LSS applied to backbone generation of Alanine dipeptide (ADP)

GitHub: https://github.com/Ferg-Lab/LSS

Paper: @article{sidky2020molecular, title={Molecular latent space simulators}, author={Sidky, Hythem and Chen, Wei and Ferguson, Andrew L}, journal={Chemical Science}, volume={11}, number={35}, pages={9459--9467}, year={2020}, publisher={Royal Society of Chemistry} }

Allocating GPU accelerator (~1 min)

```
import torch
```

```
if torch.cuda.is_available():
  print('GPU available')
else:
  print('Please set GPU via Edit -> Notebook Settings.')
```

```
GPU available
```

```
device= 'cuda' if torch.cuda.is_available() else 'cpu'
device
```

```
'cuda'
```

Uploading files (~3 mins)

(i) alanine-dipeptide-0-250ns-nowater.xtc

(ii) alanine-dipeptide-nowater.pdb

N.B. If file upload fails, try using alternate upload means by clicking on file icon in left menu and directly uploading to colab session storage or by uploading to and mounting Google Drive

```
from google.colab import files
files.upload()
```

```
Browse... alanine-dipeptide-0-250ns-nowater.xtc
alanine-dipeptide-0-250ns-nowater.xtc(n/a) - 42909936 bytes, last modified: n/a - 100% done
Saving alanine-dipeptide-0-250ns-nowater.xtc to alanine-dipeptide-0-250ns-nowater.xtc
```

Install necessary packages (~10 mins)

```
%pip install numpy scipy pandas scikit-learn jupyter ipywidgets==7.7.2 widgetsnbextension jupyter_contrib_nbex
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.25.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.11.4)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.0.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting ipywidgets==7.7.2
  Downloading ipywidgets-7.7.2-py2.py3-none-any.whl (123 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 123.4/123.4 kB 3.7 MB/s eta 0:00:00
Requirement already satisfied: widgetsnbextension in /usr/local/lib/python3.10/dist-packages (3.6.6)
```

```
Collecting jupyter_contrib_nbextensions
  Downloading jupyter_contrib_nbextensions-0.7.0.tar.gz (23.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23.5/23.5 MB 51.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Collecting MDTraj
  Downloading mdtraj-1.9.9.tar.gz (2.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.2/2.2 MB 88.7 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (4.66.2)
Requirement already satisfied: pytest in /usr/local/lib/python3.10/dist-packages (7.4.4)
Collecting pyemma
  Downloading pyEMMA-2.5.12.tar.gz (1.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 84.4 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting deeptime
  Using cached deeptime-0.4.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.1 MB)
Collecting einops
  Downloading einops-0.8.0-py3-none-any.whl (43 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.2/43.2 kB 6.2 MB/s eta 0:00:00
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.2.1+cu121)
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (0.17.1+cu121)
Collecting pytorch-lightning
  Downloading pytorch_lightning-2.2.3-py3-none-any.whl (802 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 802.2/802.2 kB 70.4 MB/s eta 0:00:00
Collecting nglview
  Downloading nglview-3.1.2.tar.gz (5.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.5/5.5 MB 91.4 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.10/dist-packages (from ipywidge
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.10/dist-packages (from ip
Requirement already satisfied: traitlets>=4.3.1 in /usr/local/lib/python3.10/dist-packages (from ipywidge
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets=
Collecting jupyterlab-widgets<3,>=1.0.0 (from ipywidgets==7.7.2)
  Downloading jupyterlab_widgets-1.1.7-py3-none-any.whl (295 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 295.4/295.4 kB 25.0 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pan
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (202
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (20
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scik
Requirement already satisfied: notebook in /usr/local/lib/python3.10/dist-packages (from jupyter) (6.5.5)
```

```
!jupyter nbextension enable --py --sys-prefix widgetsnbextension
```

```
Enabling notebook extension jupyter-js-widgets/extension...
Paths used for configuration of notebook:
        /usr/etc/jupyter/nbconfig/notebook.json
Paths used for configuration of notebook:

      - Validating: OK
Paths used for configuration of notebook:
        /usr/etc/jupyter/nbconfig/notebook.json
```

```
!jupyter nbextension enable nglview --py --sys-prefix
```

```
Enabling notebook extension nglview-js-widgets/extension...
Paths used for configuration of notebook:
        /usr/etc/jupyter/nbconfig/notebook.json
Paths used for configuration of notebook:

      - Validating: OK
Paths used for configuration of notebook:
        /usr/etc/jupyter/nbconfig/notebook.json
```

```
!nglview enable
```

```
Enabling notebook extension nglview-js-widgets/extension...
Paths used for configuration of notebook:
        /usr/etc/jupyter/nbconfig/notebook.json
Paths used for configuration of notebook:

      - Validating: OK
Paths used for configuration of notebook:
        /usr/etc/jupyter/nbconfig/notebook.json
```

```
%pip install git+https://github.com/andrewlferguson/snrv.git
```

```
Collecting git+https://github.com/andrewlferguson/snrv.git
  Cloning https://github.com/andrewlferguson/snrv.git to /tmp/pip-req-build-5kuoreb9
  Running command git clone --filter=blob:none --quiet https://github.com/andrewlferguson/snrv.git /tmp/p
  Resolved https://github.com/andrewlferguson/snrv.git to commit 63aeebc2f0253bec9f5e0ab03615c107256bf34f
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: snrv
  Building wheel for snrv (setup.py) ... done
  Created wheel for snrv: filename=snrv-0.1.0+52.g63aeebc-py3-none-any.whl size=30278 sha256=783c495dc695
  Stored in directory: /tmp/pip-ephem-wheel-cache-bvft170g/wheels/d8/83/c6/26e7926d23676778257c4238a0e7ca
Successfully built snrv
Installing collected packages: snrv
Successfully installed snrv-0.1.0+52.g63aeebc
```

```
%pip install git+https://github.com/Ferg-Lab/mdn_propagator.git
```

```
Collecting git+https://github.com/Ferg-Lab/mdn_propagator.git
  Cloning https://github.com/Ferg-Lab/mdn_propagator.git to /tmp/pip-req-build-jm9h7_u2
  Running command git clone --filter=blob:none --quiet https://github.com/Ferg-Lab/mdn_propagator.git /tm
  Resolved https://github.com/Ferg-Lab/mdn_propagator.git to commit ad8fd32faf84908b2c4f58bf7e16195a7c4f2
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: mdn_propagator
  Building wheel for mdn_propagator (pyproject.toml) ... done
  Created wheel for mdn_propagator: filename=mdn_propagator-1.0.0+32.gad8fd32-py3-none-any.whl size=15911
  Stored in directory: /tmp/pip-ephem-wheel-cache-n7vhvnqz/wheels/bc/a7/ff/4f2aa2dbe5dc942686e82380dbec7b
Successfully built mdn_propagator
Installing collected packages: mdn_propagator
Successfully installed mdn_propagator-1.0.0+32.gad8fd32
```

```
%pip install git+https://github.com/Ferg-Lab/molgen.git
```

```
Collecting git+https://github.com/Ferg-Lab/molgen.git
  Cloning https://github.com/Ferg-Lab/molgen.git to /tmp/pip-req-build-ia6k6i_r
  Running command git clone --filter=blob:none --quiet https://github.com/Ferg-Lab/molgen.git /tmp/pip-re
  Resolved https://github.com/Ferg-Lab/molgen.git to commit 533a1ccbcd5d59d5beea36d26a68ff4e6c28816c
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: molgen
  Building wheel for molgen (pyproject.toml) ... done
  Created wheel for molgen: filename=molgen-1.0.0+15.g533a1cc-py3-none-any.whl size=20455 sha256=f387b319
  Stored in directory: /tmp/pip-ephem-wheel-cache-ty1nv6df/wheels/1e/82/fb/a86e30e540a9156a4dfcf1eb19c92a
Successfully built molgen
Installing collected packages: molgen
Successfully installed molgen-1.0.0+15.g533a1cc
```

Load the different components from their respective repos (~1 min)

```
from mdn_propagator.propagator import Propagator
from molgen.models import DDPM
from snrv import Snrv
from snrv.utils import set_random_seed
```

Other dependencies

```
import mdtraj as md
from pathlib import Path
import torch
import matplotlib.pyplot as plt
import numpy as np
import nglview as nv
```

```
from google.colab import output
output.enable_custom_widget_manager()
```

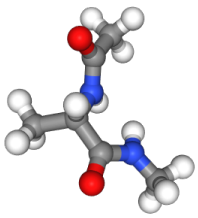
Load and prep data (~1 min)

```
trj_fnames = sorted([str(i) for i in Path('./').glob('alanine-dipeptide-*-250ns-nowater.xtc')])
top_fname = 'alanine-dipeptide-nowater.pdb'
```

```
trjs = [md.load(t, top=top_fname).center_coordinates() for t in trj_fnames]
trjs
```

```
[<mdtraj.Trajectory with 250000 frames, 22 atoms, 3 residues, and unitcells at 0x7b329c7677c0>]
```

```
v = nv.show_mdtraj(trjs[0])
v
```



```
coords_torch = list()
for trj in trjs:
    #t_backbone = trj.atom_slice(trj.top.select('backbone')).center_coordinates()
    #pdists = [torch.pdist(p)[None] for p in torch.tensor(t_backbone.xyz)]
    pdists = [torch.pdist(p)[None] for p in torch.tensor(trj.xyz)]
    coords_torch.append(torch.cat(pdists))
len(coords_torch), coords_torch[0].shape
```

```
(1, torch.Size([250000, 231]))
```

SRV fitting (~5 mins)

```
set_random_seed(42)
```

```
Setting random seed to 42
```

```
input_size = coords_torch[0].size()[1]
output_size = 3
```

```
hidden_depth = 2
hidden_size = 100
batch_norm = True
dropout_rate = 0.0
lr = 1E-2
weight_decay = 0.0
val_frac = 0.05
n_epochs = 30
batch_size = 25000
VAMPdegree = 2
is_reversible = True
num_workers = 0

model_snrv = Snrv(input_size, output_size, hidden_depth=hidden_depth, hidden_size=hidden_size,
            batch_norm=batch_norm, dropout_rate=dropout_rate, lr=lr, weight_decay=weight_decay,
            val_frac=val_frac, n_epochs=n_epochs, batch_size=batch_size,
            VAMPdegree=VAMPdegree,is_reversible=is_reversible, num_workers=num_workers,
            activation=torch.nn.GELU(), device=device)
model_snrv = model_snrv.to(device)
```

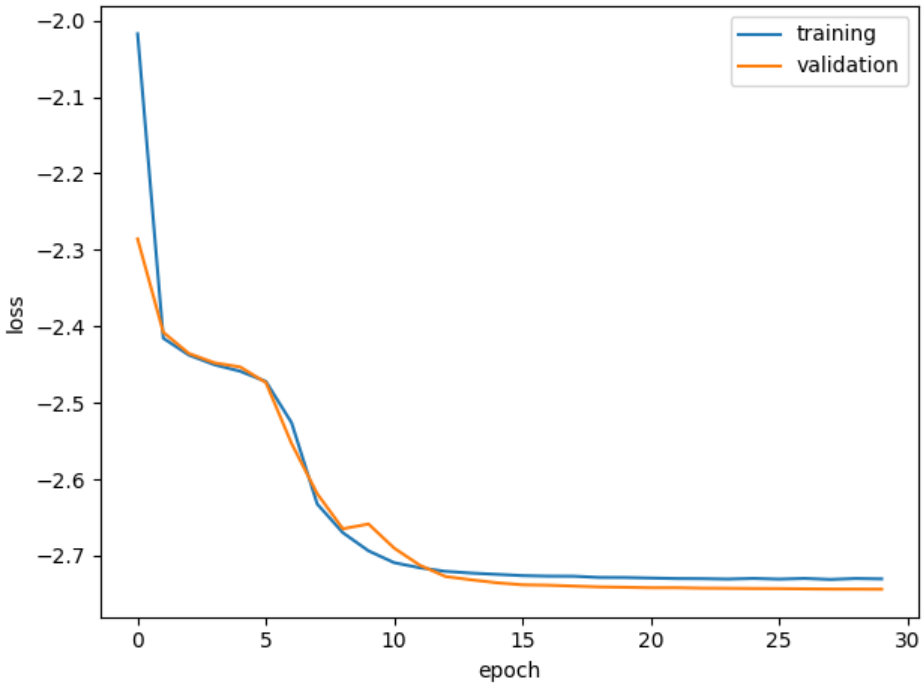
```
lag_n = 10
```

```
model_snrv.fit(coords_torch, lag=lag_n, scheduler=0.9)
```

```
Epoch 0: 100%|██████████| 10/10 [00:05<00:00,  1.98batch/s]
[Epoch 0]        training loss = -2.017  validation loss = -2.286
Epoch 1: 100%|██████████| 10/10 [00:04<00:00,  2.50batch/s]
[Epoch 1]        training loss = -2.415  validation loss = -2.408
Epoch 2: 100%|██████████| 10/10 [00:04<00:00,  2.49batch/s]
[Epoch 2]        training loss = -2.438  validation loss = -2.435
Epoch 3: 100%|██████████| 10/10 [00:03<00:00,  2.52batch/s]
[Epoch 3]        training loss = -2.450  validation loss = -2.448
Epoch 4: 100%|██████████| 10/10 [00:03<00:00,  2.52batch/s]
[Epoch 4]        training loss = -2.459  validation loss = -2.453
Epoch 5: 100%|██████████| 10/10 [00:03<00:00,  2.52batch/s]
[Epoch 5]        training loss = -2.472  validation loss = -2.473
Epoch 6: 100%|██████████| 10/10 [00:03<00:00,  2.52batch/s]
[Epoch 6]        training loss = -2.526  validation loss = -2.553
Epoch 7: 100%|██████████| 10/10 [00:04<00:00,  2.49batch/s]
[Epoch 7]        training loss = -2.632  validation loss = -2.619
Epoch 8: 100%|██████████| 10/10 [00:03<00:00,  2.51batch/s]
[Epoch 8]        training loss = -2.670  validation loss = -2.665
Epoch 9: 100%|██████████| 10/10 [00:03<00:00,  2.50batch/s]
[Epoch 9]        training loss = -2.694  validation loss = -2.658
Epoch 10: 100%|██████████| 10/10 [00:04<00:00,  2.49batch/s]
[Epoch 10]       training loss = -2.709  validation loss = -2.690
Epoch 11: 100%|██████████| 10/10 [00:04<00:00,  2.49batch/s]
[Epoch 11]       training loss = -2.716  validation loss = -2.712
Epoch 12: 100%|██████████| 10/10 [00:04<00:00,  2.49batch/s]
[Epoch 12]       training loss = -2.720  validation loss = -2.727
Epoch 13: 100%|██████████| 10/10 [00:04<00:00,  2.47batch/s]
[Epoch 13]       training loss = -2.723  validation loss = -2.731
Epoch 14: 100%|██████████| 10/10 [00:04<00:00,  2.48batch/s]
[Epoch 14]       training loss = -2.724  validation loss = -2.735
Epoch 15: 100%|██████████| 10/10 [00:04<00:00,  2.48batch/s]
[Epoch 15]       training loss = -2.726  validation loss = -2.738
Epoch 16: 100%|██████████| 10/10 [00:04<00:00,  2.49batch/s]
[Epoch 16]       training loss = -2.727  validation loss = -2.738
Epoch 17: 100%|██████████| 10/10 [00:04<00:00,  2.48batch/s]
[Epoch 17]       training loss = -2.727  validation loss = -2.740
Epoch 18: 100%|██████████| 10/10 [00:04<00:00,  2.47batch/s]
[Epoch 18]       training loss = -2.728  validation loss = -2.741
Epoch 19: 100%|██████████| 10/10 [00:04<00:00,  2.46batch/s]
[Epoch 19]       training loss = -2.728  validation loss = -2.741
Epoch 20: 100%|██████████| 10/10 [00:04<00:00,  2.48batch/s]
[Epoch 20]       training loss = -2.729  validation loss = -2.742
Epoch 21: 100%|██████████| 10/10 [00:04<00:00,  2.46batch/s]
[Epoch 21]       training loss = -2.730  validation loss = -2.742
Epoch 22: 100%|██████████| 10/10 [00:04<00:00,  2.45batch/s]
[Epoch 22]       training loss = -2.730  validation loss = -2.742
Epoch 23: 100%|██████████| 10/10 [00:04<00:00,  2.44batch/s]
```

```
[Epoch 23]       training loss = -2.730  validation loss = -2.743
Epoch 24: 100%|██████████| 10/10 [00:04<00:00,  2.47batch/s]
[Epoch 24]       training loss = -2.730  validation loss = -2.743
Epoch 25: 100%|██████████| 10/10 [00:04<00:00,  2.46batch/s]
[Epoch 25]       training loss = -2.731  validation loss = -2.743
Epoch 26: 100%|██████████| 10/10 [00:04<00:00,  2.47batch/s]
[Epoch 26]       training loss = -2.730  validation loss = -2.743
Epoch 27: 100%|██████████| 10/10 [00:04<00:00,  2.46batch/s]
[Epoch 27]       training loss = -2.731  validation loss = -2.744
Epoch 28: 100%|██████████| 10/10 [00:04<00:00,  2.45batch/s]
[Epoch 28]       training loss = -2.730  validation loss = -2.744
```

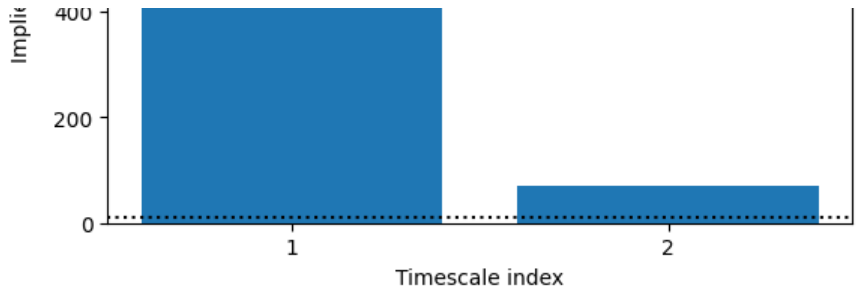
```
fig, ax = plt.subplots()
ax.plot(np.arange(len(model_snrv.training_losses)), model_snrv.training_losses)
ax.plot(np.arange(len(model_snrv.validation_losses)), model_snrv.validation_losses)
ax.set_xlabel('epoch')
ax.set_ylabel('loss')
ax.legend(['training','validation'])
fig.tight_layout()
```




```
save_freq = 1 # ps
```

```
evals = model_snrv.evals.cpu().detach().numpy()
plt.bar(range(1,evals.size), -lag_n*save_freq/np.log(evals[1:]))
plt.ylabel('Implied timescale (ps)')
plt.xticks(range(1,evals.size))
plt.xlabel('Timescale index')
plt.axhline(lag_n*save_freq, color='k', linestyle=':')
```

```
<matplotlib.lines.Line2D at 0x7b3245187370>
```




```
model_snrv.eval()
```

```
Snrv(
  (activation): GELU(approximate='none')
  (model): Sequential(
    (0): Linear(in_features=231, out_features=100, bias=True)
    (1): BatchNorm1d(100, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): GELU(approximate='none')
    (3): Linear(in_features=100, out_features=100, bias=True)
    (4): BatchNorm1d(100, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): GELU(approximate='none')
    (6): Linear(in_features=100, out_features=3, bias=True)
  )
)
```

```
evecs = model_snrv.transform(torch.cat(coords_torch)).cpu().detach().numpy()
```

```
trj_cat = md.join(trjs)
```

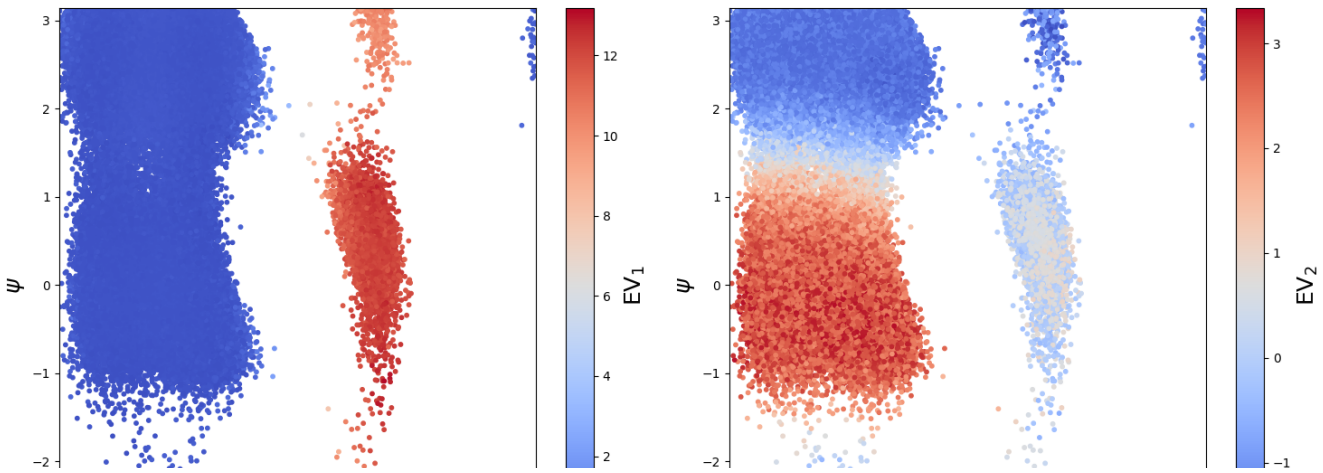
```
phi = md.compute_phi(trj_cat)[1].flatten()
psi = md.compute_psi(trj_cat)[1].flatten()
```

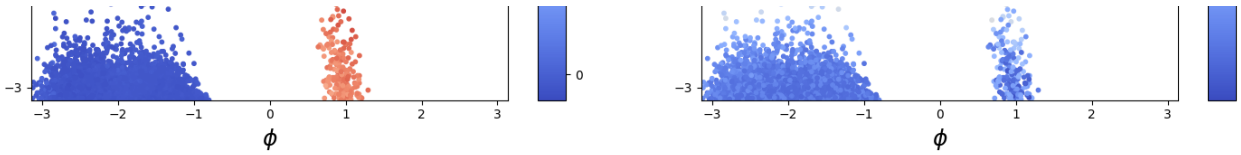
```
fig, axes = plt.subplots(1, 2, figsize = (15, 7))
axes = axes.flatten()

for e in range(1, evecs.shape[1]):
    evec = evecs[:, e]
    ax = axes[e-1]

    im = ax.scatter(phi, psi, c=evec, s=10, cmap='coolwarm')
    ax.set_xlabel('$\phi$', fontsize=18)
    ax.set_ylabel('$\psi$', fontsize=18)
    ax.set_xlim(-np.pi, np.pi)
    ax.set_ylim(-np.pi, np.pi)
    cbar = plt.colorbar(im, ax=ax)
    cbar.set_label(f'EV$_{e}$', size=18)

plt.tight_layout()
```






```
CVs = [model_snrv.transform(x).cpu().detach()[:, 1:] for x in coords_torch]
CVs[0].shape, len(CVs)
```

```
(torch.Size([250000, 2]), 1)
```

MDN propagator (~3 mins)

```
model_mdn = Propagator(dim = CVs[0].size(1))
```

```
model_mdn.fit(CVs, lag = 10, max_epochs=10)
```

```
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.utilities.rank_zero:You are using a CUDA device ('NVIDIA L4') that has Tensor Cores
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name    | Type                  | Params
-------------------------------------------------
0 | mdn     | MixtureDensityNetwork | 33.0 K
1 | _scaler | MinMaxScaler          | 0
-------------------------------------------------
33.0 K    Trainable params
0         Non-trainable params
33.0 K    Total params
0.132     Total estimated model params size (MB)
/usr/local/lib/python3.10/dist-packages/pytorch_lightning/trainer/connectors/data_connector.py:441: The '
Epoch 9: 100% 250/250 [00:04<00:00, 54.76it/s]
INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_epochs=10` reached.
Propagator(
  (mdn): MixtureDensityNetwork(
    (network): MLP(
      (mlp): Sequential(
        (0): Linear(in_features=2, out_features=128, bias=True)
        (1): SiLU()
        (2): Linear(in_features=128, out_features=128, bias=True)
        (3): SiLU()
        (4): Linear(in_features=128, out_features=125, bias=True)
      )
    )
  )
  (_scaler): MinMaxScaler()
)
```

```
n_steps = int(1E2)
x = CVs[0][0][None]
synthetic_traj_CVs = model_mdn.gen_synthetic_traj(x, n_steps)
```

```
100% 100/100 [00:00<00:00, 249.07it/s]
```

DDPM Decoder (~5 mins)

```
xyz = list()
for trj in trjs:

    t_backbone = trj.atom_slice(trj.top.select('backbone')).center_coordinates()

    n = trj.xyz.shape[0]

    xyz.append(torch.tensor(t_backbone.xyz.reshape(n, -1)).float())
```

```
model_ddpm = DDPM(xyz[0].shape[1], CVs[0].shape[1])
```

```
model_ddpm.fit(xyz, CVs, max_epochs=3)
```

```
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
/usr/local/lib/python3.10/dist-packages/pytorch_lightning/callbacks/model_checkpoint.py:653: Checkpoint d
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name              | Type              | Params
-------------------------------------------------------
0 | model             | GaussianDiffusion | 4.0 M
1 | ema_model         | GaussianDiffusion | 4.0 M
2 | _feature_scaler   | MinMaxScaler      | 0
3 | _condition_scaler | MinMaxScaler      | 0
-------------------------------------------------------
7.9 M     Trainable params
0         Non-trainable params
7.9 M     Total params
31.749    Total estimated model params size (MB)
Epoch 2: 100% 250/250 [00:24<00:00, 10.04it/s]
INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_epochs=3` reached.
DDPM(
  (model): GaussianDiffusion(
    (denoise_fn): Unet1D(
      (init_conv): Conv1d(1, 32, kernel_size=(7,), stride=(1,), padding=(3,))
      (time_mlp): Sequential(
        (0): SinusoidalPosEmb()
        (1): Linear(in_features=32, out_features=128, bias=True)
        (2): GELU(approximate='none')
        (3): Linear(in_features=128, out_features=128, bias=True)
      )
      (downs): ModuleList(
        (0): ModuleList(
          (0-1): 2 x ResnetBlock(
            (mlp): Sequential(
              (0): SiLU()
              (1): Linear(in_features=128, out_features=64, bias=True)
            )
            (block1): Block(
              (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
              (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
              (act): SiLU()
            )
            (block2): Block(
              (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
              (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
              (act): SiLU()
            )
            (res_conv): Identity()
          )
          (2): Residual(
            (fn): PreNorm(
              (fn): LinearAttention(
                (to_qkv): Conv1d(32, 384, kernel_size=(1,), stride=(1,), bias=False)
                (to_out): Sequential(
```

```
          (0): Conv1d(128, 32, kernel_size=(1,), stride=(1,))
          (1): LayerNorm()
        )
      )
      (norm): LayerNorm()
    )
  )
  (3): Conv1d(32, 32, kernel_size=(3,), stride=(2,), padding=(1,))
)
(1): ModuleList(
  (0-1): 2 x ResnetBlock(
    (mlp): Sequential(
      (0): SiLU()
      (1): Linear(in_features=128, out_features=64, bias=True)
    )
    (block1): Block(
      (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (block2): Block(
      (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (res_conv): Identity()
  )
  (2): Residual(
    (fn): PreNorm(
      (fn): LinearAttention(
        (to_qkv): Conv1d(32, 384, kernel_size=(1,), stride=(1,), bias=False)
        (to_out): Sequential(
          (0): Conv1d(128, 32, kernel_size=(1,), stride=(1,))
          (1): LayerNorm()
        )
      )
      (norm): LayerNorm()
    )
  )
  (3): Conv1d(32, 64, kernel_size=(3,), stride=(2,), padding=(1,))
)
(2): ModuleList(
  (0-1): 2 x ResnetBlock(
    (mlp): Sequential(
      (0): SiLU()
      (1): Linear(in_features=128, out_features=128, bias=True)
    )
    (block1): Block(
      (proj): WeightStandardizedConv2d(64, 64, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 64, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (block2): Block(
      (proj): WeightStandardizedConv2d(64, 64, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 64, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (res_conv): Identity()
  )
  (2): Residual(
    (fn): PreNorm(
      (fn): LinearAttention(
        (to_qkv): Conv1d(64, 384, kernel_size=(1,), stride=(1,), bias=False)
        (to_out): Sequential(
          (0): Conv1d(128, 64, kernel_size=(1,), stride=(1,))
          (1): LayerNorm()
        )
      )
      (norm): LayerNorm()
    )
```

```
    )
    (3): Conv1d(64, 128, kernel_size=(3,), stride=(2,), padding=(1,))
  )
  (3): ModuleList(
    (0-1): 2 x ResnetBlock(
      (mlp): Sequential(
        (0): SiLU()
        (1): Linear(in_features=128, out_features=256, bias=True)
      )
      (block1): Block(
        (proj): WeightStandardizedConv2d(128, 128, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 128, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (block2): Block(
        (proj): WeightStandardizedConv2d(128, 128, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 128, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (res_conv): Identity()
    )
    (2): Residual(
      (fn): PreNorm(
        (fn): LinearAttention(
          (to_qkv): Conv1d(128, 384, kernel_size=(1,), stride=(1,), bias=False)
          (to_out): Sequential(
            (0): Conv1d(128, 128, kernel_size=(1,), stride=(1,))
            (1): LayerNorm()
          )
        )
        (norm): LayerNorm()
      )
    )
    (3): Conv1d(128, 256, kernel_size=(3,), stride=(1,), padding=(1,))
  )
)
(ups): ModuleList(
  (0): ModuleList(
    (0-1): 2 x ResnetBlock(
      (mlp): Sequential(
        (0): SiLU()
        (1): Linear(in_features=128, out_features=512, bias=True)
      )
      (block1): Block(
        (proj): WeightStandardizedConv2d(384, 256, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (block2): Block(
        (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (res_conv): Conv1d(384, 256, kernel_size=(1,), stride=(1,))
    )
    (2): Residual(
      (fn): PreNorm(
        (fn): LinearAttention(
          (to_qkv): Conv1d(256, 384, kernel_size=(1,), stride=(1,), bias=False)
          (to_out): Sequential(
            (0): Conv1d(128, 256, kernel_size=(1,), stride=(1,))
            (1): LayerNorm()
          )
        )
        (norm): LayerNorm()
      )
    )
    (3): Sequential(
      (0): Upsample(scale_factor=2.0, mode='nearest')
      (1): Conv1d(256, 128, kernel_size=(3,), stride=(1,), padding=(1,))
```

```
    (1): Conv1d(256, 128, kernel_size=(3,), stride=(1,), padding=(1,))
  )
)
(1): ModuleList(
  (0-1): 2 x ResnetBlock(
    (mlp): Sequential(
      (0): SiLU()
      (1): Linear(in_features=128, out_features=256, bias=True)
    )
    (block1): Block(
      (proj): WeightStandardizedConv2d(192, 128, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 128, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (block2): Block(
      (proj): WeightStandardizedConv2d(128, 128, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 128, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (res_conv): Conv1d(192, 128, kernel_size=(1,), stride=(1,))
  )
  (2): Residual(
    (fn): PreNorm(
      (fn): LinearAttention(
        (to_qkv): Conv1d(128, 384, kernel_size=(1,), stride=(1,), bias=False)
        (to_out): Sequential(
          (0): Conv1d(128, 128, kernel_size=(1,), stride=(1,))
          (1): LayerNorm()
        )
      )
      (norm): LayerNorm()
    )
  )
  (3): Sequential(
    (0): Upsample(scale_factor=2.0, mode='nearest')
    (1): Conv1d(128, 64, kernel_size=(3,), stride=(1,), padding=(1,))
  )
)
(2): ModuleList(
  (0-1): 2 x ResnetBlock(
    (mlp): Sequential(
      (0): SiLU()
      (1): Linear(in_features=128, out_features=128, bias=True)
    )
    (block1): Block(
      (proj): WeightStandardizedConv2d(96, 64, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 64, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (block2): Block(
      (proj): WeightStandardizedConv2d(64, 64, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 64, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (res_conv): Conv1d(96, 64, kernel_size=(1,), stride=(1,))
  )
  (2): Residual(
    (fn): PreNorm(
      (fn): LinearAttention(
        (to_qkv): Conv1d(64, 384, kernel_size=(1,), stride=(1,), bias=False)
        (to_out): Sequential(
          (0): Conv1d(128, 64, kernel_size=(1,), stride=(1,))
          (1): LayerNorm()
        )
      )
      (norm): LayerNorm()
    )
  )
  (3): Sequential(
    (0): Upsample(scale_factor=2.0, mode='nearest')
    (1): Conv1d(64, 32, kernel_size=(3,), stride=(1,), padding=(1,))
```

```
      (1): Conv1d(64, 32, kernel_size=(3,), stride=(1,), padding=(1,))
    )
  )
  (3): ModuleList(
    (0-1): 2 x ResnetBlock(
      (mlp): Sequential(
        (0): SiLU()
        (1): Linear(in_features=128, out_features=64, bias=True)
      )
      (block1): Block(
        (proj): WeightStandardizedConv2d(64, 32, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (block2): Block(
        (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (res_conv): Conv1d(64, 32, kernel_size=(1,), stride=(1,))
    )
    (2): Residual(
      (fn): PreNorm(
        (fn): LinearAttention(
          (to_qkv): Conv1d(32, 384, kernel_size=(1,), stride=(1,), bias=False)
          (to_out): Sequential(
            (0): Conv1d(128, 32, kernel_size=(1,), stride=(1,))
            (1): LayerNorm()
          )
        )
        (norm): LayerNorm()
      )
    )
    (3): Conv1d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
  )
)
(mid_block1): ResnetBlock(
  (mlp): Sequential(
    (0): SiLU()
    (1): Linear(in_features=128, out_features=512, bias=True)
  )
  (block1): Block(
    (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
    (act): SiLU()
  )
  (block2): Block(
    (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
    (act): SiLU()
  )
  (res_conv): Identity()
)
(mid_attn): Residual(
  (fn): PreNorm(
    (fn): Attention(
      (to_qkv): Conv1d(256, 384, kernel_size=(1,), stride=(1,), bias=False)
      (to_out): Conv1d(128, 256, kernel_size=(1,), stride=(1,))
    )
    (norm): LayerNorm()
  )
)
(mid_block2): ResnetBlock(
  (mlp): Sequential(
    (0): SiLU()
    (1): Linear(in_features=128, out_features=512, bias=True)
  )
  (block1): Block(
    (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
```

```
          (act): SiLU()
        )
        (block2): Block(
          (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
          (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
          (act): SiLU()
        )
        (res_conv): Identity()
      )
      (final_res_block): ResnetBlock(
        (mlp): Sequential(
          (0): SiLU()
          (1): Linear(in_features=128, out_features=64, bias=True)
        )
        (block1): Block(
          (proj): WeightStandardizedConv2d(64, 32, kernel_size=(3,), stride=(1,), padding=(1,))
          (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
          (act): SiLU()
        )
        (block2): Block(
          (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
          (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
          (act): SiLU()
        )
        (res_conv): Conv1d(64, 32, kernel_size=(1,), stride=(1,))
      )
      (final_conv): Conv1d(32, 1, kernel_size=(1,), stride=(1,))
    )
  )
  (ema_model): GaussianDiffusion(
    (denoise_fn): Unet1D(
      (init_conv): Conv1d(1, 32, kernel_size=(7,), stride=(1,), padding=(3,))
      (time_mlp): Sequential(
        (0): SinusoidalPosEmb()
        (1): Linear(in_features=32, out_features=128, bias=True)
        (2): GELU(approximate='none')
        (3): Linear(in_features=128, out_features=128, bias=True)
      )
      (downs): ModuleList(
        (0): ModuleList(
          (0-1): 2 x ResnetBlock(
            (mlp): Sequential(
              (0): SiLU()
              (1): Linear(in_features=128, out_features=64, bias=True)
            )
            (block1): Block(
              (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
              (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
              (act): SiLU()
            )
            (block2): Block(
              (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
              (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
              (act): SiLU()
            )
            (res_conv): Identity()
          )
          (2): Residual(
            (fn): PreNorm(
              (fn): LinearAttention(
                (to_qkv): Conv1d(32, 384, kernel_size=(1,), stride=(1,), bias=False)
                (to_out): Sequential(
                  (0): Conv1d(128, 32, kernel_size=(1,), stride=(1,))
                  (1): LayerNorm()
                )
              )
              (norm): LayerNorm()
            )
          )
          (3): Conv1d(32, 32, kernel_size=(3,), stride=(2,), padding=(1,))
```

```
    )
    (1): ModuleList(
      (0-1): 2 x ResnetBlock(
        (mlp): Sequential(
          (0): SiLU()
          (1): Linear(in_features=128, out_features=64, bias=True)
        )
        (block1): Block(
          (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
          (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
          (act): SiLU()
        )
        (block2): Block(
          (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
          (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
          (act): SiLU()
        )
        (res_conv): Identity()
      )
      (2): Residual(
        (fn): PreNorm(
          (fn): LinearAttention(
            (to_qkv): Conv1d(32, 384, kernel_size=(1,), stride=(1,), bias=False)
            (to_out): Sequential(
              (0): Conv1d(128, 32, kernel_size=(1,), stride=(1,))
              (1): LayerNorm()
            )
          )
          (norm): LayerNorm()
        )
      )
      (3): Conv1d(32, 64, kernel_size=(3,), stride=(2,), padding=(1,))
    )
    (2): ModuleList(
      (0-1): 2 x ResnetBlock(
        (mlp): Sequential(
          (0): SiLU()
          (1): Linear(in_features=128, out_features=128, bias=True)
        )
        (block1): Block(
          (proj): WeightStandardizedConv2d(64, 64, kernel_size=(3,), stride=(1,), padding=(1,))
          (norm): GroupNorm(8, 64, eps=1e-05, affine=True)
          (act): SiLU()
        )
        (block2): Block(
          (proj): WeightStandardizedConv2d(64, 64, kernel_size=(3,), stride=(1,), padding=(1,))
          (norm): GroupNorm(8, 64, eps=1e-05, affine=True)
          (act): SiLU()
        )
        (res_conv): Identity()
      )
      (2): Residual(
        (fn): PreNorm(
          (fn): LinearAttention(
            (to_qkv): Conv1d(64, 384, kernel_size=(1,), stride=(1,), bias=False)
            (to_out): Sequential(
              (0): Conv1d(128, 64, kernel_size=(1,), stride=(1,))
              (1): LayerNorm()
            )
          )
          (norm): LayerNorm()
        )
      )
      (3): Conv1d(64, 128, kernel_size=(3,), stride=(2,), padding=(1,))
    )
    (3): ModuleList(
      (0-1): 2 x ResnetBlock(
        (mlp): Sequential(
          (0): SiLU()
          (1): Linear(in_features=128, out_features=256, bias=True)
```

```
      (1): Linear(in_features=128, out_features=256, bias=True)
    )
    (block1): Block(
      (proj): WeightStandardizedConv2d(128, 128, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 128, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (block2): Block(
      (proj): WeightStandardizedConv2d(128, 128, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 128, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (res_conv): Identity()
  )
  (2): Residual(
    (fn): PreNorm(
      (fn): LinearAttention(
        (to_qkv): Conv1d(128, 384, kernel_size=(1,), stride=(1,), bias=False)
        (to_out): Sequential(
          (0): Conv1d(128, 128, kernel_size=(1,), stride=(1,))
          (1): LayerNorm()
        )
      )
      (norm): LayerNorm()
    )
  )
  (3): Conv1d(128, 256, kernel_size=(3,), stride=(1,), padding=(1,))
 )
)
(ups): ModuleList(
 (0): ModuleList(
  (0-1): 2 x ResnetBlock(
    (mlp): Sequential(
      (0): SiLU()
      (1): Linear(in_features=128, out_features=512, bias=True)
    )
    (block1): Block(
      (proj): WeightStandardizedConv2d(384, 256, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (block2): Block(
      (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
      (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
      (act): SiLU()
    )
    (res_conv): Conv1d(384, 256, kernel_size=(1,), stride=(1,))
  )
  (2): Residual(
    (fn): PreNorm(
      (fn): LinearAttention(
        (to_qkv): Conv1d(256, 384, kernel_size=(1,), stride=(1,), bias=False)
        (to_out): Sequential(
          (0): Conv1d(128, 256, kernel_size=(1,), stride=(1,))
          (1): LayerNorm()
        )
      )
      (norm): LayerNorm()
    )
  )
  (3): Sequential(
    (0): Upsample(scale_factor=2.0, mode='nearest')
    (1): Conv1d(256, 128, kernel_size=(3,), stride=(1,), padding=(1,))
  )
 )
 (1): ModuleList(
  (0-1): 2 x ResnetBlock(
    (mlp): Sequential(
      (0): SiLU()
      (1): Linear(in_features=128, out_features=256, bias=True)
```

```
      )
      (block1): Block(
        (proj): WeightStandardizedConv2d(192, 128, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 128, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (block2): Block(
        (proj): WeightStandardizedConv2d(128, 128, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 128, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (res_conv): Conv1d(192, 128, kernel_size=(1,), stride=(1,))
    )
    (2): Residual(
      (fn): PreNorm(
        (fn): LinearAttention(
          (to_qkv): Conv1d(128, 384, kernel_size=(1,), stride=(1,), bias=False)
          (to_out): Sequential(
            (0): Conv1d(128, 128, kernel_size=(1,), stride=(1,))
            (1): LayerNorm()
          )
        )
        (norm): LayerNorm()
      )
    )
    (3): Sequential(
      (0): Upsample(scale_factor=2.0, mode='nearest')
      (1): Conv1d(128, 64, kernel_size=(3,), stride=(1,), padding=(1,))
    )
  )
  (2): ModuleList(
    (0-1): 2 x ResnetBlock(
      (mlp): Sequential(
        (0): SiLU()
        (1): Linear(in_features=128, out_features=128, bias=True)
      )
      (block1): Block(
        (proj): WeightStandardizedConv2d(96, 64, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 64, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (block2): Block(
        (proj): WeightStandardizedConv2d(64, 64, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 64, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (res_conv): Conv1d(96, 64, kernel_size=(1,), stride=(1,))
    )
    (2): Residual(
      (fn): PreNorm(
        (fn): LinearAttention(
          (to_qkv): Conv1d(64, 384, kernel_size=(1,), stride=(1,), bias=False)
          (to_out): Sequential(
            (0): Conv1d(128, 64, kernel_size=(1,), stride=(1,))
            (1): LayerNorm()
          )
        )
        (norm): LayerNorm()
      )
    )
    (3): Sequential(
      (0): Upsample(scale_factor=2.0, mode='nearest')
      (1): Conv1d(64, 32, kernel_size=(3,), stride=(1,), padding=(1,))
    )
  )
  (3): ModuleList(
    (0-1): 2 x ResnetBlock(
      (mlp): Sequential(
        (0): SiLU()
        (1): Linear(in features=128, out features=64, bias=True)
```

```
      )
      (block1): Block(
        (proj): WeightStandardizedConv2d(64, 32, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (block2): Block(
        (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
        (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
        (act): SiLU()
      )
      (res_conv): Conv1d(64, 32, kernel_size=(1,), stride=(1,))
    )
    (2): Residual(
      (fn): PreNorm(
        (fn): LinearAttention(
          (to_qkv): Conv1d(32, 384, kernel_size=(1,), stride=(1,), bias=False)
          (to_out): Sequential(
            (0): Conv1d(128, 32, kernel_size=(1,), stride=(1,))
            (1): LayerNorm()
          )
        )
        (norm): LayerNorm()
      )
    )
    (3): Conv1d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
  )
)
(mid_block1): ResnetBlock(
  (mlp): Sequential(
    (0): SiLU()
    (1): Linear(in_features=128, out_features=512, bias=True)
  )
  (block1): Block(
    (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
    (act): SiLU()
  )
  (block2): Block(
    (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
    (act): SiLU()
  )
  (res_conv): Identity()
)
(mid_attn): Residual(
  (fn): PreNorm(
    (fn): Attention(
      (to_qkv): Conv1d(256, 384, kernel_size=(1,), stride=(1,), bias=False)
      (to_out): Conv1d(128, 256, kernel_size=(1,), stride=(1,))
    )
    (norm): LayerNorm()
  )
)
(mid_block2): ResnetBlock(
  (mlp): Sequential(
    (0): SiLU()
    (1): Linear(in_features=128, out_features=512, bias=True)
  )
  (block1): Block(
    (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
    (act): SiLU()
  )
  (block2): Block(
    (proj): WeightStandardizedConv2d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (norm): GroupNorm(8, 256, eps=1e-05, affine=True)
    (act): SiLU()
  )
  (res_conv): Identity()
```

```
        (res_conv): Identity()
      )
      (final_res_block): ResnetBlock(
        (mlp): Sequential(
          (0): SiLU()
          (1): Linear(in_features=128, out_features=64, bias=True)
        )
        (block1): Block(
          (proj): WeightStandardizedConv2d(64, 32, kernel_size=(3,), stride=(1,), padding=(1,))
          (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
          (act): SiLU()
        )
        (block2): Block(
          (proj): WeightStandardizedConv2d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
          (norm): GroupNorm(8, 32, eps=1e-05, affine=True)
          (act): SiLU()
        )
        (res_conv): Conv1d(64, 32, kernel_size=(1,), stride=(1,))
      )
      (final_conv): Conv1d(32, 1, kernel_size=(1,), stride=(1,))
    )
  )
  (_feature_scaler): MinMaxScaler()
  (_condition_scaler): MinMaxScaler()
)
```

Decode synthetic traj (~2 mins)

```
xyz_gen = model_ddpm.generate(synthetic_traj_CVs)
```

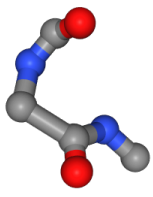
sampling loop time step: 100% 1000/1000 [00:33<00:00, 30.04it/s]

```
xyz_gen = xyz_gen.reshape(xyz_gen.size(0), -1 , 3).numpy()
fake_trj = md.Trajectory(xyz = xyz_gen, topology=t_backbone.top)
fake_trj
```

```
<mdtraj.Trajectory with 100 frames, 8 atoms, 3 residues, without unitcells at 0x7b32459e8190>
```

Visualize results (~1 min)

```
v = nv.show_mdtraj(fake_trj)
v
```



```
fake_trj.save_pdb('ADP_backbone_synthetic_traj.pdb')
```