent, and P. L. Geissler, The Impact of Conformational Fluctuations on Self-Assembly: Cooperative Aggregation of Archaeal Chaperonin Proteins, Nano Letters **9**, 292 (2009).

[79] A. Aggeli, I. A. Nyrkova, M. Bell, R. Harding, L. Carrick, T. C. B. McLeish, A. N. Semenov, and N. Boden, Hierarchical self-assembly of chiral rod-like molecules as a model for peptide β-sheet tapes, ribbons, fibrils, and fibers, Proceedings of the National Academy of Sciences **98**, 11857 (2001).

[80] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl, Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations, Proceedings of the National Academy of Sciences **106**, 19011 (2009).

[81] G. R. Bowman, K. A. Beauchamp, G. Boxer, and V. S. Pande, Progress and challenges in the automated construction of Markov state models for full protein systems, The Journal of Chemical Physics **131**, 124101 (2009).

[82] V. S. Pande, K. Beauchamp, and G. R. Bowman, Everything you wanted to know about Markov State Models but were afraid to ask, Methods (San Diego, Calif.) **52**, 99 (2010).

[83] M. Sarich, F. Noé, and C. Schütte, On the approximation quality of markov state models, Multiscale Modeling & Simulation **8**, 1154 (2010).

[84] G. R. Bowman, V. A. Voelz, and V. S. Pande, Taming the complexity of protein folding, Current Opinion in Structural Biology **21**, 4 (2011).

[85] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé', Markov models of molecular kinetics: Generation and validation, The Journal of Chemical Physics **134**, 174105 (2011).

[86] V. A. Voelz, M. Jäger, L. Zhu, S. Yao, O. Bakajin, S. Weiss, L. J. Lapidus, and V. S. Pande, Markov state models of millisecond folder ACBP reveals new views of the folding reaction, Biophysical Journal **100**, 515a (2011).

[87] V. A. Voelz, M. Jäger, S. Yao, Y. Chen, L. Zhu, S. A. Waldauer, G. R. Bowman, M. Friedrichs, O. Bakajin, L. J. Lapidus, S. Weiss, and V. S. Pande, Slow unfolded-state structuring in acyl-CoA binding protein folding revealed by simulation and experiment, Journal of the American Chemical Society **134**, 12565 (2012).

[88] F. Noé and F. Nüske, A variational approach to modeling slow processes in stochastic dynamical systems, Multiscale Modeling & Simulation **11**, 635 (2013).

[89] D. De Sancho, J. Mittal, and R. B. Best, Folding kinetics and unfolded state dynamics of the GB1 hairpin from molecular simulation, Journal of Chemical Theory and Computation **9**, 1743 (2013).

[90] G. R. Bowman, V. S. Pande, and F. Noé, *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation* (Springer Netherlands, 2014).

[91] J. D. Chodera and F. Noé, Markov state models of biomolecular conformational dynamics, Current Opinion in Structural Biology **25**, 135 (2014).

[92] R. D. Malmstrom, C. T. Lee, A. T. Van Wart, and R. E. Amaro, Application of molecular-dynamics based markov state models to functional proteins, Journal of Chemical Theory and Computation **10**, 2648 (2014).

[93] C. R. Schwantes, R. T. McGibbon, and V. S. Pande, Perspective: Markov models for long-timescale biomolecular dynamics, The Journal of Chemical Physics **141**, 090901 (2014).

[94] G. Hummer and A. Szabo, Optimal dimensionality reduction of multistate kinetic and markov-state models, The Journal of Physical Chemistry B **119**, 9029 (2015).

[95] B. E. Husic and V. S. Pande, Markov state models: From an art to a science, Journal of the American Chemical Society **140**, 2386 (2018).

[96] X. Zeng, L. Zhu, X. Zheng, M. Cecchini, and X. Huang, Harnessing complexity in molecular self-assembly using computer simulations, Physical Chemistry Chemical Physics **20**, 6767 (2018).

[97] H. Wu and F. Noé, Variational approach for learning markov processes from time series data, Multiscale Modeling & Simulation **30**, 23 (2020).

[98] J. Weng, M. Yang, W. Wang, X. Xu, and Z. Tian, Revealing Thermodynamics and Kinetics of Lipid Self-Assembly by Markov State Model Analysis, Journal of the American Chemical Society **142**, 21344 (2020).

[99] E. Suárez, R. P. Wiewiora, C. Wehmeyer, F. Noé', J. D. Chodera, and D. M. Zuckerman, What markov state models can and cannot do: Correlation versus path-based observables in protein-folding models, Journal of Chemical Theory and Computation **17**, 3119 (2021).

[100] G. Torrie and J. Valleau, Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling, Journal of Computational Physics **23**, 187 (1977).

[101] W. E, W. Ren, and E. Vanden-Eijnden, String method for the study of rare events, Physical Review B **66**, 10.1103/physrevb.66.052301 (2002).

[102] S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman, THE weighted histogram analysis method for free-energy calculations on biomolecules. I. The method, Journal of Computational Chemistry **13**, 1011 (1992).

[103] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, TRANSITIONPATHSAMPLING: Throwing Ropes Over Rough Mountain Passes, in the Dark, Annual Review of Physical Chemistry **53**, 291 (2002).

[104] A. C. Pan, D. Sezer, and B. Roux, Finding Transition Pathways Using the String Method with Swarms of Trajectories, The Journal of Physical Chemistry B **112**, 3432 (2008).

[105] V. Ovchinnikov, M. Karplus, and E. Vanden-Eijnden, Free energy of conformational transition paths in biomolecules: The string method and its application to myosin VI, The Journal of Chemical Physics **134**, 10.1063/1.3544209 (2011).

[106] S. Fischer, K. W. Olsen, K. Nam, and M. Karplus, Unsuspected pathway of the allosteric transition in hemoglobin, Proceedings of the National Academy of Sciences **108**, 5608 (2011).

[107] R. Elber, A Milestoning Study of the Kinetics of an Allosteric Transition: Atomically Detailed Simulations of Deoxy Scapharca Hemoglobin, Biophysical Journal **92**, L85 (2007).

[108] F. Pietrucci, F. Marinelli, P. Carloni, and A. Laio, Substrate Binding Mechanism of HIV-1 Protease from Explicit-Solvent Atomistic Simulations, Journal of the American Chemical Society **131**, 11811 (2009).

[109] M. Lei, M. I. Zavodszky, L. A. Kuhn, and M. F. Thorpe, Sampling protein conformations and pathways, Journal of Computational Chemistry **25**, 1133 (2004).

[110] D. Moroni, P. G. Bolhuis, and T. S. Van Erp, Rate constants for diffusive processes by partial path sampling, The Journal of Chemical Physics **120**, 4055 (2004).

[111] A. Dickson and A. R. Dinner, Enhanced Sampling of Nonequilibrium Steady States, Annual Review of Physical Chemistry **61**, 441 (2010).

[112] R. J. Allen, P. B. Warren, and P. R. Ten Wolde, Sampling Rare Switching Events in Biochemical Networks, Physical Review Letters **94**, 10.1103/physrevlett.94.018104 (2005).

[113] J. Pfaendtner, D. Branduardi, M. Parrinello, T. D. Pollard, and G. A. Voth, Nucleotide-dependent conformational states of actin, Proceedings of the National Academy of Sciences **106**, 12723 (2009).

[114] A. Barducci, M. Bonomi, and M. Parrinello, Linking Well-Tempered Metadynamics Simulations with Experiments, Biophysical Journal **98**, L44 (2010).

[115] B. W. Zhang, D. Jasnow, and D. M. Zuckerman, Efficient and verified simulation of a path ensemble for conformational change in a united-residue model of calmodulin, Proceedings of the National Academy of Sciences **104**, 18043 (2007).

[116] G. Huber and S. Kim, Weighted-ensemble Brownian dynamics simulations for protein association reactions, Biophysical Journal **70**, 97 (1996).

[117] A. L. Ferguson, A. Z. Panagiotopoulos, I. G. Kevrekidis, and P. G. Debenedetti, Nonlinear dimensionality reduction in molecular simulation: The diffusion map approach, Chemical Physics Letters **509**, 1 (2011).

[118] J. J. Juárez and M. A. Bevan, Feedback controlled colloidal self-assembly, Advanced Functional Materials **22**, 3833 (2012).

[119] X. Tang, B. Rupp, Y. Yang, T. D. Edwards, M. A. Grover, and M. A. Bevan, Optimal feedback controlled assembly of perfect crystals, ACS Nano **10**, 6791 (2016).

[120] X. Tang, J. Zhang, M. A. Bevan, and M. A. Grover, A comparison of open-loop and closed-loop strategies in colloidal self-assembly, Journal of Process Control **60**, 141 (2017).

[121] M. A. Grover, D. J. Griffin, and X. Tang, Control of self-assembly with dynamic programming, IFAC-PapersOnLine **52**, 1 (2019).

[122] A. Trubiano and M. F. Hagan, Optimization of non-equilibrium self-assembly protocols using Markov state models, The Journal of Chemical Physics **157**, 244901 (2022).

[123] F. Jamalyaria, R. Rohlfs, and R. Schwartz, Queue-based method for efficient simulation of biological self-assembly systems, Journal of Computational Physics **204**, 100 (2005).

[124] T. Keef, C. Micheletti, and R. Twarock, Master equation approach to the assembly of viral capsids, Journal of Theoretical Biology **242**, 713 (2006).

[125] M. Hemberg, SN. Yaliraki, and M. Barahona, Stochastic kinetics of viral capsid assembly based on detailed protein structures, Biophysical Journal **90**, 3029 (2006).

[126] E. C. Dykeman, P. G. Stockley, and R. Twarock, Building a viral capsid in the presence of genomic RNA, Physical review. E **87**, 022717 (2013).

[127] B. Sweeney, T. Zhang, and R. Schwartz, Exploring the parameter space of complex self-assembly through virus capsid models, Biophysical Journal **94**, 772 (2008).

[128] T. Q. Zhang and R. Schwartz, Simulation study of the contribution of oligomer/oligomer binding to capsid assembly kinetics, Biophysical Journal **90**, 57 (2006).

[129] N. Misra, D. Lees, T. Q. Zhang, and R. Schwartz, Pathway complexity of model virus capsid assembly systems, Comput. Math. Method Med. **9**, 277 (2008).

[130] M. S. Kumar and R. Schwartz, A parameter estimation technique for stochastic self-assembly systems and its application to human papillomavirus self-assembly, Physical Biology **7**, 045005 (2010).

[131] L. Xie, G. R. Smith, X. Feng, and R. Schwartz, Surveying capsid assembly pathways through simulation-based data fitting, Biophysical Journal **103**, 1545 (2012).

[132] G. R. Smith, L. Xie, B. Lee, and R. Schwartz, Applying molecular crowding models to simulations of virus capsid assembly in vitro, Biophysical Journal **106**, 310 (2014).

[133] J. Rouwhorst, C. Ness, S. Stoyanov, A. Zaccone, and P. Schall, Nonequilibrium continuous phase transition in colloidal gelation with short-range attraction, Nature Communications **11**, 3558 (2020).

[134] J. Rouwhorst, P. Schall, C. Ness, T. Blijdenstein, and A. Zaccone, Nonequilibrium master kinetic equation modeling of colloidal gelation, Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **102**, 022602 (2020).

[135] N. W. Kelley, V. Vishal, G. A. Krafft, and V. S. Pande, Simulating oligomerization at experimental concentrations and long timescales: A Markov state model approach, The Journal of Chemical Physics **129**, 214707 (2008).

[136] M. R. Perkett and M. F. Hagan, Using Markov state models to study self-assembly, The Journal of Chemical Physics **140**, 10.1063/1.4878494 (2014).

[137] M. Dibak, M. J. del Razo, D. De Sancho, C. Schütte, and F. Noé, MSM/RD: Coupling Markov state models of molecular kinetics with reaction-diffusion simulations, The Journal of Chemical Physics **148**, 214107 (2018).

[138] S. Olsson, H. Wu, F. Paul, C. Clementi, and F. Noé', Combining experimental and simulation data of molecular processes via augmented Markov models, Proceedings of the National Academy of Sciences **114**, 8265 (2017).

[139] U. Sengupta, M. Carballo-Pacheco, and B. Strodel, Automated Markov state models for molecular dynamics simulations of aggregation and self-assembly, The Journal of Chemical Physics **150**, 10.1063/1.5083915 (2019).

[140] A. Trubiano and M. Holmes-Cerfon, Thermodynamic stability versus kinetic accessibility: Pareto fronts for programmable self-assembly, Soft Matter **17**, 6797 (2021).

[141] M. Z. Miskin, G. Khaira, J. J. de Pablo, and H. M. Jaeger, Turning statistical physics models into materials design engines, Proceedings of the National Academy of Sciences **113**, 34 (2016).

[142] F. Pietrucci, Strategies for the exploration of free energy landscapes: Unity in diversity and challenges ahead, Reviews in Physics **2**, 32 (2017).

[143] H. Sidky and J. K. Whitmer, Learning free energy landscapes using artificial neural networks, The Journal of Chemical Physics **148**, 104111 (2018).

[144] Z. M. Sherman, M. P. Howard, B. A. Lindquist, R. B. Jadrich, and T. M. Truskett, Inverse methods for design of soft materials, The Journal of Chemical Physics **152**, 140902 (2020).

[145] A. Das and D. T. Limmer, Variational design principles for nonequilibrium colloidal assembly, The Journal of Chemical Physics , 1 (2021).

[146] J. Hénin, T. Lelièvre, M. R. Shirts, O. Valsson, and L. Delemotte, Enhanced sampling methods for molecular dynamics simulations [article v1.0], Living Journal of Computational Molecular Science **4**, 10.33011/livecoms.4.1.1583 (2022).

[147] D. Wang, Y. Wang, J. Chang, L. Zhang, H. Wang, and W. E., Efficient sampling of high-dimensional free energy landscapes using adaptive reinforced dynamics, Nature Computational Science **2**, 20 (2022).

[148] A. Das and D. T. Limmer, Nonequilibrium design strategies for functional colloidal assemblies, Proceedings of the National Academy of Sciences **120**, e2217242120 (2023).

[149] C. P. Goodrich, E. M. King, S. S. Schoenholz, E. D. Cubuk, and M. P. Brenner, Designing self-assembling kinetics with differentiable statistical physics models, Proceedings of the National Academy of Sciences **118**, e2024083118 (2021).

[150] U. T. Lieu and N. Yoshinaga, Inverse design of two-dimensional structure by self-assembly of patchy particles, The Journal of Chemical Physics **156**, 054901 (2022).

[151] A. Jhaveri, S. Loggia, Y. Qian, and M. E. Johnson, Discovering optimal kinetic pathways for self-assembly using automatic differentiation, Proceedings of the National Academy of Sciences **121**, e2403384121 (2024).

[152] A. I. Curatolo, O. Kimchi, C. P. Goodrich, R. K. Krueger, and M. P. Brenner, A computational toolbox for the assembly yield of complex and heterogeneous structures, Nature Communications **14**, 8328 (2023).

[153] M. F. Hagan and D. Chandler, Dynamic pathways for viral capsid assembly, Biophysical Journal **1**, 42 (2006).

[154] J. D. Perlmutter, M. R. Perkett, and M. F. Hagan, Pathways for virus assembly around nucleic acids, Journal of Molecular Biology **426**, 3148 (2014).

[155] M. Gupta, A. J. Pak, and G. A. Voth, Critical mechanistic features of HIV-1 viral capsid assembly, Science advances **9**, eadd7434 (2023).

[156] A. Zlotnick, Are weak protein-protein interactions the general rule in capsid assembly?, Virology **315**, 269 (2003).

[157] P. Ceres and A. Zlotnick, Weak protein-protein interactions are sufficient to drive assembly of hepatitis B virus capsids, Biochemistry **41**, 11525 (2002).

[158] F. M. Gartner, I. R. Graf, P. Wilke, P. M. Geiger, and E. Frey, Stochastic yield catastrophes and robustness in self-assembly, eLife **9**, e51020 (2020).

[159] C. J. Schlicksup and A. Zlotnick, Viral structural proteins as targets for antivirals, Current Opinion in Virology **45**, 43 (2020).

[160] M. F. Hagan and F. Mohajerani, Self-assembly coupled to liquid-liquid phase separation, Plos Computational Biology **19**, e1010652 (2023).

[161] A. Trubiano, Self-assembly analysis suite for HOOMD, `https://github.com/onehalfatsquared/SAASH` (2024).

[162] A. Trubiano, MultiMSM, `https://github.com/onehalfatsquared/MultiMSM` (2024).

[163] J. A. Anderson, J. Glaser, and S. C. Glotzer, HOOMD-blue: A Python package for high-performance molecular dynamics and hard particle Monte Carlo simulations, Computational Materials Science **173**, 109363 (2020).

[164] V. Ramasubramani, B. D. Dice, E. S. Harper, M. P. Spellings, J. A. Anderson, and S. C. Glotzer, Freud: A software suite for high throughput analysis of particle simulation data, Computer Physics Communications **254**, 107275 (2020).

[165] C. Sigl, E. Willner, W. Engelen, J. Kretzmann, K. Sachenbacher, A. Liedl, F. Kolbe, F. Wilsch, S. Aghvami, U. Protzer, M. Hagan, S. Fraden, and H. Dietz, Programmable icosahedral shell system for virus trapping, Nature Materials 10.1038/s41563-021-01020-4 (2021).

[166] W.-S. Wei, A. Trubiano, C. Sigl, S. Paquay, H. Dietz, M. F. Hagan, and S. Fraden, Hierarchical assembly is more robust than egalitarian assembly in synthetic capsids, Proceedings of the National Academy of Sciences **121**, 10.1073/pnas.2312775121 (2024).

[167] D. J. Wales, The energy landscape as a unifying theme in molecular science, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **363**, 357 (2005).

[168] S. N. Fejer, T. R. James, J. Hernández-Rojas, and D. J. Wales, Energy landscapes for shells assembled from pentagonal and hexagonal pyramids, Physical Chemistry Chemical Physics **11**, 2098 (2009).

[169] I. G. Johnston, A. A. Louis, and J. P. K. Doye, Modelling the self-assembly of virus capsids, Journal of Physics: Condensed Matter **22**, 104101 (2010).

[170] J. D. Weeks, D. W. Chandler, and H. C. Andersen, Role of repulsive forces in determining the equilibrium structure of simple liquids, Journal of Chemical Physics **54**, 5237 (1971).

[171] D. Rapaport, J. Johnson, and J. Skolnick, Supramolecular self-assembly: Molecular dynamics modeling of polyhedral shell formation, Computer Physics Communications **121–122**, 231 (1999).

[172] D. C. Rapaport, Studies of reversible capsid shell growth, Journal of Physics: Condensed Matter **22**, 104115 (2010).

[173] F. Noé, I. Horenko, C. Schütte, and J. C. Smith, Hierarchical analysis of conformational dynamics in biomolecules: Transition networks of metastable states, The Journal of Chemical Physics **126**, 10.1063/1.2714539 (2007).

[174] J.-H. Prinz, J. D. Chodera, and F. Noé, Spectral rate theory for two-state kinetics, Physical Review X **4**, 011020 (2014).

[175] W. C. Swope, J. W. Pitera, and F. Suits, Describing protein folding kinetics by molecular dynamics simulations. 1. Theory, The Journal of Physical Chemistry B **108**, 6571 (2004).

[176] M. E. Karpen, D. J. Tobias, and C. L. I. Brooks, Statistical clustering techniques for the analysis of long molecular dynamics trajectories: Analysis of 2.2-ns trajectories of YPGDV, Biochemistry **32**, 412 (1993).

[177] B. L. de Groot, X. Daura, A. E. Mark, and H. Grubmüller, Essential dynamics of reversible peptide folding: Memory-free conformational dynamics governed by internal hydrogen bonds11Edited by R. Huber, Journal of Molecular Biology **309**, 299 (2001).

[178] N. Singhal, C. D. Snow, and V. S. Pande, Using path sampling to build better Markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin, The Journal of Chemical Physics **121**, 415 (2004).

[179] M. Andrec, A. K. Felts, E. Gallicchio, and R. M. Levy, Protein folding pathways from replica exchange simulations and a kinetic network model, Proceedings of the National Academy of Sciences **102**, 6801 (2005).

[180] X. Zeng, B. Li, Q. Qiao, L. Zhu, Z.-Y. Lu, and X. Huang, Elucidating dominant pathways of the nano-particle self-assembly process, Physical Chemistry Chemical Physics **18**, 23494 (2016).

[181] X. Zheng, L. Zhu, X. Zeng, L. Meng, L. Zhang, D. Wang, and X. Huang, Kinetics-Controlled Amphiphile Self-Assembly Processes, The Journal of Physical Chemistry Letters **8**, 1798 (2017).

[182] X. Zeng, Z.-W. Li, X. Zheng, L. Zhu, Z.-Y. Sun, Z.-Y. Lu, and X. Huang, Improving the productivity of monodisperse polyhedral cages by the rational design of kinetic self-assembly pathways, Physical Chemistry Chemical Physics **20**, 10030 (2018).

[183] A. Mardt, L. Pasquali, H. Wu, and F. Noé, VAMPnets: Deep learning of molecular kinetics, Nature Communications **9**,

10.1038/s41467-017-02388-1 (2018).
[184] S. Schultze and H. Grubmüller, Time-Lagged Independent Component Analysis of Random Walks and Protein Dynamics, Journal of Chemical Theory and Computation **17**, 5766 (2021).
[185] T. Hempel, M. J. del Razo, C. T. Lee, B. C. Taylor, R. E. Amaro, and F. Noé, Independent Markov decomposition: Toward modeling kinetics of biomolecular complexes, Proceedings of the National Academy of Sciences **118**, e2105230118 (2021).
[186] I. Satake, *Linear Algebra*, Pure and Applied Mathematics (M. Dekker, 1975).
[187] G. R. Bowman, D. L. Ensign, and V. S. Pande, Enhanced modeling via network theory: Adaptive sampling of markov state models, Journal of Chemical Theory and Computation **6**, 787 (2010).
[188] S. Doerr and G. De Fabritiis, On-the-fly learning and sampling of ligand binding by high-throughput molecular simulations, Journal of Chemical Theory and Computation **10**, 2064 (2014).
[189] V. A. Voelz, B. Elman, A. M. Razavi, and G. Zhou, Surprisal metrics for quantifying perturbed conformational dynamics in markov state models, Journal of Chemical Theory and Computation **10**, 5716 (2014).
[190] M. I. Zimmerman and G. R. Bowman, FAST conformational searches by balancing exploration/exploitation tradeoffs, Journal of Chemical Theory and Computation **11**, 5747 (2015).
[191] Z. Shamsi, A. S. Moffett, and D. Shukla, Enhanced unbiased sampling of protein dynamics using evolutionary coupling information, Scientific reports **7**, 12700 (2017).
[192] Z. Shamsi, K. J. Cheng, and D. Shukla, Reinforcement learning based adaptive sampling: REAPing rewards by exploring protein conformational landscapes, The Journal of Physical Chemistry B **122**, 8386 (2018).
[193] M. I. Zimmerman, J. R. Porter, X. Sun, R. R. Silva, and G. R. Bowman, Choice of Adaptive Sampling Strategy Impacts State Discovery, Transition Probabilities, and the Apparent Mechanism of Conformational Changes, J Chem Theory Compute **14**, 5459 (2018).
[194] H. Wan and V. A. Voelz, Adaptive Markov state model estimation using short reseeding trajectories, The Journal of chemical physics **152**, 024103 (2020).
[195] M. F. Hagan, O. M. Elrad, and R. L. Jack, Mechanisms of kinetic trapping in self-assembly and phase transformation, Journal of Chemical Physics **135**, 104115 (2011).
[196] E. Vanden-Eijnden, Transition path theory, in *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1*, edited by M. Ferrario, G. Ciccotti, and K. Binder (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 453–493.
[197] P. Metzner, C. Schütte, and E. Vanden-Eijnden, Transition path theory for markov jump processes, Multiscale Modeling & Simulation **7**, 1192 (2009).
[198] W. E and E. Vanden-Eijnden, Transition-path theory and path-finding algorithms for the study of rare events, Annual Review of Physical Chemistry **61**, 391 (2010).
[199] L. Helfmann, E. Ribera-Borel, C. Schütte, and P. Koltai, Extending transition path theory: Periodically driven and finite-time dynamics, The Journal of Nonlinear Science **30**, 3321 (2020).
[200] C. Dellago, P. G. Bolhuis, and P. L. Geissler, Transition path sampling, Special Volume in Memory of Ilya Prigogine **123**, 1 (2002).
[201] B. Peters, Reaction coordinates and mechanistic hypothesis tests, Annual Review of Physical Chemistry, Vol 62 **67**, 669 (2016).
[202] D. Endres and A. Zlotnick, Model-based analysis of assembly kinetics for virus capsids or other spherical polymers, Biophysical Journal **83**, 1217 (2002).
[203] M. F. Hagan, Modeling viral capsid assembly, in *Advances in Chemical Physics: Volume 155* (John Wiley & Sons, Ltd, 2014) Chap. 1, pp. 1–68.
[204] I. Mizrahi, R. Bruinsma, and J. Rudnick, Spanning tree model and the assembly kinetics of RNA viruses, Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **106**, 044405 (2022).
[205] A. Zlotnick, To build a virus capsid: An equilibrium model of the self assembly of polyhedral protein complexes, Journal of Molecular Biology **241**, 59 (1994).
[206] P. Moisant, H. Neeman, and A. Zlotnick, Exploring the paths of (virus) assembly, Biophysical Journal **99**, 1350 (2010).
[207] N. Singhal and V. S. Pande, Error analysis and efficient sampling in Markovian state models for molecular dynamics, The Journal of Chemical Physics **123**, 204909 (2005), https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.2116947/15379877/204909_1_online.pdf.
[208] N. S. Hinrichs and V. S. Pande, Calculation of the distribution of eigenvalues and eigenvectors in Markovian state models for molecular dynamics, The Journal of Chemical Physics **126**, 244101 (2007).
[209] S. Bhattacharya and A. Chatterjee, Uncertainty quantification for Markov state models of biomolecules constructed using rare event acceleration techniques, The Journal of Chemical Physics **150**, 044106 (2019).
[210] N. Kozlowski and H. Grubmüller, Uncertainties in markov state models of small proteins, Journal of Chemical Theory and Computation **19**, 5516 (2023).
[211] B. Efron, Bootstrap Methods: Another Look at the Jackknife, The Annals of Statistics **7**, 1 (1979).
[212] B. Efron, Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods, Biometrika **68**, 589 (1981), https://academic.oup.com/biomet/article-pdf/68/3/589/581658/68-3-589.pdf.
[213] B. Trendelkamp-Schroer and F. Noé, Efficient estimation of rare-event kinetics, Phys. Rev. X **6**, 011009 (2016).
[214] F. Noé, S. Olsson, J. Köhler, and H. Wu, Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning, Science **365**, eaaw1147 (2019).
[215] L. Bonati, G. Piccini, and M. Parrinello, Deep learning the slow modes for rare events sampling, Proceedings of the National Academy of Sciences **118**, e2113533118 (2021).
[216] S. Falkner, A. Coretti, S. Romano, P. L. Geissler, and C. Dellago, Conditioning Boltzmann generators for rare event sampling, Machine Learning: Science and Technology **4**, 035050 (2023).
[217] A. Zlotnick, J. M. Johnson, P. W. Wingfield, S. J. Stahl, and D. Endres, A theoretical model successfully identifies features of hepatitis B virus capsid assembly, Biochemistry **38**, 14644 (1999).
[218] M. F. Hagan and O. M. Elrad, Understanding the concentration dependence of viral capsid assembly kinetics-the origin of the lag time and identifying the critical nucleus size, Biophysical Journal **98**, 1065 (2010).
[219] A. Zlotnick and S. J. Stray, How does your virus grow? Understanding and interfering with virus assembly, Trends in Biotechnology **21**, 536 (2003).

[220] P. E. Prevelige, D. Thomas, and J. King, Nucleation and growth phases in the polymerization of coat and scaffolding subunits into icosahedral procapsid shells, Biophysical Journal **64**, 824 (1993).

[221] T. Q. Zhang and R. Schwartz, Simulation study of the contribution of oligomer/oligomer binding to capsid assembly kinetics, Biophysical Journal **90**, 57 (2006).

[222] A. Tan, A. J. Pak, D. R. Morado, G. A. Voth, and J. A. G. Briggs, Immature HIV-1 assembles from Gag dimers leaving partial hexamers at lattice edges as potential substrates for proteolytic maturation, Proceedings of the National Academy of Sciences of the United States of America **118**, 10.1073/pnas.2020054118 (2021).

[223] A. Yu, A. J. Pak, P. He, V. Monje-Galvan, L. Casalino, Z. Gaieb, A. C. Dommer, R. E. Amaro, and G. A. Voth, A multiscale coarse-grained model of the SARS-CoV-2 virion, Biophysical journal **120**, 1097 (2021).

[224] D. Rapaport, The role of reversibility in viral capsid growth: A paradigm for self-assembly, Physical Review Letters **101**, 186101 (2008).

[225] A. W. Wilber, J. P. K. Doye, A. A. Louis, E. G. Noya, M. A. Miller, and P. Wong, Reversible self-assembly of patchy particles into monodisperse icosahedral clusters, The Journal of Chemical Physics **127**, 085106 (2007).

[226] A. W. Wilber, J. P. K. Doye, A. A. Louis, and A. C. F. Lewis, Monodisperse self-assembly in a model with protein-like interactions, Journal of Chemical Physics **131**, 175102 (2009).

[227] S. Cheng, A. Aggarwal, and M. J. Stevens, Self-assembly of artificial microtubules, Soft Matter **8**, 5666 (2012).

[228] D. Rapaport, Self-assembly of polyhedral shells: A molecular dynamics study, **70**, 051905 (2004).

[229] H. D. Nguyen, V. S. Reddy, and C. L. Brooks, Invariant polymorphism in virus capsid assembly, Journal of The American Chemical Society **131**, 2606 (2009).

[230] D. C. Rapaport, Molecular dynamics simulation of reversibly self-assembling shells in solution using trapezoidal particles, Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **86**, 051917 (2012).

[231] J. Wagner and R. Zandi, The robust assembly of small symmetric nanoshells, Biophys. J. **109**, 956 (2015).

[232] P. van der Schoot and R. Zandi, Kinetic Theory of Virus Capsid Assembly, Phys. Biol. **4**, 296 (2007).

[233] B. C. Bag, Nonequilibrium stochastic processes: Time dependence of entropy flux and entropy production, Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **66**, 026122 (2002).

[234] S. Vaikuntanathan and C. Jarzynski, Dissipation and lag in irreversible processes, Europhysics Letters **87**, 60005 (2009).

[235] D. Mandal, K. Klymko, and M. R. DeWeese, Entropy production and fluctuation theorems for active matter, Phys. Rev. Lett. **119**, 258001 (2017).

[236] G. T. Landi and M. Paternostro, Irreversible entropy production: From classical to quantum, Reviews of Modern Physics **93**, 035008 (2021).

[237] A. Gomez-Marin, T. Schmiedl, and U. Seifert, Optimal protocols for minimal work processes in underdamped stochastic thermodynamics, The Journal of Chemical Physics **129**, 024114 (2008), https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.2948948/6770253/024114_1_online.pdf.

[238] D. A. Sivak and G. E. Crooks, Thermodynamic geometry of minimum-dissipation driven barrier crossing, Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **94**, 052106 (2016).

[239] S. J. Large and D. A. Sivak, Optimal discrete control: Minimizing dissipation in discretely driven nonequilibrium systems, Journal of Statistical Mechanics: Theory and Experiment **2019**, 083212 (2019).

[240] S. Tafoya, S. J. Large, S. Liu, C. Bustamante, and D. A. Sivak, Using a system's equilibrium behavior to reduce its energy dissipation in nonequilibrium processes, Proceedings of the National Academy of Sciences **116**, 5920 (2019).

[241] T. R. Gingrich, G. M. Rotskoff, G. E. Crooks, and P. L. Geissler, Near-optimal protocols in complex nonequilibrium transformations, Proceedings of the National Academy of Sciences **113**, 10263 (2016).

[242] M. Nakazato and S. Ito, Geometrical aspects of entropy production in stochastic thermodynamics based on Wasserstein distance, **3**, 043093 (2021).

[243] A. Dechant, Minimum entropy production, detailed balance and Wasserstein distance for continuous-time Markov processes, Journal of Physics A: Mathematical and Theoretical **55**, 094001 (2022).

[244] S. Blaber and D. A. Sivak, Optimal control in stochastic thermodynamics, Journal of Physics Communications **7**, 033001 (2023).

[245] A. I. Brown and D. A. Sivak, Theory of nonequilibrium free energy transduction by molecular machines, Chemical Reviews **120**, 434 (2020).

[246] P. Abiuso, V. Holubec, J. Anders, Z. Ye, F. Cerisola, and M. Perarnau-Llobet, Thermodynamics and optimal protocols of multidimensional quadratic Brownian systems, Journal of Physics Communications **6**, 063001 (2022).

[247] M. D. Louwerse and D. A. Sivak, Multidimensional minimum-work control of a 2D Ising model, The Journal of Chemical Physics **156**, 194108 (2022).

[248] S. Whitelam and J. D. Schmit, Low-dissipation self-assembly protocols of active sticky particles, Journal of Crystal Growth **600**, 126912 (2022).

[249] A. Zhong and M. R. DeWeese, Limited-control optimal protocols arbitrarily far from equilibrium, Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **106**, 044135 (2022).

[250] L. K. Davis, K. Proesmans, and É. Fodor, Active matter under control: Insights from response theory, Physical Review X **14**, 011012 (2024).

[251] D.-K. Kim, Y. Bae, S. Lee, and H. Jeong, Learning entropy production via neural networks, Physical Review Letters **125**, 140604 (2020).

[252] A. Nir, E. Sela, R. Beck, and Y. Bar-Sinai, Machine-learning iterative calculation of entropy for physical systems, Proceedings of the National Academy of Sciences **117**, 30234 (2020).

[253] S. Otsubo, S. Ito, A. Dechant, and T. Sagawa, Estimating entropy production by machine learning of short-time fluctuating currents, Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **101**, 062106 (2020).

[254] S. Otsubo, S. K. Manikandan, T. Sagawa, and S. Krishnamurthy, Estimating time-dependent entropy production from non-equilibrium trajectories, Communications Physics **5**, 11 (2022).

[255] N. M. Boffi and E. Vanden-Eijnden, Deep learning probability flows and entropy production rates in active matter (2023).

[256] N. M. Boffi and E. Vanden-Eijnden, Deep learning probability flows and entropy production rates in active matter (2023), arXiv:2309.12991 [cond-mat.stat-mech].

[257] S. Whitelam, Demon in the machine: Learning to extract work and absorb entropy from fluctuating nanosystems, Physical Review X **13**, 021005 (2023).

[258] M. C. Engel, J. A. Smith, and M. P. Brenner, Optimal control of nonequilibrium systems through automatic differentiation, Physical Review X **13**, 041032 (2023).

[259] A. Zlotnick, Distinguishing reversible from irreversible virus capsid assembly, Journal of Molecular Biology **366**, 14 (2007).

[260] C. Kim, C. Schlicksup, L. Barnes, M. Jarrold, A. Patterson, B. Bothner, and A. Zlotnick, HBV core-directed antivirals and importin β can synergistically disrupt capsids, Microscopy and Microanalysis **27**, 1130 (2021).

[261] P. Kondylis, C. J. Schlicksup, N. E. Brunk, J. Zhou, A. Zlotnick, and S. C. Jacobson, Competition between normative and drug-induced virus self-assembly observed with single-particle methods, Journal of the American Chemical Society **141**, 1251 (2018).

[262] C. J. Schlicksup, J. C.-Y. Wang, S. Francis, B. Venkatakrishnan, W. W. Turner, M. VanNieuwenhze, and A. Zlotnick, Hepatitis B virus core protein allosteric modulators can distort and disrupt intact capsids, Elife **7**, e31473 (2018).

[263] A. Pavlova, L. Bassit, B. D. Cox, M. Korablyov, C. Chipot, D. Patel, D. L. Lynch, F. Amblard, R. F. Schinazi, and J. C. Gumbart, The mechanism of action of hepatitis B virus capsid assembly modulators can be predicted from binding to early assembly intermediates, Journal of Medicinal Chemistry **65**, 4854 (2022).

[264] M. Wang, J. Zhang, Y. Dou, M. Liang, Y. Xie, P. Xue, L. Liu, C. Li, Y. Wang, F. Tao, X. Zhang, H. Hu, K. Feng, L. Zhang, Z. Wu, Y. Chen, P. Zhan, and H. Jia, Design, synthesis, and biological evaluation of novel thioureidobenzamide (TBA) derivatives as HBV capsid assembly modulators, Journal of Medicinal Chemistry **66**, 13968 (2023).