Augmenting Pass Prediction via Imitation Learning in Soccer Simulations

Supplementary Material

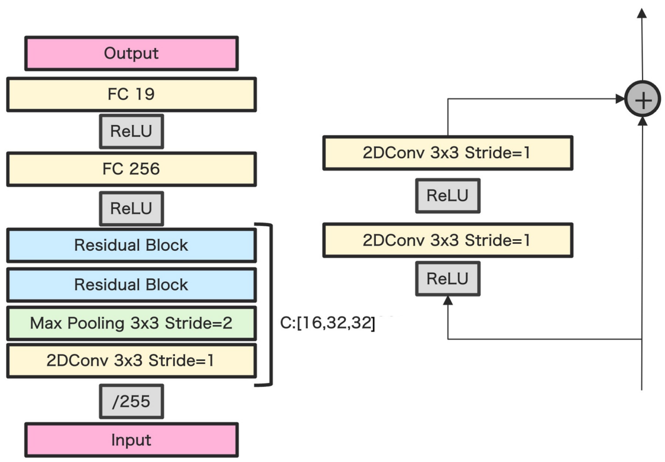

Figure 1. The structure of the three-layer CNN encoder in our method. This CNN encoder accepts player movement data as input and predicts one of the 19 types of action labels.

1. Labels and CNN Model in Behavior Cloning

In the behavior cloning model, action labels were used to define the types of movements or actions of athletes, primarily classified based on speed and direction. Details of these action labels are shown in Tab. 1.

Additionally, the labels for passing actions were classified based on the distance of the pass and its characteristics. Details of the labels are shown in Tab. 2. In the behavior cloning model, the CNN encoder used for predicting the next step action of a player making a pass is shown in Fig. 1. Considering the extensive use of simulators, this encoder modifies the CNN model used in GRF to enhance its efficiency and speed. It is a three-layer CNN encoder model with an added fully connected layer to predict 19 types of actions.

2. Scheduling Methods

In this experiment, we evaluated the impact of scheduling methods used when simultaneously training on real and synthetic data. Utilizing the number of epochs e and the epoch duration parameter E_M, we experimented with four methods to integrate the losses into L from real data L_r and synthetic data L_v. α and β are hyperparameters.

Fixed approach The loss function for the fixed approach can be represented as shown in Eq. (1).

$$L = \alpha L_r + \beta L_v \tag{1}$$

In the fixed approach, the weight for the loss of real data L_r and the weight for the loss of synthetic data L_v were kept constant throughout the entire training period, irrespective of the number of epochs E. The aim of this method is to maintain the balance between real and synthetic data unchanged from the beginning to the end of the training.

Linear approach The loss function for the linear approach can be represented as shown in Eq. (2).

$$L = \begin{cases} \alpha \frac{E}{E_M} L_r + \beta(1 - \frac{E}{E_M}) L_v & \text{if} E \leq E_M \\ \alpha L_r & \text{if} E > E_M \end{cases} \tag{2}$$

In the linear approach, the weight for real data was increased linearly with the number of epochs, while the weight for synthetic data was decreased. The aim of this method is to avoid abrupt changes in weights during the initial stages and gradually reflect more of the real data as the training progresses.

Sinusoid approach The loss function for the sinusoid approach can be represented as shown in Eq. (3).

$$L = \begin{cases} \alpha \sin\left(\frac{E}{E_M}\pi\right) L_r + \beta \cos\left(\frac{E}{E_M}\pi\right) L_v & \text{if} E \leq E_M \\ \alpha L_r & \text{if} E > E_M \end{cases} \tag{3}$$

In the sinusoid approach, the weight for real data was increased and the weight for synthetic data was decreased in the shape of a sinusoid as the number of epochs increased. The aim of this method is to introduce larger changes in weights early on compared to the linear approach, allowing real data to be reflected earlier in the training process.

Sigmoid approach The loss function for the sigmoid approach can be represented as shown in Eq. (4).

$$L = \alpha \frac{1}{1 + e^{-(E_M - \frac{E}{2})}} L_r + \beta \frac{1}{1 - e^{-(E_M - \frac{E}{2})}} L_v \tag{4}$$

In the sigmoid approach, the weight for real data was increased and the weight for synthetic data was decreased in the shape of a sigmoid curve as the number of epochs increased. The aim of this method is to smoothly transition the weights over a longer period of epochs, compared to the linear and sinusoid approaches.

3. Ablation Study on Data Generation Strategies

To demonstrate the effectiveness of strategy generation through imitation learning in our method, we generated synthetic data using two approaches: a strategy that performs

Table 1. List of behavior cloning label definitions

No.	Action name	Action definition
0	action_idle	$v_t \leq 1.0m/s$
1	action_left	$1.0m/s < v_t \leq 6.0m/s, -180 \leq \theta_t < -157.5, 135 \leq \theta_t < 180$
2	action_top_left	$1.0m/s < v_t \leq 6.0m/s, 112.5 \leq \theta_t < 157.5$
3	action_top	$1.0m/s < v_t \leq 6.0m/s, 67.5 \leq \theta_t < 112.5$
4	action_top_right	$1.0m/s < v_t \leq 6.0m/s, 22.5 \leq \theta_t < 67.5$
5	action_right	$1.0m/s < v_t \leq 6.0m/s, -22.5 \leq \theta_t < 22.5$
6	action_bottom_right	$1.0m/s < v_t \leq 6.0m/s, -67.5 \leq \theta_t < -22.5$
7	action_bottom	$1.0m/s < v_t \leq 6.0m/s, -112.5 \leq \theta_t < -67.5$
8	action_bottom	$1.0m/s < v_t \leq 6.0m/s, -157.5 \leq \theta_t < -112.5$
9	action_long_pass	long pass label
10	action_high_pass	high pass label
11	action_short_pass	short pass label
12	action_shot	shot command
13	action_sprint	$v_{t-1} \leq 6.0m/s, 6.0m/s < v_t$
14	action_release_direction	$v_{t-1} > 1.0m/s, v_t \geq 1.0m/s$
15	action_release_sprint	$v_{t-1} > 6.0m/s, v_t \geq 6.0m/s$
16	action_sliding	sliding command
17	action_dribble	$1.0m/s < v_t \leq 4.2m/s$,ball-holding player
18	action_release_dribble	$1.0m/s < v_{t-1} \leq 4.2m/s, v_t \leq 1.0m/sorv_t < 4.2m/s$, ball-holding player

Table 2. List of pass label definitions

Pass type	**Definition**
short_pass	Passes shorter than 36m
high_pass	Passes between 36m and 45m, or crosses and clearances
long_pass	Passes longer than 45m

Table 3. Comparison of pass prediction using synthetic data generated by random and AI strategies

Policy	Top-1	Top-3	Top-5	Loss
Random	57.54	90.41	97.08	1.204
Built-in AI	61.58	91.63	97.40	1.036
Cloning	66.00	93.20	97.95	0.9681

actions with equal probability and a strategy using built-in AI within the simulator. We compared the accuracy of pass prediction when trained with these synthetic datasets. Detailed comparisons are shown in Tab. 3. The Top-1 accuracies for the random AI and built-in AI strategies decreased by 8.46% and 4.42%, respectively, compared to the behavior cloning strategy of the proposed method. These results suggest that data generation using the behavior cloning strategy is more effective than data generation methods that do not capture players' strategies.