Interpolation of Molecular Dynamics with Bi-Directional Neural Networks

Ludwig Winkler & Huziel Sauceda

November 12, 2020

Molecular Dynamics (MD)

- Molecules are governed by Newton's equations of motion

Molecular Dynamics (MD)

- Molecules are governed by Newton's equations of motion
- Position $r(t)$ and momentum $p(t)$ describe the system completely

Molecular Dynamics (MD)

- Molecules are governed by Newton's equations of motion
- Position $r(t)$ and momentum $p(t)$ describe the system completely
- Dynamics f given by differential equation (DiffEq)

$$\dot{x}(t) = \begin{bmatrix} \dot{r}(t) \\ \dot{p}(t) \end{bmatrix} = f\left(\begin{bmatrix} r(t) \\ p(t) \end{bmatrix}, t \right)$$

Molecular Dynamics (MD)

- Molecules are governed by Newton's equations of motion
- Position $r(t)$ and momentum $p(t)$ describe the system completely
- Dynamics f given by differential equation (DiffEq)

$$\dot{x}(t) = \begin{bmatrix} \dot{r}(t) \\ \dot{p}(t) \end{bmatrix} = f\left(\begin{bmatrix} r(t) \\ p(t) \end{bmatrix}, t\right)$$

- Solution is a trajectory in phase space

$$x(t) = \begin{bmatrix} r(t) \\ p(t) \end{bmatrix} = \int_{t_0}^{t} f\left(\begin{bmatrix} r(t) \\ p(t) \end{bmatrix}, t\right) dt$$

Molecular Dynamics (MD)

- High-dimensional N-body problems are expensive to solve

Molecular Dynamics (MD)

- High-dimensional N-body problems are expensive to solve
- MD requirements
  - $\Delta t = 10^{-15}s$ integration step size for accurate dynamics
  - $T \in \{10^{-6}s, ..., 10^{-9}s\}$ solutions for expressive properties

Molecular Dynamics (MD)

- High-dimensional N-body problems are expensive to solve
- MD requirements
  - $\Delta t = 10^{-15}s$ integration step size for accurate dynamics
  - $T \in \{10^{-6}s, ..., 10^{-9}s\}$ solutions for expressive properties
- Dynamics f are expensive to compute for large N-body problems and long integration lengths

Molecular Dynamics (MD)

- High-dimensional N-body problems are expensive to solve
- MD requirements
  - $\Delta t = 10^{-15}s$ integration step size for accurate dynamics
  - $T \in \{10^{-6}s, ..., 10^{-9}s\}$ solutions for expressive properties
- Dynamics f are expensive to compute for large N-body problems and long integration lengths

- Yet, essentially we are dealing with a time series prediction problem ...

Molecular Dynamics (MD)

- High-dimensional N-body problems are expensive to solve
- MD requirements
  - $\Delta t = 10^{-15}s$ integration step size for accurate dynamics
  - $T \in \{10^{-6}s, ..., 10^{-9}s\}$ solutions for expressive properties
- Dynamics f are expensive to compute for large N-body problems and long integration lengths

- Yet, essentially we are dealing with a time series prediction problem ...

Can we learn the phase space dynamics with a ML algorithm?

Learning Dynamical Systems

- Given true dynamics f, learn dynamics f_θ with NN

Learning Dynamical Systems

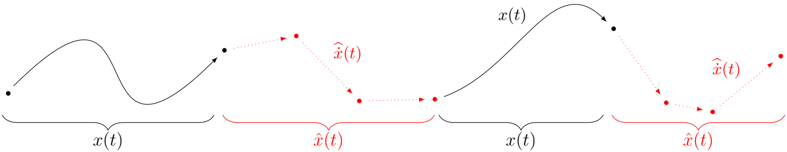
- Given true dynamics f, learn dynamics f_θ with NN
- Optimize f_θ to predict the true solutions $x(t)$

Learning Dynamical Systems

- Given true dynamics f, learn dynamics f_θ with NN
- Optimize f_θ to predict the true solutions $x(t)$
- Integrate dynamics $\widehat{\dot{x}}(t) = f_\theta$ to obtain learned solution $\hat{x}(t)$

Learning Dynamical Systems

- Given true dynamics f, learn dynamics f_θ with NN
- Optimize f_θ to predict the true solutions $x(t)$
- Integrate dynamics $\widehat{\dot{x}}(t) = f_\theta$ to obtain learned solution $\hat{x}(t)$



- ODENetwork

$$\dot{x}(t) = \begin{bmatrix} \dot{r}(t) \\ \dot{p}(t) \end{bmatrix} = f\left(\begin{bmatrix} r(t) \\ p(t) \end{bmatrix}, t \right)$$

- ODENetwork

$$\dot{x}(t) = \begin{bmatrix} \dot{r}(t) \\ \dot{p}(t) \end{bmatrix} = f\left(\begin{bmatrix} r(t) \\ p(t) \end{bmatrix}, t \right)$$

- HamiltonianNetwork

$$\dot{x}(t) = \begin{bmatrix} \dot{r}(t) \\ \dot{p}(t)) \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{H}(r(t),p(t),t)}{\partial p(t)} \\ -\frac{\partial \mathcal{H}(r(t),p(t),t)}{\partial r(t)} \end{bmatrix}$$

- ODENetwork

$$\dot{x}(t) = \begin{bmatrix} \dot{r}(t) \\ \dot{p}(t) \end{bmatrix} = f\left(\begin{bmatrix} r(t) \\ p(t) \end{bmatrix}, t\right)$$

- HamiltonianNetwork

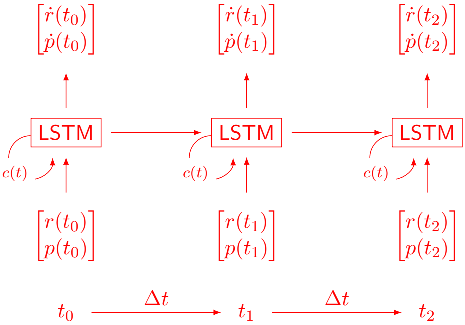
$$\dot{x}(t) = \begin{bmatrix} \dot{r}(t) \\ \dot{p}(t)) \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{H}(r(t),p(t),t)}{\partial p(t)} \\ -\frac{\partial \mathcal{H}(r(t),p(t),t)}{\partial r(t)} \end{bmatrix}$$

- RNN and LSTM

$$\dot{x}(t) = \begin{bmatrix} \dot{r}(t) \\ \dot{p}(t) \end{bmatrix} = f_\theta\left(\left[\begin{bmatrix} r(t_0) \\ p(t_0) \end{bmatrix}, \ldots, \begin{bmatrix} r(t) \\ p(t) \end{bmatrix}\right], t\right)$$

Learning Dynamical Systems with LSTM

- Best performing due to fewest assumptions and flexible parameterization
- Memory cell $c(t)$ to selectively read and write information
- Outputs $[\dot{r}(t), \dot{p}(t)]^T$ are integrated to obtain solution $\hat{x}(t)$



Bi-Directional Interpolation of Differential Equation

- Analytical simulation provides most accurate solution

Bi-Directional Interpolation of Differential Equation

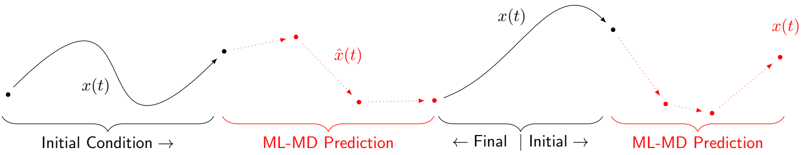
- Analytical simulation provides most accurate solution
- Combine speed of ML-MD with accuracy of analytical MD

Bi-Directional Interpolation of Differential Equation

- Analytical simulation provides most accurate solution
- Combine speed of ML-MD with accuracy of analytical MD
- Use coarse, analytical MD to provide initial and final conditions

Bi-Directional Interpolation of Differential Equation

- Analytical simulation provides most accurate solution
- Combine speed of ML-MD with accuracy of analytical MD
- Use coarse, analytical MD to provide initial and final conditions



Bi-Directional Interpolation of Differential Equation

- Predict forward solution $\overrightarrow{\hat{x}(t)}$ and backward solution $\overleftarrow{\hat{x}(t)}$ with the **same** dynamics f_θ

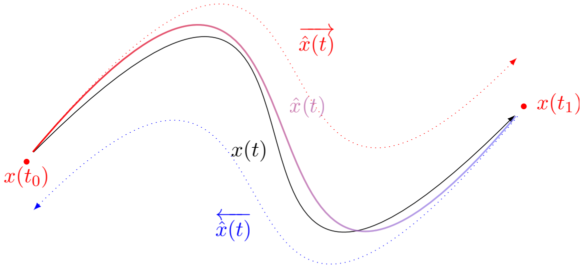
Bi-Directional Interpolation of Differential Equation

- Predict forward solution $\overrightarrow{\hat{x}(t)}$ and backward solution $\overleftarrow{\hat{x}(t)}$ with the **same** dynamics f_θ
- Use adiabatic connection to interpolate $\overrightarrow{\hat{x}(t)}$ and $\overleftarrow{\hat{x}(t)}$ to $\hat{x}(t)$

Bi-Directional Interpolation of Differential Equation

- Predict forward solution $\overrightarrow{\hat{x}(t)}$ and backward solution $\overleftarrow{\hat{x}(t)}$ with the **same** dynamics f_θ
- Use adiabatic connection to interpolate $\overrightarrow{\hat{x}(t)}$ and $\overleftarrow{\hat{x}(t)}$ to $\hat{x}(t)$

$$\hat{x}(t) = (1 - \lambda(t))\ \overrightarrow{\hat{x}(t)} + \lambda(t)\ \overleftarrow{\hat{x}(t)} \tag{1}$$

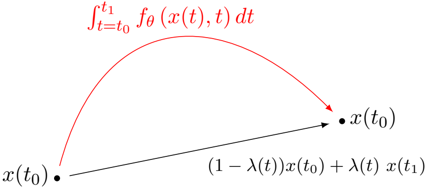


Bi-Directional Interpolation of Differential Equation

- For time-reversible solutions, we obtain

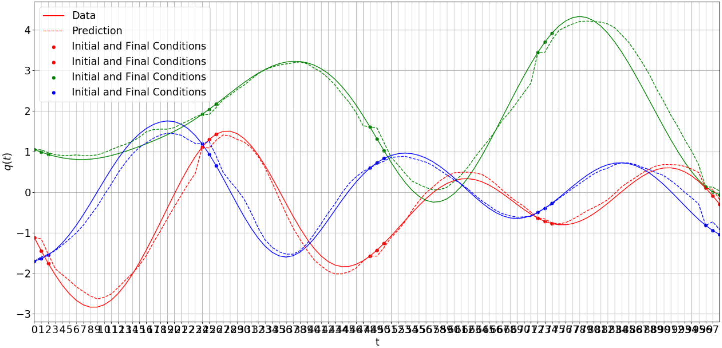
$$\begin{aligned}\hat{x}(t) &= (1-\lambda(t))\ \overrightarrow{\hat{x}(t)} + \lambda(t)\ \overleftarrow{\hat{x}(t)} \\ &= \underbrace{(1-\lambda(t))x(t_0) + \lambda(t)x(t_1)}_{\text{low frequency components}} + \underbrace{\int_{t=t_0}^{t_1} f_\theta\left(x(t), t\right) dt}_{\text{high frequency components}}\end{aligned}$$

- Adiabatic connection frees the ML model to model high frequency signals



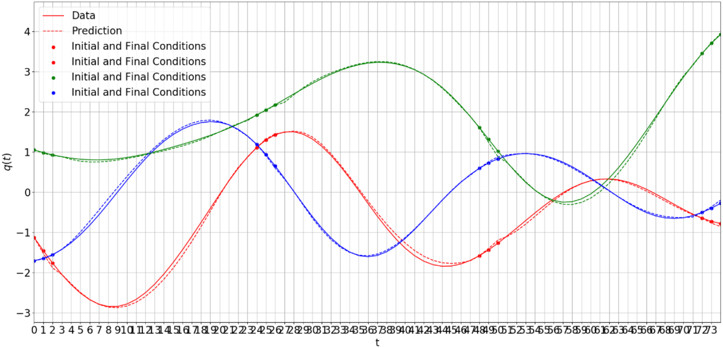
Bi-Directional Interpolation of Differential Equation

- Unidirectional LSTM architecture for Benzene MD trajectory interpolating over 20 time steps



Bi-Directional Interpolation of Differential Equation

- Bidirectional LSTM architecture for Benzene MD trajectory interpolating over 20 time steps
- Final condition and additional bidirectional training smooth trajectories significantely



Bi-Directional Interpolation of Differential Equation

- Single initial and final condition already good for sufficient performance by bidirectional LSTM