Stat-PINNs Readme

October 2024

1 Overview

This is the readme file of the code for Statistical-Physics-Informed Neural Networks (Stat-PINNs). This machine learning framework can learn the dissipative operator and free energy (or entropy) of a pure dissipative GENERIC-type isothermal (or closed) system. Following will introduce the related codes and files.

2 Stat-PINNs (Folder 'NN')

This part is included in the folder 'NN'. It contains the Stat-PINNs python code '`Stat-PINNs.ipynb`' and input data for examples shown in the manuscript, which is included in folder '`Data`'.

2.1 `Stat-PINNs.ipynb`

This code, Stat-PINNs, is coded in python (primarily using JAX) can learn the dissipative operator and free energy from data, which includes the fluctuations and evolution within a short time interval (repeated by a large number of time) for a pure dissipative GENERIC-type isothermal system, and can predict the corresponding evolution in continuum scale. It includes the following parts,

1. Preparation: Import libraries and define functions.
2. Stat-PINNs: MLP and two neural networks for learning dissipative operator and free energy, respectively.
3. Code for Continuum Evolution Prediction
4. Load data for Stat-PINNs training: Import the data (measured dissipative operator entries, short time evolution and related parameters) from MATLAB files '`AllSummaryParams.mat`' and '`AllSummariesMPK2.mat`' in folder '`NN/Data/xxx`', where '`xxx`' is the folder name of a given simulation/experiment.
5. K_NN Training and Prediction: Define training data, train neural network for dissipative operator and output training results in file '`NN/Data/xxx/K_fe_NN_model/K_params.pkl`'.
6. fe_NN Training and Prediction: Define training data, train neural network for free energy and output training results in file '`NN/Data/xxx/K_fe_NN_model/fe_params.pkl`'.
7. Continuum Prediction: Predict continuum evolution from both NN results and the long-rang analytic Arrhenius diffusion model. Then compare with the long-time Kinetic Monte-Carlo (KMC) simulation results loaded from folder '`NN/Data/yyy`', where '`yyy`' is the folder name of a given long-time KMC simulation/experiment.

2.2 Folder '`NN/Data`'

This folder contains three examples for Arrhenius process used in the manuscript, which can be found in following three folders,

```
NN/Data/Arr_Nb2e3_G25_JL0.9_L40
NN/Data/Arr_Nb2e3_G25_JL0.9_L2
NN/Data/Arr_Nb2e3_G25_JL2.2_L2
```

and three corresponding long-time evolution profiles obtained by Kinetic Monte-Carlo (KMC) simulations,

`NN/Data/ArrL_Nb2e3_G25_600t_c05A045f1_JL0.9_L40_1e3T0.32`
`NN/Data/ArrL_Nb2e3_G25_600t_c05A045f1_JL0.9_L2_1e3T0.32`
`NN/Data/ArrL_Nb1333_G17_600t_c05A04f1_JL2.2_L2_1e3T1.8`

Here the names of each folder are defined by following notations:

- `JL`: Interaction strength $J_0 L$;
- `L`: Interaction range L (in each direction) in the unit of bin number;
- `Nb`: Bin number N_b;
- `G`: N_γ, the number of shape function $\gamma_i x$;
- `600t`: The average profile is obtained from 600 independent trajectories (realizations);
- `c05A045f1`: The initial profile has average density (concentration) $\rho_{ave} = 0.5$, amplitude $A = 0.45$ and frequency $f = 1$;
- `1e3T0.32`: The total simulation time is $T = 0.32$ with 1001 steps of output at $t_i = iT/1000$, where $i = 0, 1, \cdots, 1000$.

Notice that systems with different scales/discretizations are involved in training and prediction, and rescaling are done to match them with each other.

- All training data in folders `Arr_xxx` are obtained from simulation in system ranging in $x \in [0, 1]$ with 2000 bins and 25 shape functions. The corresponding training and prediction for operator and free energy are done for the same system. After doing a rescaling, these numbers and figures will be the same as the ones shown in the manuscript, where the system is ranging in $x \in [0, 1]$ with 4000 bins and 50 shape functions.
- When predicting continuum evolution, the code rescale all variables to the system ranging in $x \in [0, 1]$ with 4000 bins and 50 shape functions in according to the manuscript. This rescaling is achieved through a variable called '`model_range_factor`' in the code.
- All long-time KMC simulations are done in system with periodic boundary condition ranging in $xin[0, 1]$ with $N_b = 2000$ and $N_\gamma = 25$, or with $N_b = 1333$ and $N_\gamma = 17$. In order to rescale them to the one used in the manuscript, these profiles are duplicated 2 times for system with $N_b = 2000$, and 3 times for system with $N_b = 1333$, then rescaled in length and time. This rescaling process is achieved though a variable called '`scale_factor_LongKMC`' in the code.