

K리그-서울시립대 공개 AI 경진대회 Track1 알고리즘 부문

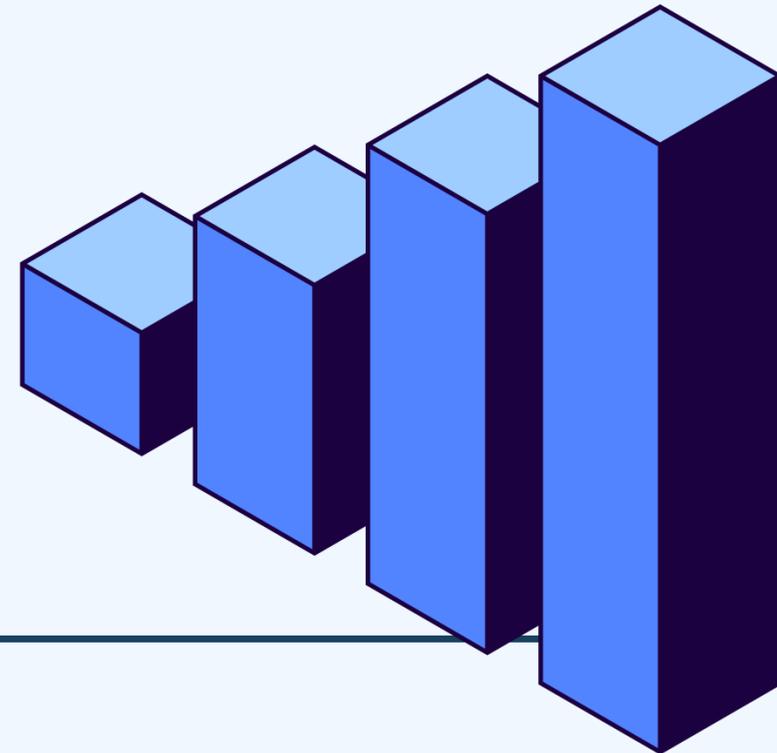
솔루션 보고서

팀명 : 루카돈치치_77

팀원 : 이종현, 명준호

목차

1. 문제 정의
2. 데이터 분석(EDA)
3. 초반: 기본 모델 구축 및 안정화
4. 중반: 핵심 개선(피처/전략) 도입
5. 종반: 최적화 및 앙상블 전략(추론 로직 완성)
6. 실험 결과 및 성능 개선 요약
7. 결론 및 부록: 상세 수치 및 설정



문제 정의

- 태스크 정의

- Task: 축구 경기 Episode별 시퀀스에서 마지막 패스 좌표 (end_x, end_y) 예측
- Input: Episode 내 액션 시퀀스(좌표/시간/타입/성공 여부 등)
- Output: 마지막 패스의 도착 좌표(또는 이를 구성하는 dx, dy 변위)

- 핵심 난점

- Episode별 시퀀스 길이 변동(2~100+ 액션) → 긴 시퀀스는 정보 희석 및 노이즈 누적
- 액션 타입(type_name)에 따라 패턴이 크게 다름
- **Hard Episode(긴 시퀀스, 복잡/변칙 패턴)**의 예측이 특히 어려움
- 직접 파생하기 어려운 선수별 플레이스타일 통계 필요

데이터 분석(EDA)

• 데이터 형태

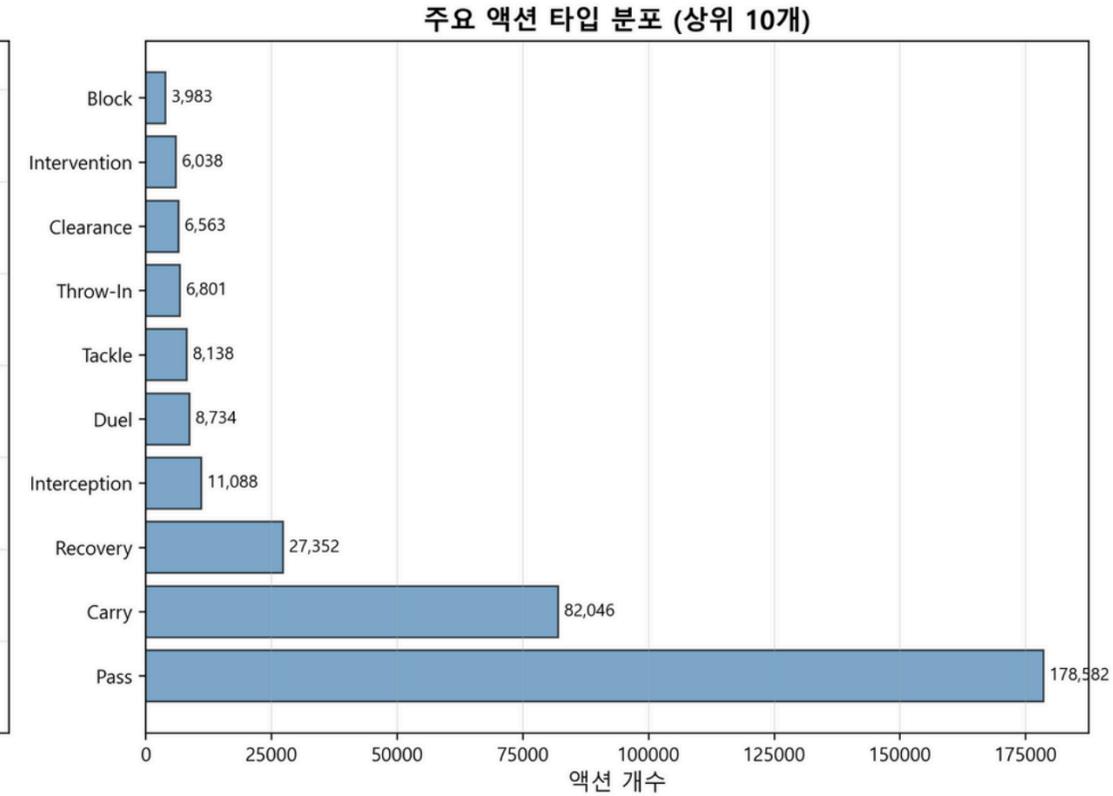
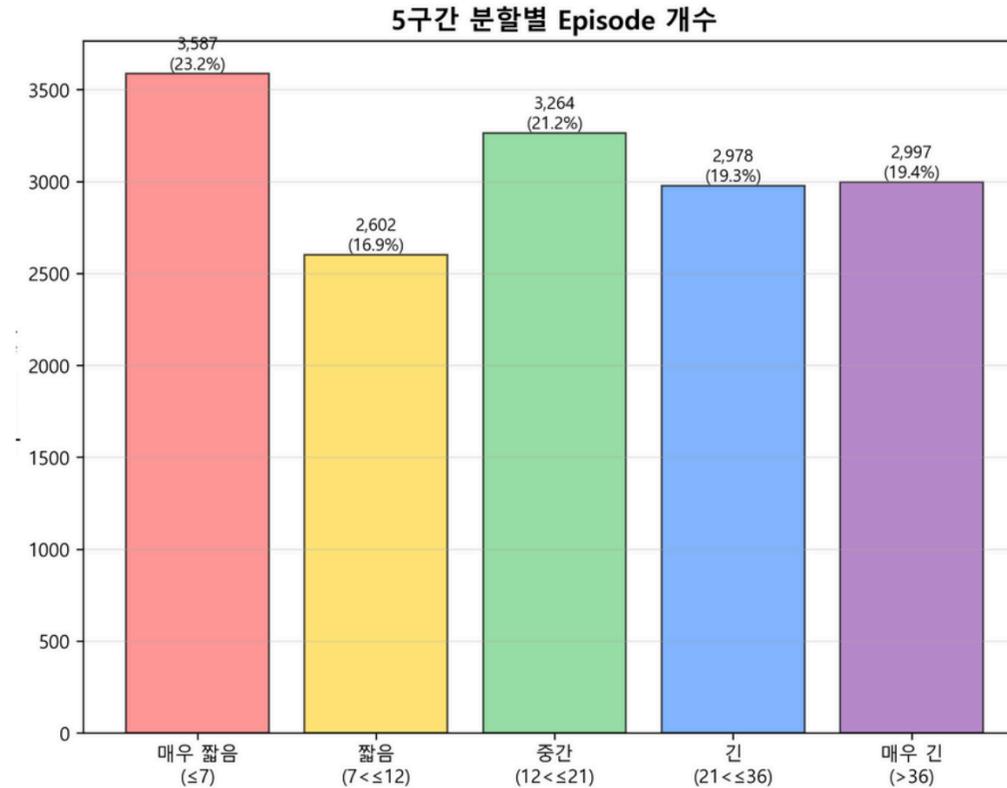
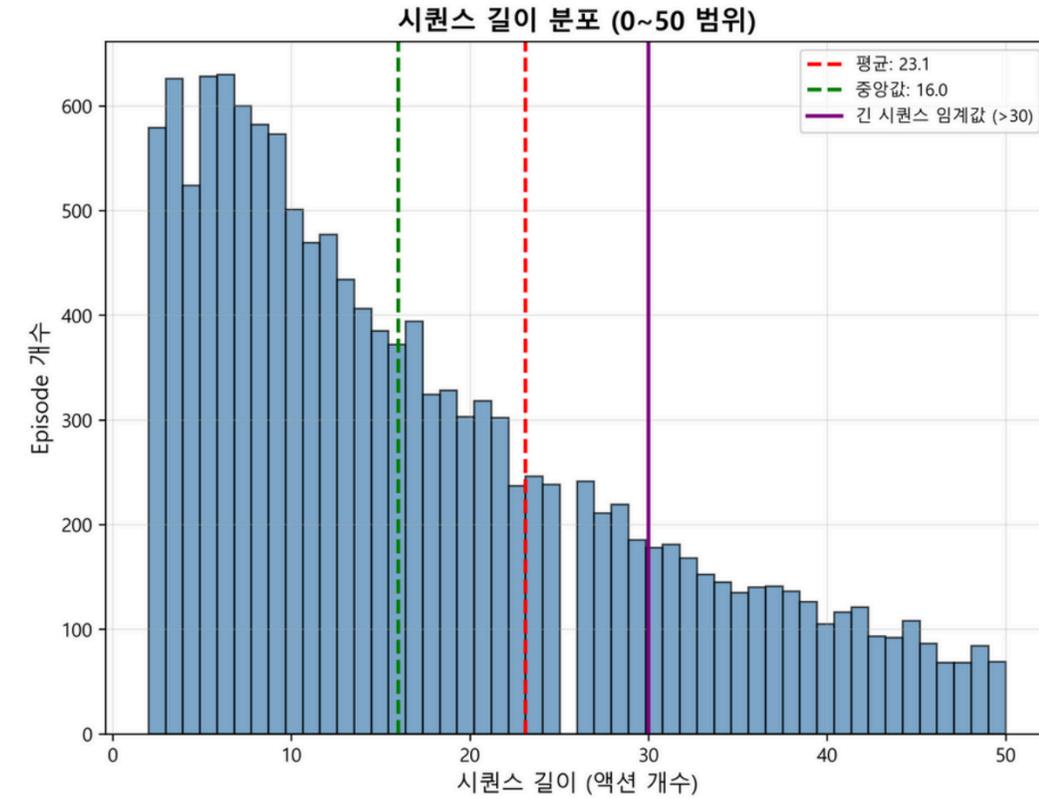
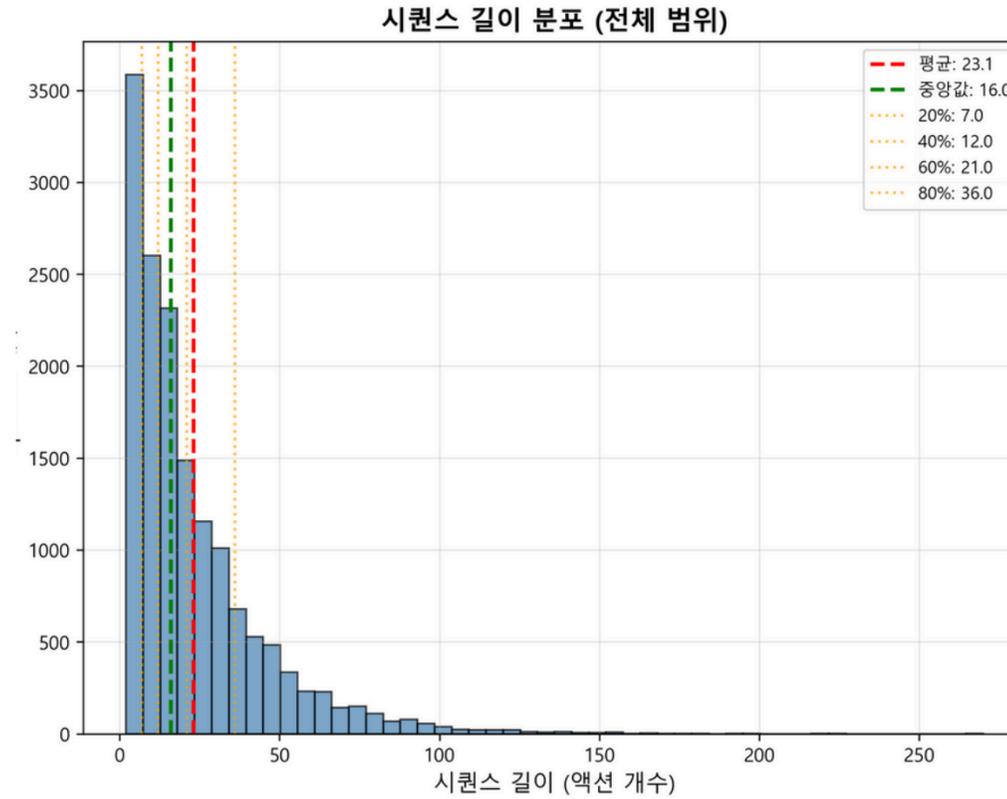
- Episode별 시퀀스 데이터(액션 시퀀스)
- 시퀀스 길이: 2~270개 (최대 270개, 긴 꼬리 분포 확인)
- 주요 액션 타입:
- Pass(178,582개), Carry(82,046개), Recovery(27,352개)가 압도적으로 많음
- 좌표 정보: start_x, start_y, end_x, end_y

• 시퀀스 길이 분포

- 평균: 23.1개 액션
- 중앙값: 16.0개 액션
- 20/40/60/80 백분위수 기반 5구간 분할 (7, 12, 21, 36 기준)
 - 매우 짧음 (≤ 7): 23.2%,
 - 짧음 (7-12): 16.9%,
 - 중간 (12-21): 21.2%,
 - 김 (21-36): 19.3%,
 - 매우 김 (> 36): 19.4%

매우 긴 시퀀스(> 30): 3,918개 (25.4%)

→ 예측 난이도 상승



데이터 분석(EDA)

• 분석 방법

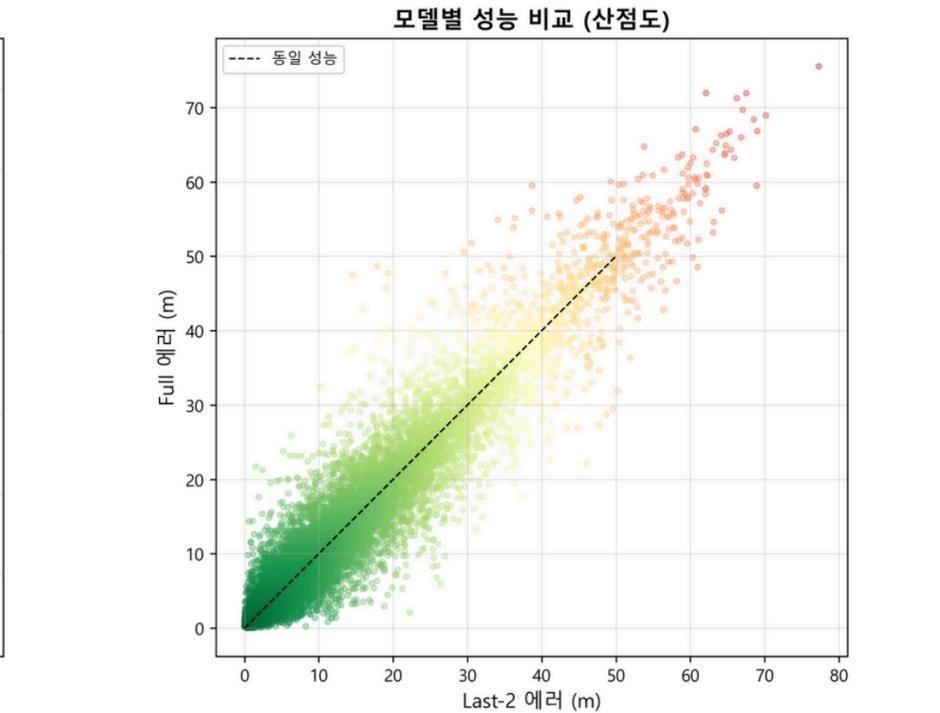
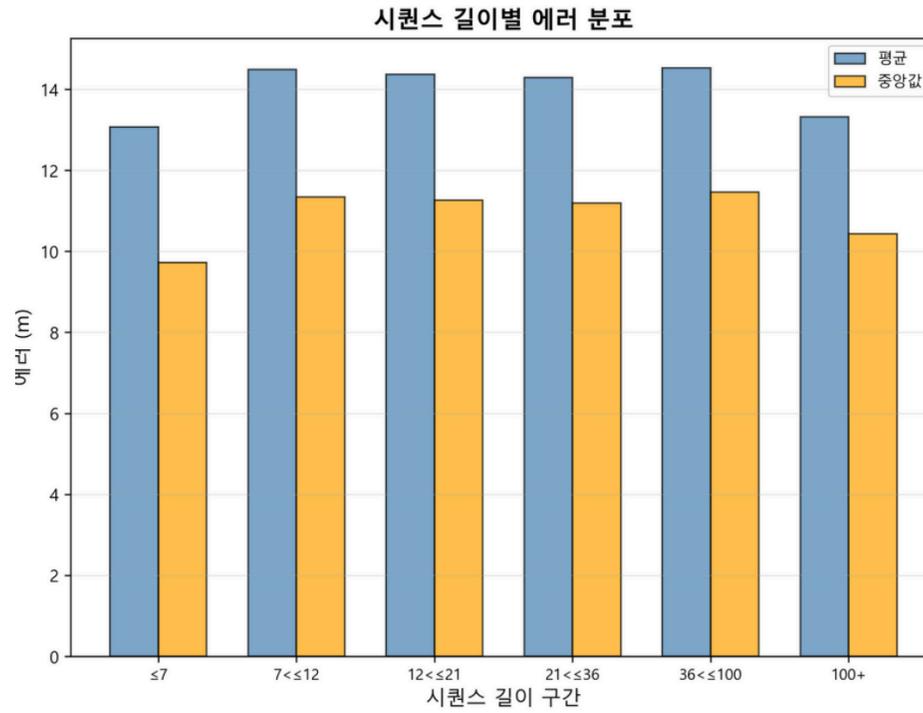
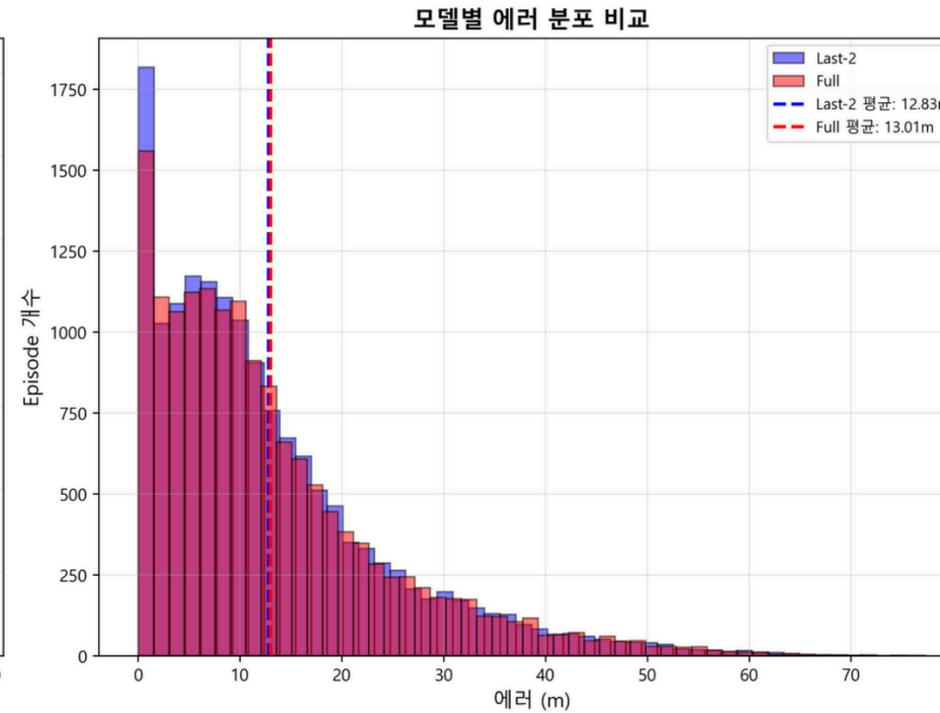
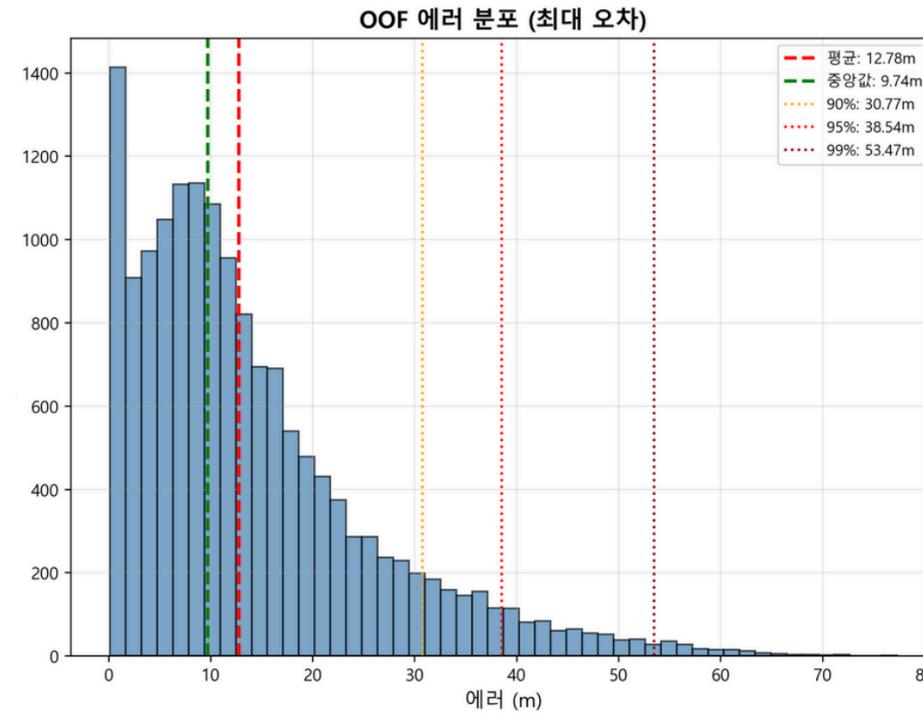
- 5-Fold CV의 OOF 예측값으로 실제 오차 계산 시퀀스 길이별/모델 타입별 성능 비교
 - Last-2: 마지막 2개 timestep만 사용하는 모델
 - Full: 전체 시퀀스를 사용하는 모델

• 핵심 발견

- 고오차 Episode 존재 (롱테일 분포)
 - OOF 에러 분포는 오른쪽으로 긴 꼬리를 가짐
 - 평균 오차: 12.78m, 중앙값 오차: 9.74m (평균 > 중앙값으로 비대칭 분포 확인)
 - 상위 10% 에피소드: 30.77m 이상, 상위 5%: 38.54m 이상, 상위 1%: 53.47m 이상
 - 극소수 에피소드에서 매우 큰 오차 발생 → 예측 난이도가 높은 "하드 에피소드" 존재

• 고오차 패턴 3종

- Zigzag: 급격한 방향 전환 ($\min_{\cos_last3} < 0$ 또는 < -0.1)
- Long-pass: 장거리 패스($pass_distance \geq 40m$)
- Long sequence: 긴 시퀀스($episode_length > 15$)에서 초기 정보 노이즈화



💡 **핵심 인사이트: 평균 오차 개선을 넘어, 극단적인 고오차 Episode(상위 10% 이상)의 특성을 파악하고 이를 해결하는 것이 LB 향상의 핵심**

데이터 분석(EDA)

• 분류 기준

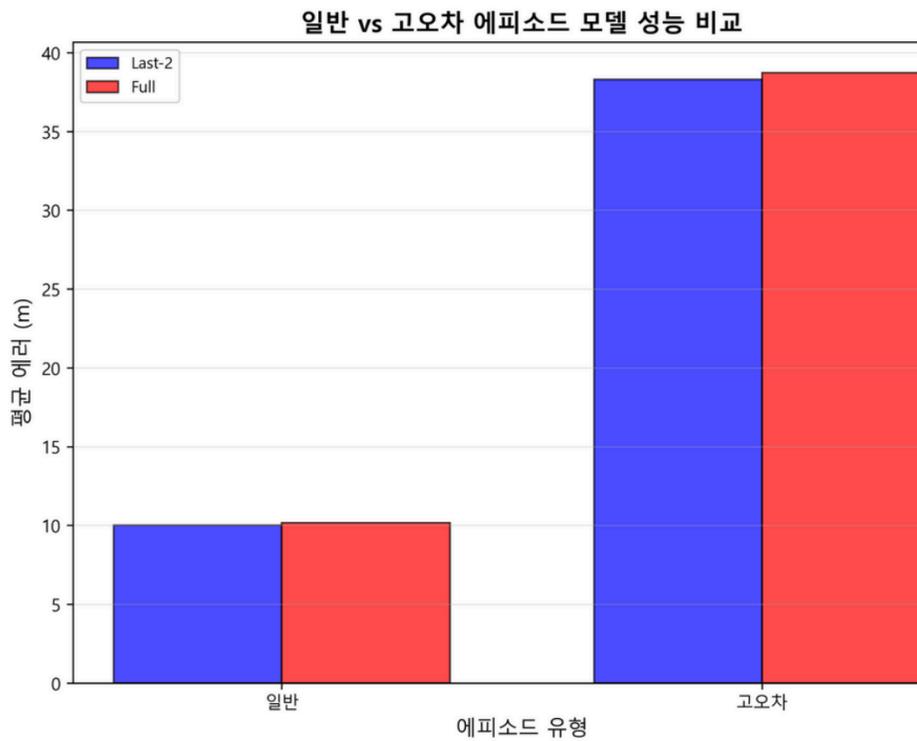
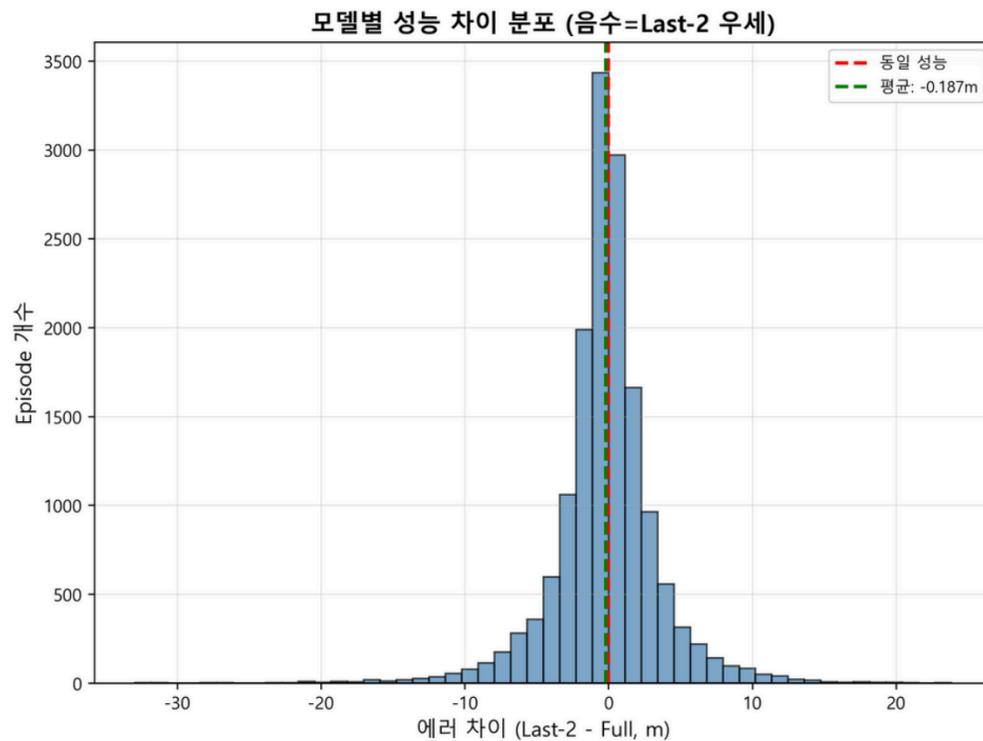
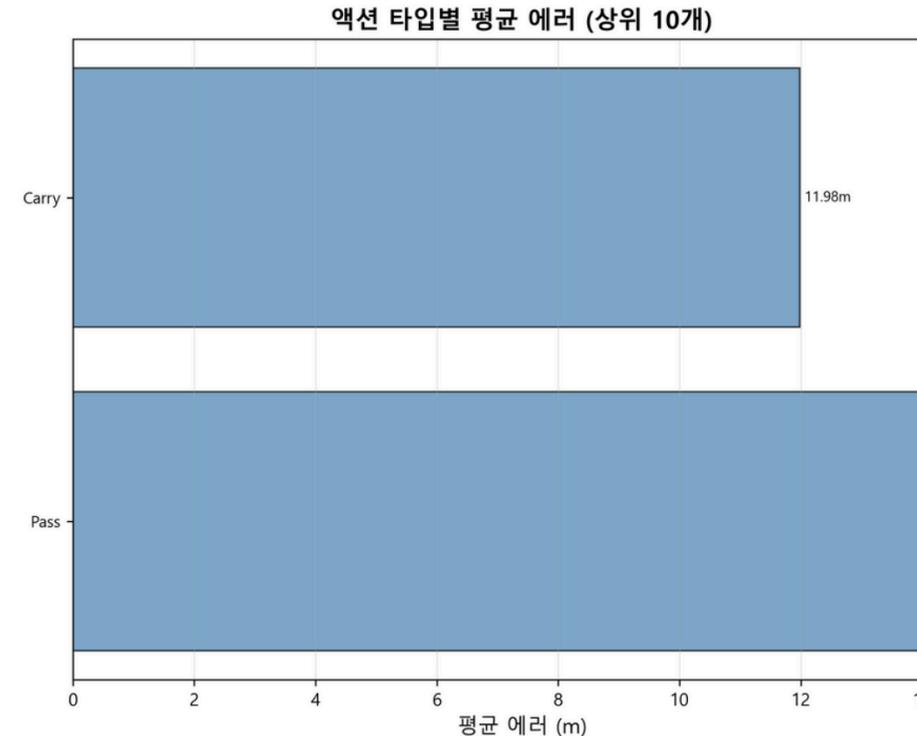
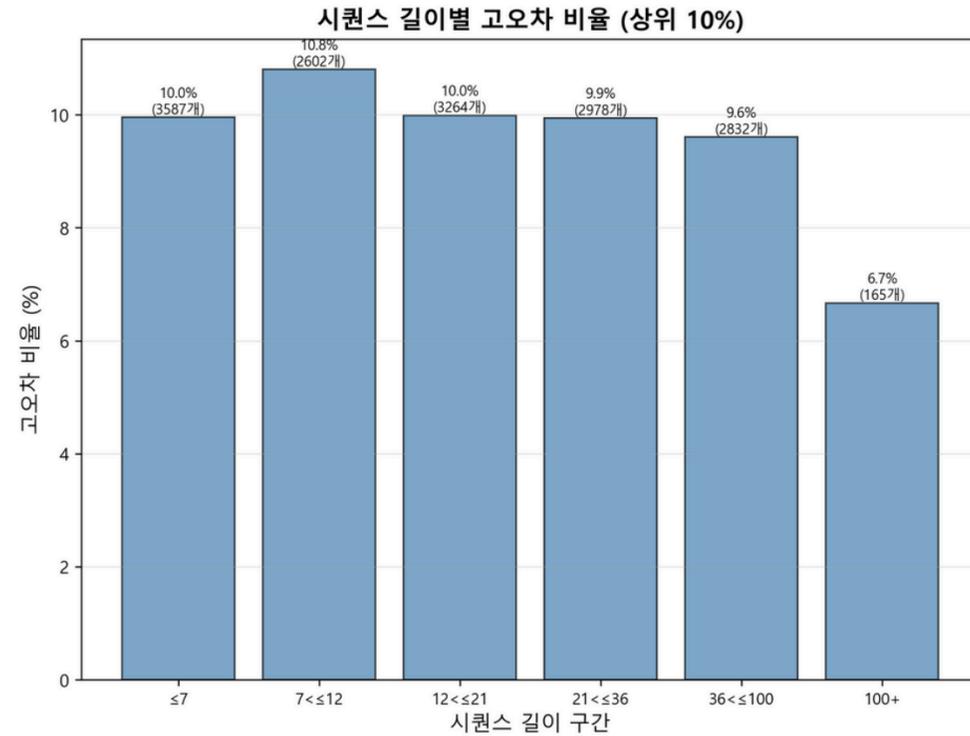
- OOF 오차 상위 10%를 고오차 에피소드로 정의 (임계값: 30.77m 이상, 총 1,543개)

• 핵심 발견

- 시퀀스 길이와 고오차 발생 비율
 - 단순히 시퀀스가 길다고 고오차 비율이 높은 것은 아님 → 특정 길이 구간에서 예측 난이도가 높음
- 액션 타입별 에러 기여도
 - Pass 액션: 평균 에러 14.1m로 가장 높은 에러 유발
 - Carry 액션: 평균 에러 12m로 두 번째로 높음

• 결론

- 오차가 큰 에피소드는 특정 플레이스타일(특히 수비/장거리 성향) 선수의 등장 빈도가 높음
- 고오차 에피소드에서 두 모델 모두 성능이 크게 저하되므로, Dynamic Weighting 및 고오차 패턴 감지 전략이 필요함
 - Player Style Feature와 고오차 대응 전략이 필요해짐



💡 **핵심 인사이트: 평균 오차 개선을 넘어, 극단적인 고오차 Episode(상위 10% 이상)의 특성을 파악하고 이를 해결하는 것이 LB 향상의 핵심**

초반: 기본 모델 구축 및 안정화

- 모델 선택: **LSTM 단일모델 선택**

- 초기 실험: 다양한 모델 시도

- 트리 모델 (XGBoost, LightGBM, CatBoost): 시도했으나 효과 없음
- GRU: 시도했으나 LSTM 대비 성능 저하
- 다른 딥러닝 모델 (Transformer, CNN 등): 시도했으나 효과 없음
- **최종 선택: LSTM**

- 트리 모델의 한계

- 시퀀스의 시간적 의존성을 직접 모델링하기 어려움
- Episode별 시퀀스 길이 차이를 효과적으로 처리하기 어려움
- 액션 간 순서와 맥락 정보를 충분히 활용하지 못함

- GRU의 한계

- GRU는 LSTM보다 단순한 구조로 장기 의존성 학습이 약함
- 긴 시퀀스(최대 270개 액션)에서 초기 정보를 기억하는 능력이 LSTM이 더 우수
- 실험 결과 LSTM이 GRU보다 일관되게 우수한 성능

- 다른 딥러닝 모델의 한계

- Transformer: 데이터 규모가 상대적으로 작아 오버피팅 위험, 긴 시퀀스에서 계산 비용 큼, CV가 개선되지 않았고 악화됨
- CNN: 시퀀스 길이가 가변적이라 고정 크기 필터 적용이 어려움
- 단순 MLP: 시퀀스의 시간적 구조를 전혀 반영하지 못함

- LSTM의 장점

- 장기 의존성 학습: Cell state를 통해 초기 액션 정보도 유지 가능
- 가변 길이 처리: 패딩 없이 다양한 길이의 시퀀스 직접 처리
- 시간적 패턴 학습: 액션 간 순서와 맥락을 효과적으로 학습
- 안정적 학습: Gradient vanishing 문제를 어느 정도 완화

- LSTM의 한계 및 해결책: Last-K 전략

- 한계: 긴 시퀀스(최대 270개 액션)에서 Gradient vanishing 문제 발생
- 해결책: Last-K 전략 도입 - 마지막 K개 timestep만 사용하여 최근 정보에 집중
- 긴 시퀀스의 초기 정보가 노이즈화되는 문제를 회피하고, 최근 액션의 물리법칙에 집중

- 모델 앙상블 시도: 여러 모델을 앙상블했으나 오히려 성능 저하

- 패턴 불일치: 트리 모델은 시퀀스의 시간적 구조를 제대로 학습하지 못해 LSTM과 다른 패턴을 예측, 앙상블 시 오히려 혼란 가중
- 데이터 특성: 시퀀스 회귀 문제의 특성상 단일 아키텍처(LSTM)에 최적화하는 것이 여러 모델을 섞는 것보다 효과적
- 오버피팅 위험: Transformer 등 복잡한 모델은 데이터 규모 대비 과도하게 복잡하여 오버피팅, 앙상블 시 일반화 성능 저하

- 결론

- **시퀀스 회귀 문제**의 특성상 LSTM이 가장 적합한 아키텍처로 판단
- 이후 모든 개선은 **LSTM 기반으로 진행하여 일관성 유지**

초반: 기본 모델 구축 및 안정화

- **특징(feature) 엔지니어링**
 - **좌표/변위 (cur_x, cur_y, dx, dy)**
 - 공의 현재 위치와 이동 벡터는 패스 예측의 기본 정보
 - cur_x, cur_y: 현재 액션의 시작 좌표 (절대 위치 정보)
 - dx, dy: 액션별 이동 변위 (상대적 이동 정보, 패스 방향/거리 직접 반영)
 - **속도/각도 (speed, smooth_speed, angle_sin, angle_cos)**
 - 공의 움직임 속도와 방향은 물리법칙에 따라 패스 예측에 직접적 영향
 - speed: 순간 속도 (dx, dy로부터 계산)
 - smooth_speed: 이동 평균 속도 (노이즈 완화, 안정적인 속도 정보)
 - angle_sin, angle_cos: 이동 각도의 sin/cos 인코딩 (각도 경계값 문제 해결, 연속성 보장)
 - **Momentum (momentum_dx, momentum_dy)**
 - 최근 5개 액션의 평균 이동 벡터 계산, 관성 정보 제공
 - 선수의 움직임 추세와 방향성 파악에 유용
 - **액션 정보 (type_val, success_val)**
 - 액션 타입과 성공 여부는 패스 패턴에 큰 영향
 - type_val: 액션 타입 인코딩 (3-2 섹션 참조)
 - success_val: 액션 성공 여부 (성공한 패스와 실패한 패스의 패턴 차이 반영)
 - **골대 관계 (dist_to_goal, sin_angle_to_goal, cos_angle_to_goal)**
 - 골대와의 거리/각도는 공격 패턴과 패스 방향 결정에 중요
 - dist_to_goal: 골대까지의 거리 (공격 상황 파악)
 - sin_angle_to_goal, cos_angle_to_goal: 골대 방향 각도 (공격 방향성 정보)
 - **시간 정보 (dt)**
 - 액션 간 시간 간격은 시퀀스의 리듬과 속도감을 반영
 - 급격한 전환(짧은 dt)과 여유로운 전환(긴 dt)의 패턴 차이 존재
- **성능 향상을 이뤘던 피쳐**
 - **type_name 피쳐**
 - 액션 타입(type_name)에 따라 패스 패턴이 크게 다름 (Pass, Carry, Shot 등)
 - 각 액션 타입별로 trajectory pattern을 구분하여 학습 필요
 - 구현 방법
 - type_name을 LabelEncoder로 정수 인덱스로 변환
 - 정규화된 type_val (0~1 범위)를 피쳐로 사용
 - Embedding 레이어로 액션 타입별 패턴 학습
 - Type Embedding: 16차원 (hidden_dim // 8)
 - 각 액션 타입별로 고유한 trajectory pattern 학습 가능
 - **Time Delta (dt) 피쳐**
 - 액션 간 시간 간격이 패스 패턴에 영향을 미침
 - 급격한 전환(짧은 dt)과 여유로운 전환(긴 dt)의 패턴 차이 존재
 - 시간 정보를 활용하여 시퀀스의 리듬과 속도감을 모델에 제공
- **Robust Scaling**
 - 이상치 제거: 물리적으로 불가능한 속도/가속도 제거 (speed > 50m/s, acc > 50m/s²)
 - 99% 분위수 기반 스케일링으로 이상치에 강건한 정규화
 - Fold별 스케일 산출로 데이터 누출 방지 및 일반화 성능 향상

초반: 기본 모델 구축 및 안정화

- 성능 개선을 이뤘던 초기의 변화

- **Metric Loss 도입: 평가 지표와 동일한 Loss 함수**
 - 초기 방식: SmoothL1Loss 등 일반적인 회귀 Loss
 - Loss와 실제 평가 지표(유클리드 거리)가 다름
 - 학습 방향과 평가 방향의 불일치 가능성
 - 개선 방식: **Metric Loss (유클리드 거리 직접 계산)**
 - 리더보드 평가 지표와 동일한 Loss 함수 사용
 - 예측값과 실제값을 역변환하여 실제 좌표로 변환
 - 유클리드 거리를 직접 계산하여 학습

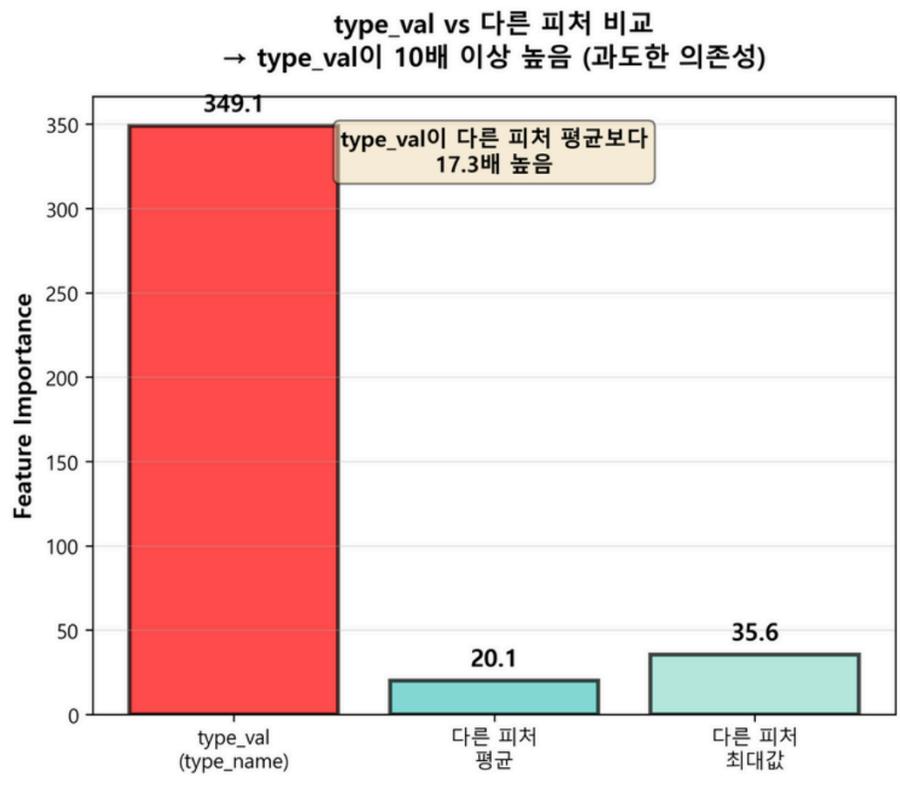
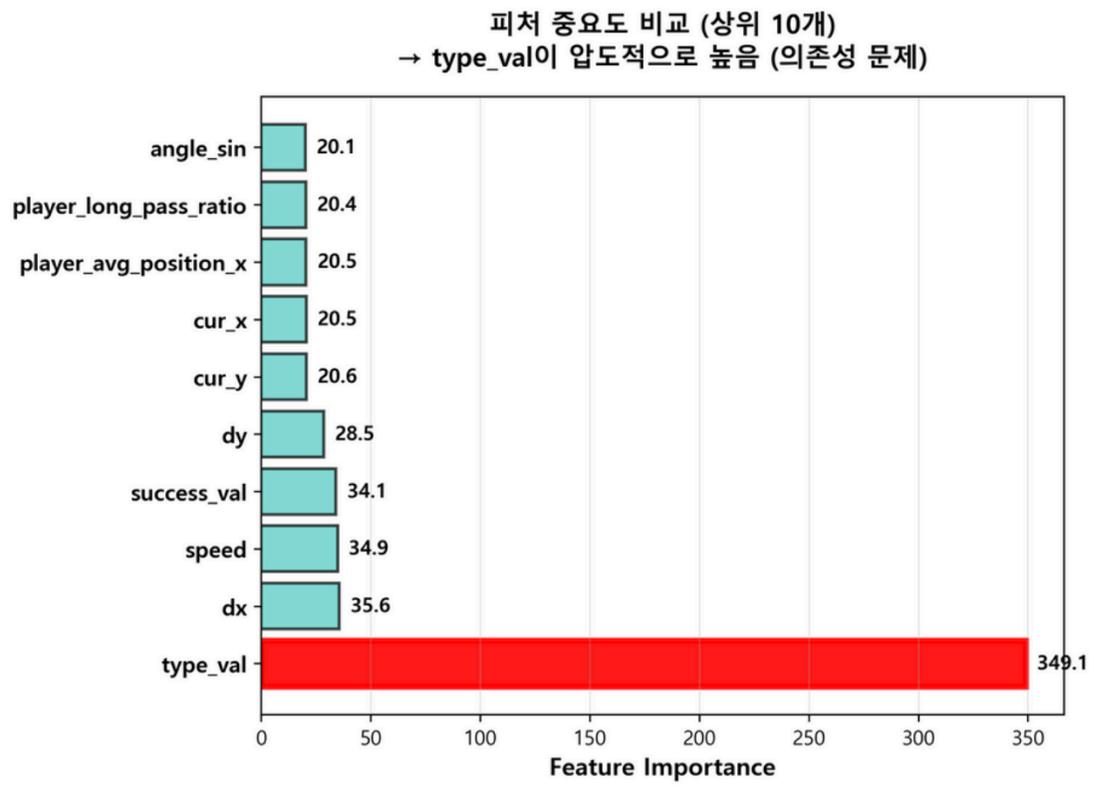
- 타깃 예측 방식 변경

- 초기 방식: (dist, angle) 예측
 - angle 오차가 거리 오차로 증폭
 - 경계값(y=0 등)에서 불안정
 - 수치적 안정성 문제
- 개선 방식: **(dx, dy) 직접 예측**
 - 선형성이 높아 학습이 안정적
 - 각도 오차 증폭이 없음
 - 평가 지표(유클리드 거리)에 직접 대응

- Residual Prediction 도입: type_name 의존성 완화

- **type_name embedding이 예측에 과도한 영향(타입에만 의존)**
 - type_name별 평균값을 먼저 만들고, 모델은 잔차(Residual)만 예측
 - 실제값 = type_name 평균 + Residual
 - 모델은 Residual만 학습 → type_name 의존성 완화
 - 추가 완화
 - Type Embedding 차원 축소(32 → 16)
 - Type Embedding Dropout(0.2) 추가
- 분석 원리: 학습된 모델의 가중치(Weight) 기반 중요도 계산
 - **type_val(type_name embedding)의 중요도가 349.1로 압도적으로 높음**

Residual Prediction 도입 근거: type_name 피쳐(type_val) 과도한 의존성

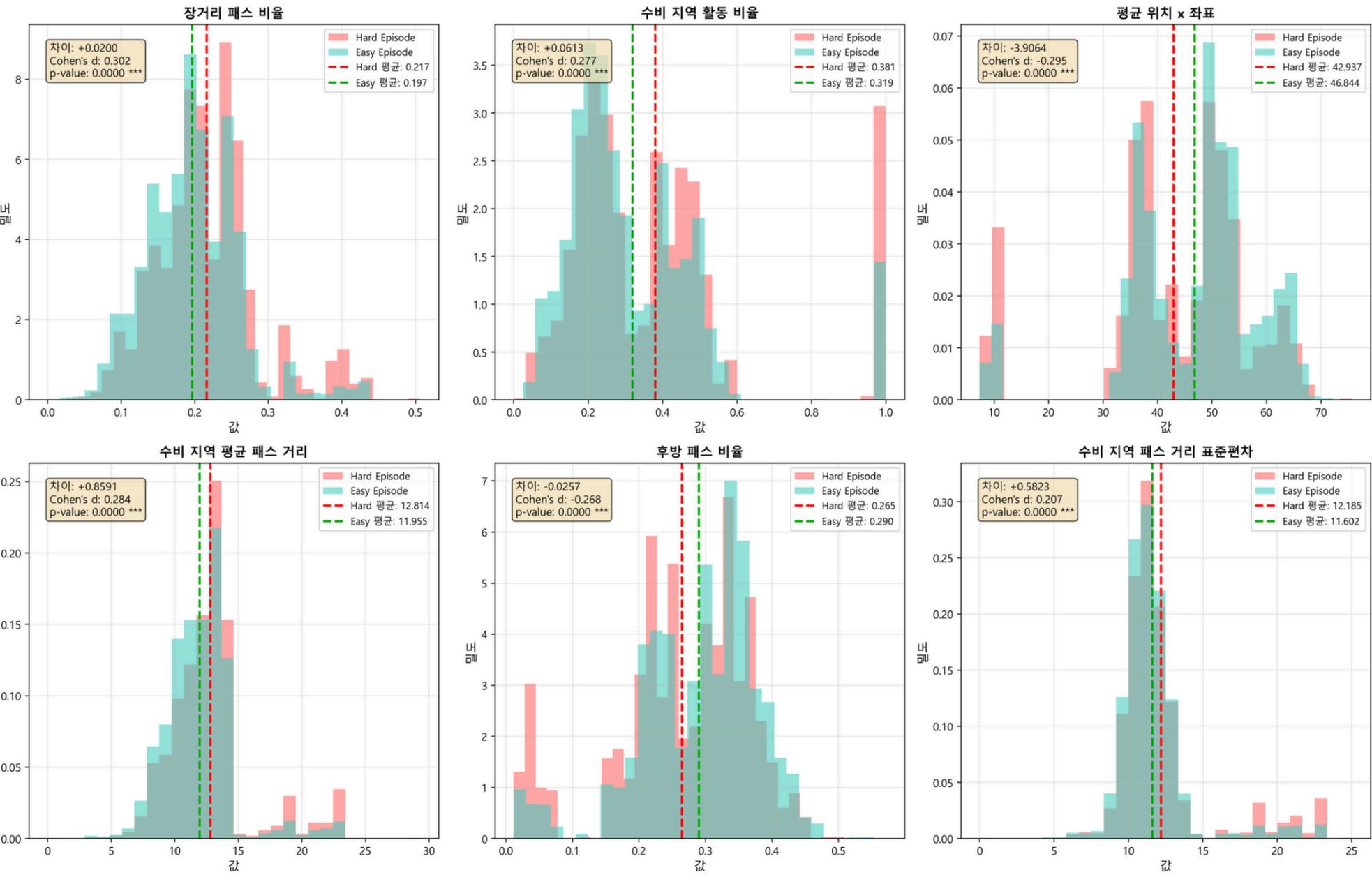


중반: 핵심 개선(피쳐/전략) 도입

- OOF 기반 문제 정의: Hard Episode 대응방안 마련
 - 고오차 Episode가 성능을 끌어내림
 - Hard Episode는 player/style 차이가 뚜렷
 - 긴 시퀀스에서는 초기 정보가 노이즈가 될 수 있음
 - 변칙 패턴(Zigzag/Long-pass)이 최근 구간에 집중되는 경향
- Player Style Feature 추가(EDA 기반 선별)
 - Episode 내부 정보만으로는 선수 성향을 충분히 표현하기 어려움
 - OOF 분석 결과, 오차가 큰 에피소드는 특정 플레이스타일(수비/장거리 성향) 선수의 등장 빈도가 높음
 - 같은 패스 상황이라도 "누가 패스하는지"에 따라 패턴이 다를 수 있음
 - 도출 과정
 - OOF 오차로 오차가 큰/작은 에피소드 분류
 - train 전체에서 선수별 통계 산출, t-test + Cohen's d로 유의성/효과 검정
 - $p < 0.05$ 및 $d > 0.2$ 기준으로 피쳐 선별
 - 선별된 6개 피쳐 및 의미
 - player_long_pass_ratio: **선수의 장거리 패스 비율** (공격 전환 시 패스 거리 패턴 반영)
 - player_defensive_ratio: **선수의 수비 지역 활동 비율** (위치 성향 반영)
 - player_avg_position_x: **선수의 평균 위치 x 좌표** (전진/후방 성향)
 - player_defensive_zone_avg_pass_distance: **수비 지역에서의 평균 패스 거리** (수비 시 패스 패턴)
 - player_backward_pass_ratio: **후방 패스 비율** (공격/수비 전환 패턴)
 - player_defensive_zone_pass_distance_std: **수비 지역 패스 거리 표준편차** (패스 다양성)
 - 효과
 - **오차가 큰 에피소드** 강건성 향상(롱패스/수비 성향 선수를 구분 가능)
 - 통계적 근거로 피쳐를 제한하여 노이즈 최소화
 - **"누가 패스하는지"** 맥락을 모델에 주입하여 개인별 패턴 학습 가능

중반: 핵심 개선(피쳐/전략) 도입

Player Style Feature 선택 근거: Hard vs Easy Episode 분포 비교 (선별된 6개 피쳐)



- 장거리 패스 비율:** Hard Episode에서 평균 0.217 vs Easy Episode 0.197 (차이 +0.020, Cohen's d=0.302)
 - 수비 지역 활동 비율: Hard Episode에서 평균 0.381 vs Easy Episode 0.319 (차이 +0.061, Cohen's d=0.277)
 - 평균 위치 x 좌표: Hard Episode에서 평균 42.94 vs Easy Episode 46.84 (차이 -3.91, Cohen's d=-0.295)
 - 수비 지역 평균 패스 거리: Hard Episode에서 평균 12.81m vs Easy Episode 11.96m (차이 +0.86m, Cohen's d=0.284)
 - 후방 패스 비율: Hard Episode에서 평균 0.265 vs Easy Episode 0.290 (차이 -0.026, Cohen's d=-0.268)
 - 수비 지역 패스 거리 표준편차: Hard Episode에서 평균 12.19 vs Easy Episode 11.60 (차이 +0.58, Cohen's d=0.207)
- 결론:** Hard Episode는 수비 성향/장거리 패스 성향 선수와 연관되어 있으며, 이 6개 피쳐가 Hard Episode를 구분하는 데 효과적임을 통계적으로 입증

중반: 핵심 개선(피처/전략) 도입

• Last-K 전략 도입(모델 다양성 확보)

- OOF 에러 분석 결과, 어려운 에피소드(고오차 에피소드)에서는 전체 시퀀스가 오히려 노이즈가 됨
- 고오차 패턴(Zigzag, Long-pass)은 최근 몇 개 액션에서만 드러나는 경우가 많음
- **마지막 몇 스텝의 물리법칙**(속도, 각도, momentum 등)이 최종 패스 예측에 더 직접적으로 중요함을 확인
- Last-2 모델이 전체 에피소드의 53.5%에서 Full 모델보다 우세
→ 최근 액션 집중의 효과성 입증
 - 실험 흐름
 - 초기: last5, last10, last15, full
 - 중기: last2, last3, last4, last5, full
 - 최종: **last2 + full (단순화가 오히려 일반화에 유리)**
 - 최종 모델 타입
 - Last-2: 마지막 2개 timestep만 사용(Epoch 25)
 - Full: 전체 시퀀스 사용(Epoch 15)

• 데이터 증강 도입(학습 + TTA)

- 축구장은 좌우 대칭 구조이므로 y축 반전이 물리적으로 타당함
 - 데이터 부족 문제 완화 및 모델 일반화 성능 향상
 - 학습 시 다양한 패턴 노출로 과적합 방지
 - Y축 반전(Horizontal Flip)
 - 축구장 좌우 대칭 구조 활용
 - 변환 공식: $y' = 68.0 - y$ (필드 높이 68m 기준)
 - 모든 y 좌표 관련 피처에 적용: cur_y, dy, angle_sin, cos_angle_to_goal 등
 - 학습 시 증강
 - 원본 Episode + 증강 Episode 학습(데이터 2배)
 - 각 에피소드마다 원본과 y축 반전 버전을 모두 학습에 사용
 - 모델이 좌우 대칭 패턴을 학습하여 일반화 성능 향상
- ## • 추론 시 TTA (Test Time Augmentation)
- 원본 예측 + 증강 예측 평균
 - 증강 예측은 y를 다시 반전해 원 좌표계로 복원 후 평균
 - TTA에 따른 효과
 - 데이터 다양성 증가 (학습 데이터 2배)
 - 예측 안정성/일반화 향상 (TTA로 예측 분산 감소)
 - 좌우 대칭 패턴 학습으로 모델 강건성 향상

종반: 최적화 및 앙상블 전략(추론 로직 완성)

- **Last-K 앙상블 최적화(단순화의 이득)**

- 최종 구성
 - 모델 타입: last2, full (2개)
 - 복잡한 다모델(5개)보다 단순한 2모델이 오히려 일반화에 유리
 - 계산 효율과 재현성이 개선됨

- **OOF 기반 Dynamic Weighting(Ridge Stacking 대체)**

- 초기 시도
 - Ridge Regression Stacking 시도
 - 한계: 길이/패턴에 따른 비선형 최적 비율을 반영하기 어려움
 - 고정된 가중치로는 Episode별 특성을 반영하기 어려움
- 개선 전략: Dynamic Weighting
 - OOF 분석 결과, 시퀀스 길이와 패턴에 따라 Last-2와 Full 모델의 최적 비율이 다름
 - OOF로 "길이별 최적 비율"을 먼저 추정
 - 추론 시 Episode 특성(길이, 패턴)에 맞게 Last-2 vs Full 비중을 동적으로 적용
 - 단순 평균보다 Episode별로 최적화된 앙상블 가능
- 기본 5구간 비율(train 기반)
 - 매우 짧음(≤ 5): Last-2 70%, Full 30% (짧은 시퀀스는 최근 정보가 더 중요)
 - 짧음(≤ 10): Last-2 55%, Full 45%
 - 중간(≤ 15): Last-2 50%, Full 50% (균형)
 - 김(≤ 23): Last-2 60%, Full 40% (긴 시퀀스는 Last-2가 더 유리)
 - 매우 김(> 23): Last-2 65%, Full 35% (매우 긴 시퀀스는 초기 정보 노이즈화)
- 효과
 - Episode별 최적 앙상블 비율 적용으로 성능 향상
 - OOF 기반 데이터 주도적 접근으로 해석 가능성 확보

- **Cluster-based Risk Assessment(고오차 패턴 감지)**

- OOF 분석 결과, **특정 패턴(Zigzag, Long-pass)**에서 고오차 발생 빈도가 높음
- 고오차 패턴은 최근 몇 개 액션에서만 드러나는 경우가 많아 Last-2 모델이 더 유리
- 패턴 기반 동적 가중치 보정으로 고오차 에피소드 대응

- **고오차 에피소드에서 Last-2가 유리한 이유**

- 정보 희석: 긴 시퀀스에서 초기 액션이 hidden state에 누적되어 최근 정보가 희석될 수 있음
- 물리법칙: 마지막 몇 스텝의 물리법칙(속도, 각도, momentum)이 최종 패스 예측에 더 직접적

- **최종 앙상블 구성(다층)**

- 모델 타입: 2가지(Last-2, Full)
- Fold 앙상블: 5-Fold
- 시드 앙상블: 3개 시드 [42, 123, 456]
- TTA: 원본 + 증강
- **총 예측 횟수: $2 \times 5 \times 3 \times 2 = 60$ 회**

실험 결과 및 성능 개선 요약

| 개선 사항 | 방법 | 효과 |
|----------------------|--|---------------------|
| Type Dependence 완화 | Residual + type embedding 축소 + dropout | 타입 의존 감소, 세밀한 패턴 학습 |
| Player Style Feature | EDA 기반 6개 선별 피쳐 | 오차가 큰 에피소드 강건성 향상 |
| Last-K 전략 | last2 + full로 단순화 | 일반화/효율 상승 |
| 데이터 증강 | Y축 반전 + TTA | 안정성/일반화 향상 |
| 고오차 대응 | Zigzag/Long-pass/긴 시퀀스 감지 | 롱테일 개선 |
| Dynamic Weighting | OOE 기반 비율 적용 | Episode별 최적 앙상블 |
| Robust Scaling | 99% 분위수 + 이상치 제거 | 스케일 안정화, 누출 방지 |

결론 및 부록:상세 수치 및 설정

• 핵심 기여(요약)

- OOF 분석으로 고오차 Episode(롱테일)를 정확히 규정
- Player Style Feature로 “누가 패스하는지” 맥락을 주입
- last2 + full 조합으로 시퀀스 길이/패턴에 따른 상보성 확보
- Dynamic Weighting + Risk Assessment로 오차가 큰 에피소드를 정밀 공략
- 증강(TTA)과 Robust Scaling으로 일반화/안정성을 보강

• 실용적 가치

- 재현 가능(전략/하이퍼파라미터가 코드와 문서에 명시)
- 다른 시퀀스 회귀 문제에도 확장 가능
- EDA/OOF 기반으로 개선 근거에 대한 해석 및 설명 가능

• 부록: 상세 수치 및 설정

- 모델 구조
 - Input dim: 24 (실제 사용 20개)
 - Hidden dim: 128
 - Layers: 2
 - Type Embedding: 16차원 + Dropout 0.2
 - Success Embedding: 16차원
 - Feature Embedding: 96차원
 - Output dim: 2 (dx_residual_norm, dy_residual_norm)

◦ 학습 설정

- Batch Size: 64
- Learning Rate: 1e-3
- Epochs: Last-2 25, Full 15
- Dropout: 0.2
- CV: 5-Fold GroupKFold (game_id 기준)
- 시드 앙상블: 3개 [42, 123, 456]

◦ 피쳐 구성(요약)

- 좌표/변위: cur_x, cur_y, dx, dy
- 속도/각도: speed, smooth_speed, angle_sin, angle_cos
- Momentum: momentum_dx, momentum_dy
- 액션: type_val, success_val
- 골대 관계: dist_to_goal, sin_angle_to_goal, cos_angle_to_goal
- 시간: dt
- Player Style(6개): long_pass_ratio, defensive_ratio, avg_position_x, defensive_zone_avg_pass_distance, backward_pass_ratio, defensive_zone_pass_distance_std

◦ Robust Scaling

- 이상치 제거: speed > 50m/s, acc > 50m/s²
- 통계: 이상치 제거 후 99% 분위수
- Fold별 계산(누출 방지)

감사합니다.

