# RNN-based Motion Prediction in Competitive Fencing Considering Interaction between Players

Yutaro Honda[1]
honda@hc.ic.i.u-tokyo.ac.jp
Rei Kawakami[2]
reikawa@c.titech.ac.jp
Takeshi Naemura[1]
naemura@hc.ic.i.u-tokyo.ac.jp

[1] The University of Tokyo, Japan
[2] Tokyo Institute of Technology, Japan

**Abstract**

The ability to accurately predict the motion of fencing athletes will help to improve the competition techniques of the players and the viewing experience of the audience. Most human-motion prediction methods only consider a single person, but in fencing, the movement of the opponent greatly affects the future movements of the player. In this paper, we propose a motion prediction model that takes into account the interaction between the two players in the game by connecting the recurrent neural networks to each other. In experiments, our model improved the accuracy of predicting movements in response to the opposing player, such as retreating to avoid the opponent's thrusts.

## 1 Introduction

Fencing is a sport where two athletes stand opposite each other and poke each other's bodies with their swords in one hand to decide the winner. A successful attack that touches the opponent's body with the sword results in a point, and a player with the specified number of points wins the game. Sports analytics is one of the areas in which computer vision is employed for purposes such as player tracking, video captioning, and action recognition [16, 20, 21, 26], while analyses conducted with cameras in stadiums have been used in popular team sports. Applications of this technology to one-on-one fighting sports are an interesting and valuable area to be explored. This paper analyzes fencing, one of the most famous one-on-one sports, by predicting two players motion simultaneously. Highly accurate motion prediction models can help to improve competition techniques and the viewing experience by visualizing the causes of the movement and by detecting technically advanced movements using deviations from predictions.

Most of the existing motion prediction methods consider only a single person and do not take into account interactions with others [8, 14, 22, 23]. In fencing, however, the players influence each other and change their movements, even within a second. Therefore, the study reported here incorporates the interaction between players for accurate motion prediction. To achieve this, we assign a prediction model formed from a recurrent neural network (RNN)

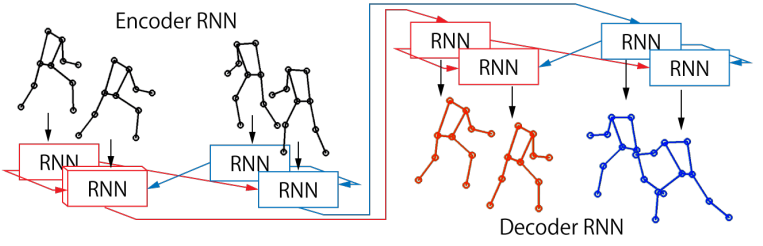


Figure 1: Overview of our prediction model. Each player's RNN observes the past few poses while receiving the opponent's information simultaneously. The connections are also used for forecasting the future poses.

to each player and mutually connect these individual models. The reason for applying a predictive model to each player in a match is that each player moves differently, which needs to be predicted with some degree of individual focus. The encoder-decoder models, which are suitable for motion prediction, are selected for the baseline prediction networks, and the models are connected mutually, as illustrated in Fig. 1.

To demonstrate the effectiveness of this method, we applied it to several baselines that predict the motion of a single person [8, 14, 22] and confirmed improvements in accuracy for all of them. The performance gain was the highest at the most accurate model. In experiments, a dataset of 2D joint positions of the two athletes was created based on match videos provided by a national fencing federation [10]. The joint positions of the two athletes in a 0.5-second period were input and those in the following 1-second period were predicted. These short time periods were decided because fencing players move fast; making a thrust and returning to the original posture happens in about one second. The results quantitatively show that the proposed method improves the prediction accuracy, and particularly, the contribution is large in the latter 0.5 second of the predicted 1-second period. Further analysis shows that the proposed methods start outperforming baselines at about 200ms to 400ms in 1 second prediction, which is close to fencing experts' reaction time. Thus, the prediction by the proposed methods should reflect the players' cognitive ability by considering interaction between them. A qualitative evaluation shows that the proposed method could predict movements that were difficult to predict without considering the interaction between players.

The contributions of this paper are summarized as follows: First, we propose a framework that embeds the mutual effect of players in motion prediction and show its effectiveness in fencing, where player interaction is immediately reflected in motion. Second, we show that our framework improves existing motion prediction methods consistently, and the accurate ones most effectively. Third, we show the performance improvement had by considering interaction is higher in the distant future, the latter half of the predicted 1-second period.

## 2 Related Work

**Human motion prediction** Many studies on human motion prediction use RNNs [3, 4, 13, 14, 22, 23], which are effective for modeling time series information. The encoder-decoder model, which predicts future movements from encoded information by observing past movements of people, is the mainstream. Within this framework, Martinez *et al*. [14]

have improved the accuracy by predicting the amount of pose change by using a residual network architecture. In addition to RNNs, GANs and convolutional autoencoders are also used [5, 11, 12], but the common denominator is that they observe only the movement information of the target's whole body in the past and predict the subsequent movements.

Some studies have focused on the dependence of each body part in order to make accurate predictions [6, 9]. In Structural-RNN [9], the skeleton and interactions between body parts are represented by a spatio-temporal graph and all factors in the graph, like the nodes, are transformed into RNNs in order to learn and predict human motion. In addition to these approaches, Corona *et al*. [2] have proposed a pose prediction model that is aware of the interaction between humans and objects. In their model, not only information on human pose but also the positions and types of objects in the environment are taken into account to improve the prediction accuracy. However, the interactions between people are not considered as deeply as in this paper, because of the different purpose of the learning dataset.

**Human interaction** Pedestrian path prediction often takes into account the interaction between people [1, 17]; its accuracy is improved by recognizing the relationship between pedestrians. Its effectiveness has also been confirmed in the field of action recognition [8, 13, 24]. One of the most common methods is to extract the features of each person by using an RNN or CNN and update the features by considering the features of the other person. For example, Sadeghian *et al*. proposed a trajectory prediction model in which the movements of different pedestrians are represented by the intermediate output of the long-short term memory (LSTM) [7] and are used to consider their interactions [17]. Ibrahim *et al*. proposed a network that takes into account relationships between players for event recognition in volleyball matches. Their method is based on a representation that a CNN extracts from video of each player [8]. Wu *et al*.'s method uses a graph neural network (GNN) to represent the characteristics of human interactions as nodes and edges of a graph; it uses nodes to represent players in a volleyball game to help understand the interaction [24].

In this paper, we propose an architecture for human motion prediction inspired by Social LSTM [1], which also considers interactions by using RNN, because fencing is a two-player competition, and there is a possibility that the graph structure does not have much meaning and an RNN is a reliable means of motion prediction.

# 3 Method

We present a two-player simultaneous human motion prediction model that takes into account the interaction between players. We assume that a single person's motion model can be represented by an RNN. The interaction between the RNNs can be realized by feeding the outputs of the middle layers to each other. Namely, the output from one player's model is fed to the other player's model by concatenating it with the input in the next time step.

**Single-player motion prediction by RNNs** In this study, one neural model is prepared for a single player. We chose RNNs as the neural models, since they handle time series information well. To incorporate the interaction between players, one might consider a neural model that takes all players together as input and models the game entirely. However, we treat a single player as a single model that interacts with another model through the exchange of intermediate outputs. This has several advantages. First, each neural model is more focused on a single player than one encoding all players, and this would reduce the number of parameters. Second, a single-player model is easier to extend. There are generally individual

differences in the same movement, which poses a challenge in predicting the movement. In fencing, this difference manifests itself as a style of play, and it is desirable that each player be understood individually in order to make accurate posture predictions. If a single model can represent a single player, it would be easier to optimize it to individual players in the future. Also, when such a model is extended to a multiplayer or team sport such as football, it would be easier to represent the local interaction between players if a single model represents a single player rather than an entire match.

**Motion prediction by an RNN with interaction** To consider the interaction of players, we use the middle layer output of the RNNs as a feature that well represents the context of the movement. The middle layer output includes past posture information; thus, that of the opponent would contain not only its posture at that time but also its posture change, that is, the context of his/her movement.

Figure 2 illustrates the core of this interaction realization. Let the input be $p$, which is a vector of joint coordinates of the players in the $x,y$ image plane, or a further encoded latent representation of it. Suppose that at each time step $t$, the RNN model of a single player outputs the hidden state $h_t$ as follows:

$$h_t \leftarrow RNN(p_t, h_{t-1}; W_r), \tag{1}$$

where $W_r$ is the set of parameters for this recurrent model. Our model that considers the interaction can be expressed as follows:

$$h_t^{left} \leftarrow RNN(p_t^{left} \frown h_{t-1}^{right},\ h_{t-1}^{left}; W_r), \tag{2}$$

$$h_t^{right} \leftarrow RNN(p_t^{right} \frown h_{t-1}^{left},\ h_{t-1}^{right}; W_r), \tag{3}$$

where $\frown$ means concatenation, and *left* and *right* indicate each player. $W_r$ are shared by the players. Rather than concatenating $h_{t-1}$ from the opponent with its own output, or specializing the RNN architecture, concatenating it with the input $p_t$ can be more natural because the motion of the opponent is information that comes from outside the player's neural model.

**interaction in encoder-decoder models** We are interested in generating a set of future frames by observing a set of frames of the two players as inputs. Encoder-decoder models are suitable for this problem setting, as they can encode the input frames into compact features and can decode them for a variable number of future frames.

The algorithm is shown in Algorithm 1. The input is a set of $p$, a vector of joint coordinates, or a further encoded latent representation of it, in $K$ frames. The output is a set of those for $T$ frames. At each time step $t$, the hidden states $h_t$ for both players are obtained from Eqs. 2 and 3. After observing $K$ frames, the RNN outputs the predicted joint position $\hat{p}_t$ for $T$ frames by $f(h_t; W_f)$, where $f$ is a projection function and $W_f$ is a projection matrix.

For the loss function to train the model, we used the Euclidean distance between the predicted and the true joint positions for each player. Namely, it is the mean of the sum of the squares of the errors:

$$\mathcal{L}_{pose} = \frac{1}{TL}\sum_{t=1}^{T}\sum_{l=1}^{L}(\|\hat{p}^l_{K+t} - p^l_{K+t}\|_2^2 + \|\hat{p'}^l_{K+t} - p'^l_{K+t}\|_2^2), \tag{4}$$

where $L$ is the total number of joints, $\hat{p}$ and $\hat{p'}$ are the predicted joint positions, and $p$ and $p'$ are the ground truth of the joint position, for each athlete.

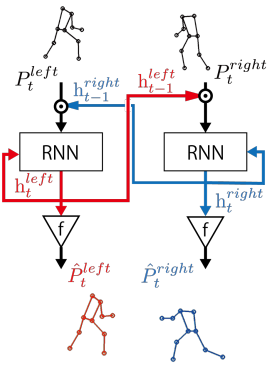


Figure 2: At each time step $t$, the previous hidden state $h_{t-1}$ of the opponent is concatenated with the input $p_t$, as shown by the red and blue arrows.

**Algorithm 1** Our basic algorithm

**Input:** $p^{left}_{1:K}, p^{right}_{1:K}$

**Output:** $\hat{p}^{left}_{K+1:K+T}, \hat{p}^{right}_{K+1:K+T}$

In encoder RNN:

**for** $t = 1$ to $K$ **do**

$h^{left}_t \leftarrow \mathrm{RNN}(p^{left}_t \frown h^{right}_{t-1}, h^{left}_{t-1}; W_r)$

$h^{right}_t \leftarrow \mathrm{RNN}(p^{right}_t \frown h^{left}_{t-1}, h^{right}_{t-1}; W_r)$

**end for**

In decoder RNN:

**for** $t = K+1$ to $K+T$ **do**

$h^{left}_t \leftarrow \mathrm{RNN}(p^{left}_t \frown h^{right}_{t-1}, h^{left}_{t-1}; W_r)$

$\hat{p}^{left}_t \leftarrow f(h^{left}_t; W_f)$

$h^{right}_t \leftarrow \mathrm{RNN}(p^{right}_t \frown h^{left}_{t-1}, h^{right}_{t-1}; W_r)$

$\hat{p}^{right}_t \leftarrow f(h^{right}_t; W_f)$

**end for**

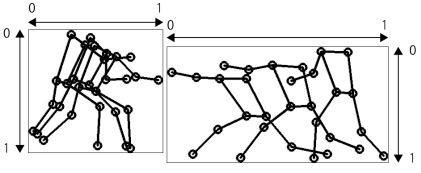


Figure 3: The joint coordinates of each player are normalized by the minimum and maximum positions appeared in the image.

To avoid low prediction accuracy due to accumulation of errors in the prediction values, our RNN has a structure that reuses its own predictions as input at training.

**Prediction target** We predict the absolute movement in the scene to some extent, rather than predict the change of each body part. This is because the change in movement due to the interaction appears prominently in the player's forward and backward movements. To implement this, the input/output joint coordinates are normalized every $K+T$ frames using the maximum and minimum values of the joint coordinates in the $K+T$ frames so that the values are in the range from -1.0 to 1.0. Figure 3 shows the image of our normalization. For each $K+T$ frames, the bounding box for normalization has different size.

**Implementation with three single-player models** It is desirable for the proposed mutual connections to work in any RNN model, and the proposed mechanism needs to be applied and validated in multiple models to show its effectiveness. Therefore, we tested our mutual connections with three types of single-person posture prediction models, LSTM-3LR' [3], LSTM [22], and rSA [14]. The network diagram of LSTM-3LR', LSTM, and rSA are shown from left to right in Figure 4. In these prediction models, the middle layer is connected at each time step to the next inputs, as shown by the red and blue arrows. When there are multiple layers in a time step, we only feed the output from the final layer to the next time step. For LSTM-3LR', we removed the noise scheduling and changed the learning method

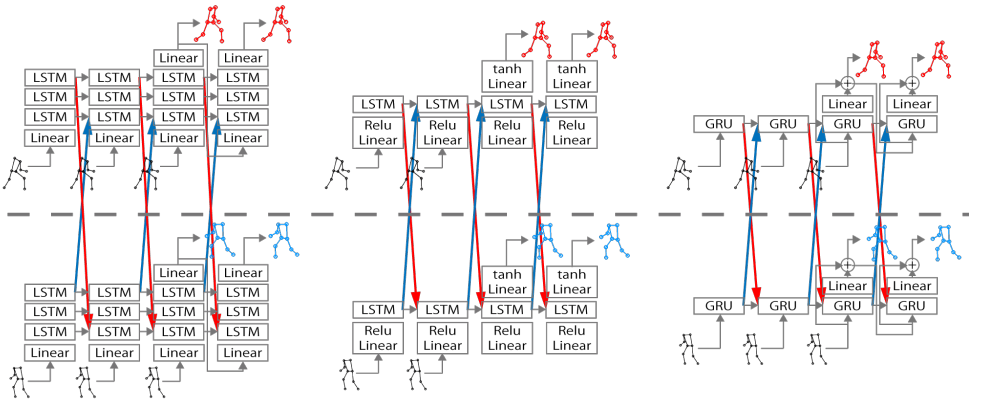


Figure 4: From left to right, diagrams of LSTM-3LR' [3], LSTM [22], and rSA [14] are illustrated. In each diagram, the networks above and below the dashed line are that of single pose prediction models. The red and blue arrows show that the middle layers of the respective models are connected to the opponent's model. The black skeletons are the input pose of the left and right players, and the red and blue skeletons are the predicted pose of them.

to free running because the original paper [3] was difficult to implement as is.

# 4 Experiment

## 4.1 Experimental setting

To demonstrate the effectiveness of the proposed method, we compared the accuracy of the single-person motion prediction models, LSTM-3LR' [3], LSTM [22], and rSA [14], with the corresponding models that consider interactions. As baselines, the models were simply applied to each athlete. Then, to validate the effectiveness of the proposed method, the intermediate layers of the models were mutually connected.

To account for the fast movements of fencing athletes, the joint positions of the two athletes in 30 frames, equivalent to 0.5 seconds, were entered as preliminary movements, and the joint positions in 60 frames, equivalent to the subsequent 1 second, were predicted. Namely, in Algorithm 1, the variable $K$ was set to 30 and the variable $T$ was set to 60.

**Dataset** A dataset was created from fencing match footage provided by a national Fencing Federation. The details after pre-processing and cleaning are shown in Table 1. First, the joint positions of each athlete were obtained by using pose estimation with HRNet [19]. The estimated 17 joints were preprocessed and reclassified based on the dominant arm with the epee and the dominant leg on the same side. In addition, we removed seven joint points: the head, whose accuracy of position estimation is low because players wear masks, and the non-dominant arm, whose accuracy is low because it is often occluded by the body. A filter was used to remove fine noises, and the scenes were divided into clips which included about 135 frames and had a 45-frame overlap with the adjacent clips.

The total number of clips was 2,326. The number of athletes, though not unique, were 4,652, as each clip included two players. We used 1,628 of the 2,326 clips in the dataset for learning and 698 scenes for testing, so the ratio of training and testing data was 7:3.

Table 1: Details of the fencing dataset

| Total numbers | |
|---|---|
| Matches | 21 |
| Athletes | 34 |
| Clips | 2,326 |
| Players for train/test | 4,652 |

Table 2: Implementation details

| | |
|---|---|
| Pose representation | x coordinate $\times$10 joints |
| | y coordinate $\times$10 joints |
| # of frames for input | 30 |
| # of frames for output | 60 |
| Reference value for PCK@0.2 | Athlete's torso distance |

**Evaluation Metric** Percentage of correct key points (PCK) [25] was used to evaluate the prediction accuracy, as in Wu *et al*. [23]. This is an index to calculate the percentage of correct answers, and if the predicted joint position is within the threshold radius from the true value, the joint is judged to be correct. We use 20% of the trunk distance (PCK@0.2), which is the distance between the player's dominant shoulder and non-dominant hip, as the threshold. The higher the value of this index, the better the accuracy. It is often used in pose estimation. We did not use this measure for the loss function because it allows for a certain amount of error in the evaluation as the correct answer.

**Implementation Details** During training, we randomly selected 90 consecutive frames from each clip and predicted the joint positions from the 31-st to 90-th frames by using those from the 1-st to 30-th frames as input. After training, a quantitative evaluation was performed using the first 90 frames of the test scenes and a qualitative evaluation was performed using the first and last 90 frames of the same scenes.

The methods were implemented by TensorFlow. The batch size in training was 128, the learning rate was set to 0.001, and the Adam optimizer was used. To prevent an RNN gradient explosion, we applied gradient clipping with a threshold of 25 to all networks. The parameters of the rSA and LSTM layers were in accordance with the original paper [14, 22]. For LSTM-3LR', the number of LSTM units is set to 512 and the output of the linear encoding layer is set to 100 dimensions.

## 4.2 Results

**Quantitative Evaluation** The values of PCK@0.2 of the baselines and the proposed method are shown in Table 3. The proposed variant of LSTM-3LR', LSTM, and rSA improved accuracy by 1.7%, 1.6%, and 2.3% over that of the original models. This indicates that the proposed RNN connections are effective regardless of the baseline network structure. The most accurate method is the one using rSA, with an accuracy of 77.1%, which is influenced by the high prediction accuracy of rSA itself. While the accuracy of all joint points was improved, a comparison of the accuracy of each joint point shows that the accuracies of the dominant hand, dominant elbow, and dominant foot joint points were lower. These areas have larger movements than other parts, especially for the dominant hand and the dominant elbow, which are faster and more difficult to predict because they belong to the epee arm.

Next, we evaluated whether the prediction accuracy changes with the prediction time. The prediction accuracy in the first 30 frames of the 60 predicted frames is shown in Table 4. In general, the closer the predicted time is to the input, the more accurate the prediction becomes, so the PCK@0.2 for each method is higher than in Table 3. Looking at the degree of improvement in prediction accuracy numerically, it is noteworthy that LSTM-3LR', LSTM, and rSA are 0.5%, 0.5%, and 1.2%, respectively, which are smaller than the prediction of the entire 60 frames. It has been shown that the prediction of times close to the input can

Table 3: Prediction accuracy for all 60 frames

| Method | 10 Coordinates | dominant hand | dominant elbow | dominant foot |
|---|---|---|---|---|
| LSTM-3LR' [8] | 70.7 | 66.0 | 59.2 | 65.7 |
| LSTM-3LR' cross | 72.4 | 66.6 | 59.9 | 67.1 |
| LSTM [22] | 71.4 | 66.7 | 60.0 | 67.0 |
| LSTM cross | 73.0 | 68.1 | 61.2 | 67.8 |
| rSA [14] | 74.8 | 69.7 | 64.2 | 70.5 |
| rSA cross | **77.1** | 71.8 | 66.4 | 72.4 |

Table 4: Prediction accuracy for the first 30 frames

| Method | 10 Coordinates |
|---|---|
| LSTM-3LR' [8] | 79.3 |
| LSTM-3LR' cross | 79.8 |
| LSTM [22] | 80.4 |
| LSTM cross | 80.9 |
| rSA [14] | 82.8 |
| rSA cross | **84.0** |

Table 5: Prediction accuracy for the last 30 frames

| Method | 10 Coordinate |
|---|---|
| LSTM-3LR' [8] | 62.2 |
| LSTM-3LR' cross | 65.0 |
| LSTM [22] | 62.4 |
| LSTM cross | 65.2 |
| rSA [14] | 66.8 |
| rSA cross | **70.2** |

be made with high accuracy from the movements of only one player without considering the interaction between the players.

Table 5 shows PCK@0.2 for each method, calculated using only the last 30 frames from the 60 predicted frames. Compared to the results in Table 4, the prediction accuracy of each method is lower because the prediction time is farther away from the input. However, it should be noted here that the degree of improvement in numerical prediction accuracy of each method is 2.8% for LSTM-3LR', 2.8% for LSTM, and 3.4% for rSA, which is greater than that of predicting the entire 60 frames. Hence, this shows that the accuracy of the prediction over a long period of time can be improved by taking the interaction into account.

We also analyzed at which point the accuracy difference appears between the baselines and the proposed variants of them. We calculated the prediction accuracy of the methods while increasing the evaluation frames from 1 to 60. We observed that, in the first few frames, our methods and baselines had almost the same accuracy or baselines outperformed our methods. However, from the particular time point, the accuracy of our methods start to outperform baselines: 250ms in rSA, 360ms in LSTM, and 380ms in LSTM-3LR'. These timings are close to the reaction time of skilled fencers [15], and it implies our methods make prediction results which reflect fencing players' cognitive ability. More details are included in our videos.

**Qualitative Evaluation**

Next, we visualized and qualitatively evaluated the prediction results of rSA with the best prediction accuracy. In the following, the true value of the motion position is depicted in black, the baseline prediction results are depicted in yellow without any relation to left and right, and the left and right players are depicted in red and blue, respectively, for the prediction results of the proposed method that takes into account the interaction.

Figure 5 shows a scene where the left player (red) retreats to avoid the thrust of the right player (blue). Although simply applying the single-person motion prediction model does not predict the regression of the left-handed player (yellow), it can be seen that the proposed

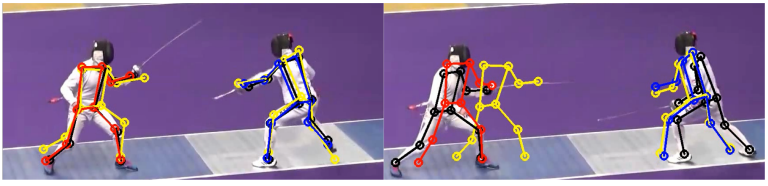

Figure 5: In response to an attack on the right flank, the player on the left fell back slightly. The true motion is depicted in black, the baseline prediction is in yellow, and the results by the proposed method are in red and blue.

method accurately predicts the regression of the player by considering the interaction (red). Such a spur-of-the-moment move in response to an opponent's attack would be a move with little or no reserve action, and thus, the predicted result would deviate significantly from the true value in the baseline without considering opponent information.

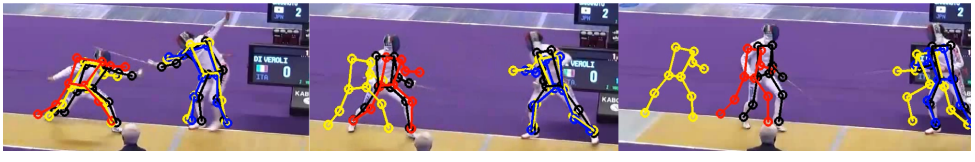

Figure 6: The left player stays in place to keep the distance between them as the right player backs away.

In Figure 6, both players approached once, and then the right player retreated significantly, so the left player can be seen to stay in place to keep the distance between them. The baseline (yellow) predicts that the player on the left who released the thrust will retreat, while the proposed method (red) predicts that the player will stay in place just like the actual player. Although the distance between players is not entered into the network, the proposed method predicts the behavior of players who maintain a certain distance by taking into account the forward and backward movement of the opponent. The video which shows more details of Figures 5 and 6 is provided in our supplementary material.

While there were results that improved the prediction accuracy by considering the interaction, there were also cases where the prediction results were incorrect as a result of considering the information of the opposing players. Figure 7 shows a scene in which the right athlete retreats in response to the left athlete's thrust, but the proposed method (blue) overreacts to the left athlete's thrust, causing the athlete to retreat farther than he actually did. One of the reasons why the information of opposing players may have a bad effect is that the distance between players is not taken into account. When an opponent makes a poking motion, the network has no information as to whether it is close or far away. Thus, even when the opponent pokes the player from a certain distance, as shown in the figure, the player is predicted to react as if he/her was poked from nearby. The same is true not only for thrusts, but also for other moves; to prevent this, the distance between players needs to be presented to the network.

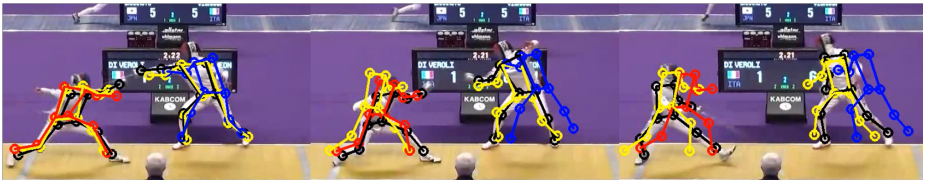

Figure 7: Example of poor prediction accuracy by the proposed method. The predicted result for the left player's thrust is excessively backward.

## 5 Conclusion

We proposed a motion prediction model for fencing athletes that considers their interaction through RNN connections. An athlete joint position dataset for fencing matches based on pose estimation by HRNet was created, and a quantitative evaluation using it showed that the prediction accuracy could be improved by taking into account the interaction between the athletes. In addition, a qualitative comparison between the baseline using rSA and the visualized prediction results of the proposed method showed that it is possible to predict the change in the player's movement when the motion prediction takes into account the information of the opposing players. Issues to be resolved in the future include improving the prediction accuracy of the epee hand and considering the distance between players.

## References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016.

[2] 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*, pages 6992–7001, 2020.

[3] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4346–4354, 2015.

[4] P. Ghosh, J. Song, E. Aksan, and O. Hilliges. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*, pages 458–466, 2017.

[5] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 786–803, 2018.

[6] 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.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997.

[8] Mostafa S Ibrahim and Greg Mori. Hierarchical relational networks for group activity recognition and retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, pages 721–736, 2018.

[9] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5308–5317, 2016.

[10] Japanese Fencing Federation. For Overseas Fencers. http://fencing-jpn.jp/for-oversea-fencers/. 2019-9-10.

[11] Jogendra Nath Kundu, Maharshi Gor, and R Venkatesh Babu. Bihmp-gan: Bidirectional 3d human motion prediction gan. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8553–8560, 2019.

[12] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee. Convolutional sequence to sequence model for human dynamics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5226–5234, 2018.

[13] Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363*, 2017.

[14] 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.

[15] Marko Milic, Aleksandar Nedeljkovic, Ivan Cuk, Milos Mudric, and Amador García-Ramos. Comparison of reaction time between beginners and experienced fencers during quasi-realistic fencing situations. *European Journal of Sport Science*, pages 1–10, 2019.

[16] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei. Detecting events and key actors in multi-person videos. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3043–3053, 2016.

[17] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1349–1358, 2019.

[18] C. Sun, A. Shrivastava, C. Vondrick, R. Sukthankar, K. Murphy, and C. Schmid. Relational action forecasting. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 273–283, 2019.

[19] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5686–5696, 2019.

[20] M. Thaler and W. Bailer. Real-time person detection and tracking in panoramic video. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1027–1032, 2013.

[21] T. Tsunoda, Y. Komori, M. Matsugu, and T. Harada. Football action recognition using hierarchical lstm. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 155–163, 2017.

[22] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to Generate Long-term Future via Hierarchical Prediction. In *ICML*, 2017.

[23] Erwin Wu and Hideki Koike. Futurepose-mixed reality martial arts training using real-time 3d human pose forecasting with a rgb camera. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1384–1392. IEEE, 2019.

[24] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu. Learning actor relation graphs for group activity recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9956–9966, 2019.

[25] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2012.

[26] H. Yu, S. Cheng, B. Ni, M. Wang, J. Zhang, and X. Yang. Fine-grained video captioning for sports narrative. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6006–6015, 2018.