USING A SINGLE INPUT TO FORECAST HUMAN ACTION KEYSTATES IN EVERYDAY PICK AND PLACE ACTIONS

Haziq Razali and Yiannis Demiris[1]

Department of Electrical and Electronic Engineering, Personal Robotics Laboratory
Imperial College London

ABSTRACT

We define action keystates as the start or end of an action that contains information such as the human pose and time. Existing methods that forecast the human pose use recurrent networks that input and output a sequence of poses. In this paper, we present a method tailored for everyday pick and place actions where the object of interest is known. In contrast to existing methods, ours uses an input from a single timestep to directly forecast (i) the key pose the instant the pick or place action is performed and (ii) the time it takes to get to the predicted key pose. Experimental results show that our method outperforms the state-of-the-art for key pose forecasting and is comparable for time forecasting while running at least an order of magnitude faster. Further ablative studies reveal the significance of the object of interest in enabling the total number of parameters across all existing methods to be reduced by at least 90% without any degradation in performance. [a]

Index Terms— Human Pose Forecasting

1. INTRODUCTION

Our ability to visually anticipate the motions of others is an essential aspect of our social intelligence when interacting with the world around us. Providing machines with access to this knowledge will enable a wide range of applications from human robot interaction [1] to surveillance and health care [2]. For instance, autonomous vehicles or delivery robots that navigate crowded scenes require the ability to anticipate human trajectories [3] whereas collaborative robots that work closely with humans need to anticipate human actions in order to provide a timely response [4].

Due to the temporal nature inherent in human motion, existing methods have used recurrent models that input and output a sequence of the human pose. The drawbacks that come with such models however, is that the errors tend to exacerbate as predictions are made further into the future and that the runtime increases with longer prediction lengths. We present in this paper a method that addresses said drawbacks when the objects of interest are known, e.g., when a person navigates around an environment to perform a pick or place action. Moreover, we show that knowledge of the object of interest allow ours and existing methods to be built with far fewer parameters without any degradation in performance. The result is a method that can be deployed in resource-limited systems, a line of research that is actively pursued by the scientific community [5, 6, 7]. To the best of our knowledge, our method is the first that uses an input from a single timestep to directly forecast the key pose the instant the pick or place action is performed, and the time taken to get to the key pose (Figure 1).

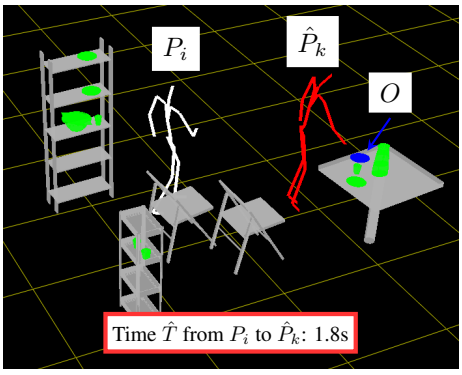


Fig. 1: Given the input pose P_i and the coordinates of the object of interest O, our method directly predicts the key pose $\hat{P}_k$ the instant the pick or place action is performed and the time $\hat{T}$ it takes to get from P_i to $\hat{P}_k$.

2. RELATED WORK

Existing works on pose forecasting learn the correlation between the input and output pose sequence. They output either the joint positions or joint rotations that are then converted to positions via forward kinematics. For instance, Martinez et al. [8] trained a GRU to output the joint velocities. Li et al. [9] replaced the GRU with 2D convolutions that receive the near and distant past as input. Corona et al. [10] modelled the relationship between joint positions and all objects using a graph recurrent network. Fragkiadaki et al. [11] and Pavllo et al.

[1]{hb620, y.demiris}@imperial.ac.uk YD is supported by a Royal Academy of Engineering Chair In Emerging Technologies.

[a]Code available at www.imperial.ac.uk/personal-robotics/software

[12] predict joint rotations that are parameterized using euler angles and quaternions respectively. There are also methods that incorporate stochasticity using variational autoencoders [13, 14]. A similarity shared by the above-mentioned works however, is that (i) they do not work with a single input, (ii) do not directly forecast the key pose and (iii) do not consider the object of interest in their input, all of which are our main contributions in this paper.

3. METHOD

Our method is briefly summarized in Figure 2. Its goal is to learn $p(P_k, T|P_i, O)$ where $O \in \mathbb{R}^3$ is the coordinates of the object of interest, $P_i, P_k \in \mathbb{R}^{N\times 3}$ the input and key pose respectively with N joints, and $T \in \mathbb{R}$ the time taken to get from P_i to P_k. It first computes the trajectory from P_i to O, then forecasts P_k, before predicting the time it takes to get from P_i to P_k. We now describe each component in detail.

Trajectory Forecasting: Our method first computes the shortest path from P_i to O. We discretize the ground plane into a grid and set the score of each node based on its $L1$ distance to O. Nodes that lie within obstacles are given very large scores. The method then traverses the nodes in a greedy fashion using Dijkstra [15] to output a path $X \in \mathbb{R}^{M\times 3}$ where M indicates the total number of points traversed from P_i to reach O. We chose Dijkstra over a learning based approach using neural networks [16] to keep the algorithm runtime as low as possible. Furthermore, our aim is not to learn the trajectory taken by the person but rather, to get a rough estimate of the person's position as he approaches the object, which we found the shortest path method to be sufficiently performant at. As such, the grid size can also be set very small to keep the runtime negligible.

Pose Forecasting: Given the trajectory, we compute the normalized approach vector defined as:

$$V_a = O - \sum_{p=1}^{P} \frac{X[-p]}{P} \tag{1}$$

$$\bar{V}_a = V_a / ||V_a||_2 \tag{2}$$

where $-p$ denotes the last p'th element. $\bar{V}_a$ tells us the direction of the object relative to the trajectory end point. Its x and y component tells us e.g., if the object is at 2 o'clock from the end point whereas the z component how high the object is above ground. It is then fed together with the input pose to predict the key pose as follows:

$$\hat{P}_k = \phi_1([P_i, \bar{V}_a], z) \tag{3}$$

where ϕ_1 is a Variational Autoencoder (VAE) [17], the brackets indicate concatenation, and z a random variable. Our reason for the VAE arises from the observation that the same object coordinate does not necessarily pair to the same key pose, primarily due to how different individuals position themselves differently when picking or placing an item. Our discretization of the navigable space further makes it even more likely that the model observes the same $\bar{V}_a$ for different values of P_k. The result is an uncertain future which can be tackled by a VAE by learning to map the approach vector to a distribution over the observed key poses. Finally, we point out that our method works equally well on both pick and place actions since O is simply a coordinate.

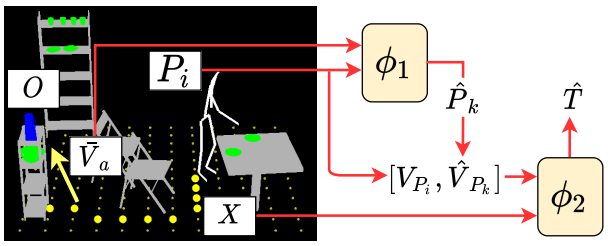


Fig. 2: Overview of our method. We discretize the navigable space then get the shortest path X shown by the bold yellow dots, before predicting P_k then T.

Time Forecasting: We also use a VAE to forecast the time T it will take to get from P_i to P_k. The reason is simply that the time a person takes to get from one pose and location to the next is also non-deterministic due to factors such as walking speed and how the person navigates around obstacles. This problem is further complicated by the fact that we use only a single timestep as input to the model. It is simply not possible to accurately forecast the time it takes without knowing the velocities of the human. These uncertainties thus necessitate the use of another VAE. We denote V_{P_i} and $\hat{V}_{P_k}$ to represent the vectors that are perpendicular to input and predicted key poses respectively. We forecast the time as follows:

$$\hat{T} = \phi_2([V_{P_i}, \hat{V}_{P_k}, \sum \triangle X], z] \tag{4}$$

where ϕ_2 is a VAE, and $\sum \triangle X$ the total distance travelled. Intuitively, $[V_{P_i}, \hat{V}_{P_k}]$ informs the model where the person faced at the start and end. We found this information helpful since there is time spent e.g., when the person does an about-face to get to the object located in the opposite direction. The trajectory returned by Dijkstra provides no such information. Finally, we train the model by minimizing:

$$L = \lambda_1 L_{\phi_1} + \lambda_2 L_{\phi_2} \tag{5}$$

where L_ϕ denotes the standard VAE loss [17] and the lambdas are both set to 1.

4. EXPERIMENTS

Dataset: We evaluate our method on the MoGaze motion-capture dataset [18] that contains 3 hours of recordings at

120 fps with 7 participants, each performing several pick and place actions on everyday household items such as cups and plates. The participants move in an area of approximately 3 by 4 meters and make use of only their right hand to perform the action. Each frame is annotated with the object the participant is currently holding on to. The dataset also contains chairs as obstacles that were placed at different locations throughout the recording. We train and test on participants 4-7 and 1-3, with 600k and 200k samples respectively.

Setup: We compare our approach to existing encoder-decoder recurrent networks for key pose forecasting: the variational architecture by Chung et al. [13], the convolutional model by Li et al. [9], the diversified latent flows by Yuan et al. [14], and the context aware architecture by Corona et al. [10]. We follow their training setup to forecast the next 2 seconds (20 frames) given a 1 second (10 frames) input. For a fair comparison, we also concatenate the coordinates of the object of interest to the input pose at every timestep. We modify their models to output T at the final encoding timestep. At test time, the decoder forecasts up till the predicted T. Methods that incorporate stochasticity [13, 14], including ours have the outputs averaged over 12 samples for evaluation purposes. Finally, we also scale all models such that their number of parameters are on par with the state-of-the-art [10] at approximately 3 million.

All models are trained until convergence using the ADAM optimizer [19] with a learning rate of 1e-3, batch size of 32 and default hyperparameters. We preprocess by subtracting the coordinates of every joint and object in the scene by the pelvis of the input pose. We do not augment the data in any way. All experiments are conducted using PyTorch on an Ubuntu machine with an Intel i7-8700k CPU and a NVIDIA RTX-2080 GPU.

4.1. Quantitative Results

Pose Forecasting: We first present quantitative results for key pose forecasting in Table 1 where we compute the L2 errors for each of the human joint in meters. Our method improves the state-of-the-art [10] by approximately 5% from 6.738m to 6.391m. The numbers also show the redundancy in using a sequence of inputs since the characteristic of our result mimics the recurrent models in that we also achieve lower errors for the right half of the body, the side of the body the participants used when performing a pick or place action. The joints on the side that was not used for the action are less constrained in their positioning, hence the higher error scores. Specifically, it can be observed that our error on the right wrist is significantly lower than the state-of-the-art.

Time Forecasting: For the task of time forecasting, we did not improve the state-of-the-art which we believe is due to our use of Dijkstra and a single input. We noted that the shortest path returned by Dijkstra may be very different than the ground truth trajectories. There is also no information about the person's velocities when using an input from a single timestep.

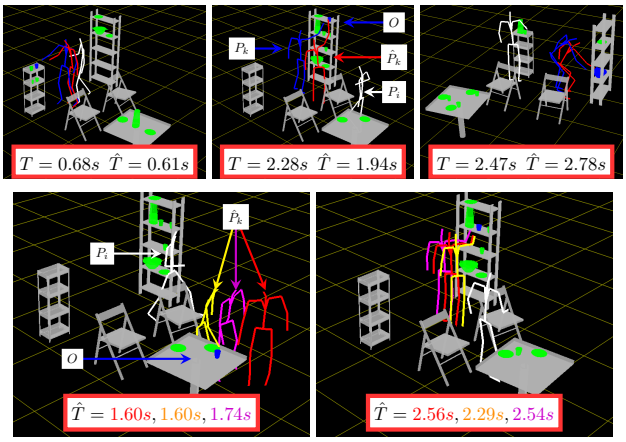


Fig. 3: **Top**: Comparing our predictions to the ground truths. **Bottom**: Multiple samples of our predictions given a single input. Our method can directly forecast key poses that are realistic with a corresponding time prediction that is accurate.

4.2. Qualitative Results

We show some qualitative results in Figure 3. The top row shows the ground truth with our predictions. Our method is able to forecast key poses that are realistic with a corresponding time prediction that is reasonably accurate. Our model predicts a person with its right arm stretched out in order to reach the object that has been placed at the top level of the shelf, or a squat to reach the object at the bottom level of the shelf. In all these examples, the distance between the connected joints are also of appropriate length. The results thus reveal to some extent, the range of valid poses the model has learnt. The bottom row shows multiple predictions given the same input and thus, the same approach vector. They show that our VAE is able to map the same vector to a distribution over the key poses.

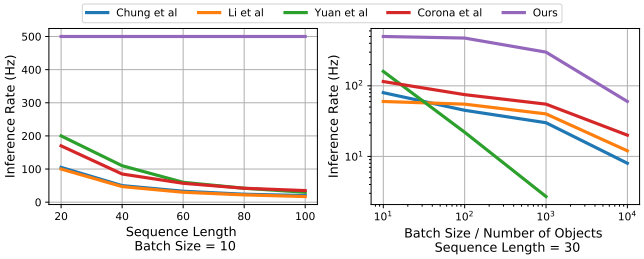
4.3. Ablation Studies

Inference Rate: We next study the rate of inference of the various methods with 2 separate experiments. (i) We fix the batch size at 10 and set the sequence length to [20,40,60,80,100] and (ii) we fix the sequence length at 30 and set the batch size to [1e1,1e2,1e3,1e4]. Methods that incorporate stochasticity now output only a single sample. The results are presented in Figure 4a and 4b respectively.

As indicated in Figure 4a, our method runs 5x faster than the state-of-the-art at 500 Hz when the sequence length is set to 20 (2 seconds). Furthermore, all four models [13, 9, 14, 10]

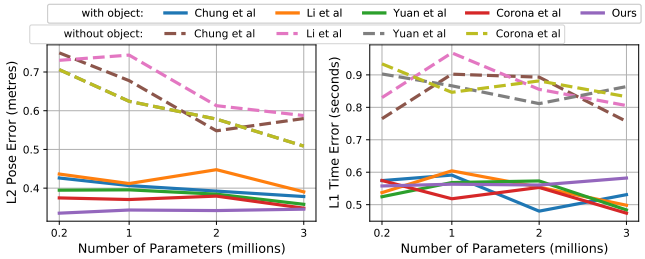
Method	Spinal Column					Left-Half							Right-Half							Total	Time
	head	neck	torso	pelvis	base	shoulder	elbow	wrist	hip	knee	ankle	toe	shoulder	elbow	wrist	hip	knee	ankle	toe		
Ours	**0.291**	**0.316**	**0.329**	**0.336**	**0.339**	**0.359**	0.439	0.438	**0.363**	0.365	0.423	0.394	**0.293**	**0.215**	**0.133**	**0.326**	**0.313**	**0.375**	**0.344**	**6.391**	0.582
Chung et al. [13]	0.359	0.365	0.366	0.368	0.368	0.397	0.446	0.442	0.388	0.379	0.425	0.401	0.356	0.360	0.339	0.355	0.338	0.378	0.361	7.191	0.530
Li et al. [9]	0.339	0.371	0.38	0.345	0.348	0.407	0.478	0.475	0.381	0.375	0.443	0.391	0.336	0.423	0.453	0.350	0.349	0.418	0.393	7.455	0.498
Yuan et al. [14]	0.338	0.344	0.35	0.351	0.346	0.375	0.433	0.436	0.366	0.366	**0.414**	0.396	0.336	0.324	0.301	0.330	0.316	**0.375**	0.345	6.842	0.483
Corona et al. [10]	0.313	0.339	0.349	0.339	0.347	0.361	**0.426**	**0.423**	0.368	**0.363**	0.418	**0.391**	0.323	0.302	0.290	0.334	0.323	**0.375**	0.349	6.738	**0.473**

Table 1: L2 key pose errors in metres for each joint and L1 time prediction errors in seconds. Our single input method improves the total L2 error by approximately 5% from 6.738 to 6.391 meters and achieves much lower scores for the joint (right wrist) that was used to perform the pick or place action.



(a) (b)

Fig. 4: Our method is invariant to the sequence length and continues to run faster than real time at approximately 60 Hz when handling an unrealistic scenario with 1e4 objects.



(a) (b)

Fig. 5: The models remain performant even after a 90% reduction in the number of parameters, signalling the strong cue the object coordinates provide for the joint task of key pose and time forecasting.

are sequence based with a runtime that increases with longer sequence lengths. Their rate of inference thus drops drastically in setups that require long-term forecasting. Such models are also faced with the dilemma of choosing between a larger step size for faster processing of longer sequences but with less accurate pose forecasting, or a smaller step size for more accurate forecasting but slower processing of longer sequences. Our method in contrast, requires no tuning by the user and runs at the same rate regardless of the start and end position of the person.

A valid concern with our method is that it requires the object of interest which may be numerous depending on the environment and task. Figure 4b shows the runtime of our method with increasing batch size. Note that the batch size can be interpreted as the number of objects of interest for 1 person. The results indicate that our method continues to run faster than real time at approximately 60 Hz even when handling an unrealistic scenario with 1e4 objects (or 10 people and 1e3 objects). Our method is thus not bottlenecked by the total number of objects in the scene. The results highlight the large gains we get by directly forecasting both the key pose and time, and are significant as it allows a system to rapidly perform inference for each person in a crowded scene or to predict multiple poses of an individual at various locations.

Importance of the object of interest: We highlight the importance of having the coordinates of the object of interest by retraining all models with different numbers of parameters: [300k, 1m, 2m, 3m]. The results are summarized in Figure 5 where the errors have been averaged over every joint. The models remain performant even after a 90% reduction in the number of parameters, signalling the strong cue the object coordinates provide for the joint task of key pose and time forecasting. Models with object information also perform significantly better than their counterparts without it. These findings enable motion forecasting models to be built with far fewer parameters when the object coordinate is made available e.g. via an object detector [20], making it an attractive add-on to any system with very little overhead. It also supports the use of our model as an alternative over the state-of-the-art recurrent models when the intermediate poses are not required as the only concern then is the mapping from the input pose to the key pose. All-in-all, our findings are useful for systems that require an estimator with a very small memory footprint running in the background.

5. CONCLUSION

We have presented a method that forecasts the action keystates in everyday pick and place actions. The main takeaways are that the object of interest (i) allows the method to directly forecast the key pose and time using an input from a single timestep, achieving lower errors with a constant runtime that is at least an order of magnitude faster than existing recurrent type methods and (ii) lets the total number of parameters across all existing methods be significantly reduced without any degradation in performance.

6. REFERENCES

[1] 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)*. IEEE, 2018, pp. 562–567.

[2] Jianxiong Zhou, Zhongyu Jiang, Jang-Hee Yoo, and Jenq-Neng Hwang, "Hierarchical pose classification for infant action analysis and mental development assessment," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 1340–1344.

[3] 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*, 2016, pp. 961–971.

[4] Ninghang Hu, Aaron Bestick, Gwenn Englebienne, Ruzena Bajscy, and Ben Kröse, "Human intent forecasting using intrinsic kinematic constraints," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 787–793.

[5] Zhe Zhang, Jie Tang, and Gangshan Wu, "Lightweight human pose estimation under resource-limited scenes," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2170–2174.

[6] Roy Miles and Krystian Mikolajczyk, "Compressing local descriptor models for mobile applications," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 1895–1899.

[7] Zhong-Han Niu, Xi-Peng Lin, An-Ni Yu, Yang-Hao Zhou, and Yu-Bin Yang, "Lightweight and accurate single image super-resolution with channel segregation network," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 1630–1634.

[8] 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*, 2017, pp. 2891–2900.

[9] 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*, 2018, pp. 5226–5234.

[10] Enric Corona, Albert Pumarola, Guillem Alenya, and Francesc Moreno-Noguer, "Context-aware human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6992–7001.

[11] 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*, 2015, pp. 4346–4354.

[12] Dario Pavllo, David Grangier, and Michael Auli, "Quaternet: A quaternion-based recurrent model for human motion," *arXiv preprint arXiv:1805.06485*, 2018.

[13] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio, "A recurrent latent variable model for sequential data," *Neural Information Processing Systems*, 2015.

[14] Ye Yuan and Kris Kitani, "Dlow: Diversifying latent flows for diverse human motion prediction," in *European Conference on Computer Vision*. Springer, 2020, pp. 346–364.

[15] Edsger W Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[16] Tomas Kulvicius, Sebastian Herzog, Timo Lüddecke, Minija Tamosiunaite, and Florentin Wörgötter, "One-shot path planning for multi-agent systems using fully convolutional neural network," *arXiv preprint arXiv:2004.00568*, 2020.

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

[18] Philipp Kratzer, Simon Bihlmaier, Niteesh Balachandra Midlagajni, Rohit Prakash, Marc Toussaint, and Jim Mainprice, "Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 367–373, 2020.

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

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.