PHYS 7810 - Computational statistical physics

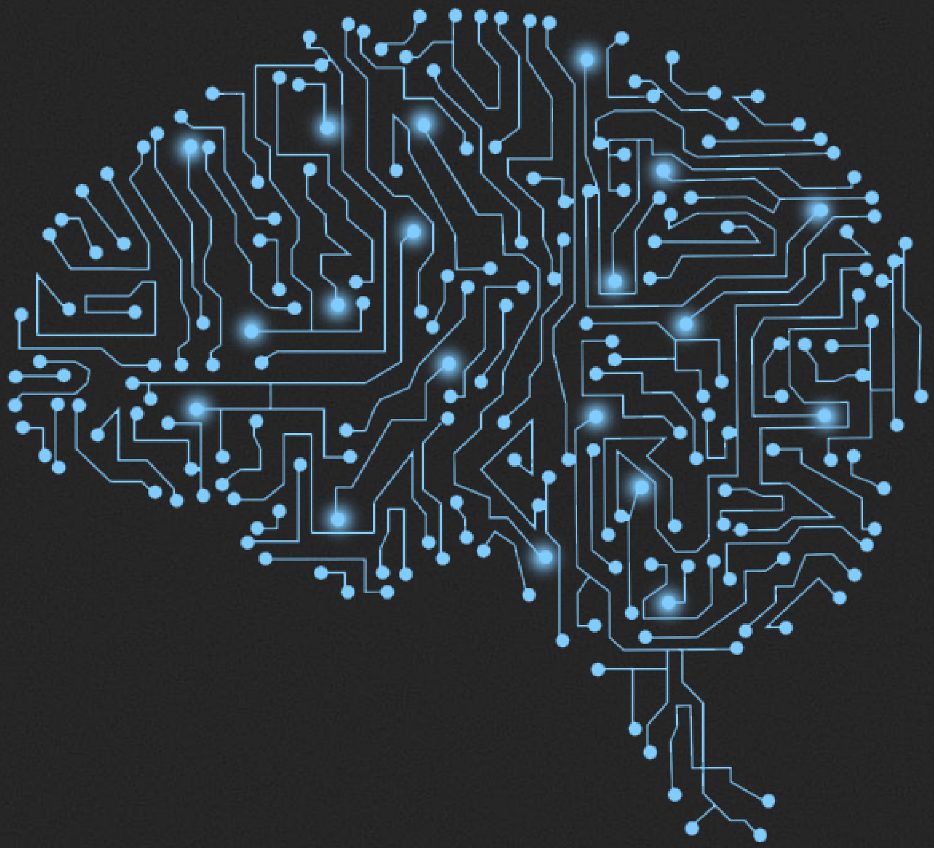
Boltzmann generators: Efficient sampling of equilibrium states of many-body system with deep learning

Final Project Presentation

Date: 04.23.2020

Lecturer: Dr. Matt Glaser

Presenter: Wei-Tse Hsu and Lenny Fobe



Be Boulder.

University of Colorado Boulder

Department of Chemical and Biological Engineering, University of Colorado Boulder

Outline

Introduction

- Challenges of sampling equilibrium states
- Architecture of a Boltzmann generators
- Flow-based generative models
- Real-valued non volume-preserving (RealNVP) transformation network
- Affine coupling layers in an NVP block
- Loss functions for training

Applications of Boltzmann generators

- Systems of interest
  - ✓ Double-well potential
  - ✓ Muller Brown potential
  - ✓ Dimer in Lennard-Jones bath
- Acquisition of training datasets
- Training of the inverse generator
- Free energy calculations

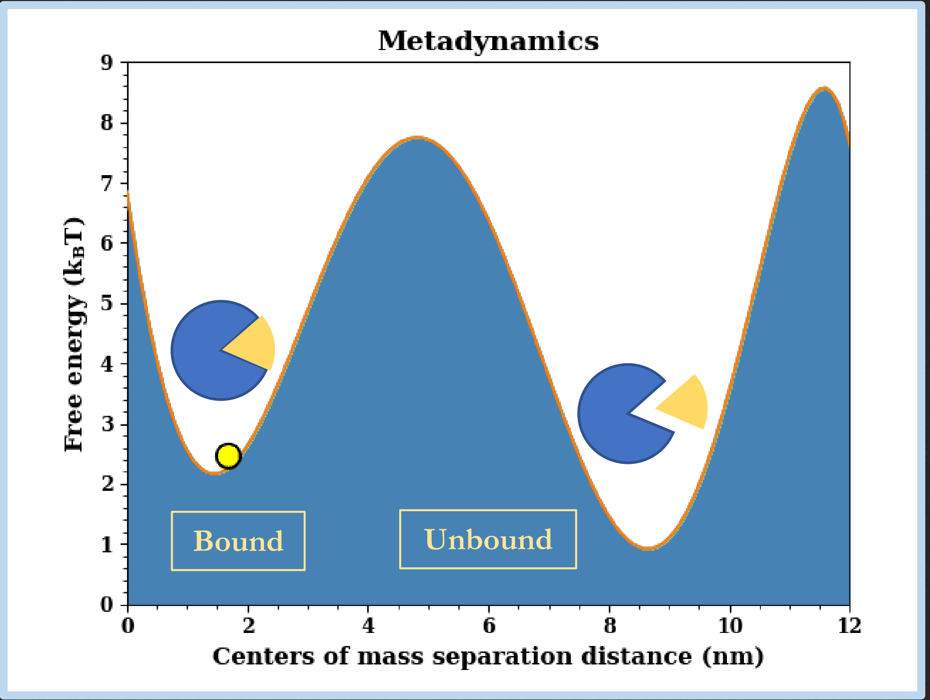
Outline

Conclusion

- Broader significance of Boltzmann Generators in modeling community
- Work expectation for the following week
  - ✓ Correct KL loss term
  - ✓ Dimer in LJ bath
- Future Applications
  - ✓ Protein/Ligand systems
  - ✓ Protein folding application

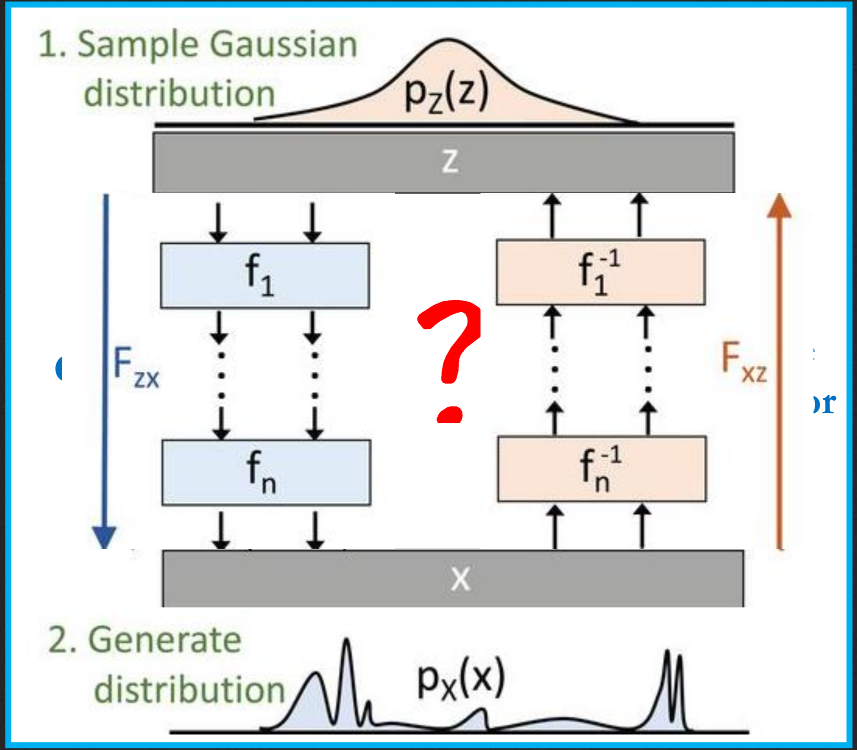
The usefulness of molecular simulations (MC or MD simulations) is limited in systems with slow kinetics

- Large free energy barriers between metastable states cause kinetic bottlenecks.
- Advanced sampling methods
  - ✓ Metadynamics and its variations
  - ✓ Umbrella sampling
  - ✓ Replica exchange, or expanded ensemble
- Boltzmann generators



A Boltzmann generator is trained to learn the transformation between probability distributions in the configuration and the latent space

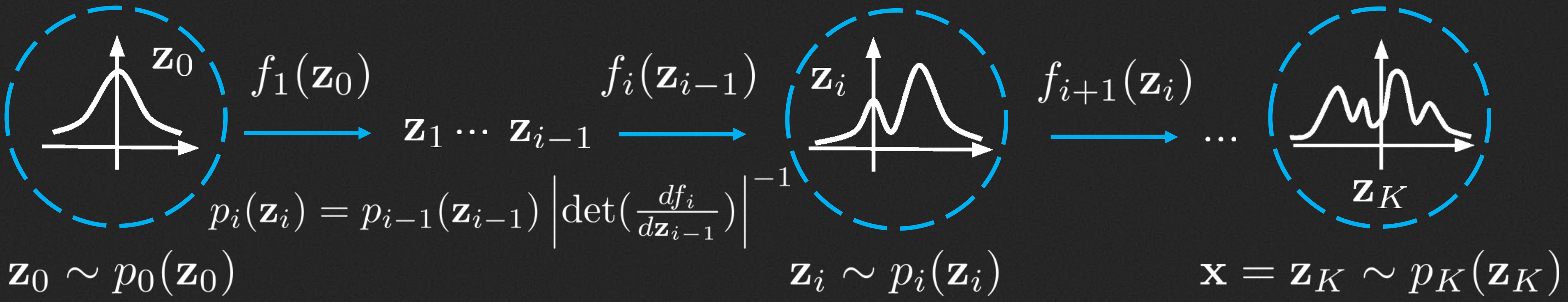
- Goal: approximate the Boltzmann distribution
- Procedures to generate $p_x(x)$
  - ✓ Taking the samples in the real space as the input
  - ✓ Train the inverse generator F_{xz} with samples x
  - ✓ Draw samples from the latent space
  - ✓ Map the latent samples back to the configuration space



Boltzmann generators are a application of flow-based generative models

- F_{xz} and F_{zx} are composed of "a flow of" non volume-persevering (NVP) blocks.
- The volume of space is changed between different spaces within the flow.

$$p_x(\mathbf{x}) = p_z(\mathbf{z}) \cdot |\det J_{zx}(\mathbf{x})|^{-1} \Rightarrow \log p_x(\mathbf{x}) = \log p_z(F_{xz}(\mathbf{x})) + \log R_{xz}$$



The affine coupling layers in each real NVP block ensure efficient computations of G^{-1} and $det(J_G)$

$$\log p_x(\mathbf{x}) = \underbrace{\log p_z(F_{xz}(\mathbf{x}))}_{\text{Inverse Generator } G^{-1}} + \underbrace{\log R_{xz}}_{\text{Log of } det(J_G)}$$

- Each affine coupling layer performs a scale-and-shift transformation

$$f_{xz}(\mathbf{x}_{1:d}, \mathbf{x}_{d+1:D}) : \begin{cases} \mathbf{z}_{1:d} = \mathbf{x}_{1:d} & \text{(Channel 1: copy)} \\ \mathbf{z}_{d+1:D} = \mathbf{x}_{d+1:D} \odot \exp(S(\mathbf{x}_{1:d}; \theta)) + T(\mathbf{x}_{1:d}; \theta) & \text{(Channel 2: transform)} \end{cases}$$

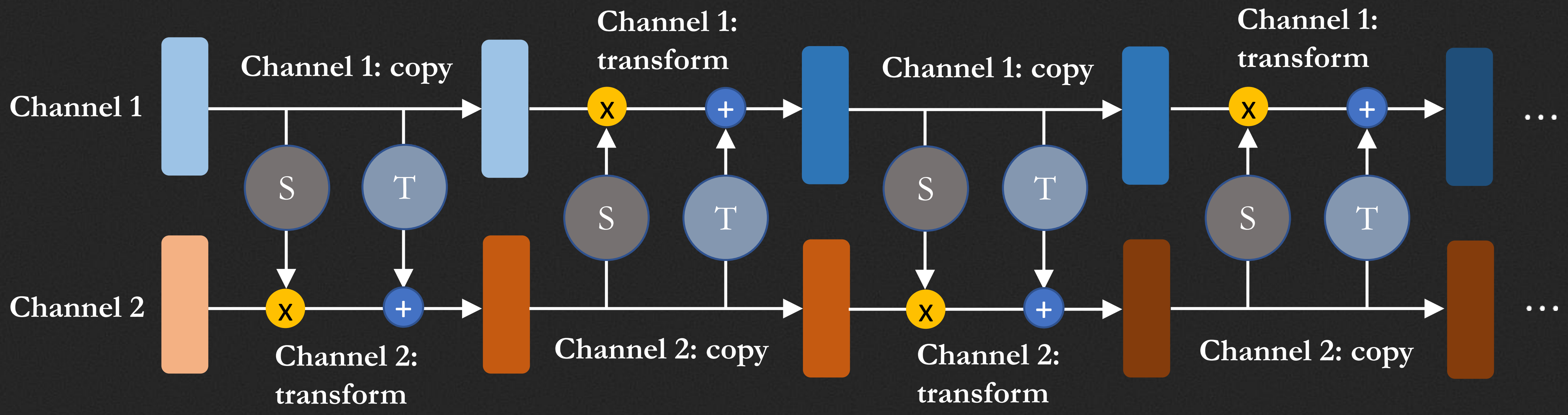
$$\mathbf{J}_{xz} = \begin{array}{c} \begin{array}{cc} \overbrace{\hphantom{\frac{\partial \mathbf{z}_{d+1:D}}{\partial \mathbf{x}_{1:d}}}}^{1:d} & \overbrace{\hphantom{\mathrm{diag}(\exp(S(\mathbf{x}_{1:d})))}}^{(d+1):D} \end{array} \\ \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_{d\times(D-d)} \\ \frac{\partial \mathbf{z}_{d+1:D}}{\partial \mathbf{x}_{1:d}} & \mathrm{diag}(\exp(S(\mathbf{x}_{1:d}))) \end{bmatrix} \end{array} \Rightarrow \log R_{xz} = \sum_{j=1}^{D-d} S(\mathbf{x}_{1:d})_j$$

The ordering of the two channels in a layer is reversed after each iteration

- Each affine coupling layer performs a scale-and-shift transformation

$$f_{xz}(\mathbf{x}_{1:d}, \mathbf{x}_{d+1:D}) : \begin{cases} \mathbf{z}_{1:d} = \mathbf{x}_{1:d} & \text{(Channel 1: copy)} \\ \mathbf{z}_{d+1:D} = \mathbf{x}_{d+1:D} \odot \exp(S(\mathbf{x}_{1:d};\theta)) + T(\mathbf{x}_{1:d};\theta) & \end{cases}$$

(Channel 2: transform)



Minimization of the loss functions ensure the robustness of the network

- Some important notations
  - ✓ $\mu_z(z)$: Gaussian prior distribution in the latent space
  - ✓ $\mu_x(x)$: Boltzmann distribution in configuration space
  - ✓ $q_z(z)$: Distribution generated by the inverse generator F_{xz}
  - ✓ $q_x(x)$: Distribution generated by the generated F_{zx}
- Maximum likelihood loss J_{ML}
  - ✓ Minimizing J_{ML} maximizes the likelihood of the configuration samples in Gaussian prior density the latent space
  - ✓ $J_{ML} = \mathbf{E}_{\mathbf{x}\sim\mu(\mathbf{x})}\left[\frac{1}{2\sigma^2}\|F_{xz}(\mathbf{x};\theta)\|^2 - \log R_{xz}(\mathbf{x};\theta)\right]$

KL loss and RC loss facilitate the sampling at different low energy states and high energy states, respectively

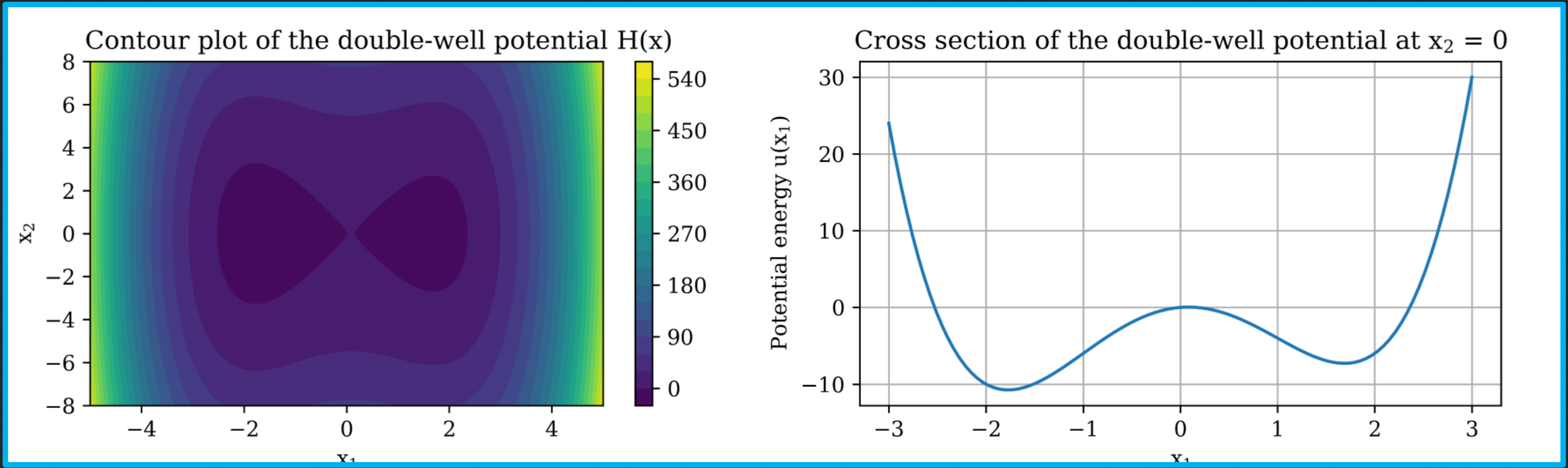
- Kullback-Leibler loss J_{KL}
  - ✓ KL divergence $D_{KL}(p||q)$ measures the distance between distribution p and q
  - ✓ In the latent space:

    $J_{KL}(\mu_z \| q_z) = \mathbf{E}_{\mathbf{z}\sim\mu(\mathbf{z})}\left[u(F_{zx}(\mathbf{z};\theta)) - \log R_{zx}(\mathbf{z};\theta)\right]$
- Reaction coordinate loss J_{RC}
  - ✓ Promotes sampling of high-energy states in a specific direction of RC
  - ✓ $J_{RC} = \int p\left(r(\mathbf{x})\right) \log p\left(r(\mathbf{x})\right) dr(\mathbf{x}) = \mathbf{E}_{\mathbf{x}\sim q_x(\mathbf{x})} \log p(r(\mathbf{x}))$
- Total loss:

$$J = w_{ML} J_{ML} + w_{KL} J_{KL} + w_{RC} J_{RC}$$

We started with double-well potential to construct Boltzmann generators

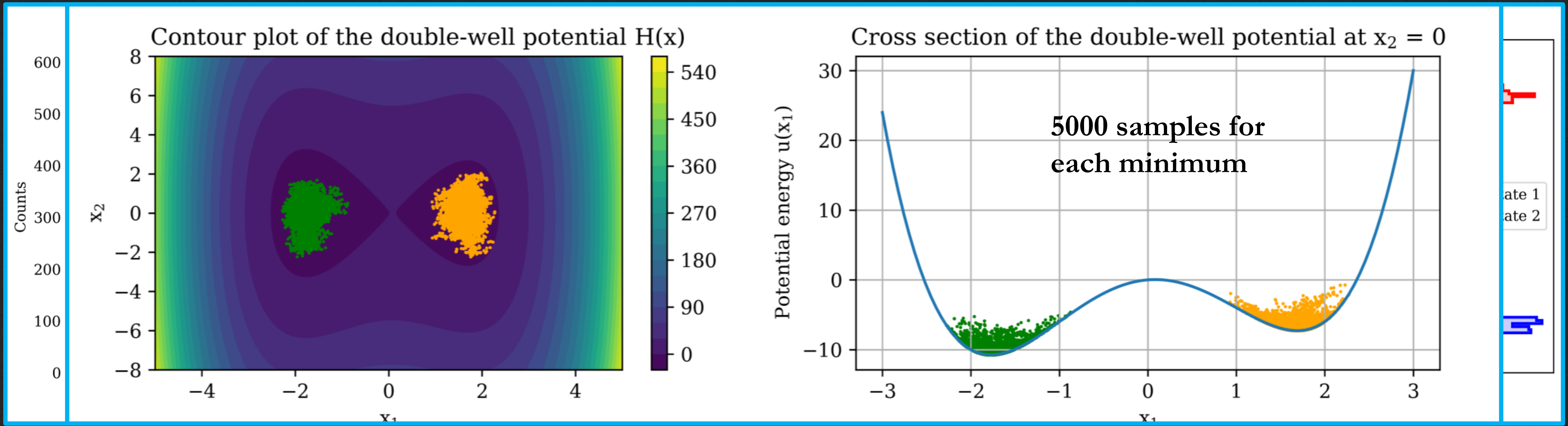
- Goals for the system:
  - ✓ Generate samples at both metastable states and compute the free energy difference.
  - ✓ Plot the free energy profile as a function of the reaction coordinate.
  - ✓ Assess the Boltzmann generators trained on different loss functions.



We extracted configuration samples from each energy minima using Monte Carlo (MC) simulations

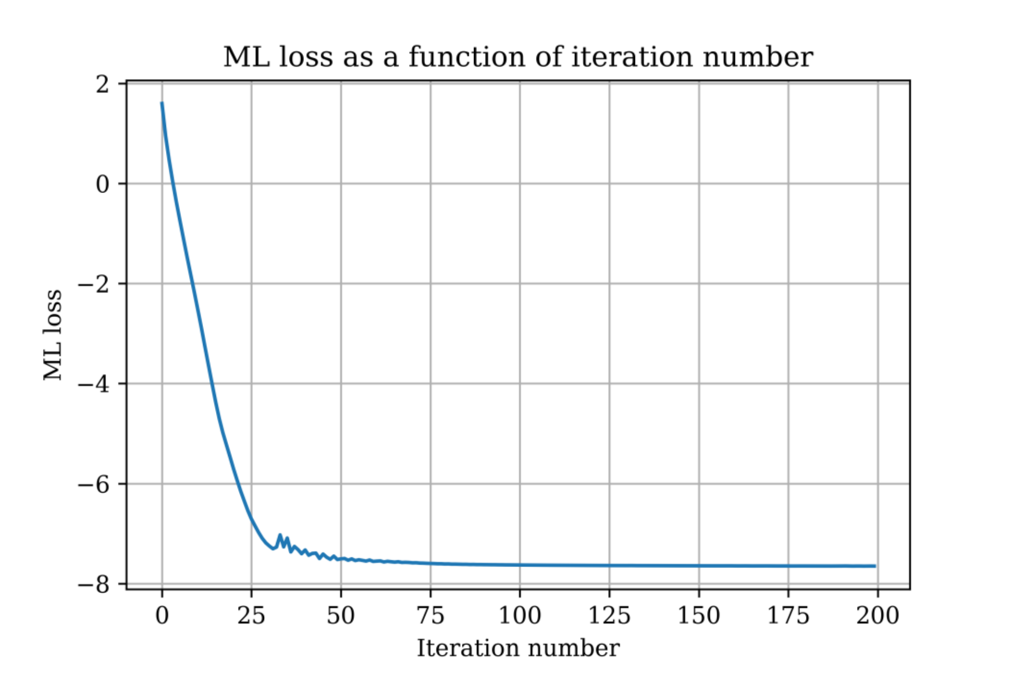
$$u(\mathbf{x}) = u(x_1, x_2) = ax_1^4 - bx_1^2 + cx_1 + dx_2^2$$

$$(a,\ b,\ c,\ d) = (1,\ 6,\ 1,\ 1)$$



A simple set of parameters is able to converge the ML loss with in a smaller number iterations

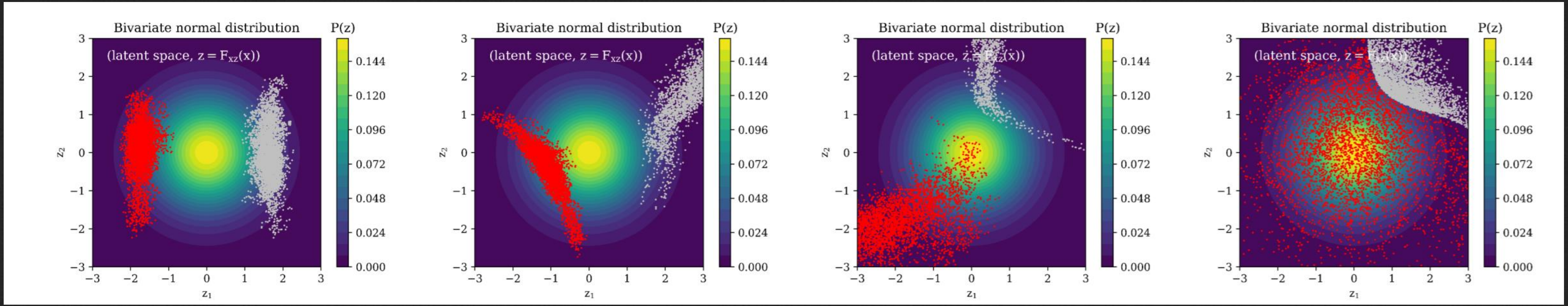
- Parameters
  - ✓ 3 NVP blocks
  - ✓ 2 coupling layers per each NVP block
  - ✓ 3 NN layers per each transformation fn
  - ✓ 100 nodes per each layer
  - ✓ Activation functions: ReLU and tanh
  - ✓ Optimizer: Adam, with LR as 0.001
  - ✓ Batch size: 1000 samples
  - ✓ Number of iterations: 200



Boltzmann generators trained on ML loss function was able to generate reasonable configurations in the real space

The process of training the inverse generator $F_{xz}(x)$

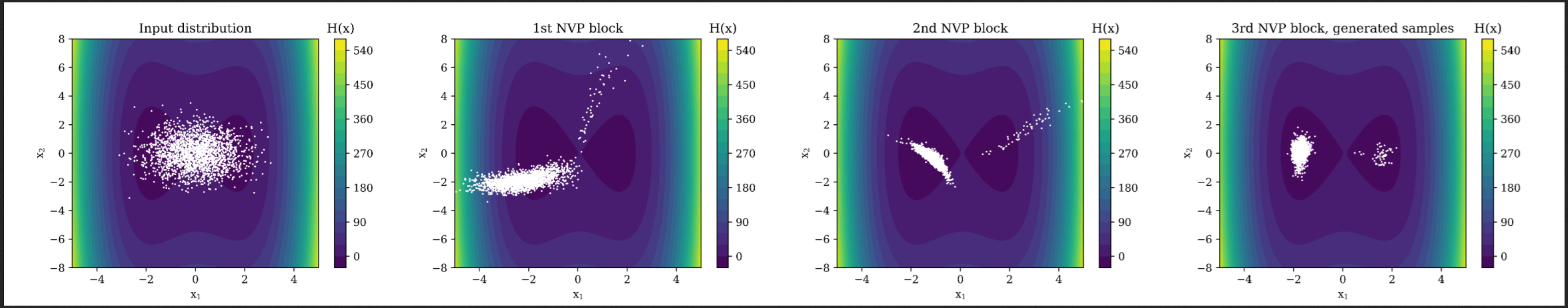
Input | **1st NVP block** | **2nd NVP block** | **3rd NVP block, output**



Boltzmann generators trained on ML loss function was able to generate reasonable configurations in the real space

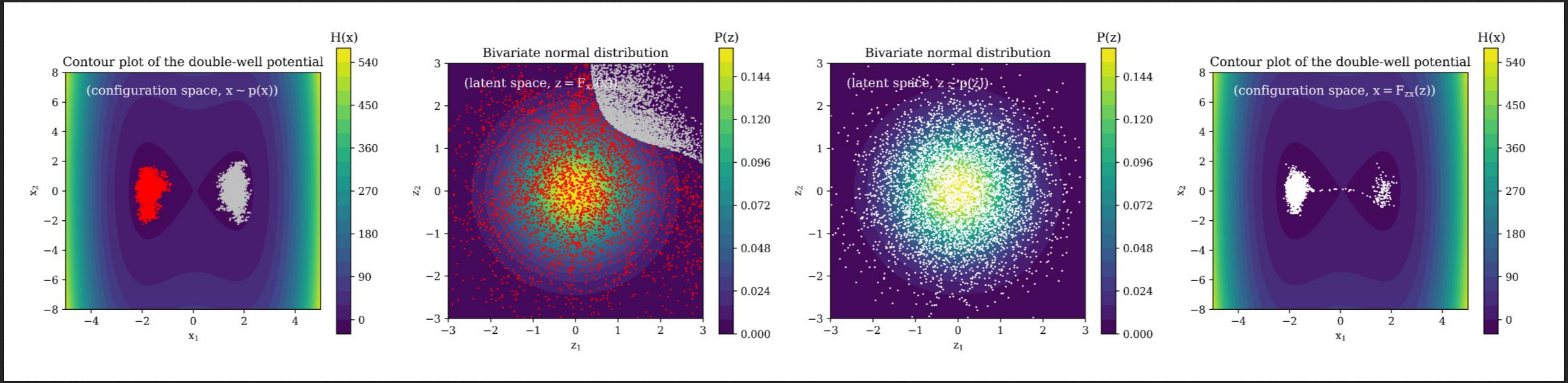
The process of training the generator $F_{zx}(z)$

Input | **1st NVP block** | **2nd NVP block** | **3rd NVP block, output**



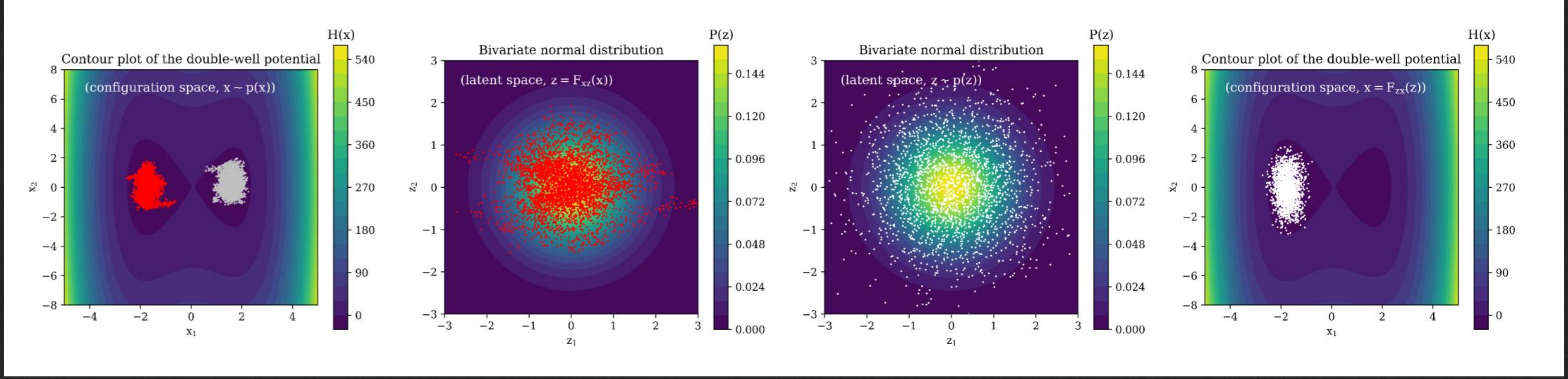
Boltzmann generators trained on ML loss function was able to generate reasonable configurations in the real space

Configurations generated by the Boltzmann generator trained on ML loss



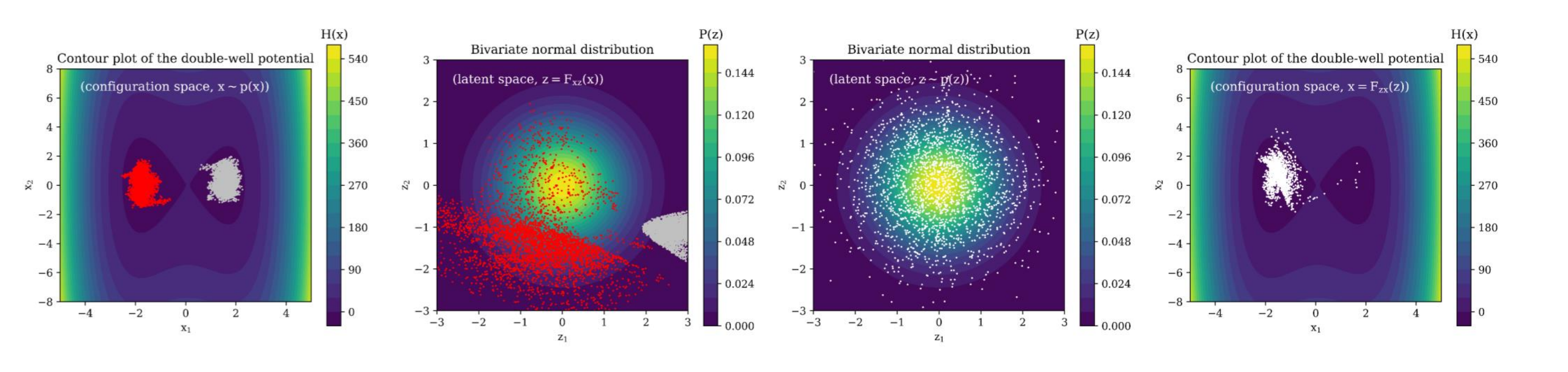
The higher-energy samples were mapped too far away from the center of the Gaussian distribution by the Boltzmann generator trained on KL loss

Configurations generated by the Boltzmann generator trained on KL loss



The Boltzmann generator trained on ML and KL loss at the same time produced an intermediate result

Configurations generated by the Boltzmann generator trained on ML + KL loss

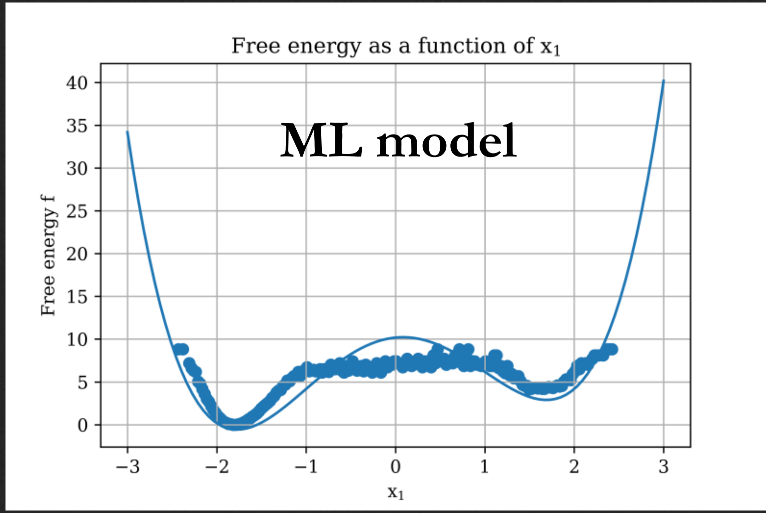


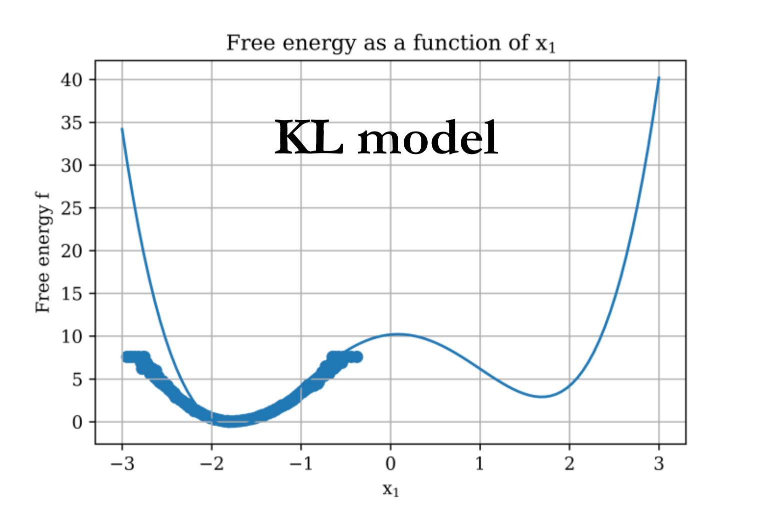
The generated distribution approximating the Boltzmann distribution requires improvements

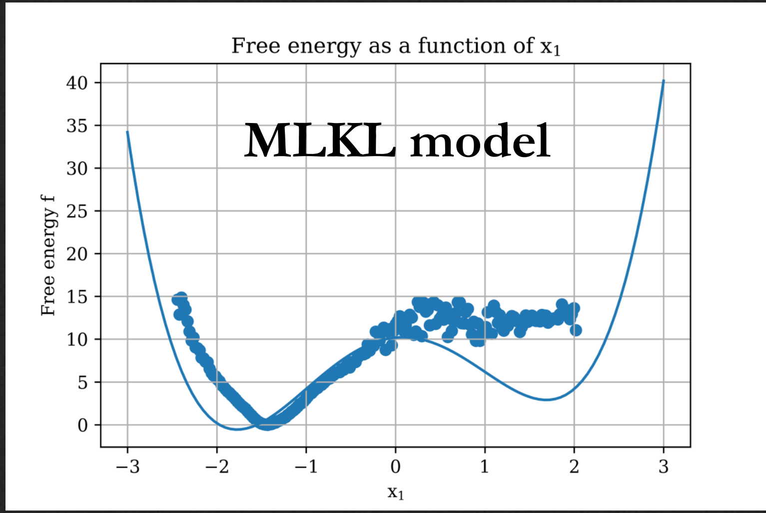
$$Z(x_1) = \int e^{-u(x_1,x_2)} dx_2 = e^{-(x_1^4 - 6x_1^2 + x_1)} \int_{-\infty}^{\infty} e^{-x_2^2} dx_2 = \sqrt{\pi} \cdot e^{-u(x_1)}$$

$$f_{analytical}(x_1) = -\ln Z(x_1) = u(x_1) - \ln(\sqrt{\pi}) = x_1^4 - 6x_1^2 + x_1 - \ln(\sqrt{\pi})$$

$$f_{estimated}(x_1) = -\ln p(x_1)$$







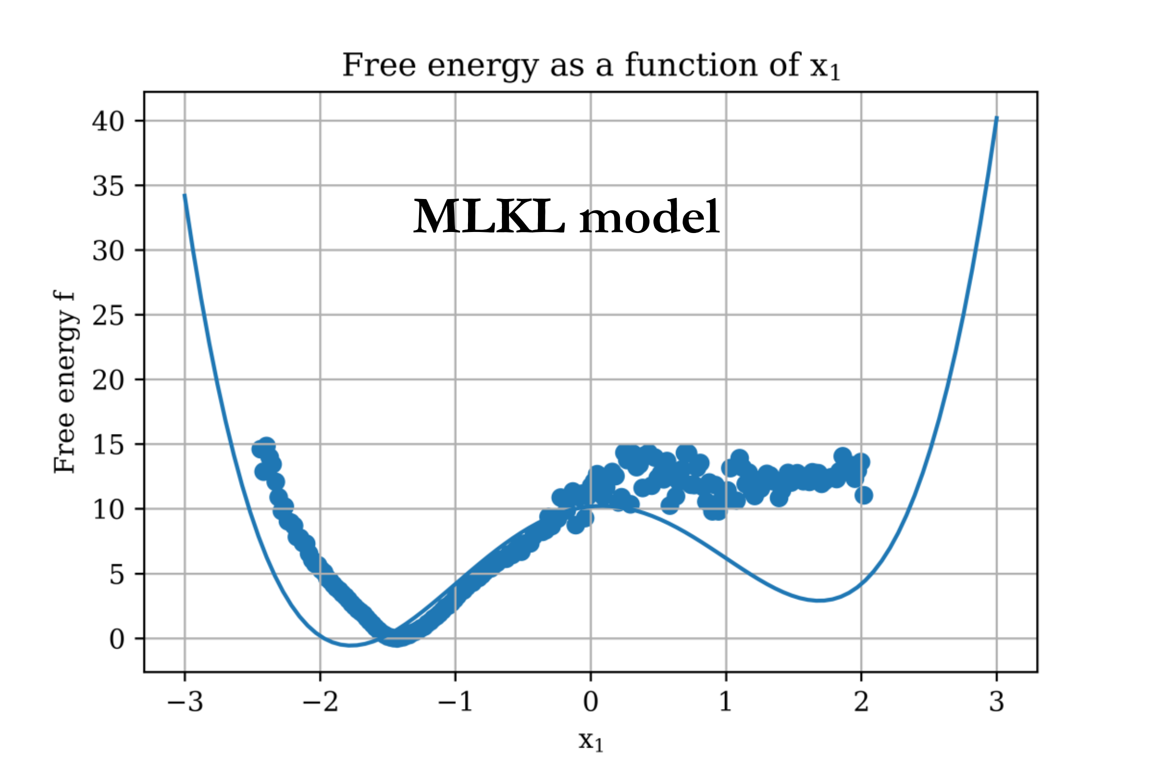
Statistical weights improve the quality of free energy estimation

$$w_x(\mathbf{x}) = \frac{\mu_x(\mathbf{x})}{q_x(\mathbf{x})} = \frac{q_z(\mathbf{z})}{\mu_z(\mathbf{z})}$$

$$w_x(\mathbf{x}) \propto e^{-u_x(F_{xz}(\mathbf{z}))+u_z(\mathbf{z})+\log R_{zx}(\mathbf{z};\theta)}$$

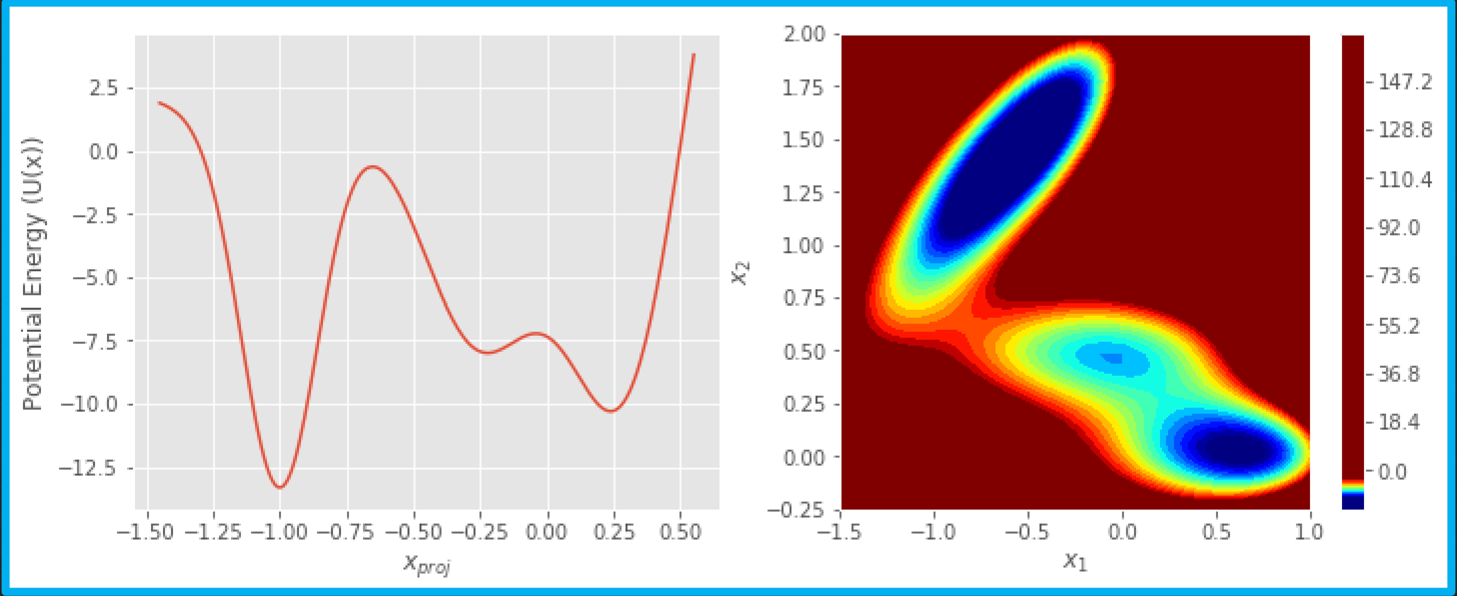
- $x_{i,k}$: k-th sample in the i-th bin
- n_i: number of events in i-th bin
- m: number of bins

$$p_i = \frac{\sum_{k=1}^{n_i} w_x(x_{i,k})}{\sum_{i=1}^{m} n_i}$$



Particle in a Muller-Brown potential to train Boltzmann generators

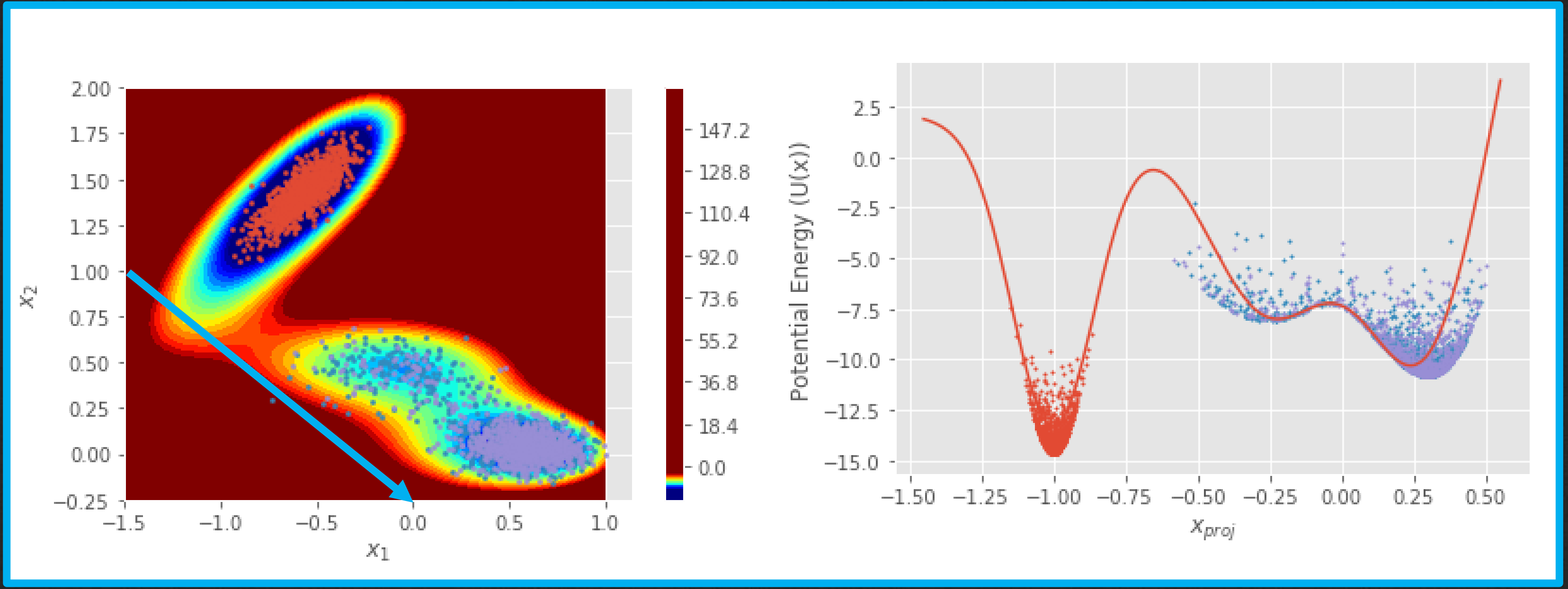
- Muller-Brown potential introduces new considerations:
  - ✓ Intermediate metastable state
  - ✓ Multidimensional transition-states



Particle in a Muller-Brown potential to train Boltzmann generators

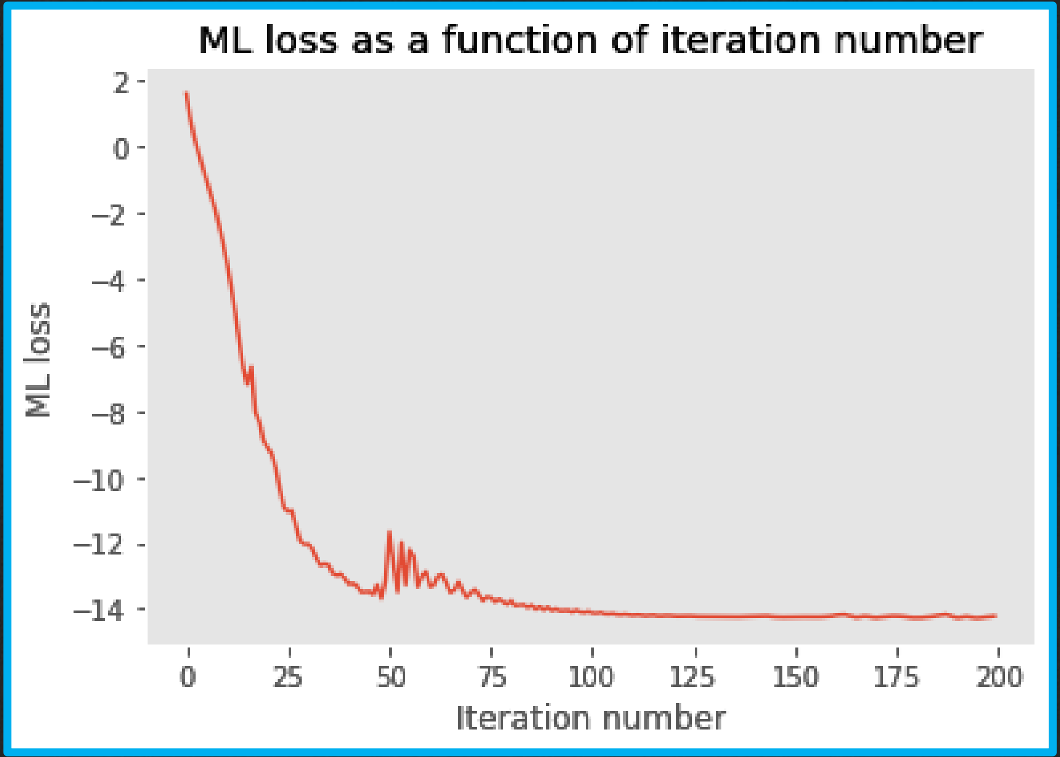
- Muller-Brown potential introduces new considerations:

$$u(x_1, x_2) = \alpha \sum_{j=1}^{4} A_j exp\left[a_i(x-\hat{x}_j)^2 + b_j(x-\hat{x}_j)(y-\hat{y}_j)) + c_j(y-\hat{y}_j)^2\right]$$



	1	2	3	4
a_j	-1	-1	-6.5	0.7
b_j	0	0	11	0.6
c_j	-10	-10	6.5	0.7
A_j	-200	-100	-170	15
$\hat{x}_j$	1	0	-0.5	-1
$\hat{y}_j$	0	0.5	1.5	1

Boltzmann generators trained on ML loss function are able to generate reasonable configurations in the real space

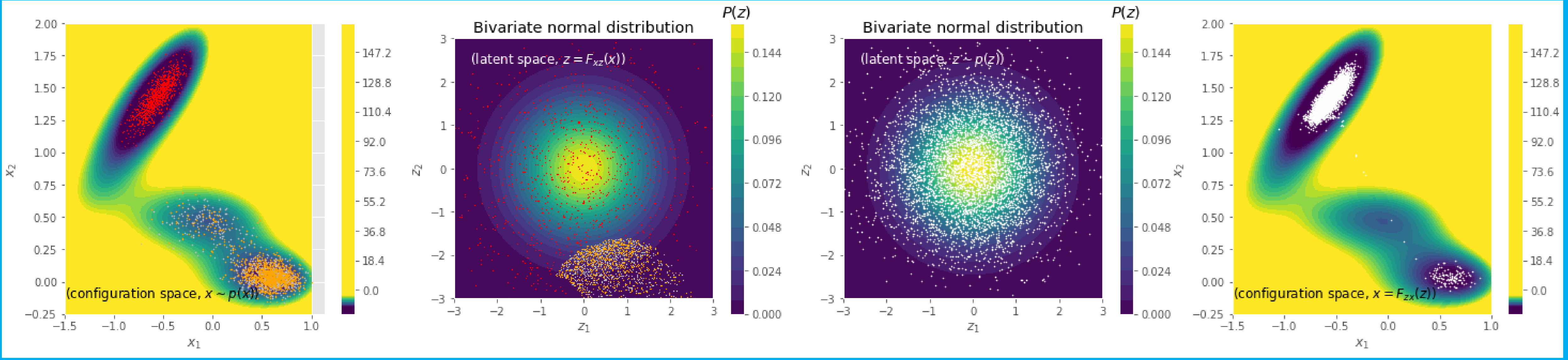


- Parameters
  - ✓ 5 NVP blocks
  - ✓ 3 layers per each NVP block
  - ✓ 100 nodes per each layer
  - ✓ Activation functions: ReLU and tanh output layer
  - ✓ Optimizer: Adam, with LR as 0.001
  - ✓ Batch size: 128 samples
  - ✓ Number of iterations: 200

Boltzmann generators trained on ML loss function are able to generate reasonable configurations in the real space

- ML loss maximizes the likelihood of the inverse transform to place samples within the latent gaussian

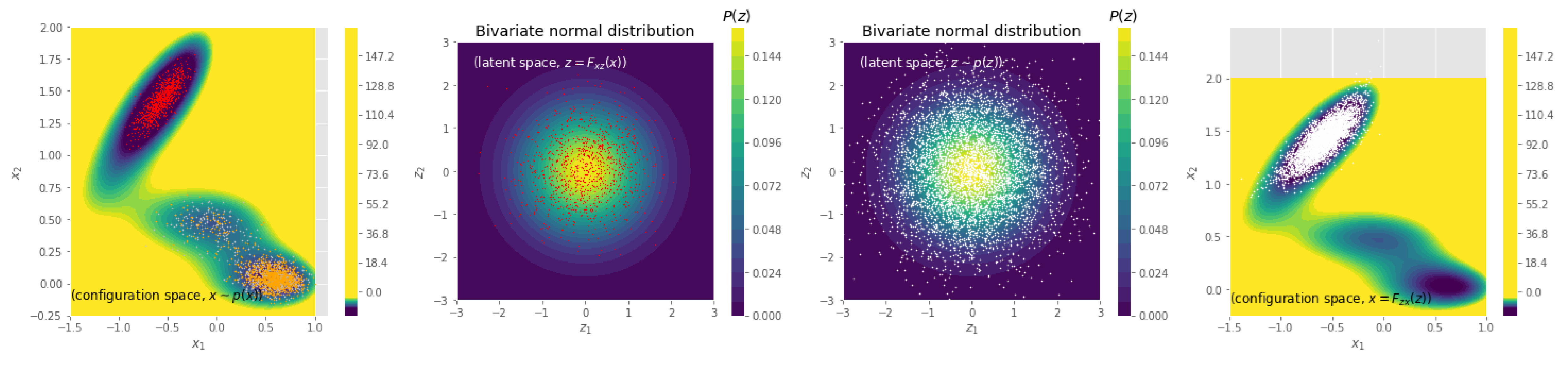
$$J_{ML} = \mathbf{E}_{\mathbf{x}\sim\mu(\mathbf{x})}\left[\frac{1}{2\sigma^2}\|F_{xz}(\mathbf{x};\theta)\|^2 - \log R_{xz}(\mathbf{x};\theta)\right]$$



Training on the KL loss function identified the global minima

- KL loss maximizes the likelihood of the forward transform to place samples within the latent the Boltzmann distribution (i.e. low energy configurations)

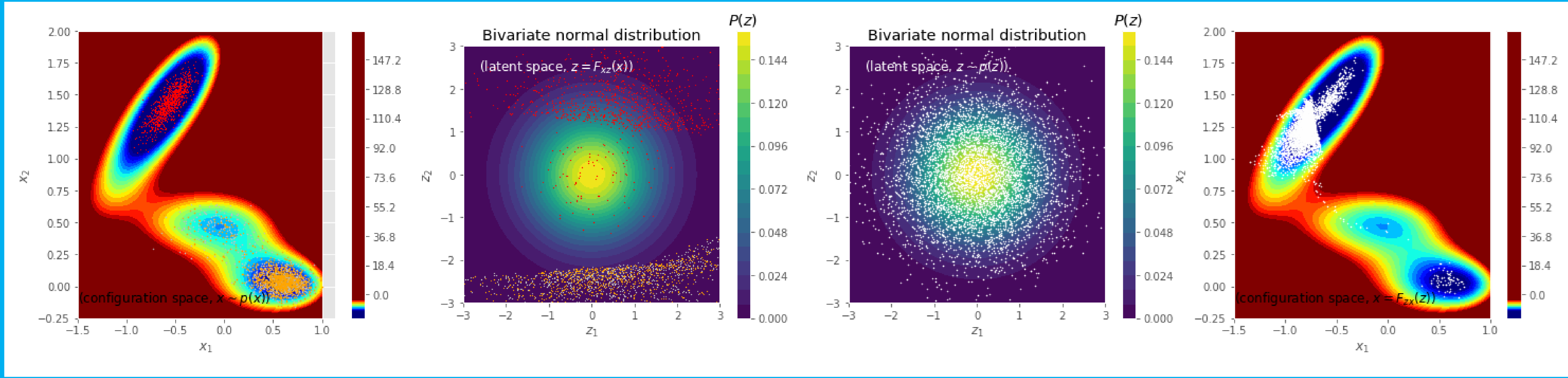
$$J_{KL}(\mu_z \| q_z) = \mathbf{E}_{\mathbf{z} \sim \mu(\mathbf{z})} \left[u(F_{zx}(\mathbf{z}; \theta)) - \log R_{zx}(\mathbf{z}; \theta) \right]$$



Training on the KL+ML loss function

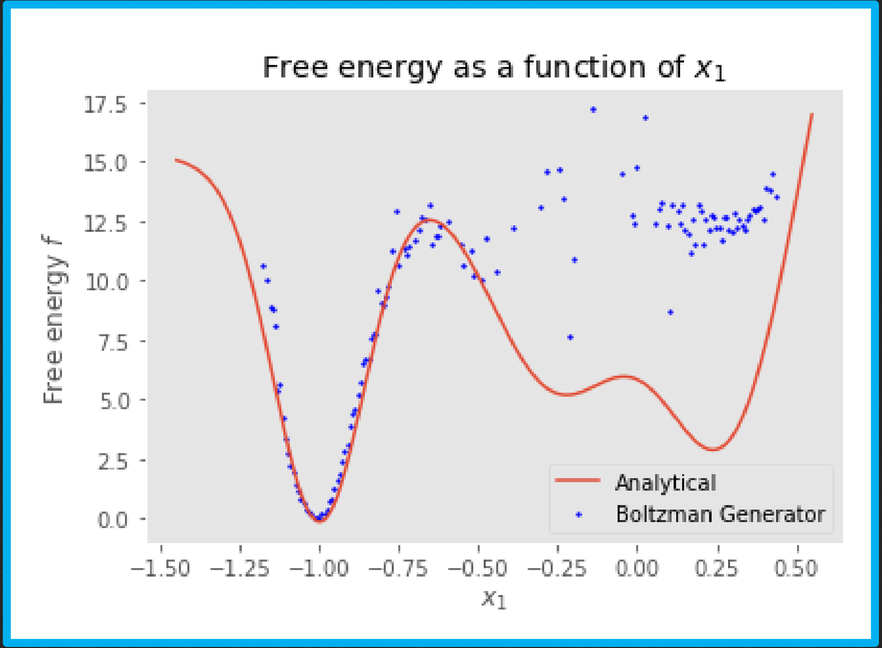
- Combination of both terms blends characteristics of both loss terms

$$J = w_{ML} J_{ML} + w_{KL} J_{KL}$$



Free energy calculation on Muller-Brown potential

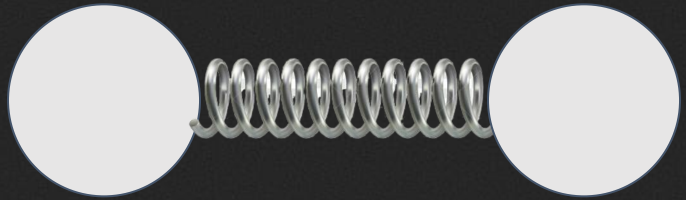
- Preliminary Muller-Brown potential free energy surface



- Preliminary Muller-Brown potential free energy surface
- Poor sampling in local minima

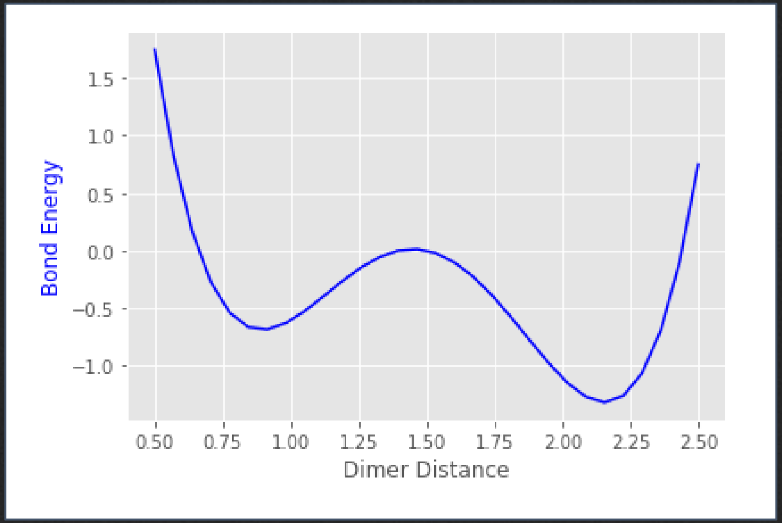
Boltzmann generator's applied to condensed matter simulations

- Bistable dimer in LJ bath considerations:
  - ✓ Inclusion of an internal coordinate potential



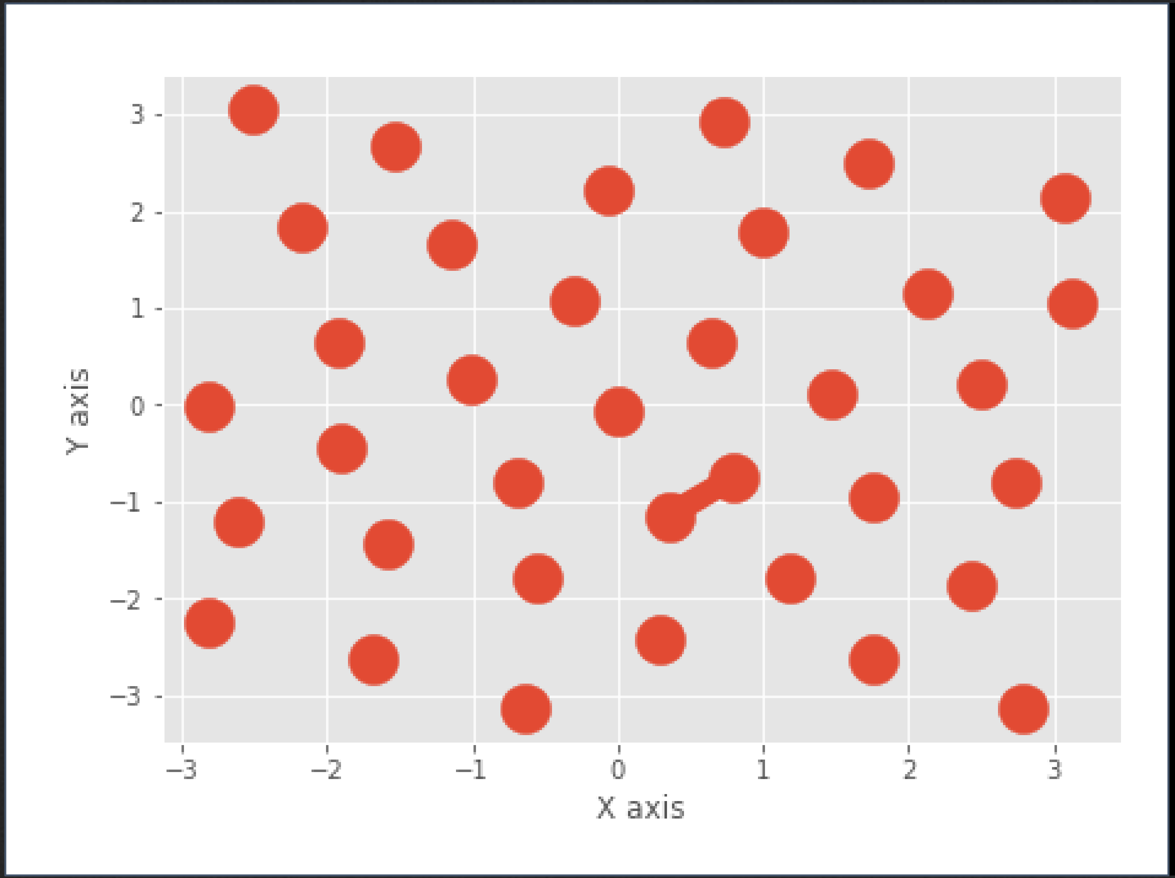
$$U_d(\mathbf{x}) = \frac{1}{4}a(d-d_0)^4 - \frac{1}{2}b(d-d_0)^2 + c(d-d_0)$$

$$U_s(r_{i,j}) = 4\epsilon\left(\frac{\sigma}{r_{i,j}}\right)^{12}$$



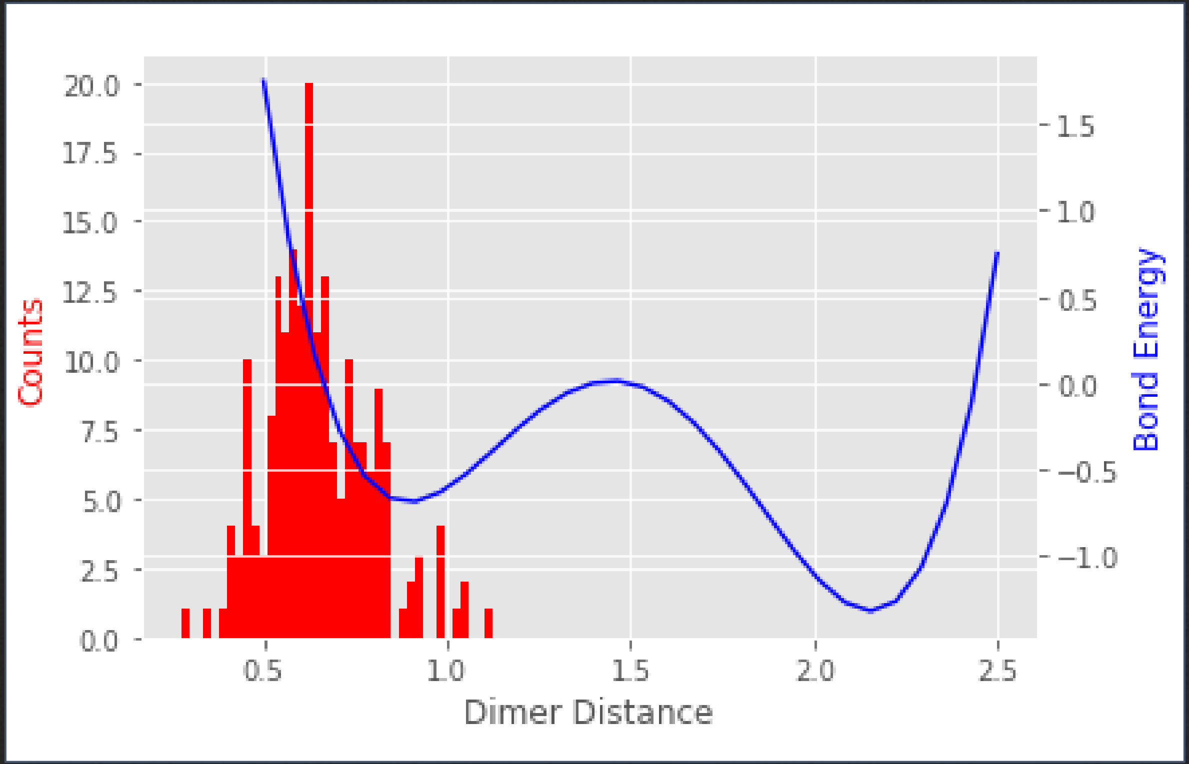
Boltzmann generator's applied to condensed matter simulations

- Bistable dimer in LJ bath considerations:
  - ✓ Inclusion of an internal coordinate potential
  - ✓ 76 dimensional problem
  - ✓ Solvent permutational invariance
  - ✓ Dense system



Boltzmann generator's applied to condensed matter simulations

- Bistable dimer in LJ bath considerations:
  - ✓ Inclusion of an internal coordinate potential
  - ✓ 76 dimensional problem
  - ✓ Solvent permutational invariance
  - ✓ Dense system



Be Boulder.
University of Colorado Boulder

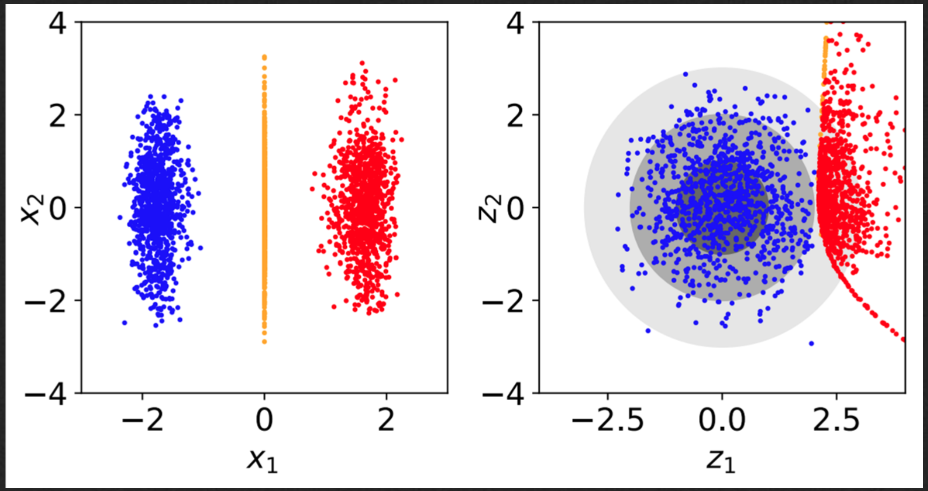
Boltzmann generators are a promising application of machine learning in molecular modeling

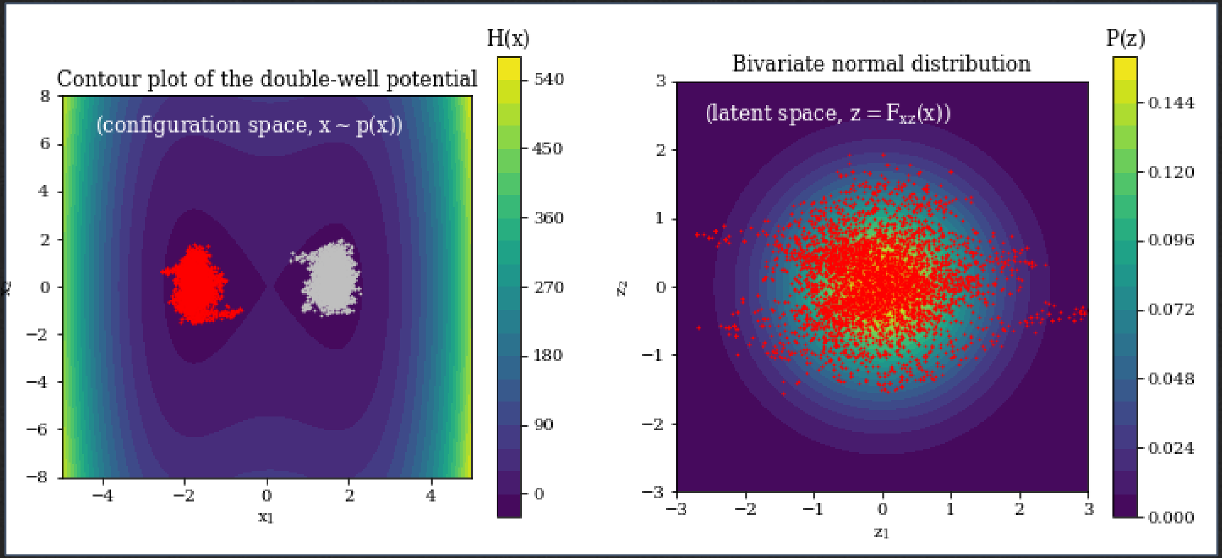
- Rare event-sampling in many-body systems is an active field of research:
  - ✓ Boltzmann generators provides a one-shot sampling of metastable states
  - ✓ Reaction coordinate free sampling method
  - ✓ Highly scalable ML algorithm
  - ✓ Lower computational cost than traditional enhanced sampling methods
- Application to biomolecular systems
  - ✓ Noé *et al.* applied Boltzmann generators to protein system
  - ✓ Many similarly framed problems in biomolecular simulations

Much of the results shared today were preliminary results and requires further work

- We are unable to reproduce KL loss results from the paper

KL

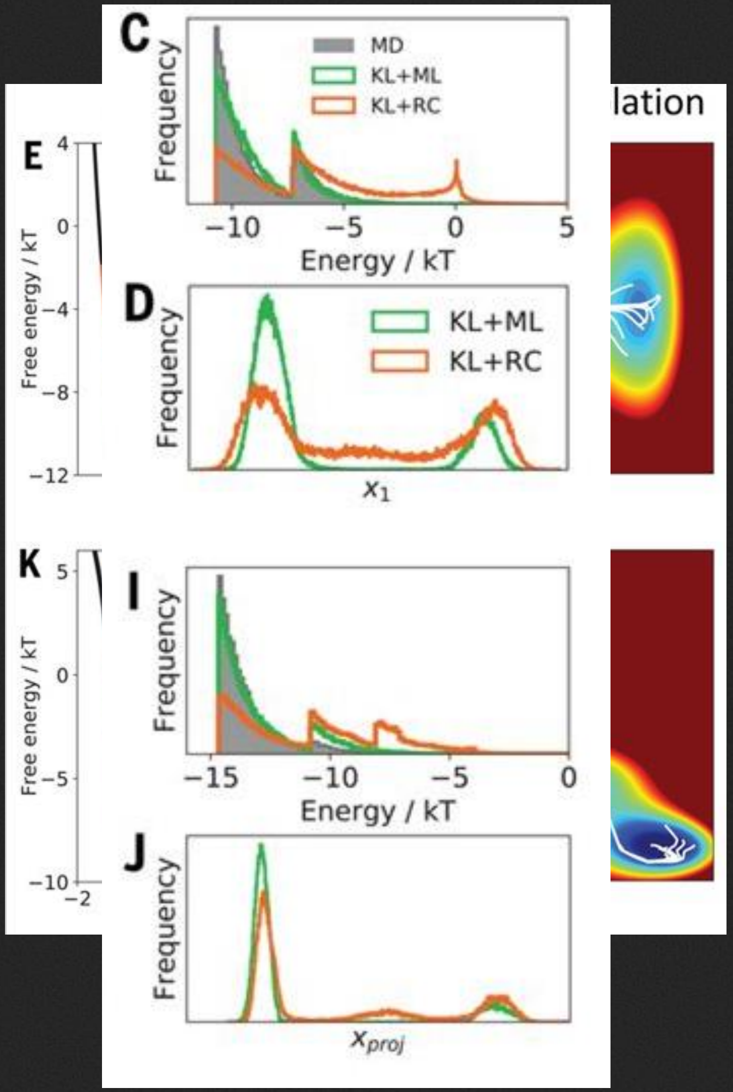




Conclusion - Continued work

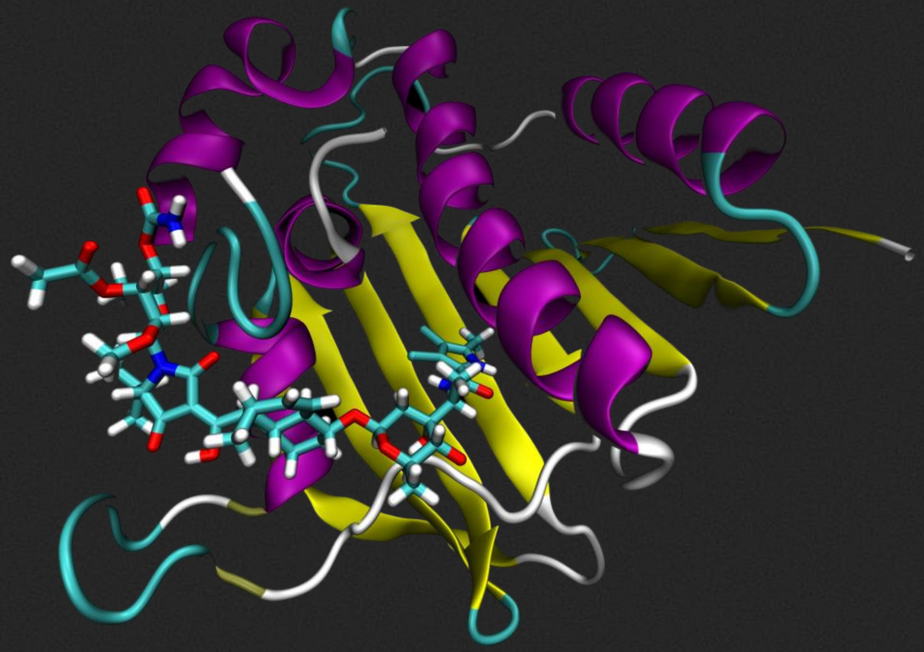
Much of the results shared today were preliminary results and requires further work

- We are unable to reproduce KL loss results from the paper
- Implement the reaction coordinate loss term (J_{RC})
- Latent space interpolation
- Apply Boltzmann generators to the dimer in LJ bath system
- Tidy up Jupyter-notebooks and github repository



Future work

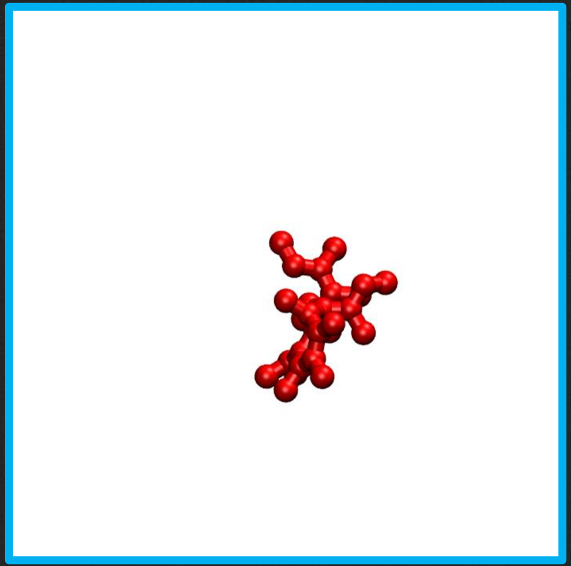
- Applying Boltzmann generators in our research:
  - ✓ Provides insight into the binding affinity of the complex.
  - ✓ Determines a binding ensemble.
  - ✓ Enables the examination of alternate binding conformations not predicted experimentally
  - ✓ Provides insight into the multimodal specificity and the binding mechanism of a binding complex.

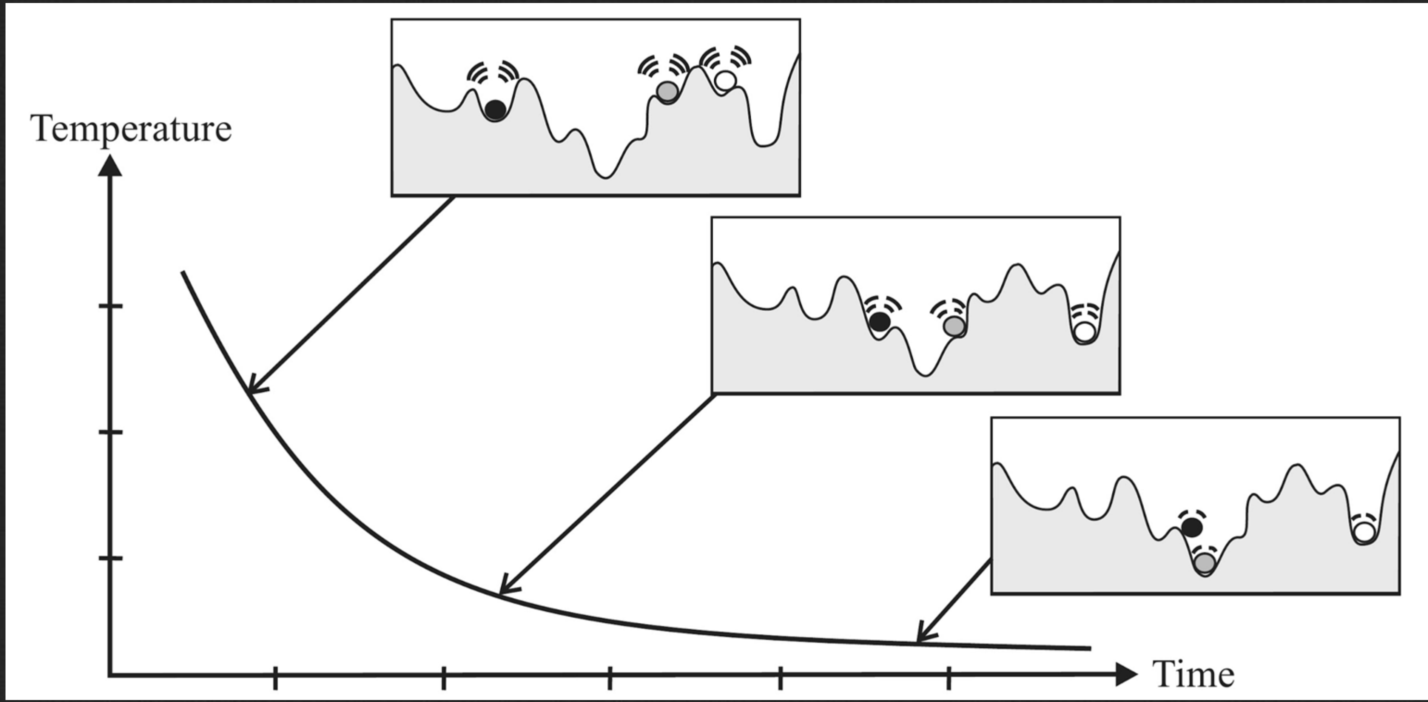


Future work

- Applying Boltzmann generators in our research:
  - ✓ Peptide folding transitions

Metropolis-Minimization

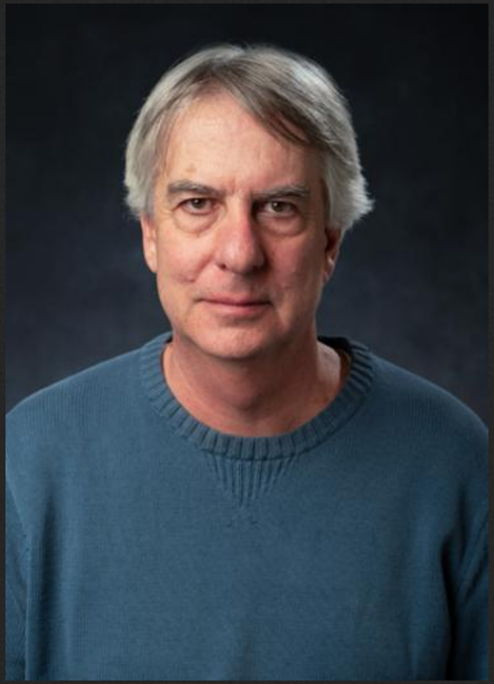




Acknowledgements

Thank you to Dr. Glaser for teaching the class and for our many discussions

Thank you to Dr. Shirts for introducing and discussing the Boltzmann generator paper with us

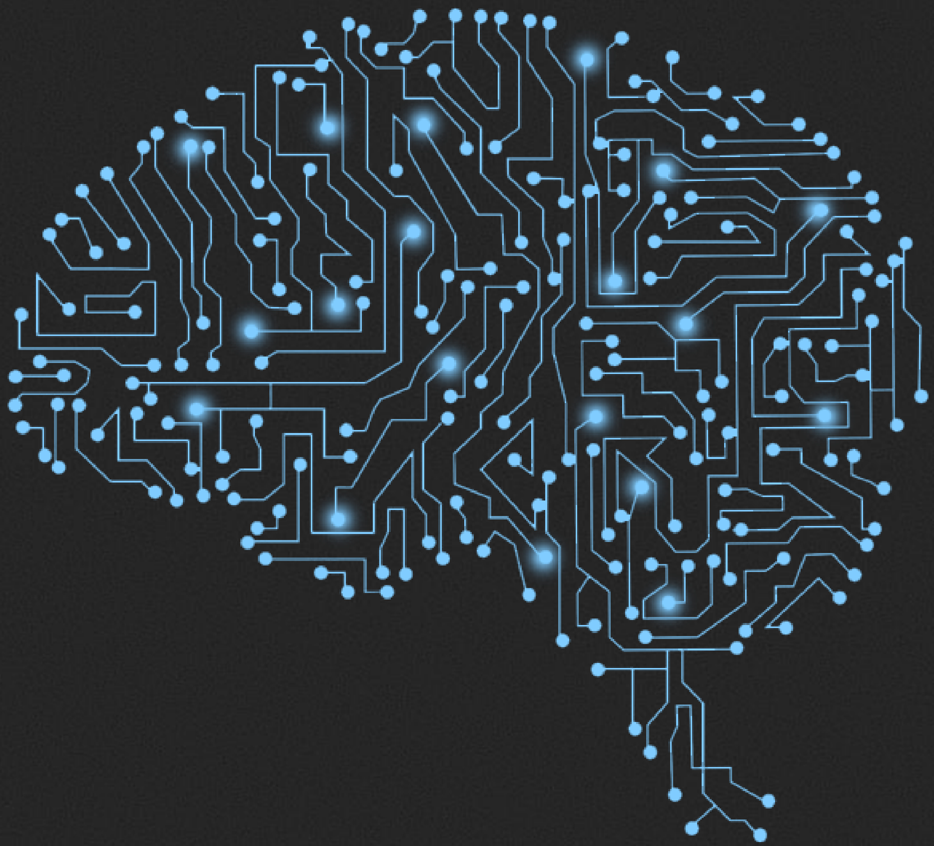


Thank you for listening!

Boltzmann generators: Efficient sampling of equilibrium states of many-body system with deep learning

Thank you for listening!

Any questions?



Be Boulder.

University of Colorado Boulder