DiseaseModelPINN_notebook_noSoftAdapt_emphasisOnPDE

December 17, 2022

[ ]:
```
import deepxde as dde
from deepxde.backend import pytorch
import torch
import matplotlib.pyplot as plt
import numpy as np

from SIRD_deepxde_DiseaseModel import SIRD_deepxde_net
from DiseaseModel import SIR,SIRD,SIRDIm,SIRDImRel, SIRDImRelSimple, SIRD2Var,␣
↪SIRD2VarRelSimple, GeneralModelSolver
from Plot import Plot

# %matplotlib widget

seed = 2
np.random.seed(seed)
dde.config.set_random_seed(seed)
```

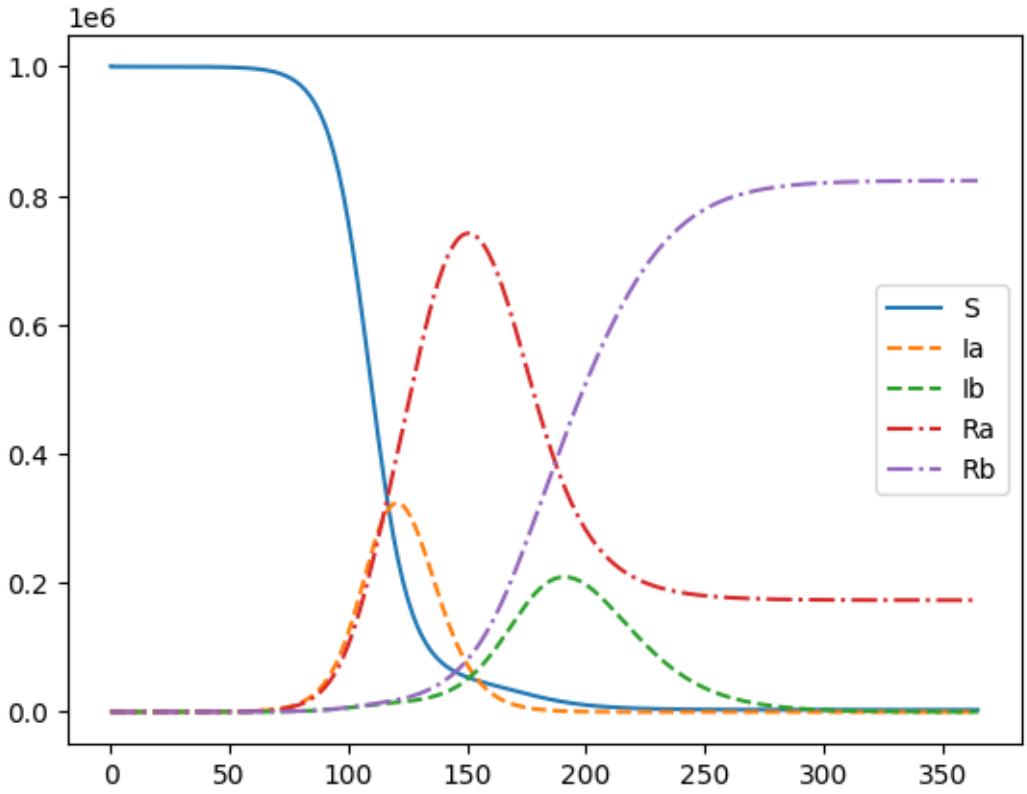
```
Using backend: pytorch


default Torch device: cpu
```

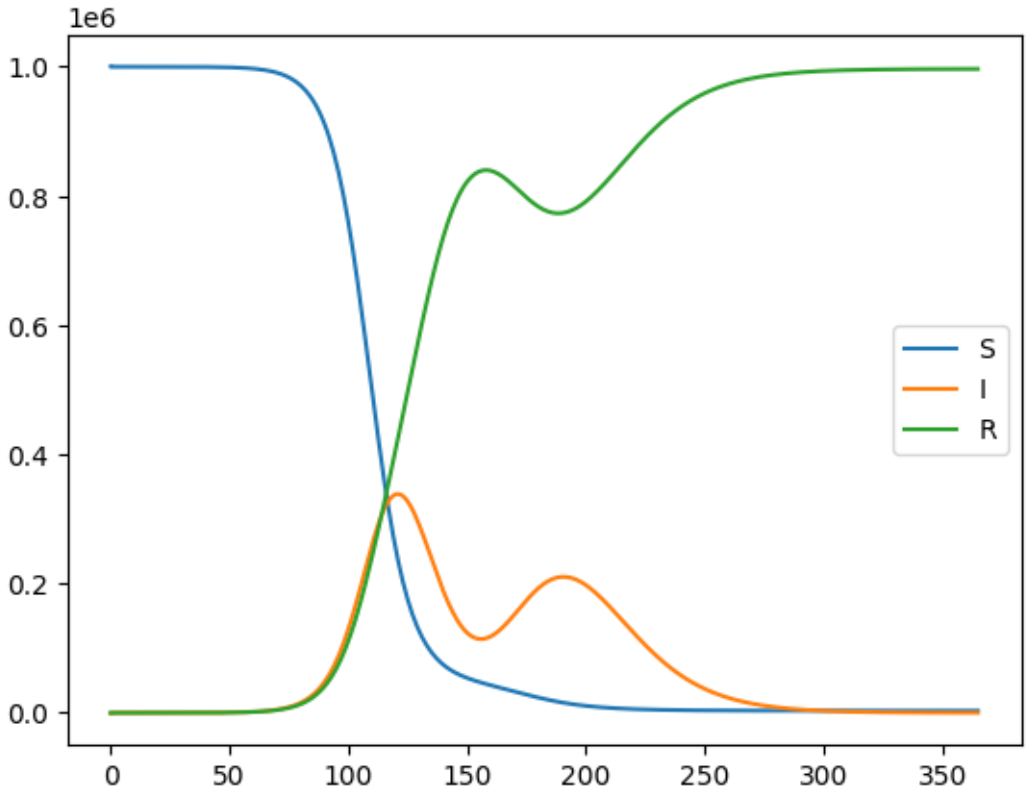
[ ]:
```
time_delta = [0,365] # use three values here for intro time of second variant

# initial_conditions = {
#     "S": 1000000,
#     "I": 1,
#     "R": 0,
#     }
# static_parameters = {
#     "alpha": (0.15),
#     "beta": (0.07),
#     }
# sird_model = SIR(initial_conditions, static_parameters, time_delta)

# initial_conditions = {
#     "S": 1000000,
#     "I": 1,
```

```
#     "R": 0,
#     "D": 0,
#     }
# static_parameters = {
#     "alpha": (0.2),
#     "beta": (0.05),
#     "gamma": (0.001),
#     }
# sird_model = SIRD(initial_conditions, static_parameters, time_delta)

# initial_conditions = {
#     "S": 1000000,
#     "I": 15,
#     "R": 0,
#     "D": 0,
#     "Im": 0, # should be between 0 and 1
#     }
# static_parameters = {
#     "alpha": 0.12,
#     "beta": 0.07,
#     "gamma": 0.02,
#     "kappa": 0.2,
#     }
# sird_model = SIRDIm(initial_conditions, static_parameters, time_delta)
# initial_conditions = {
#     "S": 1000000,
#     "I": 15,
#     "R": 0,
#     "D": 0,
#     "Im": 0, # should be between 0 and 1
#     }
# static_parameters = {
#     "lambda_": 1.5,
#     "gamma": 0.000,
#     "kappa": 0.2,
#     }
# sird_model = SIRDImRel(initial_conditions, static_parameters, time_delta)
# initial_conditions = {
#     "S": 1000000,
#     "I": 15,
#     "R": 0,
#     "Im": 0, # should be between 0 and 1
#     }
# static_parameters = {
#     "lambda_": 1.5,
#     "kappa": 0.2,
#     }
```

```
# sird_model = SIRDImRelSimple(initial_conditions, static_parameters,␣
 ↪time_delta)

# initial_conditions = {
#     "S": 1000000,
#     "Ia": 1,
#     "Ib": 0,
#     "Ra": 0,
#     "Rb": 0,
#     "D": 0,
#     "Im_a": 0, # should be between 0 and 1
#     "Im_b": 0, # should be between 0 and 1
#     }
# static_parameters = {
#     "alpha_a": 0.23  ,
#     "alpha_b": 0.18,
#     "beta_a": 0.1,
#     "beta_b": 0.08,
#     "gamma_a": 0.00,
#     "gamma_b": 0.00,
#     "kappa_a": 0.8,
#     "kappa_b": 0.5,
#     }
# static_parameters = {
#     "alpha_a": 0.11,
#     "alpha_b": 0.12,
#     "beta_a": 0.08,
#     "beta_b": 0.08,
#     "gamma_a": 0.00,
#     "gamma_b": 0.00,
#     "kappa_a": 0.1,
#     "kappa_b": 0.2,
#     }
# sird_model = SIRD2Var(initial_conditions, static_parameters, time_delta)

initial_conditions = {
    "S": 1000000,
    "Ia": 1,
    "Ib": 0,
    "Ra": 0,
    "Rb": 0,
    "Im_a": 0,
    "Im_b": 0,
    }
static_parameters = {
    "lambda_a": 2.2  ,
    "lambda_b": 1.9,
```

```
    "kappa_a": 0.8, # should be between 0 and 1
    "kappa_b": 0.5,
    }

sird_model = SIRD2VarRelSimple(initial_conditions, static_parameters,␣
 ↪time_delta)


t_synth, solution_synth_full = sird_model.simulate()
t_synth, solution_synth = sird_model.get_solution_as_sird()
sird_model.plot_solution()
sird_model.plot_sird()
```





[ ]:
```
print(sird_model)
```

```
A Disease Model with description: 'A model that simulates two concurrent
diseases and natural herd immunity as a factor of the amount of recovered for
each variant':
Parameters:
        lambda_a = 2.2
        lambda_b = 1.9
        kappa_a = 0.8
        kappa_b = 0.5

PDE groups and initial conditions:
        S = 1000000
        Ia = 1
        Ib = 0
        Ra = 0
        Rb = 0
        Im_a = 0
        Im_b = 0

PDE equations:
```

```
        dS/dt = -(0.1*lambda_a/N)*Ia*S - (0.1*lambda_b/N)*Ib*S
        dIa/dt = (0.1*lambda_a/N)*S*Ia + (0.1*lambda_a/N)*(1 - Im_a)*(Ra +
Rb)*Ia - 0.1*Ia
        dIb/dt = (0.1*lambda_b/N)*S*Ib + (0.1*lambda_b/N)*(1 - Im_b)*(Ra +
Rb)*Ib - 0.1*Ib
        dRa/dt = 0.1*Ia - (0.1*lambda_a/N)*(1 - (Im_a))*(Ra)*(Ia) -
(0.1*lambda_b/N)*(1 - (Im_b))*(Ra)*(Ib)
        dRb/dt = 0.1*Ib - (0.1*lambda_a/N)*(1 - (Im_a))*(Rb)*(Ia) -
(0.1*lambda_b/N)*(1 - (Im_b))*(Rb)*(Ib)
        dIm_a/dt = kappa_a*0.1*Ia/N
        dIm_b/dt = kappa_b*0.1*Ib/N
PINN PDE loss equations:
        dS_t - (-(0.1*lambda_a/N)*Ia*S - (0.1*lambda_b/N)*Ib*S)
        dIa_t - ((0.1*lambda_a/N)*S*Ia + (0.1*lambda_a/N)*(1 - Im_a)*(Ra +
Rb)*Ia - 0.1*Ia)
        dIb_t - ((0.1*lambda_b/N)*S*Ib + (0.1*lambda_b/N)*(1 - Im_b)*(Ra +
Rb)*Ib - 0.1*Ib )
        dRa_t - (0.1*Ia - (0.1*lambda_a/N)*(1 - (Im_a))*(Ra)*(Ia) -
(0.1*lambda_b/N)*(1 - (Im_b))*(Ra)*(Ib))
        dRb_t - (0.1*Ib - (0.1*lambda_a/N)*(1 - (Im_a))*(Rb)*(Ia) -
(0.1*lambda_b/N)*(1 - (Im_b))*(Rb)*(Ib))
        dIm_a_t - (kappa_a*0.1*Ia/N)
        dIm_b_t - (kappa_b*0.1*Ib/N)
```

[ ]:
```
# keep this even if not subsetting
t = t_synth
wsol = solution_synth

solver = GeneralModelSolver(sird_model)
# subset
# max_timestep = 300
# t_bool = t_synth < max_timestep
# t = t_synth[t_bool]
# wsol = wsol_synth[t_bool]
```

[ ]:
```
model = SIRD_deepxde_net(t, wsol,disease_model=sird_model, with_neumann=True,␣
 ↪model_name="diseasemodel_test", with_softadapt=False)
print(model)
hyper_print_every = 100
static_loss_weights = []
for key in (model.PDE_names + list(model.loss_points_dict.keys())):
    if "d" in key and "_t" in key:
        w = 1
    else:
        w = 0.2
    static_loss_weights.append(w)
```

```
model.init_model(lr=0.01, print_every=hyper_print_every, activation="tanh",␣
 ↪loss="MSE", nn_layers=1, nn_layer_width=32, loss_weights=static_loss_weights)
```

```
PINN model:
Parameters: ['lambda_a', 'lambda_b', 'kappa_a', 'kappa_b']
Loss measures: ['dS_t', 'dIa_t', 'dIb_t', 'dRa_t', 'dRb_t', 'dIm_a_t',
'dIm_b_t', 'ic_Im_a', 'ic_Im_b', 'ic_S', 'observe_S', 'observe_I', 'observe_R',
'observe_SUM', 'L2_norm_full_Ia', 'L2_norm_full_Ib', 'sign_Ia', 'sign_Ib',
'sign_Ra', 'sign_Rb', 'sign_Im_a', 'sign_Im_b', 'smooth_S', 'smooth_Ia',
'smooth_Ib', 'smooth_Ra', 'smooth_Rb', 'smooth_Im_a', 'smooth_Im_b',
'ic_neumann_S', 'ic_neumann_Ia', 'ic_neumann_Ib', 'ic_neumann_Ra',
'ic_neumann_Rb', 'ic_neumann_Im_a', 'ic_neumann_Im_b', 'fc_neumann_S',
'fc_neumann_Ia', 'fc_neumann_Ib', 'fc_neumann_Ra', 'fc_neumann_Rb',
'fc_neumann_Im_a', 'fc_neumann_Im_b']
Compiling model…
'compile' took 0.000060 s
```

[ ]:
```
prev_best_step = 0
iters = 0
plot_every=2000
```

[ ]:
```
TOTAL_ITER = 100_000
try:
    while True:
        # for n in range(TOTAL_ITER//plot_every):
        model.train_model(iterations=plot_every, print_every=hyper_print_every,␣
 ↪use_LBFGSB=False)
        params_nn, best_step = model.get_best_params(out_func=np.exp) #␣
 ↪parameters need to be extracted with the exponential functino as they have␣
 ↪been modelled in logspace
        if best_step > prev_best_step:
            prev_best_step = best_step
            break
        elif iters >= TOTAL_ITER:
            break
except KeyboardInterrupt:
    print("Training ended prematurely")

params_nn, best_step = model.get_best_params(out_func=np.exp) # parameters need␣
 ↪to be extracted with the exponential functino as they have been modelled in␣
 ↪logspace
print(static_parameters, sep="\n")
t_nn_param, wsol_nn_param, wsol_sird_nn_param = solver(*params_nn)
# params_nn= tuple(np.exp([*params_nn]))
# print(*params_nn)
model.set_synthetic_data(t_synth, solution_synth_full)
```

```
model.set_nn_synthetic_data(t_nn_param, wsol_nn_param, wsol_sird_nn_param)
plot = Plot(model, values_to_plot=sird_model.initial_conditions_keys) # class␣
 ↪that contains plotting functions
plot.show_known_and_prediction()
# plot.plot_param_history()
# plot.plot_loss_history()
```

```
Training model…

Step      Train loss
Test loss
Test metric
14000     [5.50e-06, 1.45e-06, 1.35e-06, 1.16e-06, 1.53e-06, 1.50e-06, 1.24e-06,
5.48e-07, 3.82e-06, 1.22e-06, 1.54e-05, 1.49e-04, 1.23e-04, 1.24e-05, 1.39e-06,
2.42e-06, 4.36e-06, 6.01e-09, 1.23e-06, 4.21e-07, 6.47e-09, 0.00e+00, 0.00e+00,
2.85e-05, 0.00e+00, 8.60e-06, 2.85e-06, 4.56e-08, 0.00e+00, 3.73e-08, 3.22e-06,
4.81e-06, 2.14e-06, 7.60e-07, 3.87e-07, 8.59e-08, 5.66e-11, 4.82e-10, 9.01e-10,
5.02e-10, 1.53e-09, 1.82e-09, 5.89e-09]    [5.50e-06, 1.45e-06, 1.35e-06,
1.16e-06, 1.53e-06, 1.50e-06, 1.24e-06, 5.48e-07, 3.82e-06, 1.22e-06, 1.54e-05,
1.49e-04, 1.23e-04, 1.24e-05, 1.39e-06, 2.42e-06, 4.36e-06, 6.01e-09, 1.23e-06,
4.21e-07, 6.47e-09, 0.00e+00, 0.00e+00, 2.85e-05, 0.00e+00, 8.60e-06, 2.85e-06,
4.56e-08, 0.00e+00, 3.73e-08, 3.22e-06, 4.81e-06, 2.14e-06, 7.60e-07, 3.87e-07,
8.59e-08, 5.66e-11, 4.82e-10, 9.01e-10, 5.02e-10, 1.53e-09, 1.82e-09, 5.89e-09]
[]
14100     [1.75e-06, 9.38e-07, 7.77e-07, 6.89e-07, 1.86e-06, 8.41e-07, 1.49e-06,
4.47e-10, 2.98e-10, 1.22e-09, 6.97e-06, 1.36e-04, 1.03e-04, 3.28e-07, 1.56e-06,
2.39e-06, 2.14e-07, 2.23e-07, 2.09e-09, 1.65e-09, 0.00e+00, 0.00e+00, 0.00e+00,
4.40e-09, 1.14e-09, 1.10e-08, 1.02e-08, 0.00e+00, 0.00e+00, 3.71e-09, 1.88e-10,
7.56e-11, 7.23e-10, 2.08e-11, 3.13e-11, 1.36e-13, 3.66e-11, 3.23e-10, 5.83e-10,
3.05e-10, 9.52e-10, 1.17e-09, 3.70e-09]    [1.75e-06, 9.38e-07, 7.77e-07,
6.89e-07, 1.86e-06, 8.41e-07, 1.49e-06, 4.47e-10, 2.98e-10, 1.22e-09, 6.97e-06,
1.36e-04, 1.03e-04, 3.28e-07, 1.56e-06, 2.39e-06, 2.14e-07, 2.23e-07, 2.09e-09,
1.65e-09, 0.00e+00, 0.00e+00, 0.00e+00, 4.40e-09, 1.14e-09, 1.10e-08, 1.02e-08,
0.00e+00, 0.00e+00, 3.71e-09, 1.88e-10, 7.56e-11, 7.23e-10, 2.08e-11, 3.13e-11,
1.36e-13, 3.66e-11, 3.23e-10, 5.83e-10, 3.05e-10, 9.52e-10, 1.17e-09, 3.70e-09]
[]
14200     [1.07e-06, 2.53e-06, 8.40e-07, 7.06e-07, 3.05e-06, 1.29e-06, 2.00e-05,
1.27e-05, 7.37e-08, 1.17e-06, 9.43e-06, 1.39e-04, 1.08e-04, 5.76e-06, 1.96e-06,
2.20e-06, 1.64e-07, 4.68e-07, 4.69e-08, 0.00e+00, 2.45e-07, 4.01e-10, 0.00e+00,
2.58e-07, 3.08e-07, 3.31e-07, 0.00e+00, 1.73e-06, 2.83e-09, 8.08e-07, 5.81e-07,
4.17e-07, 1.34e-07, 5.06e-07, 5.90e-07, 4.20e-06, 4.89e-11, 5.29e-10, 6.50e-10,
4.39e-10, 1.21e-09, 1.60e-09, 4.00e-09]    [1.07e-06, 2.53e-06, 8.40e-07,
7.06e-07, 3.05e-06, 1.29e-06, 2.00e-05, 1.27e-05, 7.37e-08, 1.17e-06, 9.43e-06,
1.39e-04, 1.08e-04, 5.76e-06, 1.96e-06, 2.20e-06, 1.64e-07, 4.68e-07, 4.69e-08,
0.00e+00, 2.45e-07, 4.01e-10, 0.00e+00, 2.58e-07, 3.08e-07, 3.31e-07, 0.00e+00,
1.73e-06, 2.83e-09, 8.08e-07, 5.81e-07, 4.17e-07, 1.34e-07, 5.06e-07, 5.90e-07,
4.20e-06, 4.89e-11, 5.29e-10, 6.50e-10, 4.39e-10, 1.21e-09, 1.60e-09, 4.00e-09]
```

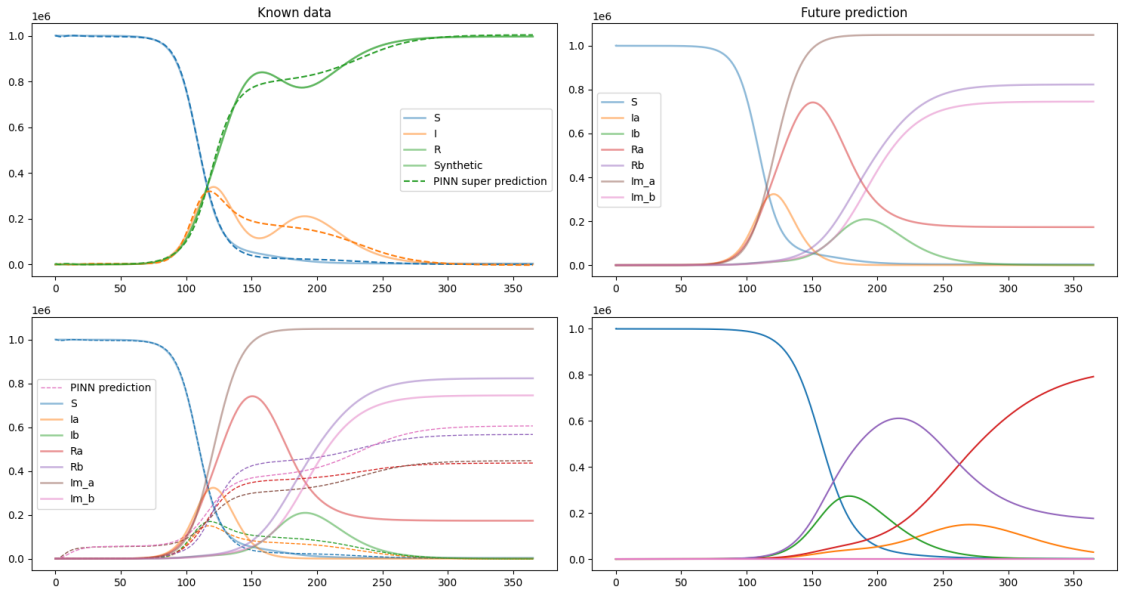
[]
14300 [5.66e-06, 1.68e-06, 1.49e-06, 1.21e-06, 2.07e-06, 4.34e-07, 2.47e-06,
2.83e-07, 6.43e-07, 3.31e-08, 3.74e-05, 2.32e-04, 2.03e-04, 4.29e-04, 2.01e-06,
2.62e-06, 1.55e-08, 0.00e+00, 5.55e-08, 0.00e+00, 1.59e-09, 3.89e-09, 0.00e+00,
1.09e-07, 0.00e+00, 3.92e-07, 0.00e+00, 1.12e-08, 2.74e-08, 2.03e-06, 2.32e-06,
6.03e-06, 4.64e-06, 2.29e-06, 2.20e-07, 2.77e-08, 7.28e-12, 7.40e-11, 1.38e-10,
7.80e-11, 2.32e-10, 3.00e-10, 8.78e-10] [5.66e-06, 1.68e-06, 1.49e-06,
1.21e-06, 2.07e-06, 4.34e-07, 2.47e-06, 2.83e-07, 6.43e-07, 3.31e-08, 3.74e-05,
2.32e-04, 2.03e-04, 4.29e-04, 2.01e-06, 2.62e-06, 1.55e-08, 0.00e+00, 5.55e-08,
0.00e+00, 1.59e-09, 3.89e-09, 0.00e+00, 1.09e-07, 0.00e+00, 3.92e-07, 0.00e+00,
1.12e-08, 2.74e-08, 2.03e-06, 2.32e-06, 6.03e-06, 4.64e-06, 2.29e-06, 2.20e-07,
2.77e-08, 7.28e-12, 7.40e-11, 1.38e-10, 7.80e-11, 2.32e-10, 3.00e-10, 8.78e-10]
[]
14400 [2.28e-06, 1.22e-06, 9.26e-07, 7.92e-07, 2.05e-06, 6.38e-07, 1.57e-06,
9.13e-10, 6.55e-09, 4.76e-08, 7.05e-06, 1.31e-04, 9.79e-05, 4.84e-06, 1.62e-06,
2.36e-06, 6.13e-08, 1.49e-07, 7.34e-10, 3.36e-09, 3.38e-12, 7.81e-11, 0.00e+00,
6.76e-09, 4.58e-08, 5.18e-09, 2.37e-08, 2.38e-11, 5.51e-10, 6.30e-10, 4.26e-08,
3.60e-08, 1.12e-10, 1.79e-08, 1.00e-08, 2.00e-08, 2.17e-11, 1.95e-10, 3.35e-10,
1.91e-10, 5.45e-10, 6.72e-10, 2.05e-09] [2.28e-06, 1.22e-06, 9.26e-07,
7.92e-07, 2.05e-06, 6.38e-07, 1.57e-06, 9.13e-10, 6.55e-09, 4.76e-08, 7.05e-06,
1.31e-04, 9.79e-05, 4.84e-06, 1.62e-06, 2.36e-06, 6.13e-08, 1.49e-07, 7.34e-10,
3.36e-09, 3.38e-12, 7.81e-11, 0.00e+00, 6.76e-09, 4.58e-08, 5.18e-09, 2.37e-08,
2.38e-11, 5.51e-10, 6.30e-10, 4.26e-08, 3.60e-08, 1.12e-10, 1.79e-08, 1.00e-08,
2.00e-08, 2.17e-11, 1.95e-10, 3.35e-10, 1.91e-10, 5.45e-10, 6.72e-10, 2.05e-09]
[]
14500 [2.17e-06, 1.11e-06, 9.01e-07, 7.79e-07, 1.94e-06, 8.27e-07, 1.50e-06,
1.14e-10, 1.50e-10, 1.30e-10, 7.00e-06, 1.28e-04, 9.51e-05, 1.63e-07, 1.57e-06,
2.35e-06, 2.09e-07, 2.02e-07, 3.67e-09, 3.49e-09, 0.00e+00, 0.00e+00, 0.00e+00,
8.09e-09, 7.98e-09, 2.36e-08, 2.40e-08, 0.00e+00, 0.00e+00, 4.63e-09, 1.84e-09,
7.55e-10, 1.22e-12, 1.57e-09, 1.14e-10, 1.88e-10, 1.95e-11, 1.73e-10, 3.00e-10,
1.69e-10, 4.86e-10, 6.06e-10, 1.83e-09] [2.17e-06, 1.11e-06, 9.01e-07,
7.79e-07, 1.94e-06, 8.27e-07, 1.50e-06, 1.14e-10, 1.50e-10, 1.30e-10, 7.00e-06,
1.28e-04, 9.51e-05, 1.63e-07, 1.57e-06, 2.35e-06, 2.09e-07, 2.02e-07, 3.67e-09,
3.49e-09, 0.00e+00, 0.00e+00, 0.00e+00, 8.09e-09, 7.98e-09, 2.36e-08, 2.40e-08,
0.00e+00, 0.00e+00, 4.63e-09, 1.84e-09, 7.55e-10, 1.22e-12, 1.57e-09, 1.14e-10,
1.88e-10, 1.95e-11, 1.73e-10, 3.00e-10, 1.69e-10, 4.86e-10, 6.06e-10, 1.83e-09]
[]
14600 [2.30e-06, 2.81e-06, 1.96e-06, 1.21e-06, 3.76e-06, 8.79e-07, 1.10e-06,
1.40e-08, 1.32e-06, 1.05e-06, 9.68e-06, 1.45e-04, 1.14e-04, 9.27e-06, 1.55e-06,
2.44e-06, 5.17e-06, 4.40e-07, 6.01e-09, 0.00e+00, 9.51e-06, 0.00e+00, 0.00e+00,
2.42e-05, 6.98e-07, 3.11e-08, 0.00e+00, 4.94e-05, 0.00e+00, 3.43e-06, 9.25e-06,
5.17e-06, 3.83e-07, 9.82e-06, 4.75e-06, 3.46e-07, 1.17e-11, 1.01e-10, 1.79e-10,
1.06e-10, 3.06e-10, 3.78e-10, 1.14e-09] [2.30e-06, 2.81e-06, 1.96e-06,
1.21e-06, 3.76e-06, 8.79e-07, 1.10e-06, 1.40e-08, 1.32e-06, 1.05e-06, 9.68e-06,
1.45e-04, 1.14e-04, 9.27e-06, 1.55e-06, 2.44e-06, 5.17e-06, 4.40e-07, 6.01e-09,
0.00e+00, 9.51e-06, 0.00e+00, 0.00e+00, 2.42e-05, 6.98e-07, 3.11e-08, 0.00e+00,
4.94e-05, 0.00e+00, 3.43e-06, 9.25e-06, 5.17e-06, 3.83e-07, 9.82e-06, 4.75e-06,
3.46e-07, 1.17e-11, 1.01e-10, 1.79e-10, 1.06e-10, 3.06e-10, 3.78e-10, 1.14e-09]

```
[]
14700     [2.08e-06, 1.08e-06, 1.01e-06, 8.31e-07, 2.02e-06, 1.68e-06, 1.30e-06,
3.20e-09, 2.55e-09, 1.94e-09, 6.96e-06, 1.24e-04, 9.11e-05, 1.25e-07, 1.57e-06,
2.38e-06, 1.35e-07, 1.68e-07, 2.02e-09, 3.19e-09, 0.00e+00, 0.00e+00, 0.00e+00,
2.02e-09, 1.61e-09, 1.42e-08, 2.25e-08, 0.00e+00, 0.00e+00, 1.54e-08, 2.51e-10,
7.71e-09, 7.74e-09, 6.31e-09, 5.65e-10, 7.32e-09, 1.35e-11, 1.16e-10, 2.09e-10,
1.16e-10, 3.32e-10, 3.97e-10, 1.30e-09]    [2.08e-06, 1.08e-06, 1.01e-06,
8.31e-07, 2.02e-06, 1.68e-06, 1.30e-06, 3.20e-09, 2.55e-09, 1.94e-09, 6.96e-06,
1.24e-04, 9.11e-05, 1.25e-07, 1.57e-06, 2.38e-06, 1.35e-07, 1.68e-07, 2.02e-09,
3.19e-09, 0.00e+00, 0.00e+00, 0.00e+00, 2.02e-09, 1.61e-09, 1.42e-08, 2.25e-08,
0.00e+00, 0.00e+00, 1.54e-08, 2.51e-10, 7.71e-09, 7.74e-09, 6.31e-09, 5.65e-10,
7.32e-09, 1.35e-11, 1.16e-10, 2.09e-10, 1.16e-10, 3.32e-10, 3.97e-10, 1.30e-09]
[]
14800     [2.51e-06, 1.17e-06, 1.01e-06, 9.49e-07, 2.39e-06, 9.47e-07, 1.30e-06,
9.25e-11, 5.90e-09, 2.27e-07, 1.37e-05, 1.74e-04, 1.34e-04, 2.67e-04, 1.91e-06,
2.80e-06, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00,
0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 2.34e-07, 7.59e-08,
2.51e-07, 3.55e-07, 3.65e-07, 1.20e-09, 9.81e-09, 1.09e-11, 9.50e-11, 1.70e-10,
9.28e-11, 2.64e-10, 3.36e-10, 1.03e-09]    [2.51e-06, 1.17e-06, 1.01e-06,
9.49e-07, 2.39e-06, 9.47e-07, 1.30e-06, 9.25e-11, 5.90e-09, 2.27e-07, 1.37e-05,
1.74e-04, 1.34e-04, 2.67e-04, 1.91e-06, 2.80e-06, 0.00e+00, 0.00e+00, 0.00e+00,
0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00,
0.00e+00, 0.00e+00, 2.34e-07, 7.59e-08, 2.51e-07, 3.55e-07, 3.65e-07, 1.20e-09,
9.81e-09, 1.09e-11, 9.50e-11, 1.70e-10, 9.28e-11, 2.64e-10, 3.36e-10, 1.03e-09]
[]
14900     [2.17e-06, 1.24e-06, 9.94e-07, 7.36e-07, 2.17e-06, 7.88e-07, 1.56e-06,
4.93e-08, 2.46e-09, 5.82e-09, 7.83e-06, 1.20e-04, 8.80e-05, 1.62e-06, 1.61e-06,
2.33e-06, 7.65e-08, 2.59e-07, 1.15e-09, 3.90e-08, 1.98e-10, 0.00e+00, 0.00e+00,
1.53e-08, 3.74e-07, 8.10e-09, 2.40e-07, 1.39e-09, 0.00e+00, 2.13e-09, 1.28e-07,
1.04e-07, 1.73e-10, 6.45e-08, 8.49e-08, 9.95e-10, 8.68e-12, 7.86e-11, 1.36e-10,
7.93e-11, 2.22e-10, 2.86e-10, 8.42e-10]    [2.17e-06, 1.24e-06, 9.94e-07,
7.36e-07, 2.17e-06, 7.88e-07, 1.56e-06, 4.93e-08, 2.46e-09, 5.82e-09, 7.83e-06,
1.20e-04, 8.80e-05, 1.62e-06, 1.61e-06, 2.33e-06, 7.65e-08, 2.59e-07, 1.15e-09,
3.90e-08, 1.98e-10, 0.00e+00, 0.00e+00, 1.53e-08, 3.74e-07, 8.10e-09, 2.40e-07,
1.39e-09, 0.00e+00, 2.13e-09, 1.28e-07, 1.04e-07, 1.73e-10, 6.45e-08, 8.49e-08,
9.95e-10, 8.68e-12, 7.86e-11, 1.36e-10, 7.93e-11, 2.22e-10, 2.86e-10, 8.42e-10]
[]
15000     [2.02e-06, 1.10e-06, 8.88e-07, 7.57e-07, 2.00e-06, 9.45e-07, 1.67e-06,
1.17e-10, 1.95e-10, 2.19e-10, 7.05e-06, 1.18e-04, 8.58e-05, 4.84e-08, 1.59e-06,
2.37e-06, 1.21e-07, 1.14e-07, 1.57e-09, 2.01e-09, 4.02e-13, 0.00e+00, 0.00e+00,
7.06e-09, 7.35e-09, 1.11e-08, 1.42e-08, 2.84e-12, 0.00e+00, 4.72e-09, 1.81e-09,
8.62e-10, 5.50e-10, 5.12e-09, 3.11e-10, 5.53e-10, 7.82e-12, 6.83e-11, 1.19e-10,
6.85e-11, 1.90e-10, 2.52e-10, 7.22e-10]    [2.02e-06, 1.10e-06, 8.88e-07,
7.57e-07, 2.00e-06, 9.45e-07, 1.67e-06, 1.17e-10, 1.95e-10, 2.19e-10, 7.05e-06,
1.18e-04, 8.58e-05, 4.84e-08, 1.59e-06, 2.37e-06, 1.21e-07, 1.14e-07, 1.57e-09,
2.01e-09, 4.02e-13, 0.00e+00, 0.00e+00, 7.06e-09, 7.35e-09, 1.11e-08, 1.42e-08,
2.84e-12, 0.00e+00, 4.72e-09, 1.81e-09, 8.62e-10, 5.50e-10, 5.12e-09, 3.11e-10,
5.53e-10, 7.82e-12, 6.83e-11, 1.19e-10, 6.85e-11, 1.90e-10, 2.52e-10, 7.22e-10]
```

[]
15100 [2.06e-05, 3.21e-06, 2.15e-06, 4.32e-06, 1.99e-06, 1.28e-06, 2.01e-06,
2.84e-06, 5.67e-08, 1.82e-05, 7.93e-05, 4.34e-04, 2.24e-04, 3.37e-04, 2.33e-06,
2.70e-06, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 5.57e-10, 0.00e+00,
0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 3.93e-09, 7.11e-06, 1.25e-05,
5.95e-08, 4.19e-11, 1.46e-06, 4.56e-10, 5.01e-10, 2.24e-11, 1.05e-10, 1.74e-10,
8.35e-11, 2.49e-10, 3.75e-10, 1.06e-09] [2.06e-05, 3.21e-06, 2.15e-06,
4.32e-06, 1.99e-06, 1.28e-06, 2.01e-06, 2.84e-06, 5.67e-08, 1.82e-05, 7.93e-05,
4.34e-04, 2.24e-04, 3.37e-04, 2.33e-06, 2.70e-06, 0.00e+00, 0.00e+00, 0.00e+00,
0.00e+00, 0.00e+00, 5.57e-10, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00,
0.00e+00, 3.93e-09, 7.11e-06, 1.25e-05, 5.95e-08, 4.19e-11, 1.46e-06, 4.56e-10,
5.01e-10, 2.24e-11, 1.05e-10, 1.74e-10, 8.35e-11, 2.49e-10, 3.75e-10, 1.06e-09]
[]
15200 [1.55e-06, 1.04e-06, 9.09e-07, 7.71e-07, 2.05e-06, 2.96e-06, 1.66e-06,
6.73e-09, 5.08e-08, 1.62e-08, 7.12e-06, 1.15e-04, 8.28e-05, 4.22e-07, 1.59e-06,
2.40e-06, 7.75e-08, 4.71e-08, 2.12e-09, 9.74e-11, 0.00e+00, 0.00e+00, 0.00e+00,
1.71e-08, 9.46e-09, 1.49e-08, 6.88e-10, 0.00e+00, 0.00e+00, 4.39e-08, 5.67e-09,
3.01e-09, 2.51e-09, 4.47e-09, 2.71e-08, 4.99e-08, 5.63e-12, 4.85e-11, 8.47e-11,
4.91e-11, 1.35e-10, 1.66e-10, 5.15e-10] [1.55e-06, 1.04e-06, 9.09e-07,
7.71e-07, 2.05e-06, 2.96e-06, 1.66e-06, 6.73e-09, 5.08e-08, 1.62e-08, 7.12e-06,
1.15e-04, 8.28e-05, 4.22e-07, 1.59e-06, 2.40e-06, 7.75e-08, 4.71e-08, 2.12e-09,
9.74e-11, 0.00e+00, 0.00e+00, 0.00e+00, 1.71e-08, 9.46e-09, 1.49e-08, 6.88e-10,
0.00e+00, 0.00e+00, 4.39e-08, 5.67e-09, 3.01e-09, 2.51e-09, 4.47e-09, 2.71e-08,
4.99e-08, 5.63e-12, 4.85e-11, 8.47e-11, 4.91e-11, 1.35e-10, 1.66e-10, 5.15e-10]
[]
15300 [2.01e-06, 1.10e-06, 9.13e-07, 7.50e-07, 2.02e-06, 1.19e-06, 1.68e-06,
1.48e-10, 2.28e-10, 3.20e-10, 7.16e-06, 1.14e-04, 8.13e-05, 3.98e-08, 1.60e-06,
2.36e-06, 9.30e-08, 8.31e-08, 7.33e-10, 7.44e-10, 0.00e+00, 0.00e+00, 0.00e+00,
3.99e-09, 4.32e-09, 5.17e-09, 5.25e-09, 0.00e+00, 0.00e+00, 4.10e-09, 9.23e-10,
3.87e-10, 2.91e-10, 2.99e-09, 6.65e-10, 1.03e-09, 4.68e-12, 4.07e-11, 7.02e-11,
4.11e-11, 1.11e-10, 1.47e-10, 4.17e-10] [2.01e-06, 1.10e-06, 9.13e-07,
7.50e-07, 2.02e-06, 1.19e-06, 1.68e-06, 1.48e-10, 2.28e-10, 3.20e-10, 7.16e-06,
1.14e-04, 8.13e-05, 3.98e-08, 1.60e-06, 2.36e-06, 9.30e-08, 8.31e-08, 7.33e-10,
7.44e-10, 0.00e+00, 0.00e+00, 0.00e+00, 3.99e-09, 4.32e-09, 5.17e-09, 5.25e-09,
0.00e+00, 0.00e+00, 4.10e-09, 9.23e-10, 3.87e-10, 2.91e-10, 2.99e-09, 6.65e-10,
1.03e-09, 4.68e-12, 4.07e-11, 7.02e-11, 4.11e-11, 1.11e-10, 1.47e-10, 4.17e-10]
[]
15400 [2.22e-06, 2.72e-06, 4.03e-06, 8.42e-07, 4.21e-06, 1.30e-06, 1.16e-06,
1.68e-06, 6.91e-07, 9.31e-06, 2.64e-05, 1.81e-04, 2.29e-04, 1.34e-05, 2.05e-06,
2.88e-06, 2.13e-06, 0.00e+00, 7.71e-07, 5.05e-06, 1.09e-08, 0.00e+00, 0.00e+00,
1.50e-05, 0.00e+00, 3.18e-06, 2.96e-05, 7.70e-08, 0.00e+00, 2.34e-07, 3.13e-06,
1.67e-05, 3.71e-07, 3.57e-06, 2.02e-07, 4.65e-08, 1.29e-11, 8.85e-11, 1.60e-10,
1.00e-10, 2.73e-10, 3.87e-10, 1.04e-09] [2.22e-06, 2.72e-06, 4.03e-06,
8.42e-07, 4.21e-06, 1.30e-06, 1.16e-06, 1.68e-06, 6.91e-07, 9.31e-06, 2.64e-05,
1.81e-04, 2.29e-04, 1.34e-05, 2.05e-06, 2.88e-06, 2.13e-06, 0.00e+00, 7.71e-07,
5.05e-06, 1.09e-08, 0.00e+00, 0.00e+00, 1.50e-05, 0.00e+00, 3.18e-06, 2.96e-05,
7.70e-08, 0.00e+00, 2.34e-07, 3.13e-06, 1.67e-05, 3.71e-07, 3.57e-06, 2.02e-07,
4.65e-08, 1.29e-11, 8.85e-11, 1.60e-10, 1.00e-10, 2.73e-10, 3.87e-10, 1.04e-09]

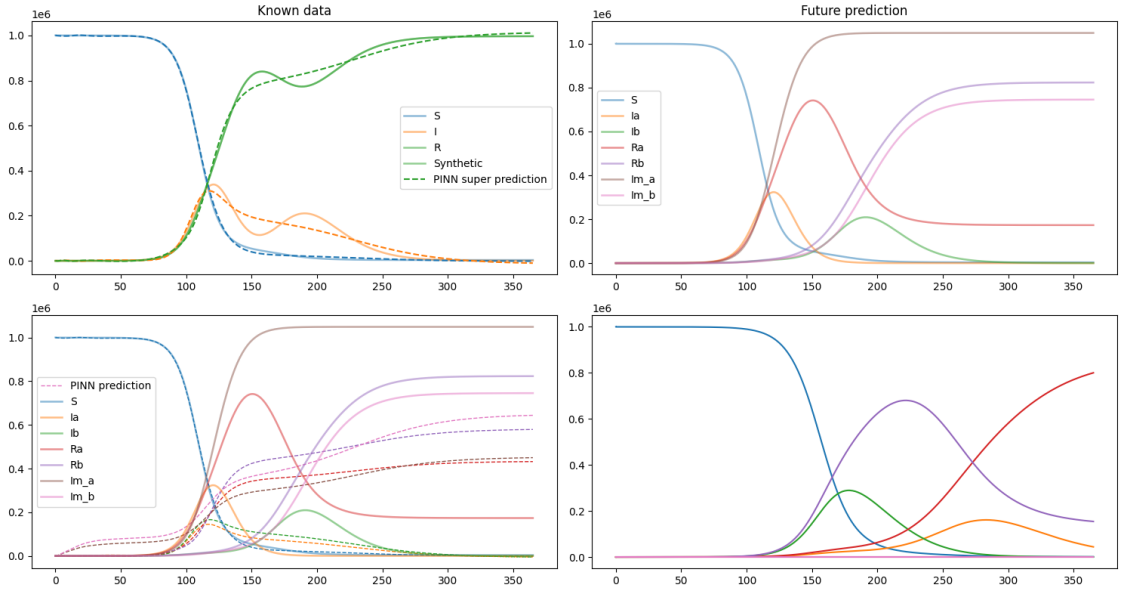
[]
15500 [1.93e-06, 1.08e-06, 8.93e-07, 7.37e-07, 2.07e-06, 1.06e-06, 1.69e-06,
3.99e-10, 6.96e-10, 5.24e-10, 7.20e-06, 1.11e-04, 7.84e-05, 4.63e-08, 1.60e-06,
2.37e-06, 6.68e-08, 6.85e-08, 2.82e-10, 4.19e-10, 0.00e+00, 0.00e+00, 0.00e+00,
7.10e-10, 8.07e-09, 1.99e-09, 2.96e-09, 0.00e+00, 0.00e+00, 5.30e-09, 7.23e-11,
1.28e-12, 3.94e-11, 4.08e-09, 5.50e-12, 5.93e-10, 3.08e-12, 2.78e-11, 4.83e-11,
2.85e-11, 7.54e-11, 1.07e-10, 2.76e-10] [1.93e-06, 1.08e-06, 8.93e-07,
7.37e-07, 2.07e-06, 1.06e-06, 1.69e-06, 3.99e-10, 6.96e-10, 5.24e-10, 7.20e-06,
1.11e-04, 7.84e-05, 4.63e-08, 1.60e-06, 2.37e-06, 6.68e-08, 6.85e-08, 2.82e-10,
4.19e-10, 0.00e+00, 0.00e+00, 0.00e+00, 7.10e-10, 8.07e-09, 1.99e-09, 2.96e-09,
0.00e+00, 0.00e+00, 5.30e-09, 7.23e-11, 1.28e-12, 3.94e-11, 4.08e-09, 5.50e-12,
5.93e-10, 3.08e-12, 2.78e-11, 4.83e-11, 2.85e-11, 7.54e-11, 1.07e-10, 2.76e-10]
[]
15600 [1.79e-06, 1.04e-06, 8.47e-07, 7.13e-07, 2.04e-06, 1.11e-06, 1.82e-06,
1.14e-10, 1.80e-10, 3.30e-10, 7.21e-06, 1.09e-04, 7.71e-05, 3.06e-08, 1.60e-06,
2.38e-06, 6.66e-08, 6.07e-08, 4.85e-10, 3.27e-10, 0.00e+00, 0.00e+00, 0.00e+00,
3.05e-09, 5.65e-09, 3.42e-09, 2.31e-09, 0.00e+00, 0.00e+00, 3.51e-09, 6.60e-10,
2.25e-11, 1.62e-10, 1.09e-09, 2.99e-10, 5.48e-10, 2.63e-12, 2.35e-11, 4.09e-11,
2.42e-11, 6.37e-11, 9.20e-11, 2.29e-10] [1.79e-06, 1.04e-06, 8.47e-07,
7.13e-07, 2.04e-06, 1.11e-06, 1.82e-06, 1.14e-10, 1.80e-10, 3.30e-10, 7.21e-06,
1.09e-04, 7.71e-05, 3.06e-08, 1.60e-06, 2.38e-06, 6.66e-08, 6.07e-08, 4.85e-10,
3.27e-10, 0.00e+00, 0.00e+00, 0.00e+00, 3.05e-09, 5.65e-09, 3.42e-09, 2.31e-09,
0.00e+00, 0.00e+00, 3.51e-09, 6.60e-10, 2.25e-11, 1.62e-10, 1.09e-09, 2.99e-10,
5.48e-10, 2.63e-12, 2.35e-11, 4.09e-11, 2.42e-11, 6.37e-11, 9.20e-11, 2.29e-10]
[]
15700 [6.22e-06, 5.94e-06, 8.39e-06, 1.19e-06, 3.50e-06, 1.96e-05, 1.60e-06,
2.05e-06, 7.32e-06, 4.35e-05, 9.17e-05, 3.92e-04, 1.91e-04, 5.22e-04, 1.39e-06,
1.84e-06, 5.96e-06, 2.04e-05, 0.00e+00, 0.00e+00, 0.00e+00, 1.64e-07, 0.00e+00,
2.34e-08, 7.48e-05, 0.00e+00, 0.00e+00, 0.00e+00, 1.16e-06, 1.19e-07, 8.52e-06,
5.67e-05, 1.24e-06, 6.77e-06, 1.27e-04, 2.30e-07, 7.26e-14, 7.60e-13, 1.30e-12,
6.41e-13, 1.97e-12, 3.48e-12, 8.46e-12] [6.22e-06, 5.94e-06, 8.39e-06,
1.19e-06, 3.50e-06, 1.96e-05, 1.60e-06, 2.05e-06, 7.32e-06, 4.35e-05, 9.17e-05,
3.92e-04, 1.91e-04, 5.22e-04, 1.39e-06, 1.84e-06, 5.96e-06, 2.04e-05, 0.00e+00,
0.00e+00, 0.00e+00, 1.64e-07, 0.00e+00, 2.34e-08, 7.48e-05, 0.00e+00, 0.00e+00,
0.00e+00, 1.16e-06, 1.19e-07, 8.52e-06, 5.67e-05, 1.24e-06, 6.77e-06, 1.27e-04,
2.30e-07, 7.26e-14, 7.60e-13, 1.30e-12, 6.41e-13, 1.97e-12, 3.48e-12, 8.46e-12]
[]
15800 [1.71e-06, 1.02e-06, 1.08e-06, 8.33e-07, 2.08e-06, 3.38e-06, 1.48e-06,
9.61e-10, 2.88e-09, 1.36e-09, 7.20e-06, 1.07e-04, 7.43e-05, 5.24e-08, 1.55e-06,
2.44e-06, 3.56e-08, 6.16e-08, 3.46e-09, 2.77e-09, 5.24e-12, 0.00e+00, 0.00e+00,
7.35e-09, 1.30e-09, 2.44e-08, 1.95e-08, 3.70e-11, 0.00e+00, 1.50e-08, 2.43e-10,
2.02e-09, 9.55e-09, 3.37e-08, 3.52e-09, 7.98e-10, 1.70e-12, 1.43e-11, 2.79e-11,
1.56e-11, 4.13e-11, 4.92e-11, 1.53e-10] [1.71e-06, 1.02e-06, 1.08e-06,
8.33e-07, 2.08e-06, 3.38e-06, 1.48e-06, 9.61e-10, 2.88e-09, 1.36e-09, 7.20e-06,
1.07e-04, 7.43e-05, 5.24e-08, 1.55e-06, 2.44e-06, 3.56e-08, 6.16e-08, 3.46e-09,
2.77e-09, 5.24e-12, 0.00e+00, 0.00e+00, 7.35e-09, 1.30e-09, 2.44e-08, 1.95e-08,
3.70e-11, 0.00e+00, 1.50e-08, 2.43e-10, 2.02e-09, 9.55e-09, 3.37e-08, 3.52e-09,
7.98e-10, 1.70e-12, 1.43e-11, 2.79e-11, 1.56e-11, 4.13e-11, 4.92e-11, 1.53e-10]

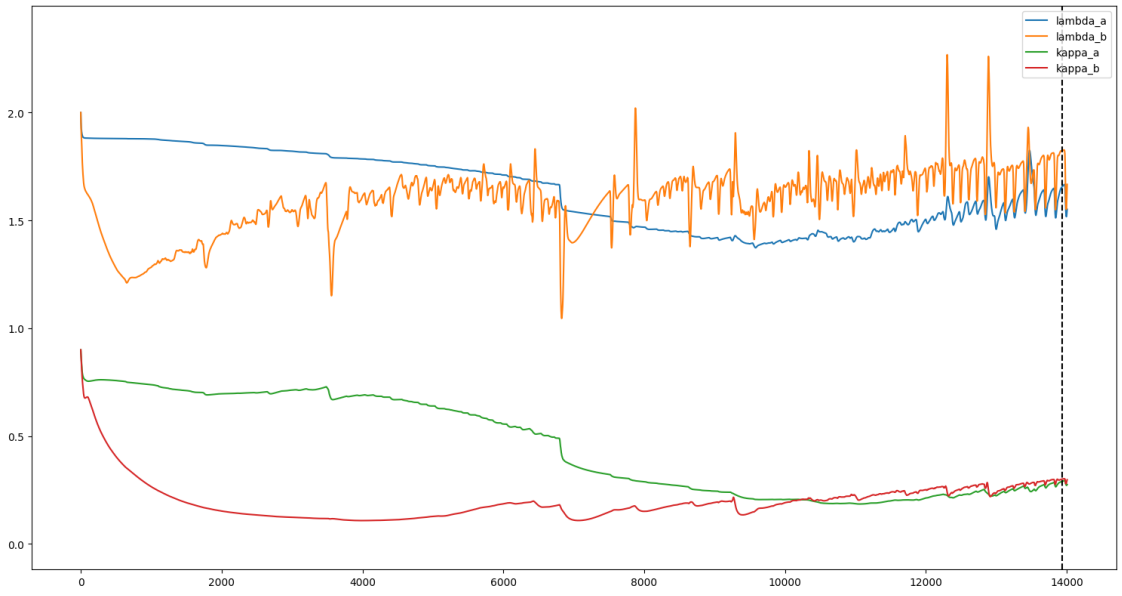
```
[]
15900     [2.03e-06, 1.13e-06, 9.65e-07, 7.63e-07, 2.12e-06, 1.33e-06, 1.63e-06,
2.00e-10, 1.27e-10, 2.45e-10, 7.32e-06, 1.05e-04, 7.32e-05, 4.40e-08, 1.58e-06,
2.40e-06, 4.63e-08, 4.11e-08, 4.88e-09, 3.64e-09, 0.00e+00, 0.00e+00, 0.00e+00,
5.41e-09, 9.67e-09, 3.45e-08, 2.57e-08, 0.00e+00, 0.00e+00, 5.60e-09, 9.76e-11,
9.56e-12, 5.41e-09, 7.99e-09, 2.70e-09, 3.06e-09, 1.48e-12, 1.29e-11, 2.33e-11,
1.37e-11, 3.54e-11, 4.43e-11, 1.24e-10]    [2.03e-06, 1.13e-06, 9.65e-07,
7.63e-07, 2.12e-06, 1.33e-06, 1.63e-06, 2.00e-10, 1.27e-10, 2.45e-10, 7.32e-06,
1.05e-04, 7.32e-05, 4.40e-08, 1.58e-06, 2.40e-06, 4.63e-08, 4.11e-08, 4.88e-09,
3.64e-09, 0.00e+00, 0.00e+00, 0.00e+00, 5.41e-09, 9.67e-09, 3.45e-08, 2.57e-08,
0.00e+00, 0.00e+00, 5.60e-09, 9.76e-11, 9.56e-12, 5.41e-09, 7.99e-09, 2.70e-09,
3.06e-09, 1.48e-12, 1.29e-11, 2.33e-11, 1.37e-11, 3.54e-11, 4.43e-11, 1.24e-10]
[]
16000     [1.91e-06, 1.14e-06, 8.87e-07, 7.32e-07, 2.09e-06, 1.13e-06, 1.83e-06,
1.38e-10, 1.62e-10, 2.65e-10, 7.35e-06, 1.04e-04, 7.21e-05, 3.85e-08, 1.59e-06,
2.40e-06, 4.18e-08, 3.56e-08, 3.49e-09, 2.48e-09, 0.00e+00, 0.00e+00, 0.00e+00,
5.94e-09, 8.35e-09, 2.47e-08, 1.75e-08, 0.00e+00, 0.00e+00, 5.31e-09, 1.39e-10,
7.27e-11, 2.33e-09, 4.77e-09, 4.50e-10, 1.64e-09, 1.25e-12, 1.09e-11, 1.95e-11,
1.17e-11, 2.96e-11, 3.78e-11, 9.94e-11]    [1.91e-06, 1.14e-06, 8.87e-07,
7.32e-07, 2.09e-06, 1.13e-06, 1.83e-06, 1.38e-10, 1.62e-10, 2.65e-10, 7.35e-06,
1.04e-04, 7.21e-05, 3.85e-08, 1.59e-06, 2.40e-06, 4.18e-08, 3.56e-08, 3.49e-09,
2.48e-09, 0.00e+00, 0.00e+00, 0.00e+00, 5.94e-09, 8.35e-09, 2.47e-08, 1.75e-08,
0.00e+00, 0.00e+00, 5.31e-09, 1.39e-10, 7.27e-11, 2.33e-09, 4.77e-09, 4.50e-10,
1.64e-09, 1.25e-12, 1.09e-11, 1.95e-11, 1.17e-11, 2.96e-11, 3.78e-11, 9.94e-11]
[]

Best model at step 16000:
  train loss: 1.97e-04
  test loss: 1.97e-04
  test metric: []

'train' took 45.169404 s

Best train step: 16000
lambda_a: 1.6877328920234462
lambda_b: 1.8191348382341581
kappa_a: 0.3026100773614343
kappa_b: 0.31828287295873003
Best train step: 16000
lambda_a: 1.6877328920234462
lambda_b: 1.8191348382341581
kappa_a: 0.3026100773614343
kappa_b: 0.31828287295873003
{'lambda_a': 2.2, 'lambda_b': 1.9, 'kappa_a': 0.8, 'kappa_b': 0.5}
```

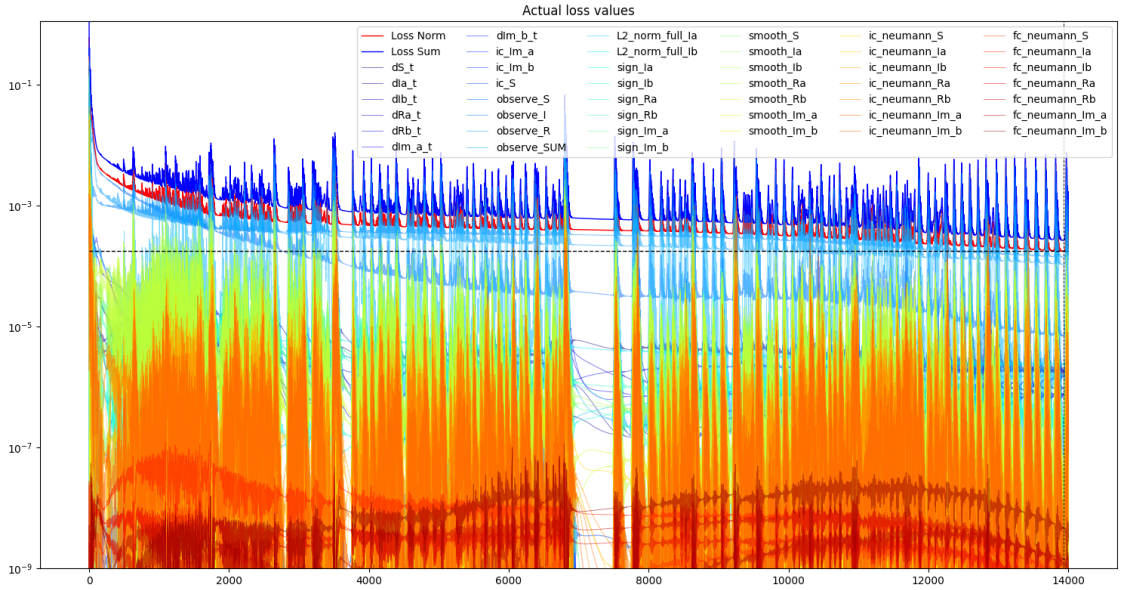


[ ]:
```
params_nn, best_step = model.get_best_params(out_func=np.exp) # parameters need to be extracted with the exponential functino as they have been modelled in logspace
print(static_parameters, sep="\n")
t_nn_param, wsol_nn_param, wsol_sird_nn_param = solver(*params_nn)

model.set_synthetic_data(t_synth, solution_synth_full)
model.set_nn_synthetic_data(t_nn_param, wsol_nn_param, wsol_sird_nn_param)
plot = Plot(model, values_to_plot=sird_model.initial_conditions_keys) # class that contains plotting functions
plot.show_known_and_prediction()
plot.plot_param_history()
plot.plot_loss_history()
```

```
Best train step: 13937
lambda_a: 1.6592537989171274
lambda_b: 1.8241637502154628
kappa_a: 0.2905775899217331
kappa_b: 0.3014625868995771
{'lambda_a': 2.2, 'lambda_b': 1.9, 'kappa_a': 0.8, 'kappa_b': 0.5}
```


Known data
Future prediction
1e6
S
I
R
Synthetic
PINN super prediction
S
Ia
Ib
Ra
Rb
Im_a
Im_b
PINN prediction


lambda_a
lambda_b
kappa_a
kappa_b



[ ]:

[ ]:
```
# from SIRD_normal_nn import SIRD_net
# max_timestep = 100000
# t_bool = t_synth < max_timestep
# t = t_synth[t_bool]
# wsol = solution_synth[t_bool]

# net = SIRD_net(t, wsol, init_num_people=1e6)
# net.train()
# net.plot(t_synth, solution_synth)
```

[ ]:
```
# fig, ax = plt.subplots()
# net.plot(ax, t_synth)
```