IMEN574/IDSC574 Programing for Data Science

Bayesian Conditional Trajectory Prediction

Forecasting ball-in-air player movement using only pre-pass tracking data for the NFL Big Data Bowl 2026.

Prize Pool: $50K
Prediction Track

Evaluation Target:
2025 Season



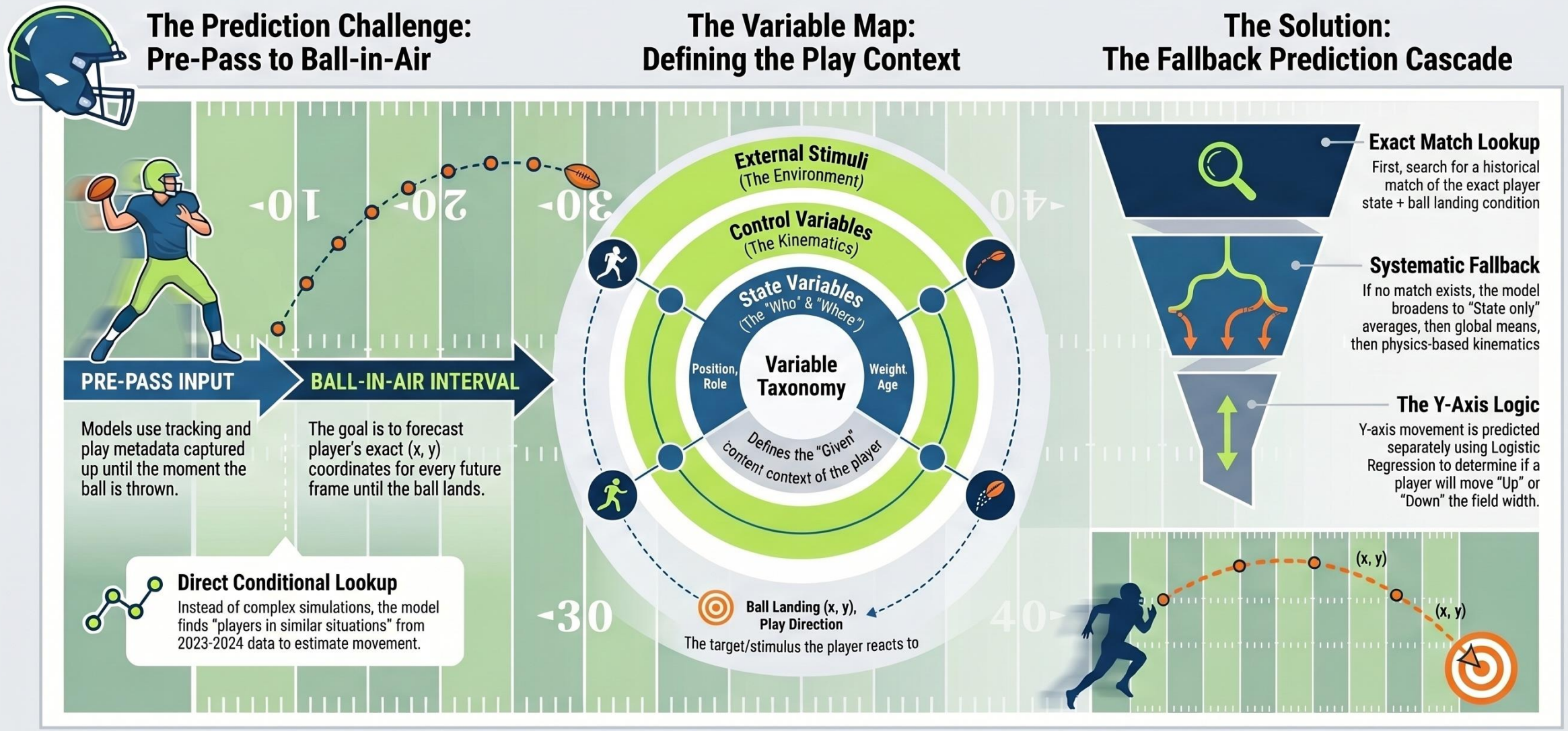
Jeongwoon Kim
Jaeyoon Shin
Seungjun Choi
Seungah Lee

Contents

1. Problem Definition
2. Data Overview
3. Feature Design & Model Architecture Overview
4. Core Methods
5. Results & Evaluation Metrics
6. Kaggle Score
7. Diagnosis
8. Conclusion

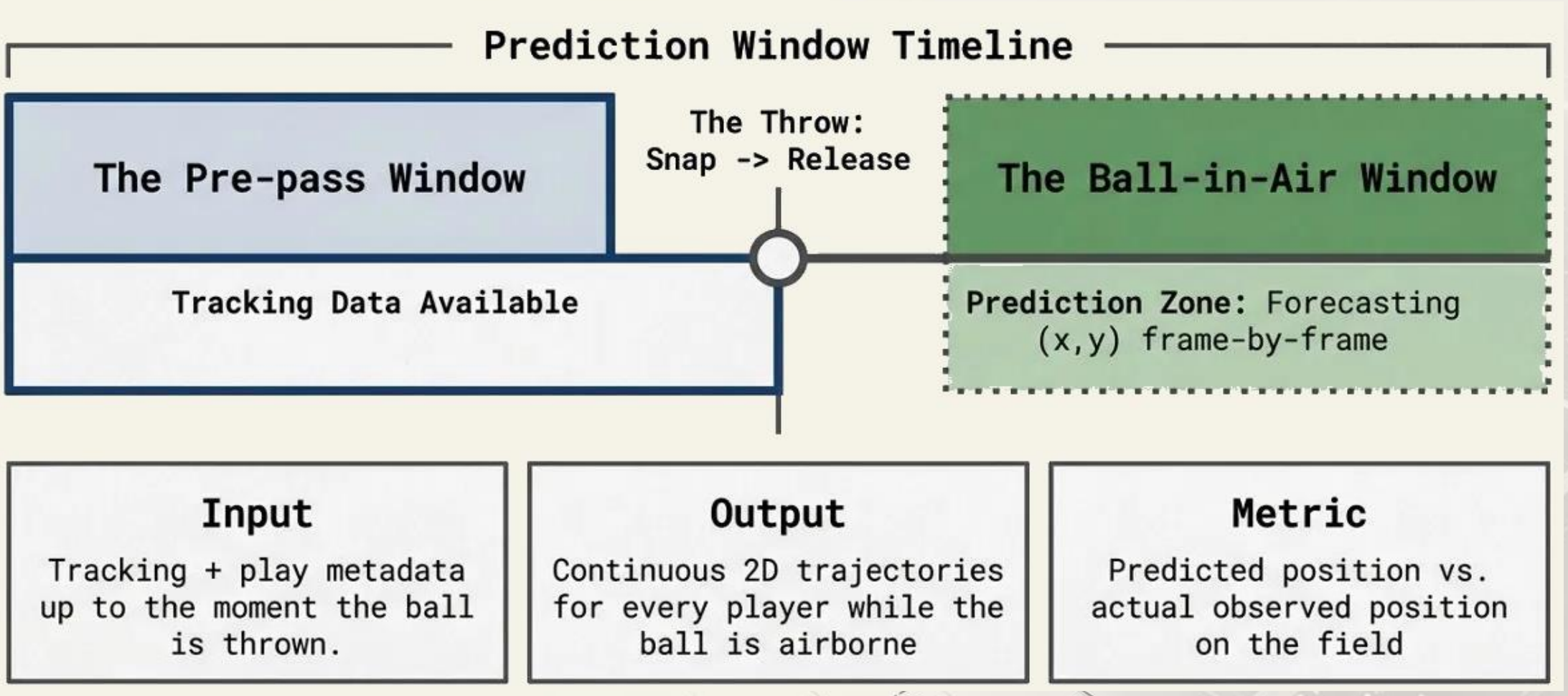


Problem Definition



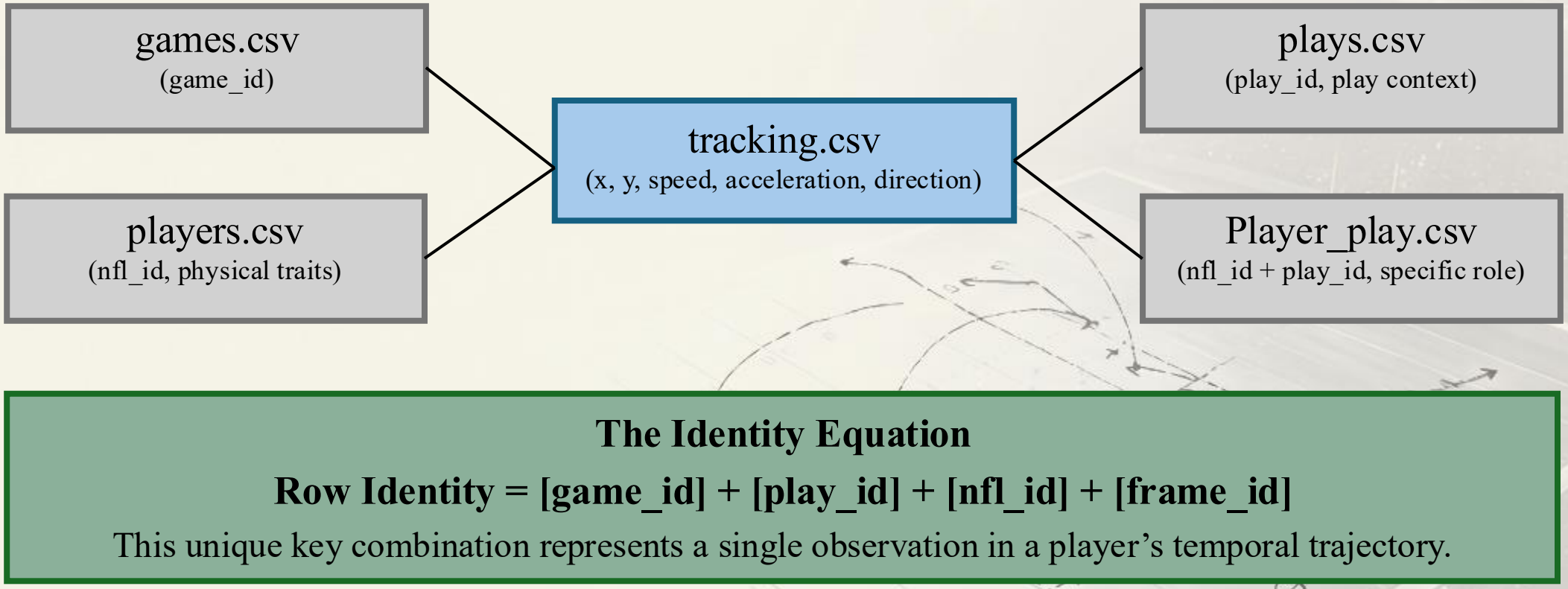
Data Overview

- Predicting ball-in-air player locations demands accurate forecasting based strictly on pre-pass tracking data.



Data Overview

- The data pipeline merges multi-level metadata into a single, unified player-frame timeline.



Feature Design & Variable Taxonomy

- Player movement is modeled as a kinematic reaction to external play stimuli, constrained by given state conditions.

State Variables: Given Context

- player_position
- player_side
- player_role
- player_weight
- x
- y

- Treated as fixed inputs, not probabilities.

External Variables: Stimulus

- play_direction
- ball_land_x/y
- absoulte_yardline_number

- The exogenous context the player reacts to.

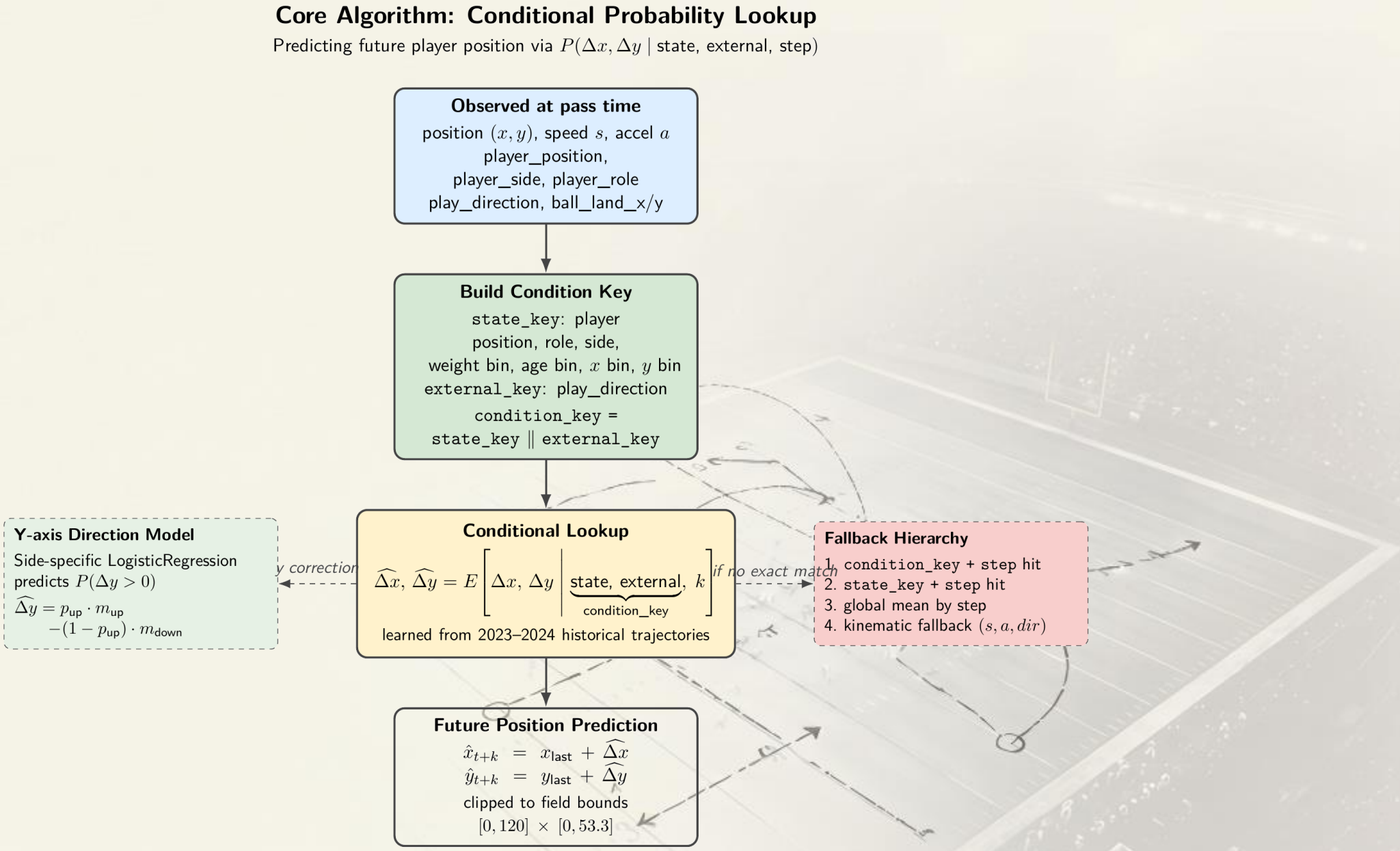
Control Variables: Reaction

- s (speed)
- a (acceleration)
- o (orientation)
- dir (direction)

- The observed kinematic adjustments.

Future Trajectory (Δx, Δy)

Model Architecture Overview



Core Method: Direct Conditional Lookup

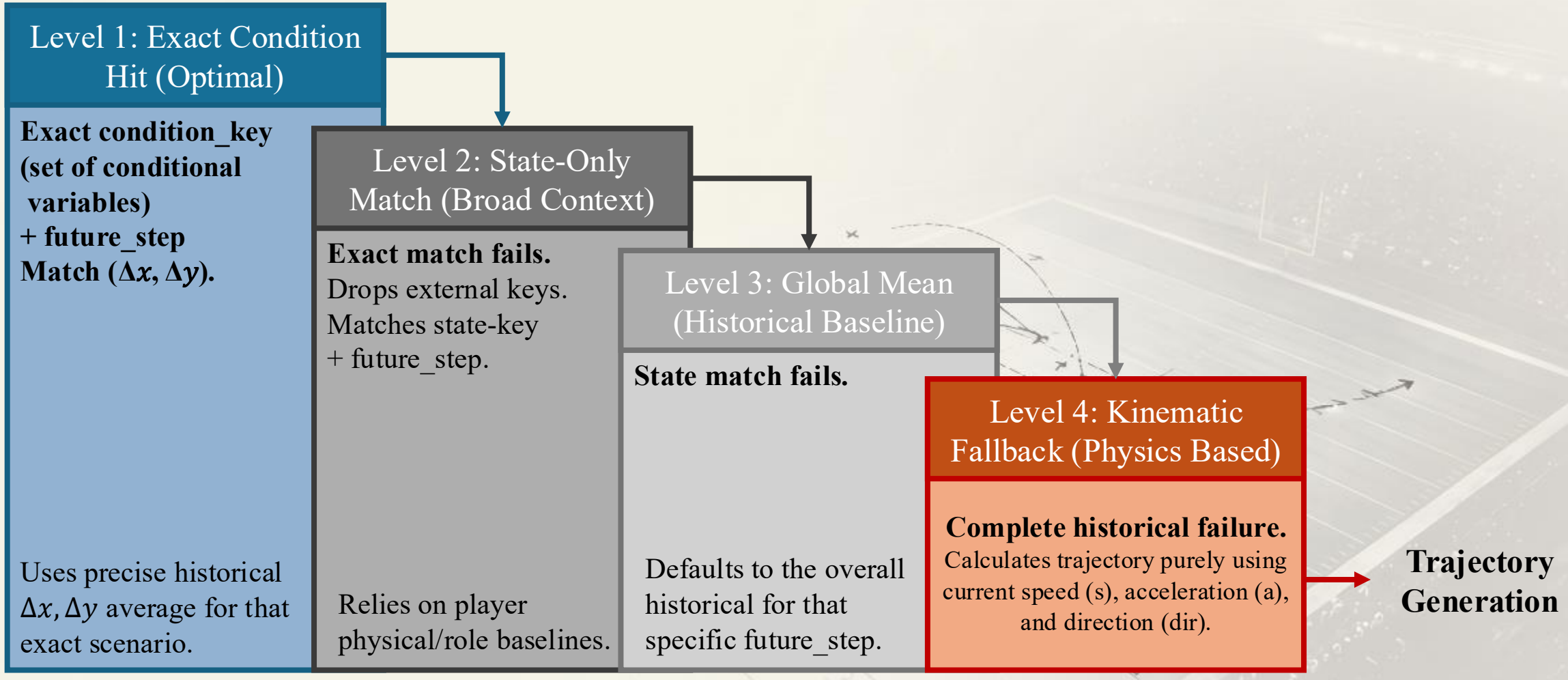
- Shifting to the current architecture eliminates posterior estimation in favor of highly efficient direct conditional lookups.

	v1~v5: Posterior Estimation	v6: Direct Lookup
State Treatment	Candidate states + posterior weighting (Treated as random variables)	Observed state treated strictly as a given variable
Probability Mapping	$P(\Delta \mid marginalized\ over\ states)$	$P(\Delta \mid \text{exact observed } state)$
Computational Cost	Expensive Kernal Similarity calculations	Instant, precise Direct Hash Lookup (condition_id)

Removing candidate filtering and similarity kernels drastically reduces inference latency while maintaining precision for the Kaggle API.

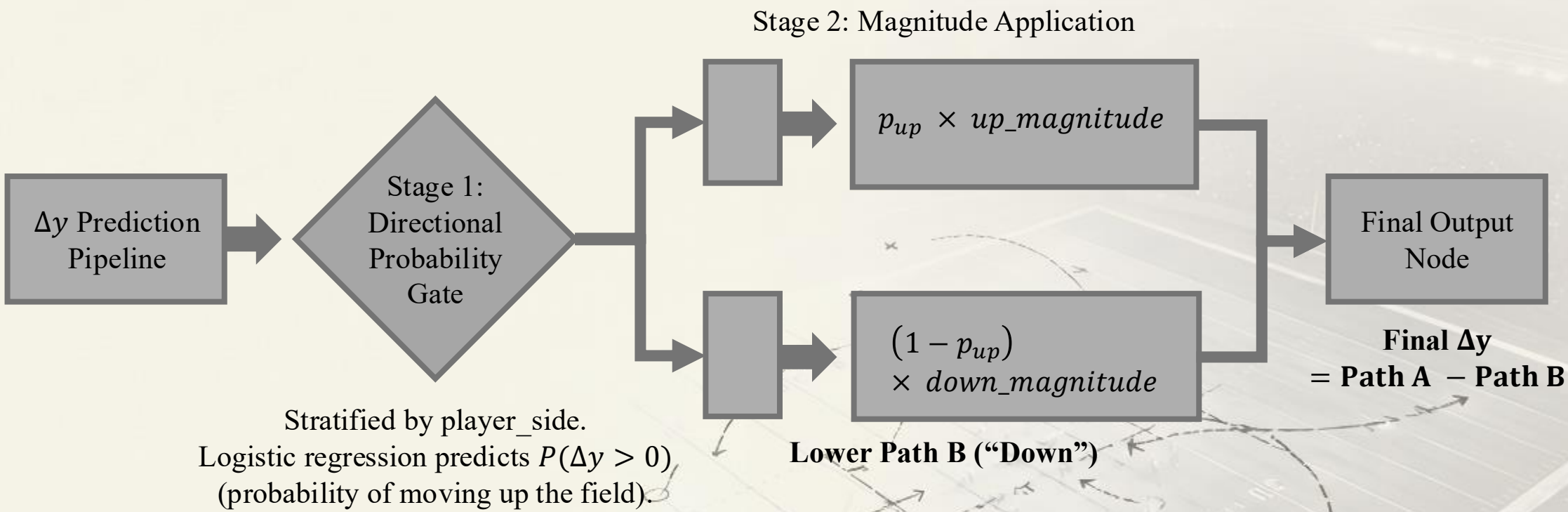
Fallback Cascade

- A four-step cascading fallback mechanism guarantees trajectory generation for unobserved edge cases.



y-axis Two-Stage Model

- A two-stage logistic regression isolates y-axis directionality to prevent zero-mean cancellation.



The Problem:
Field width (53.3 yards) causes standard historical averaging to cancel out movements (left and right movements average to zero.

Results & Evaluation Metrics

- x-axis (field length)

Metric	Value
R^2	0.927
RMSE	6.42 yards
MAE	4.40 yards

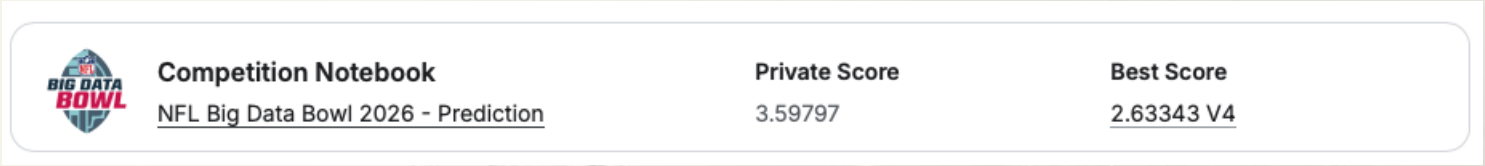
- Y-axis (field width)

Metric	Value
R^2	0.543
RMSE	7.87 yards
MAE	5.61 yards
Direction Accuracy	49.4%
Predicted “Up” Rate	5.0%

ADE	FDE	Skill vs. Persistence
7.87 yards	7.92 yards	-0.006

Kaggle Score

- Our ranking: 572nd place (723 participants)
- Top performers: score in the 0.4 range, heavily utilizing Transformer-based models
- Median performers: scores below 1.0, suggesting Transformer adoption is widespread



- Our Model in Context
  - Conditional probability lookup → interpretable but limited in capturing complex movement patterns
  - Transformer models explicitly model sequential dependencies and player interactions — key to top performance
  - **Our advantage:** significantly faster inference due to direct table lookup vs. heavy neural network computation

Diagnosis: Why Did the Model Underperform?

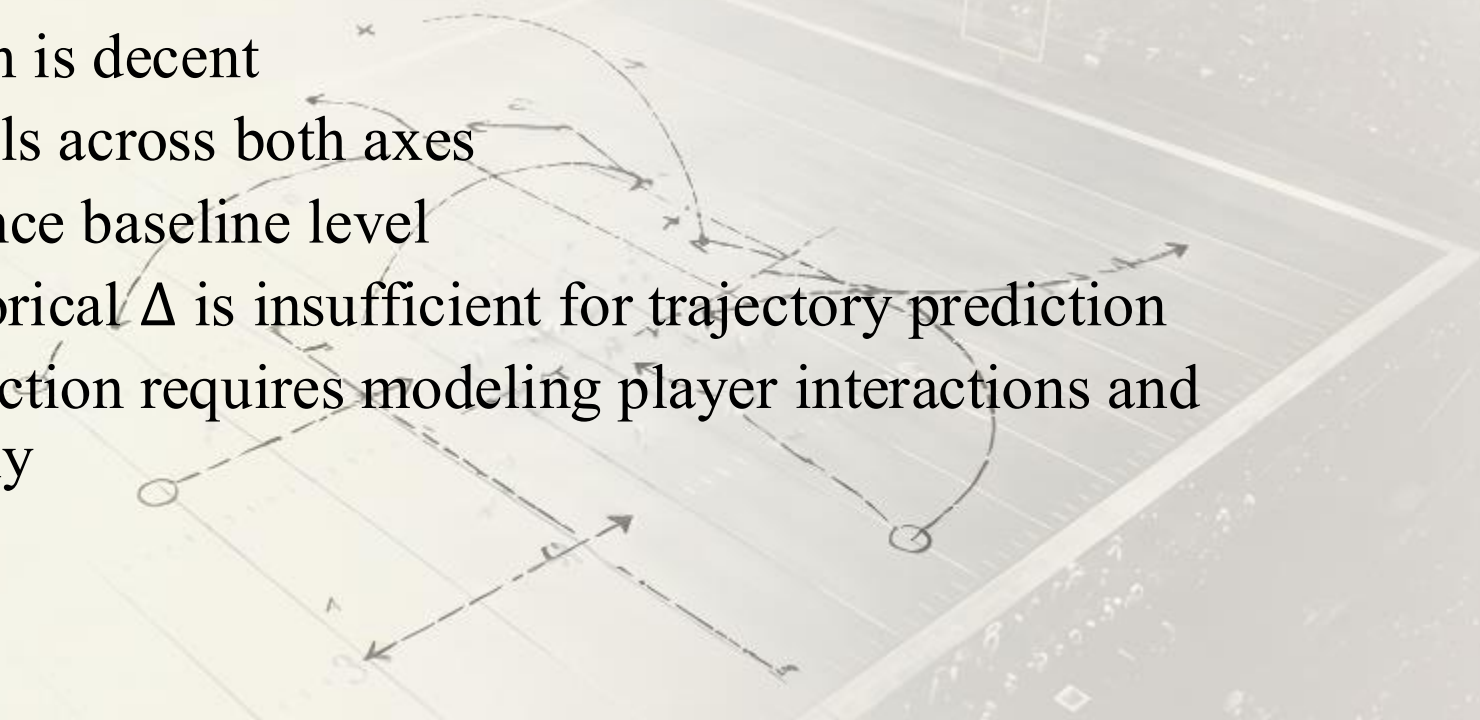
1. Averaging Effect
   - $\Delta x, \Delta y$ are averaged within each set of conditions
   - Opposing movements cancel out → predicted $\Delta \approx 0$
   - Model effectively predicts "no movement"
2. y-axis Logistic collapse
   - Predicted "Up" rate: 5.0% vs. actual 50.6%
   - y-axis directionality is not explained by position/role/side alone
   - Two-stage design failed to capture lateral movement signal
3. Sparse Lookup Table
   - 83,427 condition Keys (variables) × 55 future steps
   - Too few samples per cell → unreliable cell-level means
   - Fallback to global mean dominates prediction



Conclusion

- What We Built
  - Bayesian conditional trajectory predictor using pre-pass tracking data
  - Direct state lookup with 4-level fallback cascade
  - Side-specific two-stage y-axis model to address zero-mean cancellation

- What We Found & Learned
  - Absolute position prediction is decent
  - Displacement prediction fails across both axes
  - Model performs at persistence baseline level
  - Simple averaging over historical Δ is insufficient for trajectory prediction
  - Meaningful trajectory prediction requires modeling player interactions and temporal dynamics explicitly



Thank you

Bayesian Conditional Trajectory Prediction

