Generative Model-Enhanced Human Motion Prediction

Anthony Bourached
Department of Neurology
University College London
London, UK
ucabab6@ucl.ac.uk

Ryan-Rhys Griffiths
Department of Physics
University of Cambridge
Cambridge, UK
rrg27@cam.ac.uk

Robert Gray
Department of Neurology
University College London
London, UK
r.gray@ucl.ac.uk

Ashwani Jha
Department of Neurology
University College London
London, UK
ashwani.jha@ucl.ac.uk

Parashkev Nachev
Department of Neurology
University College London
London, UK
p.nachev@ucl.ac.uk

Abstract

The task of predicting human motion is complicated by the natural heterogeneity and compositionality of actions, necessitating robustness to distributional shifts as far as out-of-distribution (OoD). Here we formulate a new OoD benchmark based on the Human3.6M and CMU motion capture datasets, and introduce a hybrid framework for hardening discriminative architectures to OoD failure by augmenting them with a generative model. When applied to current state-of-the-art discriminative models, we show that the proposed approach improves OoD robustness without sacrificing in-distribution performance, and can facilitate model interpretability. We suggest human motion predictors ought to be constructed with OoD challenges in mind, and provide an extensible general framework for hardening diverse discriminative architectures to extreme distributional shift.

1 Introduction

Human motion is naturally intelligible as a time-varying graph of connected joints constrained by locomotor anatomy and physiology. Its prediction allows the anticipation of actions with applications across healthcare [1, 2], physical rehabilitation and training [3, 4], robotics [5, 6, 7], navigation [8, 9, 10, 11], manufacture [12], entertainment [13, 14, 15], and security [16, 17].

The favoured approach to predicting movements over time has been purely inductive, relying on the history of a specific class of movement to predict its future. For example, state space models [18] enjoyed early success for simple, common or cyclic motions [19, 20, 21]. The range, diversity and complexity of human motion has encouraged a shift to more expressive, deep neural network architectures [22, 23, 24, 25, 26, 27, 28], but still within a simple inductive framework.

This approach would be adequate were actions both sharply distinct and highly stereotyped. But their complex, compositional nature means that within one category of action the kinematics may vary substantially, while between two categories they may barely differ. This has two crucial implications. First, any modelling approach that lacks awareness of the full space of motion possibilities will be vulnerable to poor generalisation and brittle performance in the face of kinematic anomalies. Second, the very notion of *In-Distribution* (ID) testing becomes moot, for the relations between different actions and their kinematic signatures are plausibly determinable only across the entire domain of

1st NeurIPS workshop on Interpretable Inductive Biases and Physically Structured Learning (2020), virtual.

action. A test here arguably needs to be *Out-of-Distribution* (OoD) if it is to be considered a robust test at all.

To our knowledge, current predictive models of human kinematics neither quantify OoD performance nor are designed with it in mind. There is therefore a need for two *frameworks*, applicable across the domain of action modelling: one for *hardening* a predictive model to anomalous cases, and another for *quantifying* OoD performance with established benchmark datasets. General frameworks are here desirable in preference to new models, for the field is evolving so rapidly greater impact can be achieved by introducing mechanisms that can be applied to a breadth of candidate architectures, even if they are demonstrated in only a subset. Our approach here is founded on combining a latent variable generative model with a standard predictive model, illustrated with the current state-of-the-art discriminative architecture [26, 29]. [30], take an analogous approach, regularising an encoder-decoder model for brain tumor segmentation on magnetic resonance images by simultaneously modelling the distribution of the data using a variational autoencoder (VAE) [31]. Here the aim is to achieve robust performance within a low data regime, which coincides with the demand for OoD generalisation.

In short, our contributions to the problem of achieving robustness to distributional shift in human motion prediction are as follows:

1. We provide a framework to benchmark OoD performance on the most widely used open-source motion capture datasets: Human3.6M [32], and CMU-Mocap[1], and evaluate state-of-the-art models on it.
2. We present a framework for hardening deep feed-forward models to OoD samples. We show that the hardened models are fast to train, and exhibit substantially improved OoD performance with minimal impact on ID performance.

We begin section 2 with a brief review of human motion prediction with deep neural networks, and of OoD generalisation using generative models. In section 3, we define a framework for benchmarking OoD performance using open-source multi-action datasets. We then turn in section 4 to the architecture of the generative model and the overall objective function. Section 5 presents our experiments and results. We conclude in section 6 with a summary of our results, current limitations, and caveats, and future directions for developing robust and reliable OoD performance and a quantifiable awareness of unfamiliar behaviour.

2 Related Work

Deep-network based human motion prediction. Historically, sequence-to-sequence prediction using Recurrent Neural Networks (RNNs) have been the de facto standard for human motion prediction [22, 33, 24, 34, 35, 27]. Currently, the state-of-the-art (SOTA) is dominated by feed forward models [23, 25, 26, 29]. These are inherently faster and easier to train than RNNs. The jury is still out, however, on the optimal way to handle temporality for human motion prediction. Meanwhile, recent trends have overwhelmingly shown that graph-based approaches are an effective means to encode the spatial dependencies between joints [26, 29], or sets of joints [27]. In this study, we consider the SOTA models that have graph-based approaches with a feed forward mechanism as presented by [26], and the subsequent extension which leverages motion attention, [29]. We show that these may be augmented to improve robustness to OoD samples.

Generative models for Out-of-Distribution hardening. [30] use a Variational Autoencoder (VAE) [31] to regularise an encoder-decoder architecture with the specific aim of better generalisation. By simultaneously using the encoder as the recognition model of the VAE, the model is encouraged to base its segmentations on a complete picture of the data, rather than on a reductive representation that is more likely to be fitted to the training data. Furthermore, the original loss and the VAE's loss are combined as a weighted sum such that the discriminator's objective still dominates. Further work may also reveal useful interpretability of behaviour (via visualisation of the latent space as in [36]), generation of novel motion [37], or reconstruction of missing joints as in [38].

[1] t http://mocap.cs.cmu.edu/

3 Quantifying out-of-distribution performance of human motion predictors

Even a very compact representation of the human body such as OpenPose's 17 joint parameterisation [39] explodes to unmanageable complexity when a temporal dimension is introduced of the scale and granularity necessary to distinguish between different kinds of action: typically many seconds, sampled at hundredths of a second. Moreover, though there are anatomical and physiological constraints on the space of licit joint configurations, and their trajectories, the repertoire of possibility remains vast, and the kinematic demarcations of teleologically different actions remain indistinct. For this reason we propose to define OoD on multi-action motion capture datasets as being the scenario where only a single action is available for training and hyperparameter search. In appendix A, to show that the motion categories we have chosen, *walking* for the H3.6M dataset and *basketball* for the CMU dataset, can actually be distinguished at the time scales on which our trajectories are encoded we train a simple classifier and show that it can separate the selected ID action from the others with high accuracy (100% precision and recall for the CMU dataset). In this way OoD performance may be considered over the remaining set of actions.

4 Variational Graph Autoencoder (VGAE) Branch and Loss

[30] augment an encoder-decoder discriminative model by using the encoder as a recognition model for a Variational Autoencoder (VAE), [31, 40]. [30] show this to be a very effective regulariser. We take an analogous approach, we augment the SOTA discriminative models proposed by [26] (GCN), and [29] (att-GCN) with a generative model. For a full description of GCN and att-GCN and the problem formulation see appendix B. Here, for conjugacy with the discriminator (see appendix B), we consider the Variational Graph Autoencoder (VGAE), proposed by [41] as a framework for unsupervised learning on graph-structured data.

Here we define the first half (see figure 3 in the appendix) of the GCN model as our VGAE recognition model, with a latent variable $\mathbf{z} \in \mathbb{R}^{K \times n_z} = N(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}})$, where $\mu_{\mathbf{z}} \in \mathbb{R}^{K \times n_z}, \sigma_{\mathbf{z}} \in \mathbb{R}^{K \times n_z}$. $n_z = 8$, or 32 depending on training stability, and K is the number of joints as detailed in appendix B.

The decoder part of the VGAE has the same structure as the second half of the discriminative branch. We parametrise the output neurons as $\mu \in \mathbb{R}^{K \times (N+T)}$, and $log(\sigma^2) \in \mathbb{R}^{K \times (N+T)}$. We can now model the reconstruction of inputs as samples of a maximum likelihood of a Gaussian distribution which constitutes the second term of the negative Variational Lower Bound (VLB) of the VGAE.

We train the entire network together with the additional of the negative VLB:

$$\ell = \underbrace{\frac{1}{(N+T)K} \sum_{n=1}^{N+T} \sum_{k=1}^{K} |\hat{x}_{k,n} - x_{k,n}|}_{\text{Discriminitive loss}} - \lambda \underbrace{(log(p(\mathbf{C}|\mathbf{Z})) - KL(q(\mathbf{Z}|\mathbf{C})||q(\mathbf{Z})))}_{\text{VLB}}. \quad (1)$$

Where $\mathbf{C} \in \mathbb{R}^{K \times (N+T)}$ is the discrete cosine transformation of the inputs as detailed in appendix B. λ is a hyperparameter of the model. The overall network is $\approx 3.4M$ parameters. Furthermore, once trained, the generative model is not required for prediction and hence for this purpose is as compact as the original models.

5 Results

Appendix C describes the experimental setup, datasets, and methods in detail, including hyperparameter search. Consistent with the literature we report short-term ($< 500ms$) and long-term ($> 500ms$) predictions. In comparison to GCN, we take short term history into account (10 frames, $400ms$) for both datasets to predict both short- and long-term motion. In comparison to att-GCN, we take long term history (50 frames, 2 seconds) to predict the next 10 frames, and predict futher into the future by recursively applying the predictions as input to the model as in [29]. In this way a single short term prediction model may produce long term predictions.

We use Euclidean distance between the predicted and ground-truth joint angles for the Euler angle representation, we present some additional results in appendix D, including on CMU trained in

	Walking (ID)				Eating (OoD)				Smoking (OoD)				Discussion (OoD)			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
GCN	**0.22**	0.38	0.61	0.66	0.22	0.40	0.67	0.81	0.31	0.62	1.22	1.25	**0.30**	0.67	1.00	1.08
ours	0.23	**0.37**	**0.58**	**0.63**	**0.21**	**0.37**	**0.59**	**0.72**	**0.27**	**0.54**	**1.03**	**1.03**	**0.30**	**0.66**	**0.94**	**1.02**
	Directions (OoD)				Greeting (OoD)				Phoning (OoD)				Posing (OoD)			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
GCN	**0.38**	**0.58**	0.81	**0.90**	**0.48**	0.82	1.28	1.47	0.58	1.12	1.52	1.66	0.30	0.64	1.37	1.68
ours	**0.38**	**0.58**	**0.79**	**0.90**	0.49	**0.81**	**1.24**	**1.43**	**0.57**	**1.10**	**1.48**	**1.61**	**0.26**	**0.56**	**1.26**	**1.55**
	Purchases (OoD)				Sitting (OoD)				Sitting Down (OoD)				Taking Photo (OoD)			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
GCN	0.62	0.90	1.34	1.42	0.40	0.66	1.15	1.33	0.46	0.94	1.52	1.69	0.26	0.53	0.82	**0.93**
ours	**0.61**	**0.89**	**1.27**	**1.37**	**0.38**	**0.62**	**1.06**	**1.22**	**0.41**	**0.83**	**1.28**	**1.41**	**0.25**	**0.51**	**0.81**	0.95
	Waiting (OoD)				Walking Dog (OoD)				Walking Together (OoD)				Average (of 14 for OoD)			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
GCN	0.30	0.61	1.10	1.34	**0.51**	**0.85**	**1.16**	**1.32**	**0.20**	**0.42**	**0.65**	**0.69**	0.38	0.70	1.12	1.26
ours	**0.29**	**0.58**	**1.06**	**1.29**	0.52	0.88	1.17	1.34	0.21	0.44	0.66	0.74	**0.37**	**0.63**	**1.08**	**1.18**

Table 1: Short-term prediction of Eucildean distance between predicted and ground truth joint angles on H3.6M.

	Basketball (ID)					Basketball Signal (OoD)					Average (of 7 for OoD)				
milliseconds	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
GCN	**0.40**	0.67	**1.11**	**1.25**	**1.63**	**0.27**	**0.55**	**1.14**	**1.42**	2.18	0.36	0.65	1.41	1.49	2.17
ours	**0.40**	**0.66**	1.12	1.29	1.76	0.28	0.57	1.15	1.43	**2.07**	**0.34**	**0.62**	**1.35**	**1.41**	**2.10**

Table 2: Eucildean distance between predicted and ground truth joint angles on CMU. Full table in appendix, table 5.

3D cartesian coordinates. Table 1 reports the joint angle error for the short term predictions on the H3.6M dataset. Here we found the optimum hyperparameters to be $p_{drop} = 0.5$ for GCN, and $\lambda = 0.003$, with $p_{drop} = 0.3$ for our augmentation of GCN. The latter of which was used for all future experiments, where for our augmentation of att-GCN we removed dropout altogether. On average, our model performs convincingly better both ID and OoD. Here the generative branch works well as both a regulariser for small datasets and by creating robustness to distributional shifts.

From table 2 we can see that the superior OoD performance generalises to the CMU dataset with the same hyperparameter settings and a similar trend of the difference being larger for longer predictions for both joint angles and 3D joint coordinates. For each of these experiments $n_z = 8$.

Table 3, shows that the effectiveness of the generative branch generalises to the very recent motion attention architecture. For att-GCN we used $n_z = 32$. Here, interestingly short term predictions are poor but long term predictions are consistently better. This supports our assertion that information relevant to generative mechanisms are more intrinsic to the causal model and thus, here, when the predicted output is recursively used, more useful information is available for the future predictions.

	Walking (ID)				Eating (OoD)				Smoking (OoD)				Average (of 14 for OoD)			
milliseconds	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000
att-GCN	**55.4**	**60.5**	**65.2**	**68.7**	87.6	103.6	113.2	120.3	81.7	93.7	102.9	108.7	**112.1**	129.6	140.3	147.8
ours	58.7	60.6	65.5	69.1	**81.7**	**94.4**	**102.7**	**109.3**	**80.6**	**89.9**	**99.2**	**104.1**	113.1	**127.7**	**137.9**	**145.3**

Table 3: Long-term prediction of 3D joint positions on H3.6M. Here, ours is also trained with the att-GCN model. Full table is in appendix, table 7.

6 Conclusion

We draw attention to the need for robustness to distributional shifts in predicting human motion, and propose a framework for its evaluation based on major open source datasets. We demonstrate that state-of-the-art discriminative architectures can be hardened to extreme distributional shifts by augmentation with a generative model, combining low in-distribution predictive error with maximal generalisability. The introduction of a surveyable latent space further provides a mechanism for model perspicuity and interpretability, and explicit estimates of uncertainty facilitate the detection of anomalies: both characteristics are of substantial value in emerging applications of motion prediction, such as autonomous driving, where safety is paramount. Our investigation argues for wider use of generative models in behavioural modelling, and shows it can be done with minimal or no performance penalty, within hybrid architectures of potentially diverse constitution.

References

[1] Evelien E Geertsema, Roland D Thijs, Therese Gutter, Ben Vledder, Johan B Arends, Frans S Leijten, Gerhard H Visser, and Stiliyan N Kalitzin. Automated video-based detection of nocturnal convulsive seizures in a residential care setting. *Epilepsia*, 59:53–60, 2018.

[2] Manish Kakar, Håkan Nyström, Lasse Rye Aarup, Trine Jakobi Nøttrup, and Dag Rune Olsen. Respiratory motion prediction by using the adaptive neuro fuzzy inference system (anfis). *Physics in Medicine & Biology*, 50(19):4721, 2005.

[3] Chien-Yen Chang, Belinda Lange, Mi Zhang, Sebastian Koenig, Phil Requejo, Noom Somboon, Alexander A Sawchuk, and Albert A Rizzo. Towards pervasive physical rehabilitation using microsoft kinect. In *2012 6th international conference on pervasive computing technologies for healthcare (PervasiveHealth) and workshops*, pages 159–162. IEEE, 2012.

[4] David Webster and Ozkan Celik. Systematic review of kinect applications in elderly care and stroke rehabilitation. *Journal of neuroengineering and rehabilitation*, 11(1):108, 2014.

[5] Hema Swetha Koppula and Ashutosh Saxena. Anticipating human activities for reactive robotic response. In *IROS*, page 2071. Tokyo, 2013.

[6] Hema Koppula and Ashutosh Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *International conference on machine learning*, pages 792–800, 2013.

[7] Liang-Yan Gui, Kevin Zhang, Yu-Xiong Wang, Xiaodan Liang, José MF Moura, and Manuela Veloso. Teaching robots to predict human motion. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 562–567. IEEE, 2018.

[8] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.

[9] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[10] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4194–4202, 2018.

[11] Yijing Wang, Zhengxuan Liu, Zhiqiang Zuo, Zheng Li, Li Wang, and Xiaoyuan Luo. Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control. *IEEE Transactions on Vehicular Technology*, 68(9):8546–8556, 2019.

[12] Petr Švec, Atul Thakur, Eric Raboin, Brual C Shah, and Satyandra K Gupta. Target following with motion prediction for unmanned surface vehicle operating in cluttered environments. *Autonomous Robots*, 36(4):383–405, 2014.

[13] Akihiko Shirai, Erik Geslin, and Simon Richir. Wiimedia: motion analysis methods and applications using a consumer video game controller. In *Proceedings of the 2007 ACM SIGGRAPH symposium on Video games*, pages 133–140, 2007.

[14] Ahmadreza Reza Rofougaran, Maryam Rofougaran, Nambirajan Seshadri, Brima B Ibrahim, John Walley, and Jeyhan Karaoguz. Game console and gaming object with motion prediction modeling and methods for use therewith, April 17 2018. US Patent 9,943,760.

[15] Rynson WH Lau and Addison Chan. Motion prediction for online gaming. In *International Workshop on Motion in Games*, pages 104–114. Springer, 2008.

[16] Daehee Kim and J Paik. Gait recognition using active shape model and motion prediction. *IET Computer Vision*, 4(1):25–36, 2010.

[17] Zhuo Ma, Xinglong Wang, Ruijie Ma, Zhuzhu Wang, and Jianfeng Ma. Integrating gaze tracking and head-motion prediction for mobile device authentication: A proof of concept. *Sensors*, 18(9):2894, 2018.

[18] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[19] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pages 1345–1352, 2007.

[20] Ilya Sutskever, Geoffrey E Hinton, and Graham W Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in neural information processing systems*, pages 1601–1608, 2009.

[21] Andreas M Lehrmann, Peter V Gehler, and Sebastian Nowozin. Efficient nonlinear markov models for human motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1314–1321, 2014.

[22] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[23] Judith Butepage, Michael J Black, Danica Kragic, and Hedvig Kjellstrom. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6158–6166, 2017.

[24] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017.

[25] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5226–5234, 2018.

[26] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9489–9497, 2019.

[27] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223, 2020.

[28] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, et al. Learning progressive joint propagation for human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[29] Mao Wei, Liu Miaomiao, and Salzemann Mathieu. History repeats itself: Human motion prediction via motion attention. In *ECCV*, 2020.

[30] Andriy Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. In *International MICCAI Brainlesion Workshop*, pages 311–320. Springer, 2018.

[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[32] 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, 2013.

[33] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 5308–5317, 2016.

[34] Xiao Guo and Jongmoo Choi. Human motion prediction via learning local structure representations and temporal dependencies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2580–2587, 2019.

[35] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, Lee Giles, and Alexander G Ororbia. A neural temporal model for human motion prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12116–12125, 2019.

[36] Anthony Bourached and Parashkev Nachev. Unsupervised videographic analysis of rodent behaviour. *arXiv preprint arXiv:1910.11065*, 2019.

[37] Yuichiro Motegi, Yuma Hijioka, and Makoto Murakami. Human motion generative model using variational autoencoder. *International Journal of Modeling and Optimization*, 8(1), 2018.

[38] Nutan Chen, Justin Bayer, Sebastian Urban, and Patrick Van Der Smagt. Efficient movement representation by embedding dynamic movement primitives in deep autoencoders. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 434–440. IEEE, 2015.

[39] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[40] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

[41] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[42] Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu. Latent structured models for human pose estimation. In *2011 International Conference on Computer Vision*, pages 2220–2227. IEEE, 2011.

[43] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[44] F Sebastian Grassia. Practical parameterization of rotations using the exponential map. *Journal of graphics tools*, 3(3):29–48, 1998.

[45] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[46] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[47] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Appendix

The appendix consists of 6 parts. We provide a brief summary of each section below.

Appendix A: we provide results from our experimentation to determine the optimum way of defining separable distributions on the H3.6M, and the CMU datasets.

Appendix B: we provide a formulation of the problem of human motion prediction and detail the original discriminative models by [26], and [29] that we harden using a generative model.

Appendix C: we provide details of the datasets used and the experimental setup, including the policy used for hyperparameter search.

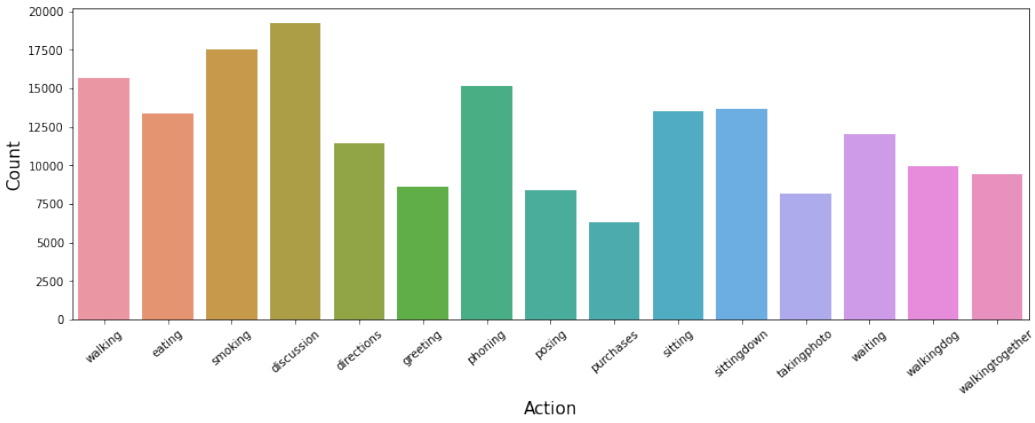
Appendix D: we provide some additional results for long term predictions on the H3.6M dataset as well as on the CMU dataset using 3D cartesian coordinates.

Appendix E: we inspect the generative model by examining its latent space and use it to consider the role that the gnerative model plays in learning as well as possible directions of future work.

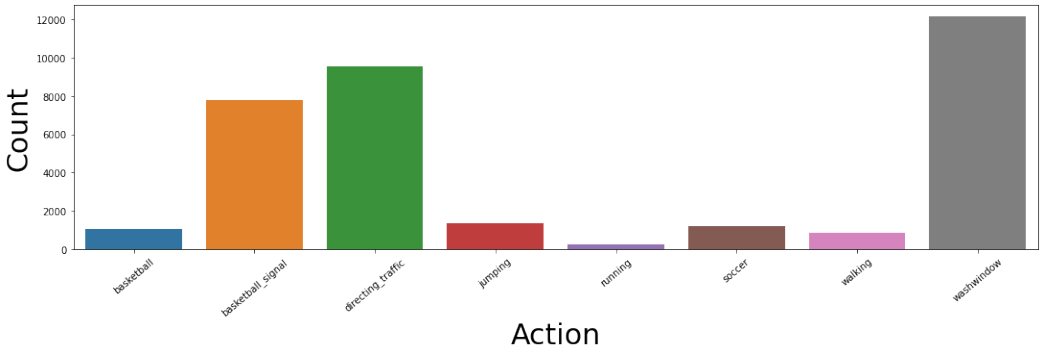
Appendix F: we provide larger diagrams of the architecture of the augmented GCN.

A Defining *Out-of-Distribution* (OoD).

Here we describe in more detail the empirical motivation for our definition of *Out-of-Distribution* (OoD) on the H3.6M and CMU datasets.



(a) Distribution of short-term training instances for actions in h3.6M.

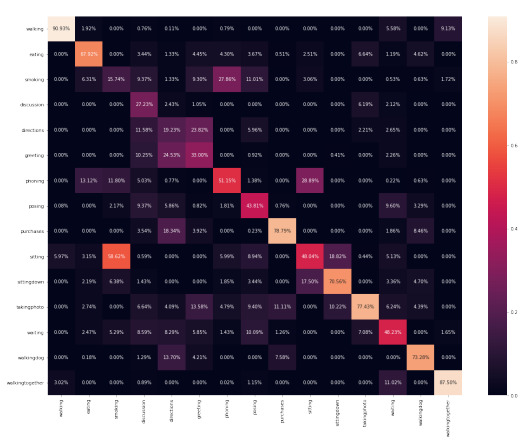


(b) Distribution of training instances for actions in CMU.

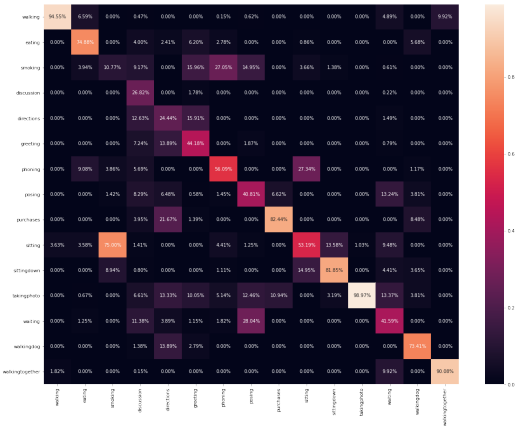
Figure 1

Figure 1 shows the distribution of actions for the h3.6M and CMU datasets. We want our ID data to be small in quantity, and narrow in domain. Since this dataset is labelled by action we are provided with a natural choice of distribution being one of these actions. Moreover, it is desirable that the action be quantifiably distinct from the other actions.

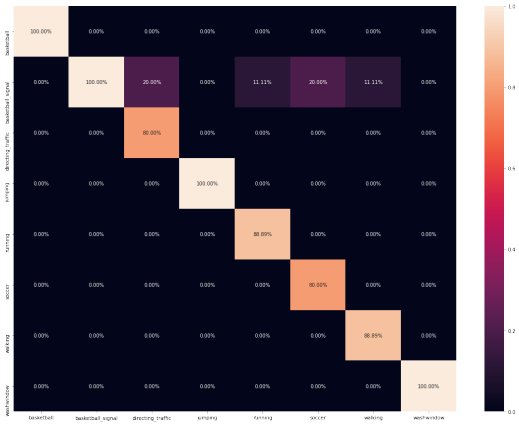
To determine which action supports these properties we train a simple classifier to determine which action is most easily distinguished from the others based on the DCT inputs: $DCT(\vec{x}_k) = DCT([x_{k,1}, \ldots, x_{k,N}, x_{k,N+1}, \ldots, x_{k,N+T}])$ where $x_{k,n} = x_{k,N}$ for $n \geq N$. We make no assumption on the architecture that would be optimum to determine the separation, and so use a simple fully connected model with 4 layers. Layer 1: *input dimensions* $\times$ 1024, layer 2: 1024×512, layer 3: 512×128, layer 4: 128×15 (or 128×8 for CMU). Where the final layer uses a softmax to predict the class label. Cross entropy is used as a loss function on these logits during training. We used ReLU activations with a dropout probability of 0.5.



(a) H3.6M dataset. $N = 10$, $T = 10$. Number of DCT coefficients = 20 (lossesless transformation).



(b) H3.6M dataset. $N = 50$, $T = 10$. Number of DCT coefficients = 20, where the 40 highest frequency DCT coefficients are culled.



(c) CMU dataset.$N = 10$, $T = 25$. Number of DCT coefficients = 35 (lossesless transformation).

Figure 2: Confusion matricies for a multi-class classifier for action labels. In each case we use the same input convention $\vec{x}_k = [x_{k,1}, \ldots, x_{k,N}, x_{k,N+1}, \ldots, x_{k,N+T}]$ where $x_{k,n} = x_{k,N}$ for $n \geq N$. Such that in each case input to the classifier is $48 \times 20 = 960$. The classifier has 4 fully connected layers. Layer 1: *input dimensions* $\times$ 1024, layer 2: 1024×512, layer 3: 512×128, layer 4: 128×15 (or 128×8 for CMU). Where the final layer uses a softmax to predict the class label. Cross entropy loss is used for training and ReLU activations with a dropout probability of 0.5. We used a batch size of 2048, and a learning rate of 0.00001.

We trained this model using the last 10 historic frames ($N = 10$, $T = 10$) with 20 DCT coefficients for both the H3.6M and CMU datasets, as well as ($N = 50$, $T = 10$) with 20 DCT coefficients additionally for H3.6M (here we select only the 20 lowest frequency DCT coefficients). We trained each model for 10 epochs with a batch size of 2048, and a learning rate of 0.00001. The confusion matrices for the H3.6M dataset are shown in figures 2a, and 2b respectively. Here, we use the same train set as outlined in appendix C. However, we report results on subject 11- which for motion prediction was used as the validation set. We did this because the number of instances are much greater than subject 5, and no hyperparameter tuning was necessary. For the CMU dataset we used the same train and test split as for all other experiments.

In both cases, for the H3.6M dataset, the classifier achieves the highest precision score (0.91, 0.95 respectively) for the action *walking* as well as a recall score of 0.83 and 0.81 respectively. Furthermore, in both cases *walking together* dominates the false negatives for *walking* (50%, and 44% in each case) as well as the false positives (33% in each case).

The general increase in the distinguishability that can be seen in figure 2b increases the demand to be able to robustly handle distributional shifts as the distribution of values that represent different actions only gets more pronounced as the time scale is increased. This is true with even the näive DCT transformation to capture longer time scales without increasing vector size.

As we can see from the confusion matrix in figure 2c, the actions in the CMU dataset are even more easily separable. In particular, our selected ID action in the paper, *Basketball*, can be identified with 100% precision and recall on the test set.

B Original discriminative models

Here we describe the current SOTA model proposed by [26] (GCN). We then describe the extension by [29] (att-GCN) which antecedes the GCN prediction model with motion attention.

B.1 Problem Formulation

We are given a motion sequence $\mathbf{X}_{1:N} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_N)$ consisting of N consecutive human poses, where $\mathbf{x}_i \in \mathbb{R}^K$, with K the number of parameters describing each pose. The goal is to predict the poses $\mathbf{X}_{N+1:N+T}$ for the subsequent T time steps.

B.2 DCT-based Temporal Encoding

The input is transformed using Discrete Cosine Transformations (DCT). In this way each resulting coefficient encodes information of the entire sequence at a particular temporal frequency. Furthermore, the option to remove high or low frequencies is provided. Given a joint, k, the position of k over N time steps is given by the trajectory vector: $\mathbf{x}_k = [x_{k,1}, \ldots, x_{k,N}]$ where we convert to a DCT vector of the form: $\mathbf{C}_k = [C_{k,1}, \ldots, C_{k,N}]$ where $C_{k,l}$ represents the lth DCT coefficient. For $\delta_{l1} \in \mathbb{R}^N = [1, 0, \cdots, 0]$, these coefficients may be computed as

$$C_{k,l} = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} x_{k,n} \frac{1}{\sqrt{1+\delta_{l1}}} \cos\left(\frac{\pi}{2N}(2n-1)(l-1)\right). \tag{2}$$

If no frequencies are cropped, the DCT is invertible via the Inverse Discrete Cosine Transform (IDCT):

$$x_{k,l} = \sqrt{\frac{2}{N}} \sum_{l=1}^{N} C_{k,l} \frac{1}{\sqrt{1+\delta_{l1}}} \cos\left(\frac{\pi}{2N}(2n-1)(l-1)\right). \tag{3}$$

Mao et al. use the DCT transform with a graph convolutional network architecture to predict the output sequence. This is achieved by having an equal length input-output sequence, where the input is the DCT transformation of $\mathbf{x_k} = [x_{k,1}, \ldots, x_{k,N}, x_{k,N+1}, \ldots, x_{k,N+T}]$, here $[x_{k,1}, \ldots, x_{k,N}]$ is the observed sequence and $[x_{k,N+1}, \ldots, x_{k,N+T}]$ are replicas of $x_{k,N}$ (ie $x_{k,n} = x_{k,N}$ for $n \geq N$). The target is now simply the ground truth $\mathbf{x_k}$.

B.3 Graph Convolutional Network

Suppose $\mathbf{C} \in \mathbb{R}^{K \times (N+T)}$ is defined on a graph with k nodes and $N+T$ dimensions, then we define a graph convolutional network to respect this structure. First we define a Graph Convolutional Layer (GCL) that, as input, takes the activation of the previous layer ($\mathbf{A^{[l-1]}}$), where l is the current layer.

$$GCL(\mathbf{A^{[l-1]}}) = \mathbf{S}\mathbf{A^{[l-1]}}\mathbf{W} + \mathbf{b} \tag{4}$$

where $\mathbf{A^{[0]}} = \mathbf{C} \in \mathbb{R}^{K \times (N+T)}$, and $\mathbf{S} \in \mathbb{R}^{K \times K}$ is a layer-specific learnable normalised graph laplacian that represents connections between joints, $\mathbf{W} \in \mathbb{R}^{n^{[l-1]} \times n^{[l]}}$ are the learnable inter-layer weightings and $\mathbf{b} \in \mathbb{R}^{n^{[l]}}$ are the learnable biases where $n^{[l]}$ are the number of hidden units in layer l.

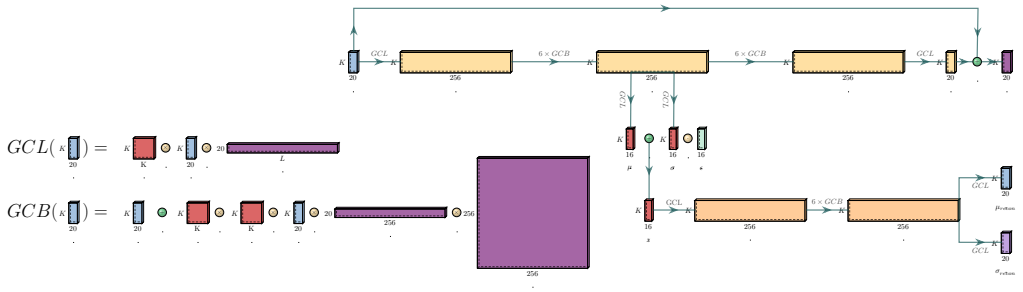


Figure 3: GCN network architecture with VGAE branch. Here $n_z = 16$ is the number of latent variables per joint.

B.4 Network Structure and Loss

The network consists of 12 Graph Convolutional Blocks (GCBs), each containing 2 GCLs with skip (or residual) connections, see figure 5. Additionally, there is one GCL at the beginning of the network, and one at the end. $n^{[l]} = 256$, for each layer, l. There is one final skip connection from the DCT inputs to the DCT outputs, which greatly reduces train time. The model has around 2.6M parameters. Hyperbolic tangent functions are used as the activation function. Batch normalisation is applied before each activation.

The outputs are converted back to their original coordinate system using the IDCT (equation 3) to be compared to the ground truth. The loss used for joint angles is the average l_1 distance between the ground-truth joint angles, and the predicted ones. Thus, the joint angle loss is:

$$\ell_a = \frac{1}{K(N+T)} \sum_{n=1}^{N+T} \sum_{k=1}^{K} |\hat{x}_{k,n} - x_{k,n}| \tag{5}$$

where $\hat{x}_{k,n}$ is the predicted k^{th} joint at timestep n and $x_{k,n}$ is the corresponding ground truth.

This is separately trained on 3D joint coordinate prediction making use of the Mean Per Joint Position Error (MPJPE), as proposed in [32] and used in [26, 29]. This is defined, for each training example, as

$$\ell_m = \frac{1}{J(N+T)} \sum_{n=1}^{N+T} \sum_{j=1}^{J} \|\hat{\mathbf{p}}_{j,n} - \mathbf{p}_{j,n}\|^2 \tag{6}$$

where $\hat{\mathbf{p}}_{j,n} \in \mathbb{R}^3$ denotes the predicted jth joint position in frame n. And $\mathbf{p}_{j,n}$ is the corresponding ground truth, while J is the number of joints in the skeleton.

B.5 Motion attention extension

[29] extend this model by summing multiple DCT transformations from different sections of the motion history with weightings learned via an attention mechanism. For this extension, the above model (the GCN) along with the anteceding motion attention is trained end-to-end. We refer to this as the attention-GCN.

C Datasets and Experimental Setup

Here we describe the datasets and experimental setup used in more detail.

Human3.6M (H3.6M) The H3.6M dataset [42, 32], so called as it contains a selection of 3.6 million 3D human poses and corresponding images, consists of seven actors each performing 15

actions, such as walking, eating, discussion, sitting, and talking on the phone. [24, 26, 27] all follow the same training and evaluation procedure: training their motion prediction model on 6 (5 for train and 1 for cross-validation) of the actors, for each action, and evaluate metrics on the final actor, subject 5. For easy comparison to these ID baselines, we maintain the same train; cross-validation; and test splits. However, we use the single, most well-defined action (see appendix A), *walking*, for train and cross-validation, and we report test error on all the remaining actions from subject 5. In this way we conduct all parameter selection based on ID performance.

CMU motion capture (CMU-mocap) The CMU dataset consists of 5 general classes of actions. Similarly to [25, 43, 26] we use 8 detailed actions from these classes: 'basketball', 'basketball signal', 'directing traffic' 'jumping, 'running', 'soccer', 'walking', and 'window washing'. We use two representations, a 64-dimensional vector that gives an exponential map representation [44] of the joint angle, and a 75-dimensional vector that gives the 3D Cartesian coordinates of 25 joints. We do not tune any hyperparameters on this dataset and use only a train and test set with the same split as is common in the literature [24, 26].

Model configuration We implemented the model in PyTorch [45] using the ADAM optimiser [46]. The learning rate was set to 0.0005 for all experiments where, unlike [26, 29], we did not decay the learning rate as it was hypothesised that the dynamic relationship between the discriminative and generative loss would make this redundant. The batch size was 16. For numerical stability, gradients were clipped to a maximum $\ell 2$-norm of 1 and $log(\hat{\sigma}^2)$ and values were clamped between -20 and 3.

Baseline comparison Both [26] (GCN), and [29] (attention-GCN) use this same Graph Convolutional Network (GCN) architecture with DCT inputs. In particular, [29] increase the amount of history accounted for by the GCN by adding a motion attention mechanism to weight the DCT coefficients from different sections of the history prior to being inputted to the GCN. We compare against both of these baselines on OoD actions. For attention-GCN we leave the attention mechanism preceding the GCN unchanged such that the generative branch of the model is reconstructing the weighted DCT inputs to the GCN, and the whole network is end-to-end differentiable.

Hyperparameter search Since a new term has been introduced to the loss function, it was necessary to determine a sensible weighting between the discriminative and generative models. In [30], this weighting was arbitrarily set to 0.1. It is natural that the optimum value here will relate to the other regularisation parameters in the model. Thus, we conducted random hyperparameter search for p_{drop} and λ in the ranges $p_{drop} = [0, 0.5]$ on a linear scale, and $\lambda = [10, 0.00001]$ on a logarithmic scale. For fair comparison we also conducted hyperparameter search on GCN, for values of the dropout probability (p_{drop}) between 0.1 and 0.9. For each model, 25 experiments were run and the optimum values were selected on the lowest ID validation error. The hyperparameter search was conducted only for the GCN model on short-term predictions for the H3.6M dataset and used for all future experiments hence demonstrating generalisability of the architecture.

D Additional results

Here we report some additional results for long term predictions on the H3.6M dataset and on the CMU dataset using 3D cartesian coordinates.

Table 4 show that we get consistent results for long term prediction. Again, emphasising that we accumulate greater benefit for predictions further into the future.

	Walking		Eating		Smoking		Discussion		Average	
milliseconds	560	1000	560	1000	560	1000	560	1000	560	1000
GCN	0.80	0.80	**0.89**	1.20	1.26	1.85	1.45	**1.88**	1.10	1.43
ours	**0.66**	**0.72**	0.90	**1.19**	**1.17**	**1.78**	**1.44**	1.90	**1.04**	**1.40**

Table 4: Long-term prediction of Eucildean distance between predicted and ground truth joint angles on H3.6M.

While table 6 we receive similar benefit when we learn from 3D joint coordinates as indeed is concluded to be more robust and scalable in [26]. For 3D joint coordinate representation we use the MPJPE as used for training (equation 6).

	Basketball (ID)					Basketball Signal (OoD)					Directing Traffic (OoD)				
milliseconds	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
GCN	**0.40**	0.67	**1.11**	**1.25**	**1.63**	**0.27**	**0.55**	**1.14**	**1.42**	2.18	0.31	0.62	1.05	1.24	2.49
ours	**0.40**	**0.66**	1.12	1.29	1.76	0.28	0.57	1.15	1.43	**2.07**	**0.28**	**0.56**	**0.96**	**1.10**	**2.33**
	Jumping (OoD)					Running (OoD)					Soccer (OoD)				
milliseconds	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
GCN	0.42	0.73	**1.72**	**1.98**	**2.66**	**0.46**	0.84	1.50	1.72	**1.57**	0.29	0.54	1.15	1.41	2.14
ours	**0.38**	**0.72**	1.74	2.03	2.70	**0.46**	**0.81**	**1.36**	**1.53**	2.09	**0.28**	**0.53**	**1.07**	**1.27**	**1.99**
	Walking (OoD)					Washing window (OoD)					Average (of 7 for OoD)				
milliseconds	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
GCN	0.40	0.61	0.97	1.18	1.85	0.36	0.65	1.23	**1.51**	2.31	0.36	0.65	1.41	1.49	2.17
ours	**0.38**	**0.54**	**0.82**	**0.99**	**1.27**	**0.35**	**0.63**	**1.20**	**1.51**	**2.26**	**0.34**	**0.62**	**1.35**	**1.41**	**2.10**

Table 5: Eucildean distance between predicted and ground truth joint angles on CMU.

	Basketball (ID)					Basketball Signal (OoD)					Directing Traffic (OoD)				
milliseconds	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
GCN	**15.7**	**28.9**	**54.1**	**65.4**	108.4	14.4	30.4	63.5	78.7	114.8	18.5	37.4	**75.6**	**93.6**	210.7
ours	16.0	30.0	54.5	65.5	**98.1**	**12.8**	**26.0**	**53.7**	**67.6**	**103.2**	**18.3**	**37.2**	75.7	93.8	**199.6**
	Jumping (OoD)					Running (OoD)					Soccer (OoD)				
milliseconds	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
GCN	**24.6**	**51.2**	111.4	139.6	219.7	32.3	54.8	85.9	99.3	**99.9**	22.6	46.6	92.8	114.3	**192.5**
ours	25.0	52.0	**110.3**	**136.8**	**200.2**	**29.8**	**50.2**	**83.5**	**98.7**	107.3	**21.1**	**44.2**	**90.4**	**112.1**	202.0
	Walking (OoD)					Washing window (OoD)					Average of 7 for (OoD)				
milliseconds	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
GCN	10.8	20.7	42.9	53.4	86.5	**17.1**	**36.4**	**77.6**	**96.0**	**151.6**	**20.0**	43.8	86.3	105.8	169.2
ours	**10.5**	**18.9**	**39.2**	**48.6**	**72.2**	17.6	37.3	82.0	103.4	167.5	21.6	**42.3**	**84.2**	**103.8**	**164.3**

Table 6: Mean Joint Per Position Error (MPJPE) between predicted and ground truth 3D Cartesian coordinates of joints on CMU.

	Walking (ID)				Eating (OoD)				Smoking (OoD)				Discussion (OoD)			
milliseconds	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000
att-GCN	**55.4**	**60.5**	**65.2**	**68.7**	87.6	103.6	113.2	120.3	81.7	93.7	102.9	108.7	**114.6**	130.0	**133.5**	**136.3**
ours	58.7	60.6	65.5	69.1	**81.7**	**94.4**	**102.7**	**109.3**	**80.6**	**89.9**	**99.2**	**104.1**	115.4	**129.0**	134.5	139.4
	Directions (OoD)				Greeting (OoD)				Phoning (OoD)				Posing (OoD)			
milliseconds	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000
att-GCN	**107.0**	123.6	132.7	138.4	**127.4**	142.0	153.4	158.6	98.7	117.3	129.9	138.4	**151.0**	**176.0**	**189.4**	**199.6**
ours	107.1	**120.6**	**129.2**	**136.6**	128.0	**140.3**	**150.8**	**155.7**	**95.8**	**111.0**	**122.7**	**131.4**	158.7	181.3	194.4	203.4
	Purchases (OoD)				Sitting (OoD)				Sitting Down (OoD)				Taking Photo (OoD)			
milliseconds	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000
att-GCN	**126.6**	144.0	**154.3**	**162.1**	**118.3**	141.1	154.6	164.0	**136.8**	162.3	177.7	189.9	**113.7**	137.2	149.7	159.9
ours	128.0	**143.2**	154.7	164.3	118.4	**137.7**	**149.7**	**157.5**	**136.8**	**157.6**	**170.8**	**180.4**	116.3	**134.5**	**145.6**	**155.4**
	Waiting (OoD)				Walking Dog (OoD)				Walking Together (OoD)				Average (of 14 for OoD)			
milliseconds	560	720	880	1000	560	720	880	1000	560	720	880	1000	560	720	880	1000
att-GCN	**109.9**	125.1	135.3	141.2	**131.3**	**146.9**	**161.1**	**171.4**	**64.5**	**71.1**	**76.8**	**80.8**	**112.1**	129.6	140.3	147.8
ours	110.4	**124.5**	**133.9**	**140.3**	138.3	151.2	165.0	175.5	67.7	71.9	77.1	**80.8**	113.1	**127.7**	**137.9**	**145.3**

Table 7: Long-term prediction of 3D joint positions on H3.6M. Here, ours is also trained with the att-GCN model.

E Latent space of the VGAE

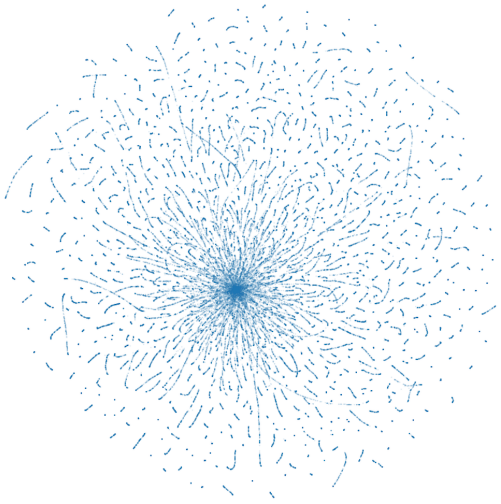
One of the advantages of having a generative model involved is that we have a latent variable which represents a distribution over deterministic encodings of the data. We considered the question of whether or not the VGAE was learning anything interpretable with its latent variable as was the case in [41].

The purpose of this investigation was two-fold. First to determine if the generative model was learning a comprehensive internal state, or just a non-linear average state as is common to see in the training of VAE like architectures. The result of this should suggest a key direction of future work. Second, an interpretable latent space may be of paramount usefulness for future applications of human motion prediction. Namely, if dimensionality reduction of the latent space to an inspectable number of dimensions yields actions, or behaviour that are close together if kinematically or teleolgically similar,

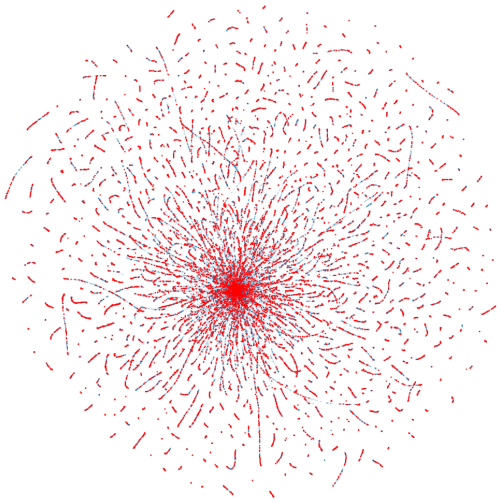
as in [36], then human experts may find unbounded potential application for a interpretation that is both quantifiable and qualitatively comparable to all other classes within their domain of interest. For example, a medical doctor may consider a patient to have unusual symptoms for condition, say, A. It may be useful to know that the patient's deviation from a classical case of A, is in the direction of condition, say, B.

We trained the augmented GCN model discussed in the main text with all actions, for both datasets. We use Uniform Manifold Approximation and Projection (UMAP) [47] to project the latent space of the trained GCN models onto 2 dimensions for all samples in the dataset for each dataset independently. From figure 4 we can see that for both models the 2D project relatively closely resembles a spherical gaussian. Further, we can see from figure 4b that the action *walking* does not occupy a discernible domain of the latent space. This result is further verified by using the same classifier as used in appendix A, which achieved no better than chance when using the latent variables as input rather than the raw data input.

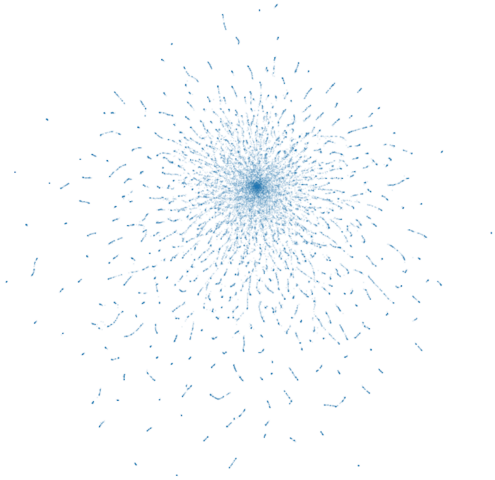
This result implies that the benefit observed in the main text is by using the generative model is significant even if the generative model has poor performance itself. In this case we can be sure that the reconstructions are at least not good enough to distinguish between actions. It is hence natural for future work to investigate if the improvement on OoD performance is greater if trained in such a way as to ensure that the generative model performs well. There are multiple avenues through which such an objective might be achieve. Pre-training the generative model being one of the salient candidates.



(a) H3.6M. All actions, opacity=0.1.



(b) H3.6M. All actions in blue: opacity=0.1. Walking in red: opacity=1.



(c) CMU. All actions in blue: opacity=0.1.

Figure 4: Latent embedding of the trained model on both the H3.6m and the CMU datasets independently projected in 2D using UMAP from 384 dimensions for H3.6M, and 512 dimensions for CMU using default hyperparameters for UMAP.

F Architecture Diagrams

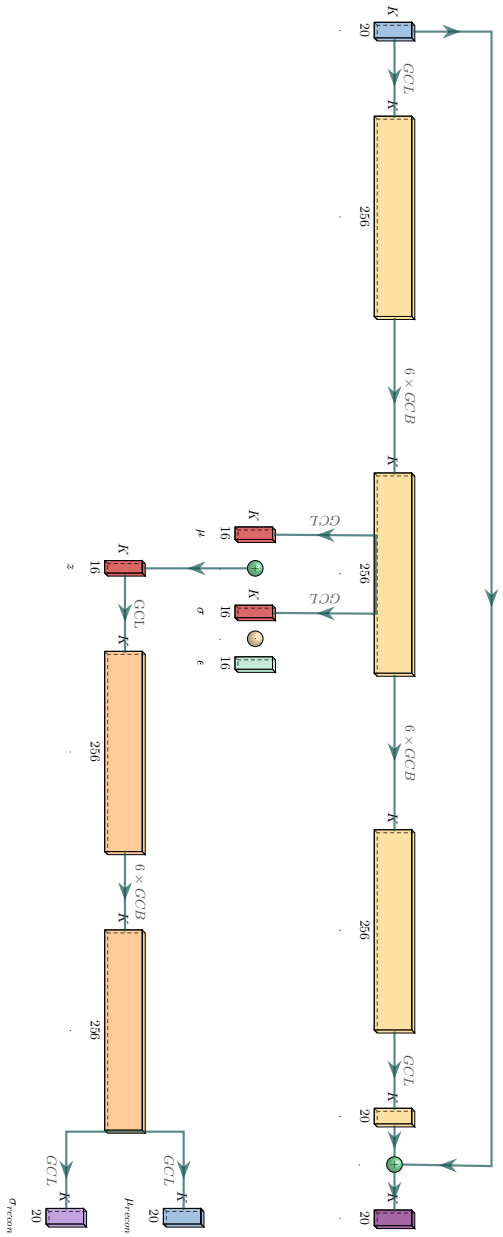


Figure 5: Network architecture with discriminative and VGAE branch.

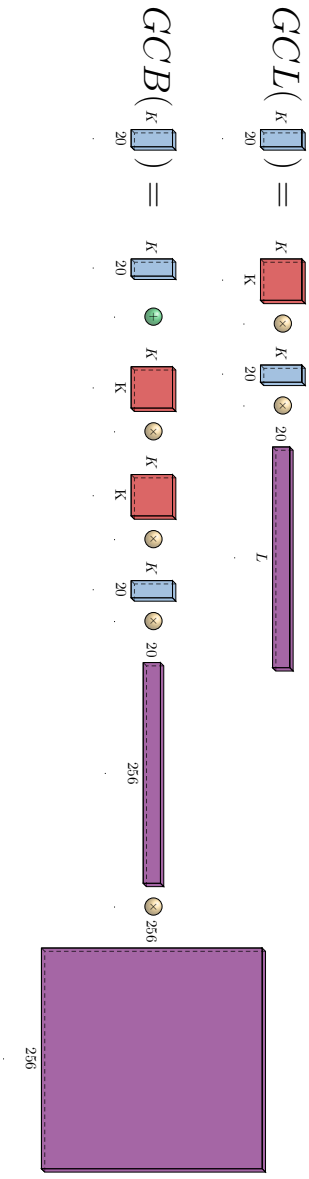


Figure 6: Graph Convolutional layer (GCL) and a residual graph convolutional block (GCB).