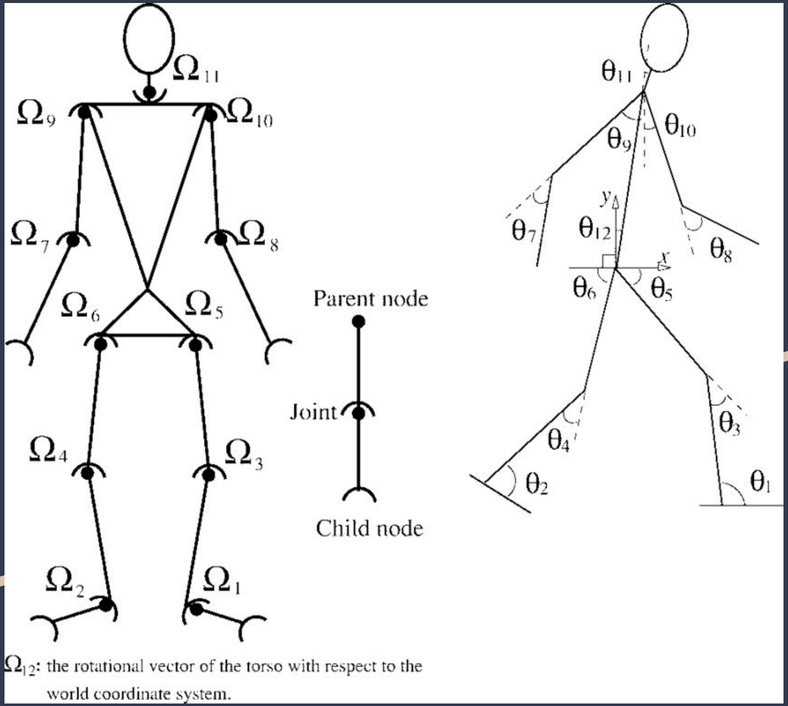
Deep Learning Transformers for Non-cyclical Kinematics

By: Coleman DuPlessie, Eddie Wei
Mentor: Andrew Gritsevskiy

Introduction



- Machine learning is a very good way to solve complex kinematics problems!
- Other techniques are very complicated and have downsides:
  - Long computations
  - Cannot generalize
- **Machine learning can fix both of these issues**

Cyclical vs Non-cyclical Kinematics

Cyclical:

- Generally consistent motion
- Ex: Walking on treadmill at constant speed
- Easier to predict
- Most studies have chosen to predict cyclical data

Non-cyclical:

- Much more complex
- More factors and variables involved in predicting
- Much greater variety
- Cannot be done well with standard models
- Not very many studies have been done on this

Previous Papers

Prediction of gait trajectories based on the Long Short Term Memory (LSTM) neural networks

- Used 4 LSTMs: Standard, stacked, bidirectional, autoencoder
- Data from 15 participants walking for 10 minutes at different speeds
- **Improvement: LSTMs cannot predict irregularities in walking**

Are Transformers Effective for Time Series Forecasting

- Transformers are the best model for time series, although there could be improvement
- Many applications:
    - Traffic flow estimation
    - Energy management
    - Financial Investment

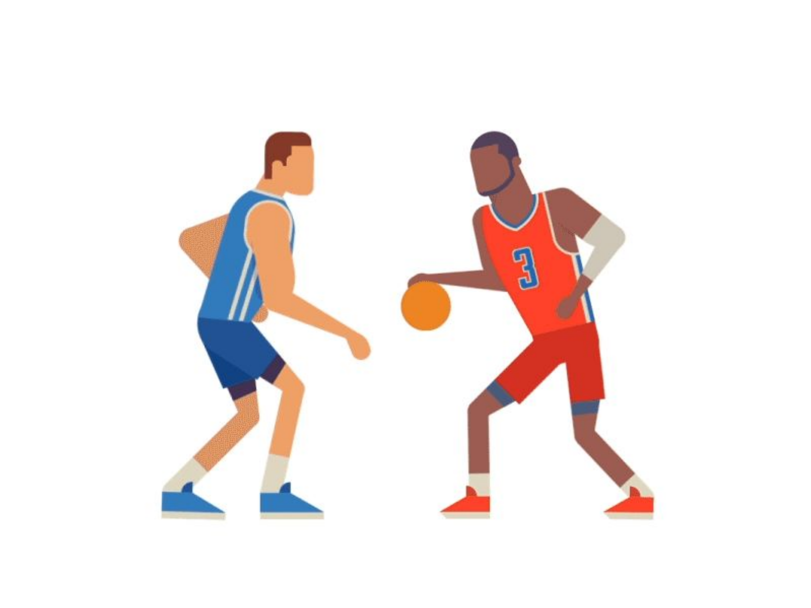
Deep Transformer Models for Time Series Forecasting

- Transformers are adaptable with little modifications needed

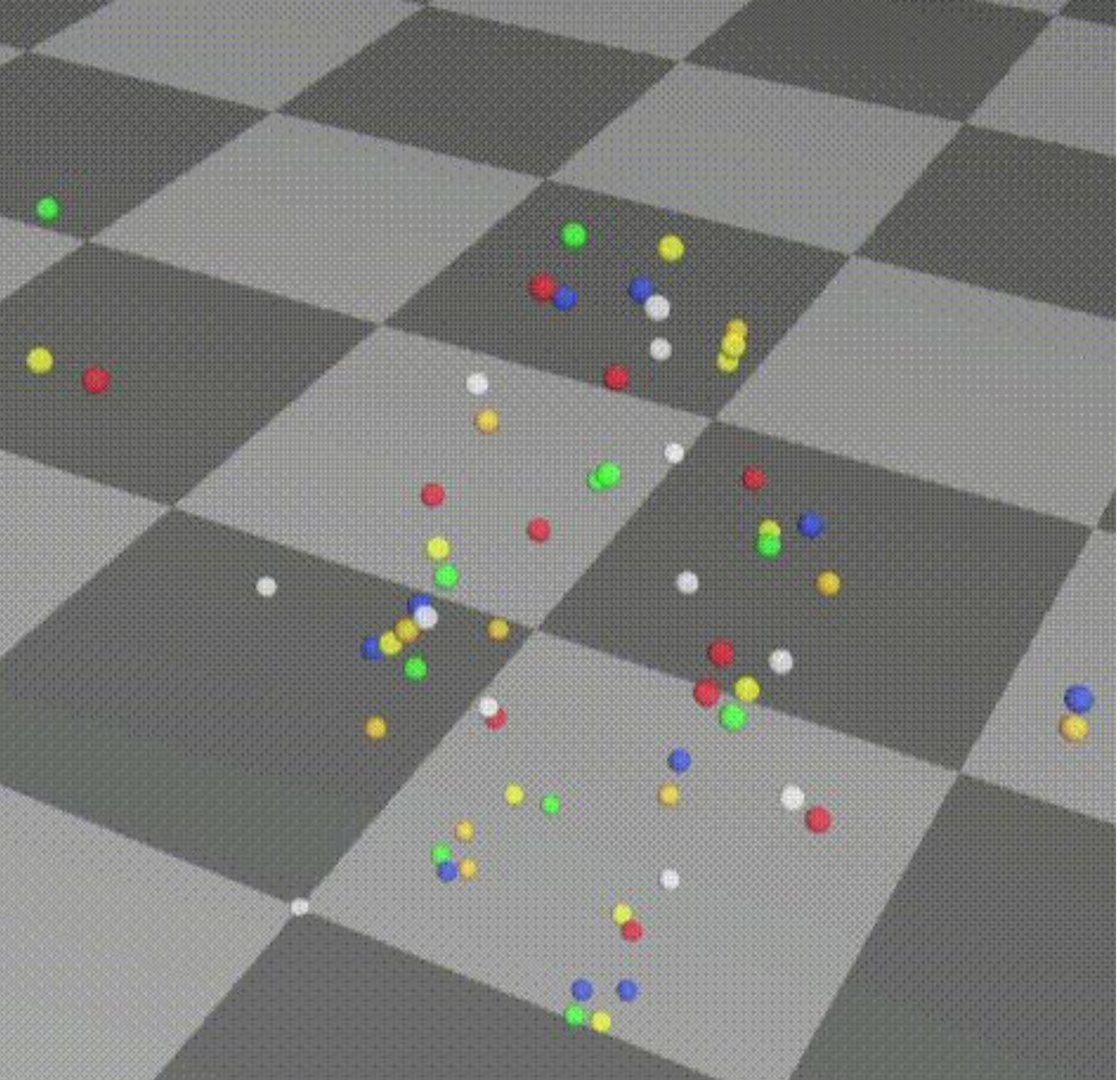
What We Did

- Trained our models on human kinematics in a kitchen
- Difficult because cooking is very complicated
- **Our models must make use of long-term dependencies**
- We wanted to see how transformers would do!

Everyday Life Applications

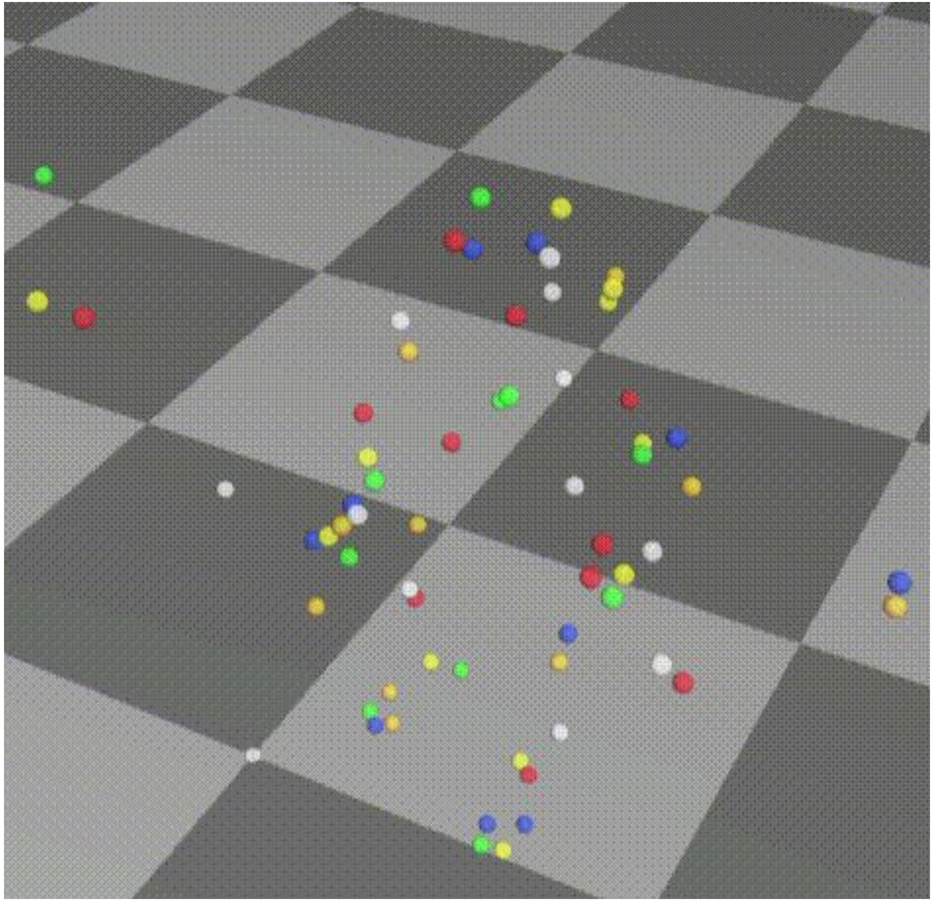


- **Humans rarely do perfectly cyclical things**
- Go from predicting running, jumping, standing up, etc. to more complex tasks requiring a process
- Greatly broadens the range of kinematics we can look at



A visualization of the Kitchen Dataset’s motion capture data

The stationary points around the test subject are part of the environment (e.g. the handles of cabinets).

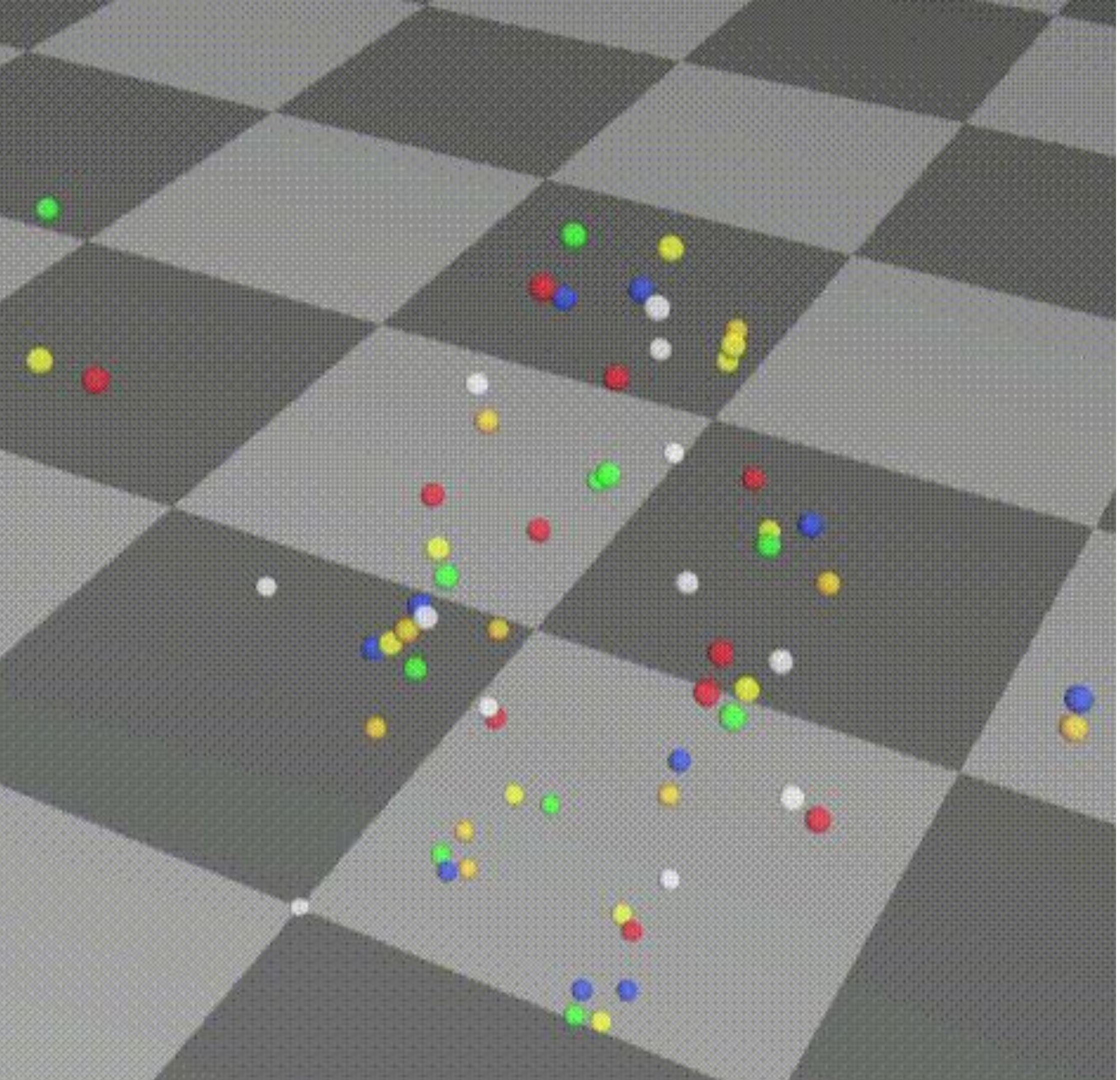


Non-cyclical data: No obvious pattern, more general: in this case, taking a brownie out of the oven

Cyclical data: an obvious, infinite pattern: left, right, left, right…

An image from Posetrack

The colorful skeletons are the points to be predicted and are not given to the models.

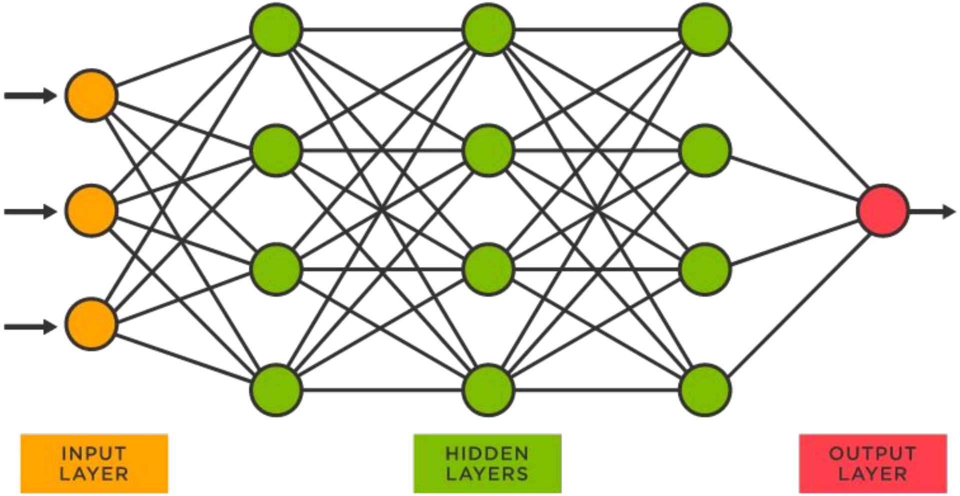


A visualization of the Kitchen Dataset’s motion capture data

The stationary points around the test subject are part of the environment (e.g. the handles of cabinets).

Neural Networks

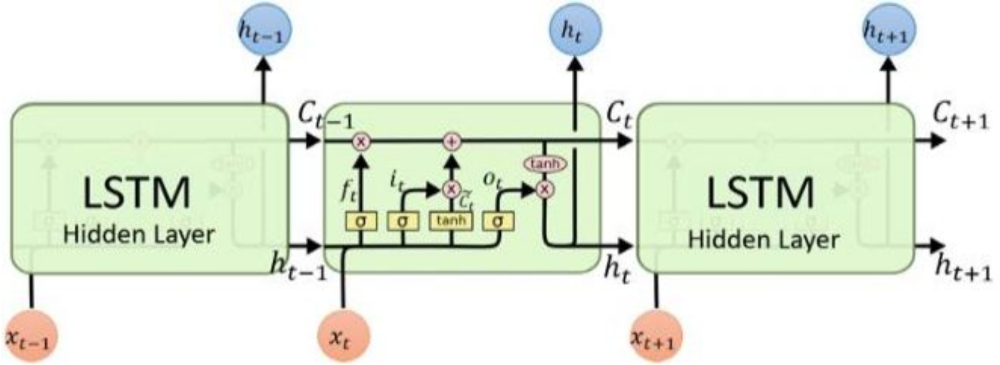
Below: A simple neural network. Each dot (a “node”) takes a weighted average of all the nodes to its immediate left (in the previous “layer”) and then applies a nonlinear function. After some number of layers, we measure the value of the final node(s) and let it be the model’s output.



Our Models:

- Transformer
- Informer
- LSTM
- Simple benchmark "model"

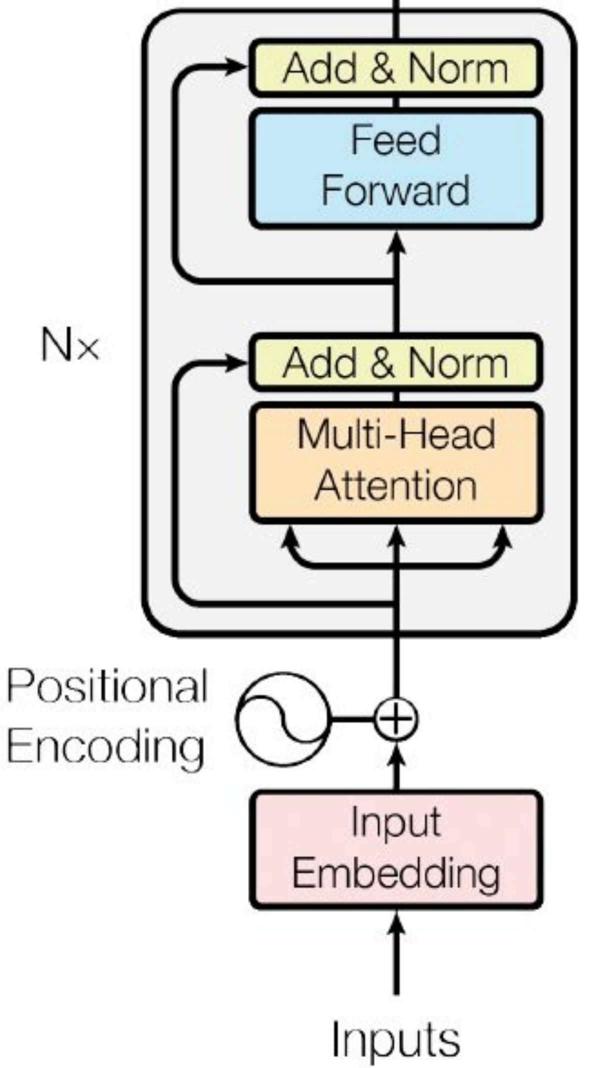
LSTMs



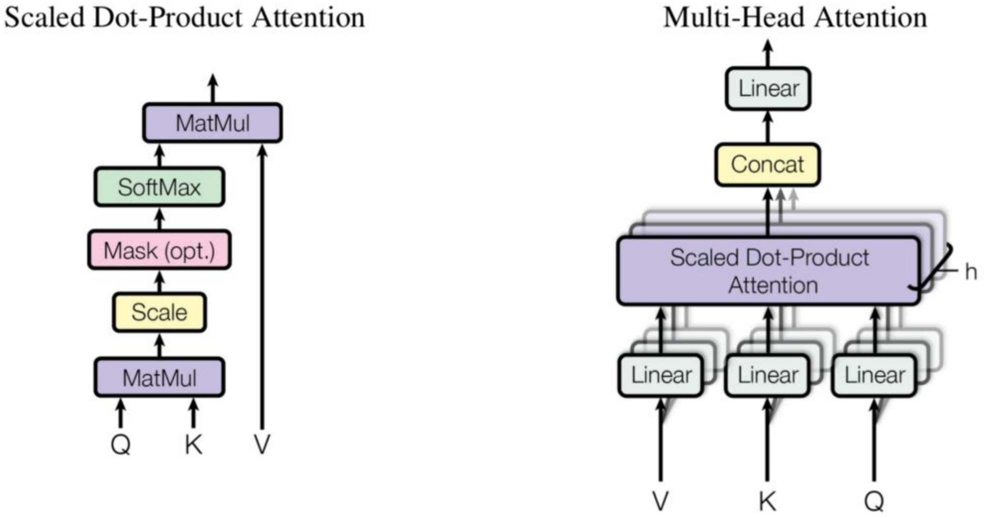
One layer of an LSTM over the course of three time steps, moving from left to right. Note the arrows (C_t & h_t), showing the passage of information from previous time steps.

Transformers

The structure of a Transformer encoder. The gray box labeled N✕ is one layer and is repeated N times in a model with N layers.



Transformers – Attention Mechanism

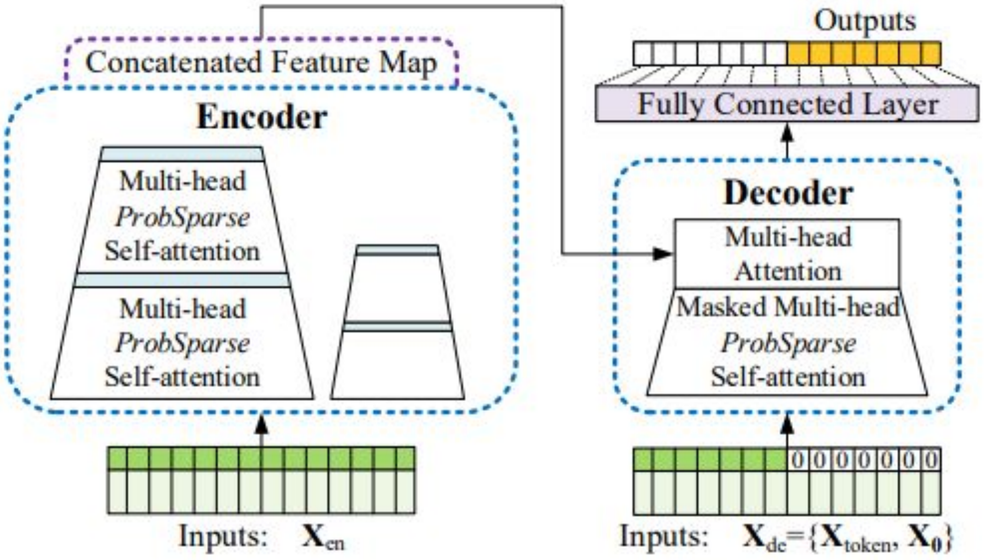


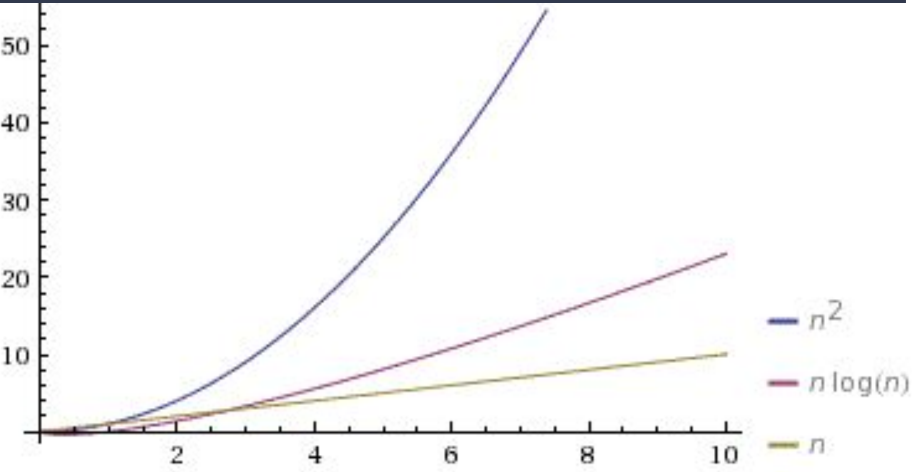
Left: An illustration of the Transformer's attention mechanism, called an attention head.

Right: Our Transformer uses a multi-head attention mechanism, 8 attention heads running in parallel.

Informers

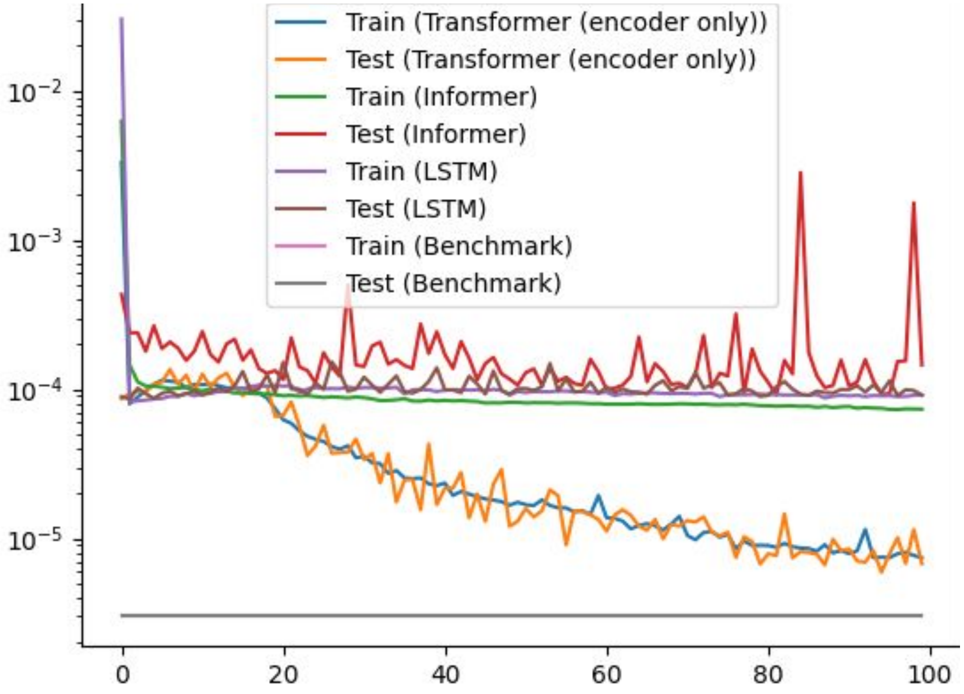
Below: O(N^2) vs. O(N log(N)) time complexity. As the length of the input sequence grows, the Informer's time complexity grows with the pink curve. It's much more efficient than the normal Transformer's time complexity, which grows with the blue curve.



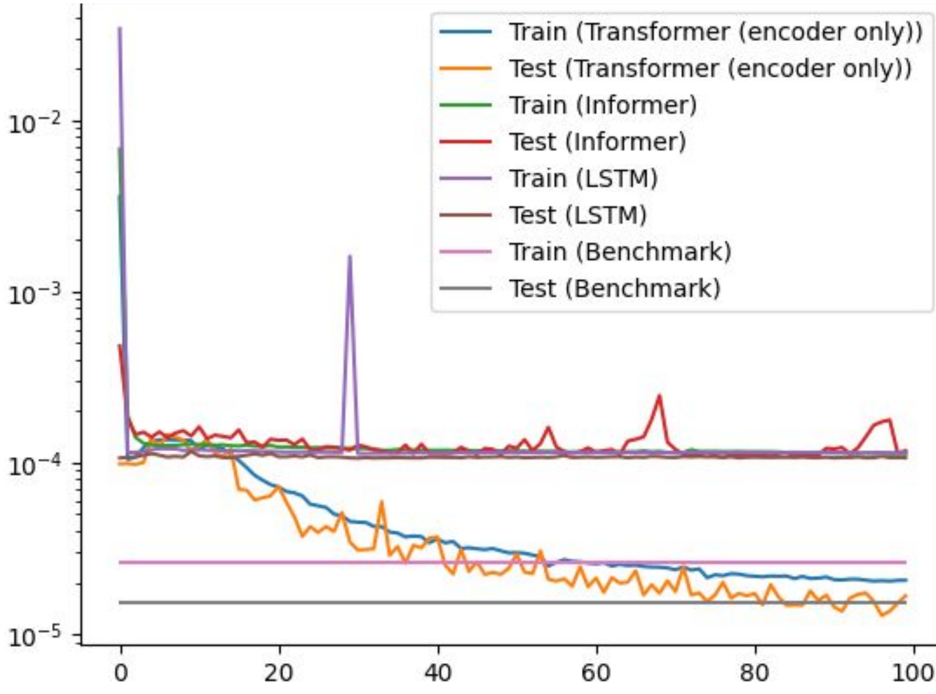


Above: An Informer machine learning model. Unlike the Transformer that we use, the Informer has a separate encoder and decoder, as well as using different attention layers.

Results



120 samples per second

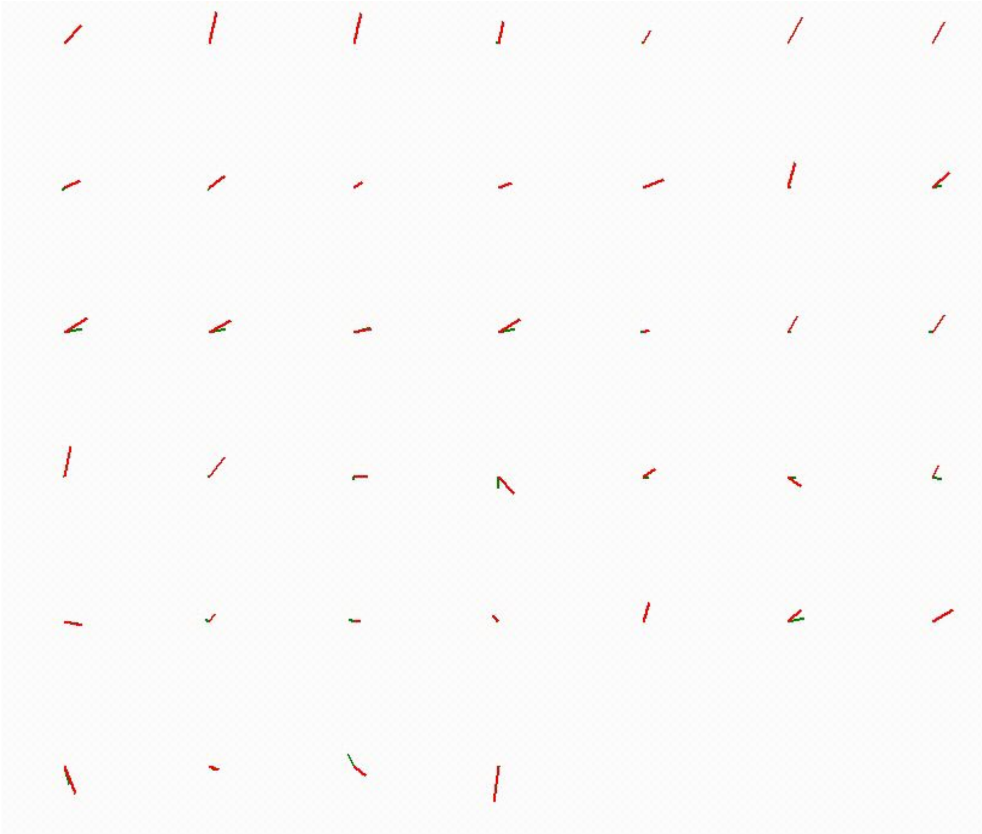


30 samples per second

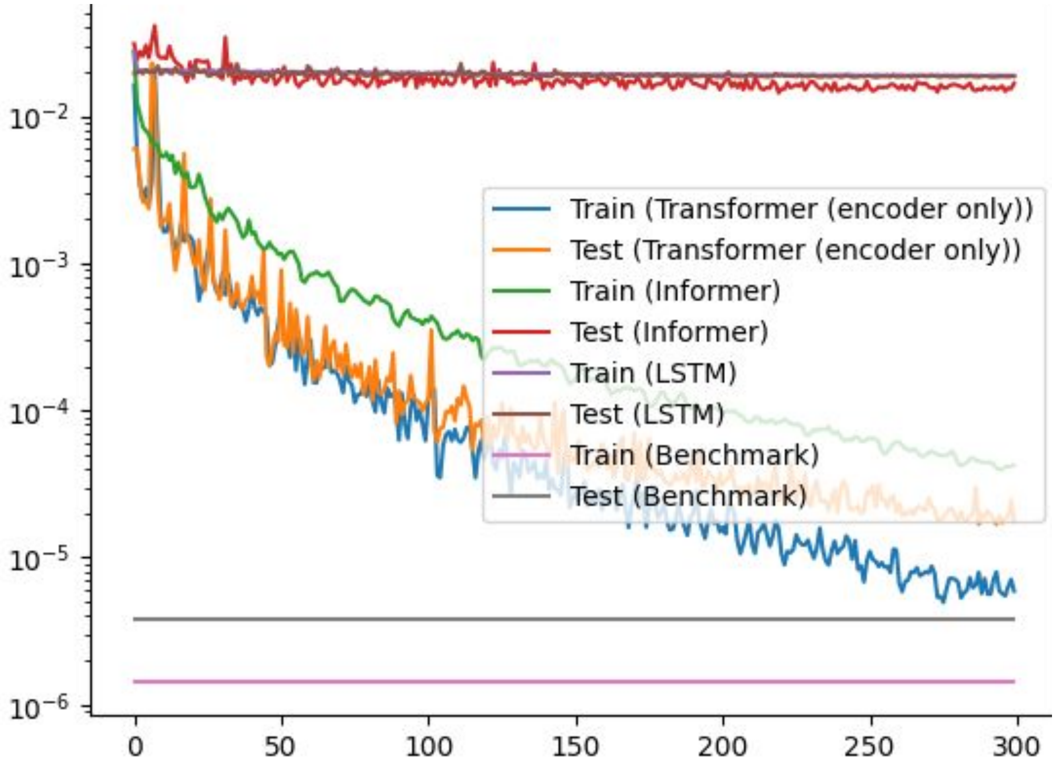
Results – Velocity Prediction GIF

Each pair of lines is the velocity of one point on the test subject's body.

Green: Actual velocity. Red: Predicted velocity



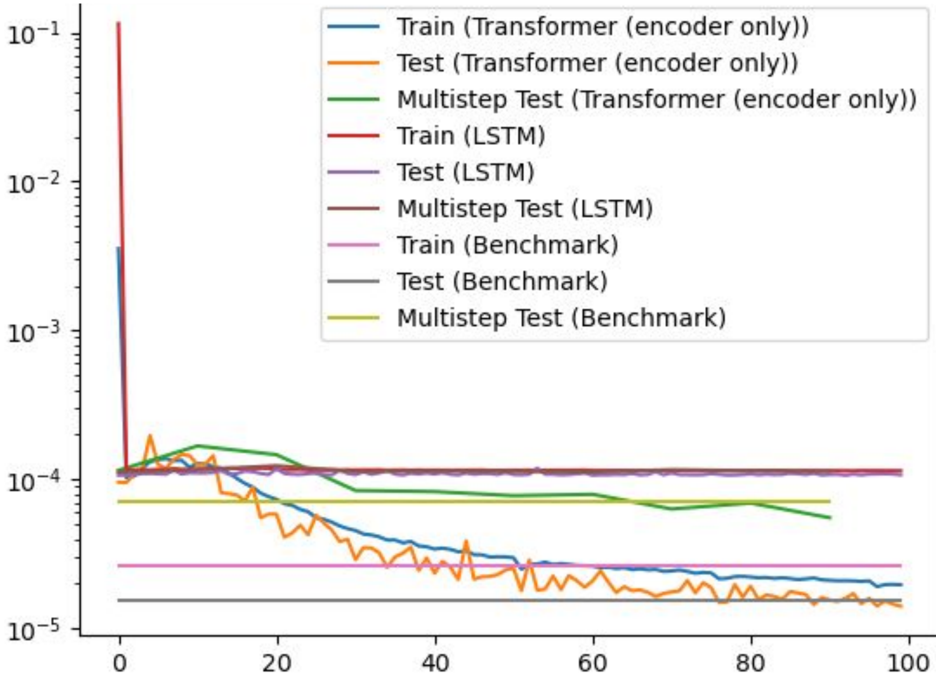
Results – Position



120 samples per second

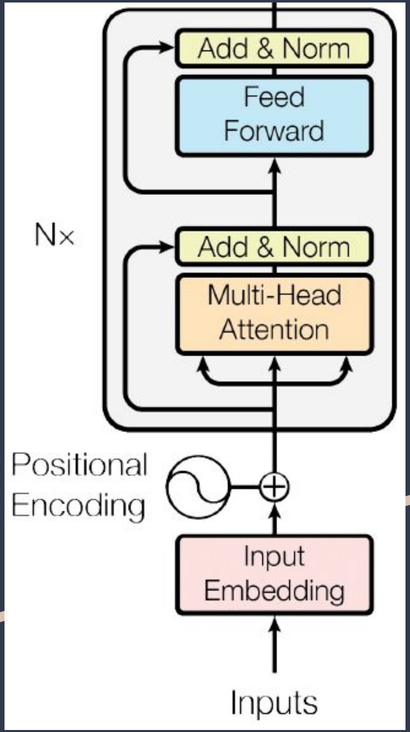
Results – Multistep

120 samples per second



30 samples per second

Conclusion



- **The Transformer does outperform the LSTM!!**
- We trained a Transformer, Informer, and an LSTM as a benchmark on CMU's kitchen dataset
- Transformers use an attention mechanism to find long-term dependencies
- Others have tried testing on time series problems, but we tried on kinematics

Future Work

- Train on other data
- Pyraformer: Uses a unique pyramidal attention mechanism
- LogTrans: Nodes only attend to log(L) frames in a sequence of length L
- Compare varying sizes of each model and study effects of individual aspects
- Multistep work:
  - Promising, but can have improvement
  - Faster computers and better hyperparameters
- Changing frame rates

Acknowledgements

We would like to thank:

- MIT PRIMES
- Andrew Gritsevskiy, our mentor
- Prof. Devadas
- Our families and friends