Microsoft

Timewarp

Transferable acceleration of molecular dynamics by learning time-coarsened dynamics

Speaker: Leon Klein
+ Main contributors: Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, Ryota Tomioka

25 October 2024

The Timewarp team

Leon

Andrew

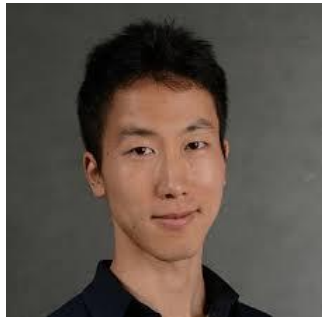
Tor

Bruno

Marc

Sebastian

Frank

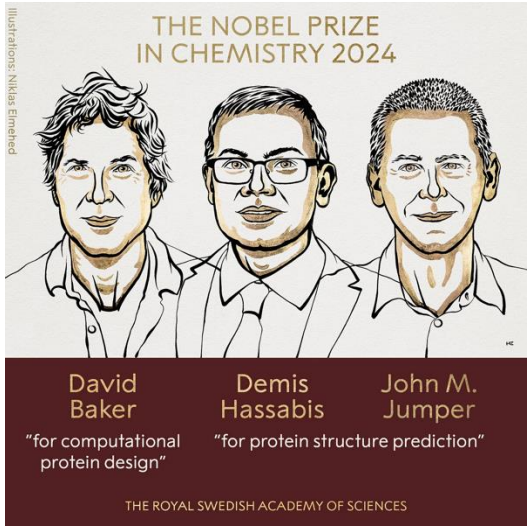


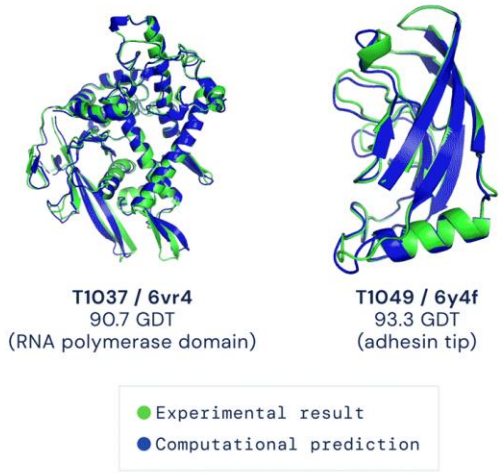
Ryota

+ entire AI4Science team

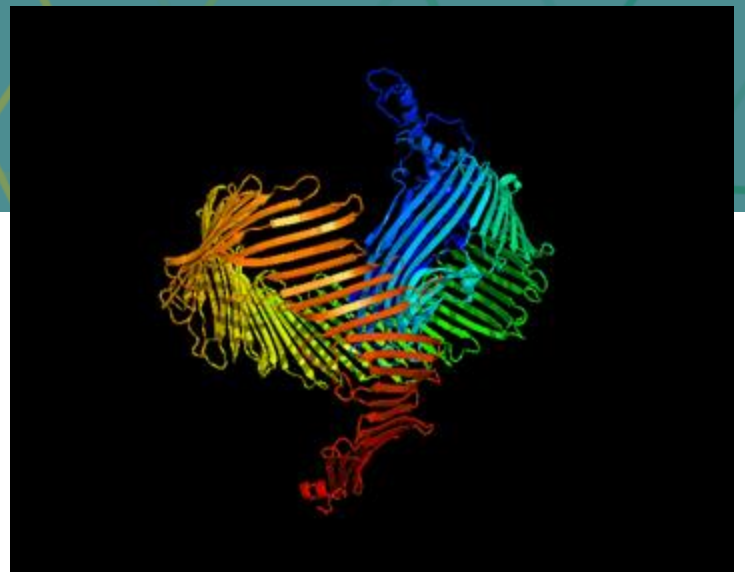
Breakthrough in ML for Proteins

- DeepMind's AlphaFold solves *protein-folding*.
- Predicts 3D structure from amino acid sequence.
- *But* static 3D protein structure not everything!
- Want to understand *dynamics* and *interactions*.
- Need to return to *Molecular Dynamics (MD)*.





Molecular Dynamics (MD)



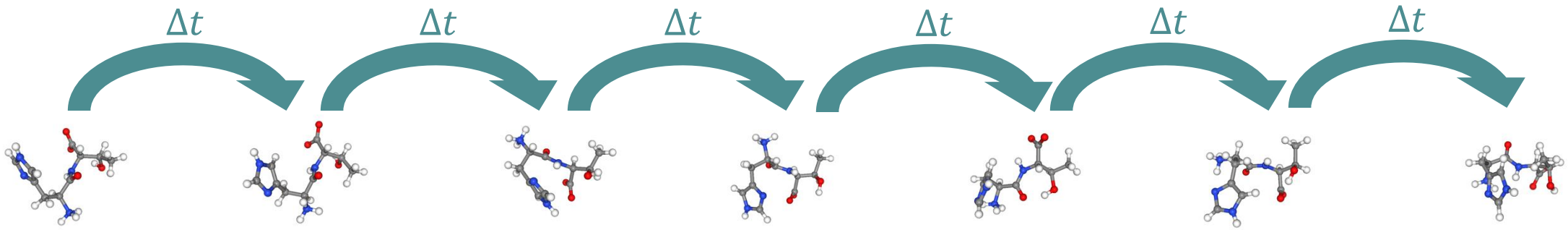
- MD simulates stochastic molecular motions

$$m\frac{d^2x}{dt^2} = -\nabla U(x) - \gamma\frac{dx}{dt} + \mathbf{R}(t)$$

- Problem: Biophysical processes take $\sim 1\mathrm{ms}$ or more. **Too long!**

→ ***Timewarp***: *can we tackle this with deep learning?*

$$\Delta t = 1\mathrm{fs} = 10^{-15}\mathrm{s}$$

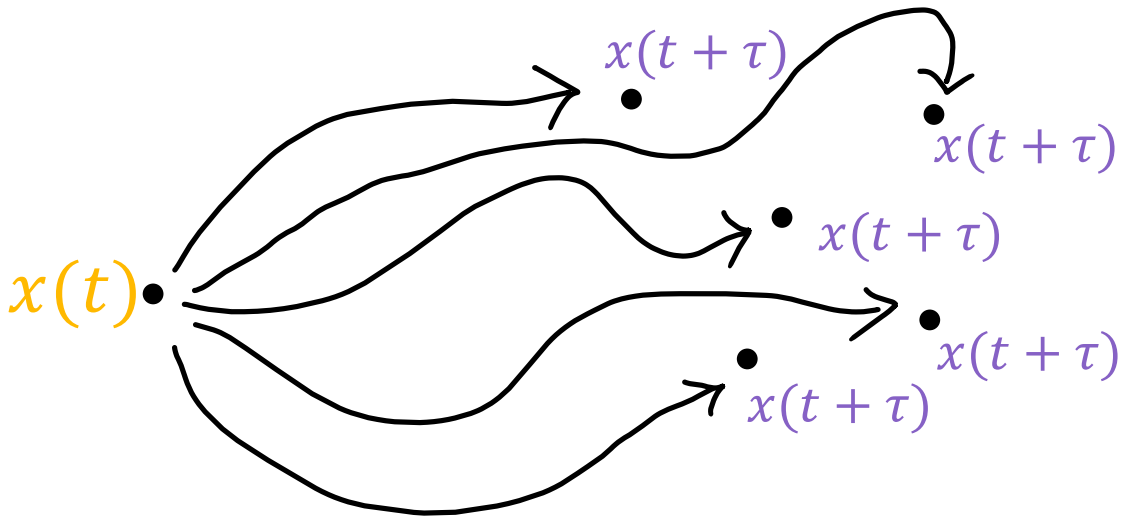


Boltzmann distribution

- We want to sample the Boltzmann distribution

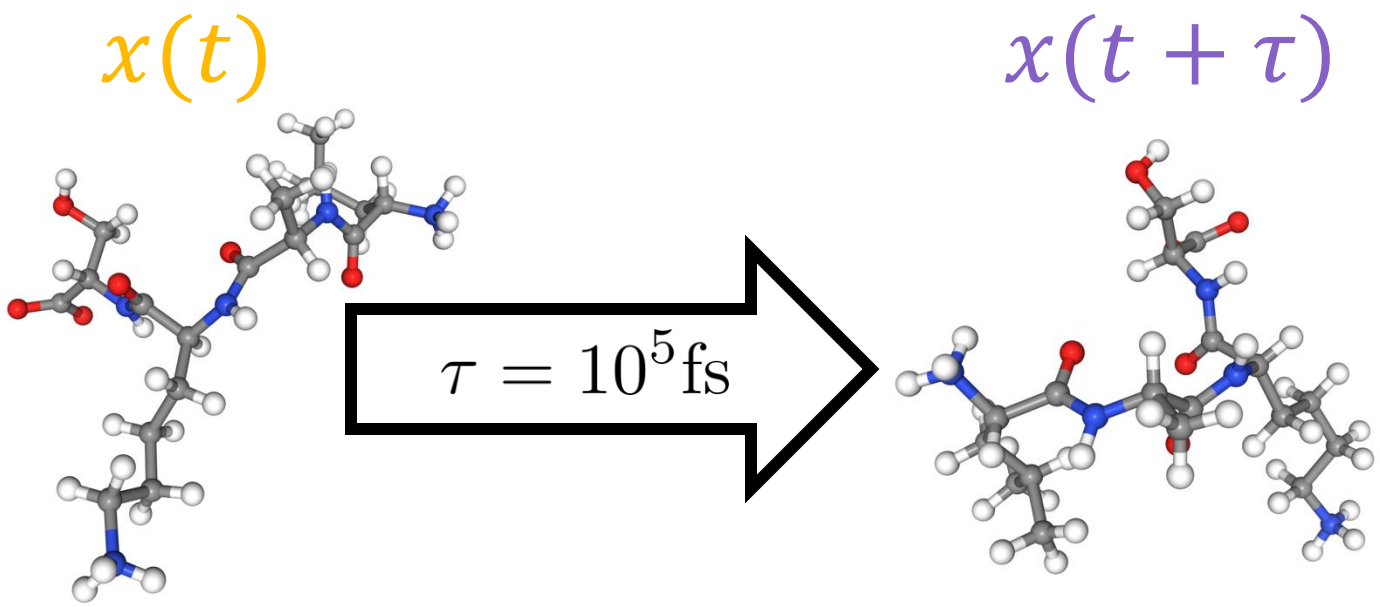
$$\mu(x) \propto \exp\left(-\frac{U(x)}{k_B T}\right)$$

- Long MD trajectories provide samples from $\mu(x)$ asymptotically.
- But first consider the conditional distribution $\mu(x(t+\tau)|x(t))$



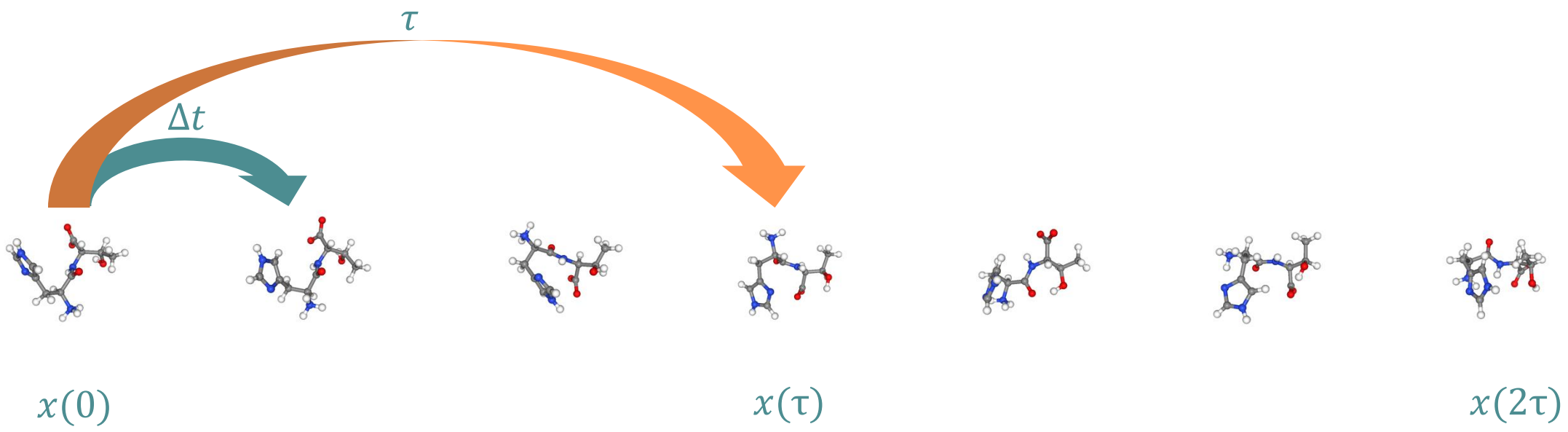
Timewarp – acceleration of molecular dynamics

- Speed-up by proposing large time steps $\tau \gg \Delta t$.
- Unbiased: correct samples with Metropolis-Hastings.



Datasets

- Generate MD trajectories of small peptides.
- Subsample the trajectories: $x(\tau), x(2\tau), x(3\tau), \ldots$
- Train model to predict $x(t+\tau)$ given $x(t)$.
- **Goal:** speed up sampling on *test* peptides.



Model desiderata

Fast sampling to quickly generate trajectories.

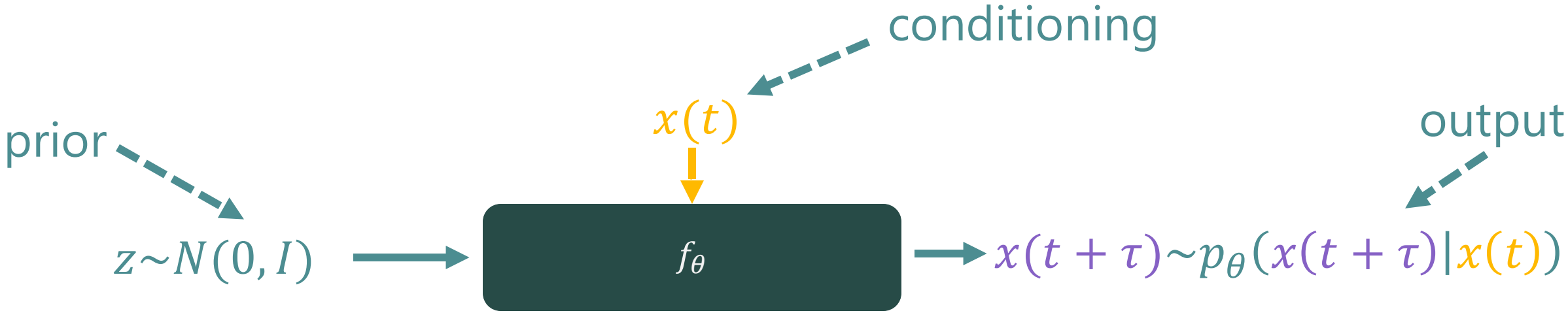
Tractable likelihood to allow for Metropolis-Hastings correction.

Train on train set of proteins, **transfer** to test set of new proteins.

Incorporate symmetries of the physical system.

Conditional normalising flows

- Want to model $\mu(x(t+\tau)|x(t))$ with $p_\theta(x(t+\tau)|x(t))$.
- Use a *conditional* normalising flow: $x(t+\tau) \coloneqq f_\theta(z; x(t))$



- Map from z to $x(t+\tau)$ is invertible, but not $x(t)$ to $x(t+\tau)$.
- Tractable formula for $p_\theta(x(t+\tau)|x(t))$.

Model desiderata

Fast sampling to quickly generate trajectories.

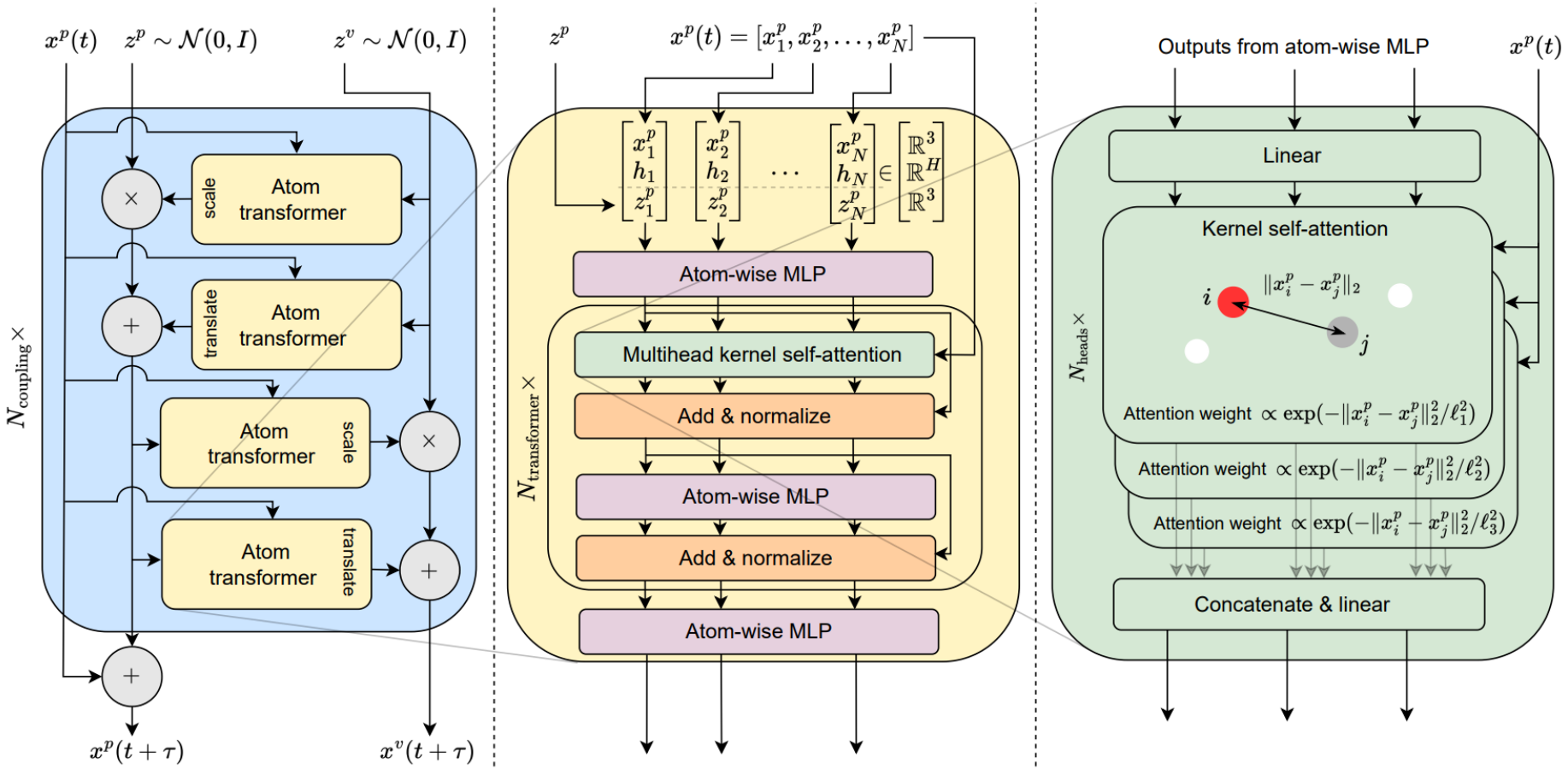
Tractable likelihood to allow for Metropolis-Hastings correction

Train on train set of proteins, **transfer** to test set of new proteins

Incorporate symmetries of the physical system.

Conditional flow architecture

- Augmented RealNVP
- All-atom representation
- Cartesian coordinates
- Permutation equivariant transformer



Model desiderata

Fast sampling to quickly generate trajectories.

Tractable likelihood to allow for Metropolis-Hastings correction

Train on train set of proteins, **transfer** to test set of new proteins

Incorporate symmetries of the physical system.

Training objective

- Two stages: *likelihood training* and *acceptance training*.
- Likelihood training: $L_{\mathrm{lik}}(\theta) = \frac{1}{K}\sum_{k=1}^{K} \log p_\theta(x_k(t+\tau)|x_k(t))$.
- Acceptance training maximises acceptance probability:

$$r_\theta(x, \tilde{x}) = \frac{\mu(\tilde{x})\, p_\theta(x|\tilde{x})}{\mu(x)\, p_\theta(\tilde{x}|x)}, \quad L_{\mathrm{acc}}(\theta) = \frac{1}{K}\sum_{k=1}^{K} \log r_\theta(x_k(t), \tilde{x}_k(t+\tau)).$$

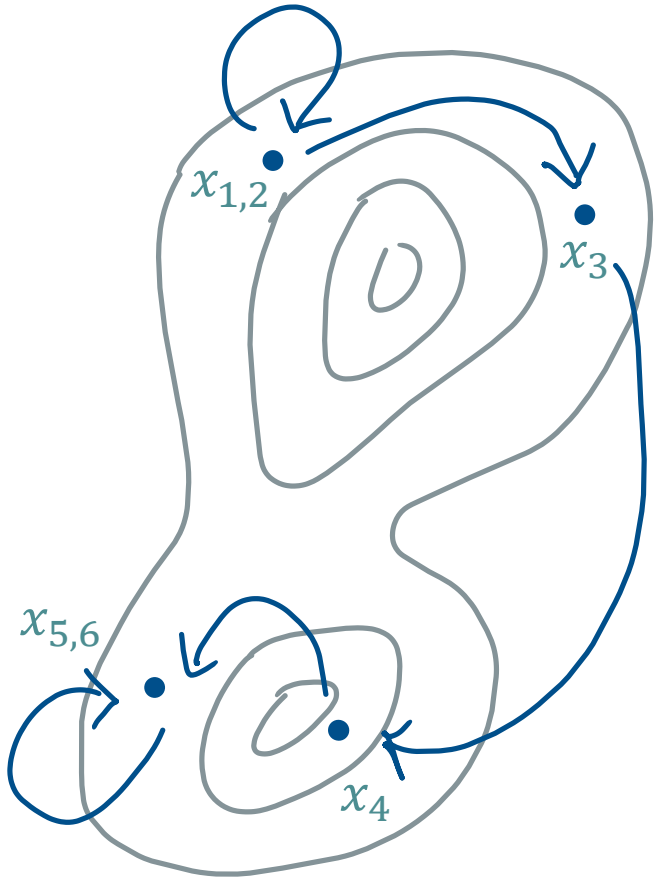
- Weighted with an entropy term to encourage exploration:

$$L_{\mathrm{ent}}(\theta) = -\frac{1}{K}\sum_{k=1}^{K} \log p_\theta(\tilde{x}_k(t+\tau)|x_k(t))$$

Sampling

Timewarp MCMC

- Sample with Metropolis-Hastings correction
- Asymptotically unbiased

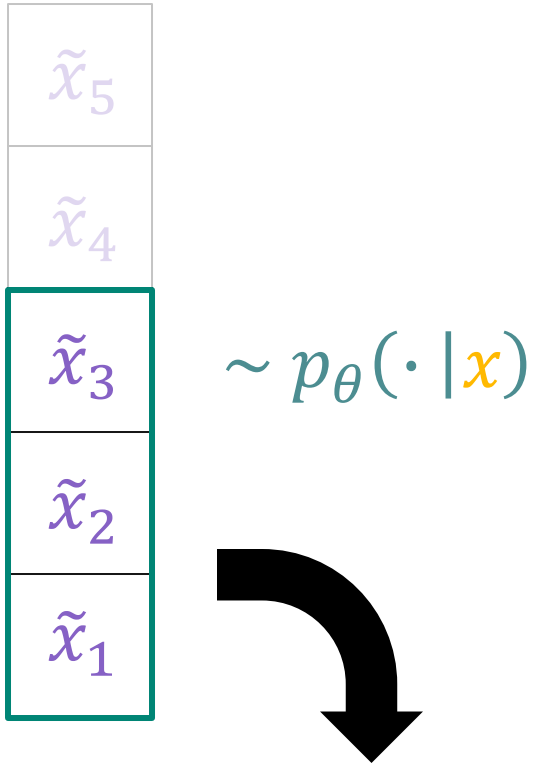


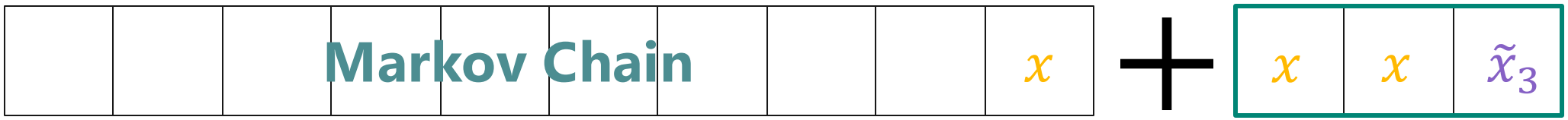
Timewarp MCMC algorithm

1. Sample $\tilde{x}_i \sim p_\theta(\cdot\,|x)$ in parallel.
2. Compute acceptance ratios

$$\alpha(x, \tilde{x}_i) = \min\left(1, \frac{\mu(\tilde{x}_i)\, p_\theta(x|\tilde{x}_i)}{\mu(x)\, p_\theta(\tilde{x}_i|x)}\right)$$

3. Accept $\tilde{x}_i$ with probability $\alpha(x, \tilde{x}_i)$.
4. Add x for each rejected sample to Markov Chain
5. Add the first accepted $\tilde{x}_i$ to the Markov Chain

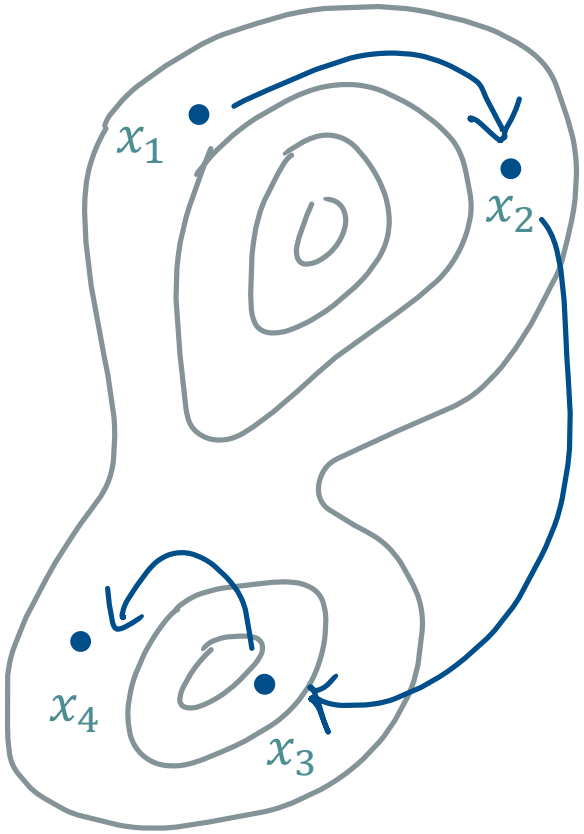




Sampling

- **Timewarp MCMC**
  - Sample with Metropolis-Hastings correction
  - Asymptotically unbiased

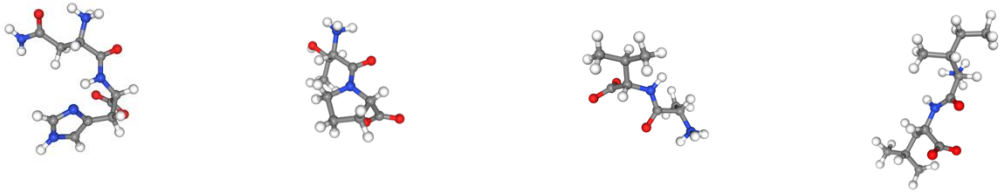
- **Timewarp exploration**
  - Every proposal is accepted
  - Potentially biased samples
  - Faster exploration



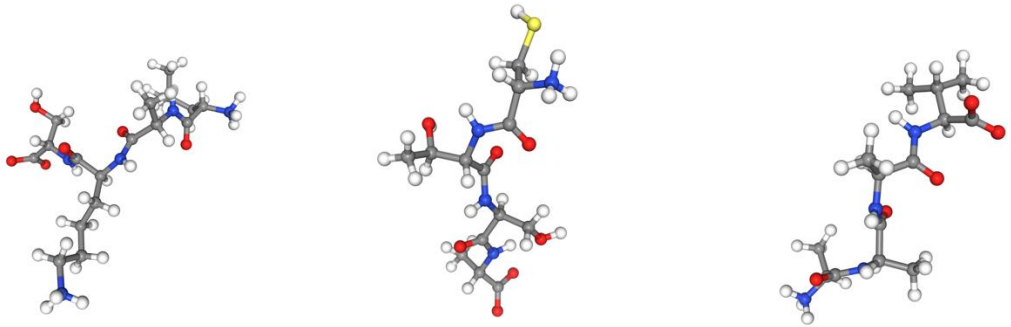
Experiments

Datasets

- Dipeptides (2 amino acids)
  - Number of peptides: 400
  - Train set: 200 dipeptides
  - Time step $\tau = 1ns = 10^6$ MD steps



- Tetrapeptides (4 amino acids)
  - Number of peptides: 20^4
  - Train set: 1500 tetrapeptides ~ 1%
  - Time step $\tau = 100ps = 10^5$ MD steps

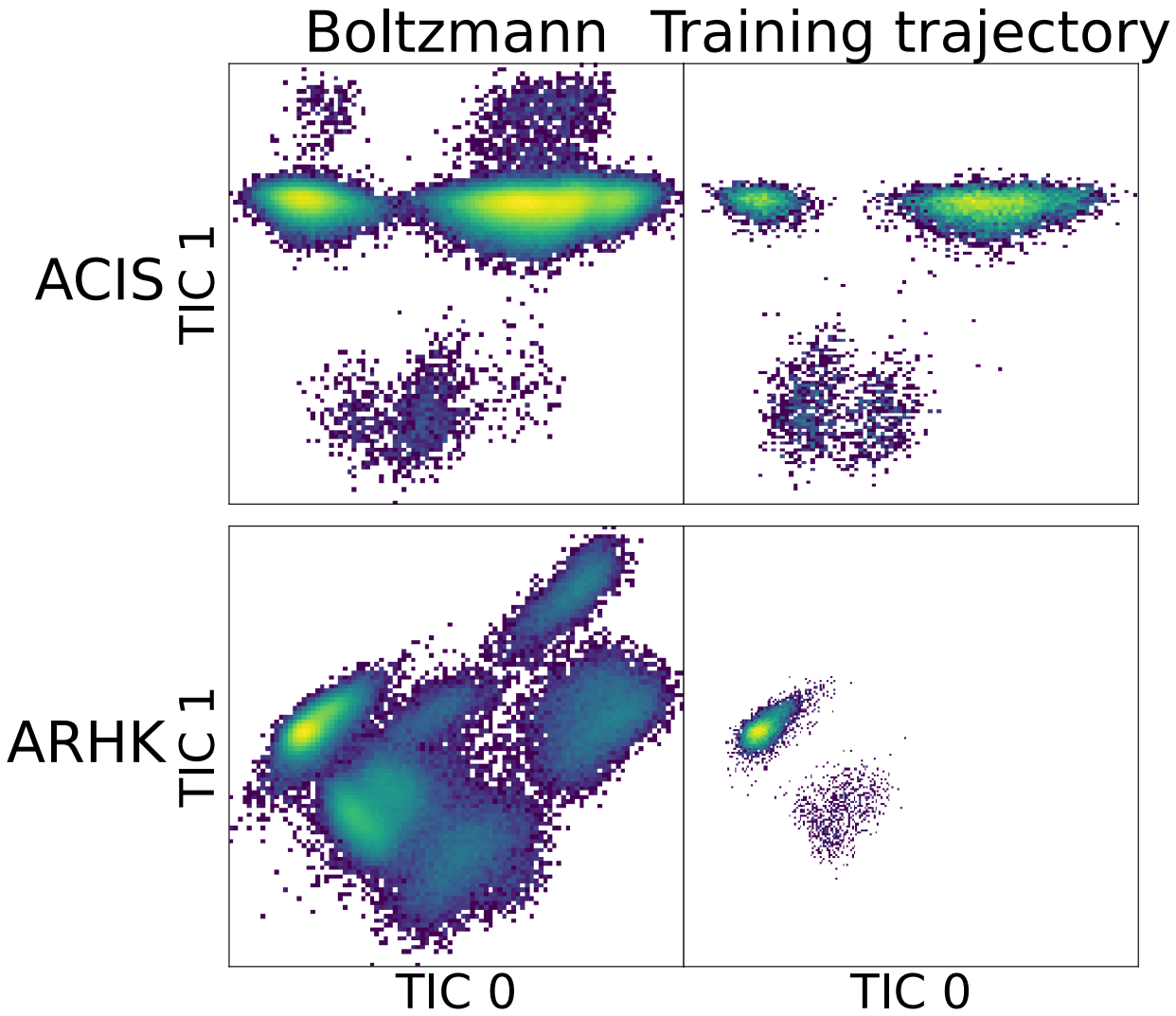


Training trajectories

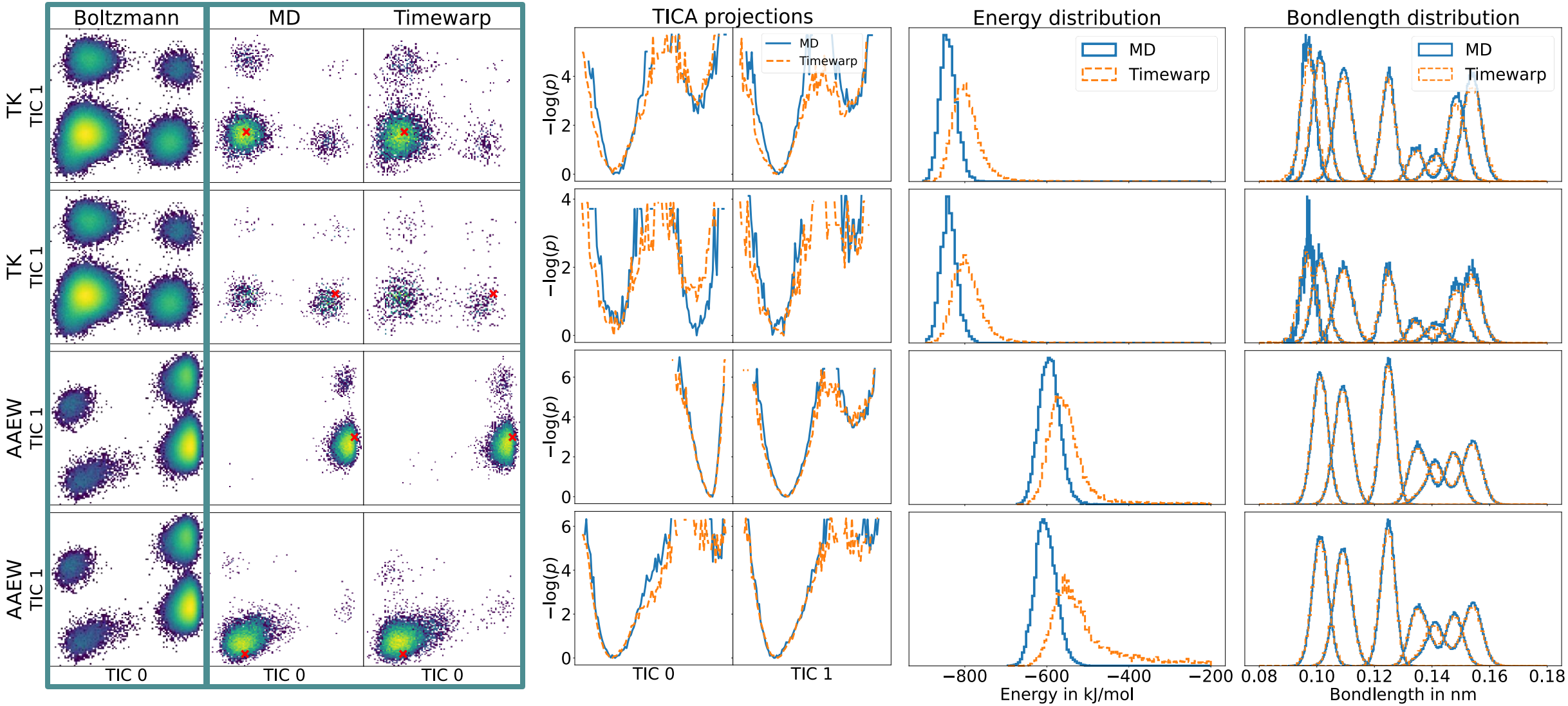
- Short training trajectories of $50ns$
- Training trajectories miss some metastable states

TICA projections:

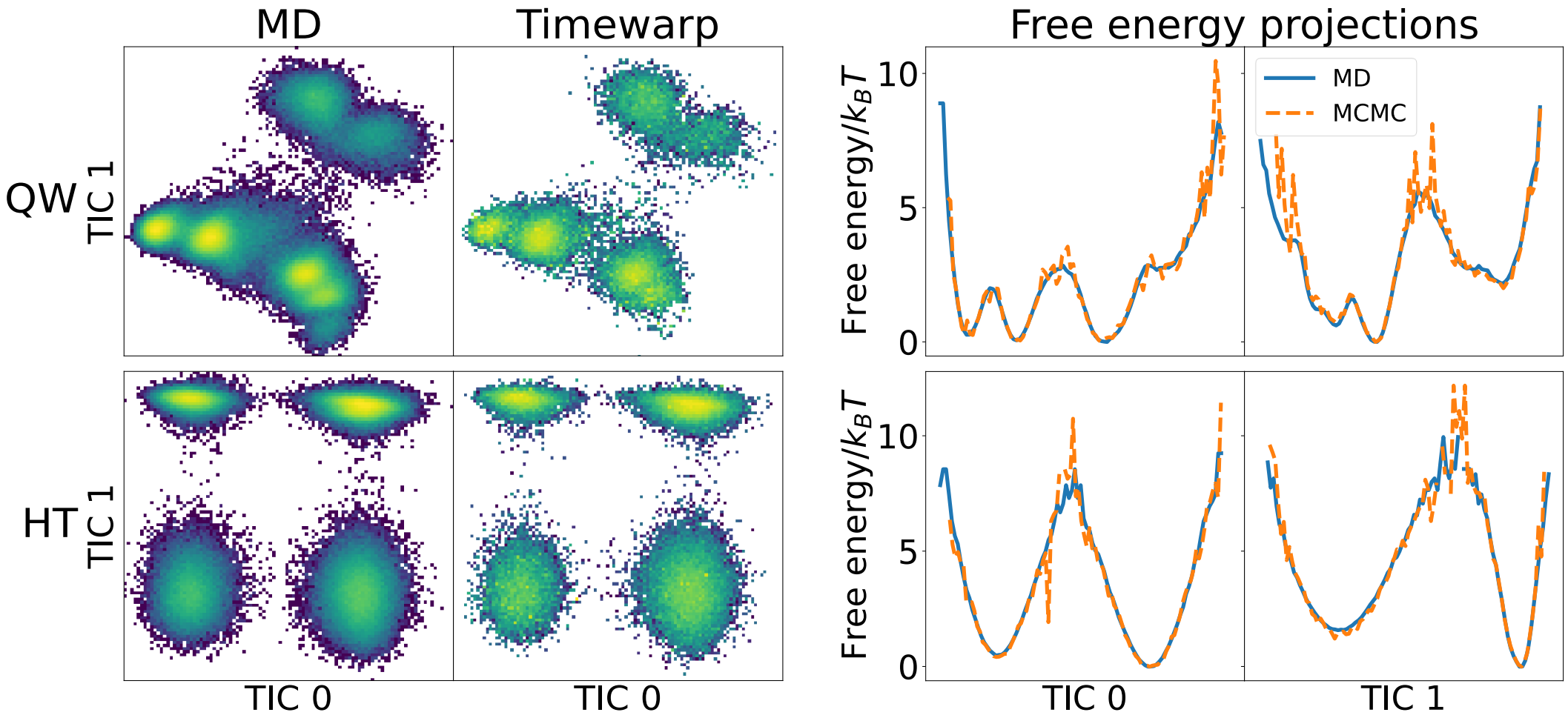
- Extract slowest processes
- Shows meta-stable states



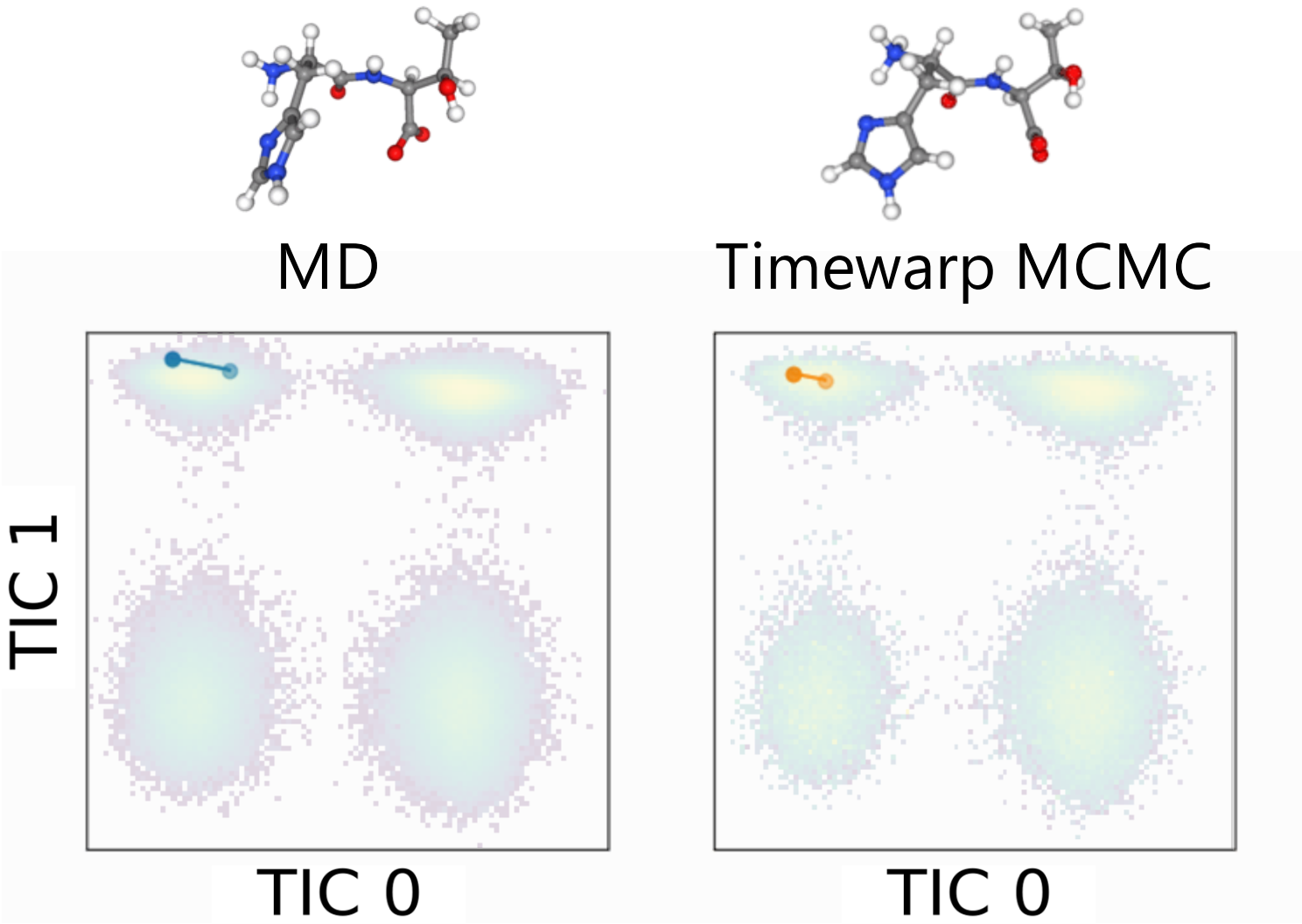
Conditional distribution $\mu(x(t+\tau)|x(t))$



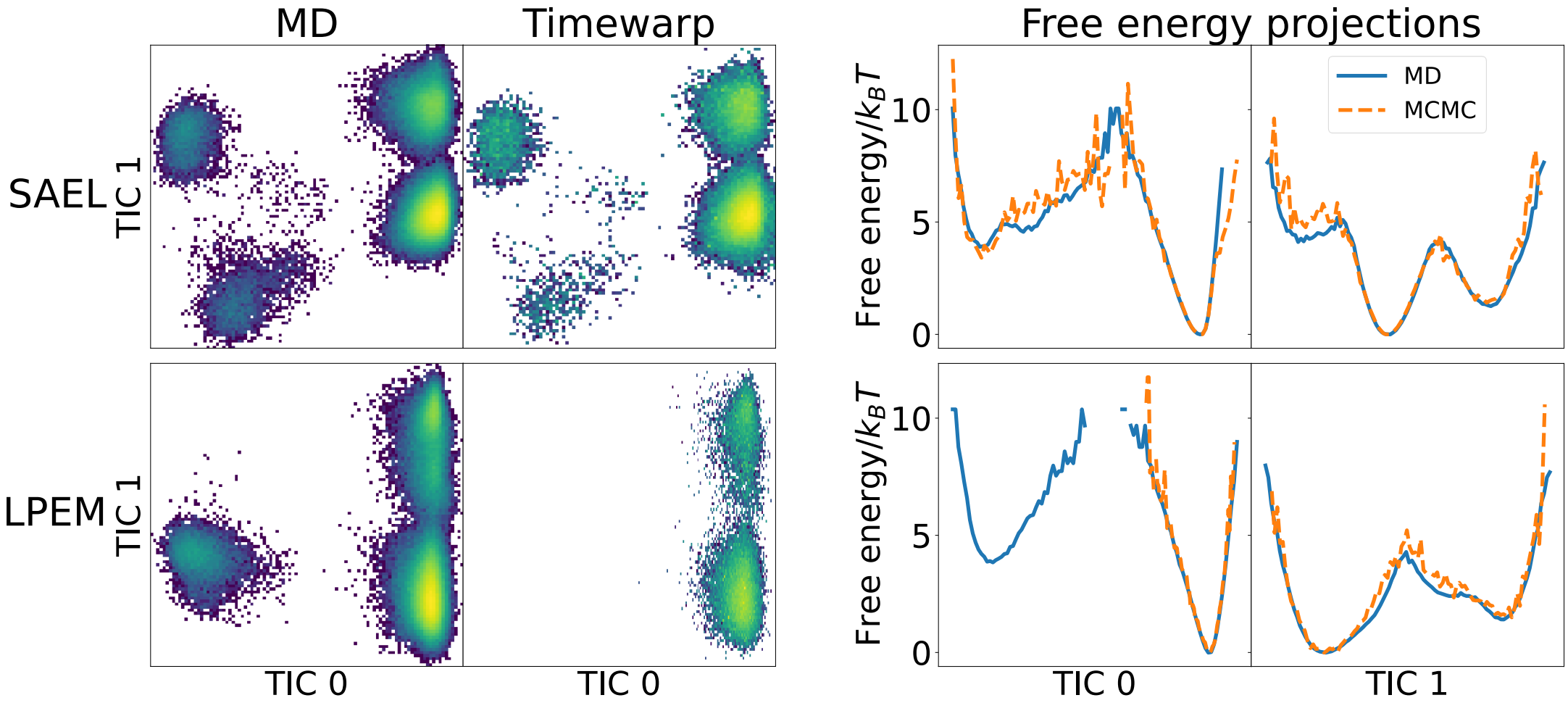
Targeting the Boltzmann distribution - dipeptides



Targeting the Boltzmann distribution - dipeptides

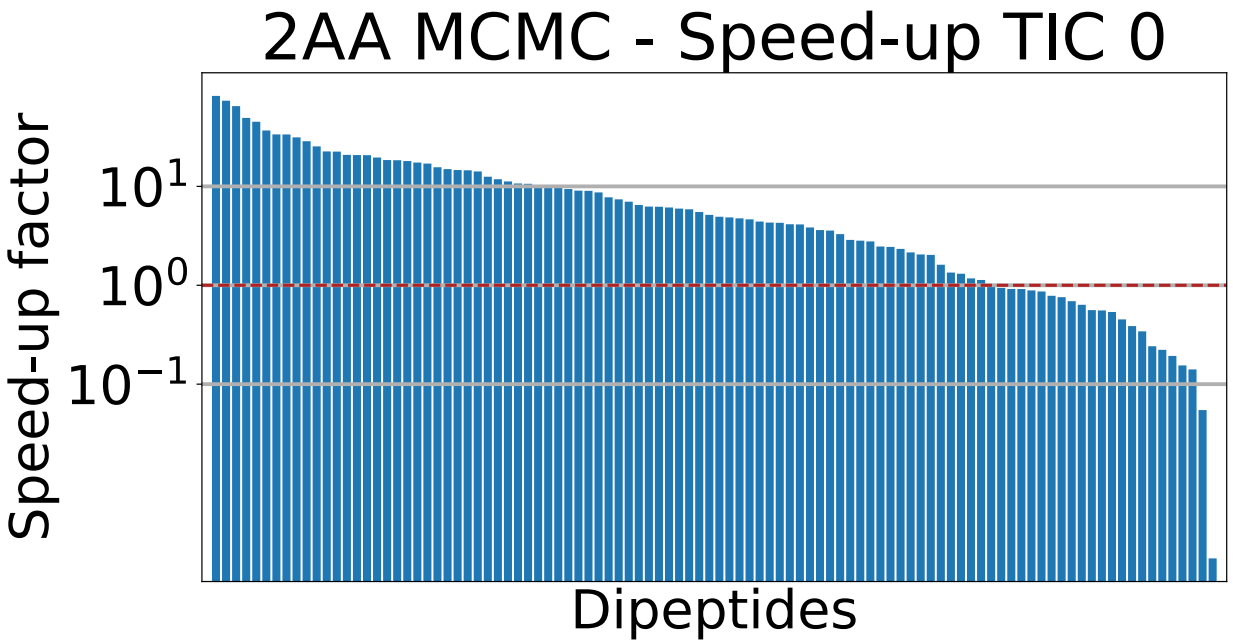


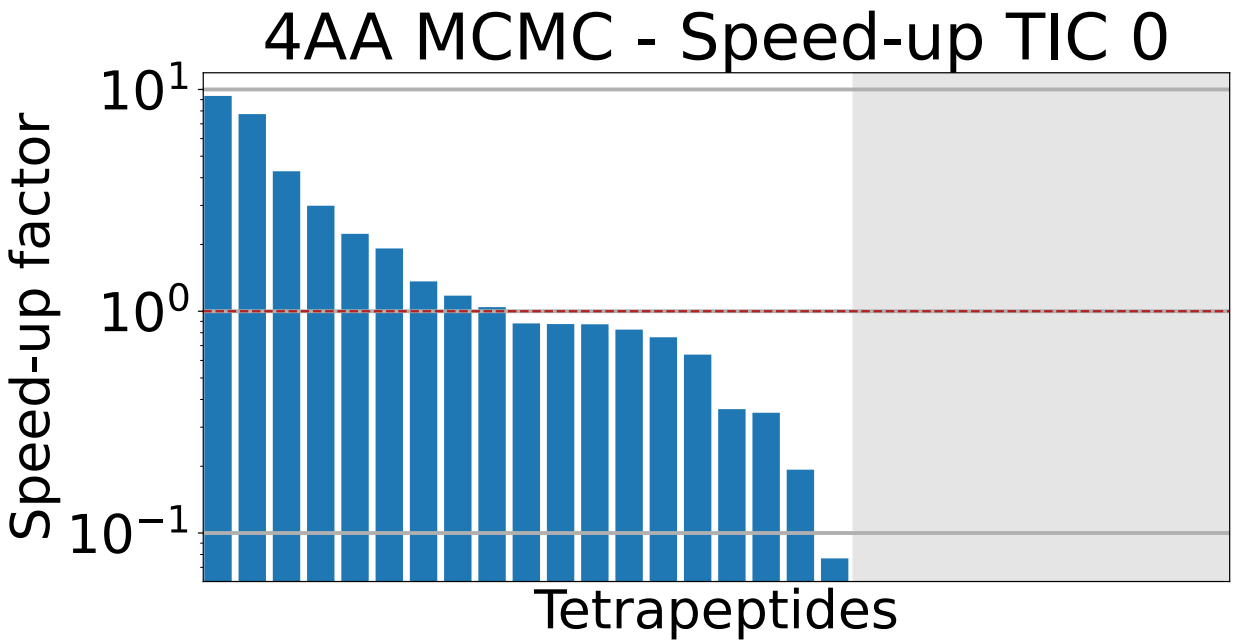
Targeting the Boltzmann distribution - tetrapeptides



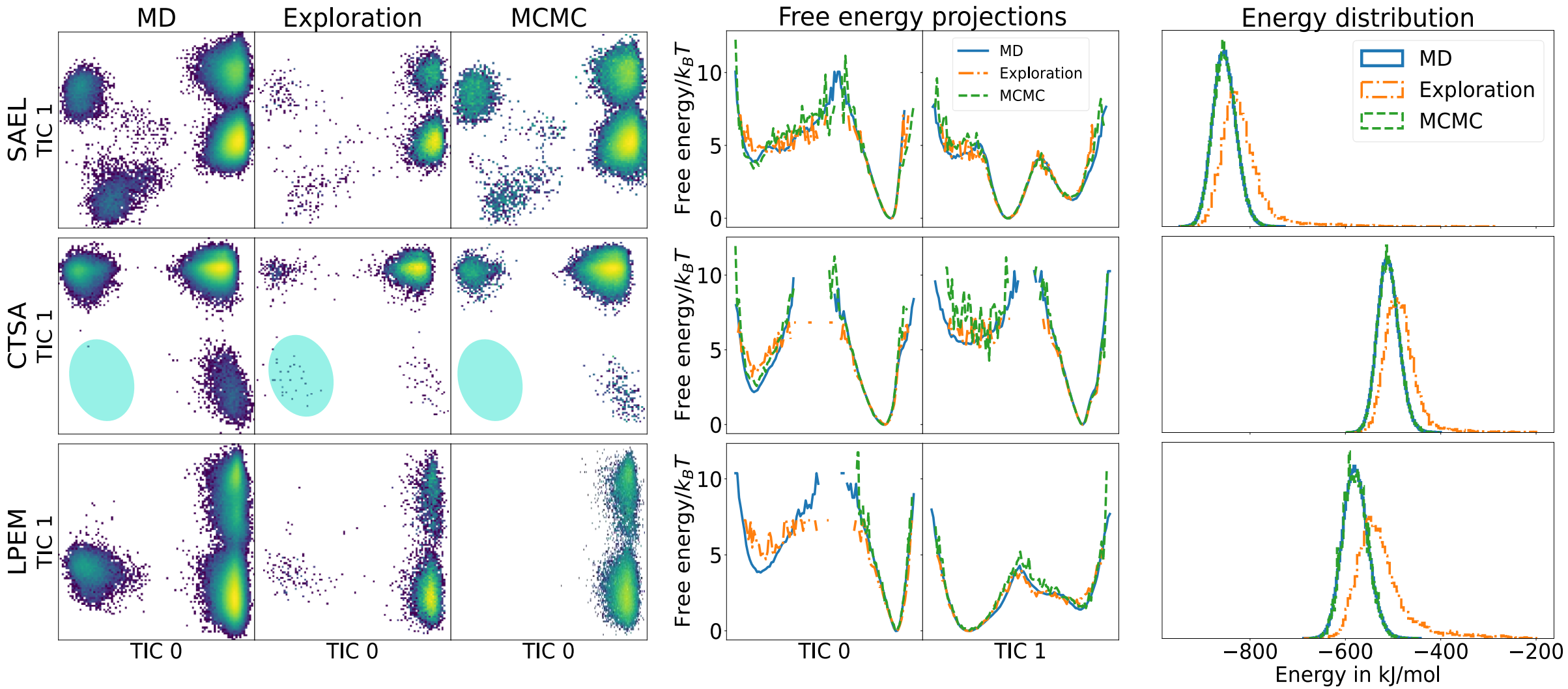
Wall-clock time speed-up – Timewarp MCMC

- Compare effective samples per second

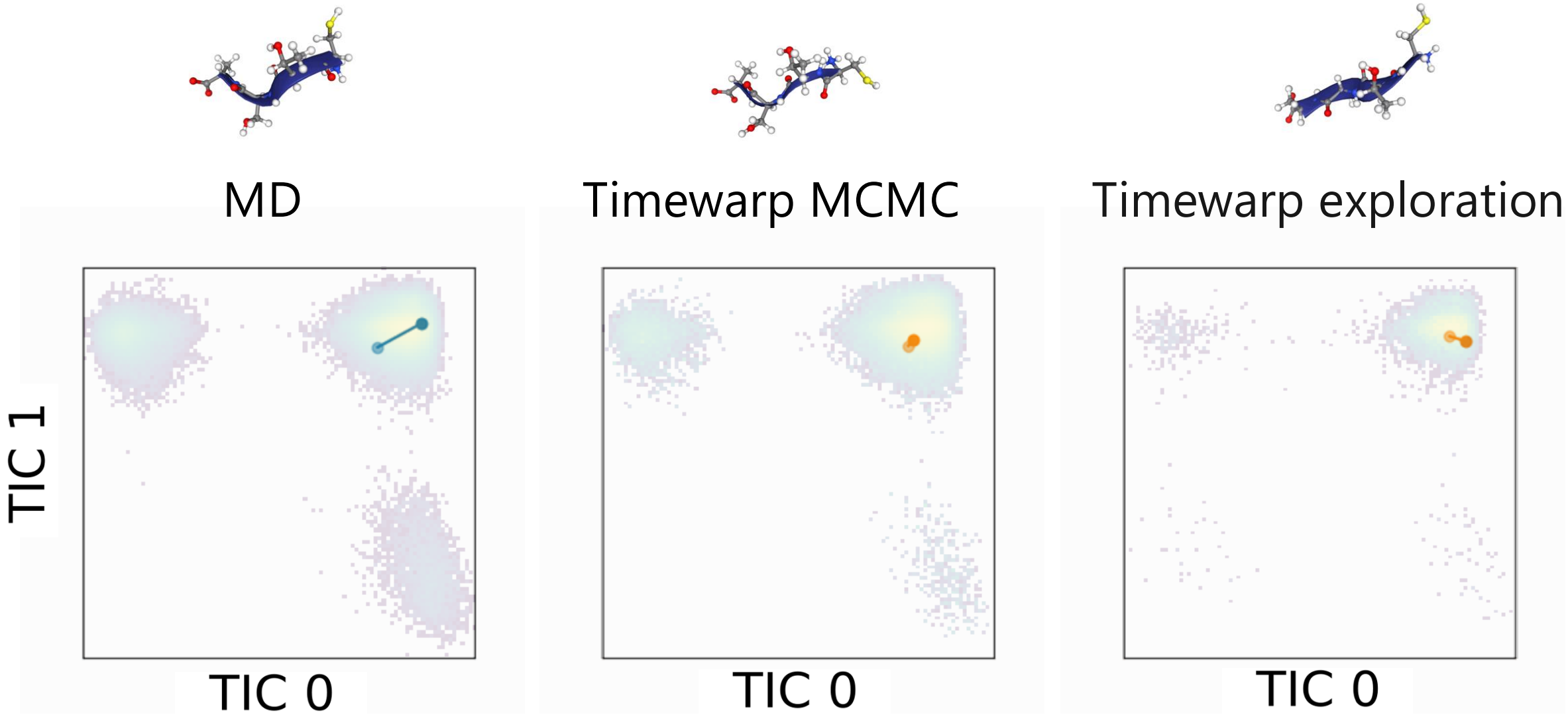




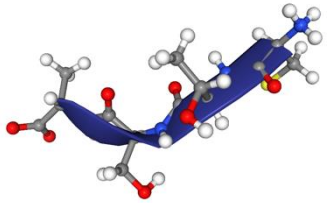
Exploration with the Timewarp model



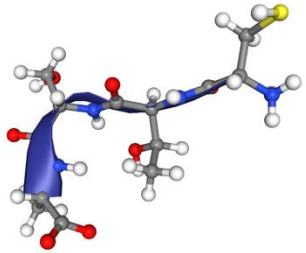
Exploration with the Timewarp model - CTSA



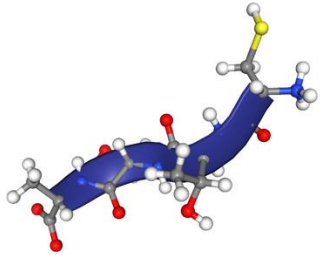
Exploration with the Timewarp model - CTSA



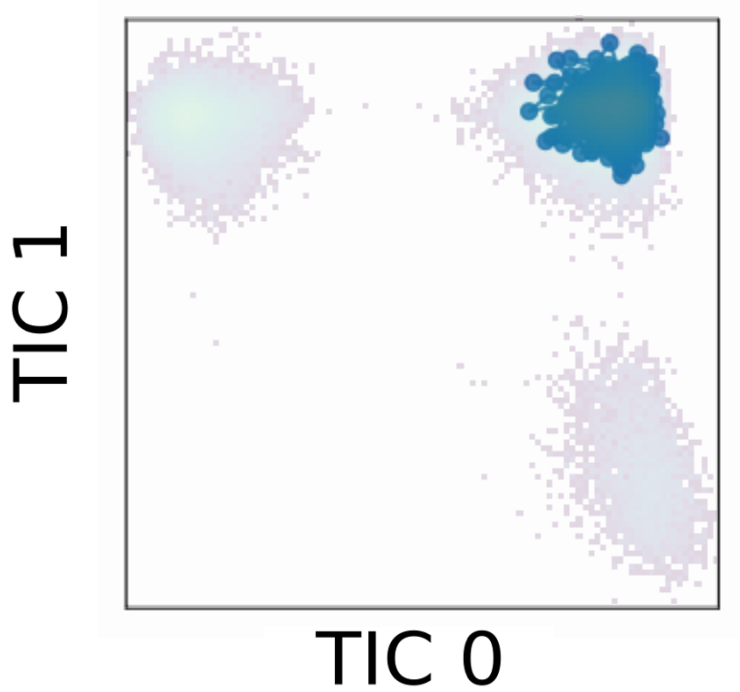
MD

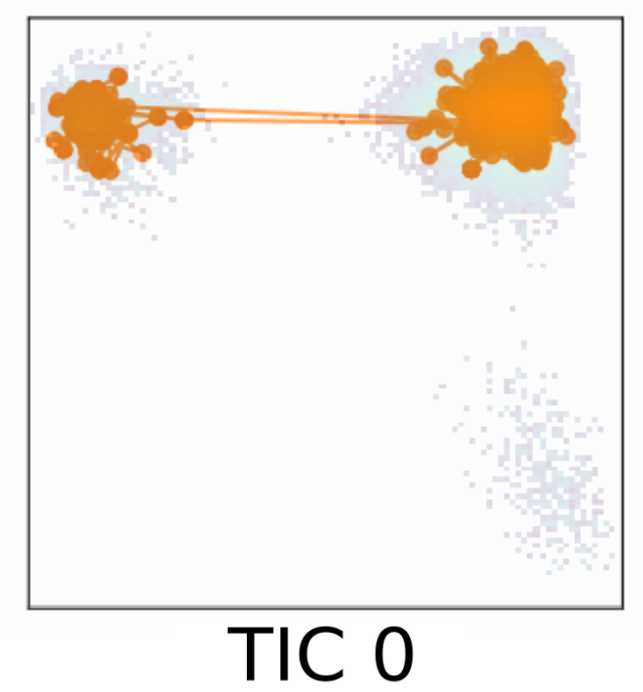


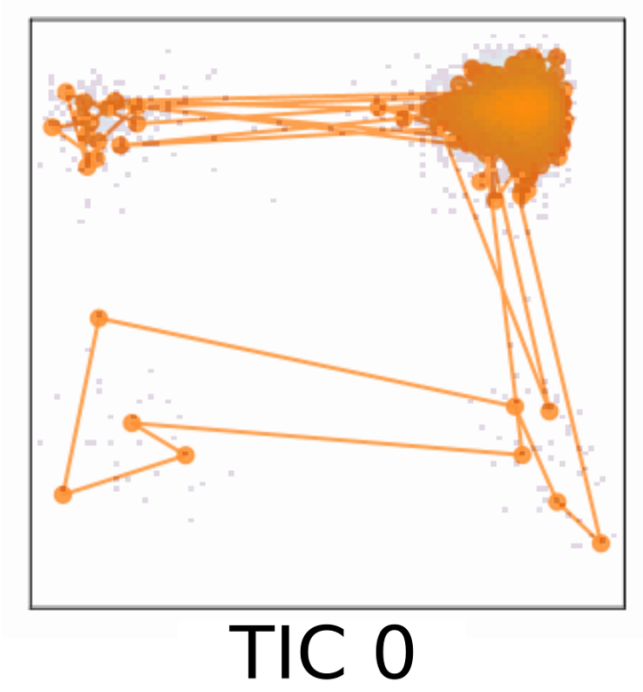
Timewarp MCMC



Timewarp exploration

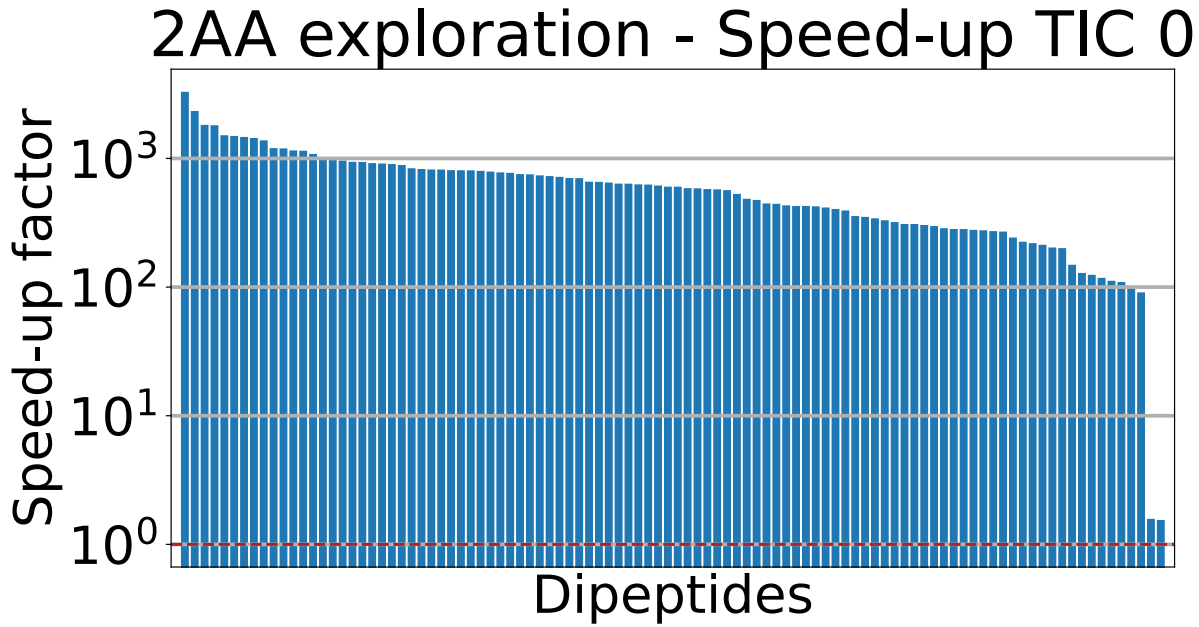


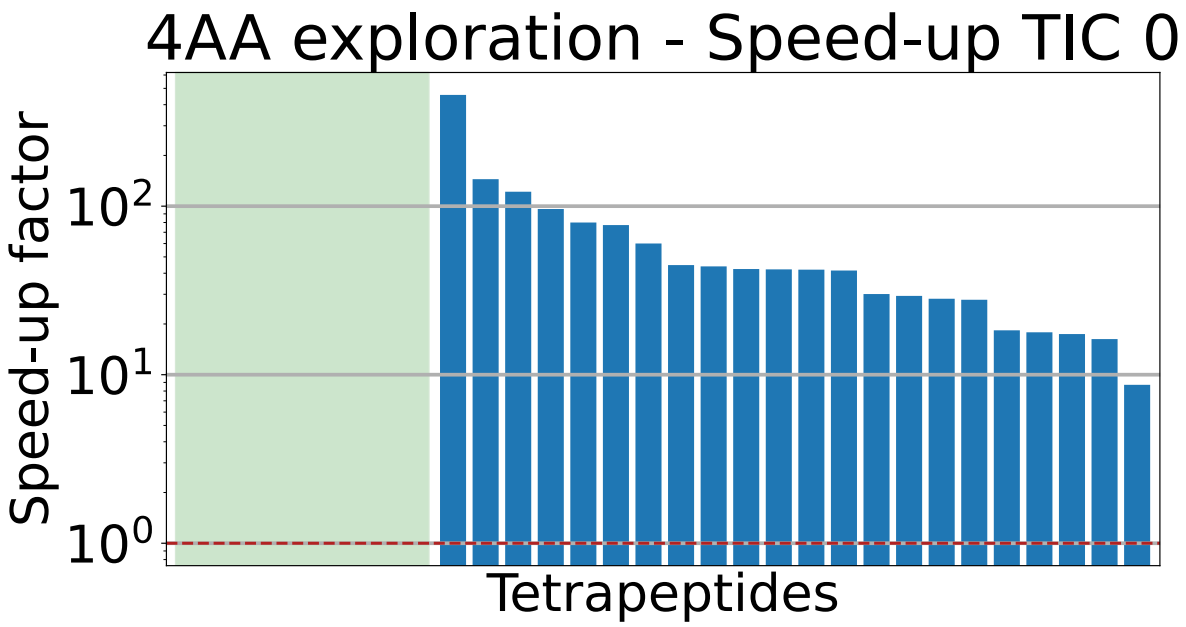




Wall-clock time speed-up – Timewarp exploration

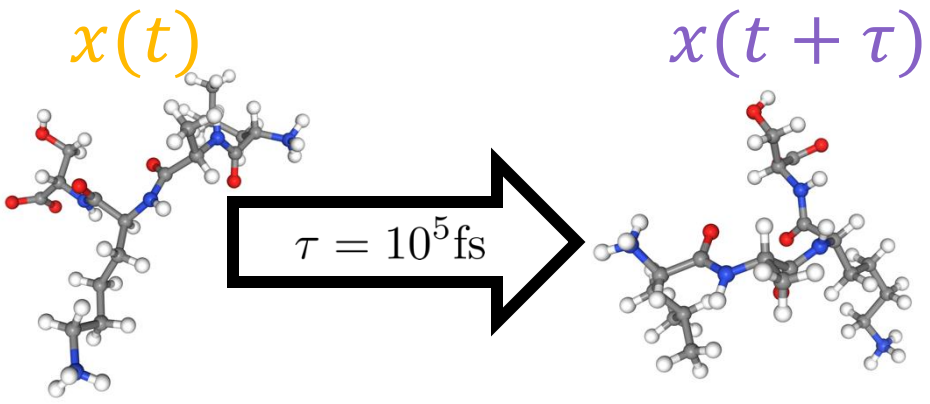
- Compare effective samples per second





Timewarp – Summary

- Transferable model that predicts future states of unseen peptides
- Two sampling algorithms
  1. Timewarp MCMC
     - Speed-up for dipeptides
     - Speed-up for minority of tetrapeptides
  2. Timewarp exploration
     - Speed-up for dipeptides and tetrapeptides
     - No longer asymptotically unbiased
     - Discovers metastable states that MD misses initially

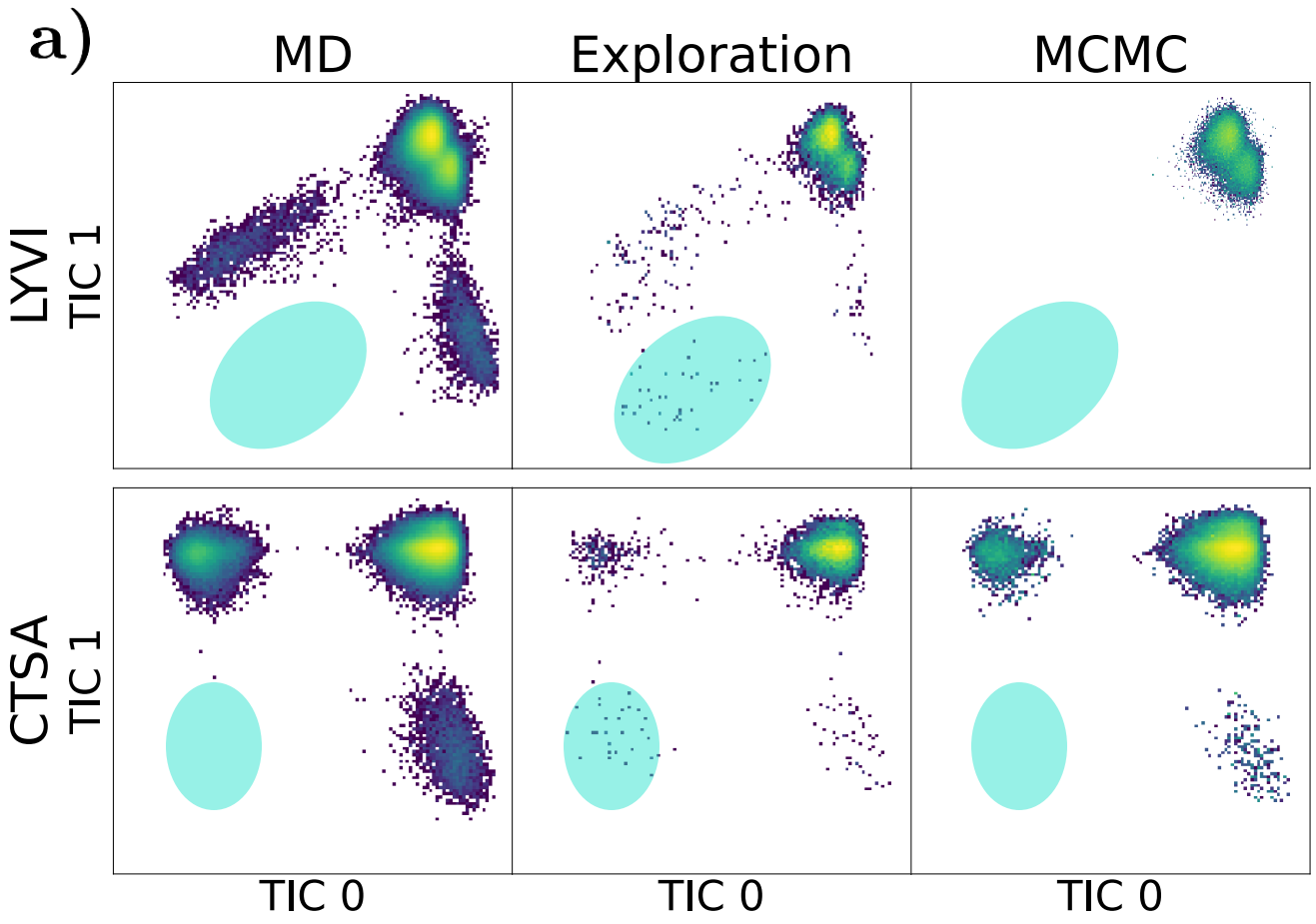


Microsoft

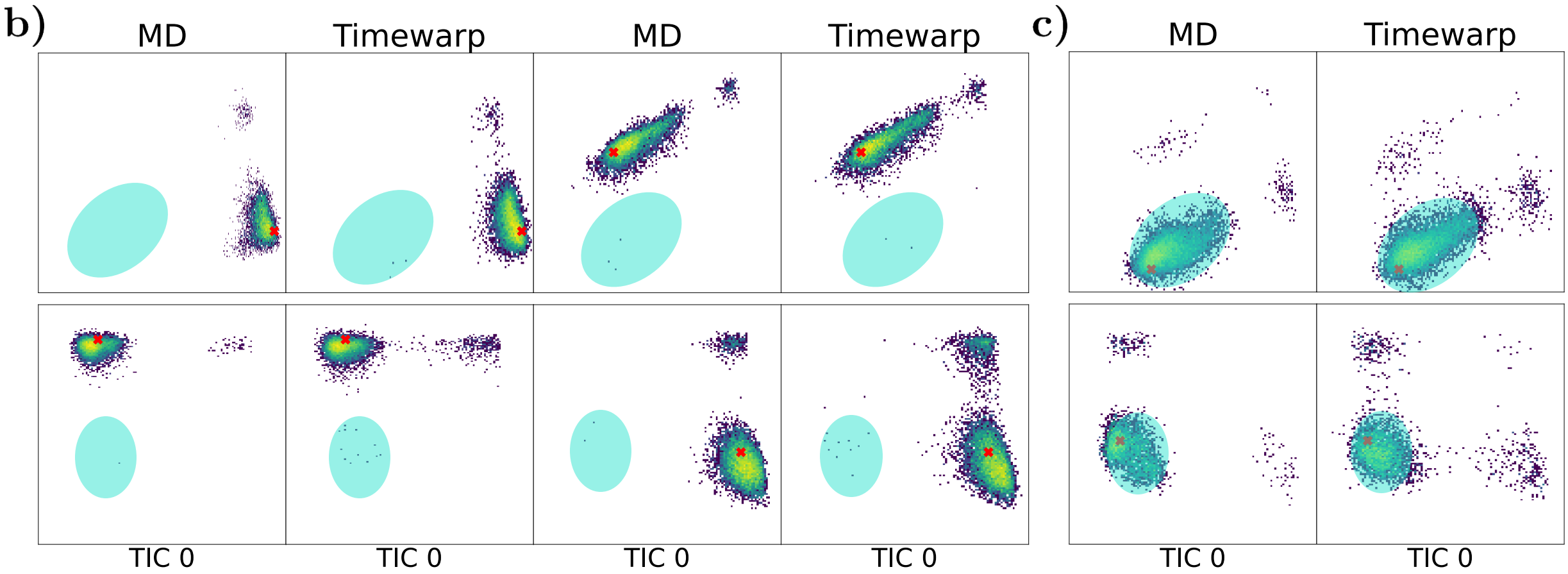
Thanks for listening!

Microsoft

Validation of new metastable states



Validation of new metastable states



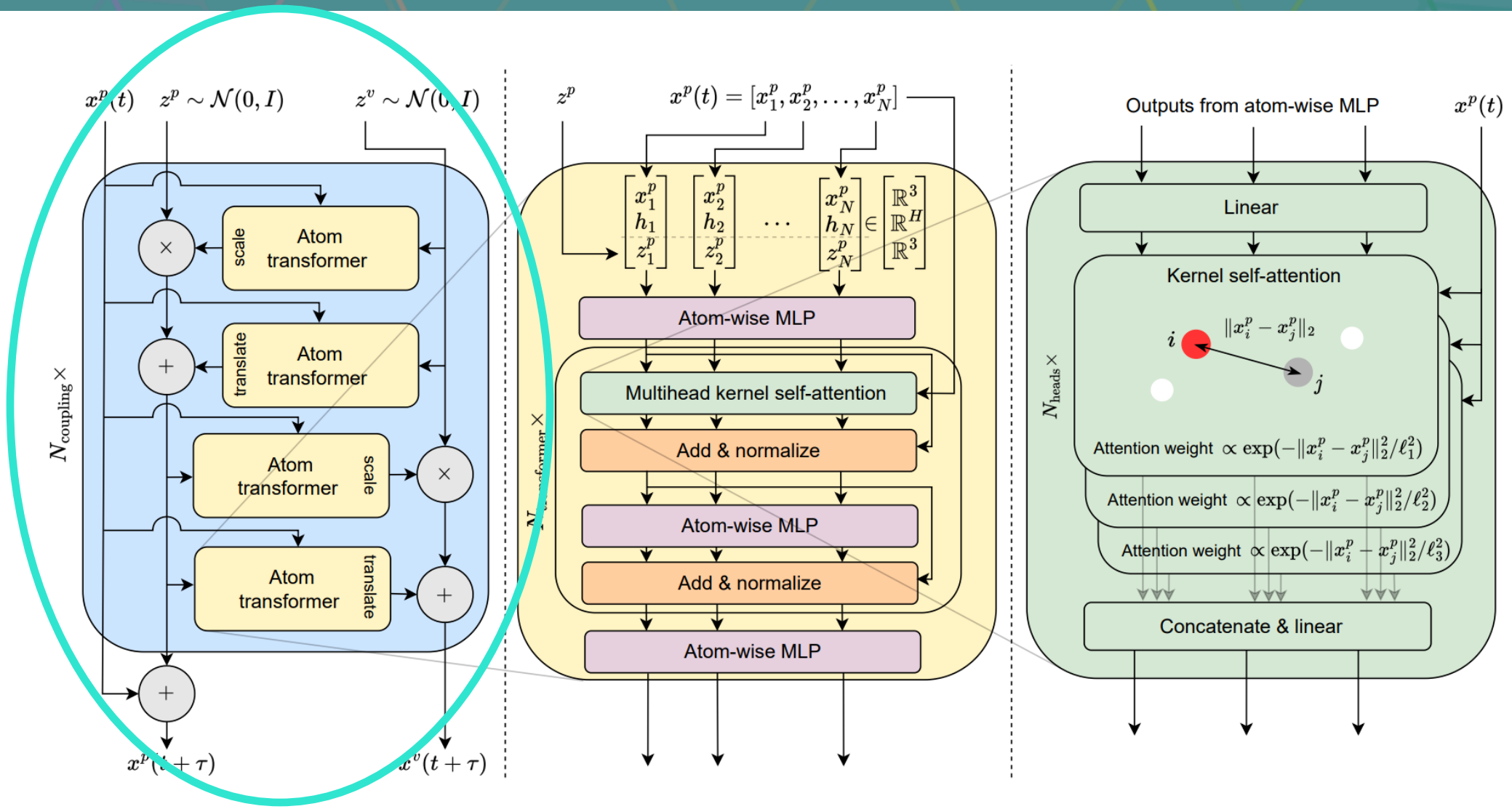
Boltzmann distribution

- Many MD applications boil down to sampling the *Boltzmann distribution*.
- Equilibrium distribution at a temperature T.

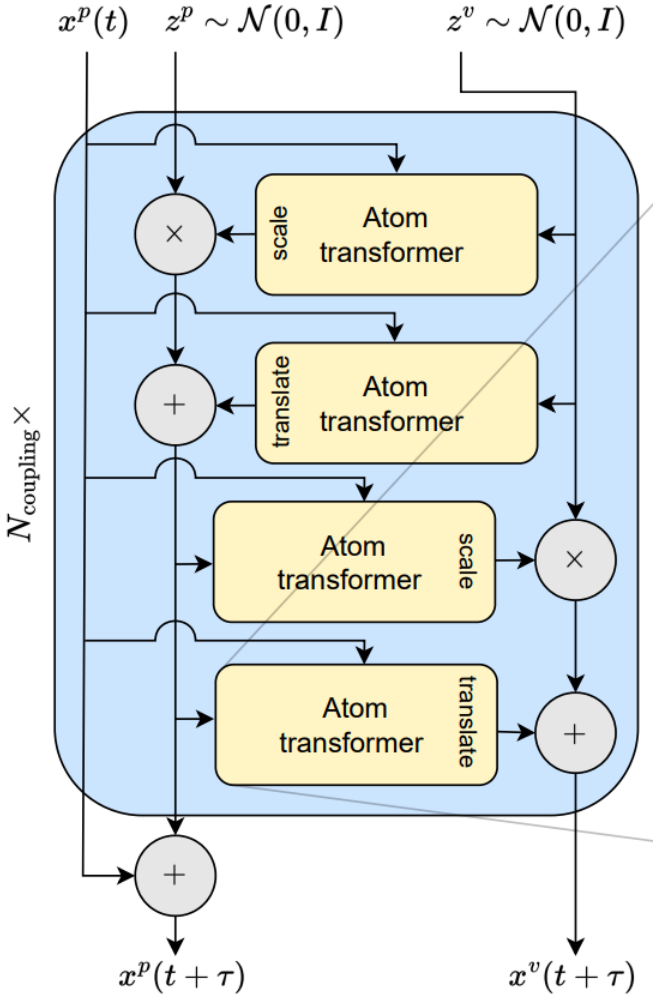
$$\mu(x) \propto \exp\left(-\frac{U(x)}{k_B T}\right).$$

- $U(x)$ is the potential energy function, k_B is Boltzmann's constant.
- Sampling i.i.d. is **intractable**.
- Long MD trajectories provide samples from $\mu(x)$ asymptotically.

Conditional flow architecture

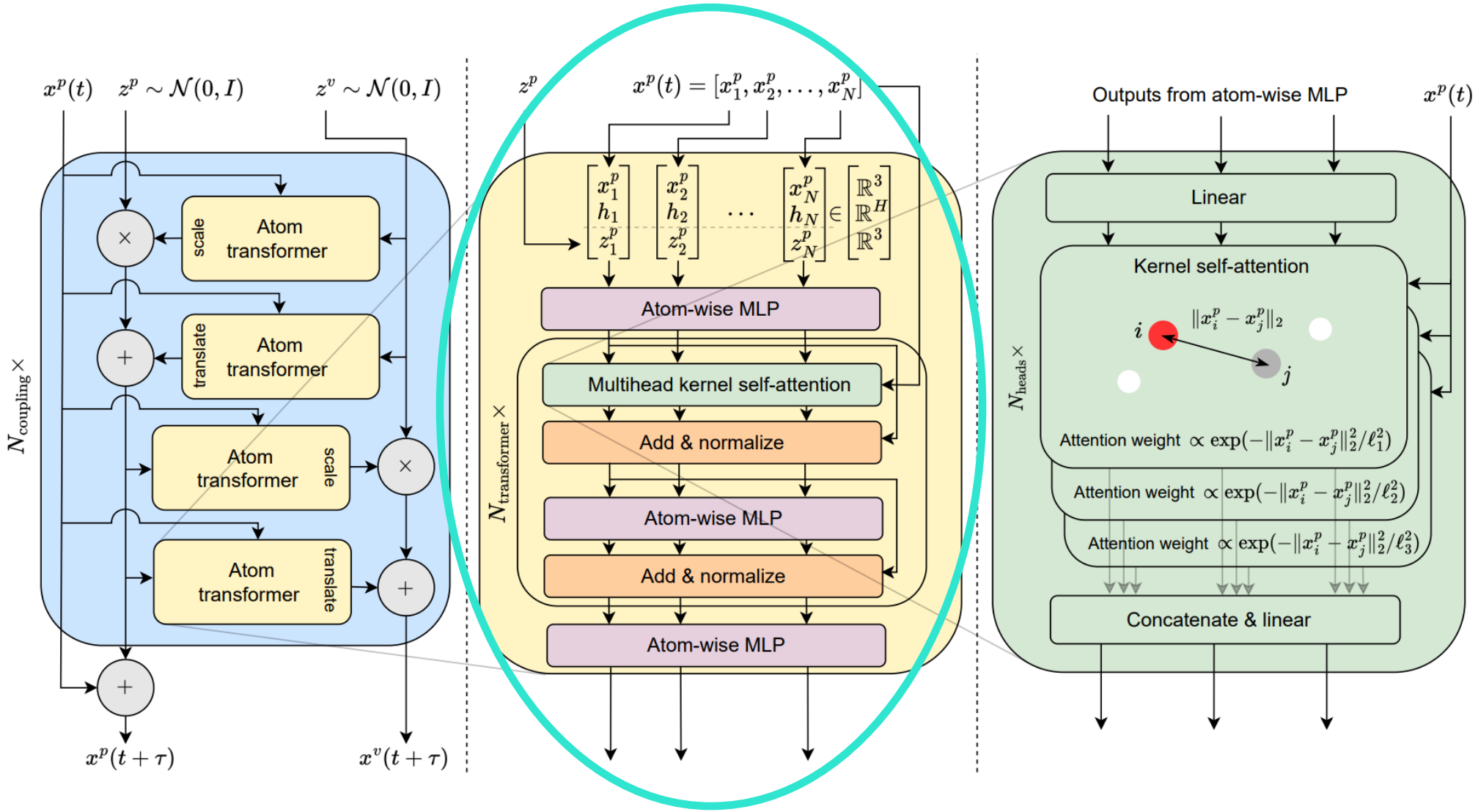


RealNVP

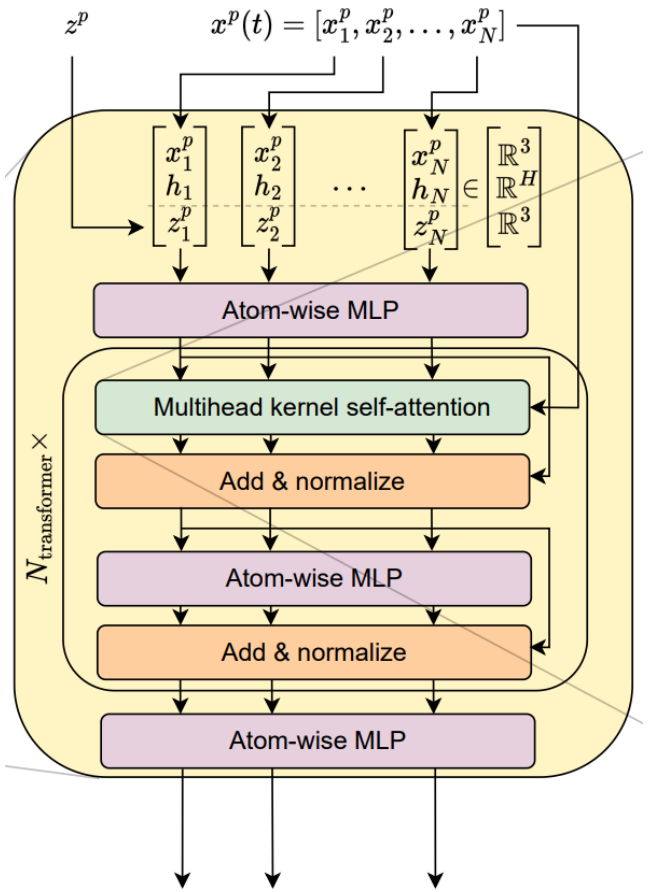


- RealNVP: affine transformation of some dimensions based on others.
- Transform z^p based on z^v and vice versa.
- Each transformation uses an atom transformer.
- Stack many transformations.
- Flow predicts the *change*, $x^p(t+\tau) - x^p(t)$.

Atom transformer

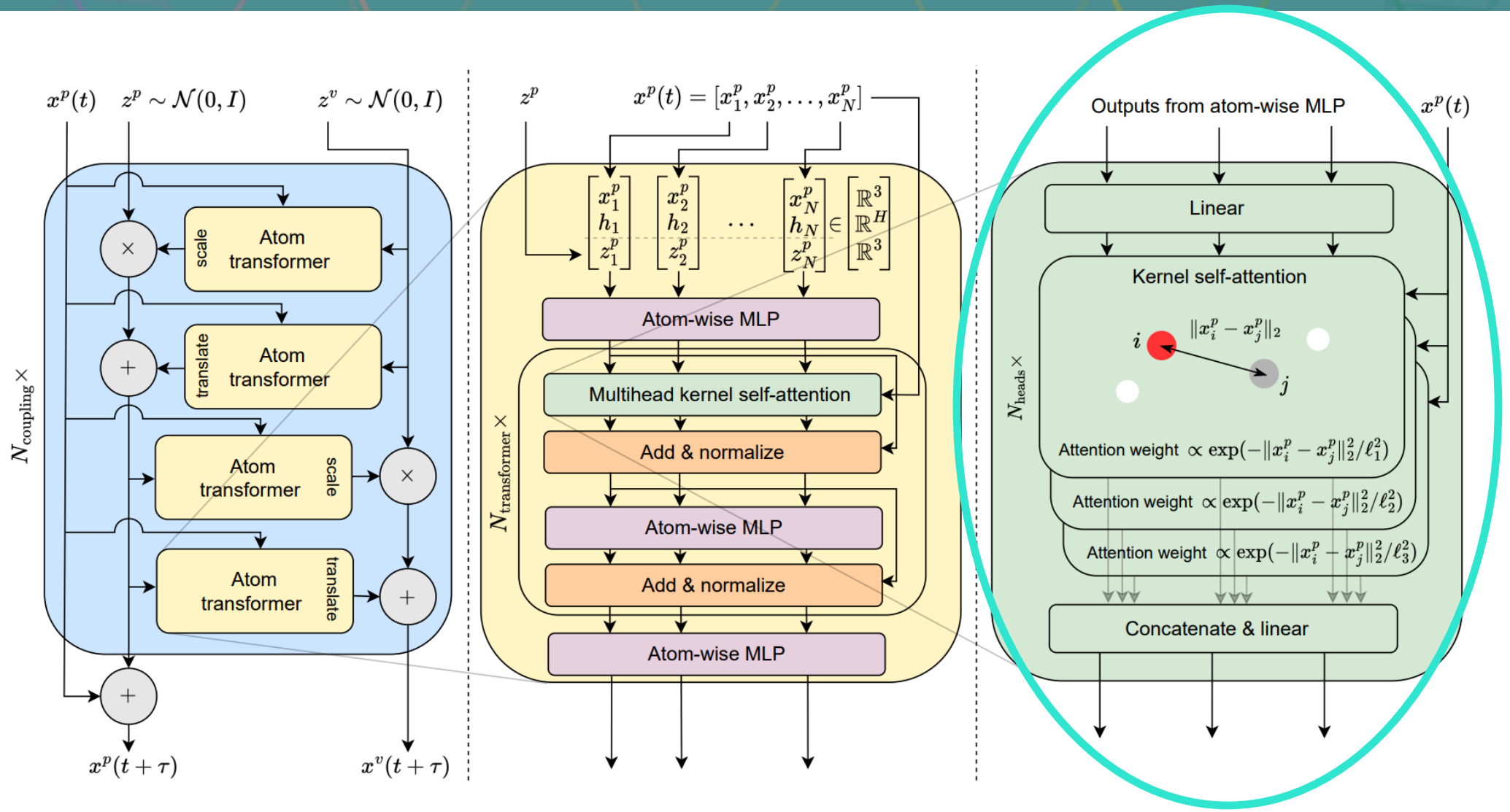


Atom transformer

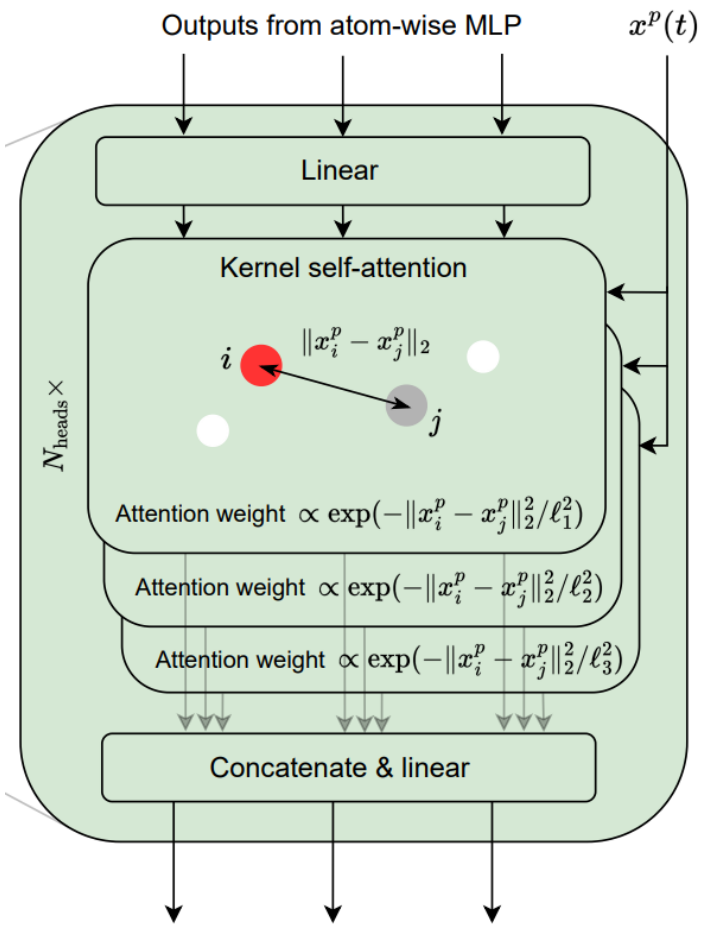


- Concatenate latent variables z, conditioning state $x(t)$ and atom feature embedding h.
- Pass through multiple transformer blocks.
- Use kernel self-attention.
- Output scale/translation factor of RealNVP.

Kernel self-attention

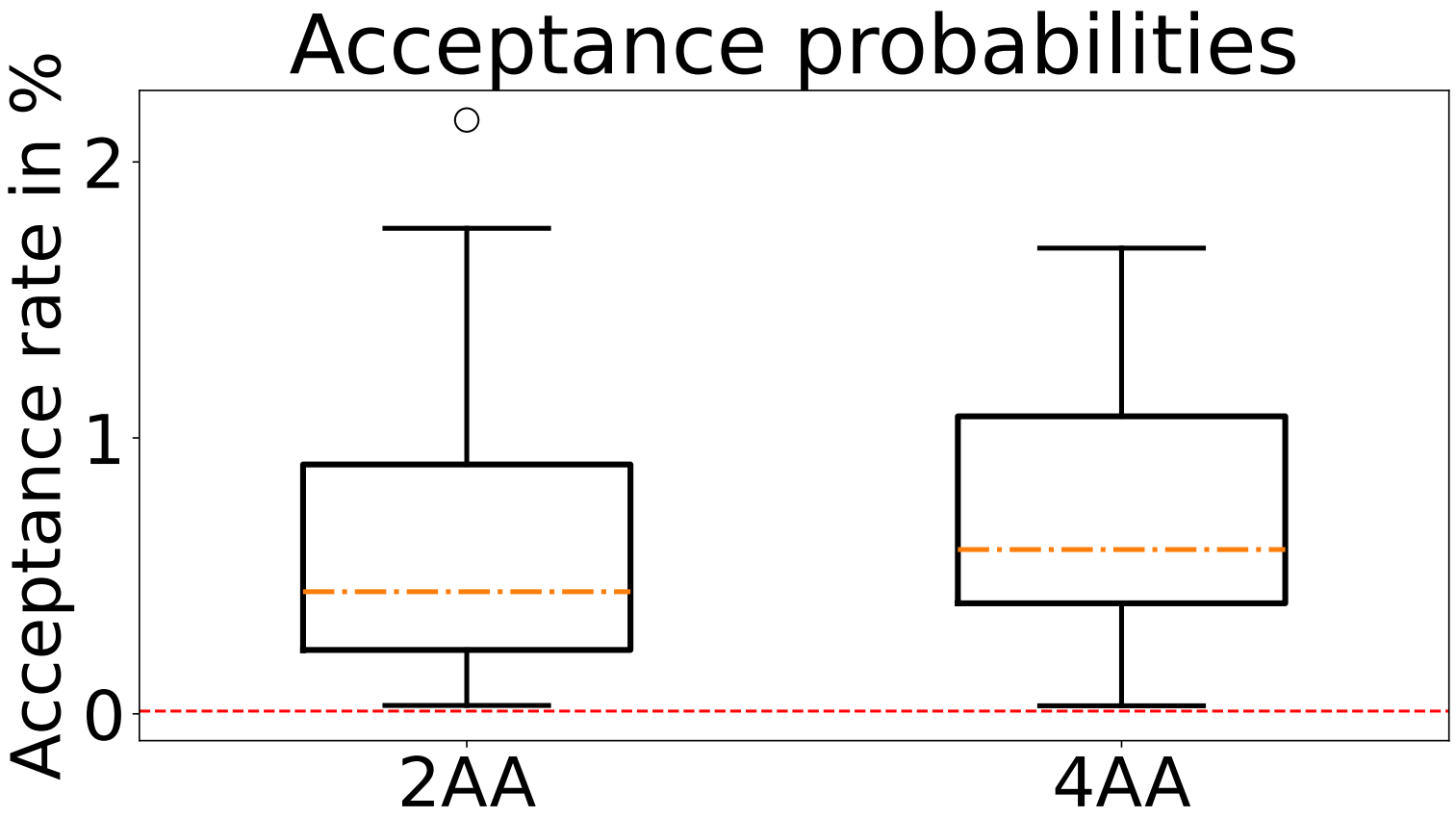


Kernel self-attention



- Gaussian RBF on interatomic distance to compute attention weights.
- Inductive bias: nearby atoms should have greater effect.
- Multihead version: each head uses a different RBF lengthscale.

Timewarp + MCMC



→ Sample proposals in parallel

Training times

Table 6. Timewarp training parameters

Dataset + training method	Batch size	No. of A-100s	Training time
AD — likelihood	256	1	1 week
AD — acceptance	64	1	2 days
2AA — likelihood	256	4	2 weeks
2AA — acceptance	256	4	4 days
4AA — likelihood	256	4	3 weeks

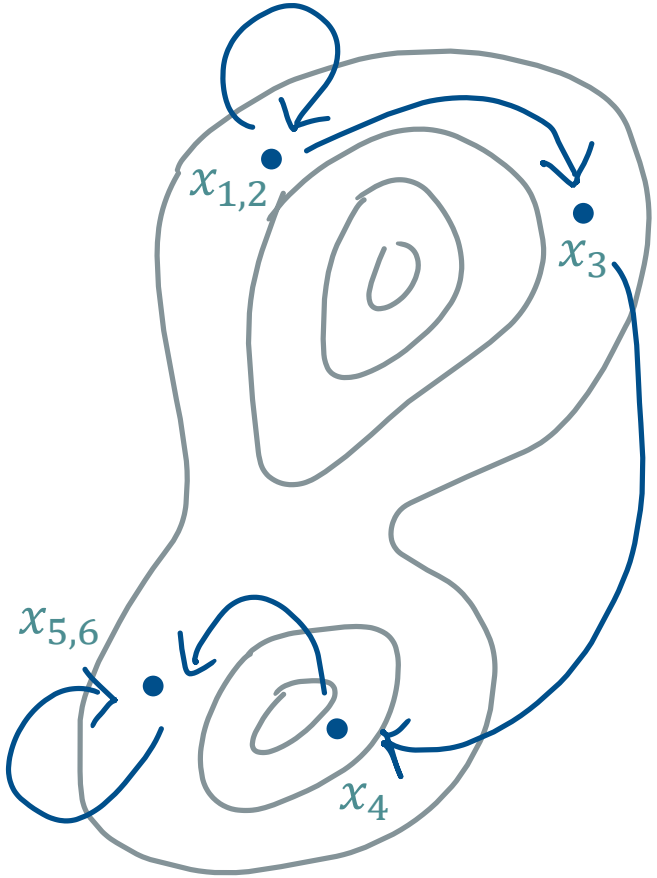
Timewarp MCMC algorithm

Metropolis Hastings proposal:

1. Sample $\tilde{x} \sim p_\theta(\cdot\,|x)$.
2. Compute acceptance ratio:

$$\alpha(x, \tilde{x}) = \min\left(1, \frac{\mu(\tilde{x})\, p_\theta(x|\tilde{x})}{\mu(x)\, p_\theta(\tilde{x}|x)}\right)$$

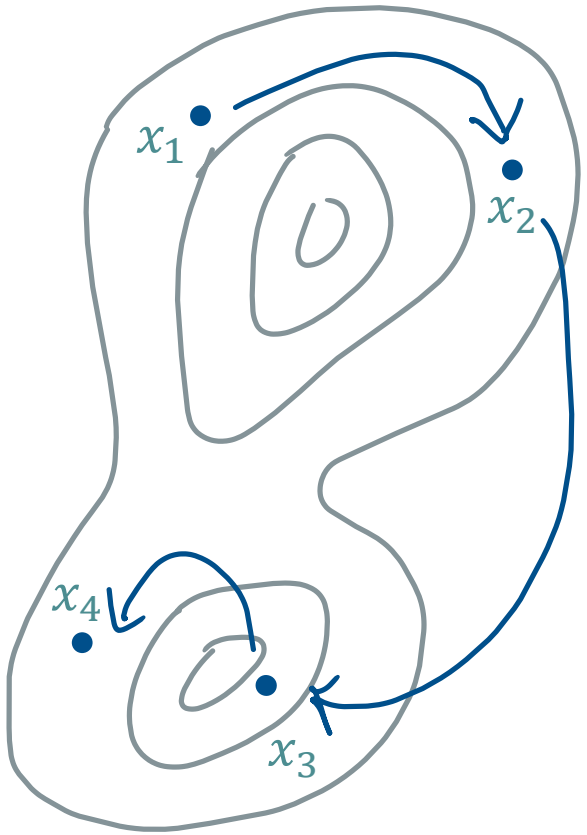
3. Accept $\tilde{x}$ with probability $\alpha(x, \tilde{x})$.



μ_{aug}

Timewarp exploration algorithm

- Timewarp MCMC can sometimes have very low acceptance.
- We also try *exploration* mode, where we accept all proposals.
- Biased samples, but can explore metastable states faster.



μ_{aug}

Augmented MCMC

- In practice we only care about positions of atoms, not velocities.
- Replace velocities with auxiliary variables $x^v \sim N(0, I)$.
- Joint augmented Boltzmann distribution:

$$\mu_{\text{aug}} \propto \exp\left(-\frac{U(x^p)}{k_B T}\right) N(x^v; 0, I).$$

- Target μ_{aug} with MCMC, then discard x^v.
- Why augment?
  - Allows more expressive distribution for x^p.
  - Easier to incorporate permutation symmetry.

Timewarp MCMC algorithm

$$x_m = (x_m^p, x_m^v)$$

1. Sample $\tilde{x} \sim p_\theta(\cdot \mid x_m^p)$
2. Compute acceptance ratios

$$\alpha(x_m, \tilde{x}) = \min\left(1, \frac{\mu_{\text{aug}}(\tilde{x})\, p_\theta(x_m \mid \tilde{x}^p)}{\mu_{\text{aug}}(x_m)\, p_\theta(\tilde{x} \mid x_m^p)}\right)$$

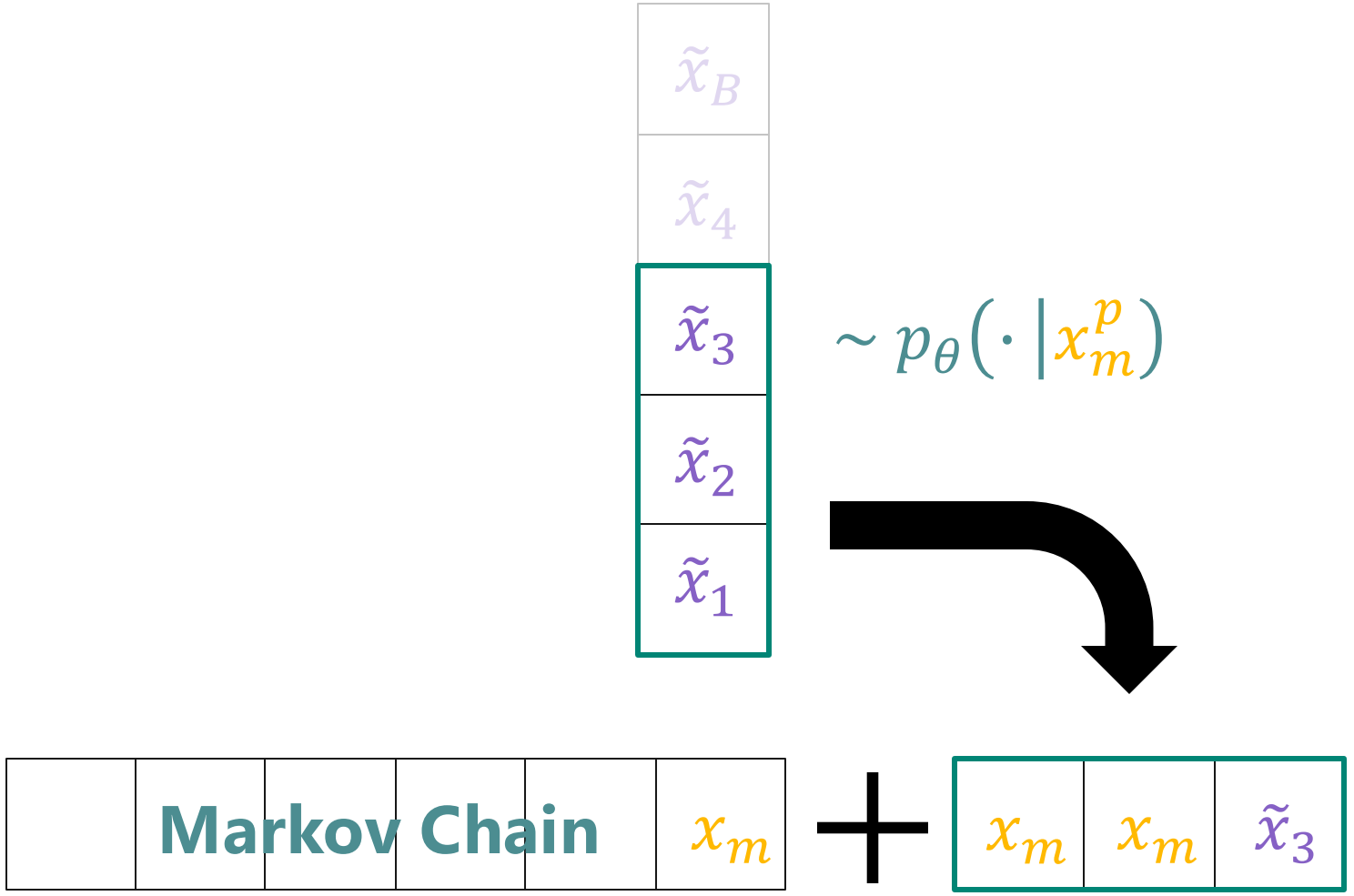
3. With probability $\alpha(x_m, \tilde{x})$ set $x_{m+1} = \tilde{x}$ else $x_{m+1} = x_m$.
4. Resample $x_{m+1}^v \sim N(0, I)$ (Gibbs update)

Batching the Timewarp MCMC algorithm

Algorithm 1 Timewarp MCMC with batched proposals

Require: Initial state $X_0 = (X_0^p, X_0^v)$, chain length M, proposal batch size B.

$m \leftarrow 0$
while $m < M$ **do**
  Sample $\tilde{X}_1, \ldots, \tilde{X}_B \sim p_\theta(\cdot \mid X_m^p)$ {Batch sample}
  for $b = 1, \ldots, B$ **do**
    $\epsilon \sim \mathcal{N}(0, I)$ {Resample auxiliary variables}
    $X_b \leftarrow (X_m^p, \epsilon)$
    Sample $I_b \sim \text{Bernoulli}(\alpha(X_b, \tilde{X}_b))$
  end for
  if $S := \{b : I_b = 1, 1 \leq b \leq B\} \neq \emptyset$ **then**
    $a = \min(S)$ {First accepted sample}
    $(X_{m+1}^p, \ldots, X_{m+a-1}^p) \leftarrow (X_m^p, \ldots, X_m^p)$
    $X_{m+a}^p \leftarrow \tilde{X}_a^p$
    $m \leftarrow m + a$
  else
    $(X_{m+1}^p, \ldots, X_{m+B}^p) \leftarrow (X_m^p, \ldots, X_m^p)$
    $m \leftarrow m + B$
  end if
end while
output $X_0^p, \ldots X_M^p$



Timewarp exploration algorithm

- Timewarp MCMC can sometimes have very low acceptance.
- We also try *exploration* mode, where we accept all proposals.
- Biased samples, but can explore metastable states faster.

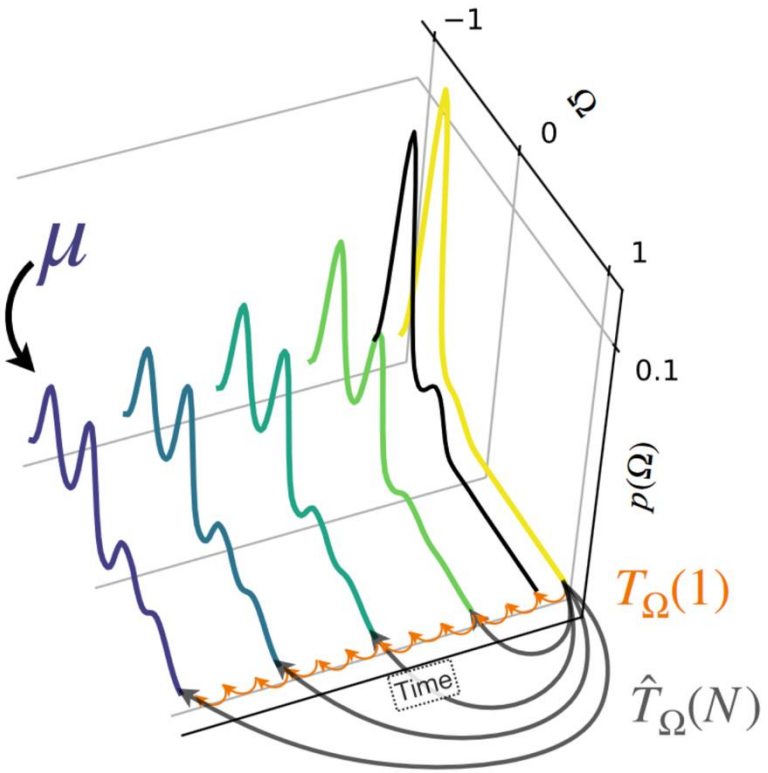
Algorithm 2 Fast exploration of the state space with Timewarp

Require: Initial state X_0^p, number of steps M, maximum allowed energy increase ΔU_{max}
  for $m = 0, \ldots, M$ **do**
    Sample $\tilde{X}_m^p \sim p_\theta(\cdot \mid X_m^p)$ {Sample from conditional flow}
    if $U(\tilde{X}_m^p) - U(X_m^p) < \Delta U_{\text{max}}$ **then**
      $X_{m+1}^p \leftarrow \tilde{X}_m^p$
    else
      $X_{m+1}^p \leftarrow X_m^p$ {Reject if energy change is too high}
    end if
  end for
output $X_0^p, \ldots X_M^p$

Related work

- **Implicit Transfer Operator Learning: Multiple Time-Resolution Surrogates for Molecular Dynamics**. *Schreiner et al. NeurIPS2023*

- Different time resolutions possible
- Accurate prediction of dynamic observables
- Not transferable yet



Future work

- Different flow architecture to scale to larger systems
- **SE(3) equivariant augmented coupling flows.** *Midgley et al. NeurIPS 2023*
- *Allow to include rotational symmetry*