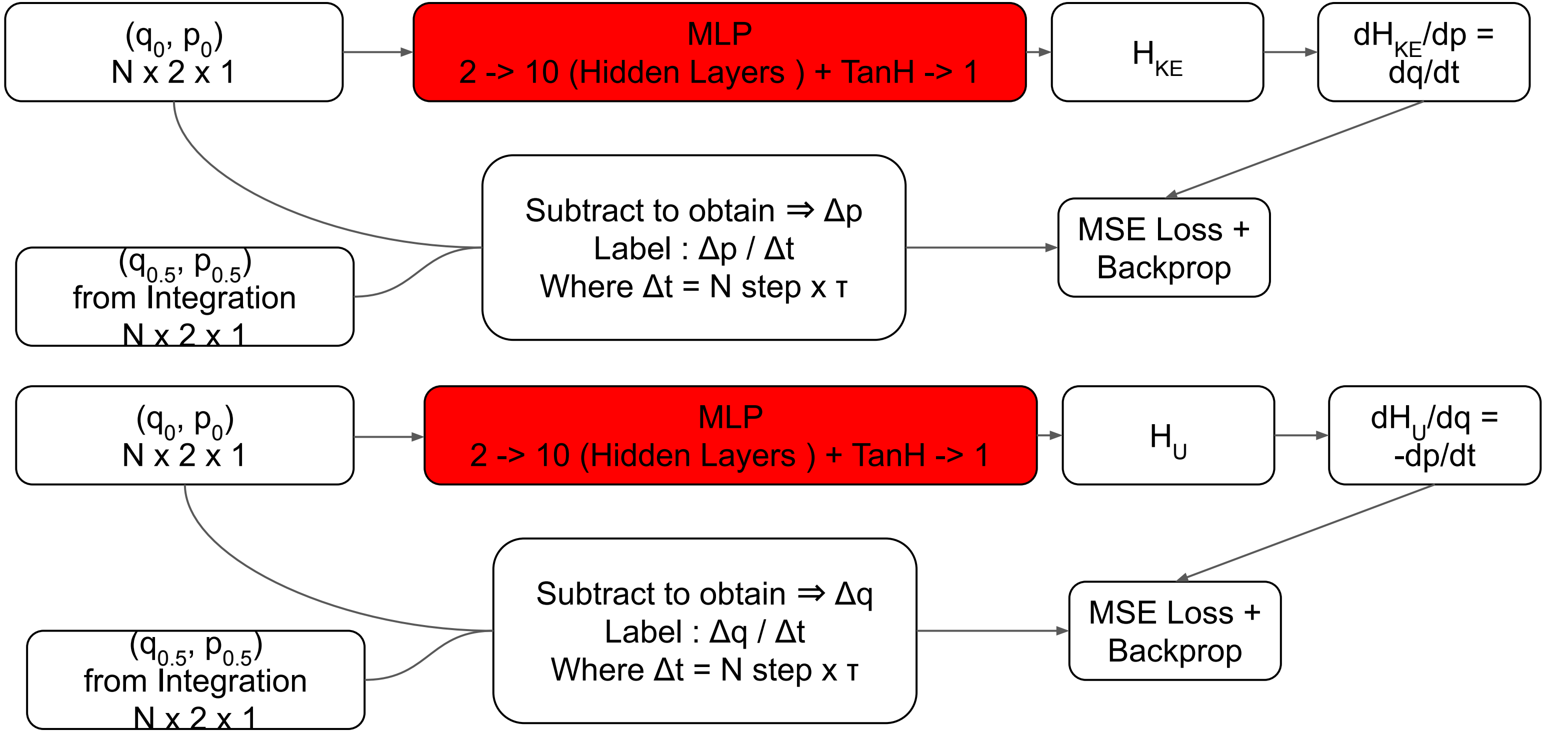
MD ML 7

2 Models

Proposed Hamiltonian Architecture



Training Setting

Optimizer : Adam

Learning Rate (LR) : 1e-3

Scheduler : None

Seed : 937162211 (9-digit Prime Number)

Epochs : 100

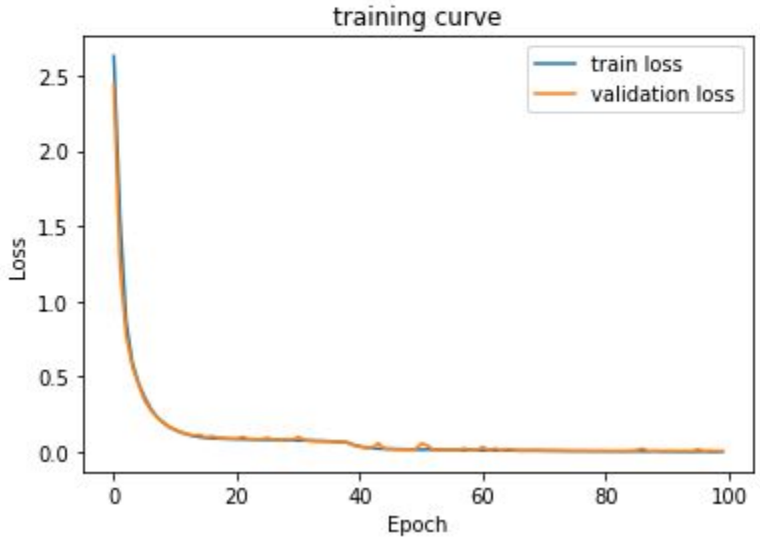
Batch Size : 32 / Shuffle

Large Time Step : 0.01 (1 Step) = 0.01

Ground Truth : 0.01

Final Statistic : Δq = 0.00034 (5 d.p)

Δp = 0.00099 (5 d.p)



Remarks :

1. *Tried using scheduler of reduce on plateau, patience = 10 , factor = 0.99, loss stuck at ~1.2*
2. *Temperature used : 1 - 10, interval 1*
3. *Tried 50 epochs, loss seems to be fluctuating*
4. *Seed to consider :*
    a. *Random.seed*
    b. *Numpy.seed*
    c. *Torch.seed*
    d. *Torch.cuda.seed*

Distribution Sampling

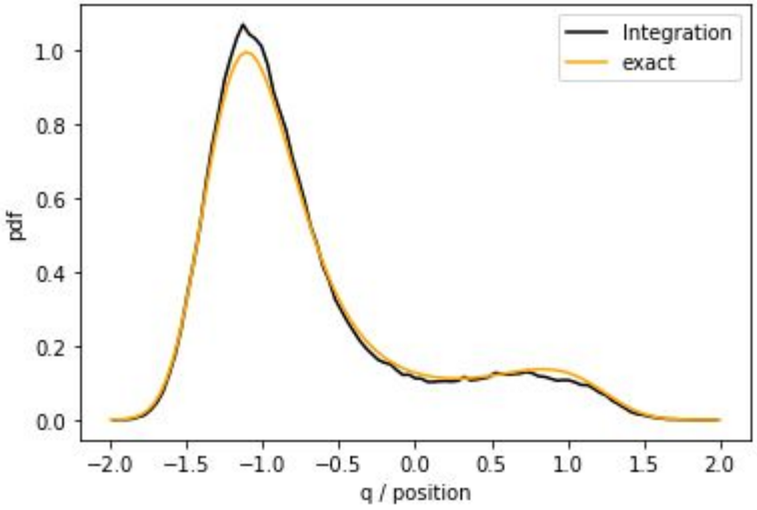
To update the distribution, we use the following splitting :

$$P(t = t + \Delta t) = [L_{\gamma} (L_{p(\Delta t/n)} L_{q(\Delta t/n)} L_{p(\Delta t/n)})^{n} L_{\gamma}]P(t = t)$$

Where $\Delta t/n = 0.01$ for the ground truth reference to calculate the **mean absolute error**

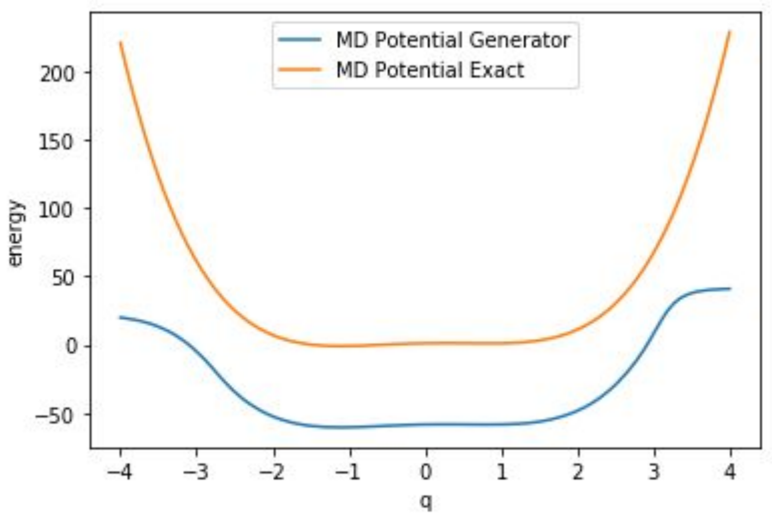
1. $(L_{p(\Delta t/n)} L_{q(\Delta t/n)} L_{p(\Delta t/n)})^{n}$ can be replaced by the model separately where n = 1
2. The constant random term is kept constant for Model and exact integration and the distribution is sampled after each update

Distribution Performance (10^4 Sampling)

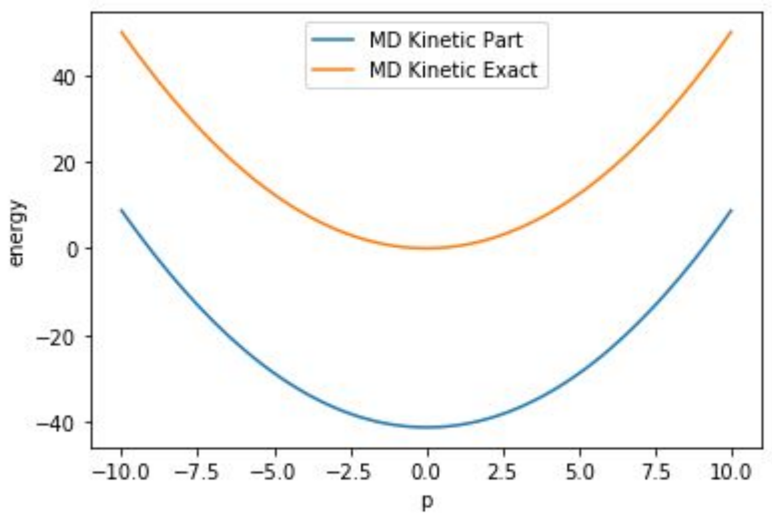


Hamiltonian Checking (time step : 0.01)

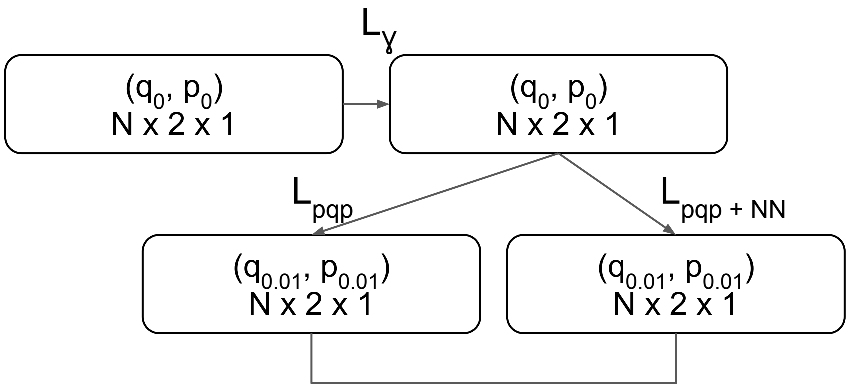
Potential



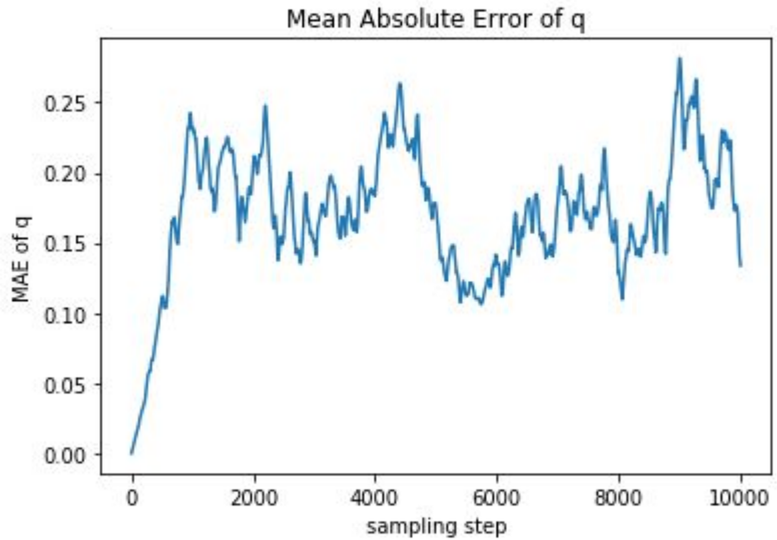
Kinetic Energy



Mean Absolute Error (time step : 0.01)

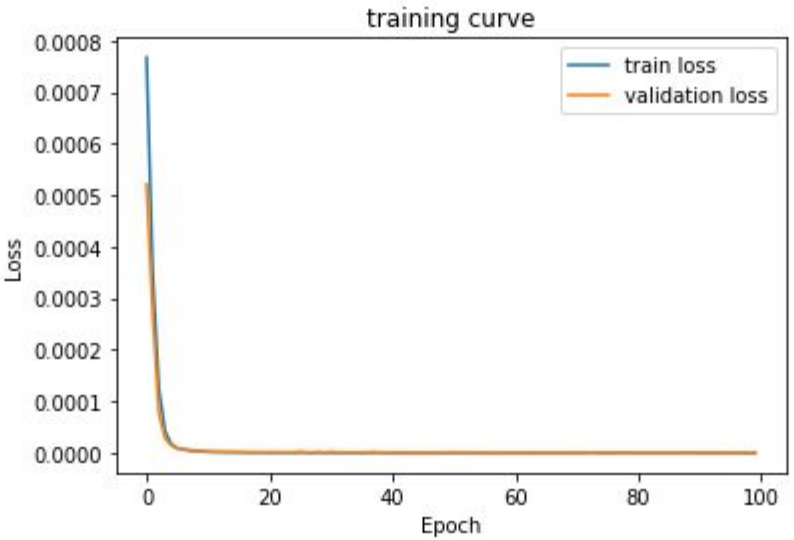


Subtract to obtain
Mean Absolute Error



Training Setting

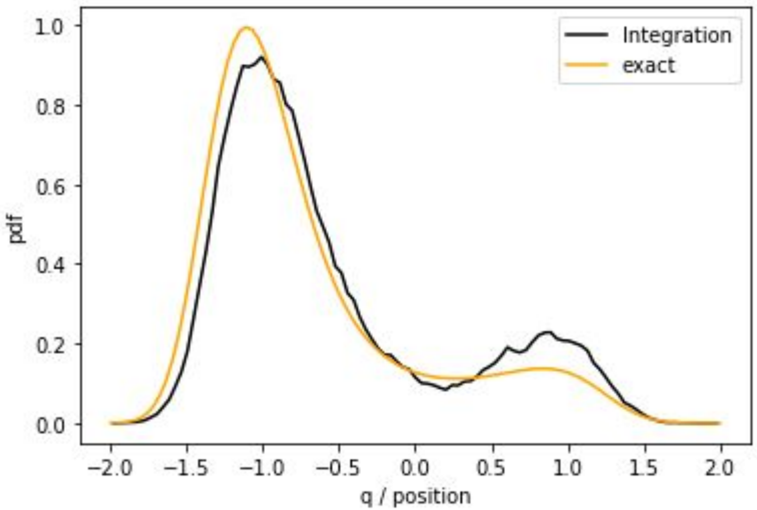
Optimizer	: Adam
Learning Rate (LR)	: 1e-3
Scheduler	: None
Seed	: 937162211 (9-digit Prime Number)
Epochs	: 100
Batch Size	: 32 / Shuffle
Large Time Step	: 0.01 (50 Step) = 0.5
Ground Truth	: 0.01
Final Statistic	: Δq = 0.00018 (5 d.p)
	Δp = 0.00023 (5 d.p)



Remarks :

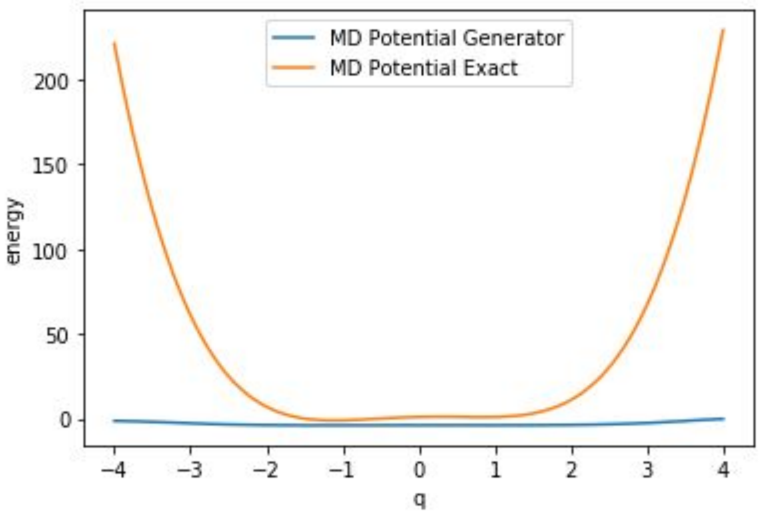
1. *Tried using scheduler of reduce on plateau, patience = 10 , factor = 0.99, loss seems to be stuck again*
2. *Temperature used : 1 - 10, interval 1*
3. *Seed to consider :*
   a. *Random.seed*
   b. *Numpy.seed*
   c. *Torch.seed*
   d. *Torch.cuda.seed*

Distribution Performance (10^4 Sampling)

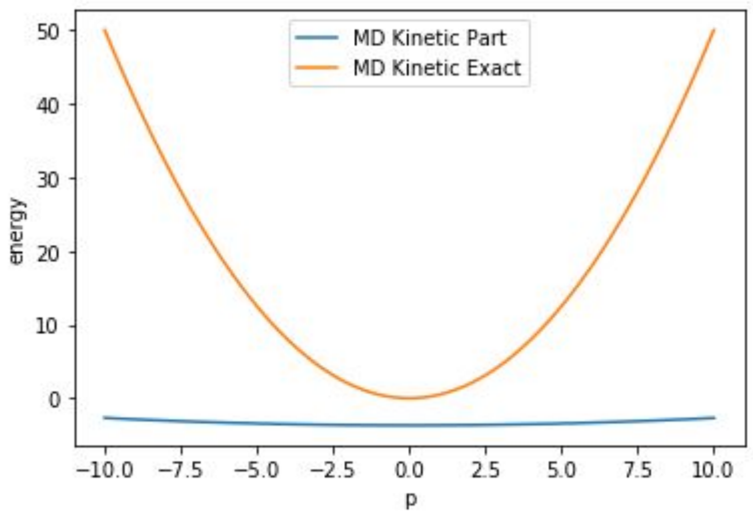


Hamiltonian Checking (time step : 0.5)

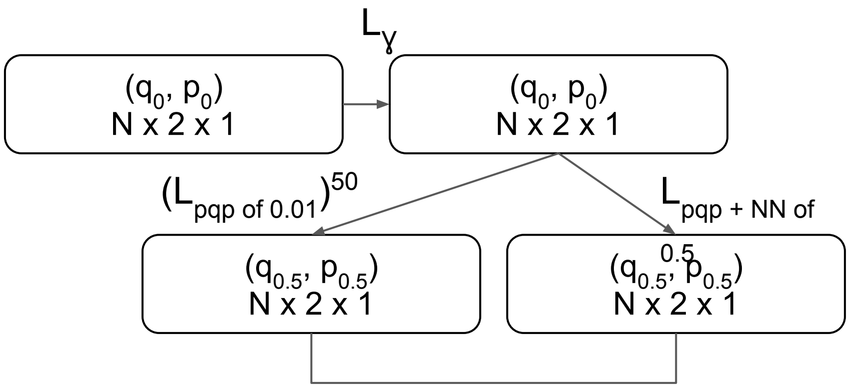
Potential



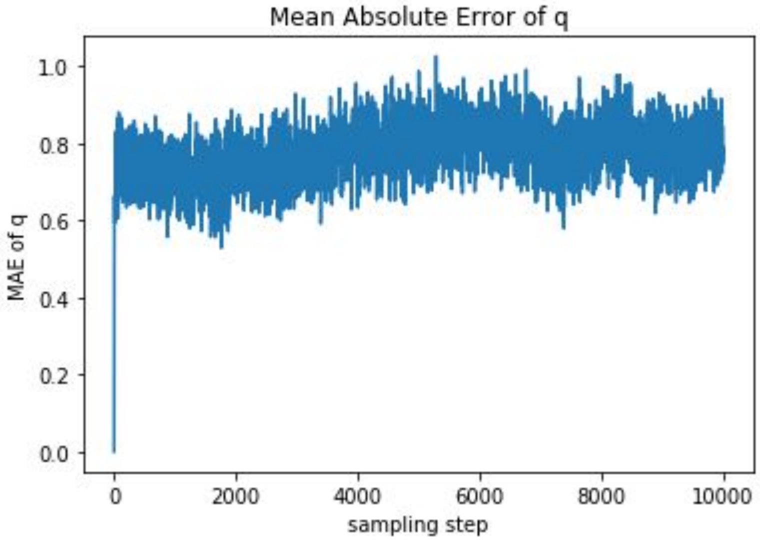
Kinetic Energy



Mean Absolute Error (time step : 0.5)



Subtract to obtain Mean Absolute Error



Analysis

Literature Review

1) Langevin Process :

1. Introduction to the theory of stochastic processes and Brownian motion problems by J. L. Garcia-Palacios (https://arxiv.org/abs/cond-mat/0701242v1) ⇒ Reference that I used
2. Overleaf Document by Liu Wei

2) Hamiltonian Neural Network :

1. Hamiltonian Neural Networks by Sam Greydanus, Misko Dzamba, Jason Yosinski (https://arxiv.org/abs/1906.01563)
2. Symplectic Recurrent Neural Networks by Zhengdao Chen, Jianyu Zhang, Martin Arjovsky, Léon Bottou (https://arxiv.org/abs/1909.13334)

3) Langevin Sampling :

1. Robust and efficient configurational molecular sampling via Langevin Dynamics by Benedict Leimkuhler, Charles Matthews (https://arxiv.org/abs/1304.3269)