Action Anticipation from SoccerNet Football Video Broadcasts

Supplementary Material

A. Extended SN-BAA dataset analysis

In Tab. S1, we provide an extended analysis of the SoccerNet Ball Action Anticipation (SN-BAA) dataset. As shown in the table, the original SoccerNet Ball Action Spotting dataset, which has been adapted for the anticipation task, consists of $C = 12$ classes. These classes follow a long-tail distribution, with specially evidenced problems in the free-kick and goal classes, where only 21 and 13 examples are observed across the entire dataset respectively. Furthermore, in the test set, these classes appear only 2 and 6 times, making evaluation metrics for these classes highly unstable, as discussed in the main paper. Consequently, these classes are removed from SN-BAA. When analyzing the occurrences of the remaining classes in SN-BAA within an anticipation window of $T_a = 5$ seconds, we observe a similar pattern, with passes and drives occurring more frequently, while all other classes have a mean occurrence rate below 0.10.

	SN-BAS				SN-BAA w. $T_a = 5$	
Action	Train	Valid.	Test	Total	μ obs.	Max. obs.
Pass	2679	585	1721	4985	0.61	6
Drive	2297	554	1449	4300	0.52	4
High Pass	465	115	181	761	0.09	2
Header	404	127	182	713	0.09	5
Out	331	75	145	551	0.07	1
Throw-in	213	54	95	362	0.04	1
Cross	177	24	60	261	0.03	2
Ball Player Block	128	28	67	223	0.03	2
Shot	100	25	44	169	0.02	3
Succesful Tackle	34	12	28	74	0.01	2
FK	15	4	2	21	-	-
Goal	6	1	6	13	-	-
All	6849	1604	3980	12433	1.52	8

Table S1. Dataset statistics for the SN-BAS action classes, showing the total number of observations for each split, and for SN-BAA, the mean (μ) and maximum number of observations per anticipation window with $T_a = 5$ seconds.

B. Details of prediction heads

In this section, we provide a more detailed description of the alternative prediction heads analyzed in the ablation studies.

Q-EOS. This approach utilizes the original anticipation head from FUTR. For each query, there are two components: (i) an action classification component, and (ii) a timestamp regressor. The main difference compared to Q-Act is the absence of the action detection component (i.e., actionness), which is replaced by an additional class in the action classification component. This extra class corresponds to an End of Sequence (EoS) class, which is activated when no ground-truth action is paired with the query, signaling the end of prediction generation for subsequent queries. Thus, when an EoS is detected during inference in one query, following queries are discarded.

Q-Bckg. This approach builds upon Q-EOS but replaces the EoS class with a background class. Similar to the EoS class, the background class is activated when no ground-truth action is paired with the query. However, unlike the EoS class, this does not lead to the discard of subsequent queries during inference, and predictions for all queries are considered.

Q-BCE. Similar to Q-EOS and Q-Bckg, this approach omits the action detection component (i.e., actionness) found in Q-Act. Additionally, it modifies the softmax and cross-entropy loss function by using a sigmoid activation function for each class and binary cross-entropy loss, treating the action classes in each query independently. As in Q-BCE predictions for all queries are considered.

Q-Hung(t). This approach adapts Q-Act by modifying the pairing between ground-truth actions and predictions. Instead of sequentially pairing ground truths and predictions, it uses the Hungarian algorithm to pair them based on temporal position, aiming to find the optimal pairing by minimizing the distance between the predicted temporal positions of the queries and the temporal positions of the paired ground-truth actions.

Q-Hung(a). Similar to Q-Hung(t), this approach modifies the pairing between ground-truth actions and predictions. However, in this case, it uses the action classes to perform the pairing. The optimal pairing is determined by minimizing the distance between the predicted scores and the class of the ground-truth action.

Anchors. In this approach, each learnable query is anchored to a temporal window of size T_a/q within the anticipation window T_a. During training, each ground-truth token is assigned the first action within its anchor window; otherwise, it is given an actionness value of 0. Only one action is considered within each anchor window, and the temporal position to predict corresponds to the position within the anchor window. During inference, for each prediction, the temporal position is determined by adding the predicted position within the anchor window to the anchor's starting point.

C. Examples from the dataset

In this section three frames are shown for each action, a frame before the action (left), the frame of the action label (center), and a frame after the action (right):

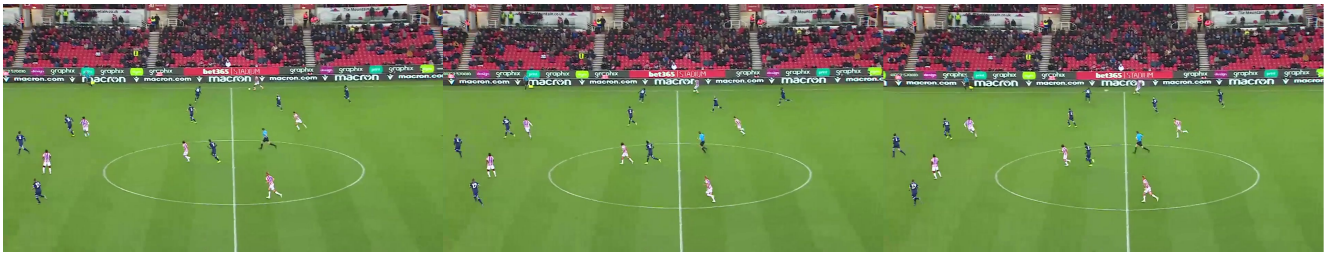

Figure S1. Example of a pass action

Figure S2. Example of a drive action

Figure S3. Example of a header action

Figure S4. Example of a high pass action

Figure S5. Example of an out action

Figure S6. Example of a throw in action

Figure S7. Example of a cross action

Figure S8. Example of a ball player block action

Figure S9. Example of a shot action

Figure S10. Example of a player successful tackle action