FEAT: Free energy Estimators with Adaptive Transport

José Miguel Hernández–Lobato
Department of Engineering
University of Cambridge

http://jmhl.org, jmh233@cam.ac.uk

Jiajun He, Yuanqi Du, Francisco Vargas, Yuanqing Wang, Carla P. Gomes, José Miguel Hernández-Lobato, Eric Vanden-Eijnden. FEAT: Free energy Estimators with Adaptive Transport. arXiv:2504.11516 [stat.ML] 2025

Collaborators

Jiajun He

Yuanqi Du

Francisco Vargas

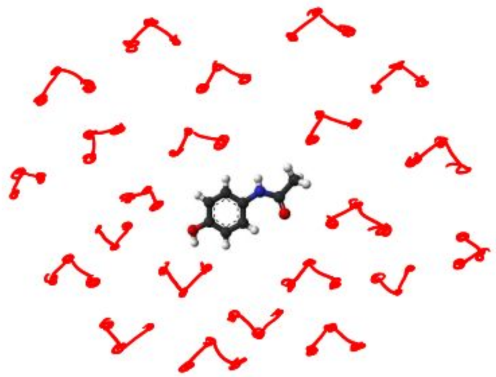
Yuanqing Wang

Carla P. Gomes

Eric Vanden-Eijnden

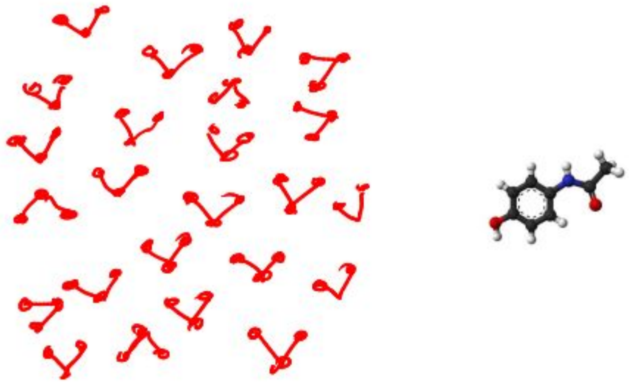
What state of a molecular system is more likely?

State A



Drug-like molecule surrounded by water molecules

State B



Drug-like molecule separated from water molecules

From energies to probabilities

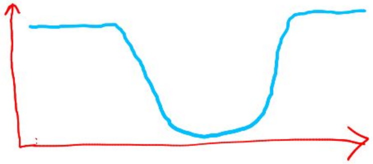
State A

State B

Energies

$u_A(x)$

$u_B(x)$



$p_A(x) = \exp\{-u_A(x) - f_A\}$

$p_B(x) = \exp\{-u_B(x) - f_B\}$

Probabilities

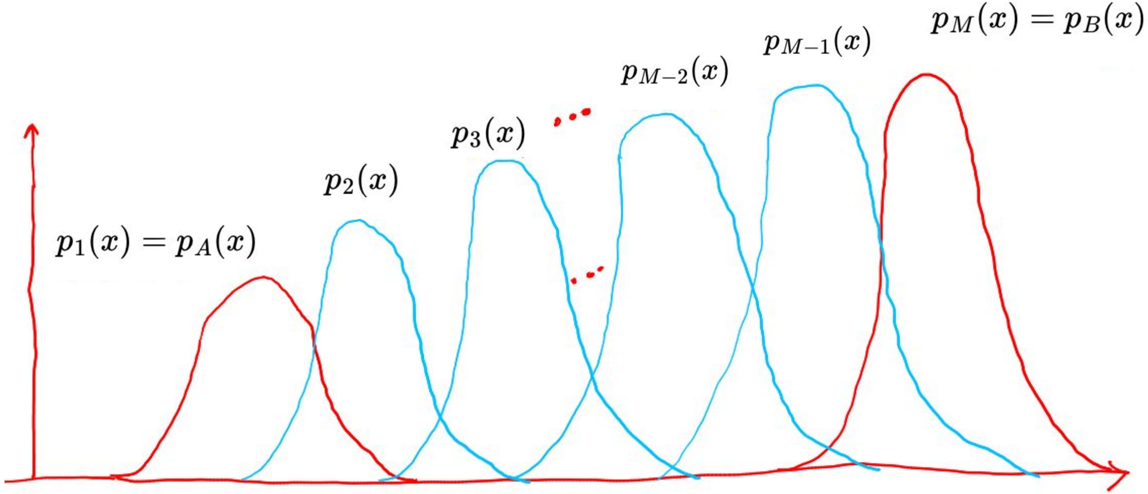
What is the relative probability mass associated with each state?

What is Δf = f_A - f_B? Δf is usually called the **free energy difference**.

Solution with **MBAR**

$$p(i_{n,m} = k \mid x_{n,m}) = \frac{\exp\{-u_k(x_{n,m}) - F_k\}}{\sum_j \exp\{-u_j(x_{n,m}) - F_j\}}$$

Use **interpolating distributions** with high overlap:



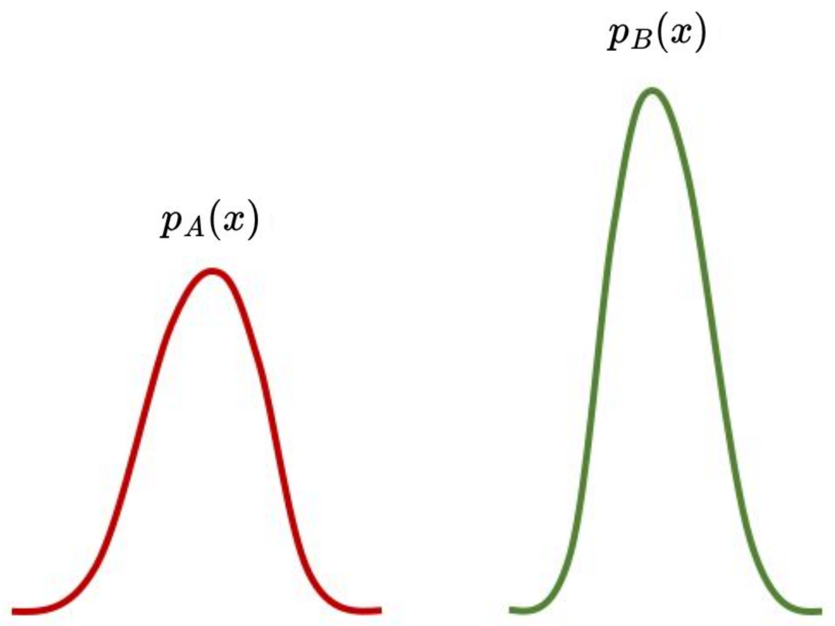
We assume we have N samples from each distribution:

$$\text{Data} = \{x_{n,m}\} \quad \text{where} \quad x_{n,m} \sim p_m(x) \quad n = 1, \ldots, N \quad m = 1, \ldots, M$$

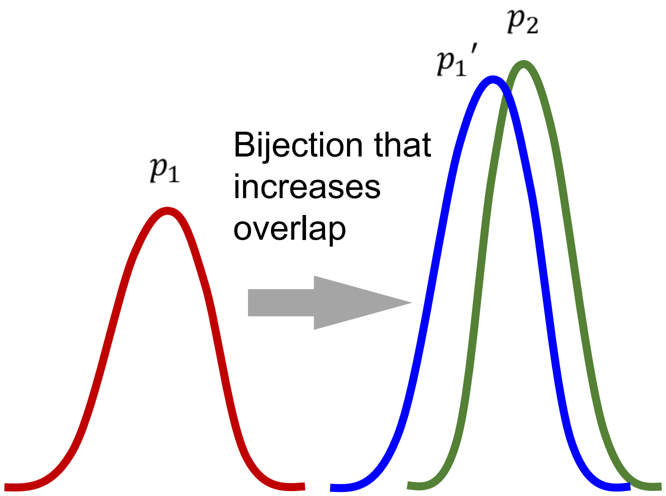
MBAR is great but what if

- No samples from intermediate distributions $p_2(x)$,…, $p_{M-1}(x)$ are available.

- We only have samples from $p_A(x)$ and $p_B(x)$.



Target FEP (use mapping to increase overlap)



$$x_0 \sim p_0(x) \qquad \frac{dx_t}{dt} = u_t(x_t)$$

$$x_t = x_0 + \int_{s=0}^{s=t} u_s(x_s) ds$$

$$\log p_t(x_t) = \log p_0(x_0) - \int_{s=0}^{s=t} \nabla \cdot u_s(x_s)\, ds$$

Limitations:

- Need to learn **flexible and accurate bijection**, e.g. using **flow matching** [3].
- But then **difficult** to calculate **changes in density due to transformation**.

[1] C. Jarzynski. Targeted free energy perturbation. *Physical Review E*, 65(4):046122, 2002.
[2] P. Wirnsberger, et al. Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics,* 153(14), 2020.
[3] L. Zhao and L. Wang. Bounding free energy difference with flow matching. *Chinese Physics Letters*, 40(12):120201, 2023.

Proposed approach

Learn forward and backward SDEs running **from t = 0 to t = 1** and **from t = 1 to t = 0** that interpolate between distributions p_A and p_B :

$$\mathrm{d}X_t = -\sigma^2 \nabla U_t(X_t)\mathrm{d}t + u_t\,\mathrm{d}t + \sigma\sqrt{2}\,\overrightarrow{\mathrm{d}B_t},\ X_0 \sim p_A$$
$$\mathrm{d}X_t = \sigma^2 \nabla U_t(X_t)\mathrm{d}t + u_t\,\mathrm{d}t + \sigma\sqrt{2}\,\overleftarrow{\mathrm{d}B_t},\ X_1 \sim p_B$$

where

- $U_t(X_t)$ is the energy of the marginal distribution for X_t , $(U_0 = U_A,\ U_1 = U_B)$.
- $u_t(X_t)$ is a control term ensuring the marginal of X_t has energy $U_t(X_t)$.

These SDEs are a **flexible mapping** between p_A and p_B .

Allow estimation of Δf without calculating expensive changes in density!

Example

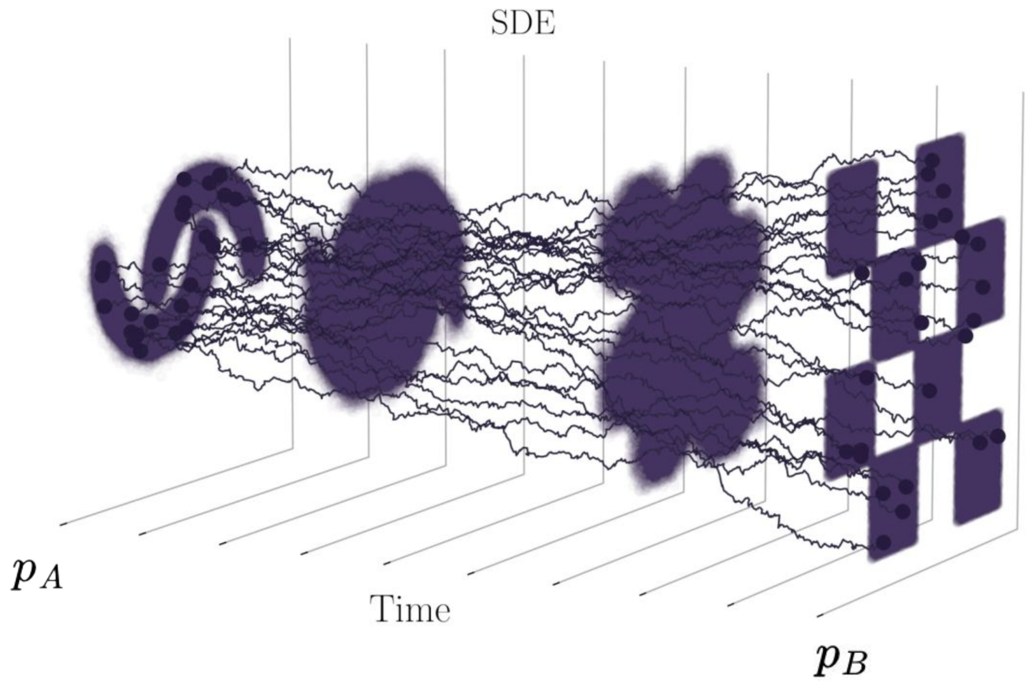


Figure source: Albergo, Michael S., Nicholas M. Boffi, and Eric Vanden-Eijnden.
"Stochastic interpolants: A unifying framework for flows and diffusions." arXiv preprint arXiv:2303.08797 (2023).

Calculating Δf with forward and backward SDEs

Let $t_1 = 0, \ldots, t_M = 1$ be times at which we discretize the SDEs trajectories:

$$\overrightarrow{X}_{t_{i+1}} = \overrightarrow{X}_{t_i} - \sigma^2 \nabla U_{t_i}(\overrightarrow{X}_{t_i})\Delta t_i + u_{t_i}(\overrightarrow{X}_{t_i})\Delta t_i + \sqrt{2\Delta t_i}\sigma\eta_i, \qquad \overrightarrow{X}_0 \sim p_A$$

$$\overleftarrow{X}_{t_{i-1}} = \overleftarrow{X}_{t_i} - \sigma^2 \nabla U_{t_i}(\overleftarrow{X}_{t_i})\Delta t_{i-1} - u_{t_i}(\overleftarrow{X}_{t_i})\Delta t_{i-1} + \sqrt{2\Delta t_{i-1}}\sigma\eta_i, \qquad \overleftarrow{X}_1 \sim p_B$$

where $\Delta t_i = t_{i+1} - t_i$ and $\eta_i \sim \mathcal{N}(0, \mathbf{I})$. Then Δf can be obtained using **importance sampling**:

$$\Delta f = -\log \mathbb{E}\underbrace{\left[\frac{\exp(-U_b(\overrightarrow{X}_1))\prod_{i=1}^{M}\mathcal{N}^-(\overrightarrow{X}_{t_{i-1}}|\overrightarrow{X}_{t_i})}{\exp(-U_a(\overrightarrow{X}_0))\prod_{i=1}^{M}\mathcal{N}^+(\overrightarrow{X}_{t_i}|\overrightarrow{X}_{t_{i-1}})}\right]}_{\exp\text{-}\widetilde{W}(\overrightarrow{X})} = \log \mathbb{E}\underbrace{\left[\frac{\exp(-U_a(\overleftarrow{X}_0))\prod_{i=1}^{M}\mathcal{N}^+(\overleftarrow{X}_{t_i}|\overleftarrow{X}_{t_{i-1}})}{\exp(-U_b(\overleftarrow{X}_1))\prod_{i=1}^{M}\mathcal{N}^-(\overleftarrow{X}_{t_{i-1}}|\overleftarrow{X}_{t_i})}\right]}_{\exp \widetilde{W}(\overleftarrow{X})}$$

$\widetilde{W}$ is **forward/backward log density ratio**. As $M \rightarrow \infty$, $\widetilde{W}$ converges to the **generalized work**.

Improvement with BAR-style approach

Recall the importance sampling estimator:

$$\Delta f = -\log \mathbb{E}\underbrace{\left[\frac{\exp(-U_b(\overrightarrow{X}_1))\prod_{i=1}^{M}\mathcal{N}^-(\overrightarrow{X}_{t_{i-1}}|\overrightarrow{X}_{t_i})}{\exp(-U_a(\overrightarrow{X}_0))\prod_{i=1}^{M}\mathcal{N}^+(\overrightarrow{X}_{t_i}|\overrightarrow{X}_{t_{i-1}})}\right]}_{\exp\text{ - }\widetilde{W}(\overrightarrow{X})} = \log \mathbb{E}\underbrace{\left[\frac{\exp(-U_a(\overleftarrow{X}_0))\prod_{i=1}^{M}\mathcal{N}^+(\overleftarrow{X}_{t_i}|\overleftarrow{X}_{t_{i-1}})}{\exp(-U_b(\overleftarrow{X}_1))\prod_{i=1}^{M}\mathcal{N}^-(\overleftarrow{X}_{t_{i-1}}|\overleftarrow{X}_{t_i})}\right]}_{\exp\ \widetilde{W}(\overleftarrow{X})}$$

Use instead a BAR-style **minimum variance method**:

$$\Delta f = \log \frac{\mathbf{E}[g(-\widetilde{W}(\overleftarrow{X}) + C)]}{\mathbf{E}[g(\widetilde{W}(\overrightarrow{X}) - C)]} \qquad g(x) = \frac{1}{1+\exp(x)}, C = \Delta f$$

1. Initialize C;
2. Calculate Δf; Set $C \leftarrow \Delta f$;
3. Repeat (2) until converge.

How to learn the SDEs?

We use the framework of **Stochastic Interpolants (SI)**, defining the **interpolant**

$$X_t = \alpha_t x_A + \beta_t x_B + \gamma_t \epsilon, \quad x_A \sim p_A,\ x_B \sim p_B,\ \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

where $\alpha_0 = 1, \alpha_1 = 0; \beta_0 = 0, \beta_1 = 1;$ and $\gamma_0 = \gamma_1 = 0$.

The distribution of the interpolant is the same as the one generated by the SDEs if

$$u_t(x) = \mathbb{E}[\partial_t X_t \mid X_t = x], \quad \nabla U_t(x) = \gamma_t^{-1}\mathbb{E}[\epsilon \mid X_t = x]$$

If we parametrize these functions as **neural networks** we can learn them by minimizing

$$\mathcal{L}_u(\psi) = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\,\mathbb{E}_{x_A,x_B,\epsilon}\left[\lambda_t \left\| u_t^{\psi}(X_t) - \partial_t X_t \right\|^2\right]$$

$$\mathcal{L}_U^{\text{DSM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\,\mathbb{E}_{x_A,x_B,\epsilon}\left[\eta_t \left\| \nabla U_t^{\theta}(X_t) - \gamma_t^{-1}\epsilon \right\|^2\right]$$

M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In 11th International Conference on Learning Representations, ICLR 2023, 2023.

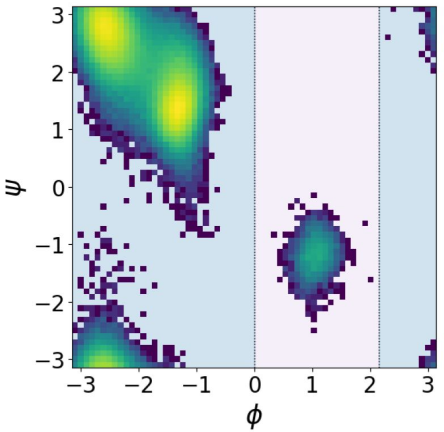
Experimental validation

- **GMM**:
  16-mode GMM and a **40-mode GMM.**

- **LJ system**:
  without LJ potential and system **with LJ potential.**

- **Alanine dipeptide – Solvation (ALDP-S)**:
  ALDP in **vacuum** and ALDP in **implicit solvent.**

- **Alanine dipeptide – Transition (ALDP-T)**:
  ALDP in **two meta-stable states.**



Results

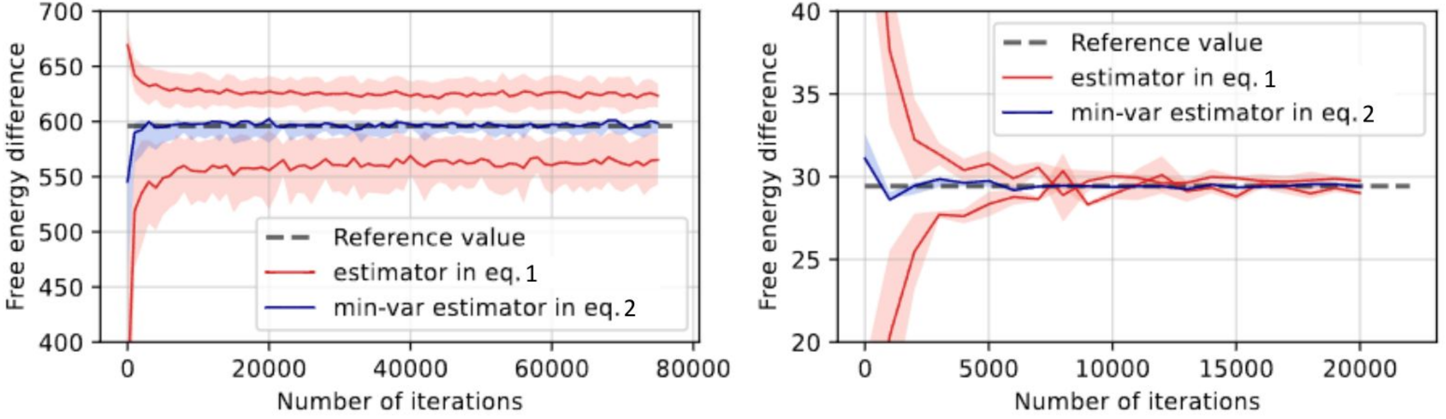
Method	GMM		LJ		
	$d = 40$	$d = 100$	$d = 55 \times 3$	$d = 79 \times 3$	$d = 128 \times 3$
Reference	0	0	234.77 ± 0.09	357.43 ± 3.43	595.98 ± 0.58
Target FEP w. FM	0.09 ± 0.26	-17.96 ± 1.49	232.06 ± 0.03	*	*
Neural TI	-181.63 ± 6.65	-402.93 ± 283.75	328.55 ± 336.39	468.76 ± 391.16	N/A
Ours	0.04 ± 0.04	-5.34 ± 1.52	232.47 ± 0.15	356.74 ± 0.79	595.04 ± 6.52

Method	ALDP-S	ALDP-T
	$d = 22 \times 3$	$d = 22 \times 3$
Reference	29.43 ± 0.01	-4.25 ± 0.05
Target FEP w. FM	29.47 ± 0.22	-4.78 ± 0.32
Neural TI	24.93 ± 3.13	-4.11 ± 2.56
Ours	29.38 ± 0.04	-4.56 ± 0.08

Results

Eq 1: $\Delta f = -\log \mathbf{E}_{\vec{\mathbf{P}}}[\exp(-W)] = \log \mathbf{E}_{\overleftarrow{\mathbf{P}}}[\exp(W)]$ **Importance sampling estimator**

Eq 2: $\Delta f = -\log \frac{\mathbf{E}_{\vec{\mathbf{P}}}[g(W-C)]}{\mathbf{E}_{\overleftarrow{\mathbf{P}}}[g(-W+C)]} + C$ **Minimum variance estimator**



(b) LJ-128. (c) ALDP-S.

Take home messages

FEAT: Free energy Estimators with Adaptive Transport is a method that

- calculates free energy differences using only samples from states A and B,
- does not need samples from intermediate distributions,
- uses SI framework to learn SDEs that flexibly map between distributions,
- avoids calculating expensive changes in density due to sample transport,
- uses a BAR-like minimum variance estimator,
- outperforms baselines like Neural TI and Neural target FEP,

Limitations: requires extra cost for training neural networks which may not be accurate enough in large systems.

Thanks!