

Archetype Modeling v2 (Merged + Expanded from Archetype Modeling v1)

1. Purpose of Archetype Modeling

Archetype modeling categorizes offensive players, defenders, and defensive units into behavioral clusters that are more predictive of production and matchup outcomes than positions alone.

Instead of saying:

- “He’s a WR1 vs CB1”

we want:

- “He’s a *Boundary Deep Threat WR* vs a *Press-Man Boundary CB*”
- “He’s a *Slot Technician WR* vs a *Soft-Zone Nickel CB*”
- “He’s a *Pass-Catching RB* vs a *Poor Coverage LB Archetype*”

Archetypes improve:

✓ Player projections (statline + fantasy)

✓ Defender-aware adjustments

✓ Archetype-response submodels

✓ Matchup difficulty indexes

✓ Start/sit logic

✓ Waiver breakout analysis

✓ Team-level game predictions

✓ Explanation clarity (Ollama layer)

Phase 4 ML models, Phase 3 baseline, and Phase 6 UI/Explainability all rely on accurate archetype classification.

2. Archetype Types (v2 Expanded)

There are three distinct archetype systems, each trained separately:

2.1 Offensive Player Archetypes

These group offensive players by behavioral traits, not by position.

Example WR archetypes:

- Boundary Deep Threat
- Slot Technician
- YAC Monster
- Alpha X WR
- Field-Stretch Specialist
- Possession WR
- Gadget/YPRR Hybrid
- Red Zone Jump-Ball WR

Example TE archetypes:

- Route-Running TE
- Inline Blocker TE
- Move TE
- Seam-Stretch TE

Example RB archetypes:

- Workhorse Power RB
- Pass-Catching Space RB
- Explosive Perimeter RB
- Inside Zone Grinder

- Gadget RB / Hybrid WR-RB

Example QB archetypes:

- Deep Passing QB
 - Checkdown Distributor
 - Scramble-First QB
 - Play-Action Vertical QB
 - Game Manager
 - High-Volume Short-Pass QB
-

2.2 Individual Defender Archetypes (Huge v2 Addition)

Defender archetypes describe:

- coverage ability
- alignment usage
- man/zone preference
- ability to handle specific offensive archetypes

Example CB archetypes:

- Shadow Press-Man CB
- Boundary Press CB
- Off-Zone Boundary CB
- Slot Zone CB
- Hybrid Safety-CB Nickel Defender

Example Safety archetypes:

- Deep Middle Safety
- Split-Safety Zone Defender
- Box Safety
- Coverage Safety / TE Eraser
- Ballhawk Deep Safety

Example LB archetypes:

- Coverage LB
 - Run-Stopping LB
 - Hybrid Safety-LB Coverage LB
 - Missed-Tackle LB
 - Spy/Contain LB
-

2.3 Team-Defense Archetypes

These describe full defensive units.

Example:

- Man-Heavy Press Defense
- Split Safety Shell
- Single-High Heavy-Blitz Defense
- Soft Zone Bend-Don't-Break Defense
- Run-Funnel Defense
- Pass-Funnel Defense

These matter for:

- matchup difficulty
 - player projection
 - scoring projections
 - high-level game scripts
-

3. Input Features for Archetypes

Archetype modeling starts with features produced in Phase 2.

3.1 Offensive Player Archetype Input Features

From `player_game_features`:

- alignment:
 - slot_rate
 - boundary_rate
 - inline_rate (TE)
 - depth profile:
 - avg_depth_target
 - deep_target_rate
 - shallow_target_rate
 - YAC metrics:
 - yac_per_reception
 - broken tackle rate
 - efficiency:
 - yards_per_route_run
 - yards_per_target
 - catch_rate
 - explosion_rate (plays >20 yd)
 - usage:
 - target_share
 - carry_share (for hybrids)
 - red_zone_share
 - route profile:
 - crossing route %, slant %, out %, corner %, fade %, seam %, etc. (*use coarse bins if fine-grained unavailable*)
 - physical traits:
 - height, weight, speed (when available)
 - role metadata (Phase 1 import)
-

3.2 Defender Archetype Input Features

From `defender_game_stats`:

- alignment_usage:
 - boundary_pct
 - slot_pct
 - deep_pct
 - box_pct
- coverage_type:
 - man_coverage_pct
 - zone_coverage_pct
- performance:
 - yards_allowed_per_target
 - yac_allowed_per_reception
 - tds_allowed
- matchup deltas:
 - delta_def_yards_last3_vs_season
 - delta_def_targets_last3_vs_season
- penalties:
 - penalty_rate

- advanced:
 - EPA per coverage target (if available)
 - target separation allowed
 - success rate allowed

3.3 Team-Defense Archetype Input Features

- man_rate
 - zone_rate
 - blitz_rate
 - pressure_rate
 - run_success_rate_allowed
 - pass_success_rate_allowed
 - EPA allowed (run/pass)
 - explosive play rates (allowed)
 - missed tackle rate
 - alignment shells (single-high, split, quarters, etc.)
-

4. Preprocessing and Normalization

Before clustering:

1. Remove extremely low-sample players/defenders
 - threshold: ~50+ snaps for WR
 - threshold: ~100+ coverage snaps for CB
 - threshold: ~50+ carries for RB
2. Fill missing metrics using:
 - positional averages
 - archetype priors (for new players)
 - conservative fallback values
3. Normalize features:
4. $z = (\text{value} - \text{mean}) / \text{std}$

This helps avoid clustering distortion.

5. Multi-Stage Archetype Modeling (v2 Architecture)

Phase 1 identified raw stats.

Phase 2 produced engineered features.

Phase 3 baseline and Phase 4 ML depend on archetypes.

Thus, Phase 2 → Phase 4 need archetypes available ahead of training.

To do this correctly, v2 uses a multi-stage clustering process:

5.1 Stage 1 – Coarse Position-Aware Clustering

For WRs separately:

- Cluster WRs based on:
 - slot_rate
 - deep_rate
 - yprp
 - target_depth_profile
 - yac_profile

Use k-Means, Gaussian Mixture Models, or HDBSCAN.

For RBs separately:

- usage (rush vs receiving)
- explosive run rate

- yards after contact
- route rate

For TEs:

- inline rate
- route rate
- target depth
- YAC

For QBs:

- deep attempt percentage
- scramble rate
- aDOT
- pressure_to_sack ratio
- EPA per play

5.2 Stage 2 – Fine-Tuning Clustering within Coarse Groups

Within each coarse archetype:

- run additional clustering for refinement
- 3–6 subtypes per cluster
- incorporate continuous embeddings

Example:

- “Boundary WR” → subdivide into:
 - Alpha X
 - Deep Threat
 - Possession Boundary
 - YAC Boundary

5.3 Stage 3 – Archetype Embedding Generation (New in v2)

Once clusters are stable:

- Encode each archetype into a dense embedding vector (e.g. 8–32 dimensions)

Embedding generation options:

1. PCA
2. UMAP (manifold learning)
3. Autoencoder (small neural network)

These embeddings serve as:

- ML model inputs
- Ensemble model metadata
- Visualization tools

6. Defender Archetype Modeling (v2 Requirement)

Defenders must also be archetyped.

Procedure similar to offense:

- cluster CBs separately
- cluster safeties separately
- cluster LBs separately

Then run fine-tuning clustering:

- slot CBs → zone slot, man slot, hybrid slot
- boundary CBs → press-man, off-zone, trail, match-quarters
- safeties → box/down, deep-middle, split-field
- LBs → run-fit, coverage, hybrid

Generate defender embeddings.

7. Team-Defense Archetypes

Apply clustering to:

- blitz_rate
- man_rate
- zone_rate
- EPA_allowed
- explosive allowed
- pressure_rate
- alignment shell rates

Resulting clusters become:

- team-defense-archetype embeddings
 - features for ML models
 - features for explanations
-

8. Archetype Stability Across Seasons

Critical requirement:

Archetypes must not shift randomly year-over-year.

v2 includes:

- centroid tracking across seasons
- distance minimization to ensure continuity
- soft reinitialization if sample sizes change

Use constrained clustering:

new_centroid =

$$0.7 * \text{old_centroid} + 0.3 * \text{new_cluster_centroid}$$

This ensures:

- historical continuity
 - defender improvements/decline reflected without total reclassification
-

9. Archetype Interaction Matrices

This is one of the most important outputs.

9.1 Offensive vs Defensive Archetype Matrix

Example matrix (simplified):

Off Archetype Press-Man CB Zone CB Slot CB Safety Help

Deep WR	-12%	+5%	n/a	-8%
Slot WR	-2%	+6%	+15%	+3%
YAC WR	+8%	+2%	+4%	+1%
Possession WR	-1%	+2%	+6%	0%

This matrix is used:

- in baseline adjustments
 - as ML features
 - in ensemble model
 - in Phase 5 explanations
-

9.2 RB Archetypes vs Box Defenders

Example:

RB Archetype	Run-Stopping	LB Coverage	LB Hybrid	LB Box	Safety
Power RB	-10%	+8%	+4%	-3%	
Pass-Catching RB	+12%	-8%	+6%	+4%	

9.3 TE Archetypes vs Safety/LB Archetypes

Example:

TE Archetype	Box	Safety	Coverage	LB Hybrid	S/LB Deep	Safety
Move TE	-4%	+6%	+12%	-8%		
Inline TE	+2%	-6%	+1%	-8%		

These tables are built from empirical data.

10. Integration into ML and Baseline Models

10.1 Baseline Model

Uses archetype adjustments:

```
predicted_yards *= (1 + off_def_archetype_matrix[off_arch][def_arch])
```

10.2 ML Models

Use:

- one-hot archetype
- embedding vectors
- interaction features

10.3 Ensemble Model

Receives:

- archetype-aware adjustments
- archetype response submodel predictions

11. Archetype Updates in Season

Archetypes update:

- weekly for defenders
- monthly for offensive roles
- but only with enough new data

We track:

- cluster drift
- centroid movement
- new rookies assigned to nearest centroid

12. Implementation Outline (Step-by-Step)

Step 1: Load Phase 2 feature tables

Step 2: Filter eligible sample players/defenders

Step 3: Normalize features

Step 4: Run coarse clustering per position group

Step 5: Fine-tuning clustering

Step 6: Generate embeddings

Step 7: Assign stable cluster IDs

Step 8: Build interaction matrices

Step 9: Store archetypes in DB

Archetype tables:

- offensive_archetypes
- defender_archetypes
- team_defense_archetypes

- archetype_embeddings
 - archetype_interactions
-

13. Output Tables

1. offensive_player_archetypes
 - player_id
 - archetype_id
 - embedding
 - metadata_json
 2. defender_archetypes
 - defender_id
 - archetype_id
 - embedding
 3. team_defense_archetypes
 - team_id
 - archetype_id
 4. archetype_interactions
 - offensive_archetype
 - defender_archetype
 - effect_multiplier
-

14. Summary of Archetype Modeling v2

Archetype Modeling v2 adds:

- ✓ Full offensive archetype system
 - ✓ Full defender archetype system
 - ✓ Full team-defense archetype system
 - ✓ Multi-stage clustering
 - ✓ Continuous embeddings
 - ✓ Interaction matrices
 - ✓ Stability across seasons
 - ✓ Integration into ML, baseline, and ensemble
 - ✓ Required metadata for Ollama explanations
- This module becomes the behavioral intelligence layer powering both prediction accuracy and explainability.