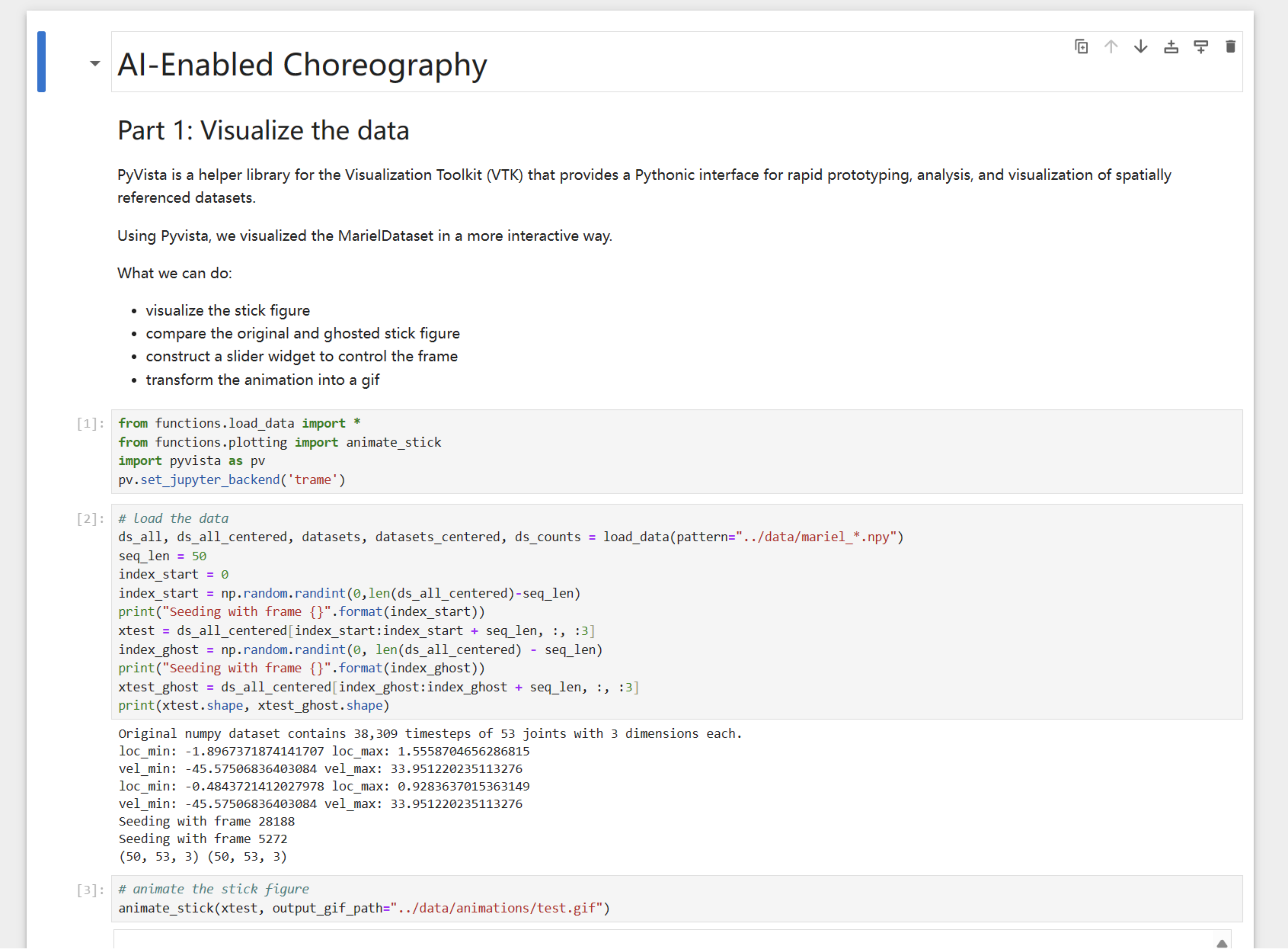


AI-Enabled Choreography

Part 1: Visualize the data

PyVista is a helper library for the Visualization Toolkit (VTK) that provides a Pythonic interface for rapid prototyping, analysis, and visualization of spatially referenced datasets.

Using Pyvista, we visualized the MarielDataset in a more interactive way.

What we can do:

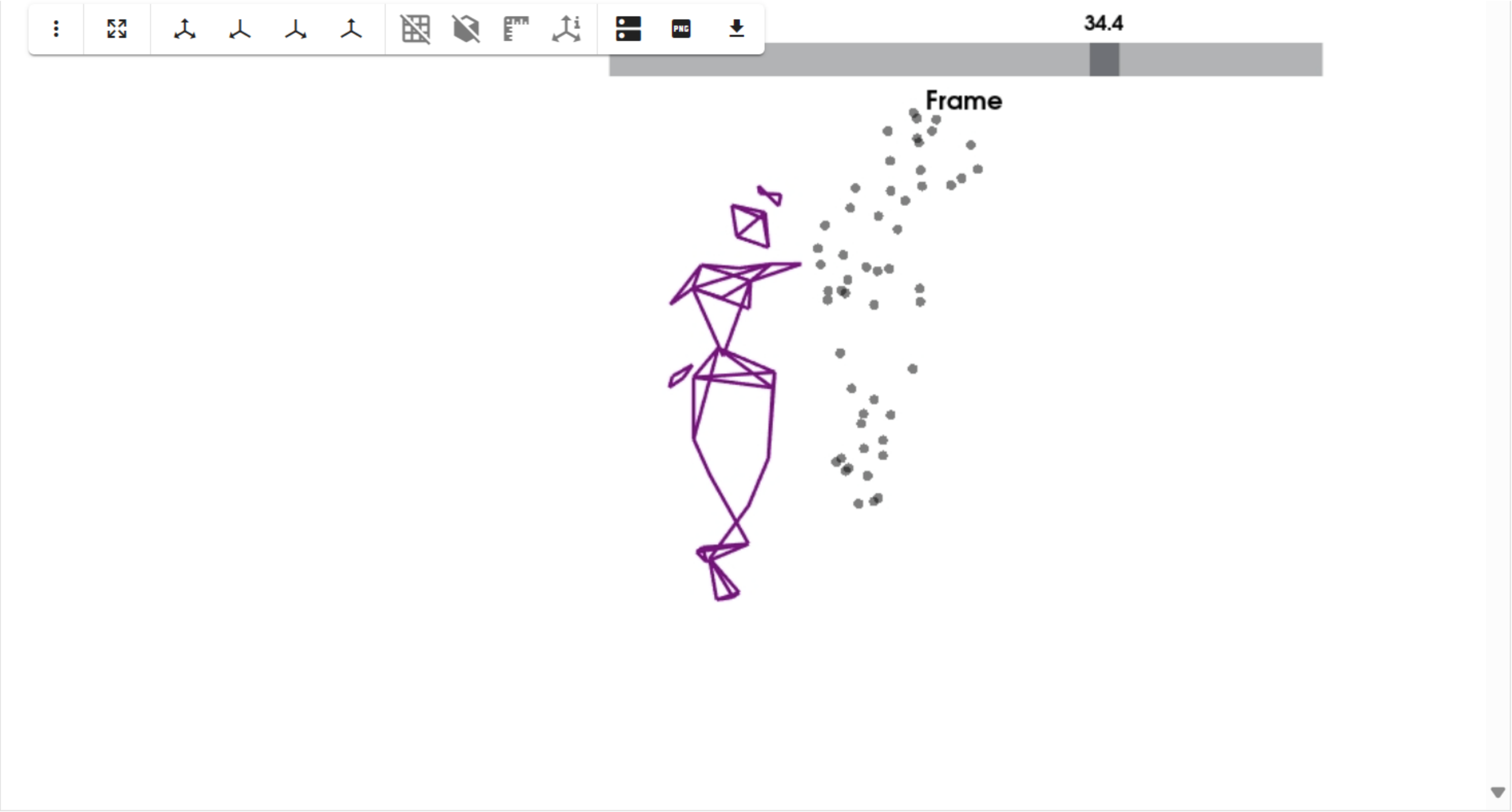
- visualize the stick figure
- compare the original and ghosted stick figure
- construct a slider widget to control the frame
- transform the animation into a gif

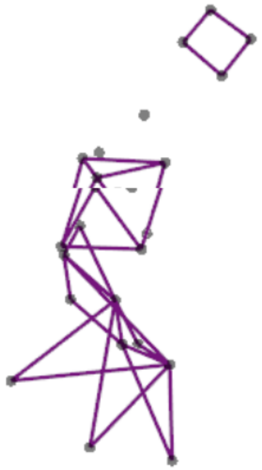
[1]:
```
from functions.load_data import *
from functions.plotting import animate_stick
import pyvista as pv
pv.set_jupyter_backend('trame')
```

[2]:
```
# load the data
ds_all, ds_all_centered, datasets, datasets_centered, ds_counts = load_data(pattern="../data/mariel_*.npy")
seq_len = 50
index_start = 0
index_start = np.random.randint(0,len(ds_all_centered)-seq_len)
print("Seeding with frame {}".format(index_start))
xtest = ds_all_centered[index_start:index_start + seq_len, :, :3]
index_ghost = np.random.randint(0, len(ds_all_centered) - seq_len)
print("Seeding with frame {}".format(index_ghost))
xtest_ghost = ds_all_centered[index_ghost:index_ghost + seq_len, :, :3]
print(xtest.shape, xtest_ghost.shape)
```

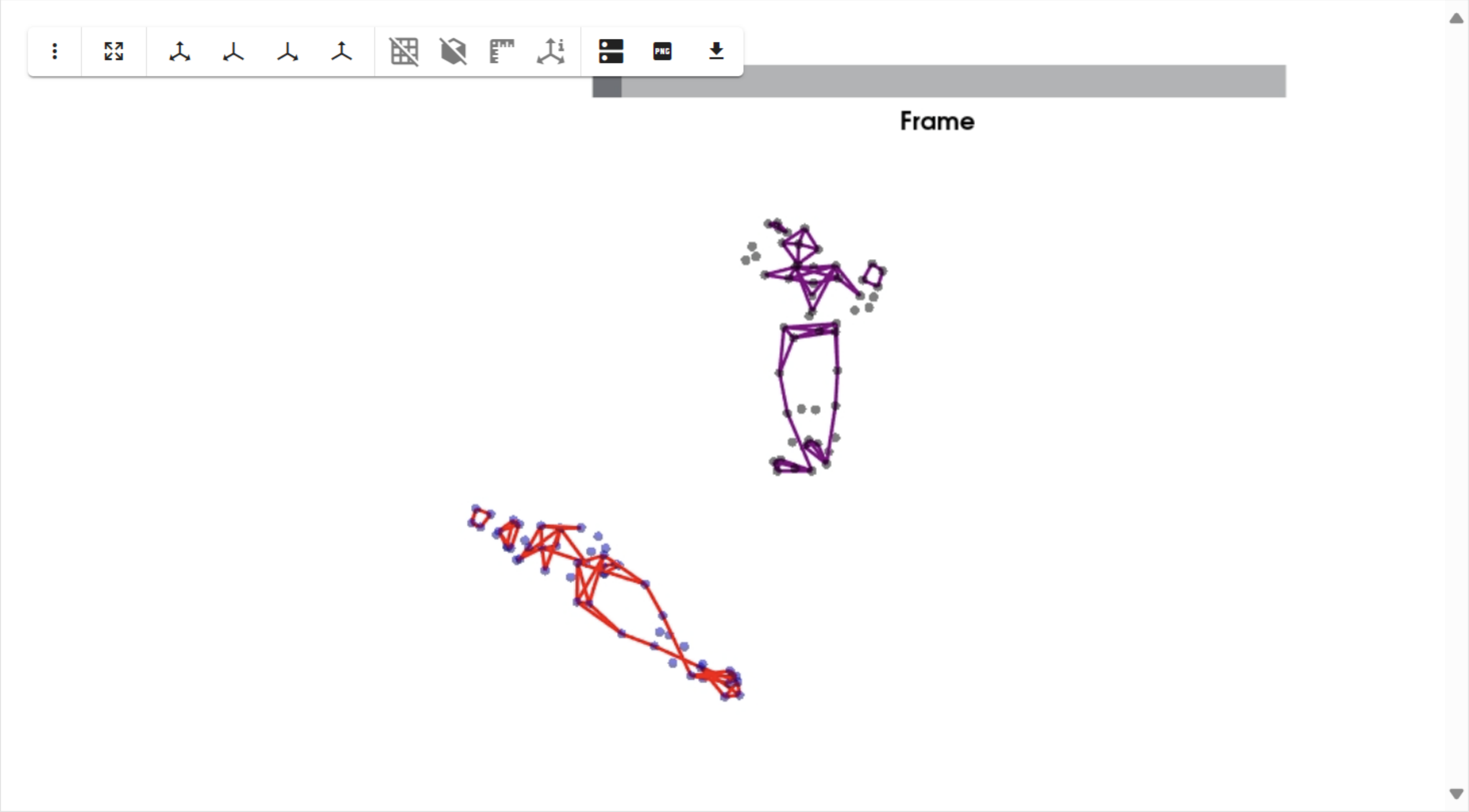
```
Original numpy dataset contains 38,309 timesteps of 53 joints with 3 dimensions each.
loc_min: -1.8967371874141707 loc_max: 1.5558704656286815
vel_min: -45.57506836403084 vel_max: 33.951220235113276
loc_min: -0.4843721412027978 loc_max: 0.9283637015363149
vel_min: -45.57506836403084 vel_max: 33.951220235113276
Seeding with frame 28188
Seeding with frame 5272
(50, 53, 3) (50, 53, 3)
```

[3]:
```
# animate the stick figure
animate_stick(xtest, output_gif_path="../data/animations/test.gif")
```

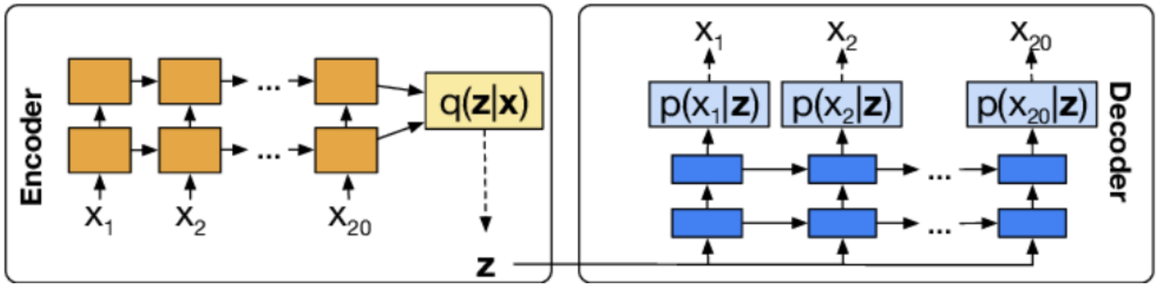

34.4
Frame



```
[4]: # compare the original and ghosted stick figure
     animate_stick(xtest, xtest_ghost, ghost_shift=0.5, output_gif_path="../data/animations/test-ghost.gif")
```



Transformers is also worth trying.



The VAE component is responsible for learning a latent representation of the input dance sequences. It consists of an encoder network that maps the input dance sequence to a lower-dimensional latent space, and a decoder network that reconstructs the original dance sequence from the latent space. The encoder network captures the essential features and patterns of the input sequences, while the decoder network generates new dance sequences based on the learned latent representation.

The LSTM component is a type of recurrent neural network (RNN) that can model the temporal dependencies in the dance sequences. It processes the latent representation generated by the VAE encoder and learns to generate dance steps one at a time, conditioned on the previous steps. By incorporating LSTM, the model can capture the sequential nature of dance movements and generate coherent and realistic dance sequences.

[5]:

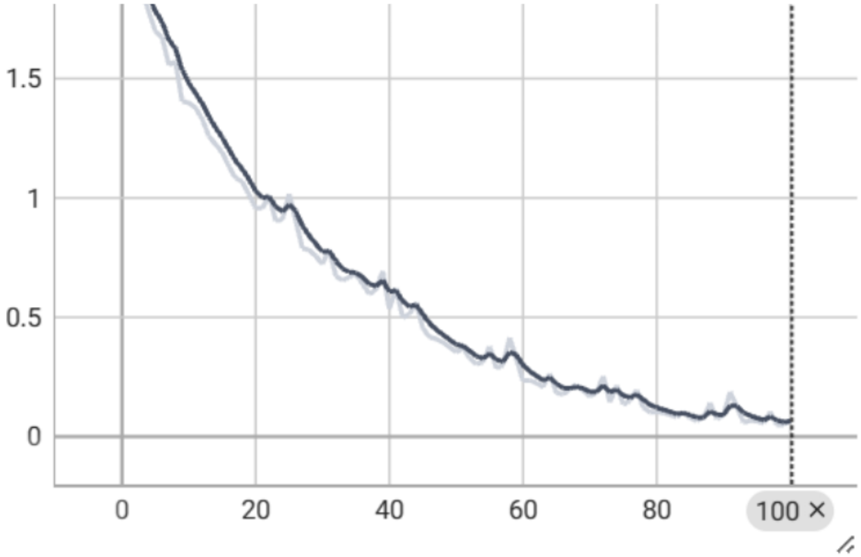
```
from functions.main import *
```

```
Original numpy dataset contains 38,309 timesteps of 53 joints with 3 dimensions each.
loc_min: -1.8967371874141707 loc_max: 1.5558704656286815
vel_min: -45.57506836403084 vel_max: 33.951220235113276
loc_min: -0.4843721412027978 loc_max: 0.9283637015363149
vel_min: -45.57506836403084 vel_max: 33.951220235113276

Generating overlapping sequences...
Using (x,y)-centering...
Using all joints...
```

[6]:

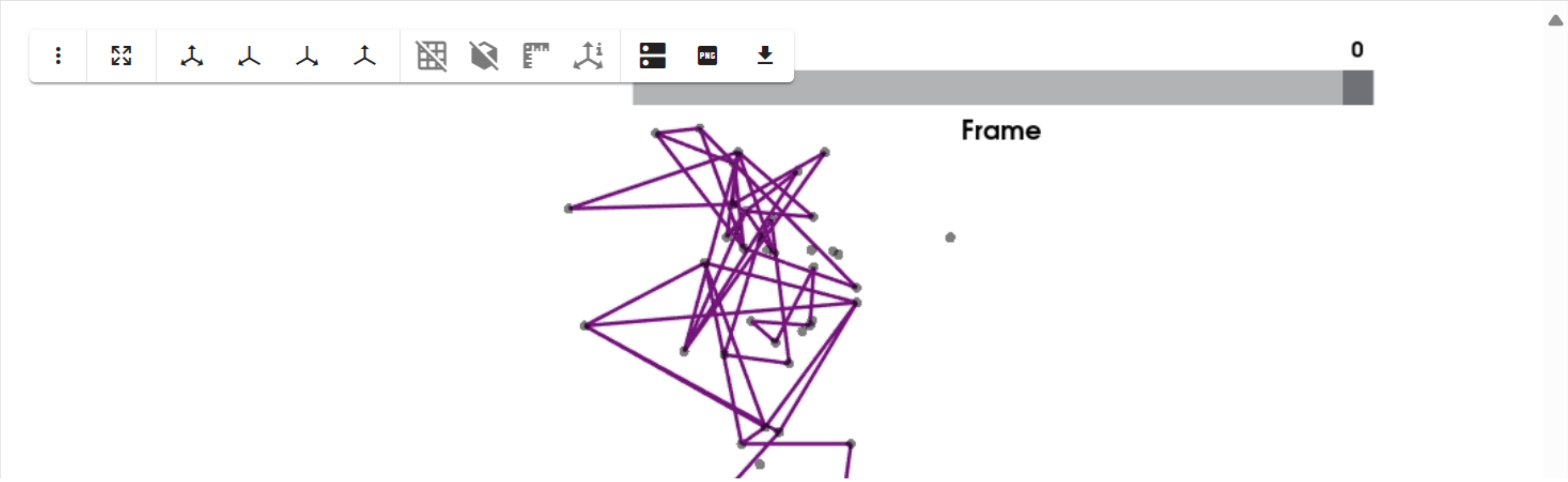
```
# training(optional)
# run_train()
```

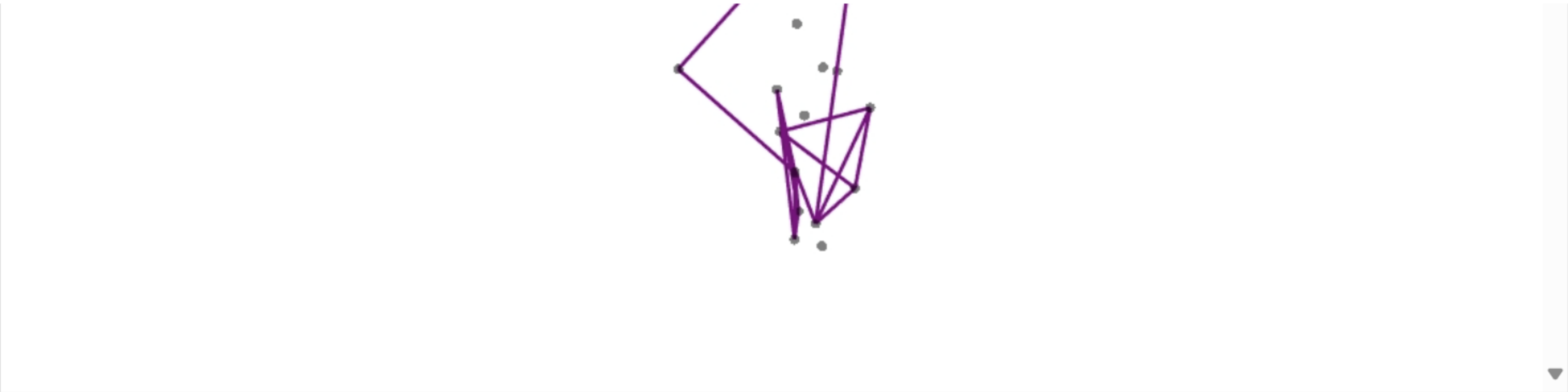


	Run ↑	Smoothed	Value	Step	Relative
●	.	0.0648	0.0704	100	39.46 min

[7]:

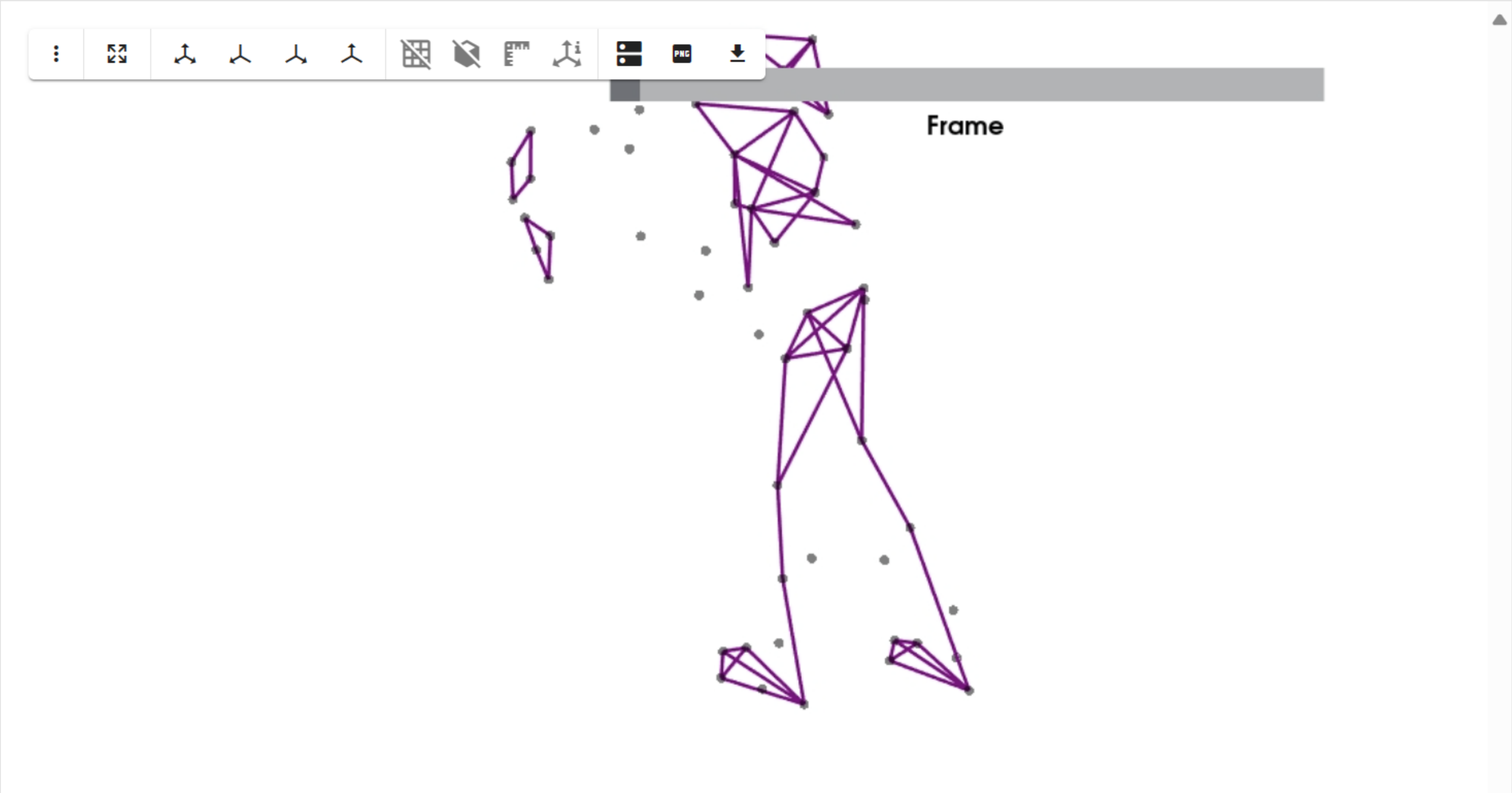
```
# or using the pretrained weights to predict the next frame
predict_frame = run_predict(1)
predict_frame = predict_frame[0].reshape(1, 53, 6).cpu().detach().numpy()[..., :3]
animate_stick(predict_frame)
```





[8]:

```
# using a loop, we can generate dance sequence of any length
predict = run_generate(1, 50)
predict = predict[0].reshape(predict.size(1), 53, 6).cpu().detach().numpy()[..., :3]
animate_stick(predict)
```



Part 3: Why this project?

I am extremely interested in this project that combines art, dance, and artificial intelligence.

I have been involved in the intersection of computer science and art for several years. I am now Master of Software Engineering at Nanjing University and got a minor in Culture and Creativity during my undergraduate study. I have worked on several AI+art projects in the past. For example, I constructed diffusion models to accomplish Text2Text, Img2Img, and Inpainting tasks for generating photos with distinctive Chinese traditional characteristics, such as shadow puppetry and New Year painting styles. I have also worked on a project called TALES FROM THE THOUSAND AND ONE NIGHTS. It involved a comprehensive exploration of emotional elements in Chinese classical films, incorporating sentiment analysis and audio-visual processing. It encompassed aspects such as film restoration and colorization.

While the VAE+LSTM architecture used in this experiment has shown promising results, there are other architectures, particularly the Transformers, that are worth exploring for this task. It can utilize self-attention mechanisms to attend to different parts of the input sequence, enabling it to learn contextual representations effectively.

AI in art opens up a realm of possibilities for artistic exploration, innovation, and engagement. I believe this project is a perfect opportunity to combine my technical skills with my artistic interests and contribute to the field of AI-enabled choreography.

[ ]: