

# Football tracking data 를 이용한 events data 예측 및 분석



202055614 최성민

202055502 강동권

지도교수 이기준

---

## 목 차

1. 서론.....	1
1.1. 연구 배경.....	1
1.2. 기존 문제점 .....	1
1.3. 연구 목표.....	2
2. 연구 배경.....	2
2.1. Tracking Data 가공.....	2
2.2. 이벤트 예측 모델 설계.....	2
3. 연구 내용 .....	2
3.1. DualCrossAttention model 설계.....	2
3.1.1. Data 및 정합 .....	2
3.1.2. 전처리 파이프라인 .....	3
3.1.3. 모델 구성.....	3
3.2. Football Manager 2024 Tracking Data 가공.....	6
3.2.1. Video 녹화 및 Image Capture.....	6
3.2.2. 선수 탐지 및 위치 좌표화.....	6
3.2.3. YOLOv8 모델을 이용한 공 궤적 데이터 추출.....	7
4. 연구 결과 분석 및 평가 .....	7
4.1. 모델 추론 실험 진행 현황.....	7
4.2. FM data 추출 현황.....	8
5. 결론 및 향후 연구 방향 .....	8
5.1. 실패 원인 진단.....	8

---

5.2. 향후 연구 방향.....	8
6. 참고 문헌.....	9

---

## 1. 서론

### 1.1. 연구 배경

현대 스포츠에서는 데이터 기반의 전략 수립과 경기 분석이 점점 더 중요해지고 있으며, 그 중심에는 트래킹 데이터(tracking data)가 있다. 트래킹 데이터란 경기 중 선수들의 위치, 이동 경로, 속도, 가속도 등 시공간적인 정보를 일정 주기로 수집한 데이터로, 전통적인 이벤트 기반 데이터(예: 패스, 슈팅 기록 등)보다 훨씬 더 정밀하고 방대한 정보를 담고 있다. 이러한 데이터는 선수의 움직임 패턴, 포지셔닝 전략, 팀 전술 분석 등 다양한 목적으로 활용될 수 있다. 특히 축구의 경우 22명의 선수와 공이 넓은 필드 위를 끊임없이 움직이며 복잡한 상호작용을 보이기 때문에, 경기 이해를 위해 트래킹 데이터를 활용한 분석이 필 수적으로 여겨지고 있다. 하지만 현재 대부분의 분석 시스템은 사후적이고 정적이며, 실제 경기 상황에서 일어나는 이벤트를 실시간으로 예측하거나 맥락적으로 해석하는 데에는 한계가 있다. 예를 들어, 패스나 슈팅과 같은 이벤트는 단순히 공의 움직임만으로 설명되기 어렵고, 주변 선수들의 위치, 압박 유무, 속도 변화 등 다양한 요소가 결합되어 발생한다. 따라서, 트래킹 데이터를 기반으로 한 이벤트 예측 모델은 보다 높은 정확도와 풍부한 경기 해석을 가능하게 하며, 나아가 코칭 스태프나 전술 분석가에게 실질적인 인사이트를 제공할 수 있다. 또한, 머신러닝과 딥러닝 기술의 발전으로 이러한 예측 문제를 해결할 수 있는 가능성이 더욱 커지고 있다. 시계열 분석, 공간적 관계 모델링, 행동 패턴 인식 등의 기법을 트래킹 데이터에 적용하면, 경기 중 발생 가능한 주요 이벤트를 사전에 예측하고, 이를 기반으로 전략적인 결정을 내릴 수 있다. 따라서 본 과제는 축구 트래킹 데이터를 활용하여 이벤트 발생을 예측하고 이를 분석하는 시스템을 개발함으로써, 기존 분석의 한계를 극복하고 보다 지능적인 스포츠 분석 플랫폼의 기틀을 마련하는 데 의의가 있다.

### 1.2. 기존 문제점

현재 상황의 문제점으로는 축구 트래킹/이벤트 데이터가 대부분 팀·리그 내부 자산으로 관리되어 공개 코퍼스가 부족하다는 것이다. 이로 인해 재현성, 벤치마킹, 모델 일반화 검증이 어려우며, 연구·산업 전반의 진입장벽이 높다. 트래킹 데이터만으로도 주요 이벤트(타입/행위자)를 자동 생성(pseudo-labeling)하는 파이프라인을 구축하면, 이벤트 로그에 대한 의존도를 줄이고 추가 수집 없이도 학습용/분석용 '축구 이벤트 데이터'를 생성할 수 있다.

---

### 1.3. 연구 목표

본 연구의 목표는 축구 트래킹 데이터로 주요 이벤트(예: Pass/Shot/DefensiveAction)와 행위자(actor)를 예측하는 모델·파이프라인을 구현하고, 불균형/라벨 정합/메모리 제약 등 현실 문제를 해결하는 것이다. 추후 데이터 보강 및 연구 개방성, 운영 효율을 높일 수 있는 기여 가능성이 있다.

## 2. 연구 배경

### 2.1. Tracking Data 가공

Football Manager 2024란 Sports Interactive에서 개발하고 SEGA에서 유통한 축구 시뮬레이션 게임으로 현실성과 재미를 모두 갖추어 많은 유저들에게 사랑받는 게임이다. 본 연구의 첫 번째 목표로 이 게임의 in-game video를 tracking data로 추출해 부족한 tracking data를 보충하여 연구에 도움이 되게끔 하는 것이 목표이다.

### 2.2. 이벤트 예측 모델 설계

본 연구의 주된 목표로, 축구 Tracking data와 Events data를 이용해 모델을 설계하여 최종 목표인 완전한 축구 데이터 생성에 기여하고자 한다.

## 3. 연구 내용

### 3.1. DualCrossAttention model 설계

#### 3.1.1. Data 및 정합

Data는 [German Bundesliga](#) (1st and 2nd divisions)[1]에서 제공한 7개의 Full matches data로 Official metadata, Official event data, Official position data로 구성되어 있다. 7개의 match data에서 data가 불충분한 DFL-MAT-J03WN1을 제외한 6개의 data를 사용하였다.

- **Source:** 25fps 트래킹(전·후반 분리), 이벤트 로그(half별 gameclock=0부터).
- **Entity:** 공 1 + 선수 22 = **23개체**(Ball/Home/Away).
- **Input Feature(프레임 단위):** x, y, vx, vy, speed, is\_actor, team\_(Ball/Home/Away), possession\_(Home/Away), ball\_status (총 12차원).
- **Window size:** 길이 T=125프레임(≈5초), stride=1, 각 윈도우의 마지막 프레임 라벨을

타킷으로 사용한다.

- **Labeling:** 이벤트 로그를 프레임에 매칭. 한 프레임에 다중 이벤트가 매칭될 때는 우선순위 규칙으로 1개만 남긴다. EventType\_Parent='Unknown'은 NoEvent로 처리.

### 3.1.2. 전처리 파이프라인

- 팀시트 기반 ID 정합(pID/xID→jID).
- 트래킹 Long-format 변환(프레임별 23×F 행렬).
- 이벤트 축약(Pass/Shot/DefensiveAction/Foul/BallControl/...), Unknown→NoEvent.
- **merge\_asof(gameclock)** 또는 half 별 인덱스로 이벤트→프레임 매칭, 중복 프레임은 우선순위로 1 건만 유지.
- 슬라이딩 윈도우 생성(T=125), 입력 텐서 (T, 23, 12) 구성.

데이터셋 예시) frame 0 시점

```
--- 데이터 샘플 ---
```

frame_id	team	player_id	x	y	possession	ball_status	EventType_Parent	EventContext	is_actor	vx	vy	speed
0	0	Ball	NaN	-5.67	0.11	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
3	0	Home	3.0	-16.45	-6.50	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
11	0	Home	7.0	-1.85	13.53	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
8	0	Home	9.0	0.99	8.39	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
1	0	Home	10.0	0.59	-8.72	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
4	0	Home	11.0	-0.12	-18.05	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
2	0	Home	25.0	-12.53	-20.32	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
5	0	Home	29.0	-4.20	-7.80	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
7	0	Home	30.0	-17.32	5.08	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
6	0	Home	31.0	-11.49	-2.73	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
9	0	Home	33.0	-40.36	-0.25	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
10	0	Home	34.0	-13.81	14.50	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
17	0	Away	1.0	-44.87	-0.01	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
13	0	Away	5.0	-10.78	-5.12	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
16	0	Away	7.0	-1.55	-13.24	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
18	0	Away	8.0	-7.02	4.81	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
20	0	Away	11.0	-15.58	-14.87	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
19	0	Away	18.0	0.44	0.10	Away	Alive	Pass SetPiece_KickOff	1	0.00	0.00	0.000000
21	0	Away	19.0	0.42	25.16	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
12	0	Away	20.0	-10.31	21.76	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
15	0	Away	23.0	-21.51	1.08	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
14	0	Away	29.0	1.42	16.90	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
22	0	Away	33.0	-20.06	11.81	Away	Alive	Pass SetPiece_KickOff	0	0.00	0.00	0.000000
23	1	Ball	NaN	-6.28	0.08	Away	Alive	NoEvent NoEvent	0	-15.25	-0.75	15.268431
26	1	Home	3.0	-16.47	-6.56	Away	Alive	NoEvent NoEvent	0	-0.50	-1.50	1.581139

### 3.1.3. 모델 구성

#### 3.1.3.1. 설계 목표(Inductive Bias)

트래킹 기반 이벤트는 (a) 공 중심 컨텍스트, (b) 팀 내부 협력과 상대 간 대치라는 공간적 상호작용, (c) 시간적 전조 패턴(준비→실행)의 결합으로 설명된다. 본 모델은 이 세 축을 아키텍처 단계에서 직접 반영하는 것을 목표로 한다.

---

### 3.1.3.2. 구성 요소와 채택 이유

#### (1) Ball gating — 선수 임베딩에 '공 맥락' 주입

- 이벤트는 대부분 공 주변에서 발생하므로 공의 위치/속도는 가장 강력한 조건부 정보이다.
- 간단한 선형 게이트로  $H = H + g(\text{Ball})$ ,  $A = A + g(\text{Ball})$  형태로 주입하여, 모든 선수 토큰이 현재 공과의 상대적 맥락을 내장한 채 다음 어텐션 단계로 넘어간다.
- 장점: concat 대비 파라미터/연산이 가벼우며, 데이터가 적어도 안정적으로 "공 중심" 신호를 전달한다.

#### (2) Dual Cross-Attention (Home↔Away) + 팀별 self-attn — 프레임 내 상호작용 구조화

- 팀 내부 self-attention 은 라인 간격, 포지셔닝, 역할 분담 같은 협력 패턴을 포착한다.
- 이어지는 양방향 cross-attention 은 마킹, 압박, 돌파 등 상대 간 상호작용을 직접 모델링한다.
- 모든 엔티티를 한꺼번에 global self-attn 으로 돌리는 것보다 도메인 유도편향이 강하고, 데이터가 제한적일 때 수렴이 안정적이다.

#### (3) 프레임 CLS + Temporal Transformer — 시간적 전조 포착

- 프레임 토큰 집합(공+선수)에 CLS 토큰을 두어 프레임의 전역 요약을 구성합니다.
- **Temporal Transformer Encoder**(temp\_layers)가 윈도우 전 구간의 준비→실행 패턴을 학습합니다.
- 라벨이 마지막 프레임에 부여되므로, 시퀀스 표현의 마지막 스텝만 읽어 예측하여 시점 정합을 보장하고 정보 누설을 방지한다.

#### (4) Two Heads (Event Type, Event Actor) — MultiTask 의 이점

- 백본 표현을 공유하되, Type(전역 context)과 Actor(지역 context)의 성격 차이를 고려해 head 를 분리한다.

#### (5) 누설 방지 마스킹 — 마지막 프레임 is\_actor=0

- is\_actor 는 정답에 직결되는 신호이므로, 마지막 스텝만 0 으로 마스킹해 치팅을 막고 일반화를 높인다.

---

### 3.1.3.3. 대안과 TradeOff

**Global self-attn 단일 구조:** 단순하지만 상호작용 구조가 희석되고, 저데이터 환경에서 수렴이 불안정할 수 있음.

**GNN(그래프 메시지패싱):** 관계표현에 강점이나 구현/튜닝 복잡도 ↑, 시간축 결합 비용 ↑.

**Conv/LSTM 기반:** 경량·고속이나 장거리/다자 상호작용 표현력이 한계.

**Ball concat vs gating:** concat 은 파라미터/학습 부담 ↑, gating 은 공 컨텍스트를 균일·경제적으로 주입.

### 3.1.3.4. 학습 설정 & Engineering Issue

#### 1. 불균형

- 1 경기 기준 라벨 분포 예시(학습셋): NoEvent >> Pass > DefensiveAction > Shot > ... (NoEvent 가 >99% 수준).
- 대응: WeightedRandomSampler(클립=10)만 사용, CE 클래스 가중치 비활성화(이중 보정 방지). label\_smoothing=0.1 적용

#### 2. 수치 안정성

- 초기 학습에서 NaN 발생 → 원인: AMP overflow/빈 마스크 계산.
- 조치: LR=2e-4 로 낮추고, 빈 마스크 시 loss=0 텐서 처리한다., AMP 일시 OFF 로 디버깅 후 재활성화, grad clip=1.0.

#### 3. 메모리/성능

np.stack 대용량 복사로 RAM **폭주** 문제 → 사전할당 + float16 저장으로 해결, DataLoader에서 float32로 캐스팅.



---

### 3.1.3.5. 손실 함수 개요

$L_{type}$  = Cross-Entropy

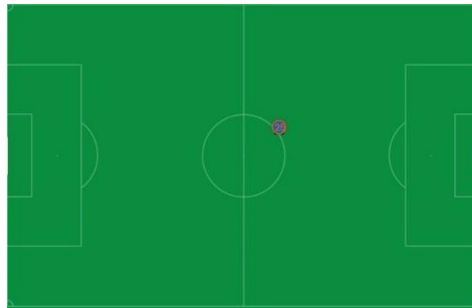
$L_{actor}$  : Masked Cross-Entropy

$L_{total} = w_{type}L_{type} + w_{actor}L_{actor}$

## 3.2. Football Manager 2024 Tracking Data 가공

### 3.2.1. Video 녹화 및 Image Capture

해당 Match의 선수 별 이동 경로 비디오를 녹화한 후에 25fps로 이미지 캡처 후 Cropping을 통해서 객체를 탐지할 수 있게 전처리한다. 예시 이미지는 다음과 같다.

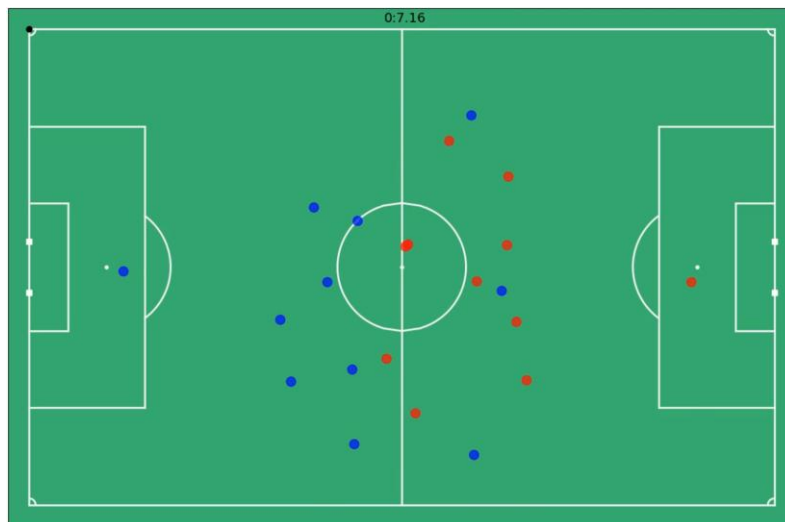


### 3.2.2. 선수 탐지 및 위치 좌표화

이미지에서 선수의 위치를 탐지하고 해당 좌표를 시각화 및 영상화 하여 비교한다.

LaurieOnTracking [2] 의 Metrica-sports sample data [3] processing code를 참조하였다.

선수의 움직임을 우선으로 tracking dataset을 구성하였다.



Period	Time [s]	Home_1_x	Home_1_y	Home_2_x	Home_2_y	Home_3_x	Home_3_y	Home_4_x	Home_4_y	Home_5_x	Home_5_y	Home_6_x	Home_6_y	Home_7_x	Home_7_y	Home_8_x	Home_8_y
1	0	43.78317	-0.0473	11.57438	0.310031	0.855867	10.43078	4.473528	-14.3183	16.0157	17.09284	18.85451	5.878774	7.831876	6.431081	17.03155	-16.8712
1	0.04	43.77362	-0.02633	11.59117	0.313501	0.867671	10.4089	4.44835	-14.3579	16.01588	17.05281	18.86907	5.854954	7.840474	6.439376	17.04254	-16.8814
1	0.08	43.76398	-0.00707	11.60536	0.317869	0.879044	10.38746	4.42493	-14.3956	16.0166	17.0151	18.88131	5.832728	7.847194	6.446595	17.05432	-16.89
1	0.12	43.75427	0.010498	11.61698	0.323136	0.889987	10.36646	4.403268	-14.4314	16.01788	16.97972	18.89124	5.812096	7.852037	6.452736	17.06689	-16.8971
1	0.16	43.74448	0.026365	11.626	0.329302	0.900499	10.34591	4.383363	-14.4652	16.0197	16.94666	18.89884	5.793057	7.855002	6.457801	17.08025	-16.9026
1	0.2	43.73461	0.040532	11.63245	0.336367	0.910581	10.3258	4.365215	-14.497	16.02206	16.91592	18.90412	5.775612	7.85609	6.461788	17.0944	-16.9066
1	0.24	43.72466	0.052999	11.6363	0.34433	0.920232	10.30614	4.348825	-14.527	16.02498	16.88751	18.90708	5.759761	7.855301	6.464699	17.10935	-16.909
1	0.28	43.71463	0.063767	11.63757	0.353193	0.929453	10.28692	4.334193	-14.555	16.02844	16.86142	18.90772	5.745504	7.852634	6.466532	17.12509	-16.9099
1	0.32	43.70451	0.072836	11.63625	0.362955	0.938243	10.26814	4.321318	-14.581	16.03245	16.83766	18.90604	5.73284	7.848089	6.467289	17.14162	-16.9092
1	0.36	43.69432	0.080205	11.63235	0.373615	0.946603	10.2498	4.3102	-14.6051	16.037	16.81622	18.90203	5.72177	7.841668	6.466968	17.15894	-16.9069
1	0.4	43.68405	0.085875	11.62586	0.385174	0.954531	10.23191	4.30084	-14.6273	16.0421	16.7971	18.89571	5.712294	7.833368	6.46557	17.17705	-16.9031
1	0.44	43.6737	0.089845	11.61678	0.397633	0.96203	10.21446	4.293238	-14.6476	16.04775	16.78031	18.88706	5.704412	7.823191	6.463096	17.19596	-16.8977
1	0.48	43.66327	0.092116	11.60512	0.41099	0.969098	10.19746	4.287393	-14.6659	16.05394	16.76584	18.8761	5.698123	7.811137	6.459544	17.21566	-16.8907
1	0.52	43.64503	0.094528	11.5817	0.438712	0.984934	10.19124	4.290105	-14.6817	16.07012	16.76038	18.84998	5.688617	7.819914	6.442164	17.24856	-16.8878
1	0.56	43.6249	0.094624	11.55427	0.459546	0.998014	10.18503	4.28716	-14.6962	16.08615	16.75695	18.82268	5.685031	7.821437	6.428546	17.28233	-16.8818
1	0.6	43.60318	0.092405	11.55798	0.458268	1.008338	10.17633	4.290105	-14.701	16.10203	16.75642	18.79723	5.686604	7.817157	6.425724	17.31639	-16.8734
1	0.64	43.59112	0.087871	11.5517	0.466252	1.015905	10.16499	4.274163	-14.7023	16.12511	16.75964	18.7712	5.691432	7.808526	6.415435	17.35017	-16.8631
1	0.68	43.59293	0.081021	11.5383	0.48205	1.020717	10.15078	4.276237	-14.7063	16.13737	16.7675	18.74581	5.698468	7.796993	6.415652	17.38307	-16.8405
1	0.72	43.61311	0.071856	11.49504	0.511538	1.044705	10.12968	4.271552	-14.7048	16.11687	16.75898	18.72217	5.706701	7.729176	6.426376	17.39258	-16.8352
1	0.76	43.60894	0.065257	11.45207	0.542617	1.040871	10.1096	4.278993	-14.7115	16.10122	16.74901	18.70134	5.715164	7.682015	6.436954	17.41417	-16.8115
1	0.8	43.60645	0.059005	11.40664	0.57047	1.030944	10.08016	4.270159	-14.7294	16.09909	16.73788	18.68429	5.722927	7.645619	6.44406	17.43256	-16.7916

### 3.2.3. YOLOv8 모델을 이용한 공 궤적 데이터 추출

공의 궤적 데이터 같은 경우 선수 데이터처럼 공만 볼 수 있는 영상이 존재하지 않아서 전체 축구 경기에서 공만 탐지하여 궤적 데이터를 추출해야 한다.

이를 위해서 YOLOv8 모델을 사용하여 공 만을 탐지하여 궤적을 추출하는 방식을 채택하여서 최종 Trajectory data sample을 완성한다.

YOLOv8 모델 학습을 위한 데이터는 아래와 같은 이미지 set을 roboflow의 Computer vision tool을 이용해서 데이터 augmentation을 포함해 총 1124장의 이미지 데이터를 사용했다.



## 4. 연구 결과 분석 및 평가

### 4.1. 모델 추론 실험 진행 현황

현재 데이터셋의 극단적 불균형(NoEvent >> Event)로 확률 보정/임계값 설정 없이는 실전 성능을 왜곡 하는 문제가 발생했다.

또한 Foul/Goalkeeping 등과 같은 이벤트는 트래킹 데이터로만 판별이 본질적으로 어려

---

운 문제가 발생했다.

액터 예측은 공 근접, 소유 맥락 제약 없이 학습 시 단일 ID 고정 경향을 보였다. 액터 손실을 제외하고 이벤트 타입 신호를 우선 확보하고 Prior 보정 +  $\tau/\beta$  튜닝 등의 극단적 예측 편향 보정 방안을 더 마련하여서 학습을 진행해야 한다.

## 4.2. FM data 추출 현황

선수 데이터는 원활하게 추출 가능했지만, YOLOv8을 이용한 공의 궤적 탐지는 정확도가 많이 떨어져서 Robust한 데이터셋을 만드는 데에는 실패했다. YOLO 모델 학습을 더 시키거나, fine-tuning 등을 통해서 모델의 정확도를 올려서 사용하는 것으로 해야 한다.

따라서 Ball의 trajectory는 interpolating과 smoothing 과정을 통한 비교적 부정확한 dataset으로 구성됐다.

## 5. 결론 및 향후 연구 방향

### 5.1. 실패 원인 진단

#### - 데이터/라벨링 측면

이벤트의 극단적 불균형 문제와 라벨 동기화 오차, 클래스 축약 영향 등으로 인한 양성 밀도 하락과 기회 손실이 원인으로 진단된다.

#### - 모델/학습 측면

사전확률 왜곡인 weighted random sampler로 학습해 샘플 분포와 자연 분포와의 괴리로 인해 보정/임계값이 없을 때 event가 쏠리는 현상이 생겼고, feature의 단 순성으로 인한 맥락 피쳐 부족, 데이터 규모에 있어 일반화에 한계를 느낀 것이 주된 실패 원인으로 판단된다.

### 5.2. 향후 연구 방향

1. **추론 보정 완성:** 로짓  $\beta \cdot \tau$  동시 튜닝 + Temperature scaling(캘리브레이션), 신뢰도 리포트(ECE).
2. **라벨 세그먼트화:** 이벤트를 \*\*구간(label segment)\*\*으로 정의(예: Pass: 시작→첫 터치 또는  $\leq 2s$ ; Shot: 시작→데드/세이프 또는  $\leq 2s$ ). 대안으로  $\pm K$  프레임 확장(dilation).
3. **이벤트 중심 샘플링:** 이벤트-앵커드 양성 윈도우 + 충분히 떨어진 음성 윈도우.

4. **자연 분포 파인튜닝**: 샘플러 학습 후 샘플러 OFF 로 3–5epoch 미세조정.
5. **액터 제약**: NoEvent→actor=0 강제, 공-거리/소유팀 기반 후보 마스킹/가중(거리 게이팅).
6. **두 단계 구조**: (1) Event vs NoEvent 검출기(예: focal/BCE) → (2) 타입 분류기(+액터), 검출기로 이벤트 밀도 안정화.
7. **피쳐 강화·정규화**: 공 기준 상대좌표, 거리/각도/압박 지표,  $a/\Delta v$ , 세트피스 플래그, 좌우 반전 통일.
8. **여러 경기 확장**: 6 경기 전부 사용, 경기 단위 split/LOOCV 로 일반화 검증, 전역 event\_type\_map 정합.
9. **프레임 내 CLS 업데이트** 층 추가(선택)로 공간 요약 개선.
10. **그래프 기반 모델**(GNN + Temporal) 혹은 Hybrid(Transformer+GNN)로 관계표현 강화.

## 6. 참고 문헌

- [1] Bassek, M., Rein, R., Weber, H., Memmert, D. (2025). An integrated dataset of spatiotemporal and event data in elite soccer. *Scientific Data*, 12(1), 195. <https://doi.org/10.1038/s41597-025-04505-y>
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [3] M. Saerens, P. Latinne, and C. Decaestecker, "Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure," *Neural Computation*, vol. 14, no. 1, pp. 21–41, 2002
- [4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826. (Label Smoothing)
- [5] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *Proc. Int'l Conf. on Learning Representations (ICLR)*, 2019. (AdamW).

- 
- [6] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in *Proc. 23rd Int'l Conf. on Machine Learning (ICML)*, 2006. (PR-AUC 근거)
- [7] Metrica Sports, "Sample Tracking and Event Data (Open Data)," GitHub repository, accessed Sep. 19, 2025. <https://github.com/metrica-sports/sample-data.git>
- [8] LaurieOnTracking, "Laurie's code for reading and working with Metrica's tracking and event data, <https://github.com/Friends-of-Tracking-Data-FoTD/LaurieOnTracking.git>