Homefield Advantage in E-Sports: Evaluating the Win-Rate Disparity in League of Legends

Furui Guan | fg436

Yize Cheng | yc2253

1. Introduction

In most traditional sports, the idea of “home-field advantage” has been widely studied. As we’ve discussed in class, teams tend to perform better when they play in their home fields, which might be a result of a variety of potential reasons. In E-Sports, however, there isn’t really a “home stadium” in the same physical sense. Players compete online, often from remote locations, on maps that are supposed to be symmetric and therefore perfectly fair. Yet, as long-time gamers, we’ve noticed that this might not always be the case. Both our first-hand experiences as players and online discussions suggest that many E-Sports games are not perfectly balanced, giving one side a slight advantage over the other, similar to the effect of the “homefield advantage.”

To further explore this phenomenon, we’ll use League of Legends, one of the world’s most popular, most played, most renowned, and most successful E-Sport titles as an example to explore how homefield-advantage plays a role in the world of E-Sports, and provide and evaluate some of the potential explanations for why homefield advantage exists in League of Legends, as well as other similar E-Sports. In doing so, we hope to provide a more structured, data-driven answer to a question that LOL players have been arguing about for years: how fair is the game, really, and why?

2. Background & Problem Statement

2.1 How League of Legends Works

To help the readers better understand the context, we'll briefly explain how League of Legends works. In League of Legends, players compete in teams. Normally, each game will have 2 teams, each assigned to one of two sides on the Summoner's Rift map: **Blue side**, which spawns on the bottom-left, and **Red side**, which spawns on the top-right. The ultimate objective of each team is to destroy the other team's Nexus, protected by turrets, minions, and players playing as champions. In theory, since the game is designed to be balanced, both sides should win roughly 50% of the time across a large number of games.

2.2 The Win-Rate Disparity

However, according to the data from leagueofgraphs.com, a widely used site by League players to gather game data, Blue Teams generally have a higher win rate than Red Teams in League of Legends.

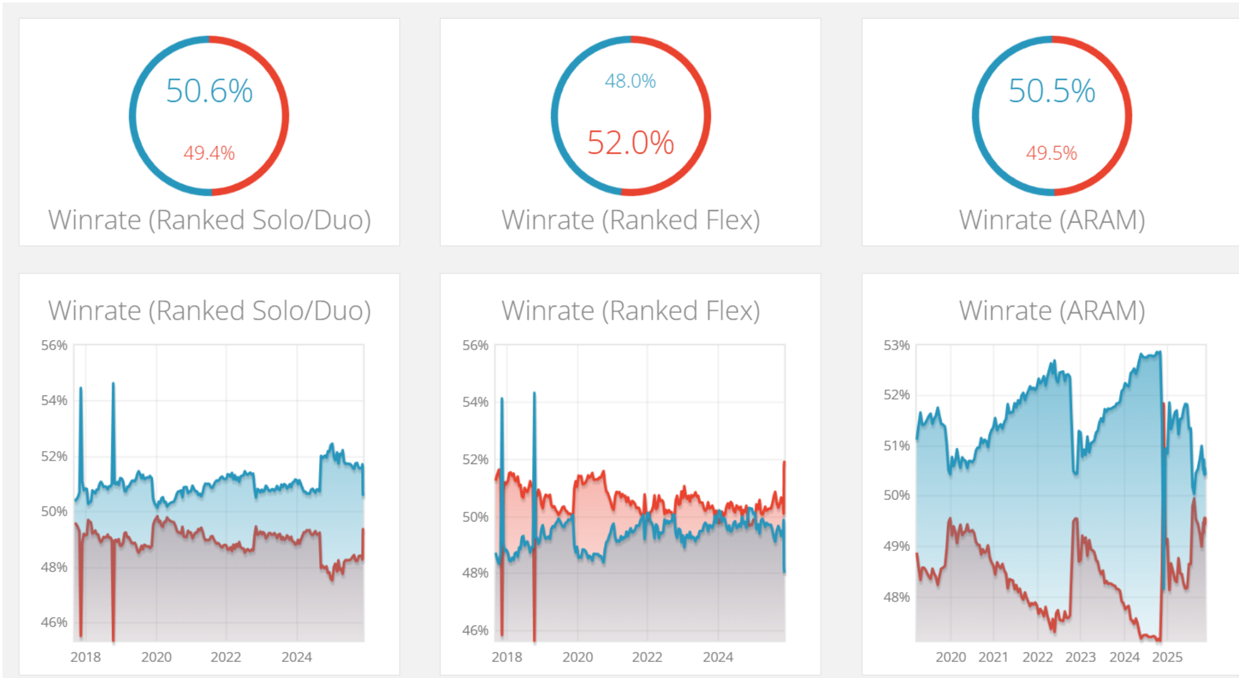

Figure 1: win-rates for blue and red teams in different modes across time. Retrieved from leagueofgraphs.com

As we can see from Figure 1, since 2018, the win rate for the Blue Team has been consistently higher than the Red Team in Ranked Solo/Duo and ARAM (we will explain the modes later), especially in the 2nd half of 2024. The difference in win-rates between the Blue and Red Teams went as high as 4% in Solo/Duo and 6% in ARAM. Ranked Flex, on the other hand, shows an advantage for the Red Team with the overall win rate of 52%. This might be a result of the difference in the matchmaking algorithms between Solo/Duo games and Flex games. But overall, the Blue Team does seem to have an advantage over the Red Team most of the time. This was also confirmed by a Data Analyst at Riot Games, the developer of League of Legends, during a lecture at the University of Southern California's School of Cinematic Arts, where he was a guest speaker for a game design class Furui was taking.

Although the differences' numbers don't seem that big, they do suggest some sort of advantage to one team over the other. Especially given that the data come from all the League of Legends games played in history, which, given League's huge player base and daily active users (4.3M+ according to activeplayer.io/), means that our data come from countless games that should, theoretically, push the win rate of each side to almost exactly 50%. Therefore, these differences in win rates, while small in numbers, do raise some questions about the balance of the maps and rules in the game because they come from a huge number of game samples.

2.3 Potential Explanations

Based on our own playing experience and prior community debates, we came up with four plausible sources of asymmetry: (1) the **map layout** on Summoner's Rift (terrain, jungle camps, objectives, pathing) might make it easier for the Blue Team to access strategic positions and/or resources, (2) The **Ban/Pick rules** in drafting mode (where the teams take turns to ban champions and then select champions, starting with the Blue team) gives the Blue team a strategic advantage in planning their team compositions, (3) the **camera angle and UI layout** could make it easier for players on the Blue team to view the game, perform certain actions, and control their champions, contributing to the higher win rates of the Blue teams, and (4) the **matchmaking system** might be, although never officially confirmed by Riot themselves, intentionally balancing the average strength of the players assigned to different sides to offset the "Blue Team Advantage" resulted from the previous 3 points, contributing to more complications of the win-rate disparity.

3. Methodology

3.1 Siloing Impactful Factors with Game Modes

Our goal in this paper is to treat these four elements as potential “hidden home-field factors” and to estimate how much each one contributes to the observed difference in win rates between Blue and Red sides.

We have gained access to the API for Riot Games’ database, which allows us to gather detailed match/game statistics to perform our analysis and evaluate the impact of each of the aforementioned sources of imbalance. But before diving into detailed data, we’ll first explain our overall methodology of isolating the impacts of each factor.

We will take advantage of the fact that different LOL game modes “switch on” or “switch off” different subsets of these factors while keeping the basic win/loss outcome the same. Basically, different game modes expose players to different combinations of the suspected factors (map layout, BP rules, camera angle, matchmaking). If we assume that the underlying player skill distribution is similar across different modes, which is largely true given the large samples of matches, then we can compare the win-rate gaps across different game modes to approximate the relative/marginal contribution of each factor.

The game modes we will be looking at include:

1. Ranked (SOLO/DUO): In this mode, the teams are matched with each other by the matchmaking system. They first complete their BP process, and then enter the game in

the classic map called Summoner's Rift to play. In this mode, players are exposed to Map Layout, BP Rules, Camera Angle, and Matchmaking Mechanism.

2. Normal / Quickplay: In this mode, the players are also matched together by the system. They can directly pick their champions all at once without having to go through the BP process, which eliminates the effect of the BP Rules on win rates. The game is also played in Summoner's Rift. In this mode, players are exposed to Map Layout, Camera Angle, and Matchmaking Mechanism.
3. ARAM: This mode is played in a different map where there's only one lane in the middle and no other strategic resources. Therefore, the Map Layout factor of the Summoner's Rift map should not have an effect on win rates in this mode. Also, the players don't get to do the BP process. Therefore, players should only be affected by Camera Angle and Matching Mechanism.

By looking at data from each game mode, we can derive the relative contribution of the BP Rules by comparing Ranked games against Normal games. Similarly, we can derive the impact of the map layout by comparing ARAM games against Normal games. Since Riot never confirmed any tweak to the matchmaking algorithm of ARAM, and given that ARAM is one of the less serious modes in League of Legends where winning doesn't matter that much, it's reasonable for us to set the ARAM matchmaking system as the benchmark (i.e., assuming that the ARAM matchmaking system is perfectly fair), and therefore derive the effect of the Camera Angle from the win-rate gaps in ARAM. Then, eventually, we can use our estimates for the previous factors to analyze the win-rate gaps across different game modes to see whether the matchmaking algorithm for specific modes (e.g., Flex) is slightly favoring one side over the other.

3.2 Data Collection Strategy

To get representative data, we used different sampling strategies for each mode tailored to player demographics. Early testing showed that high-skill players populate Ranked queues but rarely play Normal / Quickplay. A uniform approach would left us with too little Quickplay data, so we designed a twofold method.

For Ranked Solo/Duo and ARAM, we used a top-down approach with our script 'multi_queue_collector.py'. We seeded the initial player pool from the top 500 players on Riot's Leaderboard (Challenger, Grandmaster, Master tiers) and recursively pulled their match histories. This ensures our competitive data reflects high-level play where adherence to the meta and optimal draft strategies are strongest.

For Quickplay, we switched to a population-centric approach with 'robust_collector.py'. Since elite players rarely touch Blind Pick, we targeted the median playerbase, namely Gold, Silver, and Bronze tiers. We also increased the fetch depth from 20 to 50 matches per player to maximize sample volume. This seeks to capture the average player experience, and the fact that Red side advantage shows up most clearly here supports our hypothesis that the disparity is systemic, not strategic.

3.3 Statistical Methodology

To make sure our findings weren't just random noise, we used a Chi-Square Test. This is ideal for categorical win/loss data, letting us compare observed distributions against what we'd expect if the game were perfectly balanced.

For each mode, we set up:

- **Null Hypothesis (H_0):** The game is balanced; both sides have exactly 50% win probability.

- **Alternative Hypothesis (H_1):** Win rates deviate significantly from 50%.

- **Significance level (α):** 0.05 standard.

If the p-value is less than 0.05, we reject the null hypothesis and conclude there's a real structural imbalance in that mode.

4. Data Analysis

4.1 Factor 1: The Camera Angle

We started by isolating the camera angle—the fixed isometric view that supposedly favors Blue's upward perspective. We used ARAM data for this since it has a perfectly symmetric single-lane map and no draft phase, controlling for map geometry and strategic composition.

Across 2,084 ARAM matches, Blue won 50.72% of the time, meaning just a +0.72% deviation from 50%. The Chi-Square test gave us a p-value of 0.51, meaning it's not statistically significant.

We conclude that the camera angle and HUD placement don't really matter. Camera isn't driving the disparity.

4.2 Factor 2: Map Geometry & Matchmaking Interaction

Next, we looked at map asymmetry using Quickplay matches. This mode uses the standard Summoner's Rift but has no draft phase (Blind Pick), isolating map layout from strategic drafting.

Analyzing 1,500 Quickplay matches, we found Blue won only 45.93%, at a -4.07% deviation ($p=0.002$). This is highly significant, and the opposite of what we expected. Instead of a Blue advantage, we found a strong Red side advantage.

This surprising result points to matchmaking compensation. It seems the matchmaking algorithm assigns higher-skill players to Red side to offset perceived structural disadvantages. But in Quickplay, there's no draft phase to offset, so this skill bump over-corrects. Red ends up with stronger players and no draft disadvantage, leading to their dominance despite map layout.

4.3 Factor 3: The Pick/Ban (Draft) Effect

Finally, we isolated the draft phase by analyzing Ranked Solo/Duo matches. This mode brings back champion selection alongside the map and matchmaking variables present in Quickplay. Note that Blue has a first-pick advantage from the rules of the game.

Across 3,214 Ranked matches, Blue won 50.65%, basically back to equilibrium ($p=0.46$). The key insight comes from comparing this to Quickplay. Introducing the draft triggered a massive +4.72% swing in Blue's favor (from 45.93% to 50.65%).

This tells us that the strategic advantage of first pick is the primary driver of Blue side advantage. It's powerful enough to neutralize the skill premium (MMR offset) that matchmaking assigns to Red team, bringing the outcome back to a statistical tie.

4.4 Summary Table

See Appendix A for full statistical breakdown.

5. Discussion

5.1 Interpretation of the Quickplay Anomaly: Algorithmic Inertia

The most surprising finding is the pronounced Red advantage (54.07%) in Quickplay. This contradicts the historical Blue Side Advantage narrative and demands explanation. We may call this phenomenon Algorithmic Inertia.

Here's what could be happening: In Ranked, Blue has a real strategic edge thanks to first pick in draft. To keep things fair and maintain a 50/50 win rate, Riot's matchmaking uses an MMR Offset, where it intentionally gives Red team slightly higher-skilled players to compensate for Blue's draft advantage. In theory, it balances: Blue's strategic edge + lower skill ≈ Red's strategic disadvantage + higher skill.

But our data suggests this same matchmaking logic gets applied blindly to Quickplay, creating a mismatch. Quickplay has no draft phase, so the first pick advantage doesn't exist. Yet the system still seems to assign the skill premium to Red. So Red gets a double advantage: better players (via MMR offset) without suffering any draft penalty. This creates a systematic overcorrection where the artificial skill gap drives Red victories, overpowering any minor map disadvantages.

5.2 Limitations: The Challenge of Hidden Metrics

A major limitation of our study is the lack of transparency in matchmaking metrics for casual modes. In actual gameplay, we can see visible rank tiers (Diamond, Gold, etc.) that serve as proxies for skill, letting players quantify the "Elo gap" between teams. But Riot API uses a hidden internal MMR that isn't available to external users. This is why nobody has been able to quantify Riot's matchmaking mechanism. We can't directly verify the exact skill difference between Red and Blue rosters in our dataset but stuck relying on outcome data rather than input metrics. We also assume each factor's effects are additive and stable across modes, which might not fully hold if, for example, map layout interacts with champion picks or player controls in complex ways.

That said, the sheer size of the win-rate deviation serves as compelling indirect, corroborative evidence of algorithmic intervention. If matchmaking were truly skill-neutral, Quickplay win rates should mirror our ARAM baseline and hover around 50%. The sharp, statistically significant jump to 54.07% for Red implies something external is at play. Since camera and map effects proved negligible in our ARAM control, this massive deviation strongly suggests the system is actively favoring Red rosters, compensating for a strategic disadvantage that doesn't actually exist in this mode.

5.3 The Paradox of Fairness: Match vs. System

This disparity reveals an interesting tension in game balance design: the conflict between Match Fairness (micro-level) and Systemic Fairness (macro-level).

From the perspective of a single Quickplay match, the game is statistically unfair. With only a 45.93% win probability, Blue team starts at a mathematical disadvantage before the first minion even spawns. The matchmaking algorithm fails to provide an equal playing field in that specific game, handicapping Blue by withholding skilled players to offset a draft penalty that doesn't exist.

But Riot could justify this through systemic fairness. Since side assignment is random and automated, any given player will play on Red and Blue equally often over enough games. So while individual matches might be imbalanced, the experience theoretically evens out in the long run. This suggests Riot prioritizes a unified, robust matchmaking algorithm across all queues, accepting per-match inequality in casual modes as a trade-off for system simplicity and faster queue times.

5.4 Why We Excluded Professional Play: The Matthew Effect

We deliberately excluded professional esports matches (Worlds, LCK, LPL, etc.) from our dataset to avoid a critical bias. While pro games theoretically offer the highest quality gameplay data, they're compromised by the side selection mechanism.

Unlike public matchmaking where side assignment is random, professional tournaments typically let the higher-seeded team (or coin toss winner) choose their starting side. Recognizing the strategic value of first pick, stronger teams overwhelmingly choose Blue in the first scenario. This creates a Matthew Effect where the strong get stronger. A high Blue win rate in pro play reflects a compound variable (Map advantage + Team skill gap), not just the map's inherent benefits.

By restricting our analysis to algorithmically matched public games (Ranked and Quickplay/Normal) where side assignment is random and exogenous, we successfully neutralized this selection bias. Our findings reflect structural game balance rather than team skill disparities.

6. Conclusion

This study set out to unpack the widely debated Blue Side Advantage in League of Legends, treating it as a digital parallel to traditional sports' home-field advantage. By using a silo method across ARAM, Quickplay, and Ranked modes, we isolated the impacts of map geometry, camera angles, and draft rules. What we found fundamentally reshapes how we understand game balance: the disparity isn't primarily physical or geometric, it's structural and algorithmic.

Contrary to popular belief, our ARAM data confirmed that camera angle and HUD placement provide negligible competitive edge. Similarly, Quickplay data debunked the idea that map geometry alone favors Blue; in fact, without draft, the map is skewed toward Red due to matchmaking intervention. The true driver of Blue Side Advantage is unequivocally the draft phase. The strategic initiative of first pick is powerful, contributing a net +4.72% swing to Blue's win probability, effectively overriding the system's attempts to handicap them.

Our identification of what we call Algorithmic Inertia in Quickplay reveals the complexity of maintaining balance in an asymmetric game. Riot's attempt to enforce fairness via MMR offset succeeds in Ranked but fails contextually in casual modes, creating a Red Side dominance where none should exist.

Ultimately, League of Legends presents a paradox of fairness: it's a system composed of statistically unfair individual matches that relies on the law of large numbers to achieve equity over a player's lifetime. The "homefield advantage" is real, but it's less about where you spawn and more about who gets to pick first.

References

1. **Riot Games.** (2025). *Riot Developer Portal API*. Retrieved from https://developer.riotgames.com/

2. **League of Graphs.** (2025). *Winrate Statistics for League of Legends (Ranked, Flex, ARAM)*. Retrieved from https://www.leagueofgraphs.com/

3. **ActivePlayer.io.** (2025). *League of Legends Live Player Count and Statistics*. Retrieved from https://activeplayer.io/league-of-legends/

Appendix

Appendix A: Summary of Statistical Findings

Table A1: Win-Rate Analysis by Game Mode

Game Mode	**Sample Size (N)**	**Blue Win %**	**Red Win %**	**Deviation from 50%**	**P-Value**	**Significance**
ARAM	2,084	50.72%	49.28%	+0.72%	0.51	Not Significant
Quickplay	1,500	45.93%	54.07%	-4.07%	0.002	Highly Significant

Game Mode	Sample Size (N)	Blue Win %	Red Win %	Deviation from 50%	P-Value	Significance
Ranked Solo	3,214	50.65%	49.35%	+0.65%	0.46	Not Significant

Appendix B: Data Collection Methodology

To ensure the integrity of the dataset, different sampling strategies were employed based on the player demographics of each game mode.

B.1 Stratified Sampling Strategy

- High-Elo Collection (Ranked & ARAM):
    - **Target Population:** Challenger, Grandmaster, Master.
    - **Rationale:** These players adhere strictly to the "Meta," providing the cleanest data for analyzing strategic advantages (Draft).
    - **Method:** The script recursively fetches match histories from the top 500 players on the Ranked Leaderboard.
- Population-Centric Collection (Quickplay):
    - **Target Population:** Gold, Silver, Bronze.
    - **Rationale:** High-Elo players rarely participate in Quickplay/Blind Pick. To avoid sample scarcity, the collection window was shifted to the median player base.
    - **Method:** The script identifies active players in lower tiers and expands the search radius to capture casual match data.

Appendix C: Data Collection Scripts (Python)

The following Python snippets were developed to interface with the Riot Games API, implementing the stratified sampling logic described above.

C.1 High-Elo Seeding Logic (multi_queue_collector.py) Used for Ranked and ARAM data collection.

```
# From multi_queue_collector.py
def collect_seed_players(self, region="na1"):
    # Target High Tiers for Competitive Modes
    tiers = [
        ('CHALLENGER', 'I', 50),
        ('GRANDMASTER', 'I', 50),
        ('MASTER', 'I', 50)
    ]

    # ... (Fetching logic via Riot API)
    league_data = league_func(region=region, queue='RANKED_SOLO_5x5')
```

C.2 Population-Centric Seeding Logic (robust_collector.py) Used for Quickplay (Normal) data collection.

```
# From robust_collector.py
# Stratified Sampling Logic for Casual Modes
if queue_type == 'normal_blind':
    tiers_to_use = [
```

```
            ('GOLD', 'I', 30),
            ('SILVER', 'I', 30),
            ('BRONZE', 'I', 30)
        ]
        print("-> Targeting Lower Tiers (Gold/Silver/Bronze) for Normal
Blind")


    # Expanded Fetching for Casual Modes
    # Casual players play fewer games, so we fetch deeper histories
    if queue_type == 'normal_blind':
        count_to_fetch = 50
```

Appendix D: Raw Data

Please see the accompanying CSV files for the complete raw match data used in this analysis.