User guide

- Here is an example for Part 5 (Sparse KFs). Cells 1-7 contain the necessary functions, while cells 8-13 contain the main code. Cell 13 is specifically used for plotting and saving the results. Each cell is thoroughly commented to provide clarity and understanding.
- By configuring the `expconfig_field` in cell 7, this code can achieve various functionalities, such as using different types of loss functions, selecting a normalization method, adjusting the learning rate, specifying the length of training data, and choosing different dynamical systems, among other options. For example,

```
expconfig_field = collections.OrderedDict({
    "seed" : [42,],         # set random seeds
    "device" : [torch.device("cuda:0" if torch.cuda.is_available() else "cpu"),],
 # decide calculation by CPU or GPU
    "exp_name" : ['exp_test0606',],     # experimential name
    "exp_path" : ['./exp_results/',],
    "system_name" : ["Lorenz"],         # name of dynamical system
    "model_name" : ['RegularKF',],      # kernel learning method
    "normalized_type" : ['zscore',],    # normalization type, including 'zscore'
and 'minmax' normalization
    "batch_size" : [100,],              # number of samples for calculating the
denominator of $rho$ in the KFs algorithm
    "input_len" : [2,],                 # value of $tau$
    "pred_len" : [1,],                  # prediction length
    "init_ratio" : [0.33,],             # calulation for inverse matrix
    "train_ratio" : [0.80,],            # proportion of training set
    "learning_rate" : [1e-2,],          # learning rate
    "is_warmup" : [True,],              # whether to use warmup
    "seq_lenplot" : [3000,],            # number of plot
    "noise_type" : ['Gaussian_noise',], # noise type
    "noise_pct" : [0,],                 # noise level (percentage)
    "kf_metric" : ["rho"],              # the type of loss function in KFs, such
as rho and hausdorff
    "Gamma" : [0.05,],                  # hyper-paramter of L1 regularization,
gamma = 0 means regular kfs
    "Lambda" : [0,],                    # hyper-parameter of L2 regularization
})
```

where "kf_metric": ["rho"] and "Gamma": [0.05,] indicate the usage of Sparse KFs with a lasso parameter set to 0.05.

- The "data" folder includes pre-computed trajectories for 133 dynamical systems, which are obtained from the source https://github.com/williamgilpin/dysts. By changing the value of the "system_name" variable, you can observe the performance of SKFs for different systems.
- `kernel_zoo' lists more kernel functions