

LoLIntelligence: Alpha Development

ALYSSA ADAMS*

Arizona State University, Tempe
amadam15@asu.edu

Abstract

As online video games gain popularity throughout many age groups, they become an exciting new source for data on complex social systems. The online video game League of Legends has a massive, easy-to-access dataset that can be used to analyze in terms of information theory. This paper reports on the progress of the project LoLIntelligence, which analyzes the dataset in attempts to understand the dynamics of the game, as it undergoes its first round of development. A small subset of the data was used to develop information-theoretic measures across the data over time. With these measures, it is found that a small subset of game champions store and process the most information across the game over time. This suggests the massively complex game has some aspect of controllability across distributed players' behaviors.

I. INTRODUCTION

League of Legends is an online team-based strategy game that has become immensely popular over the last 5 years and is currently the most popular game of its type (1). As of 2015, there are 4 million active players (players who log in at least once per day) across the globe and this population has been steadily increasing since then. As a result of the game's popularity, many non-affiliated websites gain a lot of traffic by providing information about the game to players. In response, the game's developers have released all the game's data to the public, such that web-based applications can easily be created. In fact, the game developers host an application competition to further encourage using the game's data in creative and interactive ways (2).

One example of a popular web-based application is the website na.op.gg (3). This application will return general statistics of any given player for the past several games that player participated in (Figure 1). Often times, players will use web applications like this to learn about their team mates before a game begins.

There are several more examples of applications that are designed to access game data

in an easy-to-use way. However, none of these applications analyze data in a time series for a large ensemble of players, nor in ways that involve information-theoretic methods. These measures are useful in determining which factors change gameplay over time, as well as how player habits affect the overall play experience. Since the data dates from 2014 to current, it is surprising that this extremely rich and detailed dataset has gone untouched for network and information-theoretic analysis.

This paper reports on a first attempt to analyze the data generated by League of Legends in an information-theoretic sense. Because the data is so vast, only a very small subset of data is initially considered to develop methods of measurement and analysis. This data is translated into a time series of Boolean values (either 1 or 0) that is assumed to be generated by a dynamic Boolean network that changes nodes and edges over time. The time series is used to calculate information storage and information processing within the network. These results are translated back into gameplay to understand how player behavior affects gameplay overall.

As a disclaimer, many of the statements and assumptions made about this game are

*Expert Teemo player (Gold V)

derived from the author’s firsthand experience. For example, statements about “good” or “bad” gameplay and/or strategies are completely subjective.

II. GAMEPLAY

To gain a sense of which parts of the data are important for which measure, it is worth outlining how players interact with League of Legends. If a user wishes to play a game, they may queue up with friends, else they are placed with other random players. Two teams of five players compete against each other to win the game.

Champions Before the game starts, both teams have a chance to “ban” and pick particular champions. Champions are characters that players will play, and each player may only choose one champion to play throughout the course of a single game. An example of a champion is shown in Figure 2. If a champion is “banned”, nobody on either team will be able to play that champion, thus eliminating it from the game. Only one player can play a

given champion such that duplicate champions are not allowed in a single game.

Positions In addition to picking a champion, players must also pick a position on the team. Positions also indicate where on the game map (see Figure 3) that player will spend most of their time (top of the map, middle of the map, etc, and are named accordingly: top, middle, etc.). The concept of positions is not unlike basketball; each of the 5 members on a team has a particular role that they fulfill. Players find that some champions are good with many positions, while some are only good with a few.

Strategy Typically, a player’s combination of champion and position indicates the playstyle they wish to engage in the game and is hence considered a strategy. As an example, if a player picks the champion Teemo and the position Support, then both teams expect the player to play Teemo using a particular strategy throughout the game. The combination of all five team players’ strategies combine into a team strategy. Both individual and team strate-

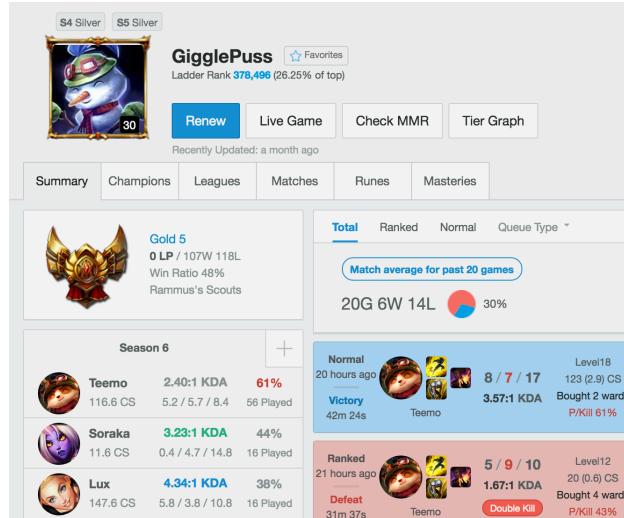


Figure 1: Screenshot from na.op.gg. In this web-based application, the statistics for the player “GigglePuss” are displayed. Any player’s username can be used to return data for that specific player.



Figure 2: One of League of Legends' most beloved champions—Teemo.



Figure 3: The main 2D map where players interact throughout the game. Team bases are highlighted in grey.

gies change over time based on what strategies are popular in the game community as a whole. Figure 4 shows an example of the screen before the game, where both teams select their champion and position.

Selecting a team strategy Teams pick their champions in a set order, and the first team is randomly decided by the game:

- Red team randomly gets to pick their first champion
- Blue team picks two champions
- Red team picks two champions
- Blue team picks two champions
- Red team picks two champions
- Blue team picks their last champion

After teams have finished picking their champions and positions, the game begins on a 2D area map (Figure 3). One team is stationed at one corner of the map, and the other team is stationed on the furthest, most opposite corner.

Winning the game The goal for each team is to destroy the other team's base by earning gold throughout the map and buying items to make their champion stronger than the enemy champions. Games can last anywhere from 20–60 minutes and ends immediately when one team surrenders or loses their base.

Games that are “ranked” place players on a skill tier system based on how many games they have won. Bronze tier houses the lowest skill players, while Challenger tier houses players that play on professional teams. Winning many games over time increases the chances a player will climb the tier. For a game, players are always randomly matched with players in the same tier.

III. ACTUAL GAMEPLAY

League of Legends is notorious for having a “toxic” player community, meaning many players can be unfriendly and bully each other

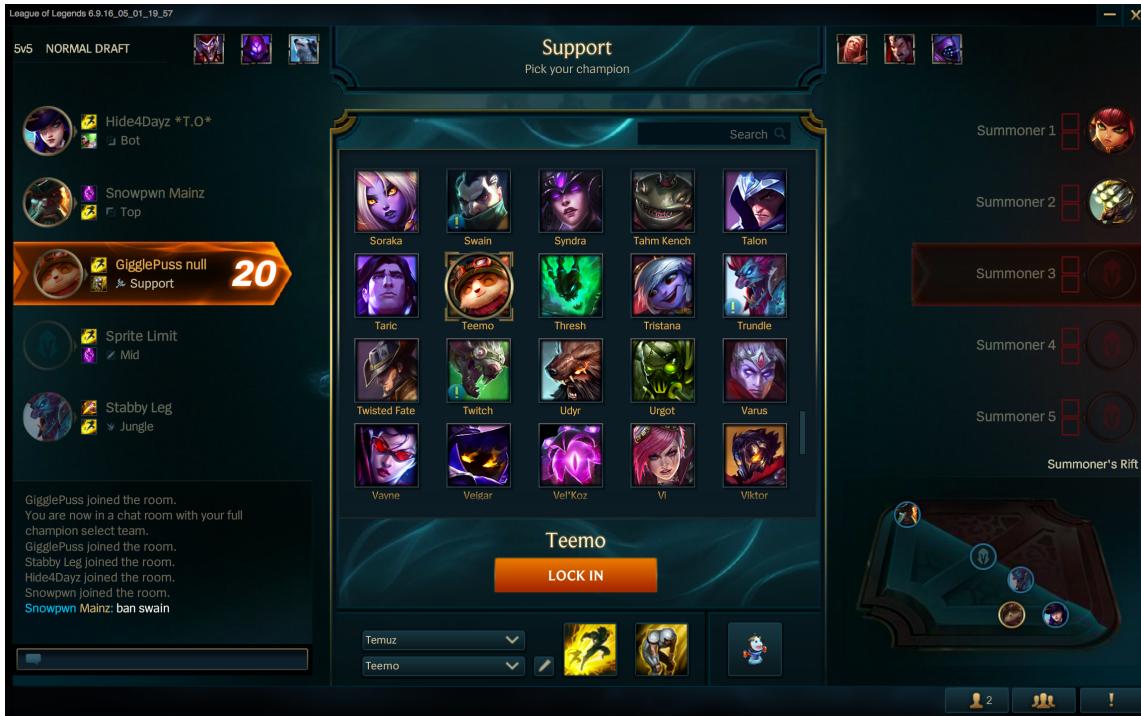


Figure 4: The pre-game screen where both teams pick their champions and positions. In this moment, the player "GigglePuss" is currently selecting the champion Teemo for the Support position. Both teams are able to see which champions the opposite team has picked, but not their subsequent positions. Banned champions are indicated by the six icons on the top of the screen. On the bottom left of the screen, the player "Snowpwn" suggests the champion Swain should be banned.

around during games. Efforts are constantly enforced to minimize the effects of toxic players by the game developers, which has had some success (4). From a behavioral-ecological standpoint, it is easy to notice the structure of the game is deeply affected by player behavior. For example, toxic players cause many players to surrender games early, which affects the overall statistics of the game. In general, toxic players often lose much more than other players, thus placing them in lower tiers. This makes it extremely difficult for non-toxic to climb out of lower tiers, and any strategies seen in lower tiers are generally considered “bad gameplay strategies”. Therefore, the majority player community avoids particular gameplay strategies because of their lower-tier association, changing the gameplay across all skill tiers.

By far, toxicity is not the only behavior that affects gameplay. Professional players set many trends that most other players work to imitate. Using Teemo as an example, one professional player started playing Teemo as a high-damage character in the Top position while other players were playing him in the Mid position. Since many people watch this player’s behavior, many imitated this player’s Teemo strategy which is a trend that continues today.

In addition to these social dynamics, when a particular champion becomes popular, players generally learn the strategy to beat it with a counter-strategy. Once the counter-strategy is popular, then the counter to that counter-strategy becomes popular, and so on. In other words, the game is not static by any means and changes according to current player behaviors. The entire game is largely dependent on feedback from the players’ strategies.

IV. CREATING A NETWORK

All the above can be represented as a network in several different ways, each highlighting an aspect of the game. On paper, there is no concept of one champion being better than another champion at any given time without players actually playing the game. As players play the game, the popularity of any given champion

(where the champion becomes popular for a variety of reasons) creates a dynamic hierarchy of preferred champions. If the champion Teemo becomes very popular, many players will assume Teemo is a very good pick at the moment. However, this might not actually reflect the actual amount of games player who play Teemo actually win. A champion’s **win-rate** is the actual number of games players who play that given champion win overall.

Champion counter-network At any given time, a network can be instructed to show which champion beats (*counters*) which champion based on the win/loss outcomes of players who play those champions. According to counterstats.net (5), Teemo beats another champion named Renekton in the Mid position because Teemo has the higher winrate for that position. In other words, Teemo wins more often than Renekton in the Mid position so Teemo is considered to counter Renekton.

This can be represented on a graph as Teemo and Renekton as nodes. A directed edge that goes from Teemo into Renekton indicates that Teemo counters Renekton. Winrate statistics can be found for every champion in every position. These statistics can be used to build a counter-network for every champion in League of Legends. This network is shown in Figure 5, where each node has a minimum of one in-degree and one out-degree. Labels for nodes are not shown.

An immediate question to ask is whether or not the vertex out-degree of a champion correlates with that champion’s overall win-rate. Figure 6 compares these two measures in color, where red indicates high values and blue indicates low values. Notice how the two measures seem to be much less correlated than intuition would lend. Champions that counter many other champions do not have the highest winrates.

Issues A problem immediately arises from the statistics that are used here. The winrates are heavily slanted by champion popularity and are not normalized. For example, the rea-

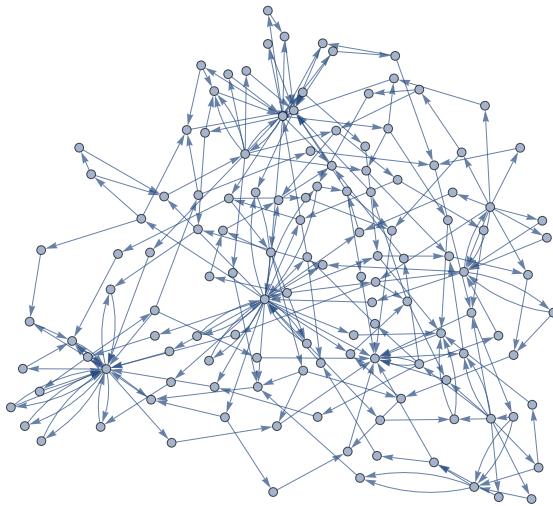
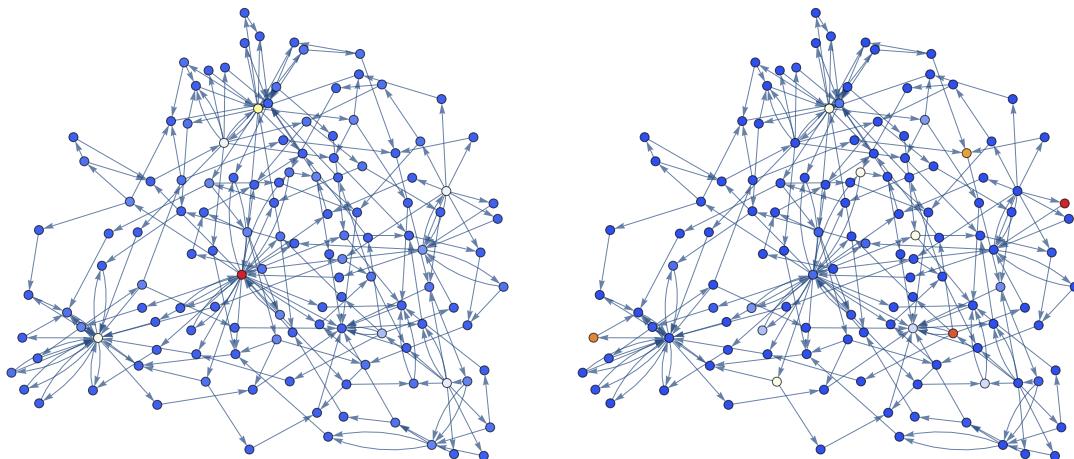


Figure 5: A network that shows how champions (nodes) beat (directed edges) other champions.



(a) Node color indicates node out-degree, or the number of
champions a given champion counters in game.

(b) Node color indicates the corresponding node's (champion's) winrate.

Figure 6: The same network in Figure 5 is shown with nodes colored given to out-degree (a) and winrate (b). Blue color indicates low values, while red color indicated high values. White colors indicate values that are in-between high and low.

son a particular node may have a high number of out-degrees is because that champion is simply popular and played in many different positions. Nodes that have only one in-degree and one out-degree might be unpopular champions for a large variety of reasons (the champion may be old and out-of-date graphically and is less aesthetically appealing).

An even bigger issue is that these networks change over time from player behavior feedback. On top of that, every few months a new node is introduced to the network when the developers add a new champion to the game. Champions are never removed from the game. In addition, old champions are changed or updated or spatial properties of the map are physically altered. All these changes affect this network drastically and as a result, edges are re-wired on a daily basis.

Collecting better data In order to avoid the most extreme network changes, any further data used for analysis will only be used from times where the game developers did not make changes to the game. Any changes to the network result from internal player feedback mechanisms.

To further reduce the data set, only the last 50 games played by the author will be used. Data is collected directly from the game developers through a set of Python 3.0 scripts known as Casseopiea (6). These games span over the course of 14 days. This automatically introduces a strong bias towards the playstyle of a single player, but the idea is to develop scalable measures to eventually analyze much larger ensembles of games for many different players. With more data, general statements about the game as a whole can be made.

In order to better understand the subset of data that is used throughout the rest of this paper, the champion counter-network for these 50 games is shown in Figure 7. Note how the center node has the highest in-degree and out-degree. This is a reflection of the bias introduced by the small amount of data from a single player used for this analysis.

Instead of analyzing the network as a typ-

ical Boolean network (since Boolean update rules are not defined to determine how node values, whatever values those may be, update from one timestep to the next), data will only be analyzed as a time series, regardless of the structure of the underlying network. It is assumed that the network changes on a daily basis so fully understanding how the network evolves is not yet possible. Keeping these things in mind, the analysis on the data based on information measures will be not depend on the network structure of dynamics, and instead based on Boolean values of nodes as they change over time (regardless of edges).

V. INFORMATION MEASURES

Active information and transfer entropy are Shannon-based ways of measuring information storage and processing (respectively) in a system (7). They do not require more than a time series of discrete values to calculate, which is a convenient measure to use with this particular data set, since it does not depend on any network dynamics.

Active information In particular, active information is defined as the ability for a particular node to store information. Over discrete time steps, a node x will have discrete values. For any given time n , active information is calculated by summing over all possible time steps the odds of finding a particular pattern of size k :

$$AI_x = \sum_n \log_2 \frac{p(x_{n+1}, x_n^k)}{p(x_{n+1})p(x_n^k)} \quad (1)$$

The size of k is analogous to the history length, or memory, or a system. The more often a given pattern of size k given a value at time n is found over the time series in a given node, the higher the active information that node has. For nodes with Boolean values (only having a value of 0 or 1), the maximum active information a value can have is 1.

Transfer entropy Almost similar to active information, transfer entropy calculates the like-

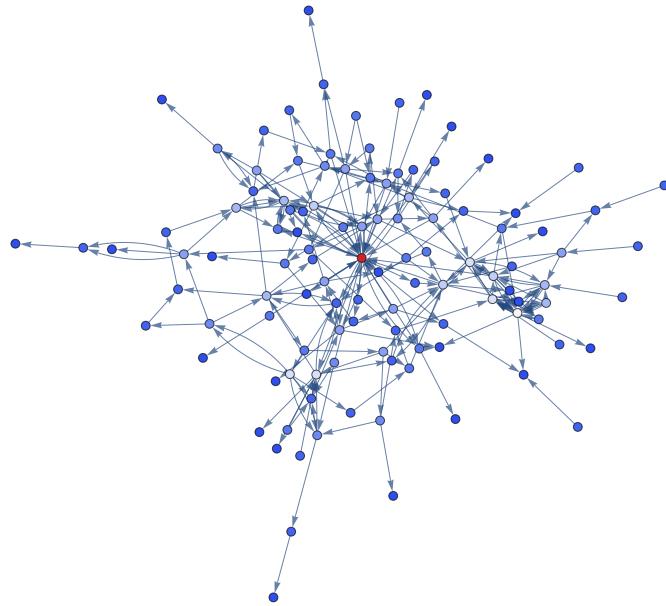


Figure 7: A network that shows how champions (nodes) beat (directed edges) other champions for the last 50 games the author played. Nodes colors indicate the out-degree for the corresponding node. Blue color indicates low values, while red color indicated high values. White colors indicate values that are in-between high and low. Node how the central node has the highest out-degree value. This is the main champion (Teemo) played by the author and highlights the data biases for this analysis.

lihood that node y will affect what node x 's value is given some history length of node x , k . For a single node, the transfer entropy is calculated for all possible interactions between other y nodes:

$$TE_{y \rightarrow x} = \sum_n \log_2 \frac{p(x_{n+1}|x_n^k, y_n)}{p(x_{n+1}|x_n^k)} \quad (2)$$

The more often a given pattern of size k given a value at time n for x , while also considering the value for another node y is found over the time series in a given node, the higher the transfer entropy that node has. If a node has a high value (close to 1) for transfer entropy, then that node is easily affected by the states of other nodes and thus processes much information about the entire system.

Generating two time series Since the subset of data being considered is so small, no complete networks (one where all champions are

present) can be extracted. As a baseline to generate a time series, the data was sorted into days as a time unit. Two types of values for the time series were considered: champion winrate and champion popularity. The winrate was only considered between that champion and their opponent with the same position. In other words, even if Teemo's team lost the game, if Teemo beat his opponent in the middle position by earning more gold, then it is counted as a win for Teemo. If Teemo wins more games than he loses in a day, then the Teemo node is assigned a 'winrate = 1' value for that day, otherwise the node is assigned a 'winrate = 0'. Similarly, if Teemo was in more than a third of all games played that day, then that node is assigned a 'popularity = 1', otherwise it gets a 'popularity = 0'. These thresholds were arbitrarily chosen, and need further justification by investigation.

VI. RESULTS

To gain a general sense of the information dynamics in this data, both the AI and TE were calculated for both the popularity time series and the winrate time series (Figures 8 and 9, respectively). The values are ranked by frequency and different k values are indicated by color. In general, AI increases with larger k values, while TE decreases with larger k values. AI values are much larger than TE values in both types of time series. Overall, this general information analysis reveals little difference between the two types of time series.

Both of the observables in the network (winrate and popularity) show similar trends in the distribution of AI and TE value. This suggests that in the data used, both observables have the same affect on the game. Intuitively, one might think that a winrate data would have a higher impact on player behavior but these graphs suggest champion popularity has an equal effect.

For both TE and AI, different champions occupy the highest values, but are fairly constant with different k values. This may be a reflection of the small data size, but it could also indicate that certain champions may play a pivotal role in the network as a whole. For example, the champion Master Yi was in the digests regimes of AI for every k value, which indicates this champion might play a pivotal role in the game dynamics as a whole. **Regardless of data bias, the fact that a small, single set of nodes populates the highest AI and TE values is an interesting feature of the game, and is worth exploring in further detail with a larger set of data.**

If TE is calculated only for nodes that are connected by an edge, the distribution of TE values changes a bit more between the two time series' (Figure 10). On average, TE values are slightly higher for winrate states. Most TE values are found between edges with a low average degree but in general, there is no clear trend between average vertex degree and TE.

Again, the lack of a clear trend may be from small data bias. Figure 7 shows how the use

of a single champion slants most of the player interactions to involve Teemo. Since one champion is unproportionally represented in this dataset, it probably has an affect on Figure 10.

Because the data in Figure 10 lack a clear trend, it is more informative to plot the total TE processed in each dataset as a function of k . Figure 11 shows a clear trend, and as seen in Figure 9, the total amount of information processed over a time series decreases as k increases. The effect is more dramatic in the winrate time series, which suggests **information on what champion is winning has more causal power over smaller timescales than longer timescales, while information on champion popularity has more causal power over longer timescales than information on champion winrates**. Again, this is a trend worth exploring with a larger set of data to see if the same holds true for the game as a whole.

Finally, it will be useful to see the trend between AI and TE. Figures 12 and 13 show a heatmap of TE values as a function of AI. Nodes are arranged on the x and y axes according to their AI value. Nodes with low AI are towards the origin, while nodes with high AI are away from the origin. Both the x and y axis have the same arrangement for nodes. The distribution of the nodes' AI values is shown below the heatmap and it only shown for the x axis since it is the same in the y axis.

TE values are not symmetric, so the heatmap is not symmetric. The TE for a node A using B is different than node B using A. Each Figure shows data for a different k value. These figures only represent data from the winrate time series, since the same figures produced by the popularity time series were not drastically different.

For low values of k , nodes with low AI values have, in general, higher TE values but as k increases, this trend becomes more distributed over various AI values. Probably, this indicates that for this dataset, there is a core set of interactions that are the most meaningful in terms of causal power. Since the small data that is sampled involves many interac-

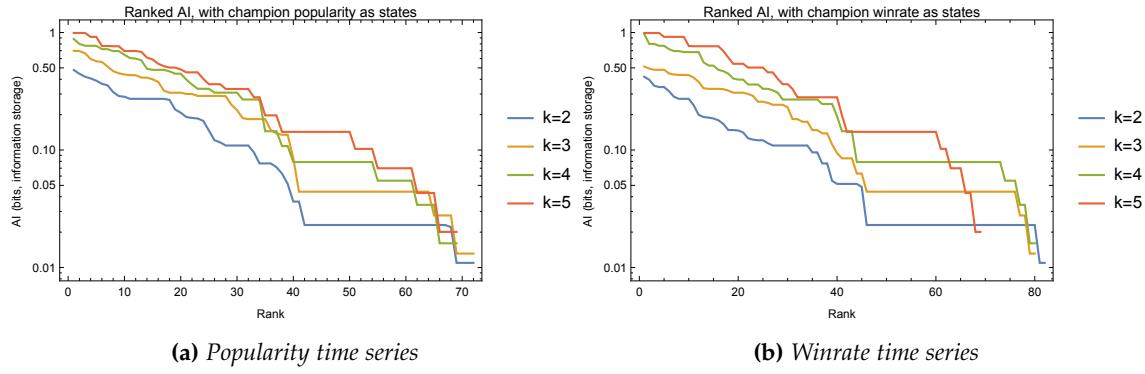


Figure 8: The distribution of active information (AI) values are sorted by rank for both time series. Different colors indicate different values of k .

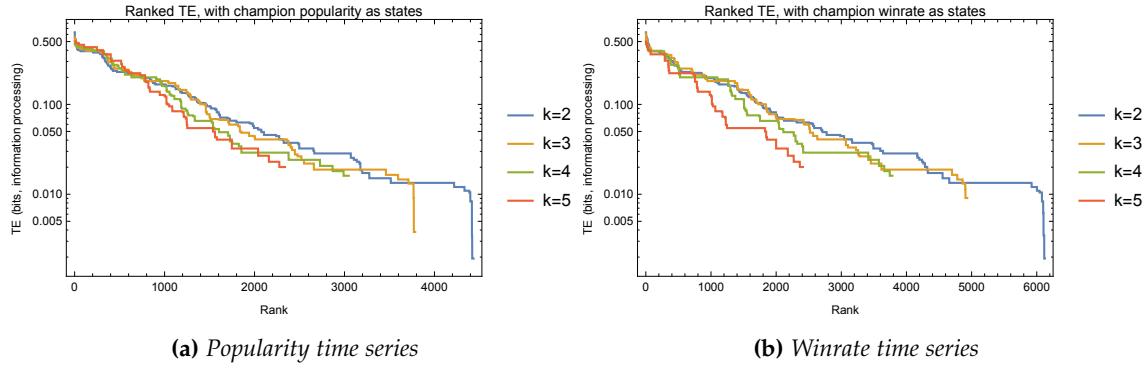


Figure 9: The distribution of transfer entropy (TE) values are sorted by rank for both time series. Different colors indicate different values of k .

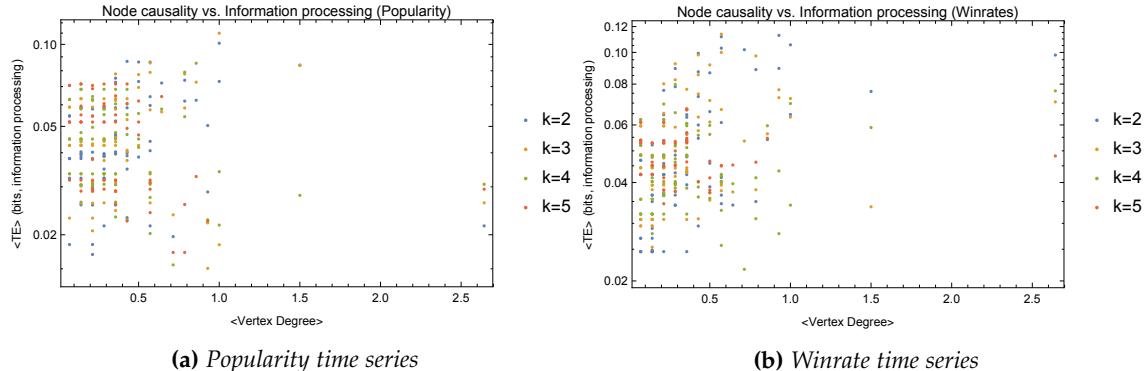


Figure 10: Average node degree (since node connectivity changes over time) vs. that node's average transfer entropy value. This is over the popularity time series. As before, different values for k are shown by color.

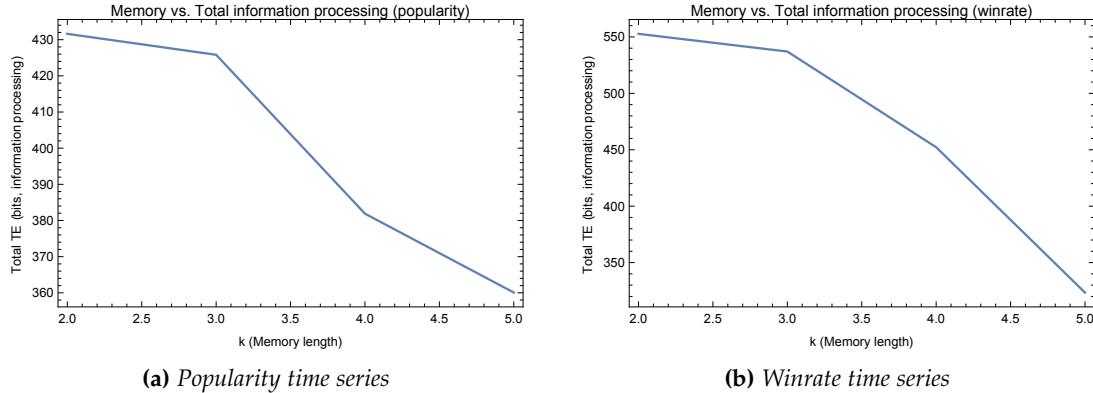


Figure 11: Total TE as a function of k for both time series.

tions with the same champion, many nodes were not included and results in many 0 values for TE and AI. However, **this relationship between TE and AI suggests a core number of champions have pivotal roles in the game dynamics**, as already suggested by Figure 8 (since the highest values for TE and AI are populated by a small subset of champions).

VII. DISCUSSION

The most striking results are shown by Figures 8 and 12. These figure indicate that most of the dynamics in the subset of games center around a small subset of champions. Interestingly, none of these “central” champions were Teemo, as intuition would suggest. Even though the dataset is small, this indicates a new notion of controllability. Since the same champions have high AI and TE values, they are perhaps controlling the overall game in some way.

Figure 11 indicates that information processing occurs at a greater rate with a shorter memory in winrate dynamics than popularity dynamics. This might indicate that players process the most information about winrates in the last few days, but not over longer timescales. Over longer periods of time, champion popularity trends seem to have a bigger affect on the game in terms of information processing. This might be an indication of players’ memory. Perhaps it is easier to remember which

champions that were played, rather than which champions that have won.

once again, all these results may be artifacts of a tiny sample size. The play habits of the single player are geared more towards playing Teemo over and over again, no matter the win or loss outcome. Regardless of this fact, the results are interesting, and it is worth investigating a larger dataset to see if these trends are a central feature of the game.

VIII. WHAT’S NEXT?

The algorithms developed for this analysis are certainly scalable with data size, so the most immediate step is to use data from a representative sample of the League of Legends community. Several questions are worth exploring to understand how this network behaves. Here are a few questions, ordered by tractability:

- How do these results change with different thresholds for winrate and popularity?
- How do these results change with a representative sample of the entire League community?
- Is there a small subset of champions that “control” the game overall?
- How do these results change as a function of player skill (sampling from each tier separately)?
- How do these results change between patches per tier, specifically when a new

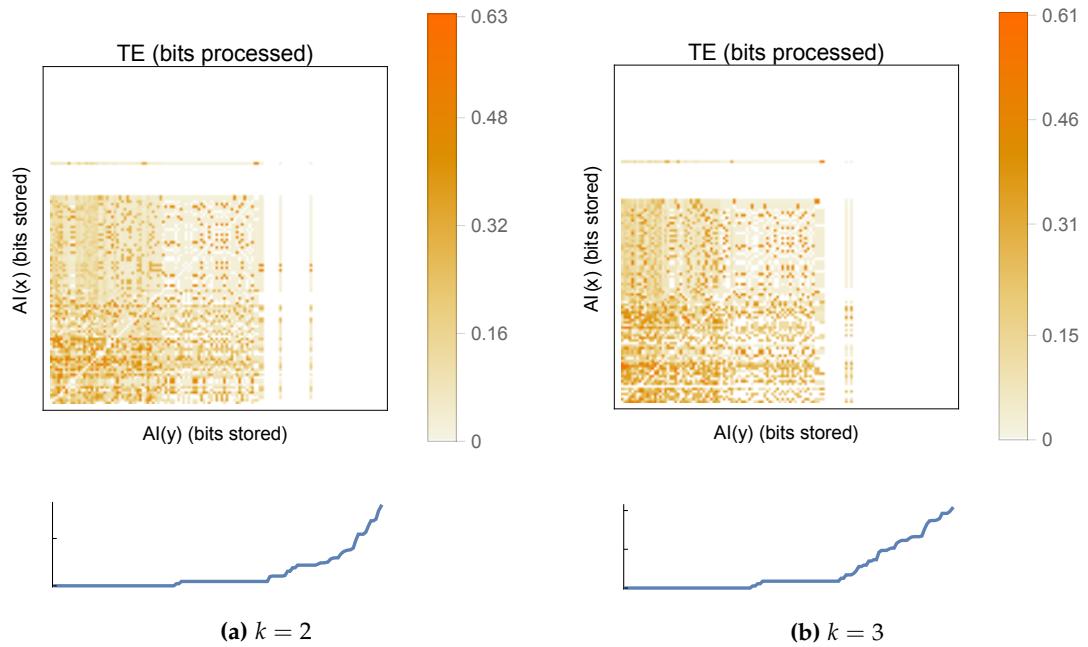


Figure 12: Heatmap of TE values between nodes from the winrate time series. Nodes are arranged on the x and y axes according to their AI value. Nodes with low AI are towards the origin, while nodes with high AI are away from the origin. Both the x and y axis have the same arrangement for nodes. The distribution of the nodes' AI values is shown below the heatmap and it only shown for the x axis since it is the same in the y axis.

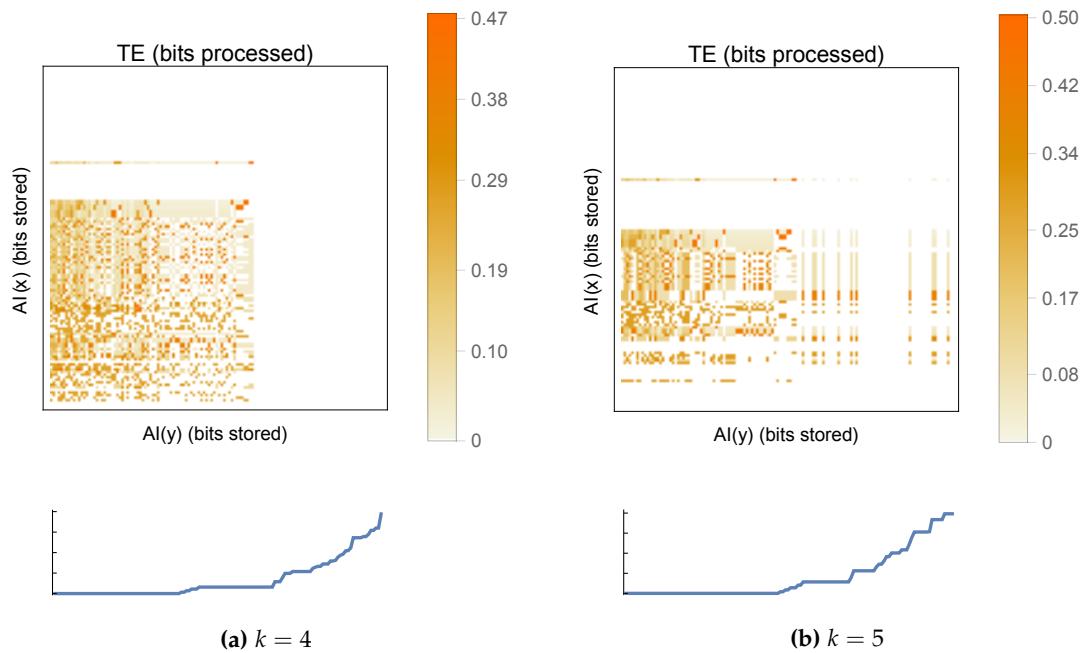


Figure 13: Heatmap of TE values between nodes from the winrate time series. Nodes are arranged on the x and y axes according to their AI value. Nodes with low AI are towards the origin, while nodes with high AI are away from the origin. Both the x and y axis have the same arrangement for nodes. The distribution of the nodes' AI values is shown below the heatmap and it only shown for the x axis since it is the same in the y axis.

- node is added?
- Are there nodes that have consistent AI and/or TE values?
 - What is the most sensitive part of the network, and does it change over time?

The goal of the project LoLIntelligence is to understand how League of Legends behaves as an ecosystem of player behaviors, and to use this information to make predictions about the future. It seems like players perform a distributed computation to evolve the network, and then respond to the network's evolution by changing the network structure in a state-dependent way. State dependence is the central dynamical driver of the network, and with the game developers constantly adding nodes and changing properties of the game, the network never falls into a constant, unchanging state.

Understanding how the game evolves with topology and responds to changes can help predict future dynamics and player strategies. It can also help develop a notion of controllability without fully understanding the network dynamics, given that time series data is available. This type of analysis on a network can be extended to other social networks or security systems. The LoLIntelligence project will be con-

tinue to develop and will eventually be synthesized into a web application where users can learn about information-theoretic techniques, and how League of Legends can be used to study human behavior and distributed computation.

References

- (1) *Player Numbers*. Riot Games. riotgames.com/tags/player-numbers. 2015.
- (2) *Stats Straight From the Source*. Riot Games. developer.riotgames.com. 2016.
- (3) *What is OP.GG?*. OP.GG. <http://na.op.gg>. 2016.
- (4) *Player Behavior*. Riot Games. <http://www.riotgames.com/tags/player-behavior-team>. 2015.
- (5) *Counter Picking Statistics*. CounterStats.net. <http://www.counterstats.net/>. 2016.
- (6) *Casseopeia Documentation*. Casseopeia. <http://cassiopeia.readthedocs.io/en/latest/>. 2016.
- (7) H. Kim, PCW Davies, & SI Walker. *New Scaling Relation for Information Transfer in Biological Networks*. 2015. J. Roy. Soc. Interface 12 20150944.