Connected to Python 3.10.12

In [ ]:

```
from pathlib import Path

import scimba.nets.training_tools as training_tools
import scimba.pinns.pinn_losses as pinn_losses
import scimba.pinns.pinn_x as pinn_x
import scimba.pinns.training_x as training_x
import scimba.sampling.sampling_parameters as sampling_parameters
import scimba.sampling.sampling_pde as sampling_pde
import scimba.sampling.uniform_sampling as uniform_sampling
import torch
from scimba.equations import domain, pdes

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"torch loaded; device is {device}")

torch.set_default_dtype(torch.double)
torch.set_default_device(device)

PI = 3.14159265358979323846
ELLIPSOID_A = 4 / 3
ELLIPSOID_B = 1 / ELLIPSOID_A


class PoissonDisk2D(pdes.AbstractPDEx):
    def __init__(self, space_domain, rhs='8*pi*pi*sin(2*pi*x)*sin(2*pi*y)', diff='(1,0,0,1)', g='0',):
        super().__init__(
            nb_unknowns=1,
            space_domain=space_domain,
            nb_parameters=1,
            parameter_domain=[[0.5, 1]],
        )

        self.rhs = rhs
        self.diff = diff
        self.g = g
        self.first_derivative = True
        self.second_derivative = True

    def bc_residual(self, w, x, mu, **kwargs):
        u = self.get_variables(w)
        x1, x2 = x.get_coordinates()
        g_evaluated = eval(self.g, {'x': x1, 'y': x2, 'pi': PI, 'sin' : torch.sin})
        return u - g_evaluated

    def residual(self, w, x, mu, **kwargs):
        x1, x2 = x.get_coordinates()
        alpha = self.get_parameters(mu)
        u_xx = self.get_variables(w, "w_xx")
        u_yy = self.get_variables(w, "w_yy")

        f = eval(self.rhs, {'x': x1, 'y': x2, 'pi': PI, 'sin' : torch.sin})
        diff = eval(self.diff)

        return u_xx * diff[0] + u_yy * diff[3] + f


    def reference_solution(self, x, mu):
        x1, x2 = x.get_coordinates()
        x1_0, x2_0 = self.space_domain.large_domain.center
        f = eval(self.rhs, {'x': x1, 'y': x2, 'pi': PI, 'sin' : torch.sin})
        return 0.25 * f * (1 - (x1 - x1_0) ** 2 - (x2 - x2_0) ** 2)


class Poisson_2D(pdes.AbstractPDEx):
    def __init__(self, space_domain,  rhs='8*pi*pi*sin(2*pi*x)*sin(2*pi*y)', diff='(1,0,0,1)', g='0',):
        super().__init__(
            nb_unknowns=1,
            space_domain=space_domain,
            nb_parameters=1,
            parameter_domain=[[0.50000, 0.500001]],
        )
        self.rhs = rhs
        self.diff = diff
        self.g = g
        self.first_derivative = True
        self.second_derivative = True

    def bc_residual(self, w, x, mu, **kwargs):
        u = self.get_variables(w)
```

```
        # Évaluation de la condition aux limites g
        x1, x2 = x.get_coordinates()
        g_evaluated = eval(self.g, {'x': x1, 'y': x2, 'pi': PI, 'sin' : torch.sin})
        return u - g_evaluated

    def residual(self, w, x, mu, **kwargs):
        x1, x2 = x.get_coordinates()
        alpha = self.get_parameters(mu)
        u_xx = self.get_variables(w, "w_xx")
        u_yy = self.get_variables(w, "w_yy")

        f = eval(self.rhs, {'x': x1, 'y': x2, 'pi': PI, 'sin' : torch.sin})
        diff = eval(self.diff)

        return u_xx * diff[0] + u_yy * diff[3] + f


    def post_processing(self, x, mu, w):
        x1, x2 = x.get_coordinates()
        return x1 * (1 - x1) * x2 * (1 - x2) * w

    def reference_solution(self, x, mu):
        x1, x2 = x.get_coordinates()
        alpha = self.get_parameters(mu)
        return eval(self.rhs, {'x': x1, 'y': x2, 'pi': PI, 'sin' : torch.sin})


class Poisson_2D_ellipse(pdes.AbstractPDEx):
    def __init__(self, space_domain):
        super().__init__(
            nb_unknowns=1,
            space_domain=space_domain,
            nb_parameters=0,
            parameter_domain=[[0.99999, 1]],
        )

        self.first_derivative = True
        self.second_derivative = True

    def make_data(self, n_data):
        pass

    def bc_residual(self, w, x, mu, **kwargs):
        u = self.get_variables(w)
        return u

    def residual(self, w, x, mu, **kwargs):
        x1, x2 = x.get_coordinates()
        u_xx = self.get_variables(w, "w_xx")
        u_yy = self.get_variables(w, "w_yy")
        f = 1
        return u_xx + u_yy + f

    def reference_solution(self, x, mu):
        x1, x2 = x.get_coordinates()
        x1_0, x2_0 = self.space_domain.large_domain.center
        a, b = ELLIPSOID_A, ELLIPSOID_B
        rho = 0.5 / (1 / a**2 + 1 / b**2)
        return rho * (1 - ((x1 - x1_0) / a) ** 2 - ((x2 - x2_0) / b) ** 2)


def disk_to_ellipse(x):
    x1, x2 = (x[:, i, None] for i in range(2))
    return torch.cat((x1 * ELLIPSOID_A, x2 * ELLIPSOID_B), axis=1)


def Jacobian_disk_to_ellipse(x):
    x1, x2 = (x[:, i, None] for i in range(2))
    return ELLIPSOID_A, 0, 0, ELLIPSOID_B


def disk_to_potato(x):
    x1, x2 = (x[:, i, None] for i in range(2))
    x = x1 - 0.5 * x2**2 + 0.3 * torch.sin(x2)
    y = x2 + 0.1 * x + 0.12 * torch.cos(x)
    return torch.cat((x, y), axis=1)


def Jacobian_disk_to_potato(x):
    x1, x2 = (x[:, i, None] for i in range(2))
    raise ValueError("Jacobian_disk_to_potato is not implemented")
    return 0, 0, 0, 0
```

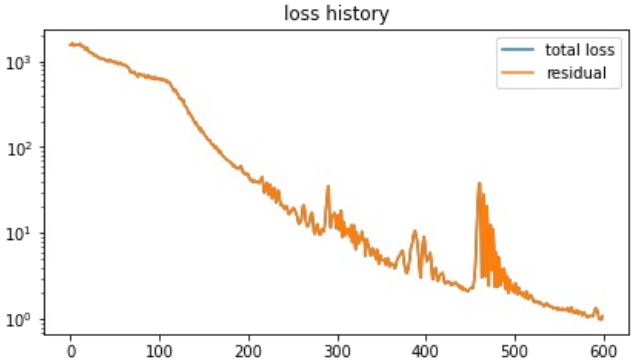
```python
def Run_laplacian2D(pde, bc_loss_bool=False, w_bc=0, w_res=1.0):
    x_sampler = sampling_pde.XSampler(pde=pde)
    mu_sampler = sampling_parameters.MuSampler(
        sampler=uniform_sampling.UniformSampling, model=pde
    )
    sampler = sampling_pde.PdeXCartesianSampler(x_sampler, mu_sampler)

    file_name = "test.pth"
    #new_training = False
    new_training = True

    if new_training:
        (
            Path.cwd()
            / Path(training_x.TrainerPINNSpace.FOLDER_FOR_SAVED_NETWORKS)
            / file_name
        ).unlink(missing_ok=True)

    tlayers = [20, 20, 20, 20, 20]
    network = pinn_x.MLP_x(pde=pde, layer_sizes=tlayers, activation_type="sine")
    pinn = pinn_x.PINNx(network, pde)
    losses = pinn_losses.PinnLossesData(
        bc_loss_bool=bc_loss_bool, w_res=w_res, w_bc=w_bc
    )
    optimizers = training_tools.OptimizerData(learning_rate=1.2e-2, decay=0.99)
    trainer = training_x.TrainerPINNSpace(
        pde=pde,
        network=pinn,
        sampler=sampler,
        losses=losses,
        optimizers=optimizers,
        file_name=file_name,
        batch_size=5000,
    )

    if not bc_loss_bool:
        if new_training:
            trainer.train(epochs=600, n_collocation=5000, n_data=0)
    else:
        if new_training:
            trainer.train(
                epochs=600, n_collocation=5000, n_bc_collocation=1000, n_data=0
            )

    trainer.plot(20000, reference_solution=True)
    # trainer.plot_derivative_mu(n_visu=20000)


if __name__ == "__main__":
    # Laplacien strong Bc on Square with nn
    xdomain = domain.SpaceDomain(2, domain.SquareDomain(2, [[0.0, 1.0], [0.0, 1.0]]))
    print(xdomain)

    pde = Poisson_2D(xdomain,  rhs='8*pi*pi*sin(2*pi*x)*sin(2*pi*y)', g='0')
    Run_laplacian2D(pde)


    pde = Poisson_2D(xdomain, rhs='-1.0-4*y*x+y*y', g='x')
    Run_laplacian2D(pde)

    xdomain = domain.SpaceDomain(2, domain.DiskBasedDomain(2, center=[0.0, 0.0], radius=1.0))
    pde_disk = PoissonDisk2D(space_domain=xdomain)
    Run_laplacian2D(pde_disk)
```

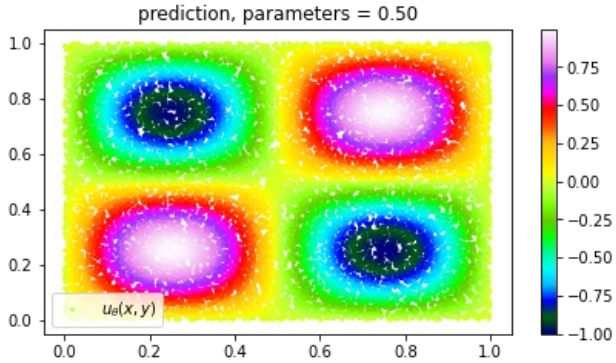
```
Using device: cpu
torch loaded; device is cpu
<scimba.equations.domain.SpaceDomain object at 0x7fe820e43d60>
>> load network /workspaces/2024-m1-scimba-feelpp/networks/test.pth
network was not loaded from file: training needed
epoch     0: current loss = 1.55e+03
epoch     0: best loss = 1.55e+03
epoch     1: best loss = 1.54e+03
epoch     4: best loss = 1.50e+03
epoch    12: best loss = 1.49e+03
epoch    14: best loss = 1.49e+03
epoch    15: best loss = 1.44e+03
epoch    17: best loss = 1.35e+03
epoch    20: best loss = 1.33e+03
epoch    21: best loss = 1.28e+03
epoch    22: best loss = 1.27e+03
epoch    23: best loss = 1.25e+03
epoch    24: best loss = 1.24e+03
epoch    25: best loss = 1.23e+03
```

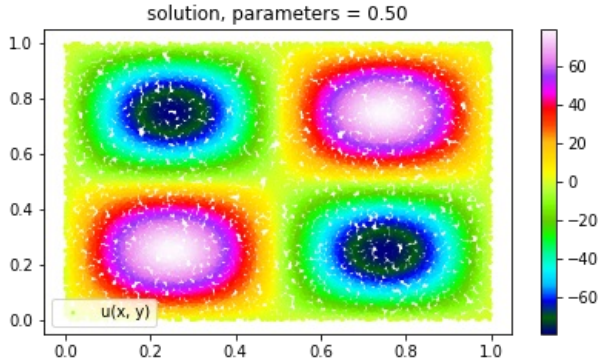
```
epoch    26: best loss = 1.18e+03
epoch    27: best loss = 1.17e+03
epoch    29: best loss = 1.15e+03
epoch    30: best loss = 1.13e+03
epoch    31: best loss = 1.11e+03
epoch    32: best loss = 1.08e+03
epoch    35: best loss = 1.05e+03
epoch    39: best loss = 1.03e+03
epoch    41: best loss = 9.97e+02
epoch    43: best loss = 9.92e+02
epoch    47: best loss = 9.75e+02
epoch    48: best loss = 9.70e+02
epoch    49: best loss = 9.65e+02
epoch    51: best loss = 9.52e+02
epoch    52: best loss = 9.42e+02
epoch    54: best loss = 9.17e+02
epoch    58: best loss = 9.10e+02
epoch    60: best loss = 8.88e+02
epoch    63: best loss = 8.72e+02
epoch    64: best loss = 8.60e+02
epoch    65: best loss = 8.43e+02
epoch    66: best loss = 8.25e+02
epoch    67: best loss = 7.55e+02
epoch    68: best loss = 7.44e+02
epoch    69: best loss = 7.37e+02
epoch    73: best loss = 7.19e+02
epoch    74: best loss = 6.91e+02
epoch    75: best loss = 6.85e+02
epoch    76: best loss = 6.55e+02
epoch    84: best loss = 6.48e+02
epoch    86: best loss = 6.35e+02
epoch    91: best loss = 6.34e+02
epoch    92: best loss = 6.30e+02
epoch    94: best loss = 6.17e+02
epoch    95: best loss = 6.12e+02
epoch    99: best loss = 6.10e+02
epoch   102: best loss = 6.00e+02
epoch   107: best loss = 5.84e+02
epoch   108: best loss = 5.78e+02
epoch   110: best loss = 5.59e+02
epoch   113: best loss = 5.28e+02
epoch   114: best loss = 5.08e+02
epoch   115: best loss = 4.83e+02
epoch   116: best loss = 4.55e+02
epoch   117: best loss = 4.49e+02
epoch   120: best loss = 4.15e+02
epoch   121: best loss = 4.15e+02
epoch   122: best loss = 3.85e+02
epoch   123: best loss = 3.65e+02
epoch   125: best loss = 3.59e+02
epoch   126: best loss = 3.40e+02
epoch   128: best loss = 3.11e+02
epoch   129: best loss = 3.02e+02
epoch   130: best loss = 2.95e+02
epoch   131: best loss = 2.81e+02
epoch   132: best loss = 2.69e+02
epoch   133: best loss = 2.45e+02
epoch   135: best loss = 2.43e+02
epoch   136: best loss = 2.28e+02
epoch   137: best loss = 2.24e+02
epoch   138: best loss = 2.12e+02
epoch   139: best loss = 2.00e+02
epoch   140: best loss = 1.89e+02
epoch   142: best loss = 1.80e+02
epoch   143: best loss = 1.80e+02
epoch   144: best loss = 1.70e+02
epoch   145: best loss = 1.62e+02
epoch   146: best loss = 1.55e+02
epoch   148: best loss = 1.55e+02
epoch   149: best loss = 1.46e+02
epoch   151: best loss = 1.38e+02
epoch   152: best loss = 1.34e+02
epoch   153: best loss = 1.32e+02
epoch   154: best loss = 1.24e+02
epoch   155: best loss = 1.22e+02
epoch   156: best loss = 1.20e+02
epoch   157: best loss = 1.20e+02
epoch   158: best loss = 1.12e+02
epoch   159: best loss = 1.08e+02
epoch   160: best loss = 1.06e+02
epoch   162: best loss = 1.00e+02
epoch   163: best loss = 9.62e+01
epoch   165: best loss = 9.34e+01
```

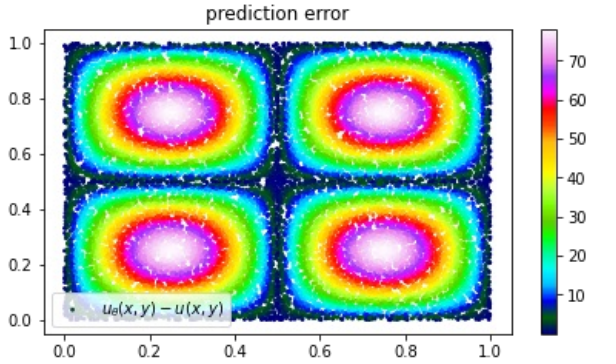
```
epoch   167: best loss = 8.67e+01
epoch   170: best loss = 8.25e+01
epoch   171: best loss = 8.04e+01
epoch   172: best loss = 7.86e+01
epoch   173: best loss = 7.69e+01
epoch   174: best loss = 7.46e+01
epoch   175: best loss = 7.35e+01
epoch   176: best loss = 7.13e+01
epoch   177: best loss = 6.99e+01
epoch   178: best loss = 6.90e+01
epoch   179: best loss = 6.84e+01
epoch   180: best loss = 6.68e+01
epoch   181: best loss = 6.53e+01
epoch   182: best loss = 6.34e+01
epoch   183: best loss = 5.98e+01
epoch   186: best loss = 5.77e+01
epoch   187: best loss = 5.73e+01
epoch   188: best loss = 5.63e+01
epoch   189: best loss = 5.58e+01
epoch   194: best loss = 5.12e+01
epoch   195: best loss = 4.96e+01
epoch   197: best loss = 4.77e+01
epoch   201: best loss = 4.41e+01
epoch   202: best loss = 4.17e+01
epoch   204: best loss = 4.08e+01
epoch   205: best loss = 3.93e+01
epoch   209: best loss = 3.92e+01
epoch   212: best loss = 3.83e+01
epoch   217: best loss = 3.56e+01
epoch   218: best loss = 2.94e+01
epoch   223: best loss = 2.79e+01
epoch   227: best loss = 2.52e+01
epoch   232: best loss = 2.25e+01
epoch   237: best loss = 2.09e+01
epoch   240: best loss = 1.99e+01
epoch   241: best loss = 1.90e+01
epoch   245: best loss = 1.72e+01
epoch   246: best loss = 1.65e+01
epoch   255: best loss = 1.60e+01
epoch   256: best loss = 1.44e+01
epoch   257: best loss = 1.28e+01
epoch   266: best loss = 1.23e+01
epoch   267: best loss = 1.19e+01
epoch   274: best loss = 1.04e+01
epoch   275: best loss = 9.79e+00
epoch   280: best loss = 9.79e+00
epoch   281: best loss = 9.49e+00
epoch   306: best loss = 8.97e+00
epoch   317: best loss = 7.93e+00
epoch   318: best loss = 7.77e+00
epoch   321: best loss = 6.47e+00
epoch   332: best loss = 5.44e+00
epoch   343: best loss = 5.37e+00
epoch   344: best loss = 4.99e+00
epoch   347: best loss = 4.37e+00
epoch   355: best loss = 4.27e+00
epoch   359: best loss = 4.05e+00
epoch   365: best loss = 3.89e+00
epoch   378: best loss = 3.41e+00
epoch   394: best loss = 3.03e+00
epoch   408: best loss = 2.90e+00
epoch   413: best loss = 2.81e+00
epoch   414: best loss = 2.76e+00
epoch   421: best loss = 2.67e+00
epoch   422: best loss = 2.60e+00
epoch   428: best loss = 2.52e+00
epoch   429: best loss = 2.47e+00
epoch   437: best loss = 2.39e+00
epoch   439: best loss = 2.25e+00
epoch   442: best loss = 2.16e+00
epoch   446: best loss = 2.14e+00
epoch   447: best loss = 2.10e+00
epoch   493: best loss = 2.01e+00
epoch   500: current loss = 2.39e+00
epoch   503: best loss = 1.93e+00
epoch   508: best loss = 1.89e+00
epoch   511: best loss = 1.81e+00
epoch   512: best loss = 1.76e+00
epoch   515: best loss = 1.72e+00
epoch   521: best loss = 1.63e+00
epoch   522: best loss = 1.57e+00
epoch   525: best loss = 1.56e+00
epoch   527: best loss = 1.54e+00
```

```
epoch   529: best loss = 1.53e+00
epoch   530: best loss = 1.52e+00
epoch   531: best loss = 1.49e+00
epoch   532: best loss = 1.44e+00
epoch   533: best loss = 1.41e+00
epoch   540: best loss = 1.39e+00
epoch   543: best loss = 1.35e+00
epoch   544: best loss = 1.32e+00
epoch   548: best loss = 1.29e+00
epoch   552: best loss = 1.27e+00
epoch   556: best loss = 1.27e+00
epoch   558: best loss = 1.27e+00
epoch   561: best loss = 1.22e+00
epoch   562: best loss = 1.21e+00
epoch   566: best loss = 1.20e+00
epoch   567: best loss = 1.18e+00
epoch   568: best loss = 1.18e+00
epoch   571: best loss = 1.16e+00
epoch   572: best loss = 1.15e+00
epoch   576: best loss = 1.10e+00
epoch   581: best loss = 1.09e+00
epoch   582: best loss = 1.04e+00
epoch   595: best loss = 1.01e+00
epoch   596: best loss = 1.01e+00
epoch   597: best loss = 9.82e-01
epoch   599: current loss = 1.08e+00
load network: /workspaces/2024-m1-scimba-feelpp/networks/test.pth
jsbdbshd
network loaded
```

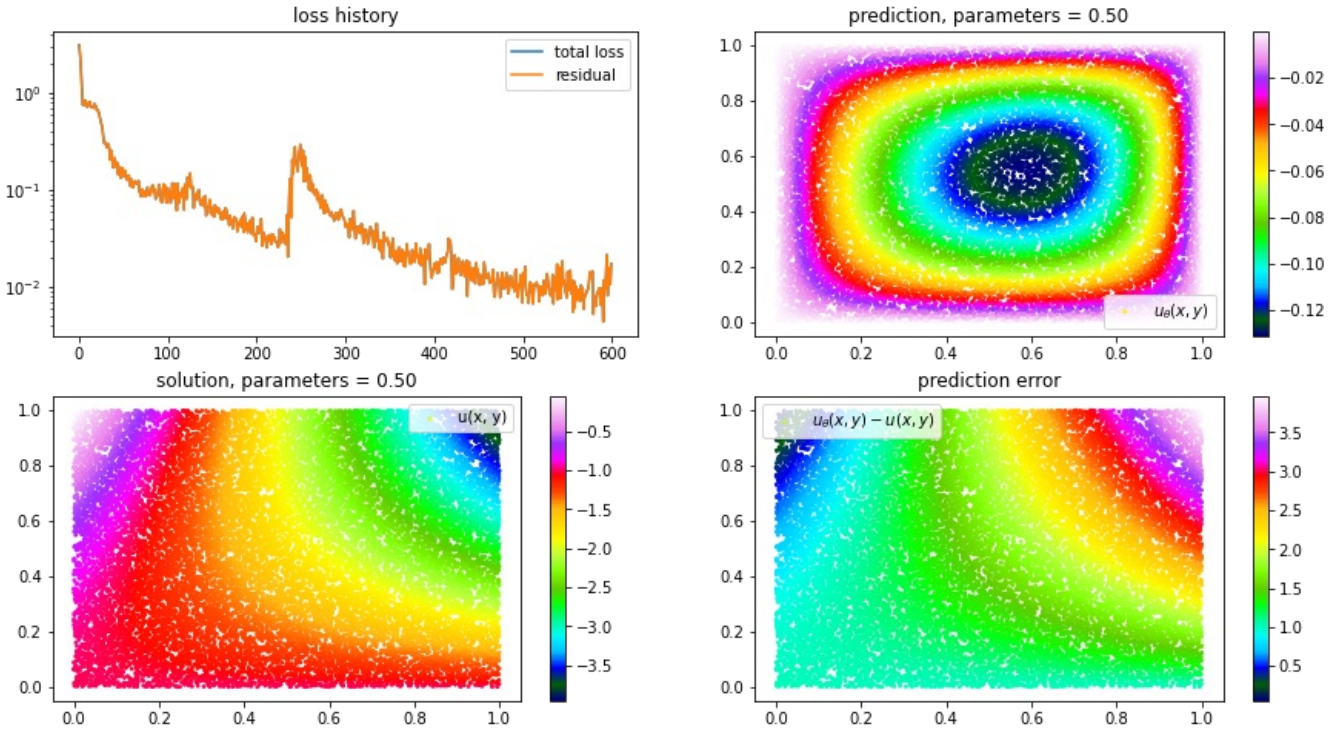








```
>> load network /workspaces/2024-m1-scimba-feelpp/networks/test.pth
network was not loaded from file: training needed
epoch     0: current loss = 3.11e+00
epoch     0: best loss = 3.11e+00
epoch     1: best loss = 2.51e+00
epoch     2: best loss = 1.84e+00
epoch     3: best loss = 1.18e+00
epoch     4: best loss = 7.55e-01
epoch     5: best loss = 7.41e-01
epoch     9: best loss = 7.16e-01
epoch    14: best loss = 7.07e-01
epoch    16: best loss = 7.01e-01
epoch    20: best loss = 6.55e-01
epoch    22: best loss = 6.36e-01
epoch    23: best loss = 5.77e-01
epoch    24: best loss = 5.34e-01
epoch    25: best loss = 4.79e-01
epoch    26: best loss = 4.48e-01
epoch    27: best loss = 3.73e-01
epoch    28: best loss = 3.04e-01
epoch    29: best loss = 3.04e-01
epoch    31: best loss = 2.74e-01
epoch    34: best loss = 2.65e-01
epoch    35: best loss = 2.04e-01
epoch    38: best loss = 1.71e-01
epoch    39: best loss = 1.69e-01
epoch    44: best loss = 1.44e-01
epoch    47: best loss = 1.32e-01
epoch    53: best loss = 1.30e-01
epoch    54: best loss = 1.26e-01
epoch    57: best loss = 1.21e-01
epoch    58: best loss = 1.09e-01
epoch    61: best loss = 1.05e-01
epoch    68: best loss = 9.44e-02
epoch    69: best loss = 9.16e-02
epoch    70: best loss = 8.49e-02
epoch    86: best loss = 8.04e-02
epoch    92: best loss = 7.80e-02
epoch    93: best loss = 7.36e-02
epoch    98: best loss = 7.35e-02
epoch   103: best loss = 7.32e-02
epoch   110: best loss = 6.10e-02
epoch   143: best loss = 5.42e-02
epoch   152: best loss = 5.25e-02
epoch   160: best loss = 4.96e-02
epoch   162: best loss = 4.96e-02
epoch   167: best loss = 4.95e-02
epoch   169: best loss = 4.33e-02
epoch   173: best loss = 3.99e-02
epoch   178: best loss = 3.72e-02
epoch   185: best loss = 3.66e-02
epoch   190: best loss = 3.62e-02
epoch   194: best loss = 3.52e-02
epoch   196: best loss = 3.47e-02
epoch   197: best loss = 3.42e-02
epoch   198: best loss = 3.41e-02
epoch   199: best loss = 2.94e-02
epoch   206: best loss = 2.65e-02
epoch   211: best loss = 2.53e-02
epoch   235: best loss = 2.03e-02
epoch   342: best loss = 1.95e-02
epoch   355: best loss = 1.90e-02
epoch   356: best loss = 1.85e-02
epoch   358: best loss = 1.73e-02
epoch   365: best loss = 1.65e-02
epoch   367: best loss = 1.62e-02
epoch   389: best loss = 1.06e-02
epoch   420: best loss = 9.44e-03
epoch   437: best loss = 8.01e-03
epoch   468: best loss = 7.83e-03
epoch   474: best loss = 7.76e-03
epoch   499: best loss = 6.89e-03
epoch   500: current loss = 1.40e-02
epoch   506: best loss = 6.78e-03
epoch   511: best loss = 6.69e-03
epoch   556: best loss = 5.52e-03
epoch   579: best loss = 5.22e-03
epoch   590: best loss = 4.36e-03
epoch   599: current loss = 1.74e-02
load network: /workspaces/2024-m1-scimba-feelpp/networks/test.pth
jsbdbshd
network loaded
```




```
>> load network /workspaces/2024-m1-scimba-feelpp/networks/test.pth
network was not loaded from file: training needed
epoch     0: current loss = 1.63e+03
epoch     0: best loss = 1.63e+03
epoch     2: best loss = 1.58e+03
epoch     3: best loss = 1.55e+03
epoch     9: best loss = 1.55e+03
epoch    10: best loss = 1.53e+03
epoch    11: best loss = 1.47e+03
epoch    12: best loss = 1.43e+03
epoch    14: best loss = 1.36e+03
epoch    16: best loss = 1.28e+03
epoch    19: best loss = 1.27e+03
epoch    21: best loss = 1.25e+03
epoch    23: best loss = 1.23e+03
epoch    27: best loss = 1.23e+03
epoch    28: best loss = 1.22e+03
epoch    30: best loss = 1.22e+03
epoch    33: best loss = 1.20e+03
epoch    34: best loss = 1.20e+03
epoch    36: best loss = 1.19e+03
epoch    37: best loss = 1.17e+03
epoch    38: best loss = 1.14e+03
epoch    40: best loss = 1.13e+03
epoch    42: best loss = 1.13e+03
epoch    43: best loss = 1.09e+03
epoch    44: best loss = 1.06e+03
epoch    45: best loss = 1.06e+03
epoch    46: best loss = 9.95e+02
epoch    47: best loss = 9.72e+02
epoch    48: best loss = 9.12e+02
epoch    49: best loss = 8.90e+02
epoch    50: best loss = 8.75e+02
epoch    64: best loss = 8.31e+02
epoch    71: best loss = 8.23e+02
epoch    74: best loss = 8.11e+02
epoch    78: best loss = 8.06e+02
epoch    80: best loss = 8.03e+02
epoch    83: best loss = 7.95e+02
epoch    87: best loss = 7.90e+02
epoch    93: best loss = 7.85e+02
epoch   103: best loss = 7.80e+02
epoch   105: best loss = 7.76e+02
epoch   113: best loss = 7.71e+02
epoch   127: best loss = 7.66e+02
epoch   142: best loss = 7.62e+02
epoch   153: best loss = 7.55e+02
epoch   160: best loss = 7.52e+02
epoch   170: best loss = 7.51e+02
epoch   172: best loss = 7.47e+02
epoch   179: best loss = 7.38e+02
epoch   187: best loss = 7.38e+02
epoch   195: best loss = 7.36e+02
```

```
epoch   200: best loss = 7.30e+02
epoch   202: best loss = 7.27e+02
epoch   205: best loss = 7.01e+02
epoch   208: best loss = 6.98e+02
epoch   209: best loss = 6.90e+02
epoch   211: best loss = 6.75e+02
epoch   212: best loss = 6.74e+02
epoch   213: best loss = 6.63e+02
epoch   215: best loss = 6.44e+02
epoch   217: best loss = 6.29e+02
epoch   218: best loss = 6.28e+02
epoch   219: best loss = 5.98e+02
epoch   220: best loss = 5.92e+02
epoch   221: best loss = 5.60e+02
epoch   222: best loss = 5.54e+02
epoch   223: best loss = 5.38e+02
epoch   224: best loss = 5.20e+02
epoch   225: best loss = 5.03e+02
epoch   226: best loss = 4.81e+02
epoch   228: best loss = 4.51e+02
epoch   229: best loss = 4.32e+02
epoch   230: best loss = 4.11e+02
epoch   231: best loss = 4.05e+02
epoch   232: best loss = 3.76e+02
epoch   233: best loss = 3.46e+02
epoch   234: best loss = 3.24e+02
epoch   235: best loss = 3.18e+02
epoch   236: best loss = 2.92e+02
epoch   237: best loss = 2.71e+02
epoch   238: best loss = 2.52e+02
epoch   239: best loss = 2.41e+02
epoch   240: best loss = 2.34e+02
epoch   241: best loss = 2.21e+02
epoch   242: best loss = 2.02e+02
epoch   243: best loss = 1.76e+02
epoch   244: best loss = 1.71e+02
epoch   245: best loss = 1.67e+02
epoch   246: best loss = 1.41e+02
epoch   249: best loss = 1.23e+02
epoch   250: best loss = 1.18e+02
epoch   252: best loss = 1.14e+02
epoch   253: best loss = 1.06e+02
epoch   254: best loss = 1.02e+02
epoch   255: best loss = 1.00e+02
epoch   256: best loss = 8.90e+01
epoch   258: best loss = 8.66e+01
epoch   260: best loss = 8.11e+01
epoch   261: best loss = 7.18e+01
epoch   263: best loss = 6.66e+01
epoch   265: best loss = 5.99e+01
epoch   269: best loss = 5.67e+01
epoch   270: best loss = 5.43e+01
epoch   272: best loss = 5.38e+01
epoch   273: best loss = 5.07e+01
epoch   274: best loss = 4.83e+01
epoch   276: best loss = 4.47e+01
epoch   278: best loss = 4.38e+01
epoch   279: best loss = 4.10e+01
epoch   281: best loss = 3.64e+01
epoch   286: best loss = 3.45e+01
epoch   292: best loss = 3.17e+01
epoch   293: best loss = 2.96e+01
epoch   298: best loss = 2.78e+01
epoch   299: best loss = 2.55e+01
epoch   304: best loss = 2.51e+01
epoch   305: best loss = 2.23e+01
epoch   310: best loss = 2.01e+01
epoch   316: best loss = 1.81e+01
epoch   320: best loss = 1.74e+01
epoch   322: best loss = 1.69e+01
epoch   323: best loss = 1.68e+01
epoch   325: best loss = 1.61e+01
epoch   326: best loss = 1.53e+01
epoch   327: best loss = 1.51e+01
epoch   329: best loss = 1.41e+01
epoch   332: best loss = 1.36e+01
epoch   333: best loss = 1.29e+01
epoch   342: best loss = 1.22e+01
epoch   349: best loss = 1.16e+01
epoch   355: best loss = 1.02e+01
epoch   360: best loss = 9.85e+00
epoch   361: best loss = 9.75e+00
epoch   368: best loss = 8.66e+00
```

```
epoch   374: best loss = 8.14e+00
epoch   381: best loss = 7.45e+00
epoch   390: best loss = 6.93e+00
epoch   392: best loss = 6.86e+00
epoch   401: best loss = 6.73e+00
epoch   402: best loss = 5.90e+00
epoch   406: best loss = 5.86e+00
epoch   424: best loss = 5.70e+00
epoch   425: best loss = 5.11e+00
epoch   435: best loss = 5.10e+00
epoch   442: best loss = 4.06e+00
epoch   450: best loss = 4.04e+00
epoch   456: best loss = 3.43e+00
epoch   466: best loss = 3.31e+00
epoch   476: best loss = 3.25e+00
epoch   480: best loss = 3.15e+00
epoch   487: best loss = 2.90e+00
epoch   500: current loss = 1.60e+01
epoch   531: best loss = 2.65e+00
epoch   532: best loss = 2.62e+00
epoch   533: best loss = 2.34e+00
epoch   536: best loss = 2.16e+00
epoch   540: best loss = 2.11e+00
epoch   541: best loss = 2.05e+00
epoch   551: best loss = 1.83e+00
epoch   562: best loss = 1.78e+00
epoch   563: best loss = 1.71e+00
epoch   574: best loss = 1.55e+00
epoch   580: best loss = 1.53e+00
epoch   585: best loss = 1.47e+00
epoch   599: current loss = 1.55e+00
load network: /workspaces/2024-m1-scimba-feelpp/networks/test.pth
jsbdbshd
network loaded
```