

ok so i am thinking of designing a problem statement where we can check the compatibility between 2 players . Like designing a formula which uses graph based calculations to arrive at a conclusion how well a combination of 2 or more players fit. Is there some way?

ChatGPT said:

Thought for 21s

Absolutely—this is a great fit for graph thinking. Here's a concrete, ML-friendly framework to quantify **player compatibility** (for a pair or a small group) from StatsBomb Events. No code—just the blueprint and formulas/workflows you can implement.

Compatibility as a Graph Problem

Think of a match (or set of matches) as a **multi-relational, temporal player graph**:

- **Nodes:** players (with attributes like position, average locations, event rates). Positions and average pitch coordinates come straight from the event spec (e.g., `position`, `location`) .
- **Edges (typed):** passes, carries handoffs, press/duel chains, interceptions, etc., with features like success, distance, angle, start/end locations (`pass.recipient`, `pass.end_location`, `carry.end_location`, `interception.outcome`, etc.) .
- **Temporal aspect:** sequences by `timestamp/index` within possessions; possessions are explicitly tracked (`possession`, `possession_team`) .
- **Pressure context:** “under_pressure” and “counterpress” tags enable linking defensive actions to on-ball events within 5s windows .

We'll score a pair (A,B) with a **Compatibility Score** built from components that each measure a different kind of fit. You can compute per-match and aggregate/weight by minutes played together.

The Compatibility Score (for a pair A,B)

$\text{Compat}(A, B) = w_1\text{CQ} + w_2\text{CS} + w_3\text{SC} + w_4\text{RC} + w_5\text{DI} + w_6\text{TS}$

$w_1\text{CQ} + w_2\text{CS} + w_3\text{SC} + w_4\text{RC} + w_5\text{DI} + w_6\text{TS}$

Weights can be (a) learned to predict outcomes you care about (e.g., xG differential when both are on) or (b) set by domain judgement (equal or tuned). Each component below is on [0,1] after normalization.

1) CQ – Connection Quality (direct synergy)

How reliably and productively do they connect?

- **Graph view:** directed pass edges A→B and B→A with features: count, success rate, distance, angle, height, pressure context (from pass attributes and `under_pressure`) .
 - **Compute:**
 - **Quality:** $q_{A \rightarrow B} = \text{weighted success rate}_{A \rightarrow B}$ = weighted success rate $q_{A \rightarrow B} = \text{weighted success rate}$ (weight by pass difficulty—distance/angle/height—and pressure flags).
 - **Volume-normalized:** scale by opportunities (minutes together, proximity in average locations).
 - **Bidirectional CQ:** $CQ = q_{A \rightarrow B} + q_{B \rightarrow A} / 2$ $CQ = 2q_{A \rightarrow B} + q_{B \rightarrow A}$.
 - **Graph-ML twist:** train a simple **edge existence/quality model** (link prediction) from team graphs; **CQ is the residual** (actual – expected). Positive residual ⇒ they outperform what the system expects.
-

2) CS – Chain Synergy (indirect, sequence value)

Do A and B make each other more dangerous across sequences, not just direct passes?

- **Graph view:** a **possessions hypergraph** (nodes: players, hyperedges: possession sequences). Include pass+carry+dribble edges; shots as terminal nodes with `statsbomb_xg` (if available) or your xT proxy derived from shot/location features .
- **Compute:** for sequences that include both A and B in any order within N steps:
 - **Lift in value:** $\Delta = xT/xG$ of sequence–baseline xT/xG of similar sequences without that pair $\Delta = xT/xG$ of sequence – xT/xG of baseline xT/xG of similar sequences without that pair $\Delta = xT/xG$ of sequence–baseline xT/xG of

similar sequences without that pair.

- ○ **CS:** mean positive $\Delta\Delta$ clipped to [0,1].
 - **Notes:** Use `carry` and `dribble` events to credit ball progression between the pair even without a pass .
-

3) SC – Spatial Complementarity

Do their preferred spaces **overlap (bad)** or **interlock (good)**?

- **Graph view:** discretize the pitch into zones (as per spec's locations framework) and build a **player↔zone bipartite graph** weighted by time on ball / events per zone (using `location` and `end_location`) .
- **Compute:**
 - Build two zone distributions $PA(z), PB(z), P_A(z), P_B(z)$.
 - **Overlap penalty:** $OL = \sum z \min(PA(z), PB(z))$
 - **Adjacency bonus:** reward when their peaks are in **adjacent** lanes (facilitates triangles and safe passing lanes).
 - **SC:** $1 - OL + AdjacencyBonus$, normalized to [0,1].

4) RC – Role Complementarity

Do their on-ball profiles fit (e.g., creator + runner, progressor + finisher)?

- **Graph view:** per player, a **role vector** from event mix: pass types (`switch`, `cross`, `cut-back`, through-like “technique”), carries, dribbles, pressures, duels, interceptions, shot involvement; all available in event objects .
- **Compute:** learn **player embeddings** via a graph model (e.g., team pass graph with multi-relational edges). Define a **compatibility kernel** where *complementary* roles score higher than identical ones (e.g., via a learned bilinear form $eA^\top MeBe_A^\top M e_B A^\top MeB$). Start simple: cosine distance inverted + bonuses for creator→finisher

pairs (based on labels like frequent `shot-assist` vs `first_time` shots) .

- **RC:** scale to [0,1].
-

5) DI – Defensive Interlock

How well do they act as a **press-trap** pair or cover for each other?

- **Graph view:** chains linking **pressure** → **(opponent on-ball event)** → **recovery/interception** within the 5s “counterpress” window (available as `counterpress/under_pressure`) .
 - **Compute:** count A’s pressures leading to B’s recoveries/interceptions (and vice-versa), weighted by outcome quality (win ball high, create shot within k actions).
 - $DI = \frac{\text{success chains involving both}}{\text{all joint press chains}}$ $DI = \frac{\text{all joint press chains}}{\text{success chains involving both}}$ (minutes-adjusted).
 - **Include tackling/duel outcomes** (`duel.outcome`, `interception.outcome`) for success credit .
-

6) TS – Triangle Stability (third-man patterns)

Do they form robust **triangles** with others?

- **Graph view:** on the pass graph, compute **closed triads** that contain A and B; weight by edge quality and zone diversity.
 - **Compute:** clustering coefficient restricted to A,B subgraph neighborhood, plus a **third-man run** frequency (A→X→B within t seconds; include `carry` edges as A→X handoffs) .
 - **TS:** normalize by minutes/opportunities.
-

Workflow (end-to-end)

1. **Assemble on-pitch segments** where A and B both play. Slice events by `timestamp`, `period`, and player presence.
 2. **Build graphs per segment:**
 - o **Pass multigraph** (directed, weighted, with features like distance/angle/height/pressure) .
 - o **Possession hypergraphs** for chain synergy (include pass, carry, dribble, shot nodes/edges; shots have outcomes/xG) .
 - o **Press-chains graph** (pressure→opponent on-ball→recovery/interception within counterpress window) .
 - o **Zone bipartite** for spatial distributions (from `location`, `end_location`) .
 3. **Feature computation:** compute CQ, CS, SC, RC, DI, TS as above; normalize to [0,1].
 4. **Weighting/Scoring:**
 - o **Supervised option:** learn `wiw_iwi` to maximize correlation with a target when A+B are on (e.g., team xG per 90, goal diff per 90, entries into final third).
 - o **Unsupervised option:** equal weights or principal-components weighting (more stable when data is sparse).
 5. **Validation:**
 - o **Back-test:** do top-10 compatible pairs improve your target metric vs randomly sampled pairs with similar positions?
 - o **Ablations:** recompute score with each component dropped; check which components drive predictive signal.
 - o **Robustness:** control for opponent quality, game state, minutes, sample size (e.g., ignore pairs < 300 shared minutes).
 6. **Deployment:** use rolling windows to keep scores current; expose pair and small-group (triplets) scores to scouting or lineup tools.
-

Extensions (optional but powerful)

- **Small-group compatibility:** generalize to **hyperedges** for triplets/units (e.g., CB-LB-DM triangle) and re-use CS/TS.
 - **Context awareness:** compute **situation-specific** scores (vs low/mid/high block; leading vs trailing; first vs second half).
 - **Uncertainty:** bootstrap possessions to add confidence intervals around each component.
 - **Explainability:** show top sequences and zones that most increased each component (e.g., top A→B passes under pressure).
-

Why this works with StatsBomb Events

- Rich pass features (recipient, angle, distance, height, crosses/cut-backs) let you rate connection difficulty and intent for **CQ/CS** .
- Carry/dribble and their end locations capture non-pass progression between the pair for **CS/TS** .
- `under_pressure` and `counterpress` flags allow tight **defensive chain** modeling for **DI** within the 5-second window .
- Positions and `location` fields support **SC** (zone maps) and role vectors for **RC** .