Deep Learning Transformers for Non-cyclical Kinematics

Coleman DuPlessie, Eddie Wei

December 31, 2022

Transformers are powerful machine learning models that are especially good at capturing long-distance relationships in data. However, they have not been applied to human kinematics, a field which has seen significant success from the application of other machine learning models.

Abstract

Machine learning is a useful tool in the field of kinematics because of its ability to easily analyze high-dimensional temporal data and recognize patterns that are often not discernible to humans. Many machine learning models have already been applied to human kinematics, yet the transformer, a model that is especially good at capturing long-distance relationships in data, has not yet been applied to this field. Because common models such as LSTMs perform much worse on non-cyclical data than on cyclical data, their usefulness in the field of kinematics is limited. We theorize that, because Transformers can better represent long-term dependencies, they will achieve superior performance on tasks in this field, where the time series data is significantly aperiodic. In this work, we have compared Transformers and similar models to an LSTM model and a heuristic benchmark on non-cyclical, 3-dimensional positional data from CMU's Quality of Life Grand Challenge Kitchen dataset and found that vanilla Transformers are able to outperform both LSTMs and simple heuristics.

1 Introduction

Kinematics prediction problems for machine learning can be divided into two broad categories: cyclical and non-cyclical prediction. Cyclical tasks focus on predicting a generally consistent motion, like the future pose of someone walking on a treadmill at a constant speed, whereas non-cyclical tasks require predicting movement that does not follow a simple pattern. Non-cyclical kinematics are much more applicable to everyday life, because people rarely do things that are perfectly cyclical. Instead of only being able to predict basic tasks such as the kinematics of walking, jumping, standing up, etc., we would figure out how an individual combines all these basic tasks to perform more complex tasks that require thought and a process that our model needs to learn. By refusing to limit ourselves to a single, repetitive motion, we greatly broaden the range of possible kinematics datasets

and problems that we can approach. Typically, non-cyclical prediction tasks are much more difficult than their cyclical counterparts, because there are many more variables involved, and models must figure out what is going to happen next, rather than knowing that the data is going to almost repeat itself. Therefore, most studies on using machine learning for kinematics have only focused on cyclical prediction tasks.

In this paper, we train Transformer-like models (and an LSTM as a benchmark) on CMU's Quality of Life Grand Challenge Kitchen dataset, which is noncyclical. The models are trained to predict the subject's position one frame in the future (ranging from 8 to 33 milliseconds). Although accurately predicting the kinematics of cyclical motion is useful, accurately predicting the kinematics of non-cyclical motion is much more applicable to everyday life, since humans rarely do things that are perfectly cyclical.

2 Background

LSTMs have been trained on kinematics problems before. For example, [Zar+21] attempts to predict the acceleration and angular velocity of the lower limbs during walking. However, this is a cyclical kinematics problem; the subjects are simply doing the same thing over and over. Instead of learning general human kinematics, the LSTMs learn the kinematics of normal humans walking, and they are minimally useful on other prediction tasks. For example, the LSTM would never be able to effectively predict irregularities in the person's walking, such as falling, because falling isn't part of the cyclical pattern they were trained on.

We attempt to train neural networks on people moving about naturally in a kitchen, cooking a meal, in the hope that our networks can perform reasonably well on a wide variety of different motions. However, because we are tackling a more general problem, it's much harder because there are fewer constants to rely on. For example, when predicting people walking, a neural network can safely learn that when the left foot touches the ground, the right foot will soon lift off the ground. However, on more general prediction tasks, the neural network cannot assume that the subject will keep walking. We theorize that, in order to make accurate predictions on non-cyclical datasets, we must be able to make use of long-term dependencies. This is because there is much more variation in non-cyclical data, and our prediction task requires the model to accurately find the overall scope of what the subject is doing (e.g. walking to the stove), as opposed to a strictly granular understanding (e.g. picking up the left foot). On the other

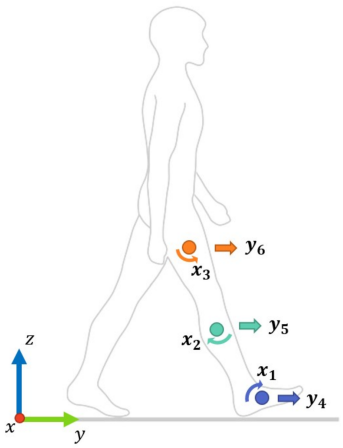


Figure 1: A normal kinematics problem for an LSTM.

hand, in cyclical data we know that the future is going to be very similar to the past, so a model only needs to know what happened recently (since, hopefully, the model has learned the pattern of the motions) to predict what is going to happen next. Although LSTMs can make use of shorter-term dependencies, we need a different type of model to more accurately make use of these long-term dependencies.

Transformers [Vas+17] are extremely useful for time-series problems because they can model dependencies of any length without a loss in detail, due to the attention mechanism. Recently, Transformers have been applied to problems involving time-series data with great success, including many papers ([Li+19], [Liu+21], [Zho+21]) using slightly modified Transformers to achieve record-breaking performance on time-series problems. However, Transformers have not yet been extensively applied to kinematics problems, so we decided to test it out.

3 The Dataset

We began by experimenting with the PoseTrack18 dataset [tea18]. PoseTrack18 was developed for machine learning models to estimate people's two-dimensional poses from a video. We originally trained a small LSTM on PoseTrack's labels (the actual poses of people) in an attempt to predict the future labels. However, PoseTrack was not an ideal dataset. Not only were the sets of datapoints extremely short (video clips of approximately one second), many were unusable because people would often be partially out of the frame. Because of these problems, there was not enough data to train a reasonably-sized neural network on PoseTrack18.

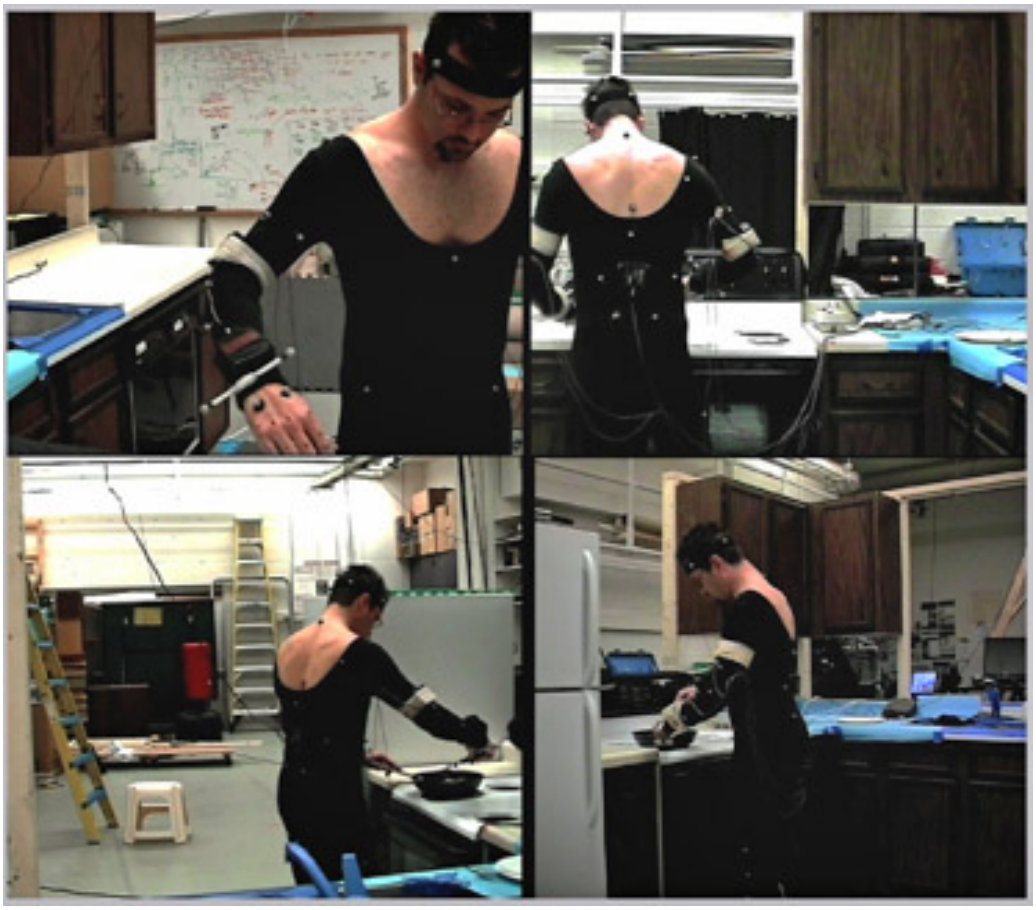


Figure 2: The creation of the Kitchen pilot dataset.

Instead of PoseTrack, we chose to train our models on the pilot portion of CMU's Quality of Life Grand Challenge Kitchen dataset. The Kitchen pilot dataset consists of several people cooking several dishes in a kitchen while being monitored by several different sensors, including multiple cameras, microphones, and a motion-capture suit. We use the 3D motion-capture data (in the form of .c3d files) as both our data and labels. The motion-capture system captures the positions of 51 key points on the human body (as well as the positions of a few inanimate objects the subject interacts with, which we ignore). We use 18 of the 20 motion captures in the pilot dataset (the other two were formatted differently from the 18 we used). We preprocess the .c3d files by normalizing them, removing any frames where the subject's pose is lost, and differentiating the position to find the velocity. We then store the preprocessed and batched data to be loaded at train and test time. When running on each batch, we do not predict for the first 100 frames, so that the models always have a minimum number of past frames to base their predictions off of. We store

the dataset in two forms, at 120 frames per second, the framerate it originally had, and downsampled to 30 frames per second.

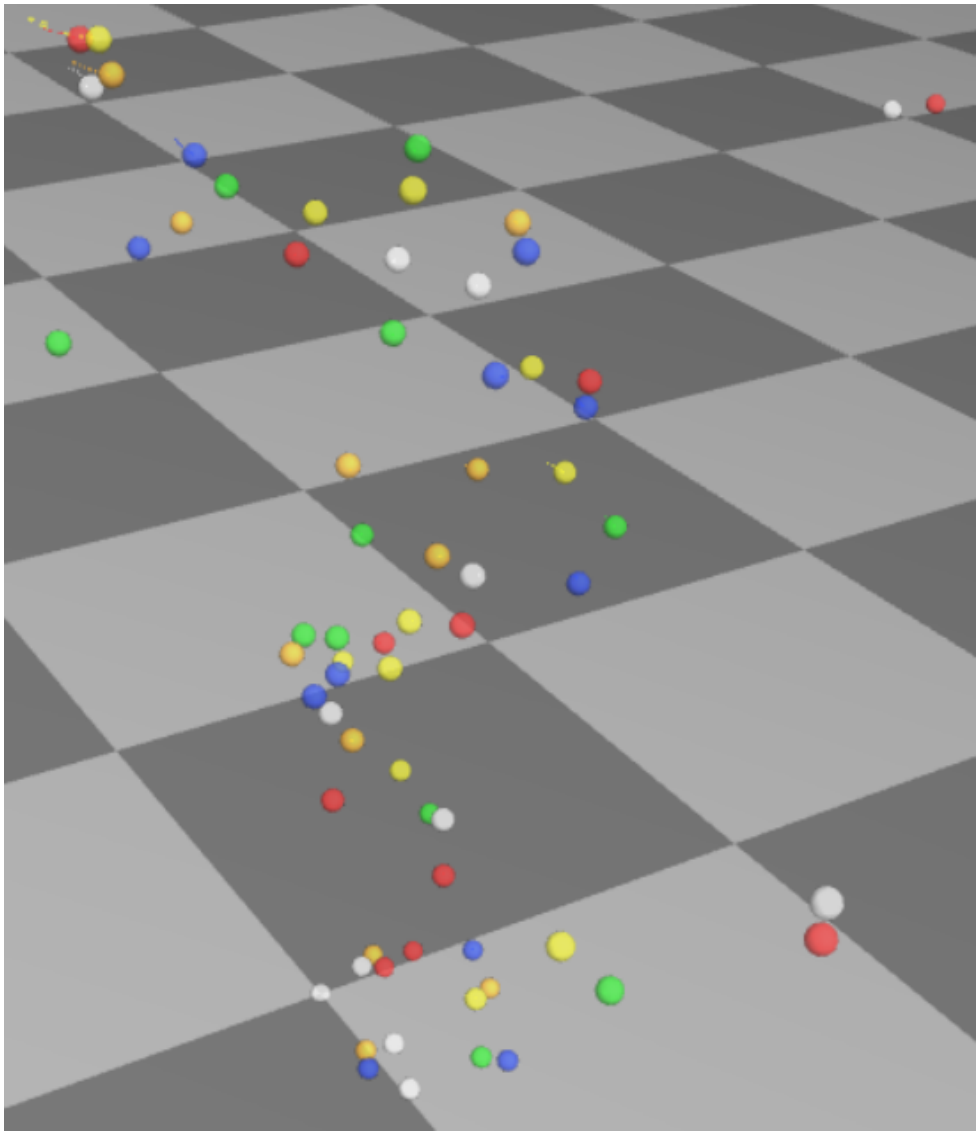


Figure 3: A visualization of a frame of the motion-capture data. Each dot is a datapoint (either on the subject or on an inanimate object nearby) captured by the motion-capture system.

4 The Models

We have trained three separate models on the Kitchen pilot dataset at both framerates in addition to a simple heuristic benchmark.

4.1 Heuristic Benchmark

We benchmark all our models against a simple "model" that always predicts that the person will remain where they last were, ignoring all but the $n-1$th frame, which it copies exactly.

4.2 LSTM Benchmark

Since LSTMs are currently the standard, state-of-the-art model for kinematics, we also benchmark our Transformer-like models against them. We use a fairly standard LSTM model as in [HS97] with no dropout.

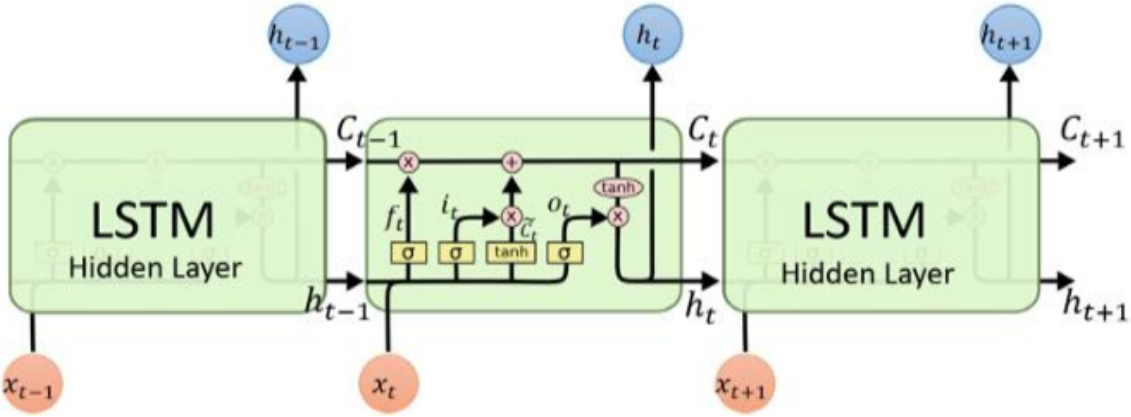


Figure 4: A single layer of LSTM nodes (Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

4.3 Transformer

We use a Transformer model based off of the encoder in Attention Is All You Need [Vas+17], with 8 attention heads. Unlike with more traditional Transforer tasks such as natural language processing, kinematics data is non-discrete. Therefore, we do not need a tokenizer, which we replace with a simple fully connected layer. We then add the output of our positional encoding and input it into the Transformer itself. We then use the inverse of the first, fully-connected layer to get our final prediction. By using the encoding layer in reverse instead of a different fully connected layer, we ensure that we use the same transformation both times. This means that, if all the Transformer's weights were zero, it would behave like our heuristic benchmark, because Transformers add the output of the $n-1$th layer to the output of the nth layer, and therefore, the Transformer's input would carry through unchanged and become its output.

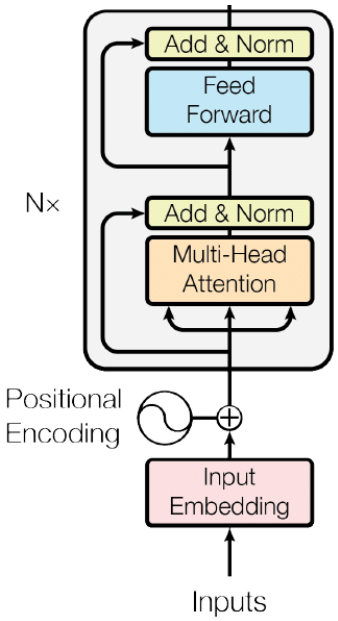


Figure 5: Our Transformer encoder architecture. (Note that, to get our output, we multiply by the inverse of the input embedding, for which we just use a fully-connected layer.

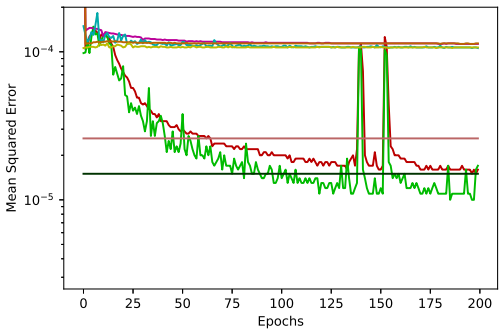
4.4 Informer

Finally, we also test an Informer [Zho+21], a Transformer-like model that uses a unique *ProbSparse* attention mechanism for $O(L\,log\,L)$ time and space complexity and self-attention distilling to decrease the size of successive layers. Like our Transformer, our Informer has 8 attention heads. Unlike our Transformer, the Informer is an encoder-decoder model, with half (rounded up) of the layers being part of the encoder and the other half of the layers being part of the decoder.

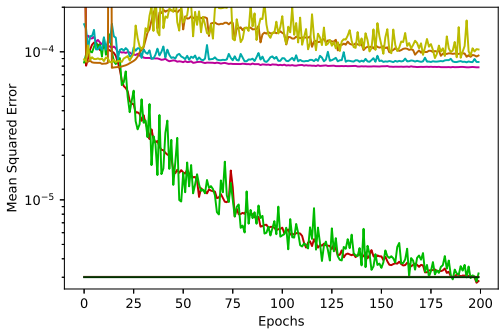
5 Results

We trained our models using the Adam optimizer [KB14] with a learning rate of 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$. Our loss function was Mean Squared Error.

As shown by these graphs, the Transformer slightly outperforms our benchmark when training on the 30 fps dataset, but is inferior to the benchmark when training on the 120 fps dataset. The Informer and LSTM perform significantly worse than the Transformer and benchmark, both converging to a loss of approximately 10^{-4} at both framerates.

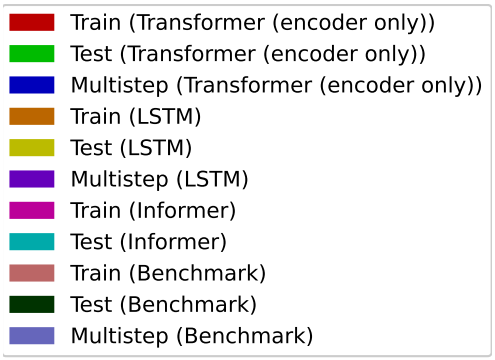


(a) MSE losses of all our models over 200 epochs with 30 FPS data



(b) MSE losses of all our models over 200 epochs with 120 FPS data

Figure 6: Loss graphs created over the course of our models' training. (Not shown: all models' train losses in the first epoch are on the order of 10^{-2})



Legend

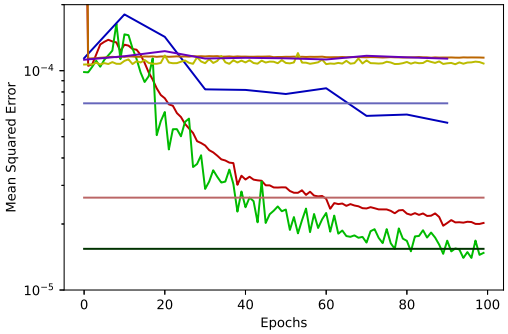


Figure 7: MSE losses of our Transformer and LSTM predicting 3 frames into the future at 30 FPS. (Not shown: all models' train losses in the first epoch are on the order of 10^{-2})

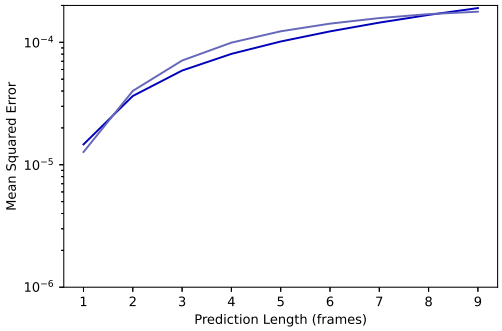
5.1 Multistep Results

We have also experimented with training Transformers and LSTMs to predict multiple steps into the future. Unlike traditional multistep Transformers, in which the Transformer is trained from the beginning to predict some number of frames into the future, we train our multistep Transformer on predicting just one frame into the future, then test it on both single- and multi-frame prediction. This has the advantage of being easily adjustable to predict any number of frames into the future without having to retrain the model.

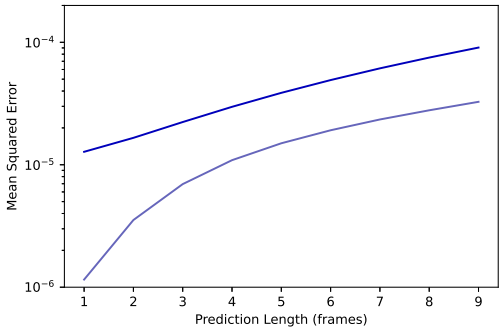
Figure 7 shows that, as expected, the LSTM consistently fails to perform and the Transformer and benchmark perform worse when predicting 3 steps into the future as

opposed to predicting just one. However, the gap between the Transformer and benchmark widened. In other words, the Transformer's accuracy decays slower than the benchmark's does as they predict more steps into the future.

Together with the method we used to train our Transformer, this is a very good trend: the Transformer gets comparatively better the farther into the future we predict, and, because of how it was trained, it can easily predict an arbitrary number of timesteps into the future where other Transformers have to resort to various hacks that reduce both their accuracy and efficiency if they want to predict more timesteps into the future than they were trained to. We have tested out Transformer's predictions up to 9 timesteps in the future, equivalent to 0.3 seconds. As can be seen in Figure 8, it is able to consistently beat our benchmark by a small, but not insignificant margin when predicting between 2 and 8 frames into the future, and is less consistent when predicting 1 or 9 timesteps ahead.



(a) Transformer and Benchmark losses at 30 FPS predicting 1-9 frames into the future



(b) Transformer and Benchmark losses at 120 FPS predicting 1-9 frames into the future

Figure 8: Our Transformer and benchmark's losses when predicting varying distances into the future.

6 Conclusion

In this work, we studied the unique difficulties inherent in non-cyclical kinematics prediction tasks and compared multiple machine learning models' performances. As hypothesized, we can see that the transformer does indeed outperform other models in kinematics prediction, most notably the "state-of-the-art" LSTM. It is more accurate at determining the "bigger picture," or what the person is going to do and the steps they are going to take, rather than the nuances of each individual step. As a result the transformer was the only model able to have a consistently decreasing loss graph.

7 Future Work

In the future, we plan to compare several other transformer-like models, including Pyraformer [Liu+21], which uses a unique pyramidal attention mechanism and LogTrans [Li+19], whose nodes only attends to $\log(L)$ frames in a sequence of length L. We also plan to compare varying sizes of each model. We would also like to study the effect of individual aspects of these models, including, but not limited to, their embeddings and their activation functions.

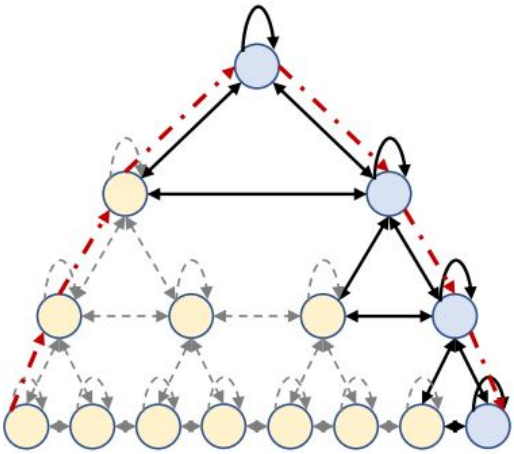


Figure 9: The Pyraformer's attention system.

Finally, we may try to test our models on other, more ambitious non-cyclical datasets; not just people cooking in a kitchen. If we are able to obtain good results on a dataset of something important, our models could have real-life implications. Also, hopefully this would be able to again confirm our results that transformers are the best model to use for predicting non-cyclical kinematics.

8 Acknowledgements

The data used in this paper was obtained from kitchen.cs.cmu.edu and the data collection was funded in part by the National Science Foundation under Grant No. EEEC-0540865.

We wouldn't have been able to do this if it weren't for the MIT PRIMES program and the assistance of our great mentor, Andrew Gritsevskiy, who directed us to the resources we needed and helped us figure out the state-of-the-art, and Professor Srini Devadas, who generously let us train our models on his machine.

We would also like to thank Dr. Daniel Nolte of MathWorks, who provided the original idea behind this research. This project was partially funded by MathWorks.

References

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[KB14] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[Li+19] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/6775a0635c302542da2c32aa19d86be0-Paper.pdf.

[Liu+21] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting". In: *International Conference on Learning Representations*. 2021.

[tea18] The DensePose team. *PoseTrack - Dataset and Benchmark*. 2018. URL: https://posetrack.net/.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need". In: (2017). DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[Zar+21] Abdelrahman Zaroug, Alessandro Garofolini, Daniel TH Lai, Kurt Mudie, and Rezaul Begg. "Prediction of gait trajectories based on the Long Short Term Memory neural networks". In: *PLoS One* 16.8 (2021), e0255597.

[Zho+21] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting". In: *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*. Vol. 35. 12. AAAI Press, 2021, pp. 11106–11115.