Supplementary Material

S1. CONNECTION BETWEEN DGA AND MARKOV STATE MODELING

Here, we describe in detail the connection between DGA and certain dynamical estimates calculated using a MSM. To map the general dynamics onto the state space of the Markov Chain, we make three assumptions.

Assumption S1.1. *Each Markov state S_i is contained entirely in either D or in D^c.*

Assumption S1.2. *The boundary conditions b can be expressed as*

$$b(x) = \sum_{l \in D^c}^{M'} b_l \mathbb{1}_{S_l}(x). \tag{1}$$

Assumption S1.3. *For any $\mathcal{L}_p^\dagger$ considered, p can be written as*

$$p(x) = \sum_{j \in D}^{M} \frac{p_j}{\langle \mathbb{1}_j \rangle} \mathbb{1}_j(x) + \sum_{l \in D^c}^{M'} \frac{p_l}{\langle \mathbb{1}_l \rangle} s\mathbb{1}_l(x). \tag{2}$$

The first assumption is necessary for the basis set to obey the homogeneous boundary conditions, and can be enforced explicitly in the construction of the MSM. The second two assumptions will be required to make the action of $\mathcal{L}$ representable as the action of matrices on vectors over the MSM states. While these assumptions should not be expected to hold for general b and p, in the correct limit of infinite sampling and sufficiently small Markov states, we expect (1) and (2) to be arbitrarily good approximations. In fact, for most b in Section III, assumption S1.2 can hold exactly. We also note that the vector p_i sums to one, as

$$\begin{aligned}
1 &= \int p(x)\mu(dx) \\
&= \int \sum_{j \in D}^{M} \frac{p_j}{\langle \mathbb{1}_j \rangle} \mathbb{1}_j(x) + \sum_{l \in D^c}^{M'} \frac{p_l}{\langle \mathbb{1}_l \rangle} \mathbb{1}_l(x)\mu(dx) \\
&= \sum_{j \in D}^{M} p_j + \sum_{l \in D^c}^{M'} p_l.
\end{aligned}$$

Consequently, p_i is a probability distribution over the MSM state-space.

A. Equations with the Transition Operator

We first consider equations that take the form of (33). As our guess, we will use (38). Substituting into (42), applying Assumption S1.2, and dividing by $\langle 1_{S_i} \rangle$, we arrive at

$$\sum_{j \in D}^{M} \frac{1}{\Delta t} (P - I)_{ij} a_j = \eta_i - \sum_{l \in D^c}^{M'} \frac{1}{\Delta t} (P - I)_{il} b_l. \tag{3}$$

Here P_{ij} is the MSM transition matrix defined in (3) with a time lag of Δt, and η_i is defined as

$$\eta_i = \frac{\langle \mathbb{1}_i, h \rangle}{\langle \mathbb{1}_i \rangle} \tag{4}$$

This can be rewritten as

$$\begin{gathered}
\sum_{j} \frac{1}{\Delta t} (P - I)_{ij} a_j = \eta_i \text{ for } i \in D \\
a_i = b_i \text{ for } i \in D^c
\end{gathered} \tag{5}$$

where the sum is over states on the entire domain. This is equivalent to (33) for the dynamics given by the MSM.

B. Equations with Transition Adjoints

For equations that take the form of (34) we again begin with (42), this time with terms defined by equations (48), (50), and (49). Substituting in our guess function and Assumptions S1.3 and S1.1, we have

$$\begin{aligned}&\sum_{j\in D}^{M} \langle \mathcal{L}\mathbb{1}_i, \mathbb{1}_j\rangle \left(\frac{p_j}{\langle \mathbb{1}_j\rangle}\right) a_j \\ &= \left(\frac{p_i}{\langle \mathbb{1}_i\rangle}\right) \langle \mathbb{1}_i, h\rangle - \sum_{l\in D^c}^{M'} \langle \mathcal{L}\mathbb{1}_i, \mathbb{1}_l\rangle\, b_l \left(\frac{p_l}{\langle \mathbb{1}_l\rangle}\right)\end{aligned} \tag{6}$$

We then divide both sides by p_i. Applying the definition of P_{ij}, we arrive at

$$\sum_{j\in D}^{M} p_i^{-1} \left(P - I\right)_{ij}^{T} p_j a_j = \eta_i - \sum_{l\in D^c}^{M'} p_i^{-1} \left(P - I\right)_{il} p_l b_l. \tag{7}$$

which, as before, is equivalent to solving

$$\begin{aligned}\sum_{j} p_i^{-1} \left(P - I\right)_{ij}^{T} p_j &= \eta_i \text{ for } i \in D \\ a_i &= b_i \text{ for } i \in D^c.\end{aligned} \tag{8}$$

Comparing with (26), we see that the matrix with elements $p_i^{-1} \left(P - I\right)_{ij}^{T} p_j$ is the weighted adjoint of the MSM generator against p_i. Consequently, (8) is equivalent to (34) for the MSM.

S2. DETAILS OF DIFFUSION MAP CONSTRUCTION

Here, we give the specific kernel and parameter choice used in our calculations used to construct the diffusion map in our calculations. Our procedure closely follows work in references 1 and 2. Specifically, our algorithm corresponds to the parameter choice $\alpha = 0$ and $\beta = -1/d$ in reference 2 and not performing the bandwidth normalization in equation (5).

A. Kernel Construction

As in Section V, let x_m be a collection of N datapoints. We define the initial bandwidth function

$$\varsigma_0(x_m) = \frac{1}{k_0} \sum_{l=1}^{k_0} ||x_m - x_{I(m,l)}||^2$$

where $I(m,l)$ is the index of the l'th nearest neighbor to point x_m (not including x_m). Here k_0 is a neighborhood parameter giving the number of nearest neighbors considered, we follow reference 2 and set it to 7. We then construct the kernel density estimate

$$\begin{aligned}q(x_m) &= \frac{(2\pi\varepsilon_0)^{-d/2}}{N\varsigma_0(x_m)^d} \sum_{n=1}^{N} K_0(x_m, x_n; \varepsilon_0), \text{ where} \\ K_0(x_m, x_n; \varepsilon_0) &= \exp\left(\frac{-||x_m - x_n||^2}{2\varepsilon_0\varsigma_0(x_m)\varsigma_0(x_n)}\right)\end{aligned}$$

where d is the intrinsic dimensionality of the data manifold and ε_0 is a bandwidth parameter. To estimate d and a good choice for ε_0, we consider all possible choices of eps_0 of the form 2^k with $k = -40, -39, \ldots, 39, 40$. We then set

$$d = \frac{2}{\ln(2)} \max_k \left[\ln\left(\frac{\sum_{m,n} K_0(x_m, x_n, 2^{k+1})}{\sum_{m,n} K_0(x_m, x_n, 2^{k})}\right)\right] \tag{9}$$

Reference 1 suggests setting ε_0 by using the k where the right-hand-side attains its maximum. In practice we find this can be overly aggressive, so we subsequently multiply ε_0 by 2. We then construct the Diffusion map kernel matrix as

$$K(x_m, x_n) = \exp\left(\frac{||x-y||^2}{\varepsilon q(x_m)^{-1/d} q(x_n)^{-1/d}}\right) \tag{10}$$

where we select the ε using the same procedure as before.

B. Out-of-sample Extension for the Diffusion-Map Basis

To predict the values of the quantities in Section III at new datapoints, we will need to extend the diffusion-map basis and guess functions to new configurations. Initially, one might attempt this by constructing a new diffusion map matrix that contains both the old and the new points and recomputing the guess and eigenvectors. However, not only would this procedure be expensive, it would change the values of the basis and guess functions on the old points. Consequently, the estimates of a_j would be incorrect, and the entire DGA scheme would need to be repeated. We therefore seek a method for extending the basis and guess functions to new points that leave their values on older points unchanged.

Let x_ν be a new point added to the dataset. To extend the basis functions to x_ν, we can use the established method of Nyström extension.[3, 4] Let φ_i be an eigenvector of the submatrix discussed in Section V, and let κ_i be the associated eigenvalue. The estimate of the basis function on x_ν is given by

$$\varphi_i(x_\nu) = \frac{1}{\kappa_i}\frac{\sum_m K_\varepsilon(x_m, x_\nu)\varphi_i(x_m)}{\sum_m K_\varepsilon(x_m, x_\nu)} \tag{11}$$

To extend the guess function to new configurations, we introduce a new method based on the Jacobi method.[5] We first consider $\hat{P}$, a new diffusion map matrix built using both the old datapoints $x_{1..N}$ and the new datapoint x_ν. The guess function associated with $\hat{P}$ would then solve the problem

$$\left(\hat{P} - I\right) g = h \tag{12}$$

for all of the points in D. We will construct our estimate of g at the new point by considering a single iteration of the Jacobi method for solving (12). Our initial vector takes values of g_m on x_m and 0 on x_ν. This gives us the following out-of-sample extension formula

$$g_\nu = \frac{1}{\hat{P}_{\nu\nu} - 1}\left(h_\nu - \sum_{m=1}^{N} \hat{P}_{\nu m} g_m\right) \tag{13}$$

where the sum runs only over points in the original dataset. This can be further simplified using the definition of $\hat{P}$ to

$$g_\nu = \frac{\sum_{m=1}^{N} K_\varepsilon\left(x_\nu, x_m\right) g_m}{\sum_{m=1}^{N} K_\varepsilon\left(x_\nu, x_m\right)} - h_\nu\left(1 + \frac{K_\varepsilon\left(x_\nu, x_\nu\right)}{\sum_{m=1}^{N} K_\varepsilon\left(x_\nu, x_m\right)}\right). \tag{14}$$

S3. DERIVATION OF TRANSITION PATH THEORY REACTIVE FLUX AND RATE IN DISCRETE TIME

Transition path theory was originally formulated for diffusion processes[6] and was extended to finite-state Markov jump processes.[7] Here, we derive analogous equations for discrete-time Markov chains on arbitrary state spaces. The derivation closely follows reference 6. Let $x(t)$ be a single trajectory ergodically sampling the stationary measure. We will extend the trajectory both forwards and backwards in time so the time index t takes values from $-\infty$ to ∞. For all t, let

$$t^+_{AB}(t) = \min\left\{t' | t' \geq t, x(t') \in A \cup B\right\} \tag{15}$$

$$t^-_{AB}(t) = \max\left\{t' | t' \leq t, x(t') \in A \cup B\right\} \tag{16}$$

be the next time the system entered A or B and the most recent time the system left A or B, respectively. Now let C be as in (28). The total reactive current is defined as

$$\begin{aligned} I_{B\to A} = \lim_{T\to\infty} \frac{1}{2T} \sum_{t\in[-T,T]} & \left[\mathbb{1}_C\left(x(t)\right)\mathbb{1}_{C^c}\left(x(t+\Delta t)\right)\right. \\ & \left.-\ \mathbb{1}_{C^c}\left(x(t)\right)\mathbb{1}_C\left(x(t+\Delta t)\right)\right] \\ & \left[\mathbb{1}_A\left(x(t^-_{AB}(t)\right)\mathbb{1}_B\left(x(t^+_{AB}(t+\Delta t)\right)\right] \end{aligned} \tag{17}$$

Using ergodicity and the strong Markov property, we can rewrite this as an average against $\rho_{\Delta t}$.

$$I_{B\to A} = \int \mathbb{1}_{C^c}(y)q_+(y)q_-(x)\mathbb{1}_C(x)\pi(x)\rho_{\Delta t}(dx,dy) \tag{18}$$

$$- \int \mathbb{1}_C(y)q_+(y)q_-(x)\mathbb{1}_{C^c}(x)\pi(x)\rho_{\Delta t}(dx,dy) \tag{19}$$

This is the discrete-time equivalent of equation (30) in reference 6. Applying the definition of the generator and observing that that $\mathbb{1}_C(x)\mathbb{1}_{C^c}(x) = 0$ everywhere gives (28). We then arrive at (29) in our work by the same arguments as in reference 8.

S4. GRID-BASED REFERENCE SCHEME

Here we discuss the scheme used to calculate the reference values for our test system in Sections V and VI. Instead of considering the discrete time process directly, we will attempt to approximate the dynamics of the continuous-time Brownian dynamics on the test potential. To this end, we define a Markov hopping process on a grid that converges to the continuous time dynamics as the grid becomes finer. Specifically, we allow nearest neighbor hops on a square grid with spacing ϵ. The hopping probabilities are given by

$$\begin{aligned} P(x+\epsilon,y) &= \left(\frac{1}{4}\right)\left(\frac{1}{1+\exp\left[U(x+\epsilon,y)-U(x,y)\right]}\right) \\ P(x-\epsilon,y) &= \left(\frac{1}{4}\right)\left(\frac{1}{1+\exp\left[U(x-\epsilon,y)-U(x,y)\right]}\right) \\ P(x,y+\epsilon) &= \left(\frac{1}{4}\right)\left(\frac{1}{1+\exp\left[U(x,y+\epsilon)-U(x,y)\right]}\right) \\ P(x,y-\epsilon) &= \left(\frac{1}{4}\right)\left(\frac{1}{1+\exp\left[U(x,y-\epsilon)-U(x,y)\right]}\right) \\ P(x,y) =& 1 - P(x+\epsilon,y) - P(x-\epsilon,y) \\ & - P(x,y+\epsilon) - P(x,y-\varepsilon). \end{aligned} \tag{20}$$

Here $P(x \pm \epsilon, y)$ is the probability of hopping one grid point to the right or left, $P(x, y \pm \epsilon)$ is the probability of hopping up or down the grid, and $P(x, y)$ is the probability of remaining in place.

We will not give a full proof of convergence. Instead we merely demonstrate that as $\epsilon \to 0$, we approximate the infinitesimal generator $\mathcal{L}^{\text{brwn}}$ for Brownian Dynamics. Let P be the transition matrix associated with the transition probabilities given by (20), f be a three-times continuously differentiable function, and the vector $\vec{f}$ the values of f evaluated at each grid point. In the limit of $\epsilon \to 0$,

$$\frac{16(P-I)\vec{f}}{\epsilon^2}(x,y) = \mathcal{L}^{\text{brwn}}f(x,y) + \mathcal{O}(\epsilon) \tag{21}$$

where $\mathcal{L}^{\text{brwn}}$ is the infinitesimal generator for Brownian dynamics with isotropic diffusion constant,

$$\begin{aligned} \mathcal{L}^{\text{brwn}}f(x,y) =& -\partial_x U(x,y)\partial_x f(x,y) - \partial_y U(x,y)\partial_y f(x,y) \\ & + \partial_x^2 f(x,y) + \partial_y^2 f(x,y). \end{aligned}$$

To demonstrate this, we write $(P-I)\vec{f}$ explicitly as

$$\begin{aligned}(P-I)f(x,y) =&P(x+\epsilon,y)f(x+\epsilon,y)\\ &+P(x-\epsilon,y)f(x-\epsilon,y)\\ &+P(x,y+\epsilon)f(x,y+\epsilon)\\ &+P(x,y-\epsilon)f(x,y-\epsilon)\\ &+P(x,y)f(x,y)+\mathcal{O}(\epsilon^3)\end{aligned}$$

If we expand f to second order around (x,y), the zeroth order term cancels, leaving

$$\begin{aligned}(P-I)f(x,y) =&P(x+\epsilon,y)\left(\epsilon\partial_x f+\frac{1}{2}\epsilon^2\partial_x^2 f\right)\\ &-P(x-\epsilon,y)\left(\epsilon\partial_x f-\frac{1}{2}\epsilon^2\partial_x^2 f\right)\\ &+P(x,y+\epsilon r)\left(\epsilon\partial_y f+\frac{1}{2}(\epsilon r)^2\partial_y^2 f\right)\\ &-P(x,y-\epsilon r)\left(\epsilon\partial_y f-\frac{1}{2}(\epsilon r)^2\partial_y^2 f\right)+\mathcal{O}(\epsilon^3)\end{aligned}$$

We then expand the transition probabilities to first order, giving

$$\begin{aligned}P(x\pm\epsilon,y)&=\frac{1}{8}\left(1\mp\frac{1}{2}\partial_x U(x,y)\epsilon\right)+\mathcal{O}(\epsilon^2)\\ P(x,y\pm\epsilon r)&=\frac{1}{8}\left(1\mp\frac{1}{2}\partial_x U(x,y)\epsilon\right)+\mathcal{O}(\epsilon^2).\end{aligned}$$

Substituting, simplifying, and multiplying by $16/\epsilon^2$ gives (21).

To estimate the reference quantities for our test system, we constructed a square grid on the interval $-2.5\leq x\leq 1.5$ and $-1.5\leq y\leq 2.5$ with grid spacing of 0.005. We then construct the transition rate matrix $16(P-I)/\epsilon^2$, and estimate the dynamical quantities using the corresponding formulas in Section III.

S5. BASIS SIZE CHOICE FOR MÜLLER-BROWN MODEL

In Figure S1, we show the dependence of the root-mean-square error in the committor on basis size for the Müller-Brown model. While using 1000 basis functions gives a slightly better result at higher dimensions, it is not enough to appreciably change the trends depicted in Figure 2. However, choosing 1000 or more basis functions gives worse results for the two-dimensional system. We therefore chose to use 500 dimensions to avoid giving the impression that the diffusion-map basis outperforms the MSM basis at low dimensions.

S6. NUMERICAL EFFECT OF ENFORCING DETAILED BALANCE

To test the effect of enforcing detailed balance in MSMs through a maximum likelihood procedure, we returned to our two-dimensional test potential without any additional nuisance degrees of freedom. Using the clusterings described in Section V A, we constructed MSMs in PyEMMA both with and without the reversible option set to True. We then estimated the mean first-passage time from state B into state A, using the states depicted in Figure 1A. As before, we repeated this procedure over thirty replicates. Moreover, we also varied the number of short trajectories included in the dataset to observe trends in statistical convergence.

Our results are given in Figure S2. The mean first-passage time calculated using reversible MSMs, depicted in panel A, grows unboundedly with increasing basis size. To demonstrate that this not due to the nature of the data, we repeated the calculation on a long equilibrium trajectory of commensurate length. Our results, shown in panel B, exhibit the same phenomenon. We also varied the error tolerance for convergence, as well as the minimum count required for connectivity. Neither affected the results. Moreover, an in-house code for the iteration described in

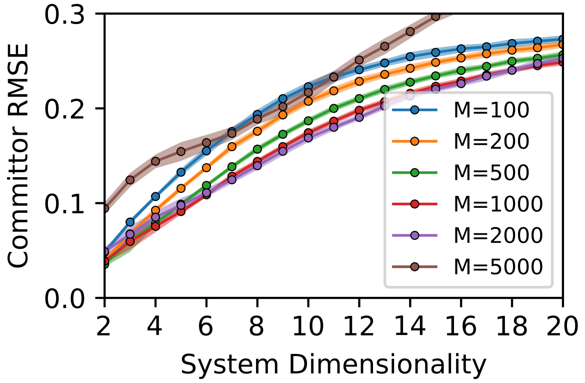

FIG. S1. Dependence of the MSM committor root-mean-square error (RMSE) on the number of clusters. Different curves correspond to different numbers of Markov states.

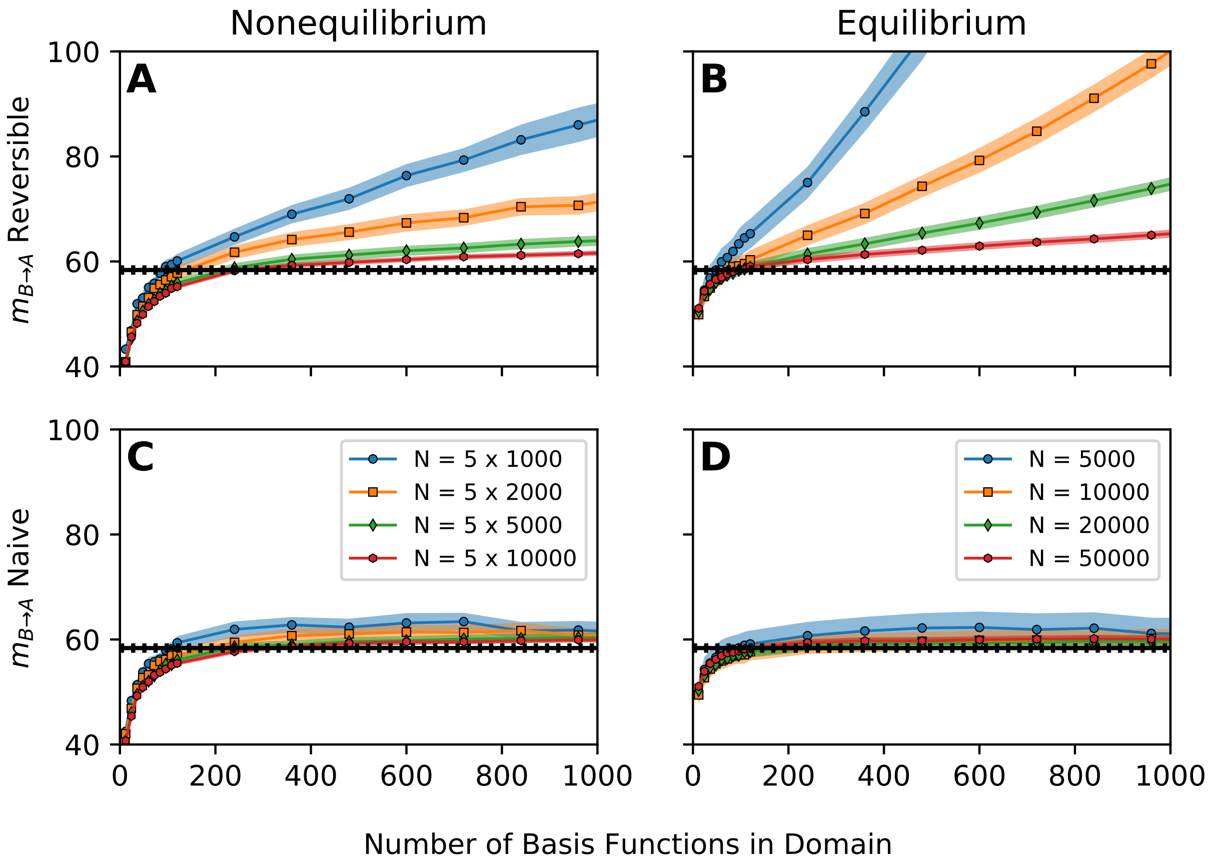

FIG. S2. Effect of enforcing MSM reversibility on the estimated mean first-passage time from state B to state A on the scaled Müller-Brown potential. Estimates in the top row are constructed using the reversible MSM estimator and estimates in the bottom row are not. The columns correspond to two different datasets: the left column shows estimates constructed from the nonequilibrium dataset detailed in section V, and the right column shows estimates constructed from a long equilibrium trajectory. Different curves correspond to MSMs constructed from datasets of different sizes.

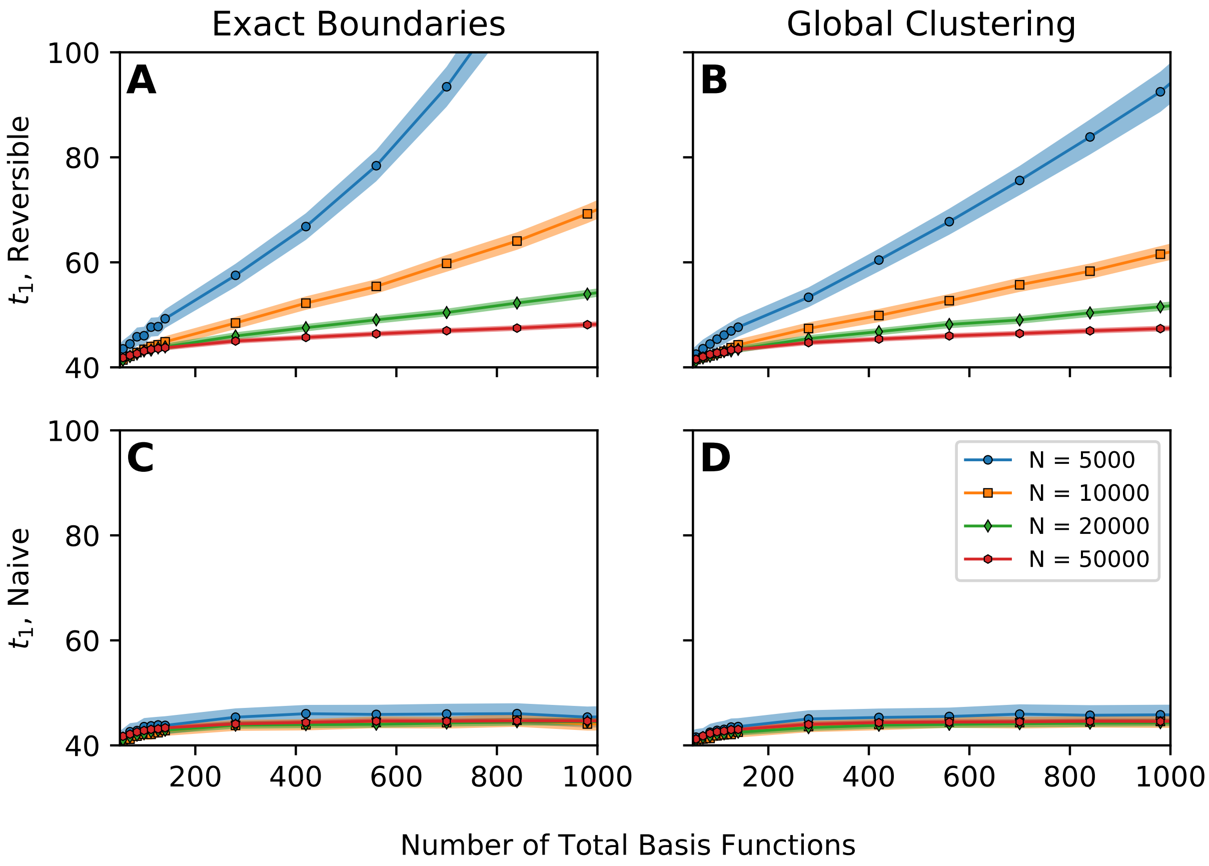

FIG. S3. Dominant implied timescale for MSMs constructed on a long equilibrium trajectory on the scaled Müller-Brown potential. Estimates in the top row are constructed using the reversible MSM estimator and estimates in the bottom row use the naive estimator. Columns correspond two different clustering schemes. The left column gives estimates constructed using the clustering described in Section V A, and the right column gives estimates obtained by clustering the data without regard for the boundary conditions (i.e., globally). Different curves correspond to MSMs constructed on different size datasets.

reference 9 gave the same results as PyEMMA. Rather, we see that the bias decays with increasing dataset sizes, suggesting that it is statistical in nature.

In panels C and D, we show estimates constructed without enforcing reversibility, which we term the naive estimator. The naive estimator does not have the same bias. This suggests that the maximum likelihood iteration introduces a large, slowly decaying statistical error. To ensure that this is not an artifact of the clustering procedure, we also constructed MSMs by applying k-means globally to the data, without regard to boundary conditions. We then estimated the dominant implied timescale for both clustering schemes, which we plot in Figure S3. We see the same trends as in the mean first-passage time: for reversible MSMs, the implied timescale grows unboundedly with the number of basis functions for both clustering methods. In contrast, both clustering methods converge equally well when using the naive estimator.

S7. SUPPLEMENTARY PLOTS FOR DELAY EMBEDDING THE MÜLLER-BROWN MODEL

In Figure S4, we give implied timescales for the MSMs constructed in Section VI. To test the effect of trajectory length on the one-dimensional, delay-embedded data, we repeated the calculation for three additional datasets. The total number of points in each dataset is fixed, but each nonequilibrium trajectory is of different length. We plot the resulting curves in Figure S5. In all cases, we see an anomalous behavior when the delay length or lag time approaches the total length of the trajectory.

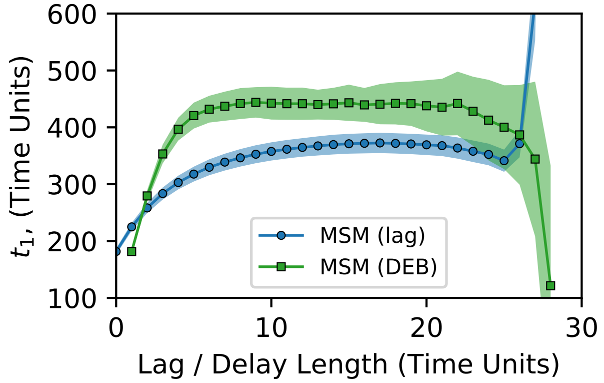

FIG. S4. Implied timescales for the delay-embedded MSM and lagged MSMs in Section VI.

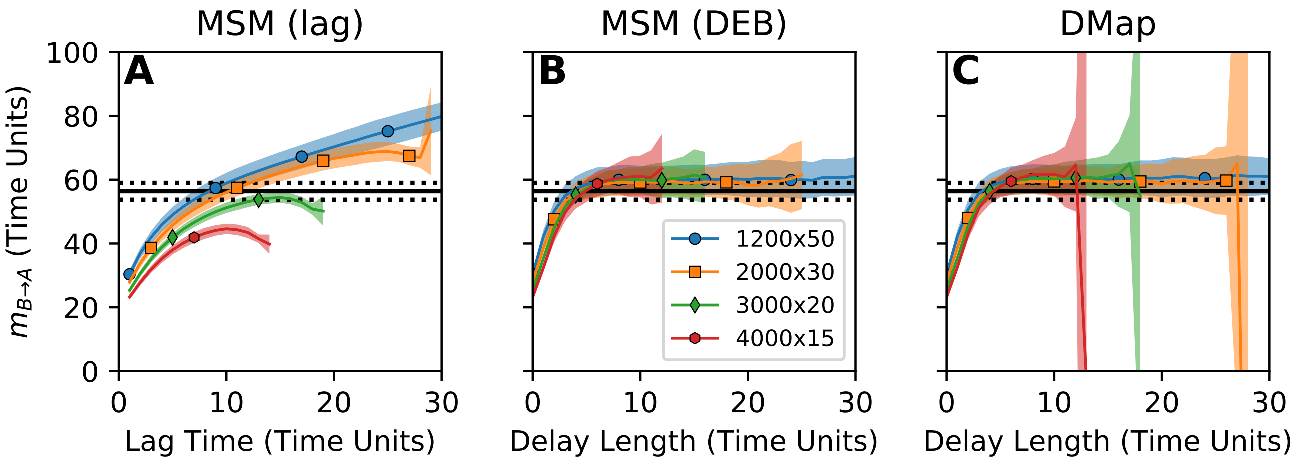

FIG. S5. Comparison of methods for controlling the projection error in an incomplete CV space. Plots are as in Figure 4, with the addition of three new datasets. The curves correspond to datasets consisting of 1200 trajectories, each 50 time units long (blue circles), 200 points, each 30 time units long (orange squares, the same data as pictured in Figure 4), 3000 trajectories, each 20 units long (green diamonds), and 4000 trajectories, each 15 units long (red hexagons).

[1] T. Berry, D. Giannakis, and J. Harlim, Physical Review E **91**, 032915 (2015).

[2] T. Berry and J. Harlim, Applied and Computational Harmonic Analysis **40**, 68 (2016).

[3] C. K. Williams and M. Seeger, in *Advances in Neural Information Processing Systems* (2001) pp. 682–688.

[4] A. W. Long and A. L. Ferguson, Applied and Computational Harmonic Analysis (2017).

[5] I. Bronshtein, K. Semendyayev, G. Musiol, and H. Muehlig, *Handbook of Mathematics*, 3rd ed. (Springer, 2007).

[6] E. Vanden-Eijnden, in *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1* (Springer, 2006) pp. 453–493.

[7] P. Metzner, C. Schütte, and E. Vanden-Eijnden, Multiscale Modeling & Simulation **7**, 1192 (2009).

[8] W. E and E. Vanden-Eijnden, Annual Review of Physical Chemistry **61**, 391 (2010).

[9] G. R. Bowman, K. A. Beauchamp, G. Boxer, and V. S. Pande, The Journal of Chemical Physics **131**, 124101 (2009).