Computing Committors in Collective Variables Using Mahalanobis Diffusion Maps

Luke Evans

Joint work with:

Dr. Maria Cameron Dr. Pratyush Tiwary

Applied Mathematics & Statistics, and Scientific Computation
University of Maryland, College Park

Funding:

AFOSR MURI grant FA9550-20-1-0397 (MC)

NSF CAREER grant CHE-2044165 (PT)

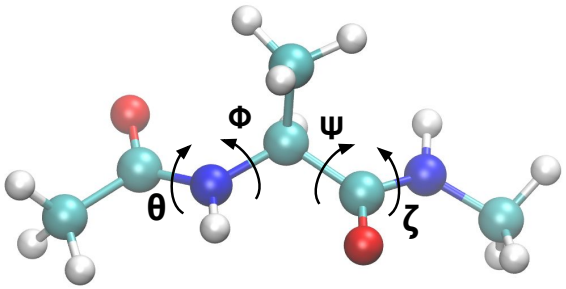
Ann G. Wylie Dissertation Fellowship (LE)

Summer 2023: Joining the Flatiron Institute as Postdoc, Center for Computational Mathematics, Structural and Molecular Biophysics Group (Pilar Cossio and Sonya Hanson)

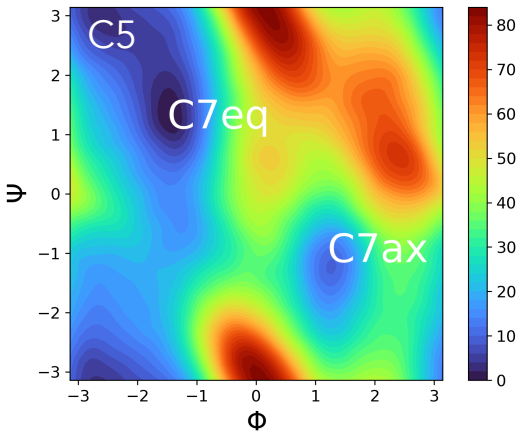
Model Reduction: Rare Transitions in Molecular Dynamics

Main example: Conformal changes in molecules

Phenomena: Long residence times in stable states, transitions between stable states are very quick



Free Energy $F(x) = \beta^{-1} \log p(x)$



All-atom simulation: Langevin Dynamics

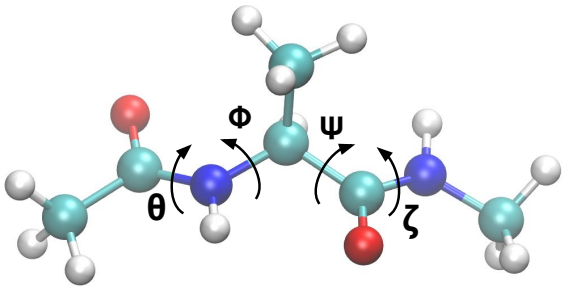
$$dy_t = m^{-1} y_t dt$$

$$dp_t = (-\nabla V(y_t) - \gamma p_t)dt + \sqrt{2\gamma m \beta^{-1}} dw_t$$

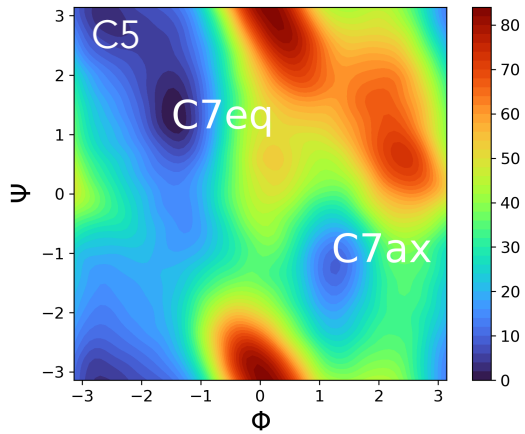
Model Reduction: Rare Transitions in Molecular Dynamics

Main example: Conformal changes in molecules

Phenomena: Long residence times in stable states, transitions between stable states are very quick



Free Energy $F(x) = \beta^{-1} \log p(x)$



All-atom simulation: Langevin Dynamics

$$dy_t = m^{-1} y_t dt$$

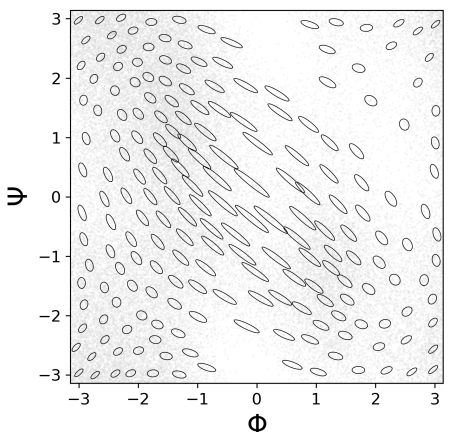
$$dp_t = (-\nabla V(y_t) - \gamma p_t)dt + \sqrt{2\gamma m \beta^{-1}} dw_t$$

Diffusion matrix $M(x)$

$$M_{ij} \approx \frac{1}{N} \sum_{k=1}^{N} \left[\sum_{\ell=1}^{n} \frac{\partial x_i(t_k))}{\partial y_\ell} \frac{\partial x_j(t_k))}{\partial y_\ell} \right]$$

Collective variable dynamics*:

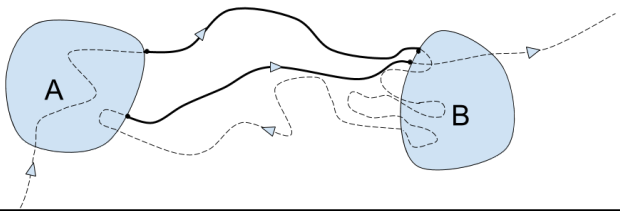
$$dx_t = [-M(x_t)\nabla F(x_t) + \beta^{-1} \nabla \cdot M(x_t)]dt + \sqrt{2\beta^{-1}} M(x_t)^{1/2} dw_t$$

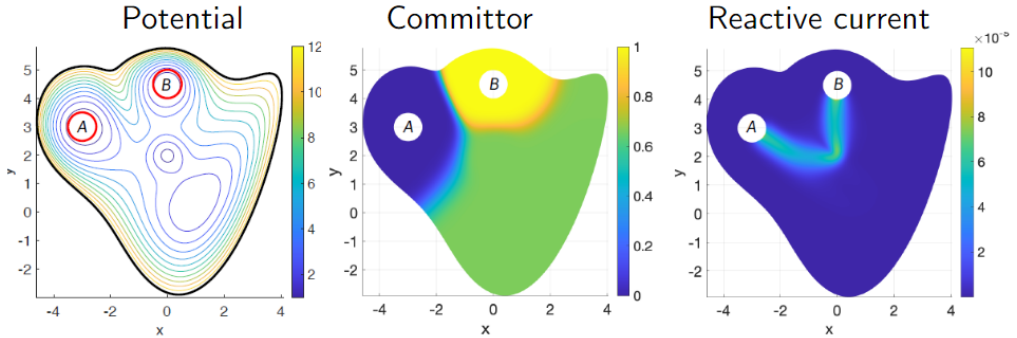


Maragliano, Fischer, Vanden-Eijnden, Cicotti *J. Chem. Phys* (2006).
Carter, Ciccotti, Hynes, Kapral, *Chem. Phys. Letters* (1987)
Legoll, Leliévre, *Nonlinearity* (2010)

Framework: Transition Path Theory

Weinan E, Eric Vanden-Eijnden, *J. Stat. Phys. 2006*





Subject: reactive trajectories

Key function: committor

$$q(x) = \mathbb{P}(\tau_B < \tau_A | x_0 = x)$$

Elliptic BVP for committor

$$\begin{aligned} \mathcal{L}q(x) &= 0 \qquad x \in (A \cup B)^c \\ q(\partial A) &= 0 \quad q(\partial B) = 1 \end{aligned}$$

$\mathcal{L}$: generator for the SDE

Reactive current

Transition rate

Collective Variable dynamics*

$$dx_t = [-M\nabla F + \beta^{-1}\nabla \cdot M]dt + \sqrt{2\beta^{-1}M}dw_t$$

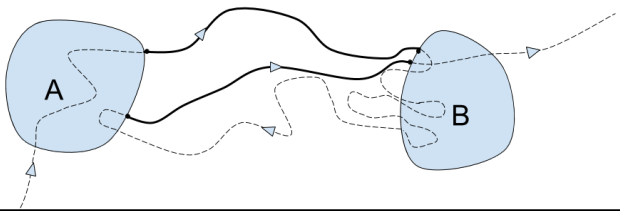
$$\mathcal{L} = \beta^{-1}e^{\beta F}\nabla \cdot (e^{-\beta F}M\nabla)$$

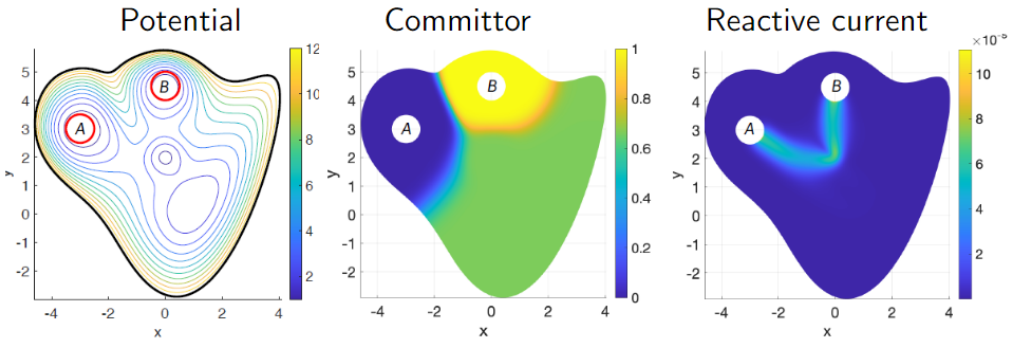
Key Challenge: Solve Committor BVP

Then, extract reaction rate and reactive current.

Framework: Transition Path Theory

Weinan E, Eric Vanden-Eijnden, *J. Stat. Phys. 2006*





Subject: reactive trajectories

Key function: committor

$$q(x) = \mathbb{P}(\tau_B < \tau_A | x_0 = x)$$

Elliptic BVP for committor

$$\begin{aligned} \mathcal{L}q(x) &= 0 \qquad x \in (A \cup B)^c \\ q(\partial A) &= 0 \quad q(\partial B) = 1 \end{aligned}$$

$\mathcal{L}$: generator for the SDE

Reactive current

$$\mathcal{J}(x) \propto e^{-\beta F(x)} M(x) \nabla q(x)$$

Transition rate

$$\nu_{AB} \propto \int_{\Omega_{AB}} e^{-\beta F} \nabla q^\top M \nabla q dx$$

Collective Variable dynamics*

$$dx_t = [-M\nabla F + \beta^{-1} \nabla \cdot M]dt + \sqrt{2\beta^{-1} M} dw_t$$

$$\mathcal{L} = \beta^{-1} e^{\beta F} \nabla \cdot (e^{-\beta F} M \nabla)$$

$$= \beta^{-1} \operatorname{tr}(M \nabla \nabla) + (-M\nabla F + \beta^{-1} \nabla \cdot M) \cdot \nabla$$

Key Challenge: Solve Committor BVP

Then, extract reaction rate and reactive current.

Problem Space: High-Dimensional PDE Solvers

Our Contribution: Committor solver based on Mahalanobis Diffusion Maps (for moderately sized dimensions)

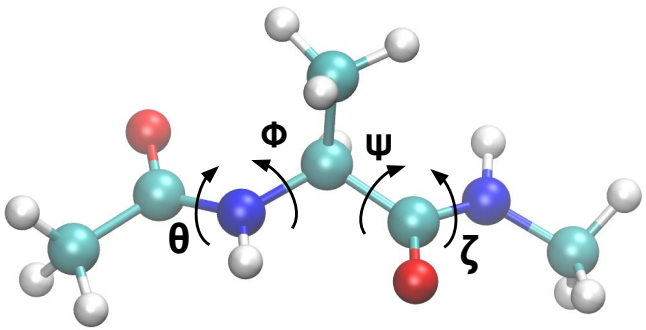
Solving For Committor in High Dimensions:

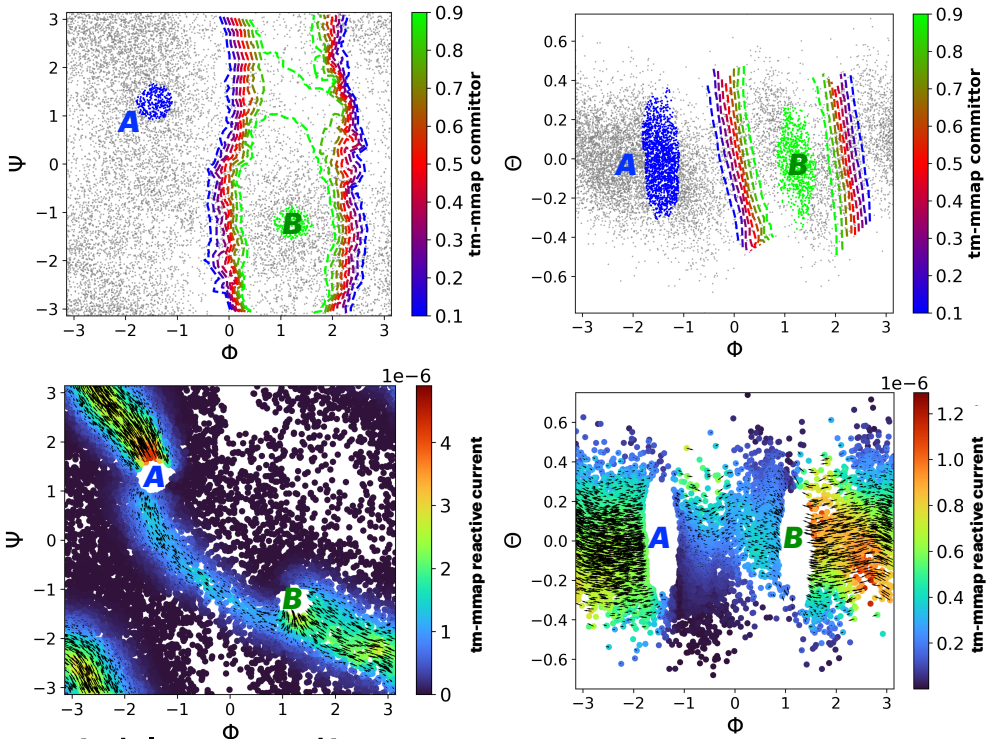
- Neural Network: (Li, Lin, Ren 2019), (Khoo, Lu, Ying 2019), (Rotskoff, Mitchell, Vanden-Eijnden 2021), (Li, Khoo, Ren, Ying 2021)
- Tensor Trains: (Chen, Khoo, Lindsey 2021)
- Diffusion Maps: (Trstanova, Leimkuhler, Leliévre 2020)

High Dimensional PDE Solvers:

- Neural Network: W.E, G. Karniadakis, Y. Khoo, W. Ren, E. Vanden-Eijnden, H. Yang, ...
- Tensor Trains: S.Dolgov, A. Gorodetsky, J. Hoskins, Y. Khoo, M. Lindsey, ...
- Diffusion Maps: H. Antil, T. Berry, D. Giannakis, J. Harlim, ...

Solve Committor BVP On a Point Cloud





Given data $\{x_i\}_{i=1}^N$ in collective variable coordinates,

find $N \times N$ matrix L so that $\sum_{i=1}^N L_{ij} f(x_j) \approx (\mathcal{L}f)(x_i)$.

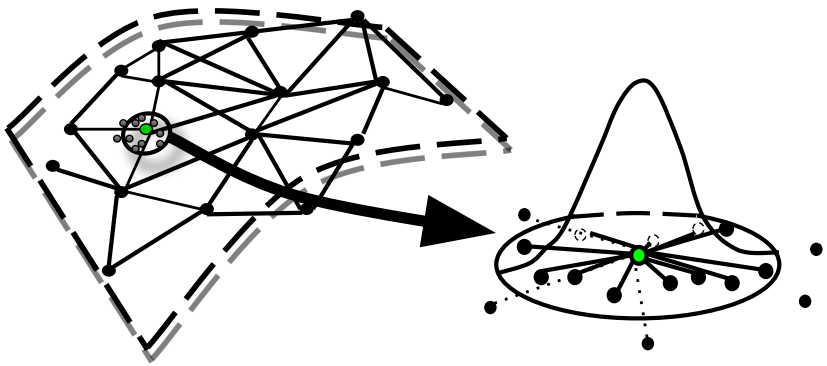
Solve:

$$\begin{cases} [Lq]_i = 0 & i \in \mathcal{I}(\Omega \backslash A \cup B) \\ [q]_i = 0 & i \in \mathcal{I}(A) \\ [q]_i = 1 & i \in \mathcal{I}(B) \end{cases}$$

Challenge: Approximate $\mathcal{L} = \beta^{-1} e^{\beta F(x)} \nabla \cdot (e^{-\beta F(x)} M(x) \nabla)$

Diffusion Maps

Laplacian Eigenmaps (Belkin, Niyogi, *Neural Comp.* 2003)
Diffusion Maps (Coifman, Lafon, Lee, Maggioni, Nadler, Warner, Zucker *PNAS* 2005)



Idea: Analyze a data set by constructing a Markov chain P on it

Similarity function: Gaussian kernel

$$k_\epsilon(\mathbf{x_i}, x_j) = \exp\left(\frac{-||\mathbf{x_i} - x_j||^2}{2\epsilon}\right)$$

- P_{ij} proportional to $k_\epsilon(x_i, x_j)$

Theory: *discrete* generator $L = \epsilon^{-1}(P - I)$ approximates *continuous* generator $\mathcal{L}$

Prototypical Diffusion Map Convergence Theorems

As $N \to \infty$,

$$\sum_{j=1}^{N} (L_{\epsilon,\alpha})_{ij} f(x_j) = \mathcal{L}_\alpha f(x_i) + \mathcal{O}(\epsilon)$$

Diffusion Maps: Building Blocks

Heat kernel $k_\epsilon(x, x') := e^{-\frac{||x-x'||^2}{2\epsilon}}$ for small ϵ (short "time"):

$$\boxed{\frac{1}{(2\pi\epsilon)^{d/2}} \int_{\mathbb{R}^d} e^{-\frac{||x-x'||^2}{2\epsilon}} f(x')dx' = f(x) + \frac{\epsilon}{2}\textcircled{\Delta f(x)} + O(\epsilon^2)}$$

For N i.i.d samples $\{x_i\}_{i=1}^N$, sampling density $\rho(x)$:

$$\lim_{N\to\infty} \frac{1}{N} \sum_j e^{-\frac{||x-x_j||^2}{2\epsilon}} f(x_j) = \int_{\mathbb{R}^d} e^{-\frac{||x-x'||^2}{2\epsilon}} f(x')\rho(x')dx'$$

Diffusion Maps: Building Blocks

$$\lim_{N\to\infty} \frac{\sum_j e^{-\frac{||x-x_j||^2}{2\epsilon}} f(x_j)}{\sum_j e^{-\frac{||x-x_j||^2}{2\epsilon}}} = \frac{\int_{\mathbb{R}^d} e^{-\frac{||x-x'||^2}{2\epsilon}} f(x')\rho(x')dx'}{\int_{\mathbb{R}^d} e^{-\frac{||x-x'||^2}{2\epsilon}} \rho(x')dx'}$$

$$= \frac{(f\rho)(x) + \frac{\epsilon}{2}\Delta(f\rho)(x) + O(\epsilon^2)}{\rho(x) + \frac{\epsilon}{2}\Delta\rho(x) + O(\epsilon^2)}$$

$$= f(x) + \frac{\epsilon}{2}\left([\Delta f(x) + \nabla f(x)\cdot\nabla \log \rho^{\mathbf{2}}(\mathbf{x_i})]\right) + O(\epsilon^2).$$

$$\text{Markov Matrix } [P_\epsilon]_{ij} := \frac{e^{-\frac{||x_i-x_j||^2}{2\epsilon}}}{\sum_\ell e^{-\frac{||x_i-x_\ell||^2}{2\epsilon}}}$$

$$[L_\epsilon f]_i := (2/\epsilon)([P_\epsilon f]_i - f_i) \approx \left(\Delta f(x_i) + \nabla f(x_i)\cdot\nabla \log \rho^{\mathbf{2}}(\mathbf{x_i})\right)$$

Diffusion Maps: Framework

α : renormalization parameter

$\rho(x)$: sampling density

$\{x_i\}_{i=1}^N$ samples from $\rho(x)$

1. Kernel:
   $k_\epsilon(x, x') = \exp(-\frac{||x-x'||^2}{2\epsilon})$ $\quad [K_\epsilon]_{i,j} = k_\epsilon(x_i, x_j)$

2. Right normalization:
   $\rho_\epsilon(x) := \int k_\epsilon(x, x')\rho(x')dx'$ $\quad [\rho_\epsilon]_i := \sum_j [K_\epsilon]_{ij}$
   $k_{\epsilon,\alpha}(x, x') := k_\epsilon(x, x')\rho_\epsilon^{-\alpha}(x')$ $\quad K_{\epsilon,\alpha} := K_\epsilon \text{diag}(\rho_\epsilon)^{-\alpha}$

3. Left normalization:
   $\rho_{\epsilon,\alpha}(x) := \int k_{\epsilon,\alpha}(x, x')\rho(x')dx'$ $\quad [\rho_{\epsilon,\alpha}]_i := \sum_j [K_{\epsilon,\alpha}]_{ij}$
   $\mathcal{P}_{\epsilon,\alpha} f(x) := \frac{\int k_{\epsilon,\alpha}(x,x')f(x')\rho(x')dx'}{\rho_{\epsilon,\alpha}(x')}$ $\quad P_{\epsilon,\alpha} := \text{diag}(\rho_{\epsilon,\alpha})^{-1} K_{\epsilon,\alpha}$

4. Construct generator:
   $\mathcal{L}_{\epsilon,\alpha} f = \frac{\mathcal{P}_{\epsilon,\alpha} f - f}{\epsilon}$ $\quad L_{\epsilon,\alpha} = \frac{P_{\epsilon,\alpha} - I}{\epsilon}$

Limiting Operator, $N \to \infty$, $\epsilon \to 0$

$$\boxed{\mathcal{L}f = \Delta f + \nabla f \cdot \nabla \log \rho^{2-2\alpha}}$$

- $\alpha = 0$: normalized graph Laplacian
  $\mathcal{L}f = \Delta f + \nabla f \cdot \nabla \log \rho^2$
- $\alpha = \frac{1}{2}$: backward Kolmogorov operator
  $\mathcal{L}f = \Delta f + \nabla f \cdot \nabla \log \rho \quad = \quad \boxed{\rho^{-1} \nabla \cdot (\rho \nabla f)}$
- $\alpha = 1$: Laplace-Beltrami operator
  $\mathcal{L}f = \Delta f$

Typical use: Solve an eigenvalue problem with $\mathcal{L}$, use top eigenfunctions as coordinates (the "diffusion map").

Usage in molecular dynamics: for Gibbs distribution $\rho \propto e^{-\beta V}$ and $\alpha = \frac{1}{2}$, $\beta^{-1}\mathcal{L}$ is the generator of the overdamped Langevin dynamics.

Coifman, Kevrekidis, Lafon, Maggioni, Nadler, *Multiscale Modeling & Sim.* (2008).

Extending to Collective Variables: mmap

Mahalanobis Kernel: (Singer, Coifman 2008), "mmap"

$$\exp\left(\frac{-(x_i - x_j)^\top (M^{-1}(x_i) + M^{-1}(x_j))(x_i - x_j)}{4\epsilon}\right) \not\approx \exp\left(\frac{-||z_i - z_j||^2}{2\epsilon}\right).$$

Our Case: No diffeomorphism, our $M \neq \mathrm{J}\mathrm{J}^\top$.
Is the Mahalanobis kernel still useful?

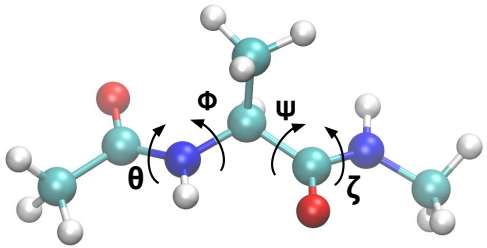
Yes, for a user-input $M(x)$

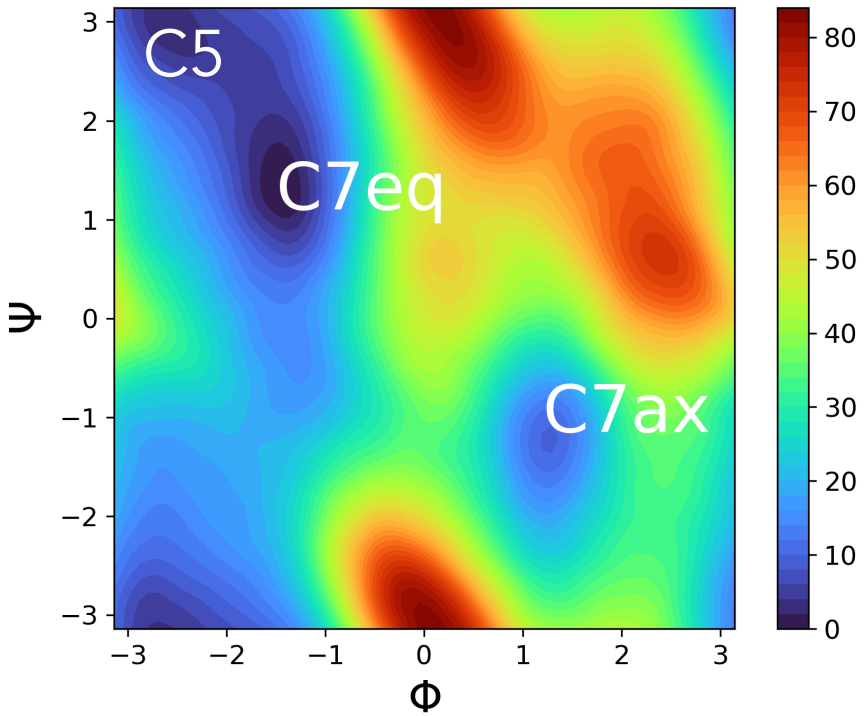
$$\mathcal{L}_{\epsilon,1/2} f \longrightarrow \rho(x)^{-1} \nabla \cdot \left(\rho(x) \boxed{M(x)} \nabla f(x)\right)$$

Evans, Cameron, Tiwary, *Applied and Comp. Harmonic Analysis* (2023).

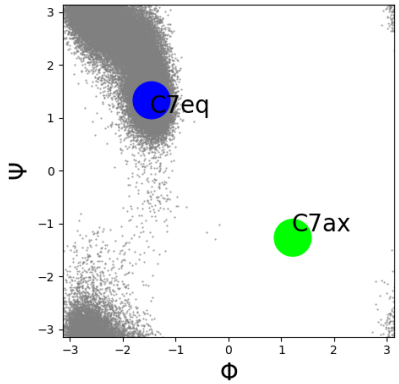
Related Work:
Berry, Sauer, *Applied and Comp. Harmonic Analysis* (2016).
Banisch, Trstanova, Bittracher, Klus, Koltai *Applied and Comp. Harmonic Analysis* (2020).

Extending to Enhanced Sampling: tm-mmap

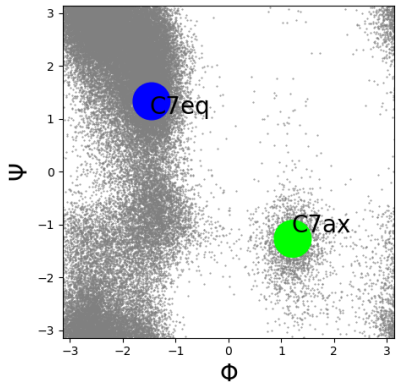




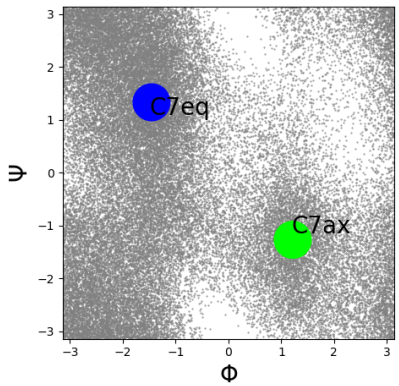
10ns trajectory, 300K



10ns trajectory, 500K



10ns trajectory, Metadynamics



Laio, Parrinello *PNAS* (2002).
Barducci, Bussi, Parrinello *Phys. Review Letters* (2008).

tm-mmap: Target Measure Mahalanobis Diffusion Maps

$$k_\epsilon(x, x') = \exp\left(-\frac{(x - x')^\top \left[M^{-1}(x) + M^{-1}(x')\right](x - x')}{4\epsilon}\right)$$

Right Normalization:

$$k_\epsilon(x, x') \to k_\epsilon(x, x')\frac{\left(\det M(x')^{-1/2}\mu(x')\right)^{1/2}}{\rho_\epsilon(x')}$$

Theorem(s) (Evans, Cameron, Tiwary)

$$\lim_{\epsilon\to 0} \mathcal{L}_{\epsilon,\mu} f(x) = \frac{\beta}{2}\mathcal{L} f(x) \quad \forall x \in \mathcal{M}, \qquad (\texttt{tm} - \texttt{mmap})$$

$$\lim_{\epsilon\to 0} \mathcal{L}_{\epsilon,\alpha} f(x) = \frac{\beta}{2}\mathcal{L}_\alpha f(x) \quad \forall x \in \mathcal{M}, \qquad (\texttt{mmap})$$

$$\mathcal{L} f = \beta^{-1}\mu^{-1}(x)\nabla \cdot (\mu(x)M(x)\nabla f(x))$$

tm-mmap: Evans, Cameron, Tiwary, *J. Chem. Phys.* (2022).

Related Work:

tm-dmap: Banisch, Trstanova et al., *Applied and Comp. Harmonic Analysis (2020).*

Trstanova, Leimkuhler, Leliévre, *Proc. Royal Soc. A*, (2020)

tm-mmap for transition path theory: Recap

☑ **Discrete Operator**:

Find $N \times N$ matrix L so that $\sum_{i=1}^{N} L_{ij} f(x_j) \approx (\mathcal{L}f)(x_i)$.

$$\mathcal{L}f = \beta^{-1} e^{\beta F(x)} \nabla \cdot (e^{-\beta F(x)} M(x) \nabla f(x))$$

After constructing discrete operator, we need to find:

1. ☑ **Committor**: Solve
$$\begin{cases} [L_\epsilon q]_i = 0 & i \in \mathcal{I}(\Omega \backslash (A \cup B)) \\ [q]_i = 0 & i \in \mathcal{I}(A) \\ [q]_i = 1 & i \in \mathcal{I}(B) \end{cases}$$
2. ✓**What if F(x)(or target measure) is unknown?**
3. ✓**Reactive Current**? $\mathcal{J}(x) = \beta^{-1} Z^{-1} e^{-\beta F(x)} M(x) \nabla q(x)$

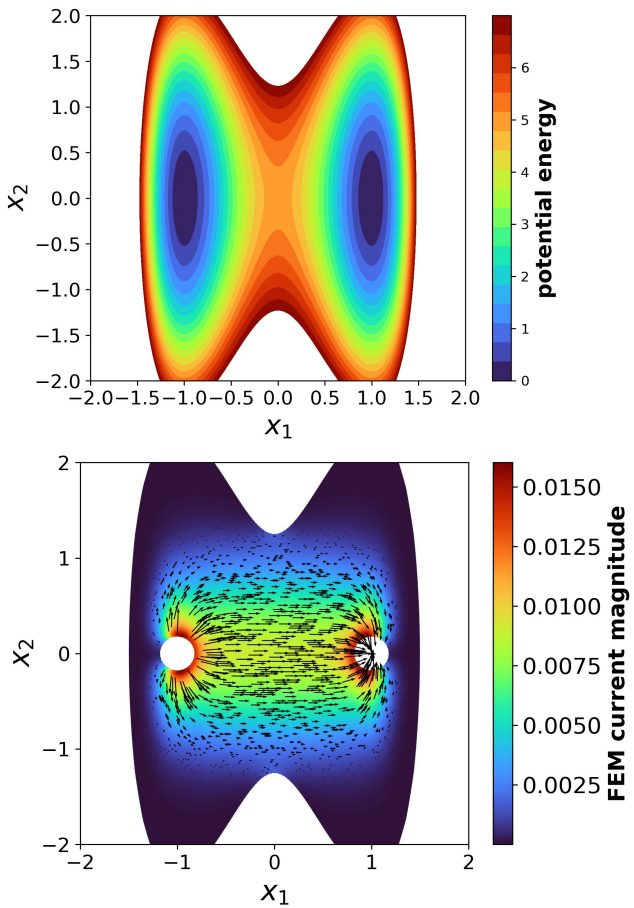
tm-mmap: Evans, Cameron, Tiwary, *J. Chem. Phys.* (2022).

2. Related: Rydzewski, Chen, Ghosh, Valsson, *J. Chem. Theory Comput.* (2022)

tm-mmap: 2D Toy Example With Saddle Avoidance

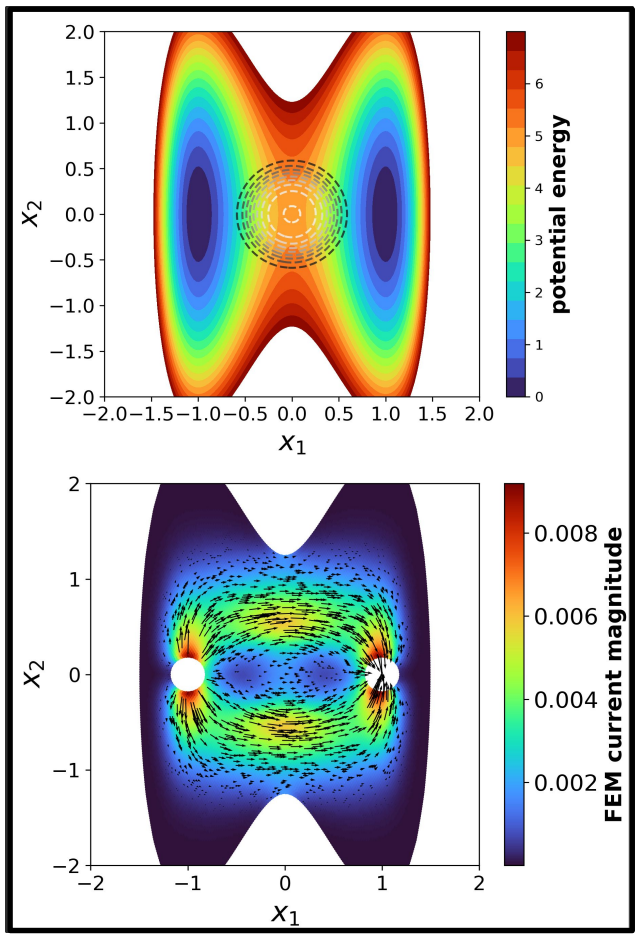
Two-Well Potential

$V(x)=5(x_1^2-1)^2+10\alpha x_2^2$



With Pos-Dep. Diffusion

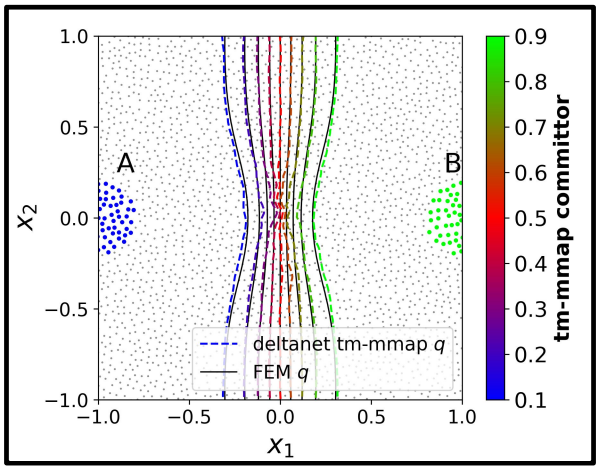
$M(x)=(1+8\exp(-||x||^2/2\sigma^2))^{-1}I_{2\times 2}$



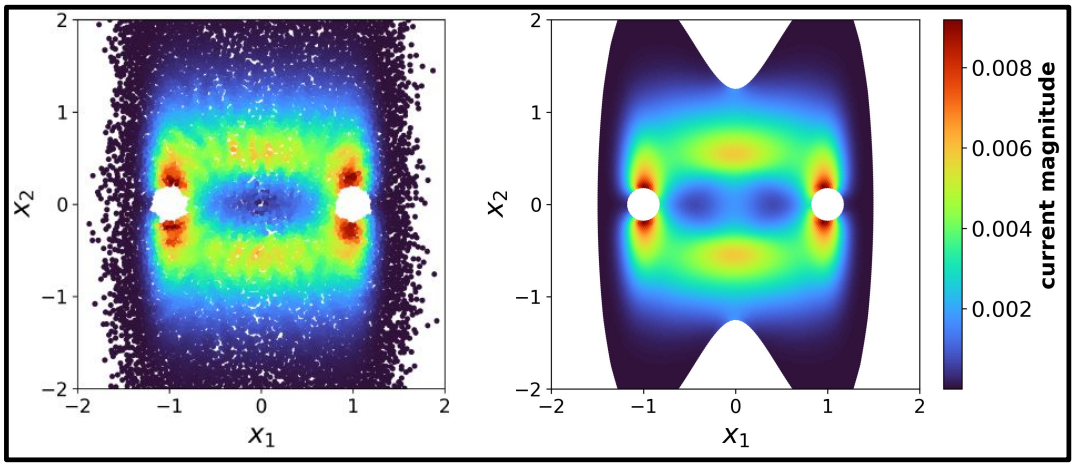
Moro, Cardin. "Saddle point avoidance due to inhomogeneous friction". Chemical Physics. 1997

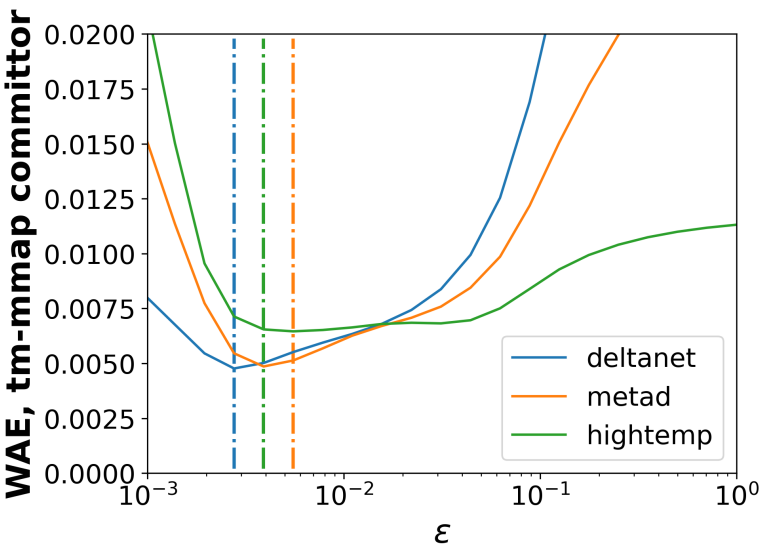
Errors and Rates: 2D Toy Example

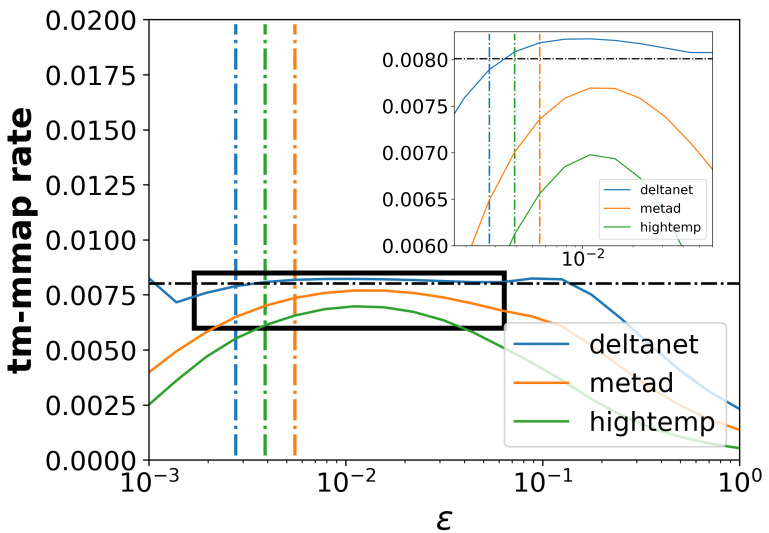
Committor



Current



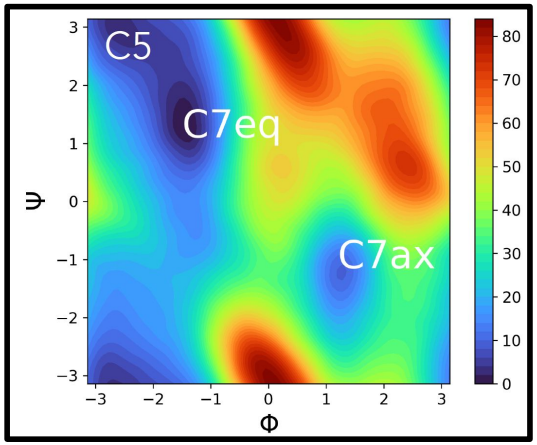




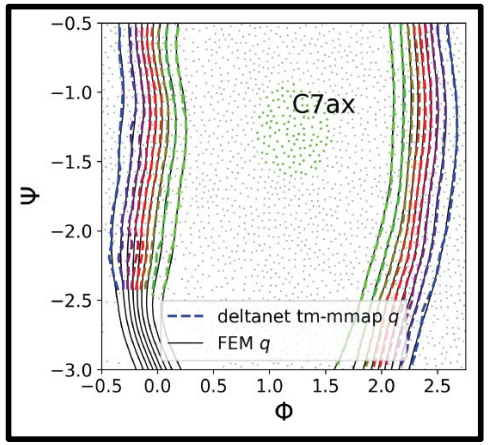
Error: $\sum_i |q(x_i) - q_{FEM}(x_i)|\nu_i$ Rate: $\frac{\beta^{-1}}{NZ_\epsilon}\sum_{ij}\frac{\mu_i}{[p_\epsilon]_j}L_{ij}(q_j - q_i)^2$

Errors and Rates: Alanine Dipeptide, 2 dihedrals

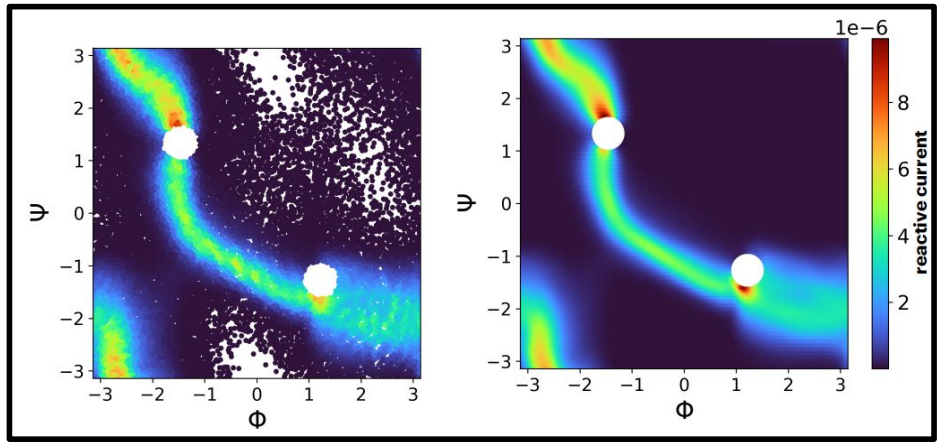
Free Energy

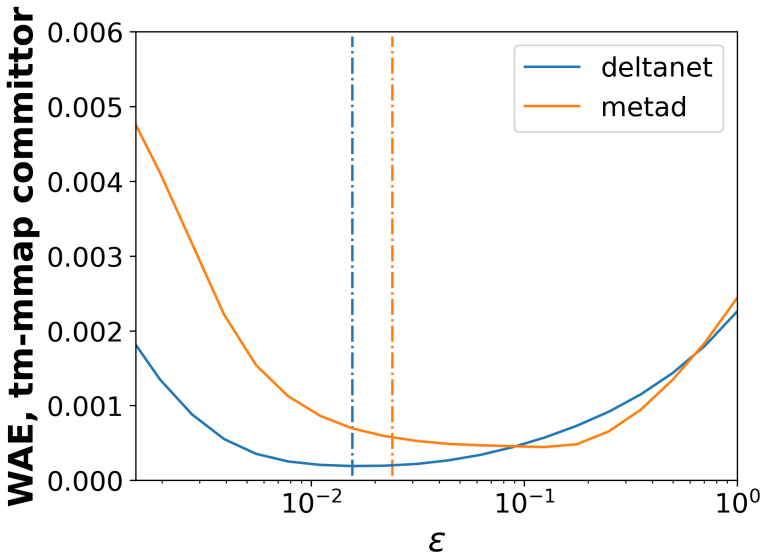


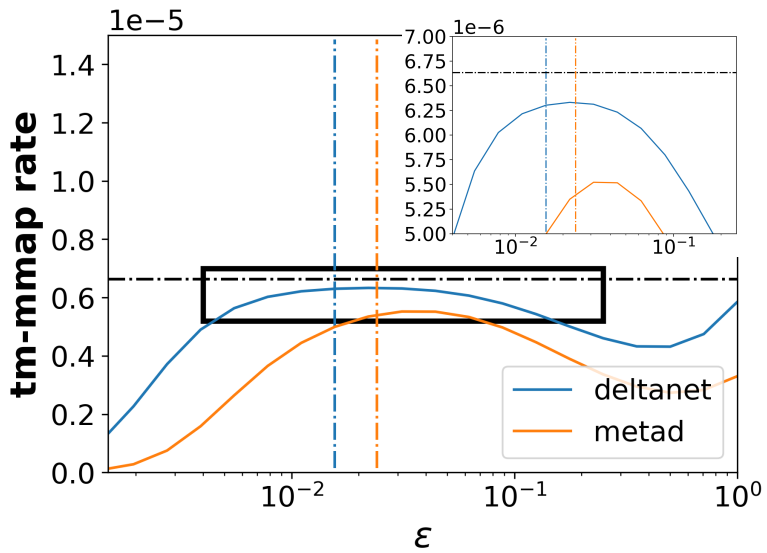
Committor



Current

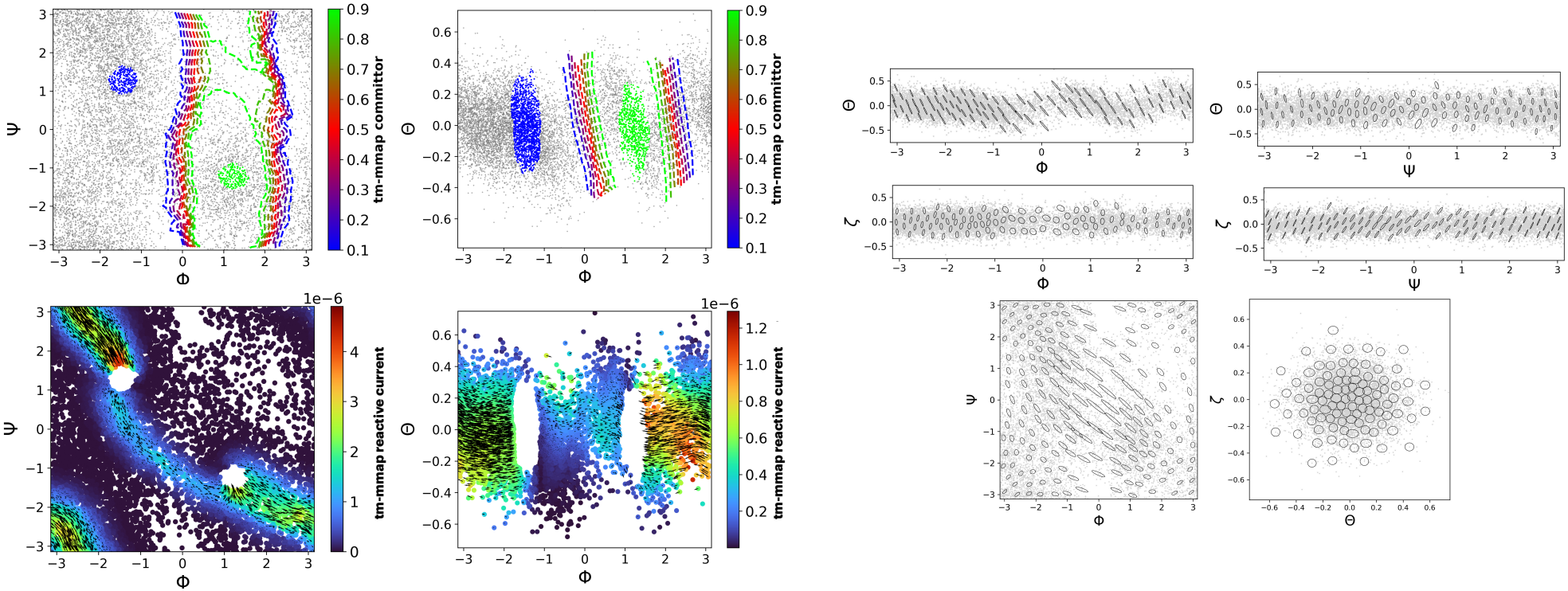






Error: $\sum_i |q(x_i) - q_{FEM}(x_i)|\nu_i$ Rate: $\frac{\beta^{-1}}{NZ_\epsilon} \sum_{ij} \frac{\mu_i}{[p_\epsilon]_j} L_{ij}(q_j - q_i)^2$

Alanine Dipeptide, 4 dihedrals



biased traj. $x_t = (\Phi_t, \Psi_t, \Theta_t, \zeta_t)$, diffusion matrices $M(x_t) \in \mathbb{R}^{4\times 4}$

ν_{AB}	method	dataset
$6.3 \cdot 10^{-6}$ ps^{-1}	`tm-mmap`	2 dihedral (ϕ, ψ)
$2.0 \cdot 10^{-6}$ ps^{-1}	`tm-mmap`	4 dihedral $(\phi, \psi, \theta, \zeta)$
$1.4 \cdot 10^{-6}$ ps^{-1}	`direct`	2.4μs trajectory*

*Vani, Weare, Dinner, *J. Chem. Phys. (2022)*

Extensions and Future Directions

- Current work: `tm-dmap` **error analysis** - see poster of Shashank Sule
- Assessment of **model reduction error** in molecular dynamics:
  - physical & machine-learned collective variables, **autoencoders**
  - spectral/rate criteria (Zhang, Hartmann, Schütte 2016)
- Overdamped Langevin **regularization for autoencoder latent space** (with *Dedi Wang*, Yihang Wang, Pratyush Tiwary)

Future Considerations:

- Higher Dimensions: Free Energy, Diffusion Tensor computation
- Reactive currents and related - using available vector fields
- Software module in Plumed (enhanced sampling software) for `tm-mmap`

Evans, Cameron, Tiwary, *Journal of Chemical Physics*. (2022).
Evans, Cameron, Tiwary, *Applied and Computational Harmonic Analysis* (2023).

Appendix

Algorithm 1: Target Measure Mahalanobis Diffusion Map (`tm-mmap`)

Input: data $X = \{x_i\}_{i=1}^N$, diffusion matrices $\{M(x_i)\}_{i=1}^N$
bandwidth ϵ, target measure μ,

Output: Generator matrix $L_{\epsilon,\mu}$

Construct Mahalanobis kernel, estimate sampling density

1 $[K_\epsilon]_{i,j} = k_\epsilon(x_i, x_j), i, j = 1, \ldots, N$

2 $c_i = (2\pi\epsilon)^{d/2}|M_i|^{1/2}, \; i = 1, \ldots, N$

3 $[\rho_\epsilon]_i = \frac{1}{Nc_i}\sum_j [K_\epsilon]_{ij}, i = 1, \ldots, N$

Right normalize the kernel

4 $$[K_{\epsilon,\mu}]_{ij} := \frac{[K_\epsilon]_{ij}(\mu_j|M_j|^{-1/2})^{1/2}}{[\rho_\epsilon]_j}, \; i, j = 1, \ldots, N$$

Left normalize the kernel

5 $$[P_{\epsilon,\mu}]_{ij} := \frac{[K_{\epsilon,\mu}]_{ij}}{\sum_\ell [K_{\epsilon,\mu}]_{i\ell}}, \; i, j = 1, \ldots, N$$

Construct generator

6 $$[L_{\epsilon,\mu}]_{ij} = \frac{[P_{\epsilon,\mu}]_{ij} - \delta_{ij}}{\epsilon}, i, j = 1, \ldots, N$$

Reweighting in Higher Dimensions

Suppose that an enhanced sampling algorithm samples from the Gibbs density

$$\rho(x) \propto e^{-\beta(F(x)+U(x))},$$

where U is a *known* bias potential, while $F(x)$ is the *unknown* free energy. The desired target measure is $\mu(x) = \exp(-\beta F(x))$. We approximate the sampling density $\rho(x)$ by $\rho_\epsilon(x)$ and obtain the following estimate for the target measure:

$$\boxed{\mu(x) \approx \rho_\epsilon(x) \exp(\beta U(x)).}$$

We approximate the normalizing constant Z as

$$Z_\epsilon := \frac{1}{N} \sum_{i=1}^{N} \frac{\mu(x_i)}{[p_\epsilon]_i}$$

as for sufficiently large N and small ϵ we have

$$Z_\epsilon \approx \int_{\mathbb{R}^d} \frac{\mu(x)}{\rho(x)} \rho(x) dx = Z.$$

Getting Gradients from a Generator Matrix

Observation: $\boxed{\Delta(fg) = f\Delta g + g\Delta f + 2\nabla f^\top \nabla g}$

$$\mathcal{L} = b\cdot\nabla + \beta^{-1}\,\mathrm{tr}(M\nabla\nabla) \quad | \quad dx_t = b(x)dt + \sqrt{2\beta^{-1}}M^{1/2}(x)dw_t$$

$$\boxed{\mathcal{L}(fg) = f\mathcal{L}g + g\mathcal{L}f + 2\beta^{-1}\nabla f^\top M\nabla g}$$

Deviation from "product rule":

$$\nabla f^\top M\nabla g = (\beta/2)\left[\mathcal{L}(fg) - f\mathcal{L}g - g\mathcal{L}f\right]$$

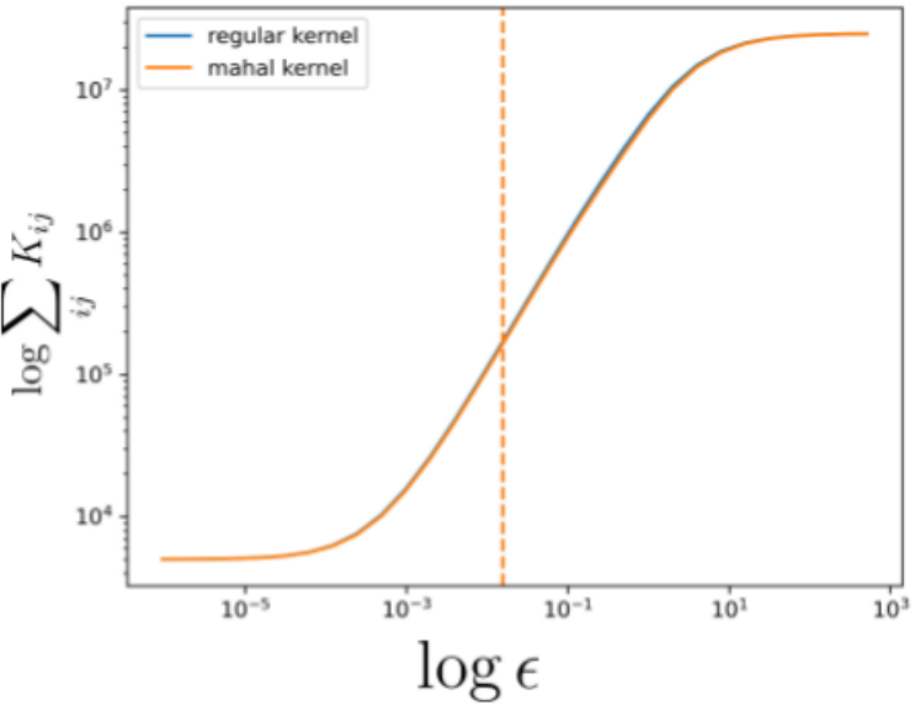
Discrete formulation: since $\sum_j L_{ij} = 0$...

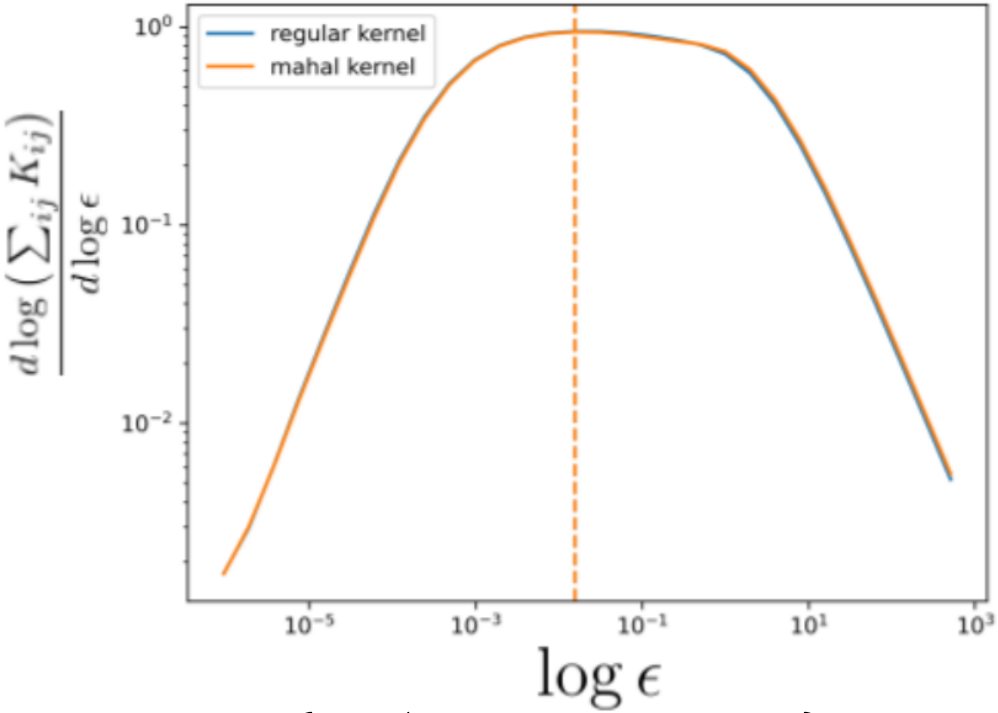
$$\nabla f(x_i)^\top M(x_i)\nabla g(x_i) \approx (\beta/2)\sum_j L_{ij}(f_i - f_j)(g_i - g_j)$$

For reactive current coordinate ℓ at i-th data point :

$$\boxed{[\hat{\mathcal{J}}_\epsilon]_{i,\ell} := \frac{\mu(x_i)}{\beta Z_\epsilon}\sum_{j=1}^{N}[L_\epsilon]_{ij}([q_\epsilon]_i - [q_\epsilon]_j)(x_{i,\ell} - x_{j,l}).}$$

ϵ-Picking Heuristic





1. Pick range of ϵ values, e.g: `eps-list` $= \{2^m \mid m = -15, \ldots, 1\}$
2. Compute $\sum_{ij} [K_\epsilon]_{ij}$ for each ϵ in `eps-list`
3. **Method**: Make a $\log$-$\log$ plot, pick ϵ in near-linear region
4. **Method++**: Pick ϵ maximizing $\frac{d(\log \sum_{ij} [K_\epsilon]_{ij})}{d \log \epsilon}$

---

3. Coifman, Shkolinsky, Sigworth, Singer (2008).
4. Berry and Harlim (2016).

Effective Dynamics in Collective Variables

The **free energy** F is defined for $x = (x_1, \ldots, x_d)^\top \in \mathbb{R}^d$ as

$$F(x) = -\beta^{-1} \ln \left(\int_{\mathbb{R}^n} Z^{-1} e^{-\beta V(y)} \prod_{i=1}^{d} \delta(\theta_i(y) - x_i) dy \right)$$

The **diffusion tensor** $M(x) \in \mathbb{R}^{d \times d}$ is defined in mass-weighted y-coordinates as

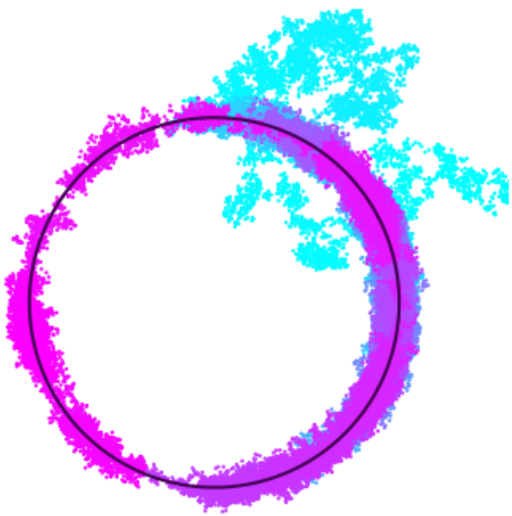
$$M_{ij}(x) = Z^{-1} e^{\beta F(x)} \int_{\mathbb{R}^n} \sum_{k=1}^{d} \frac{\partial x_i}{\partial y_k} \frac{\partial x_j}{\partial y_k} e^{-\beta V(y)} \prod_{\ell=1}^{d} \delta(\theta_\ell(y) - x_\ell) dy$$

Maragliano, Fischer, Vanden-Eijnden, Cicotti *J. Chem Phys* (2006)

Computing ∇F and M

Restrain system via "extended " potential

$$U(y; k, x) = V(y) + \frac{k}{2}||\theta(y) - x||^2$$



With restrained trajectory $\{y_t\}_{t=0}^{T}$:

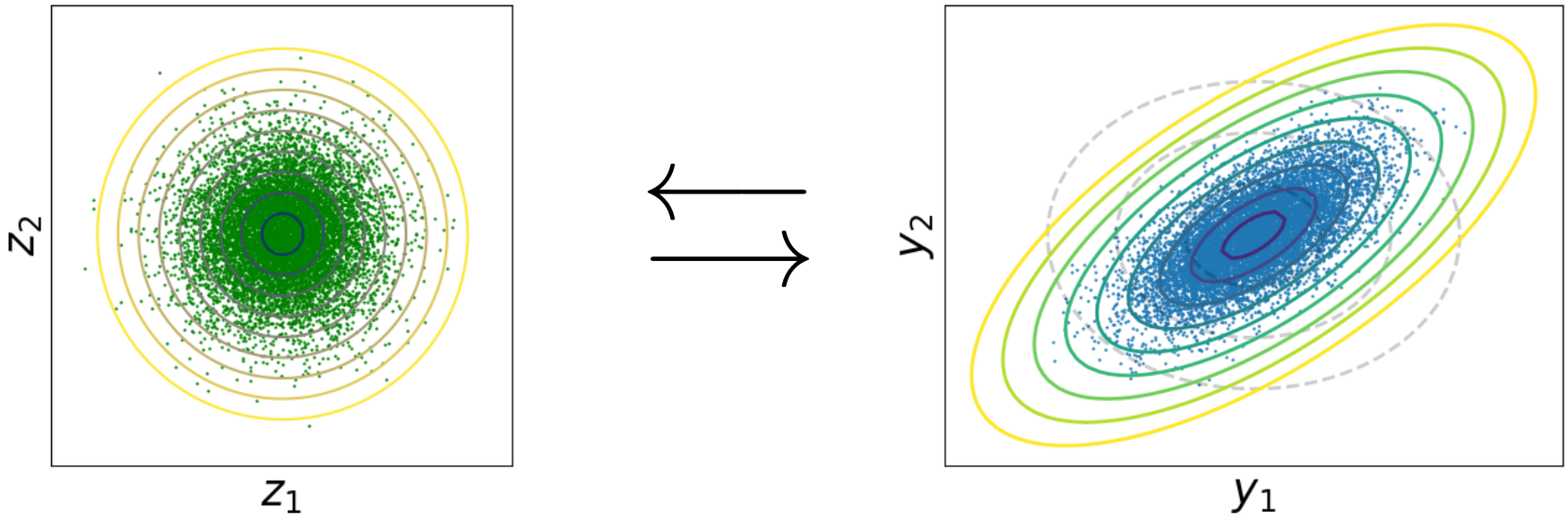
$$\nabla F(x) \approx \frac{k}{T} \int_0^T (x - \theta(y_t))dt$$

$$M_{ij}(x) \approx \frac{1}{T} \int_0^T \frac{\partial x_i}{\partial y_k} \frac{\partial x_j}{\partial y_k} dt$$

Maragliano, Fischer, Vanden-Eijnden, Cicotti, *J Chem. Phys.* (2006)

Mahalanobis Distance

Mahalanobis, Prasanta Chandra. "On the generalized distance in statistics." National Institute of Science in India, 1936.



Common appearance:

$$p(y) = \frac{1}{\sqrt{2\pi|C|}} \exp(\underbrace{-(y-\mu)^\top C^{-1}(y-\mu)}) $$

Definition: for distribution with covariance C, given samples y_1, y_2

$$||\mathbf{y_1} - \mathbf{y_2}||_M = \sqrt{(\mathbf{y_1} - \mathbf{y_2})^\top \mathbf{C}^{-1}(\mathbf{y}_1 - \mathbf{y}_2)}$$

Extending to Collective Variables: mmap

Singer, Coifman, *Applied and Comp. Harmonic Analysis*, (2008)

Mahalanobis Kernel: "mmap"

$$\exp\left(\frac{-(x_i - x_j)^\top (M^{-1}(x_i) + M^{-1}(x_j))(x_i - x_j)}{4\epsilon}\right) \approx \exp\left(\frac{-||z_i - z_j||^2}{2\epsilon}\right).$$

Scenario: Data x is output of an unknown diffeomorphism $z \mapsto x$ on $\mathbb{R}^d$.

Jacobian matrix $\mathrm{J}_{k\ell}(z) = \dfrac{\partial x^{(k)}}{\partial z^{(\ell)}}$ $\qquad M(x) := \mathrm{J}(z)\mathrm{J}(z)^\top$

$$(x - x_0)^\top \left[\frac{M^{-1}(x) + M^{-1}(x_0)}{2}\right](x - x_0) = ||z - z_0||_2^2 + \mathcal{O}(||z - z_0||^4)$$