Generating Animated Videos of Human Activities from Natural Language Descriptions

Angela S. Lin[1]*, Lemeng Wu[1]*, Rodolfo Corona[2], Kevin Tai[1], Qixing Huang[1], Raymond J. Mooney[1]
alin@cs.utexas.edu, lm.wu@utexas.edu, r.coronarodriguez@uva.nl, kevin.r.tai@utexas.edu, huangqx@cs.utexas.edu, mooney@cs.utexas.edu

Introduction

Generating realistic character animations is of great importance in computer graphics and related domains. In this paper, we introduce a sequence-to-sequence model that maps a natural language (NL) description to an animation of a humanoid skeleton.

This problem is challenging because:
- the output is much longer and higher dimensional than input
- language is ambiguous
- motion capture (mocap) data is limited
- there is a large imbalance in activities

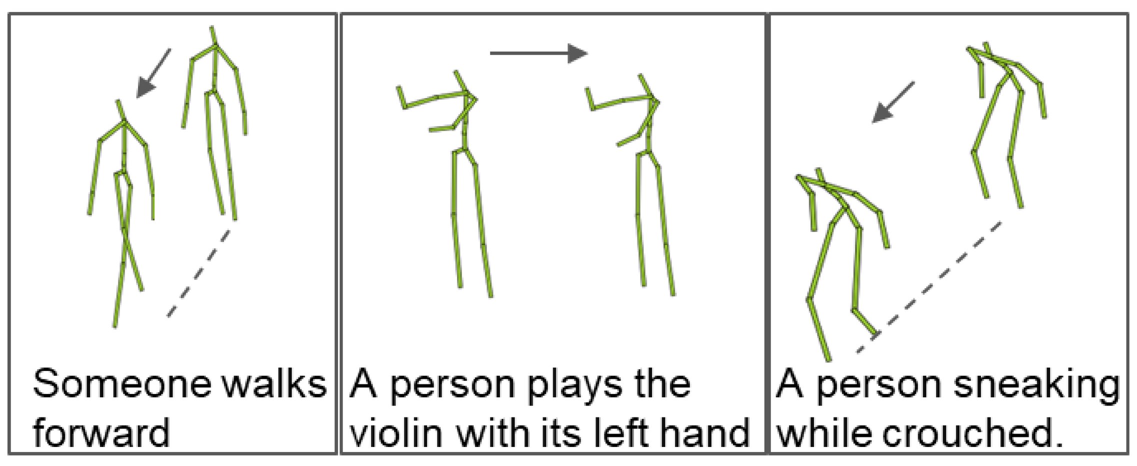
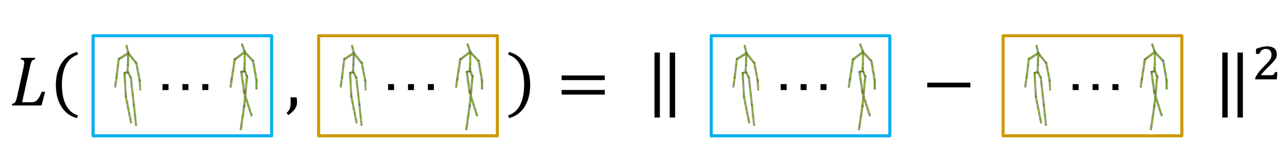


Figure 1: Examples from our dataset. Solid arrows show the passage of time and dotted lines show movement in space.

Approach

Step 1. We pretrain the animation decoder using an autoencoder objective. We train the autoencoder to reconstruct the input using the L2 distance between the predicted and gold-standard animation as the loss function L:



We use the data representation proposed by Holden et al. [2]:

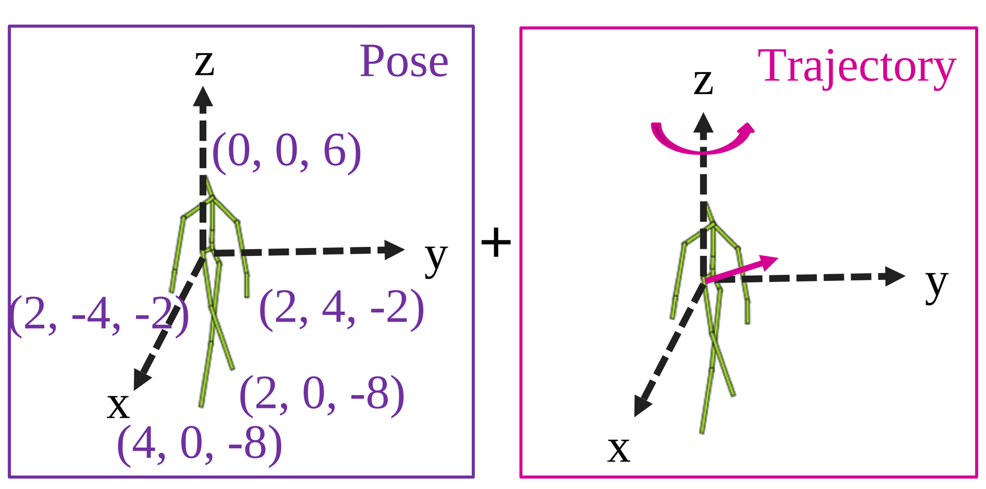


Figure 2: (Left) The character's pose is represented by the joint positions in the local coordinate frame. (Right) The character's trajectory is represented by the rotational velocity about the z-axis and translational velocity on the yz-plane.

Network Architecture:
- The decoder is the GRU with residual connections proposed by Martinez et al. [4]
- Trajectory prediction module is inspired by Agrawal et al. [1]

Training data: KIT Motion-Language Dataset [5] and Human3.6M [3]

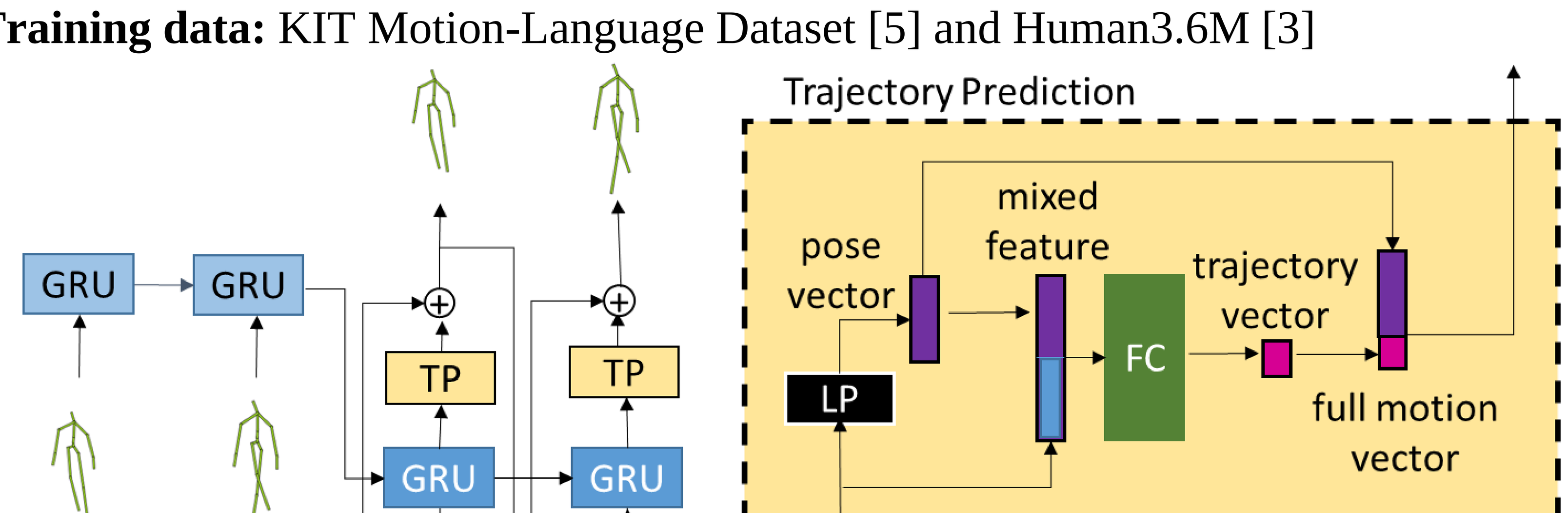


Figure 3: (Left) The network architecture for the autoencoder. (Right) The network architecture for the trajectory prediction (TP) module. LP indicates a linear projection layer and FC indicates a fully connected layer.

Step 2: We train the end-to-end network for generating animations from text using the same loss function.
Training data: KIT Motion-Language Dataset [5] and additional paired data that we collected on Amazon Mechanical Turk (AMT)

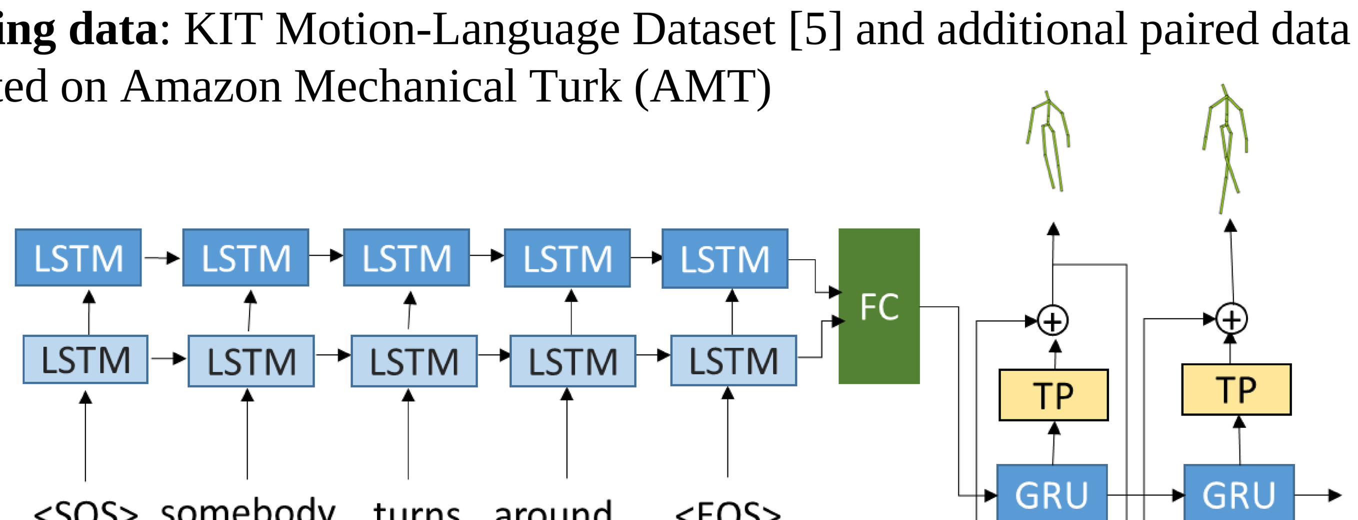


Figure 4: Network architecture for our full pipeline.

Experimental results

Baseline methods:
- **Nearest neighbor:** Our simplest baseline is a standard TF-IDF bag-of-words nearest neighbor method.
- **Plappert et al. [6]'s method:** This method also generates animations from text descriptions, but their animated character moves in place because their model does not predict the character's trajectory.

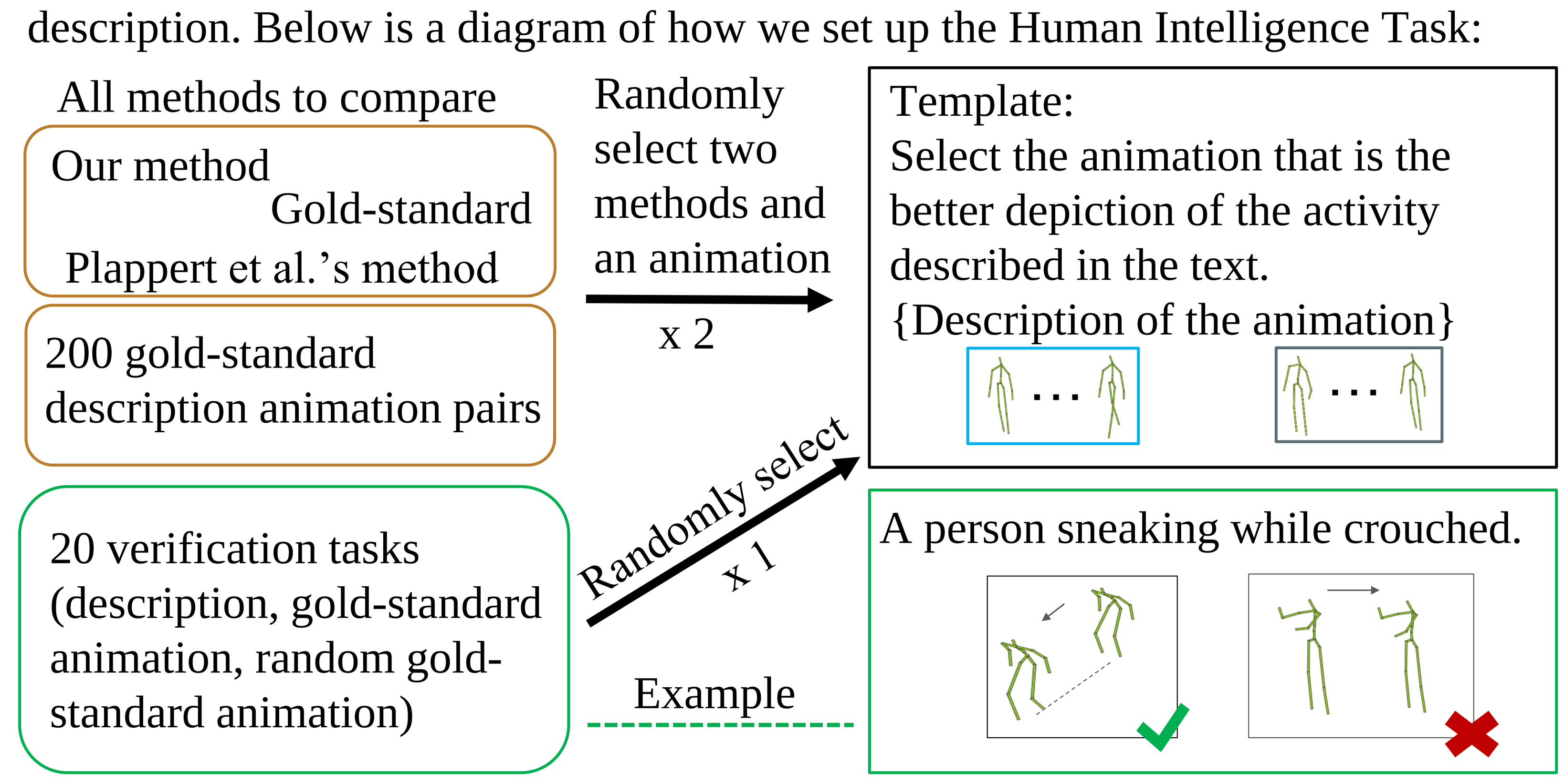
Evaluation metrics:
Dynamic time warping mean absolute error (DTW-MAE):
1. Use the dynamic time warping algorithm to warp animations to same length
2. Compute the absolute error at each time step and average across time

DTW-MAE-T is DTW-MAE on animations with the trajectory information removed.

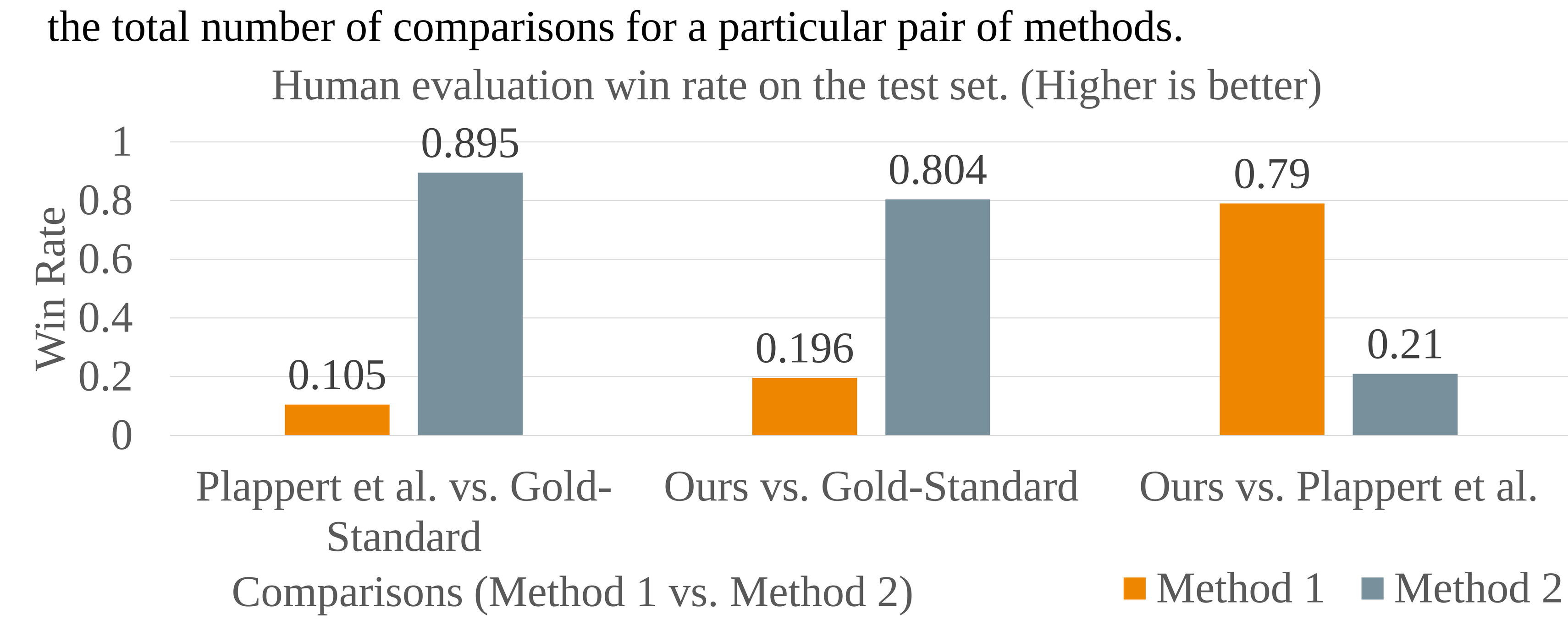
	DTW-MAE	DTW-MAE-T
Nearest neighbors	9.80 ± 5.79	9.76 ± 5.77
Plappert et al.'s method	N/A	8.44 ± 3.99
Our method	9.74 ± 4.34	9.71 ± 4.32

Table 1: Dynamic time warping mean absolute error metric on the test set. (Lower is better)

Human evaluation: We conducted a crowd-sourced human evaluation of the generated animations using AMT to evaluate the generated animations for faithfulness to the description. Below is a diagram of how we set up the Human Intelligence Task:



The win rate is defined as the number of comparisons won by the method divided by the total number of comparisons for a particular pair of methods.



Discussion

Evaluation metrics:
- DTW-MAE results do not agree well with human evaluation win rate
- We need better automatic metrics for comparing animations
- Our method outperforms Plappert et al. [6]'s method on the human evaluation win rate but it might not be fair because many descriptions describe global movement
- There is room for improvement for both animation generation methods

Main failure cases:
- Producing animations that fail to depict the description for rare activities
- Producing animations that are physically impossible

Future work:
- Improve our loss function to capture more semantic meaning
- Explore physically-based controller approaches to generate more realistic animations

References

[1] Pulkit Agrawal, Ashvin V. Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to Poke by Poking: Experiential Learning of Intuitive Physics. In *Advances in Neural Information Processing Systems*, pages 5074–5082, 2016.
[2] Daniel Holden, Jun Saito, and Taku Komura. A Deep Learning Framework for Character Motion Synthesis and Editing. *ACM Transactions on Graphics*, 35(4):138:1–138:11, July 2016. ISSN0730-0301. doi:10.1145/2897824.2925975. URL http://doi.acm.org/10.1145/2897824.2925975.
[3] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014.
[4] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. *In Proceedings of CVPR*, pages 4674–4683. IEEE, 2017.
[5] Matthias Plappert, Christian Mandery, and Tamim Asfour. The KIT Motion-Language Dataset. *Big Data*, 4(4):236–252, December 2016. doi: 10.1089/big.2016.0028. URL http://dx.doi.org/10.1089/big.2016.0028.
[6] Matthias Plappert, Christian Mandery, and Tamim Asfour. Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *Robotics and Autonomous Systems*, 109:13–26, 2018.

The University of Texas at Austin
Department of Computer Science
College of Natural Sciences