Applying Machine Learning to Enhance Esport Broadcast Narratives

Alan Pedrassoli Chitayat

Doctor of Philosophy

University of York
Computer Science
January 2025

Abstract

Esports has emerged as one of the fastest-growing entertainment phenomena, blending the excitement of traditional sports with the immersive, digital environment of online gaming. Fuelled by the popularity of games broadcast, esports commands a massive audience that demands increasingly engaging and immersive experiences. This drives tournament organisers and broadcasters to seek innovative ways to meet these expectations. The digital nature of esports provides a wealth of accessible data, making it an ideal domain for advancements in Artificial Intelligence (AI) and Machine Learning (ML). This motivates research in the domain as a meaningful data-driven solution to creating more engaging esport broadcasting experiences.

However, leveraging ML to enhance esports broadcasts presents unique challenges. The fast-paced, complex nature of esports requires models that generate meaningful insights while integrating seamlessly with live coverage. Current research often struggles to bridge the gap between theoretical advancements and practical application, leaving many ML innovations underutilised in live esports broadcasts, which limits their real-world impact.

This thesis addresses these challenges by exploring how ML can be applied to enhance esports broadcast narratives. It provides insights for designing models with greater longevity, ensuring they remain functional across multiple game patches. It also offers considerations for integrating ML models into live broadcast environments, enabling them to more easily complement ecological contexts in real-world applications. Furthermore, it emphasises the importance of seamless integration with existing broadcasting strategies, from production workflows to narrative creation. These findings are then condensed into a framework aimed at ML practitioners that provides practical guidance on how to apply them in to future work in the domain.

By addressing these key areas, this thesis advances the practical and long-term impact of ML research in esports broadcasting and contributes to the continued evolution of this dynamic field.

Acknowledgements

The process of this PhD was full of challenges, tears and doubts. It was also full of wonderful moments, and it thanks to many very special people that I managed to reach this point.

Firstly, I would like to thank my supervisors for, above all, believing in me. Going from development work to research was a difficult transition, but it was important to know that I was on the right path. Your support throughout the years - leading up to and during my PhD - was immensely valuable and I am incredibly thankful to Florian Block, James Walker and Anders Drachen.

I would also like to thank my partner, Steph. It is difficult to put into written words how much you have helped me. Your persistent reminders of how much I had achieved and how much I was worth - usually rebutted by my passionate protests - prevented me from just giving up in many occasions. Your care for me, your support, you being there for me, I cannot begin to imagine where I would be without you.

To my mother, Milene, I must thank you for pushing me to pursue my dreams, and giving me the environment I needed to do so. I still remember not thinking it would be possible for me to attend university and you telling me that you would "make it work". Thank you for making it work, as you always do. Thank you for teaching me, for raising me into who I am, and for always being there for me.

I would also like to thank my brother, Samy. Twice I needed a lifeline and twice you were there for me. I do not know what I could have done without your support, and I am incredibly thankful for you help. Most importantly, I am thankful for just knowing you are there for me when I need it.

To all of my friends in the IGGI PhD program. I know you can all understand how difficult the PhD process is. Having you all to rely on, be it for advice, empathy, or to share special milestones and good moments, I am so grateful for being part of the IGGI community. I have special fond moments of the IGGI SIG. I am glad we all got that opportunity to share our news, complain about things together and just have a good time. The SIG was always a way to ensure I would have a good Friday, so thank you to everyone for making it a reality.

I would also like to thank all of my gaming friends. You were all a huge part of my life and you kept me sane when things were too difficult. You would listen when I had problems, celebrate it with me when I was happy. Though I know on many occasions I was just saying words at you, as I was not always very good at giving context, I knew it did not matter and that you were there for me.

Lastly, I would like to thank Nano and Pixel, our two cats. Though you often made writing mechanically hard, your presence and constant source of affection has been instrumental in me completing this thesis.

To anyone and everybody who supported and aided me throughout this long and difficult process, thank you!

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references.

The following list of publications contains the works that were completed as part of the contribution of this thesis:

> Chapter 3 is based on:
> [123] Alan Pedrassoli Chitayat, Florian Block, James Walker, and Anders Drachen. Beyond the meta: Leveraging game design parameters for patch-agnostic esport analitics. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 116–125, 2023
>
> Chapter 4 is based on:
> [124] Alan Pedrassoli Chitayat, Florian Block, James Alfred Walker, and Anders Drachen. Applying and visualising complex models in esport broadcast coverage. In *Proceedings of the 2024 ACM International Conference on Interactive Media Experiences*, pages 108–116, 2024
>
> Chapter 5 is based on:
> [125] Alan Pedrassoli Chitayat, Florian Oliver Block, James Alfred Walker, and Anders Drachen. How could they win? an exploration of win condition for esports narratives. In *Annual Symposium on Computer-Human Interaction in Play (CHI PLAY 2024)*. ACM, 2024

Additionally, I was also involved in some collaboration work that has been published in relevant fields during the completion of this thesis. While these works were not completed as part of the contribution of this thesis, the publications have been included (as is) in the appendix for transparency and completeness. The list of these publications is as follows:

> Appendix D includes:
> [168] Marko Tot, Michelangelo Conserva, Alan Pedrassoli Chitayat, Athanasios Kokkinakis, Sagarika Patra, Simon Demediuk, Alvaro Caceres Munoz, Oluseji Olarewaju, Marian Ursu, Ben Kirmann, et al. What are you looking at? team fight prediction through player camera. In *2021 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2021
>
> Appendix E includes:
> [126] Alan Pedrassoli Chitayat, Alistair Coates, Florian Block, Anders Drachen, James Alfred Walker, James Dean, Mark McConachie, and Peter York. From passive viewer to active fan: Towards the design and large-scale evaluation of interactive audience experiences in esports and beyond. In *Proceedings of the 2024 ACM International Conference on Interactive Media Experiences*, pages 94–107, 2024

Contents

List of Tables

List of Figures

1 Introduction

In an industry where viewer engagement defines success [105, 65, 157], esports broadcasting faces the unique challenge of delivering dynamic, real-time content to an ever-growing global audience [10, 113, 46]. This increasing demand for more engaging and insightful coverage has motivated academic interest in the application of artificial intelligence (AI) and machine learning (ML) within the esports domain [24].

However, integrating these technologies into the live coverage of esports remains a complex challenge from both technical and usability perspectives [18, 168, 82, 70]. The intricate and fast-paced nature of esports further complicates the training and evaluation of ML models [70, 59, 174]. As a result, ML practitioners must navigate a range of obstacles to ensure their models positively impact esports broadcast narratives and enhance the audience experience [168, 126].

Despite advancements in ML, a significant gap remains in adapting these models for live broadcast environments [29, 147, 141]. This limits the tangible impact of such research and ultimately affects the viewer experience that streamers and broadcasters aim to enhance [56]. The relevance of this gap is amplified by the rapid growth of the esports industry, which has become one of the fastest-growing streaming markets and a key sector within the gaming industry [111, 113, 114].

This growth creates strong financial incentives for broadcasters to adopt increasingly complex and engaging coverage strategies to cater to diverse audience needs [56, 21, 82]. However, much of the ML research in esports focuses on performance metrics such as model accuracy, leaving limited emphasis on practical implementation in live broadcasts [70, 141, 59, 29, 30, 168, 22]. This disconnect prevents valuable findings from reaching their intended audiences and contributing to the broadcasting ecosystem.

Additionally, as ML models are primarily evaluated with metrics such as accuracy, the impact such models can have in the way in which esports content is broadcast and consumed is not fully explored. This can lead to an over-reliance on ML models, or situations in which ML insights can retract from the audience experience. For example, a model that is too accurate at predicting the winner of a match could have a negative impact in the overall audience experience [59, 70]. Thus, it is important to design models in a way that can enhance the esport broadcast narrative.

This thesis aims to address this issue and provide a framework for applying ML to enhance esport broadcast narratives. Firstly, it investigates the current literature on esports, particularly focusing on how ML is utilised in the domain. It also examines how data is integrated into broadcasts, with an emphasis on narratives and storytelling. Through this analysis, the thesis identifies gaps in the knowledge and establishes the methodological approaches used in the field. Secondly, it addresses these gaps through a series of case studies that explore:

- Methods to extend the longevity of ML models, ensuring they remain functional across multiple in-game patches (see Chapters 3).

- Strategies for integrating models into live esports broadcasts, ensuring usability in ecologically valid contexts (see Chapters 4).

- Techniques for designing and evaluating models to align seamlessly with existing broadcast narratives and production workflows (see Chapters 5).

These case studies form the basis for a empirically derived systematic framework. This framework offers methodological guidelines for developing ML solutions that enhance live esports coverage while integrating seamlessly into broadcast narratives. It provides ML practitioners with actionable insights into designing models with long-term functionality, effective evaluation strategies, and sustainable deployment methods.

1.1 Research Motivation

The rise of esports has led to significant academic interest across various fields [37]. In particular, the field of ML has generated a substantial body of research [51, 24]. While much of this research is motivated by the popularity of esports and its global audiences [22, 59, 30, 151, 103], it has largely focused on improving ML performance and techniques for training more complex and accurate models [29, 147, 70, 8].

Advancements in model performance and accuracy are undoubtedly important for developing ML solutions. However, without deployment into live broadcast streams or integration with esports coverage, their impact remains limited. Even the most accurate models would fail to contribute to enhancing the audience experience unless applied within the broadcasting ecosystem. Many studies suggest that ML models can significantly enrich esports narratives [22, 59, 29, 168, 98], yet few explore how these models could be effectively implemented in live broadcasts [191, 177].

Furthermore, applying ML models to esports broadcasts raises distinct challenges. Models are often trained using data from a specific time frame, which can limit their effectiveness when games are updated through patches. This raises concerns about their long-term viability for broadcast use [174, 166, 8]. Additionally, ML models typically rely on consistent access to detailed input data [97, 161, 198]. Such data may not always be available in live esports coverage due to technical constraints or tournament regulations [10, 126].

Another challenge arises from differences in performance between professional and amateur players. Models trained on amateur data often perform worse with professional players, necessitating the use of professional datasets for training and evaluation [22, 141, 59]. However, professional datasets are typically smaller, further complicating model development.

It is also worth noting that some AI/ML work in esports focuses on game AI, such as optimising player agents or analysing strategies, rather than broadcasting applications [175, 7]. While valuable, these approaches fall outside the scope of this thesis, which is centred on enhancing live esports broadcasts.

Conversely, esports audiences demand engaging and diverse viewing experiences [21, 55, 10, 126, 18, 17]. Data-driven storytelling has proven effective in delivering engaging narratives within esports [82], and ML has shown strong potential to provide the data-driven insights necessary for such storytelling [191, 94, 168, 140, 186]. This demonstrates the untapped potential for applying ML to enhance esports broadcasts, despite the current gap in research.

This thesis aims to address this gap by studying how ML can be applied to enhance broadcast narratives in esports. By examining various facets of ML application in live coverage, the project seeks to expand the reach and impact of ML research in esports, ensuring future findings can be more readily integrated into real-world contexts.

The thesis focuses on applying ML to professional esports broadcasts, using Dota 2 as a case study. Section 1.2 defines the key terms and describes Dota 2, a popular title chosen for its accessible datasets and significant academic interest [174, 59, 94, 38, 35, 197, 95]. Dota 2 serves as an ideal environment for ML research, with insights and resources that may also apply to similar titles like League of Legends.

While the framework developed in this thesis are framed broadly to support ML applications across esports, certain findings may be more directly applicable to Dota 2 and similar games. The goal is to provide a foundation for developing ML models that seamlessly integrate into live esports coverage, enriching broadcast narratives and enhancing audience engagement.

1.2 Terms and Background Domain Knowledge

This thesis explores the ways in which ML can be used in the broadcast of esports to enhance narratives. In order to aid in contextualising the reader, this section provides descriptions of some of the background domain knowledge that can assist with understanding the work presented. In addition, this section also provides definitions for some of the terms used throughout this thesis. Note that Appendix A provides a comprehensive description of specialists' terms that may be used in this thesis. The terms and descriptions provided in this section refer to how they are understood and utilised within this thesis.

As this work focuses on esports, it is important to understand what is meant by the term. Esports refers to the electronic sports domain. In particular, the act of playing a video game competitively. The broad term refers to any instance where a video game is played competitively. Furthermore, the esport title being played refers to the specific game being played. There is a wide range of esport titles available in the market, this can include titles such as League of Legends[1], Dota 2[2], Valorant[3], Counter Strike[4] and many others. This thesis focuses on the title *Dota 2* throughout the proposed case studies. As such background knowledge on the title is also provided (see Section 1.2.1).

Moreover, it is important to note that the spelling of the term can sometimes be found in varied forms, such as "e-sports", "Esports" (even when in the middle of a sentence) or "eSports" The general accepted spelling of esports through academia is "esports" with no hyphen and written all in lower case, unless proceeding a full stop [37] (in which case the leading "e" is written in upper case). This spelling is adopted throughout the rest of the thesis.

Esport broadcast refers to the broadcast of esport. In other words, an

[1] `https://www.leagueoflegends.com/en-gb/` (accessed 13-05-2025)
[2] `https://www.dota2.com/home` (accessed 13-05-2025)
[3] `https://playvalorant.com/` (accessed 13-05-2025)
[4] `https://www.counter-strike.net/` (accessed 13-05-2025)

esport title is played by any number of players, the game footage is then broadcast to audiences, who are not playing the game themselves, but instead consuming the video and audio footage as a form of digital media for entertainment purposes. While not limited to, esport broadcast is typically provided in a live format - i.e. audiences are watching the game as the players are playing. While pre-recorded matches can be accessed and do also gain popularity, this is typically an on-demand version of the same live content. This means that the esport broadcast was first performed and delivered live to audiences, and the event and delivery is also recorded. This recording is then made available to watch as a video on-demand (VOD) at any point afterwards, as a video log. However, this thesis focuses on the live broadcast of esport, as this is the most conventional form of esport broadcast as well as the typical starting point for VOD versions.

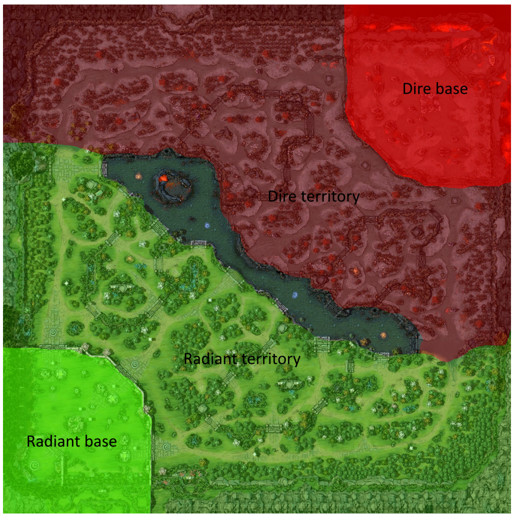
Furthermore, there are varying degrees of proficiency of esport broadcast. This refers to both the skill level of the players as well as the degrees of production placed on the broadcasting itself. For example, highly produced esport broadcast will typically consist of both player footage and commentary done by professional broadcasters [72]. The commentator's role is to both describe what is happening in the game and entertain the audience throughout the match [21]. Professional esports is typically played in a tournament or league setting, where the broadcasting is highly produced, and will include monetary strategies, such as adds rollouts and sponsored content. This thesis will primarily focus on professional esports, as this form of broadcast generally has the highest intensive for integrating and producing more and more engaging experiences.

Similarly, the esport broadcast narrative refers to the storytelling provided alongside the coverage. As professional esport typically includes commentary that creates a narrative [21, 10], this thesis aims to enhance the narrative that is created through storytelling provided by the commentator [17].

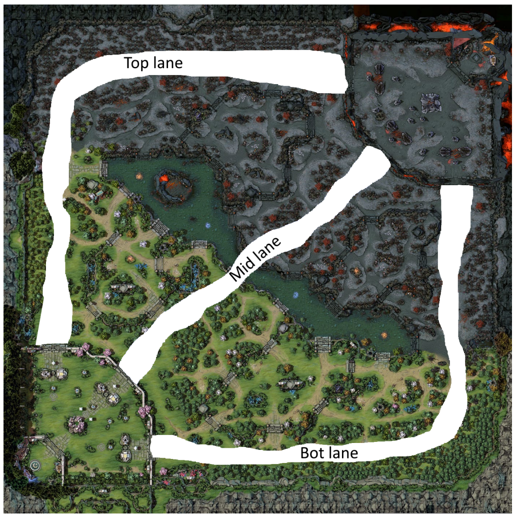
1.2.1 Dota 2

Dota 2 is a Multiplayer Online Battle Arena (MOBA) game in which, two teams of 5 players compete for several in-game resources with the objective of destroying the main structure located in the opponent's base. The two teams in Dota 2 are named "Radiant" and "Dire", which are colour coded throughout the design of the game as green (Radiant) and red (Dire). The map of Dota 2 has a quasi symmetry diagonally, with the Radiant base located at the bottom left corner of the map, and the Dire base on the top right corner of the map as depicted in Figure 1.1a. Additionally, the map is split into three main paths, called lanes, commonly referred to as the top, mid (middle) and bot (bottom) lanes, also depicted in Figure 1.1b.

Dota 2 is a top-down game, in which players control their individual hero. Each character can only be picked by one player, who will only play that character for the entire game. At regular intervals of time, computer-controlled units called "creeps" will spawn on both sides of each lane (i.e. the Radiant and the Dire side). These units will follow the path of the lanes towards the opponent base and will attack any enemy units along their path. Lanes are protected by structures called "towers" that determine each team's territory. Towers, like creeps, will attack any enemies within range, however



(a) The Dota 2 map with both team's territory and base highlighted



(b) The Dota 2 map with each of the three lanes highlighted

Figure 1.1: A depiction of the Dota 2 map

unlike creeps, towers are static. Each tower in a lane can only be destroyed if the previous tower on that lane has already been destroyed (starting from the one nearest the enemy base). Once all towers in one of the lanes have been destroyed, the main structure (the Ancient) can then be destroyed. Destroying the enemy's Ancient is the main objective of the game, therefore the team who manages to destroy the enemy's Ancient first wins the game. Figure 1.2 depicts the towers and Ancients.

In the game of Dota 2, players must select from a pool of characters called "heroes". At the time of writing there are *125* unique heroes available[5], each containing their own set of abilities and attributes. Furthermore, heroes are divided into 4 groups, or categories, based on their primary attribute. There are 3 possible attributes within Dota 2, which are "strength" (str), "agility" (agi) and "intelligence" (int). Heroes are then divided with either one or none

[5]`https://dota2.fandom.com/wiki/Heroes` (accessed 13-05-2025)

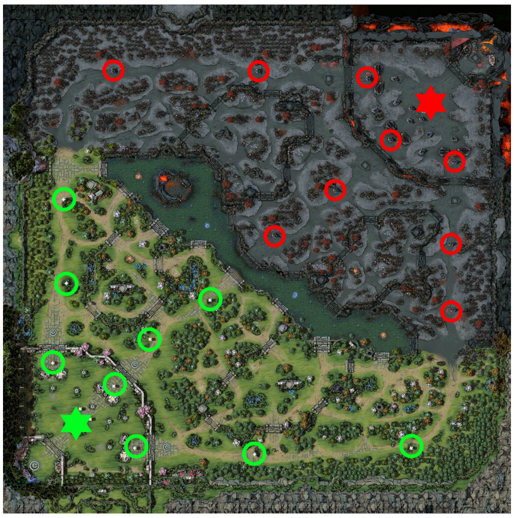


Figure 1.2: A depiction of the Dota 2 map's main structures. Towers are circled and Ancient are depicted with a star

of those attributes to be their primary (i.e. one of str, agi, int or none of them). Heroes that are assigned no attribute as their primary are refereed to as "Universal" hero, conversely, heroes with a primary attribute are called by the attribute name (for example, a "strength hero").

Attributes[6] are represented in game with numerical values, which outlines how much of each attribute a hero (or player) has at any given moment in a match. The amount of strength a character has determines their maximum health and health regeneration (the rate at which any lost health passively restores over time). Similarly, agility is used to determine the amount of armour (a resource that reduces incoming damage) and attack speed (the rate at which a hero can perform an attack action). Lastly, intelligence determines the amount of mana (the resource needed to utilise, or cast, spells and abilities) and mana regeneration (the rate at which any lost mana passively restores over time). The primary attribute of a hero determines which

[6] https://dota2.fandom.com/wiki/Attributes (accessed 13-05-2025)

attribute benefits that hero the most, proving an additional unit of damage for each point in that attribute (Universal heroes gain 0.7 for every point in each attribute).

In addition to their primary attribute, heroes in a match will typically fulfil a role in the team composition [29]. Roles are typically refereed to as positions 1 through 5, which can generally identifies the priority of resources (i.e. the player at position 1 has highest priority of resources while the position 5 has the lowest). Another name used to refer to the positions are the carry, core and supports; where carry typically refers to the positions 1 and 2, core refers to position 3 and support typically refers to positions 4 and 5. Unlike primary attributes which are determined by the game design itself and is coded as a part of the game, positions are community determined and are merely linked to the way in which players choose to play the characters within the team. As a result, the same character may be played in different positions depending on the other characters in the team and the individual player preference.

Every match of Dota 2 begins with the draft phased. This is the character selection part of the game. The most commonly played game mode in esport tournaments is the "Captains Mode", in which teams have one of the players as their designated captain who will choose 5 characters for their team from the list of available heroes. In addition to selecting the characters that their team will play, the draft phase also allows captains to ban heroes. Banned heroes are removed from the list of available characters for that match, which prevents either team from selecting them for that particular game. The order in which captains pick/ban characters is predetermined by the game, while tournament structures will typically determine which team will be the first to pick/ban and which team will play each faction (i.e. Radiant/Dire). Once both captains have selected the 5 characters for their team, characters are allocated within the team players and the main phase of the game begins.

As the game phase begins, players start with very limited resources and attributes. Attributes can be improved by levelling up, or by buying in-game items. Levels can be acquired by accumulating experience (exp), while items can be purchased with in-game currency called gold. Both exp and gold can be acquired by defeating enemy units and structures, particularly creeps, towers and enemy heroes (i.e. enemy players). When a player character is defeated (i.e. killed), the character is temporarily removed from the game until it re-spawns in their base. Killing an enemy player will yield a large exp and gold reward. Additionally, during the death period, that player will be unable to acquire any gold or exp until they re-spawn. This places a high importance on killing enemy characters and surviving confrontations or encounters [168, 146].

Lastly, Dota 2 is an imperfect information game. This means that players do not have access to all game state data at all times. This is achieved through the Fog of War (FoW) mechanic, where information about entities in an area is hidden from players unless a friendly unit is present. To allow for information gathering the game supports a vision mechanic, through an in-game item called "wards" [22]. Figure 1.3 demonstrates the mechanic. In this figure, the left-side image demonstrates an area without vision, depicted with a darker shading. No information about the state of the game, or the units present in the area is given to the players. The same area is then shown on the right side of the figure, where a ward has been placed. This outlines the presence of an enemy unit that was previously hidden. Wards can be destroyed by the enemy team, in a process called de-warding. Additionally, as the FoW mechanic directly impacts the visual information a player can extract, warding and FoW are generally jointly called "vision" within Dota 2.

Figure 1.3: A depiction of the FoW & warding mechanic in Dota 2.

1.3 Research Questions

This thesis seeks to address the main research question (RQ) "How can ML be applied to enhance esport broadcast narratives?" In order to investigate this RQ, the related academic literature is explored. This is done to understand the current state of the art of the available knowledge. Therefore, in order to address the main RQ, a set of sub-RQs are posed.

To begin with, the current state of the art of the related academic literature is explored in RQ1 "how is ML currently being used within the esport broadcast domain for storytelling and broadcasting narratives?" Through the process of addressing this RQ1 three clear gaps in the knowledge available in the literature are identified. These are then used to inform the three subsequent RQs, which are investigated in this thesis. For ease of parsing, the main RQ and the four subsequent RQs are listed below:

- Main RQ: How can ML be applied to enhance esport broadcast narratives?
- RQ1: How is ML currently being used within the esport broadcast domain for storytelling and broadcasting narratives?
- RQ2: How can ML models sustain performance and functionality beyond initial game patches?
- RQ3: How can ML models be integrated with the live coverage of esports?
- RQ4: How can ML models enhance esport broadcast narratives?

By addressing these four sub-RQs, this thesis then reflects on the findings in order to answer the main RQ, providing knowledge of how to apply ML to enhance esport broadcast narratives.

1.4 Thesis Structure

This thesis is composed of 7 chapters. Firstly, this chapter outlines the research motivation and provides background contexts to aid the reader. Then, the current state of the ML and data-driven storytelling literature for esports is outlined in Chapter 2, which provides the necessary context to address RQ1. While addressing this RQ, three clear gaps are identified within the current state of the art in the academic literature in regards to the application of ML models into esport broadcast, particularly in relation to narratives and storytelling. This informs the need to address the subsequent RQs. Chapter 3 then investigates the way in which ML models can be used beyond the original patch in which the model was trained in. This seeks to address RQ2. RQ3 is investigated in Chapter 4, which provides insights into the design considerations that allow for pre-existing complex systems (such as pre-trained ML models) to be integrated and utilised within the live coverage of esport broadcast, as well as reflecting on the impact this has had into the esport narrative. This knowledge is then taken further in Chapter 5, which addresses RQ4, investigating how ML models can be designed to have a positive impact in the broadcast narrative.

Chapter 6 then compiles the knowledge learned from the previous chapters into systematic methodological framework. This framework reflects on the empirical findings outlined in this thesis aimed at ML pracitioners to integrate the findings of this thesis, on how to apply ML to enhance esport broadcast narratives.

Lastly, the thesis is then summarized and concluded in Chapter 7. This describes how the research goals proposed in this chapters are achieved throughout the thesis. Additionally, Chapter 7 also reflects on the limitations and future work that can be perused as a continuation of the contributions described.

2 Literature Review

This thesis seeks to explore the use of ML to enhance esport broadcast narratives. In the previous chapter, the motivation for this research is outlined and some of the contextual domain knowledge is explained. In this chapter, the body of work of the literature is investigated.

In particular, this chapter seeks to outline and critically evaluate how ML is currently being studied in the context of esport broadcasting and how it impacts storytelling and narratives. In doing so, this chapter seeks to address RQ1; by providing an analysis and critique of the current State-Of-The-Art (SOTA) in the associated research domains, define the gaps in the literature and synthesize the SOTA. These gaps of knowledge in turn inform the compilation of three RQs which are explored in the following chapters. These ultimately permit us to address the main RQ of how to apply ML to enhance esport broadcast narratives.

This chapter is constructed as follows:

This chapter first defines the knowledge structure of the relevant literature (Section 2.1), which outlines the various disciplines of interest, as well as how and where knowledge may be available beyond academic literature. Section 2.2 details the search strategies within this review of the literature. This includes the search criteria with a detail overview of the query string utilised, the databases investigated as well as the additional refining criteria employed to allow for a relevant and comprehensive review of the academic literature. The literature is then presented in the following sections. Lastly, this chapter concludes with a knowledge synthesis (Section 2.6), which reflects on the current SOTA, the details of the identified knowledge gaps as well as how this thesis seeks to address these gaps.

2.1 Knowledge Structure

Esports research encompasses a wide range of fields and academic disciplines [37], including player psychology [80, 12, 182], education [41, 193, 64], monetisation [128, 2, 99], gender studies [53, 75, 131] and many others. Within the wider cross-disciplinary literature which either directly focuses on esports, or uses esports as a research tool, this thesis focuses on the fields of AI/ML, data science, game analytics, Human Computer Interaction (HCI), data visualisation and Computer-Supported Cooperative Work (CSCW).

Scientific domains such as AI/ML, data science and game analytics provide a wealth of technical and computational knowledge [174, 178, 147, 140] in relation to how esport data can provide meaningful insights over what happens in a game. Similarly, traditional sports broadcast shares several constraints, considerations and goals to esports broadcast. Therefore, sports analytics can complement esports analytics to inform how insights can be applied to esports. Collectively, the fields of AI/ML, data science and game analytics (both esports and traditional sports) encompass the SOTA for how ML, and data more broadly, can be used to generate insights into esports.

Furthermore, HCI, data visualisation and CSCW investigate how humans perceive, process and utilise computer systems and data more broadly. Within esports, these fields provide insights over how audiences, broadcasters and other stakeholders utilise computer systems. In other words, while fields such as AI/ML can inform how to use data to generate insights, disciplines such as HCI can inform how these insights can be applied into systems that enable users to benefit from the insights.

Therefore, the combination of these disciplines are needed to understand the relevant SOTA of knowledge. This knowledge is valuable to understand how complex systems (such as ML powered solutions) can be applied to successfully enhance esport broadcast narratives, in particular in live settings. As such, this chapter will investigate these disciplines to outline the state of the art available in the academic literature with esport broadcast and its narratives.

It is important to note, however, that due to financial incentives present in the esports industry [111, 113, 114], a large part of the SOTA of knowledge encompass industry-lead research. As a consequence, a lot of the existing knowledge in the domain is not publicly available. While some of the literature investigates and contrasts audience facing solutions that are commercially available, much of what drives these solutions is inaccessible for open academic research, particularly in relation specific to AI and ML techniques and methodologies that are deployed in them. While this thesis endeavours to learn from the full SOTA available in the domain, only publicly available sources can be investigated. Despite this limitation, this chapter addresses how ML is generally studied to be used within the esport broadcast domain and its implications on storytelling and narratives.

2.2 Search Strategies

In order to identify the relevant literature, four main criteria were devised (C1-4).

- **C1** : Publications that consider esports
- **C2** : Publications that consider audiences
- **C3** : Publications that consider machine learning
- **C4** : Publications that consider storytelling or narratives

When combined, all four criteria provide a curated subset of the literature that encompass the SOTA for the relevant knowledge. In particular, C1 provides the initial and broad framing on esports. This is then further curated by C2, which introduces the need for audiences. This is done in-line with the esport broadcast definition provided in Chapter 1, which necessitates the presence of audiences, making a distinction between esports more broadly to esports broadcast. As this chapter aims to address RQ1, C3 introduces the need for ML work in the relevant explored literature. This ensures the SOTA of how ML is used within esports broadcast can be explored. Similarly, C4 is necessary to ensure that the SOTA explored in this review relates to the way in which ML is used for storytelling and broadcasting narratives. Therefore, the four criteria can then be used to outline the relevant literature needed to investigate RQ1 "how is ML currently being used within the esport broadcast domain for storytelling and broadcasting narratives?"

To identify relevant literature that meets the four criteria, a set of search sub-queries was developed. These sub-queries were formulated by reviewing existing literature known prior to the review, focusing on similar words and terms commonly found in relevant sources. The final search terms for each criterion are as follows:

- **C1** : (esport* OR "competitive gam*" OR e-sport*)
- **C2** : (audience OR viewer* OR spectat* OR broadcast*)
- **C3** : ("machine learning" OR ml OR "artificial intelligence" OR ai)
- **C4** : (storytelling OR narrative* OR commenta*)

These subqueries were then combined with *AND* operators in the form (*C1 AND C2 AND C3 AND C4*) to be used as a search query in four distinct databases. The four database utilised in this search were the ACM Digital Library, IEEE Xplore, Web of Science, SCOPUS. Table 2.1 outlines the total results obtained in all databases.

In order to further refine the search results, additional criteria were employed. Only peer reviewed conference and journal publications from the years of 2014 to 2024 written in English were included. This ensured that only publications of the last decade were present in this review, to more accurately reflect the current SOTA.

Furthermore, some fields and topics of research were excluded from this review, as they fall outside the scope of this thesis. These are generally grouped into education & serious games, representation & wellbeing, and monetisation. The full exclusion criteria used in the search can be found in Appendix B.

Table 2.1 outlines the number of publications identified on each of the databases before and after the additional criteria were applied. Note that publications which fall under the exclusion criteria but were not automatically removed through the query were manually excluded from this review. This can happen if the database does not correctly identify or categorize publications as under the relevant fields and thus the publications are still listed in the result from the query. Each publication was investigated sequentially and individually, if a particular publication was found to be under the exclusion criteria based on topic, language or year of publication it was excluded from the analysis provided in this chapter. Additionally, any work that has been performed as part of the contribution of this thesis was also excluded from this review and it is instead covered in detail through the contributions of this thesis (see Declaration).

Table 2.1: Number of results for each database after applying the exclusion criteria.

Database	Total results	Post-exclusion
ACM Digital Library	172	157
IEEE Xplore	3	3
Web of Science	5	5
SCOPUS	397	214

2.3 Previous Reviews

Following the search strategy described in Section 2.2, the resulting set of publications identified included a range of previous literature reviews. Therefore, this section present the relevant literature reviews identified that are already available in the wider literature. In particular, the main points outlined in those reviews are presented. However, these identified reviews have varying focus, which do not fully address RQ1. For this reason, this chapter also investigates the remaining of the literature identified following the search strategy described above. This is then presented in the next two sections.

Tuyls et al. [170] provides a review of how ML and AI is applied - not to esports - but to regular sports, specifically football. This review shows several techniques and ways in which AI is applied to football and the benefits and consequences of it. Within football, AI has been shown to aid in many areas, from training analyses tools for player performance improvements to live and post game analysis tools for commentators and audiences. Importantly, this review poses many instances where ML is successfully integrated into the sports broadcast narrative, such as how Automated Video Assistant Coach (AVAC) [163] has revolutionised the way in which broadcasters discuss in-match events. The football case-study is an example where audience and commentator-focused innovation enabled ML to have a meaningful impact in the broadcasting narrative, ultimately enhancing the audience experience [170]. AVAC as an example has enabled broadcasting narratives to be built that were previously unfeasible. The ability to automatically replay moments of importance in the match aids in explaining complicated scenarios, and enable broadcasters to identify multiple narrative features that may be missed in real time if they are not presented with the instant replay. Thus integrating AVAC into football broadcast has enhanced the way in which storytelling is done by commentators. This exemplifies the importance of conducting narrative-focused research and investigating effective ways to apply such techniques to the live coverage of esports.

By contrast, two esport focused literature reviews investigate the use of

AI and ML into MOBAs specifically [51, 24]. In both cases, the literature outlined persistent emphasis on performance improvement and innovation. While this is a crucial aspect of ML and AI research, very limited insights into the real-life application of such techniques in any situation (whether broadcasting, streaming, betting or team informatics) can be found in this literature.

Furthermore, the review by Guzmán and Medina [51] has identify a major limitation in relation to the application of such models into live broadcast. The review investigates the input features that are commonly used within the AI/ML MOBA literature, revealing that many of the features utilised in the current SOTA are not accessible live, which prevent many of these models to be incorporated into live broadcast. Moreover, the authors also outlined that there is an inconsistent use of input features even when models perform the same or similar tasks. The authors conjecture that this inconsistency suggests that ML practitioners are generally designing models based on their own understanding of the environment rather than following a data-driven approach to feature engineering. Three notable exceptions identified by the authors were that of character selection, in-game items and the character roles within the team composition, which were more consistently utilised as inputs for ML models. The authors outlined that these features are highly mutable in the context of the game patch, with game developers reliably altering the fundamental design of characters, abilities and many other aspects of the game. As such, Guzmán and Medina emphasized the importance of representing the characters and the items in a patch-aware format, that can allow for patch-agnostic analysis for better longevity and model performance.

Both reviews [51, 24], however, identified a strong emphasis on model performance as the primary evaluation metric within the relevant literature. Limited to no reflection of how models can be integrated into the live broadcast was identified. Additionally, the reviews did not investigate how the application of such models could impact the broadcasting narrative, as this was not the focus of those reviews. No additional review that focuses on the direct application of ML for esport broadcast could be found. For this reason, this chapter will review the ways in which ML technologies and ML powered solutions are (or can be) applied to the live coverage of esport broadcasts with a critical reflection of the impact this has (or could have) in the broadcasting narrative from a storytelling perspective.

2.4 Understanding user needs

Lessel et al. [88] explores viewer perceptions and preferences when watching livestream games. This outlines how viewers tend to prefer the use of interactive elements, as long as they do not interfere too heavily with the streamer's game performance. Importantly, this applied to both active and passive viewers, who the authors described as the viewers that do not seek to engage with the livestream directly, be it in the live-chat or through social media platforms outside of the livestream itself. Additionally, the authors also concluded that several aspects of games livestream that are common place in platforms are generally not well received by audiences, for example, while audiences are generally favourable to streamer-created interactive pulls, the ability for audiences to create or stop their own pulls is not perceived positively. The authors also outlined that, in regards to elements related to the screen/audio composition of the stream, the use of game specific overlays that provide explanations and additional information about the game was the most well received component of games livestream across audiences. Such overlays are directly applicable to esports, both built in to the stream itself [126, 10], or presented as third party interactive dashboards [18, 191] as well as companion apps to be used alongside streams [82]. This is a clear example of the identification of audience needs in relation to broadcast narrative and how data-driven storytelling can aid in addressing this need within esports.

Robertson et al. [142] explores how commentary is delivered in esports broadcasts, specifically focusing on Dota 2. The study highlights that expert broadcasters primarily describe and explain *what* is happening in a match. However, the authors note that research in Explainable AI has generally shown users prefer *why* explanations [92, 93]. To investigate this apparent contradiction, a controlled experiment was conducted with three groups:

- One watched a Dota 2 match without additional tools
- Another used an interactive map providing *what* explanations
- The third used a map offering detailed *why* explanations

All groups were asked to watch a Dota 2 match, followed by a recall questionnaire. The study concluded that while *why* explanations are effective for understanding complex systems, they impose too high a cognitive load for live esports coverage. As a result, audiences were unable to fully benefit from detailed information presented in real-time. The authors also observed that combining multiple modalities (e.g., audio and visual cues) helped viewers process information more effectively. This work underscores the importance of aligning live esports coverage with audience cognitive capacity, favouring *what* explanations over *why* explanations.

This preference for *what* explanations has been corroborated in other esports contexts. For instance, Penney et al. [127] examined how domain experts, such as commentators, use AI tools to provide explanations in StarCraft, an RTS esport. They similarly found that *why* explanations were cognitively costly, even for experts, due to the complexity of these games.

Further evidence comes from Rijnders et al. [139], who studied player cognitive capacity through the use of a live companion application in both digital and physical formats. Although this study was not aimed at storytelling or broadcasting, it investigated how such tools could improve player performance during gameplay. The authors raised concerns about fairness, noting that such systems might be considered cheating depending on game terms of service. Nevertheless, their findings also highlighted the challenges of processing dense, real-time information, particularly in the physical modality, where the tool was less integrated into the playing environment.

The consistency of findings across different genres [142, 127] and modalities [139] demonstrates the general cognitive limitations of esports audiences and participants when absorbing live information. This highlights the need to carefully design ML-powered solutions and applications that enhance live broadcasts without overburdening audiences or broadcasters.

One example of a proposed ML powered solution is in anomaly detection proposed for League of Legends [149]. This paper explores how archetypal analysis - a form of unsupervised ML that identifies extreme examples on a population for classification - can be used to detect and classify important moments in a match that are unexpected. This is done with the intent to be a broadcasting tool, to aid commentators in identifying important moments in the match as well as have better opportunities to understand and create narratives that are engaging and enhance the audience experience through storytelling.

The authors utilise dynamic time segments, divided based on important in-game events (in particular team-fights). Each time segment would then be represented by a representative aggregate of telemetry data, such as the number of kills, or experience gained by a team. By comparing the performance of each segment, the authors first trained an archetypal classification of segments which informs what is happening in the game. Additionally, the classification can also highlight outliers in how players are playing the game, which can provide useful insight for broadcasters to build narratives. This can be done by both identifying the frequency in which archetypes are detected in the training data, as well as comparing the specific features in the data distribution. For example, if an aggregated feature such as experience gained is above the 95^{th} or below the 5^{th} percentile it could be considered an anomaly. Conversely, if the archetype itself is not commonly observed in the training data, it can also be considered an anomaly.

By comparing the performance of previously unseen data with the pre-trained (offline) model described by the author, a new segment can be classified as one of the existing archetypes. This classification, and whether or not it is an outlier (i.e., an anomaly), can then be made available to broadcasters who can utilise their intuition and game expertise to build their narratives.

The interpretation of the archetypes themselves (i.e., how they were named and what they mean in relation to game contexts) was done with consultation with game experts. However, it is unclear if professional broadcasters were included in expert consultation.

The author evaluated the work in a complete (not live) game, but they conjecture that the same approach can be applied in a live scenario to perform the archetype analysis and detect anomalies. Additionally, Sifa et al. [149] reflected on how game changes can impact the way in which players play the game [86]. The introduction of changes to the core game design can lead to fundamental changes game balance [129], which impact the way players interact with the game [81], altering their strategies and what is common or not common for the given patch. In other words, the player community identifies what is considered to be the optimum way of playing the game for any given patch, which is then called the "meta" [166]. This could alter what is considered an anomaly for future game patches, as well as change the distribution of data in the space [174], which could have an impact in the archetypes directly. The authors address this by posing that the model can be reactively updated to fit with the new changes. While this could lead to sustaining the performance of the model through a patch in its entirety, it is important to note that this is a reactive solution that relies on enough games to be played in the new patch to allow for an update in the model. During this data-collection period, insights provided by the model may not be a true reflection of the anomalies in the patch.

Lastly, although Sifa et al. [149] offer valuable reflections on how this model could be applied to the live coverage of esports and how it could impact the broadcasting narrative, it is not clear what the impact of the models is in relation to actual commentators. The archetypes identified were entirely data-driven with post-hoc interpretations by domain experts. This is important as it offers largely data-driven insights, avoiding biases and providing an objective definition of what is happening in the game. However, commentators may already have their own methodologies and strategies for talking about the game and building narratives which may not, necessarily, benefit from how the archetypes are presented, particularly if broadcasters were not directly involved in the interpretation of the archetypes. For example, viewers rely on player positioning to understand the game context and use player movements to interpret their decision-making [78]. If an archetype detected does not match a broadcaster's interpretation of player positioning and movement, it may force commentators to provide post-hoc interpretations even when they do not agree with what is highlighted. This could have a negative impact on the broadcaster's strategy to narrate the game, particularly as they are already following a cognitively arduous environment [142, 127]. Therefore it is important to ensure that the highlighted insights match broadcasters expectations and convey meaningful information that can be leveraged by commentators to build and expand narratives that suit their storytelling needs.

Despite this, it is clear that the advances posed by Sifa et al. [149] offer great insight and knowledge on ML techniques, as well as how they can be applied to live esport broadcast coverage. This work was later also expanded to Dota 2, following a similar methodology [134]. In this version of the work, the authors outlined how anomaly detection, when applied to the wider playing population, can serve as a form of cheat detection by identifying unexpected performances from players which can indicate performance altering techniques, such as playing on someone's account on a lower/higher skill bracket, the use of cheating software. This wider application - while outside the scope of this project - would not suffer from the same narrative and storytelling constraints detailed in this review. However, it is still important to ensure that the model performance is not affected by the changes to the game environment in patches to ensure results are reliable across the continued development of the game, if applied to cheat detection.

Another approach to applying ML technologies to enhance broadcast narratives has been proposed through cooperative storytelling [187]. Xu et al. explored the use of a pre-trained generative language model to co-commentate an esports match alongside a professional broadcaster. This approach involves providing the language model with varying levels of input, such as the human commentator's remarks (converted to text) and additional contextual information, like team names or tournament details, enabling the model to generate informed follow-up commentary. For example, if the system is being used in a final match of a major tournament, between "Team Sigma" and "Team Alpha", three versions of inputs could be used to enable the language model to provide a follow-up commentary:

- "This was a great fight for Team Sigma, picking up three kills with no deaths"
- Tournament final - "This was a great fight for Team Sigma, picking up three kills with no deaths"
- Tournament final - Team Sigma vs. Team Alpha - "This was a great

fight for Team Sigma, picking up three kills with no deaths"

In this case, the model was evaluated purely on a performance-based metrics on the quality of the language generated compared to the actual (human produced) comments used in the training dataset. This evaluation method may not fully reflect the model performance in the environment, as different broadcasters may choose to talk about different aspects. It is important to consider the purpose and utility of such solutions, while collaborative storytelling is a promising area of research, the work does not offer any considerations to the commentator need. For example, a qualitative evaluation of how human broadcasters utilise the solution, and the impact it had on the audience may provide a more representative evaluation of the model performance. However, the work described also did not offer any considerations of how such solutions could be integrated with a live coverage of esports.

Firstly, the human commentator speech would need to be transcribed live in order to feed into the model as an input feature. This can be computationally expensive and inaccurate, particularly - as the authors outlined - as commentary is often provided in a fast manner, making it difficult for automated systems to detect punctation, which is critical for the quality of the follow up commentary. Furthermore, esports commentary is commonly loaded with technical terms that are related to the game itself, this is another difficulty for an automated speech-to-text solution that must precede the ML-powered commentary.

Secondly, the solution proposed in the work does not account for any game state input, which limits the capabilities of any commentary of in-game play-by-play to be purely reactionary to what the human caster comments on. This prevents the model from offering any additional insights that the human caster does not already offer.

Thirdly, the broadcaster may not be accustomed to collaborating language model commentator, as this was not designed to cater for their needs it is unclear if this solution would benefit the narrative flow, potentially leading to interruptions and other forms of disruptions between human and AI commentary. Of course, this could be mitigated with human training, to allow broadcasters to learn how to produce commentary in this collaborative fashion, however it is unclear if this would be beneficial or if this solution would be positively received by professional commentators.

2.4.1 Data Visualisation

Other authors have investigated different aspects of user needs in regards to esport broadcast narratives. Data visualisation as a broadcaster tool has been explored as an alternative way of applying data-driven insights into esports commentary. One of the first instances of visualisation tools within academia includes Block et al. [10], who investigated the use of historical performances and integrated it into the direct feed of esport coverage. In this work, the authors created a tool for parsing a current live game of Dota 2, and compare a large set of KPIs with previous performances for all players in a match. This tool was then deployed to the commentators who could choose to integrate the insights into their narrative. By using this tool, broadcasters could create a "narrative bite" which displayed the performance metric in a KPI to the broad audience as part of the feed being broadcast. This is exemplified in Figure 2.1, which is extracted from the original authors' work (Block et al.

[10]) for demonstration purposes.

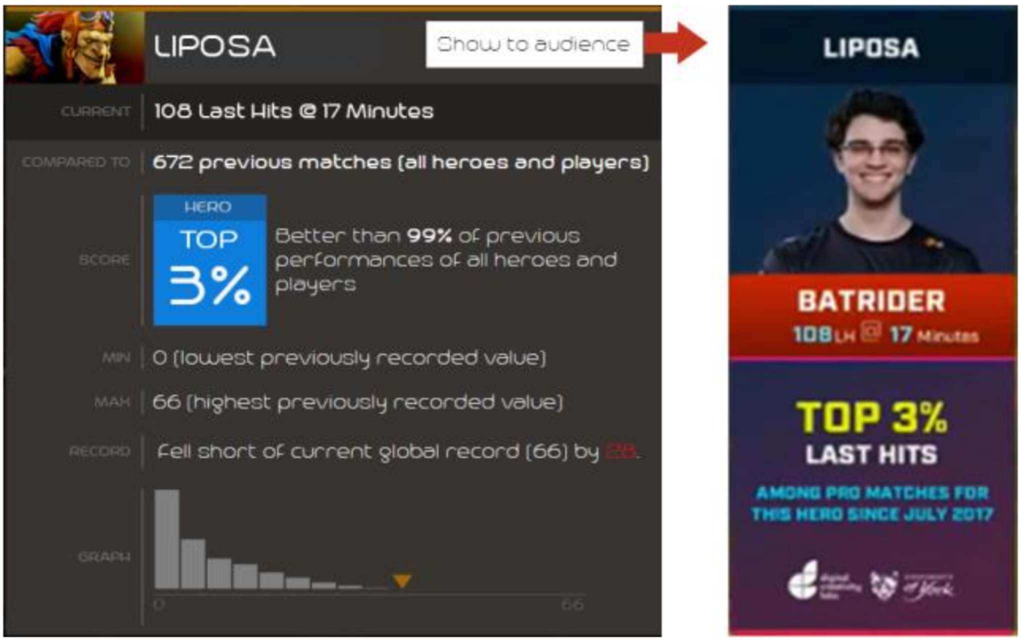


Figure 2.1: An example of how broadcasters could outline KPI metrics by integrating a graphic into the linear feed [10], originally labelled as Figure 3 in the source material. Full authorship of the figure belongs to Block et al. [10] and is used with permission from the authors.

As depicted in Figure 2.1, and discussed by Block et al. [10], the use of the tool provides full control of the visualisation to the commentators. This includes the KPI to be displayed, the time at which the metric is shown and the duration for which the graphic is visible to audiences. Additionally, commentators had the option to integrate the use off the tool into their broadcast, or omit it, given them the freedom to gain insights from the tool without actually utilising the graphic. This complete freedom to operate enables an easier integration into their existing systems, without having to compromise any narratives they deem relevant and instead enhance the storytelling with the use of data.

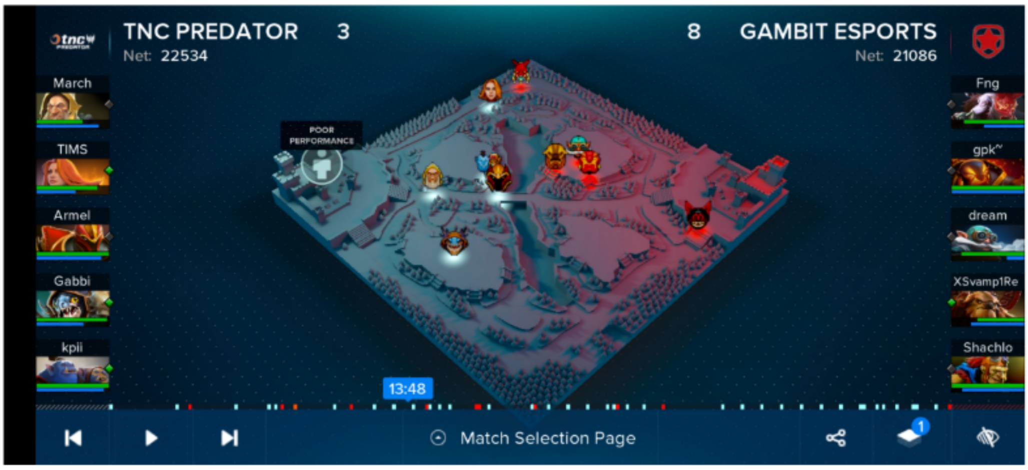
Furthermore, the design of the graphic generated by the tool can provide further insights. This info-graphic contains a relatively small amount of information, and - when shown to audiences - only obscures a small area to the right of the screen. This unintrusive design was created to ensure the main gameplay coverage is not disrupted, allowing audiences to gain insights from the graphic without interfering with the immersion into the game footage.

Another noteworthy aspect of this design is to the ranking of the KPI as a form of qualitative percentile cue. In the example graphic depicted in Figure 2.1, the ranking is shown as a "Top 3%" performance. This text is given a unique font colour and also contains the largest font size of all of the text in this visualisation. That also places a significant amount of importance in this aspect of the image, which is expected as the ranking itself is the key messaging of the graphic. This percentile is further expanded on by the description of the KPI itself, which is written in full underneath the performance text. In addition to the core information displayed, the authors also included a second level qualifier that added further context to the narrative item. This was done to give the option to broadcasters as well as more keen viewers to explore this story in more detail, however the authors did not expect the majority of the viewers to actively consume this additional content.

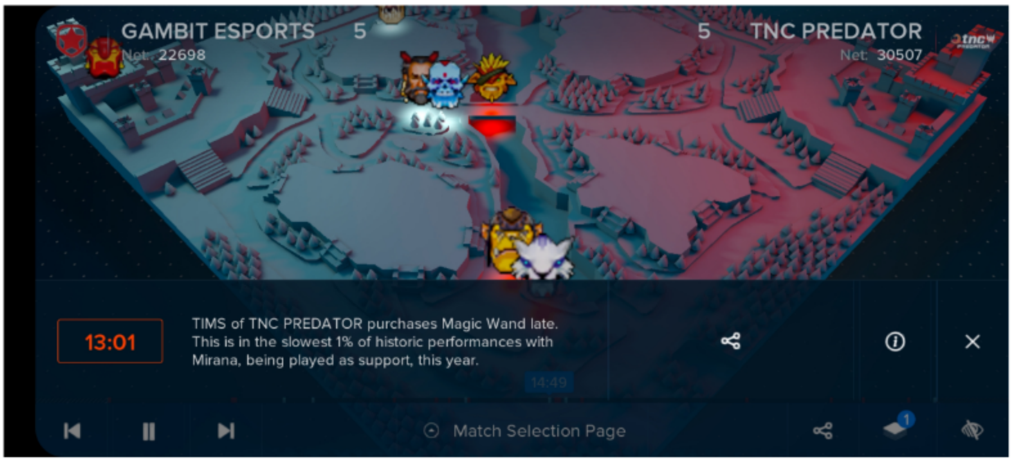
Lastly, the graphic generated by this tool also contains branding information. This is done in the form of the researcher's affiliation, but that outlines the opportunity for tournament organisers to incorporate sponsorship branding into the graphic. This form of sponsorship and advertising has been shown to have positive impact into the brand recognition without negatively impairing in the audience experience [121, 5]. This is an important design consideration as the majority of the revenue for esport tournaments are derived from sponsorship [111]. Therefore integrating effective and unintrusive forms of advertising can lead to the successful application of such tools that can enhance the audience experience without compromising the overall user experience.

In addition to adding the visualisation into the broadcast feed directly, the authors' subsequent work has also explored the use of a companion app for visualisation [82], taking the tool into the audience directly. In this work, Kokkinakis et al. [82] have explored immersive data driven audience experience. This companion app explored a wide range of spatial-temporal displays, placing the playing-map at the forefront of the display with events being added to their respective locations and giving the users the ability to scroll through time and rewind what has happened before hand. Figure 2.2 contains a depiction of how users are presented with spacial-temporal information, which have been extracted form the original publication by Kokkinakis et al. [82].

Figure 2.2a depicts how the main screen of the app places the *Dota 2* map in the centre of the display. The location for all characters are also updated in the map to match the actual player location for any given time. Users can watch the game live, or scroll through time with a time bar at the bottom. The time bar itself also presents viewers with additional information, highlighting the game events detected by the app for any given timestamp, allowing users to easily navigate to investigate the different events that hap-



(a) The main screen of the Weavr companion app [82]. Originally labelled as Figure 1 in the source material. Full authorship of the figure belongs to Kokkinakis et al. [82] and is used with permission from the authors.



(b) The Weavr companion app displaying a game event [82]. Originally labelled as Figure 2 in the source material. Full authorship of the figure belongs to Kokkinakis et al. [82] and is used with permission from the authors.

Figure 2.2: A depiction of the Weavr companion app. showcasing the main screen users are presented when using the app (a) as well as how events can be selected for additional information (b).

pen sporadically throughout the match.

In addition to navigating through the match times to highlight events, users are also able to touch - or click - on the events themselves, which will provide them with more information in textual form. This is displayed in Figure 2.2b, in which the location of the event is zoomed into the map, further highlighting the spatial location, the exact timestamp is highlighted on the bottom-right of the app, followed by the description of the event. This description functions as a summary - or a recap - of the narrative story being displayed. Similar approaches using game logs to create textual recaps of game events have also been posed in the literature [181, 117, 85, 98]. Users are then able to close or share the information using various social media platforms.

The design outlined by Kokkinakis et al. [82] was an outcome of the authors extensive investigation into user needs. In describing their findings, the authors highlight the importance of user control. The authors comment how viewers are commonly bound to the observer (i.e. the camera operator), and having the ability to control what info they can see was highly valued by the users. Additionally, users were also observed to utilise the companion app in ways the authors did not expect, such as a review tool for past matches. This was made possible as users had the ability to "rewatch" past matches as if they were live, as well as skip and rewind through the time bar.

Similarly, when reflecting on user feedback, the authors depicted an observed "healthy distrust in algorithms" by audiences [82]. This demonstrates the importance of carefully designing ML solutions, as audiences would commonly question the features that relied in more complex algorithms. One example of this is in the app's win prediction feature, which audiences have demonstrated distrust in the validity of the prediction. This distrust, particularly with win prediction models, form an example of a 'win prediction dilemma' first outlined by Hodge et al. [59] and discussed further in Section 2.5 of this thesis.

Another example of a viewer-focused data-driven storytelling tool is by Charleer et al. [18] who described the process of designing a real-time dashboard for esport audiences of both League of Legends and CS:GO. In this study the author highlighted a few criteria that inform the design process of this dashboard, including general dashboard design choices (as described by Bowman et al. [13]), and based on game specific considerations.

The general dashboard design choices outlined by the authors include a range of considerations. Firstly, temporal usage, in which the authors describe how the dashboard provides continuous information throughout the game, in addition to described game event presented sporadically through the game. This is a similar use case as the one described by Kokkinakis et al. [82], but collapsed into one source view rather than in multiple screens.

The second consideration was of visual complexity. The authors outlined how spectators - who are also players - may have a higher cognitive capacity to consume data, as they often demonstrate a high degree of data literacy. This enables the dashboard to provide a high data density to audiences, and thus their dashboard could be designed with a large volume of data being displayed. This is in somewhat agreement to the findings of [82], which describe that a degree of information density is needed as users are seeking - what the authors described as - the right amount of data. However, while Kokkinakis et al. [82] approach seeks to provide a large amount of data through different areas and navigation strategies, Charleer et al.'s [18] dashboard provides all of the data in a single source, and addresses information overload with separation in design. Notably, there is a distinction in the findings by Charleer et al. [18] - who found viewers are able to absorb high volumes of data - and Robertson et al. [142] - who outlined that *why* questions are too cognitively demanding to be absorbed live by esports viewers. Thus, while viewers may be too cognitively overloaded to digest data in too much depth [142], they may be able to process a wide range of shallower data [18].

The next design consideration posed by Charleer et al. [18] focuses on immersion and integration. This refers to the form in which the dashboard is presented to audiences and it is another way in which the authors tackle information overload. The author describe how their dashboard takes a side bar approach that mimics League of Legend's interactive overlay. By implementing the dashboard on the side of the stream, the main game broadcast can be shown unobstructed, which avoid cluttering the broadcast. Furthermore, by appealing to viewer familiarity to existing design choices, League of Legends audiences may be able to adapt more easily to this design and be able to more readily take advantage of the dashboard.

Following from the design considerations described, Charleer et al. [18] then utilised a range of audience surveys to understand user needs for both games. This advised in the creation of a dashboard, which consists of three main sections providing different information. For the League of Legends dashboard, the three sections provide information on gold advantage. This is a time-series graph outlining which of the two teams have the most amount of gold at any given moment. The second section provides damage information, allowing viewers to see how much damage has each individual player done to the other team. Lastly, the bottom most section provides vulnerability information, outlining which sources of damage are most effective against each character. Similarly, the CS:GO dashboard is also divided into three main sections, with the top most providing information for each player. This includes the player name, amount of kills, deaths and assists (K/D/A) and some other game specific features which were highlighted as important given the user survey. The middle section contains a money distribution graph. Unlike with the League of Legends' dashboard, however, the CS:GO money distribution only accounts for the current state and does not reflect on past states. This was also done to better suit the audience needs for that particular title. Lastly, the bottom most section for the CS:GO dashboard depicts the average damage dealt per player per round. This allows audiences to quickly ascertain the general impact of each player across the whole match, as CS:GO is consists of multiple short rounds that reset and restart, in contrast to the long-form format of League of Legends and other MOBAS.

Both dashboards (League of Legends and CS:GO) were then evaluated in design. In order to enable this evaluation, the authors employed eye-tracking. By inviting a range of participants to watch a match of the respective titles with the dashboard available to them, the authors found that the areas around and including the dashboard were more frequently gazed, which suggests that both dashboards were utilised by audiences. This form of evaluating highlights some of the different ways in which such tools can be measured. If a user is looking at the information being displayed more frequently than - or with similar frequency to - the main game stream, it suggests that the tool is being utilised. Other forms of evaluations have also been explored, such as post game questionnaires and interviews [10] or direct measurement of interactivity [82] as well as user customization analysis [173].

In the three example of data visualisation solutions discussed (Block et al. [10]; Kokkinakis et al. [82]; Charleer et al. [18]), the data being displayed was limited to simple metrics extracted directly from the game. Although in the two Dota 2 examples [10, 82] there was a machine learning powered win probability feature, very little reflection was provided by the authors on this features, with the evaluation largely focusing on the simpler metrics. This demonstrates that even simpler data-driven solutions can lead to enhanced storytelling, particularly when combined with the broadcast narratives [10]. Furthermore, this also outlines the need for continued studies of how ML solutions can be incorporated to enable and enhance commentary and narrative, particularly as it has been shown to have positive impacts in similar fields such as football broadcast [170, 69].

2.4.2 Spectators and Broadcasters

Yu et al. [195] investigated what leads spectators to want to watch an esport broadcast, particularly on livestream platforms such as Twitch. In this explorative work, the authors surveyed 56 participants on the way in which they consume League of Legends esport content. Participants could only be included if they had watched at least one League of Legend esport match in the previous month. Importantly, the majority of the respondents (90%) indicated that they watch esport content at least once a month, with 23% of the participants indicating they watch esports at every opportunity. The survey contained opened questions, with the responses qualitatively analysed through thematic analysis. The authors found that several of the motivations stems from primarily four causes.

1. The competitive nature of esports as well as the tournament formats creates an stimulating environment that draws audiences in.

2. Some audience members are particularly engaged fans that are following the performance of one or more player or team. Their investment on a particular team or player can serve as an intensive to follow esport tournaments live, ensuring they can keep up to date with how their player of interest is performing.

3. Some audience members are also keen to translate what they watch into what they play. These spectators seek to learn from watching professional players in order to improve their own gameplay performances.

4. Audiences are particularly drawn to professional esport broadcast because of the suspense and unpredictability of the games. The emerging narrative present in these games are a key factor that keeps viewers engaged and draws them to watch professional esport broadcast live.

As these results were reached through qualitative analysis, generalisability can not be claimed. However, these findings are largely aligned with a previous work by Cheung and Huang [21] that seeks to identify the different types of esport audience members, even though the two works encompass two different genres of esports (MOBA and RTS). This can indicate a level of generalisability to esports audiences more broadly.

Another factor that has been identified in the literature as an important motivator for esport audiences is the commentator who enhances the audience experience for the viewers [21, 195, 10, 26]. For this reason, Li et al. [91] has investigated the needs of considerations of esport commentators specifically. Through in-depth interviews that lasted between 45 to 90 minutes, 19 professional esports broadcasters were asked a range of questions, from informal ice-breaker questions, followed by information questions about the broadcasters and ending on semi-structure interview questions following self-presentation theory [4] that aimed at providing information about the broadcaster's practices, needs and strategies. Commentators interviewed had experiences with a wide range of esports genres, including FPS, fighting games, RTS and MOBAs, with participants experience in their respective domain ranging from have only just started commentating professionally to 10 years of experience.

The work outlines the high demand on broadcasters to keep up with the game environment - the meta - the general player base and how professional players are performing, as well as the specific events that happen across each match they work on, in order to produce detailed and engaging play-by-play descriptions. This can be a challenging process, yet their contribution is vital for the esport broadcasting experience, with some viewers choosing not to consume esport content unless their favoured broadcaster is narrating the match [73, 135].

Li et al. [91] outlines that commentators must study the game environment carefully prior to broadcasting to ensure they can accurately understand and describe the game. They also must carefully follow the match closely, while attempting to convey everything in an emotionally engaging way to ensure the audience is informed and invested in the game. Yet commentators must strive to achieve a balance between describing the intense moments of gameplay and producing engaging content during the quieter moments, allowing viewers to "reflect on what they just saw and heard". This challenging task requires broadcasters to choose between information selection and presentation. Broadcasters describe how they often rely on multiple sources of knowledge, such as collating community based strategies, to aid in their general descriptions. However, in a live broadcast, commentators must rely on their own intuition and previously collected knowledge to interpret the game, which the author emphasise the difficulty of this process. This clearly demonstrates a need to broadcasting tools that can be integrated to commentators existing strategies to understand the environment. A tool designed to seamlessly integrate with broadcasters existing strategies are crucial, as commentators typically already have established practices and approaches to information gathering and storytelling. Therefore, it is important to implement such tools with the broadcaster's specific use-case in mind, to allow the tool to aid the broadcaster, rather than introducing further complexity of them having to, not only interpret and understand the gameplay, but also interpret the complicated results of a tool that has not been designed for their ease of use.

Similarly, Kempe-Cook et al. [73] investigated the practice of producing commentary in live games. In this work, the authors conducted semi-structured interviews with 20 amateur broadcasters. Additionally, the study also collected extensive diary data (participant self reflection on their performance in broadcasting events) as well as logs containing chronological data of activities for participant observations. Interview and diary data were analysed following a grounded theory approach [20], which allowed the authors to qualitatively analyse the data collected in-depth. The authors identified the importance of play-by-play description, which provides more evidence for the need to cater to this difficult task. Furthermore, the authors described another part of broadcast that is present in commentary, referring to it as "color commentary" [73].

When producing play-by-play commentary, the broadcaster is focused on describing exactly what is happening in that moment. By contrast, colour commentary provides analysis of what has been happening in the match so far. This is typically done in moments of low excitement and it serves as an opportunity to describe the cadence of the match and the decision making employed by players. According to Kempe-Cook et al. [73], commentators often use this analysis to build engaging narratives, reflect on the pacing of the game as well as to stipulate possible future game events and the impact that could have in the match.

In both types of commentary (i.e. play-by-play and colour), broadcasters are faced with difficult tasks, as the game environment can often be difficult to understand or predict, even for experts [82, 22]. Furthermore, the fast and unpredictable nature of esports may force broadcasters to have to abruptly switch from colour commentary to play-by-play, if a moment of importance suddenly takes place [146, 168, 149, 70]. To address this, Kempe-Cook et al. [73] stipulate that broadcasting tools can be used to assist commentators with this difficult task, such as the use of data-driven interfaces that can provide broadcasters with insights in and outside the current game.

This was later explored further by Margetis et al. [101], who designed two live dashboard experiences: a directed dashboard (aimed at esports directors and other members of the production team) and a colour commentary dashboard (aimed at the commentator). This study focused primarily in the role of data-driven storytelling to enhance esport broadcast narratives. Crucially, the authors described the importance of understanding the needs and requirement of the broadcasters in designing such solutions, to ensure the dashboard can be integrated into commentary and esport production more broadly. To achieve this, the work describes a collaborative iterative design process which included 14 broadcasters into the design and development of the two dashboards following the Human-Centred Design (HCD) framework [100]. Both dashboards were qualitatively evaluated through the relevant users, receiving significant positive responses. In particular, the colour commentary dashboard, which was specifically aimed at the commentator (similar to what is described by Block et al. [10]) received interest from the broadcaster user-base who outlined the need for such tools. The authors describe how they received usability related feedback for future improvements to enable faster information retrieval by broadcasters in a live setting. Notably, however, the authors outlined the positive reception by commentators who emphasised the importance of integrating data-driven storytelling into their colour commentary. This was similarly also observed in the other examples of commentator-focused dashboard solutions in the literature [10].

2.5 Understanding ML work

As discussed in Section 2.4, video highlights offer many storytelling applications to enhance the audience experience within esports [144]. The use of ML to automatically generate highlight videos that summarize a match have been proposed in many different ways across the literature. One example comes from outside the esport domain, proposed by Park et al. [120] for generating match highlights for baseball games. This was done using deviations in win probability models to automatically identify moments of key importance, reflecting when the balance of the game shifted, be it either from one team being favoured to win to another, or a team consolidating their advantages to secure their victory.

In terms of methodology, the model proposed by Park et al. [120] divides a match into temporal segments. Segments of time where the win probability shifts are identified throughout the match. They are then ordered with highest shifts first and selected until there are enough segments to fill a predetermined minimum and maximum desired length of the highlight. Lastly, the selected clips are then reordered back into chronological order to create the automatic highlight that displays all of the crucial - win probability shift - moments in the game. In addition to the neutral highlight generation, the authors also investigated the effect of team-biased highlights, giving moments which more heavily favoured one team over the other a higher weight in the segment selection phase to create a fan catered highlight that shows the viewers preferred team perform better. These highlights, neutral and biased, were then tested with baseball spectator fans and qualitatively evaluated showing positive reception from audiences, who have demonstrated both interest as well as higher recall. This work demonstrates how ML can be used to power solutions that cater to sport audiences, and provides good insights that can be readily applicable to esport contexts, particularly as esports focused attempts can also be identified in the literature [194, 95, 40, 140]. In these esport focused examples, however, focus was primarily put into effective ways of generating highlight videos, in terms of computational needs, attempting to recreate human created highlights from a training dataset. While this can be a powerful methodology for creating high performance and high impact solutions, and the advances in computational capabilities are undoubtedly valuable, it is still important to consider the audiences reception to such solutions. Similarly, post-game analysis narrative can also benefit from a diverse way of generating highlights, yet automatic systems proposed in the literature that have been identified with the search strategy described in Section 2.2 offer no consideration or reflections of how these systems can be integrated with or utilised for broadcasting narratives, showing a vast contrast over the way in which ML for sports broadcast are studied in comparison to esports. This is particular salient as viewers have been shown to seek highlight videos for various needs [144], from spectating crucial moments of gameplay that includes high level of player skill to identifying moments in which the less favourable team gain advantage of the match, both motivators that benefit from the solution proposed by Park et al. [120].

Marshall et al. [102] compares a Convolutional Neural Network (CNN) [185], a Recurrent Neural Network (RNN) [49] and a Long Short Term Memory (LSTM) network [57] for performing *microprediction* (deaths) for CS:GO. The author found that LSTM produced the most accurate results when accounting for the player's average in-game death count, number of enemies in close proximity, character health, and equipment value. Using these features, the proposed LSTM could predict the character death with a F_1 score of 0.38, when compared to 0.37 and 0.36 for the CNN and RNN respectively.

Furthermore, the authors utilised the LIME explainer [138] to identify the features needed to allow non-experts to gain insight into the causes of the model's predictions. As the features identified correlated with their intuition of CS:GO gameplay, the author posed that the model would be interpretable (as defined by the LIME explainer [138]) and thus would enhance storytelling for esport broadcast. This raises the importance of explainability for ML models, as well as meaningful feature engineering in a way that broadcasters can understand and build narratives with.

Another form of death prediction in Dota 2 was proposed by Katona et al. [70]. This study utilised a predictive deep neural network to predict the chances of a character being killed within five seconds. Later this work was extended by Ringer et al. [141]. In both these examples [70, 141] the authors took advantage of a shared weights technique for features related to individual players and characters. This methodology allows for greater generalisation as it more directly allows for order independent features to be processed and trained, as the same weights of a network could be trained multiple times across all players for a given match. This creates a condensed architecture where the same repeated features across multiple individual can hold the same intrinsic meaning regardless of each player it represent, or the arbitrary order in which they are stored in the dataset, for example: "player1, player2" should have the same intrinsic meaning as "player2, player1". By using shared weights for the layers which handle player input, this methodology can ensure the same underlying meaning and patterns are learned. This simplifies the environment improving the accuracy of the results. The authors describe how the use of a death prediction model can allow broadcasters to firstly not miss important moments in the match, and secondly build engaging narrative on top of the predictions.

In the three highlighted examples of death predictions [102, 70, 141] the authors described powerful methodologies for training predictive ML models. However, it is important to note that these approaches are subject to the impact of changes to game design [174, 149, 86]. Thus it is uncertain if such solutions could be reused across future iterations of their respective games. Furthermore, while the authors offered some reflection of how this system could have an impact into the storytelling narrative, the implementation and integration of such systems into the live coverage of broadcast was not explored. This can bring additional challenges as the data needed to effectively run these models would need to be retrieved during the live game, then processed in real-time as well as being incorporated into the broadcast as either a commentator tool or an audience facing visualisation. The design considerations and effective strategies for implementing such solutions are yet to be explored.

Another example of ML being utilised to aid with the complexity of esport titles can be found in the work by Demediuk et al. [29]. On this example, the author utilises player performances in *Dota 2* to classify players into roles. As explained in Section 1.2.1, *Dota 2* players typically fulfil strategic roles within a team composition, ranging from positions 1 through 5, which refers to the order of priority of resources. These roles are not defined in the game rules, but are playing conventions that are widely adopted by the community. Thus, while roles are key in the ways characters are played, there is no trivial way to automatically classify players into roles, particularly as characters can typically be played in multiple roles if varied play-styles are adopted. To achieve this classification, the authors utilised a range of unsupervised learning techniques to identify differences in performances of key performance indicators (KPI). By identifying peaks and troughs in different KPIs that are commonly observed through different players, the authors were able to investigate a range of clustering techniques [145] within the data, which lead to an automatic classification of player performances. As professional players in high level esport events are presumed to perform at the highest level of proficiency, due to the self-curating nature of the *Dota 2* esport scene, the performances of players are assumed to be representative to the way in which the different roles are expected to perform. Therefore, by manually labelling the different clusters based on expert domain knowledge, the authors were able to automatically classify players performance, and assign them to the different roles.

While the impact of automatic role detection [29] on esport broadcast narratives was not investigated, the detected roles were further utilised within additional esport research by the authors. Demediuk et al. [30] expanded the work of automatic role detection to evaluate the way in which players are performing, given their roles. This was similarly done through unsupervised learning, where common KPI performances for the different roles are highlighted given the final outcome of a match to detect the ways in which the different roles can have a greater impact into the final outcome of the game. In other words, role 'X' characters that perform well in '*these*' three KPIs *tend* to win games. Thus, by identifying relevant KPIs and their expected values, a metric can be composed which not only classifies, but also quantifies the performance of a player. To achieve this, the authors utilised archetype analysis [27], which provides a set relevant features given each role as well as the optimum target values. This can then be averaged together to formulate a performance index classification metric. This performance index then represents a numerical value, which is a quantifiable measure of the performance of players within their teams for a single match.

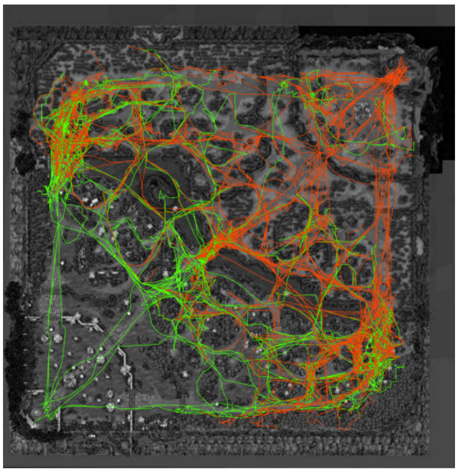
In both examples discussed [29, 30], a clear progression of data can be established, through the use of ML. By first reducing the problem to a single objective (the automatic classification of player roles), the authors were able to then complete a more complex task. This outlines how addressing different areas of the game can lead to improving the performance of different models by providing a richer dataset. However, both examples of the work [29, 30] are highly subjected to the changes in game.

Firstly, the role classification uses the characters as a starting point and compares within character performance. This means that character 1 will always be compared to other character 1 performances within the population. Thus, if this character is fundamentally changed in design, the way in which players play it would be impacted, which could significantly change the distribution of the performance data, potentially leading to new clustering models to be developed. However, as the model relies on telemetry data, significant data collection would need to take place to allow for a new model to be trained. This can be a slow process in terms of real-world time needed for matches to happen in a new patch, as well as in relation to computationally processing the new patch data for every patch. Secondly, the development of a new clustering model would also require labelling, which is done manually with consultation of game experts. Similarly, if a new clustering model is needed to classify player roles, the player index model would also need to be reconstructed, as it relies on the identified roles as an input.

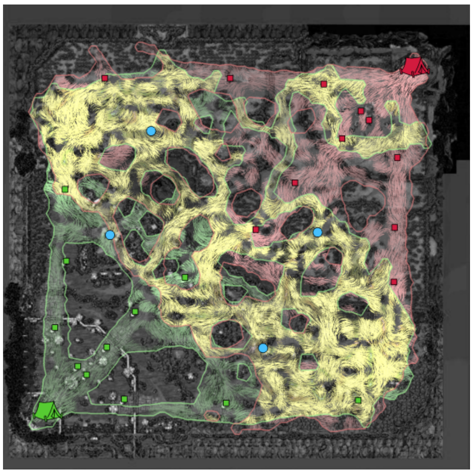
Furthermore, neither the role identification work [29], nor the player performance index [30] were evaluated in terms of the storytelling capability. The authors offered some reflection in how these models can simplify the environment for audiences, allowing them to better understand the game, however the ways in which this information can integrated into the live broadcast - or visualised and presented to audiences - have not been explored.

In some cases, ML has been used to enable visualisations. Wallner and Drachen [177] investigate player trajectories and how they can be visualised to convey information about the match. By plotting player trajectory for a match, the authors then applied a vectorisation which is used to generate a Line Integral Convolution (LIC) texture [15]. The LIC texture map is then combined with an auxiliary image which provides information about trajectory frequency, as well as an α-shape map that provides information about the geometry - or shape - of the player trajectory. By combining these images, the resulting graph represents an analysis of player trajectory, displaying how players are generally navigating through the map across the whole match. This can be observed in Figure 2.3.

The authors offer considerations towards the application of such tools. In particular the author discusses the computational time needed to generate such maps in real-time. While the authors details that the majority of the process can be compute in linear time, the α-shape computation requires $O(nlogn)$ time, where - in this instance - n is the number of points included



(a) An example depiction of the raw player trajectory. Originally labelled as Figure 4a in the source material. Full authorship of the figure belongs to Wallner and Drachen [177] and is used with permission from the authors.



(b) An example of player trajectory when blended using alpha-blending. Originally labelled as Figure 5 in the source material. Full authorship of the figure belongs to Wallner and Drachen [177] and is used with permission from the authors.

Figure 2.3: A depiction of transforming the source image of raw player trajectories (a) into the blended player trajectory visualisation (b). Both images were extracted from Wallner and Drachen [177] and are used with permission from the authors.

in the map. For this reason, the authors outlined how they utilised a map simplification to reduce the number of points, however this approach can lead to high information density per location as several points get merged into one. However, this simplification can enable this methodology to be run in real-time, which the authors conjecture that this maps can be used to display player movement across different timestamps, or to perform analysis of general player movement across a map, such as what has been outlined by Xenopoulos et al. [186]. This can serve as both a training tool for players to improve on their playing ability as well as having balancing implications for game developers to make informed decisions on future iterations of the game map.

Furthermore, Wallner and Drachen [177] called for continued work to enable this visualisation to be audience facing. This outlines that, despite the clear methodological advances posed by the authors, continued work needs to be done to enable such techniques to be integrated into esport broadcast coverage to enhance the audience experience.

Yang et al. [192] raises an important criticism to the current way in which ML predictors function within the esport domain. The authors outline how outcome prediction models have the potential to provide valuable contributions in many systems, including in the field of game playing agents through the use of AI as well as in an AI-based commentator. However, the authors suggest that current approaches to outcome predictions are limited by both the lack of accessible and large-scale datasets of in-game features as well as the lack of interpretable results that models offer. It is important to note that, while certain esport titles, such as League of Legends, offer limited access to data independently, particularly in relation to live telemetry data at, other titles such as Dota 2 or Counter Strike offer comprehensive APIs with high fidelity and high accuracy datasets of in-game features [70, 141, 10, 59, 18]. In other cases, researchers have compiled comprehensive datasets by relying on various input features including community comments and computer vision techniques [188, 23]. Furthermore, game developers may have access to more detailed datasets which are not available to the general public. Industry collaboration has been known to enable analysis of such datasets that are not typically available to the general public [22]. On the other hand, the authors concern with interpretability raises an important issue that is generally applied to most ML powered in-game prediction solutions, regardless of dataset access or data features used.

Beyond this point of data accessibility, Yang et al. [192] postulates that a black-box model would offer less interpretable insights than a prediction model that offers *why* explanations of the predicted results. In particular, the work investigates if highlighting the features most responsible for producing the output has a positive impact on interpretability. To evaluate this, the authors proposed a 'fidelity' metric, which borrows from Natural Language Processing methodologies for evaluating fidelity in language [90]. To achieve this evaluation, the authors first trained a predictor model using a full set of input features. By itself, this model provides a black-box prediction which are not yet interpretable beyond what values that are being predicted. Then the authors performed a feature exploration, training the same model multiple times with reduced features. By comparing the performance of the full set model to the simplified versions, the features that produced results which most closely matched the full features were considered high fidelity. This was performed with a range of features in order to explore which produced the highest fidelity both in conjunction and individually, including a range of 100, 10, 5 and individual (1) features. This investigation was done under four distinct prediction tasks, including the overall winner (i.e. outcome prediction), next player to kill an enemy, next player to be killed by an enemy as well as the next team to defeat an important non-playing character that provides a high reward to the entire team (Tyrant).

It is important to note that, while the authors successfully evaluated which features are most prominent under the black-box ML model, this study did not explore the impact that could have on users. The methodology itself has been explored under different, more general, contexts [162, 152], however as explored in other discussed works of literature [142, 127] *what* questions may be more suited under live esport broadcast domains. Therefore, while it is clear that the proposed model offers more interpretable results, it is still important to explore the usability of such models and how that may fit within the overall broadcasting application. Should this model be made available to audiences, it is still important to understand how are audiences utilising such tools and if this tool addresses the underlaying audience experience needs. Similarly, if this tool is applied in a narrative context, either in combination to or as a possible replacement for commentator narratives, it is important to understand how such narratives are built and what can be done to enhance them. This interpretable model was architectured to produce the most accurate results, designed firstly based on the available feature-set. While this is indubitably a reliable way to produce accurate predictions, it may not address the narrative needs of audiences of commentators. For example, audiences may disregard the insights of predictive models if they believe that they can reach the same conclusions without the aid of predictive models, as observed in preliminary results with experienced players watching esports [179]. Thus, it may be more beneficial to architecture the model in a way that the output itself, and the insights being generated, are already in-line with the user narrative needs. A less accurate model at predicting game events may possess more storytelling implications if it is fundamentally designed for existing narrative structures, rather than pure accuracy with the usability needs considered post-hoc.

Another notable example of an esport win prediction model was proposed by Hodge et al. [59]. This study describes a Random Forest (RF) [14] model to predicting the winner of a *Dota 2* match. The topic of win prediction in esports has been extensively studied within the academic literature [147, 97, 58, 68, 77, 154, 180, 190, 189, 66, 71], however this is the only example identified in which the model was deployed to a major esport tournament (the ESL One Hamburg 2017[1].

The authors noted that matches with higher skilled players are more challenging to predict the outcome, with professional games being the hardest within the overall match population. This observation was also highlighted in other works in the literature [147, 97, 22]. In this particular example, the authors identified that a mixed-population training set produced better results in professional games than a model that is trained using only professional games. The authors conjecture that this is due to the limited number of professional games that is available in a single patch of the game, which creates insufficient data to train a predictive model.

In this example [59], game state data is used to predict the outcome of the game, which includes features such as total damage by a team, total gold and networth, as well as kills or score for a team. The authors described their model as the first time a win prediction model was integrated alongside an actual professional tournament, which was qualitatively analysed alongside the commentator dashboard in the collaborative work by Block et al. [10]. In this win prediction work Hodge et al. [59] focused on evaluating the model's performance.

Initially, the authors reflected on the work needed to enable the model to

[1] `https://liquipedia.net/dota2/ESL_One/Hamburg/2017` (accessed 13-05-2025)

be integrated into the live coverage of the tournament. In this example, the authors took advantage of the collaboration with the tournament organisers to be included in the local lobby of all games as a non-broadcasting observer. This means that a member of the researching team was invited to the game directly, alongside the broadcasters, this allowed the model to retrieve game state data from Dota 2's Game State Integration (GSI) system, which offers automatic and live updates with the information needed for the model to produce predictions. A custom UI was designed and implemented alongside the broadcasting dashboard described in [10], allowing the model to be injected into the broadcast and the narrative.

The authors then reflected on the performance accuracy and the swings (sharp changes on the model's predicted winning team) in prediction observed throughout the tournament. According to the evaluation, the accuracy of the model ranged from 70% to 90% throughout the series. The authors also identified two archetypes in the prediction swing. (1) where the model did not swing throughout the whole match and (2) where two or more swings were observed in a single match.

It is important to note that, while the model was integrated into all matches in the tournament, this only accounted for 28 games, which does not provide enough samples to comprehensively evaluate the model quantitatively. However, the authors offered some reflections on the overall performance and use case of the model. In particular, they noted a discrepancy between model performance expectations and audience needs. The authors outlines that the audience experience benefits from the suspense of the narrative of the match [195]. Therefore, a model that is too accurate would detract from the audience experience. Conversely, a model that is too inaccurate would lead to mistrust of the results by users, which would limit the impact of such models. Similar conclusions and reflections have also been highlighted throughout the literature [82, 195, 139, 91]. This creates an accuracy dilemma in relation to win prediction models, where the model have to be "just accurate enough" to be useful.

In addition to the reflections offered by the authors, it is also important to critically analyse the work. Firstly, this paper offers a valuable contribution to the field as it provides detailed descriptions of how the model was trained and evaluated, and it describes how it was implemented alongside the live broadcast coverage. However, it is unclear how the model performance is impacted by changes in game design as new patches are released by game developers. This is an under explored area of esports ML work that must be addressed to enable long-lasting contributions in the domain.

Secondly, it is important to understand what the model output represents. As the authors outlined [59], such win prediction models have the potential to describe to the audience which team is currently winning at any given time. This is important as Dota 2, and other esports, are complex environments that can often be difficult to parse, therefore a win prediction can serve as a proxy for which team is winning at the time. Therefore, it is important to note that, accurately predicting the final winner of the game at all times would not, necessarily, provide this benefit. Thus, the ideal use case of a win prediction model - as described by the authors - is not of a forward facing model that predicts the final result of the game, but a backwards facing model that reflects on current and past states of the game to indicate which team is currently in the lead, with no reflection on future states. In this case, measuring the performance of the model based on average accuracy of the final results is suboptimal and generally not representative of how the model is intended to be used. The authors outlined the accuracy of the model in the works as ranging from 70% to 90%. This was calculated by averaging all of the predictions for every timestamp in a match, but multiplying any incorrect predictions (i.e. where the loosing time was predicted to win) by *-1*. One alternate way of interpreting this result is purely a reflection on how Dota 2 games play out, where the team that is winning through 70% to 90% of the match wins overall, and the model is purely accurately reflecting that. This is also evidenced by the archetypes of match swings outlined by the authors, where 11 out of the 28 matches in the tournament had no swings and only 7 had more than 5 swings, indicating a high degree of consistency throughout the match. Another evidence to this interpretations can be found the work by Viggiato and Bezemer [174], where a predictive model was used to study matches rather than offer audience facing insights. In this work, the authors investigated blowout matches (i.e. matches which are particularly one-sided). In this study, the author found that win prediction models were significantly more accurate at predicting the winner in blowout matches. This is particularly relevant as the win prediction model described by Viggiato and Bezemer [174] does not use any in-game features, focusing only on character selection data and other variables which are known before the start of the game. This is a contrast to the methodology described by Hodge et al. [59], which uses only in-game features to produce the prediction. Furthermore, the definition used by Viggiato and Bezemer [174] for blowout matches found an almost even split of the entire population of professional games for the years ranging from *2012* to *2020*, i.e. there were nearly as many blowout games as non-blowout games. Therefore, it is important to critically reflect on the evaluation metrics, in particular when relating to the use case.

Lastly, while the collaborative work by Block et al. [10] qualitatively evaluates the impact of data-driven storytelling in the audience experience, the work focuses primarily on the stats presented by broadcasters - as described in Section 2.4 - offering no direct reflection on the win prediction model described by Hodge et al. [59]. Thus, although successfully integrated within a tournament, the impact of the model into the broadcasting narrative has not been evaluated. This is particularly relevant as the authors outlined the contribution as an aid for audiences to parse the current winner, specially when considered that this was a commentator lead integration [10]. Understand how commentators utilise this tool to form narrative and what the impact it has on the storytelling is, therefore, crucial to evaluate the impact of the contribution. Designing the model with the broadcaster needs can elevate the contribution of the work further, as the commentator is generally trying to summarize and inform audiences on the past, current and future states of the game [91, 73, 135, 21].

2.6 Literature Synthesis

This chapter has outlined the relevant state of the art literature that investigates the use of ML to enhance esport broadcasts. As it can be observed, much of the current literature that investigates user needs, be it audience facing [88, 195, 18] or broadcaster facing [91, 73, 10] has called for continued work to enhance storytelling in esports. In particular, collaborative tools that aid the broadcaster on building and enhancing narratives, which demonstrate a clear demand for continued work in the subject [73, 135].

Despite this, ML literature in esports has focused primarily in methodological improvements to computational needs and model accuracy [70, 141, 147, 68, 77, 154, 180, 190, 71]. With few exceptions [59, 177], considerations into designing tools and solutions that are suited for the live broadcast are limited, or not present in the existing literature despite the continued call for future work to enable the application of such models [29, 30, 149, 134].

Furthermore, it is well understood that esport games undergo frequent and significant changes across time (patches) [81, 166]. Similarly, there is also evidence of this factor limiting ML training data, to ensure consistency of the environment improving model performance [141, 134, 29]. Despite this known limitation, very little work has been proposed to address this issue, or perusing patch agnostic models. This major limitation within the domain prevents models from having a long-lasting impact, even if they are successfully integrated into broadcast systems.

In relation to the methodologies currently adopted, unsupervised systems have demonstrated promising capabilities to label, qualify and quantify data [29, 147, 134, 30, 149], while supervised learning methodologies have displayed powerful predictive capabilities [59, 102, 70]. Similarly, ML has been shown to be used as a beneficial methodology for analysing the game, be it to enable continued research or to allow for visual analysis of complex patterns [185, 29, 177].

Overall, it is clear that ML for esports is currently primarily being studied as a methodological contribution, advancing the performance of models in terms of computational demands and accuracy. In relation to storytelling and broadcasting narrative there are continued calls for the application of ML and ML powered solutions, but very little work can be identified where ML is integrated with the broadcast and fewer still where this process is critically analysed. It is well understood that esport commentators face demanding and difficult requirements, with vast opportunities for ML to be adopted. Despite this, no ML work directed at fulfilling such needs could be identified, demonstrating a clear gap in the current state of the art in relation to broadcast narratives. Thus, this chapter provides an overview of the literature addressing the first research question (RQ1) posed in Chapter 1.

Therefore, following the careful review described in this chapter, this thesis poses that - in order to address the gaps identified - three additional steps must be investigated. Firstly, the means to design patch agnostic models needs to be developed (RQ2). This would allow for any impact into the broadcast narrative to not be limited by a small period of time, ensuring a long-lasting contribution to the domain. Secondly, the ways to design and implement live broadcasting tools must be studied, ensuring ML models can be integrated into the live coverage of esport broadcast (RQ3). This can ensure that models can be incorporated in ecological settings, maximising the impact of future research. Thirdly, the steps and considerations needed to integrate ML systems into broadcasting narratives must be explored (RQ4). A full understanding of how ML models can be built for enhancing esport broadcast narratives can enable storytelling, aiding commentators and ultimately enhancing the esport audience experience.

To this end, the subsequent three chapters in this thesis will investigate these three points (RQ2-4).

3 Beyond the Meta: Leveraging Game Design Parameters for Patch-Agnostic Esport Analytics

3.1 Introduction

In the previous chapter, the current SOTA of ML for esport broadcast narrative was presented. In doing so, the review of the academic literature outlined three clear gaps in the way in which ML is currently researched within academia. This chapter investigates a way to address one such gap, where ML models proposed in the literature are typically bound to a single patch, limiting the potential impact of any contribution, particularly as game developers release balancing patches frequently with the intent to alter the way in which the game is played by players [81, 166].

As outlined in the literature, the full set of features typically used for training ML models is varied, even within the same task [51]. However, the characters present in a match are within the most commonly utilised features [51, 70, 141, 29, 30]. For this reason, this chapter will focus primarily on character selection data, and how this feature can be used within ML in a patch aware format, allowing for patch-agnostic analysis.

3.1.1 Character Selection Data within ML Literature

Typically, characters are represented by their unique Character IDs (which are arbitrarily assigned numerical identifiers), which most commonly undergo a one-hot encoding or a similar variation [161, 97, 70, 174, 30, 141]. In this encoding, the unique identifier of a character is represented as a vector class. This means that if a game only has four characters, with IDs 1 through 4 respectively, character 1 can be presented as the vector *(1,0,0,0)*, while character 2 would be *(0,1,0,0)*, etc... This can lead to an architectural problem where, as the number of Character IDs increases so does the number of dimensions needed to encode them. Using the same example, if a new character is then introduced with Character ID 5, the vector needed to represent all previous characters would need an additional dimension. Thus character 1 would then be represented as the vector *(1,0,0,0,0)*. A machine learning model developed prior to the introduction of the 5th character would be architectured to support a 4-dimensional input vector for each character, and thus not support a 5-dimensional input needed given the new game design parameters. This could lead the model to be unusable due to the technical constraints of how it was trained and applied.

Additionally, characters within esport games usually have unique abilities and traits which allow them to be played in different ways [29, 30]. If a character is re-designed, and several of their original abilities or traits are altered, the way in which they are played could significantly change. Those changes would not be encompassed by the Character IDs. Using the same example as above, if character 1 originally had several supportive traits (such as healing or otherwise improving the in-game performance of allies) but then is re-designed to instead have several offensive abilities, the way in which the character is played could be changed significantly. However, as the character ID would remain the same, a model that uses IDs could have its performance impacted, as it had been trained in a different state that does not account for the new changes to the character. In these hypothetical scenario, while the model would still be able to produce results, it would be unclear if the results are reliable due to the uncertainty of the impact of the change to the game's environment.

Thus, this chapter conjectures that changes to game design parameters may impact pre-trained ML models in three ways:

- Breaking changes - These are fundamental changes to the design of the game which would require a change in model architecture in order to produce any results.
- Impactful changes - These do not incur changes in architecture, however have a substantial impact in model performance.
- Unimpactful changes - These do not alter the state of the game significantly enough to affect the performance of previously trained models.

While breaking changes are usually trivial to identify, due to their severe disruption to the application of models, differentiating between impactful and unimpactful changes may require substantial analysis. Furthermore, if either breaking or impactful changes are identified, a previously trained model may need to undergo the training process again to account for the new parameters. This can be a cumbersome process, especially as esport titles typically change rapidly and abruptly [161, 81].

3.1.2 Chapter Contribution

In order to address this issue of changes through patches, this chapter proposes a novel technique for representing character selection data. This methodology provides valuable insights into addressing RQ2, investigating how ML models can sustain performance and functionality beyond initial game patches.

This form of representing characters utilises patch specific game design data, which is then clustered to allow for a fixed and reusable notation that can be readily applied to future models. The Clustered Character Representation (CCR) method holds meaningful information about the character's capabilities and it is sensitive to changes introduced in patches. Within the Dota 2 case-study, each playable character has a range of unique abilities - which are active skills that can be used by players during play with in-game effects, such as causing damage or healing allies - and stats, including "*Intelligence*", "*Agility*", "*Strength*", etc... The patch specific values for each ability and character stats are leveraged to produce this noval notation.

CCR is then tested and evaluated in a case-study, which simulates a hypothetical future work within the esport literature. Three versions of a Neural Network (NN) are trained using professional games of Dota 2 from patches 7.27 to 7.33. These models attempt to predict the number of kills (also referred to as the score) for each team at the end of the match. It is important to note that these NNs bear no direct contribution within the chapter other than as a evaluation metric, however kill prediction was selected as a case study as it is a promising, under-explored area of the literature [146, 70, 141]. Thus, this serve as an illustration of a possible use case within future research, and outline the capabilities of this methodology. (NN1) was used as a simple base-line, where only the match duration was used as an input. No additional features were included in this baseline, therefore this network would have no way of determining or modelling the characters present in the game. (NN2) was utilised as an additional control, where Character IDs (one-hot encoded) were used to represent the characters present in each team, as well as the duration of the match. This network controls for the standard encoding typically used in the literature, which holds information about the characters that have been selected in the match. Lastly, (NN3) was trained using CCR to represent character selection data, as well as the match duration.

Towards addressing the impact of game changes in machine learning models and other forms of data analytics within esports, which poses a problem of short life-spam, this study provides 3 major contributions:

- A feature set of character traits is compiled from the data made available by the game's publisher.
- A novel way of representing characters (CCR) is proposed and validated through performance, outlying how it is sensitive to patch specific context as well as reliant to fundamental changes to the core game environment.
- Access to CCR, including standardized format, centroids, clustered abilities and characters are made freely available for use for future research in the field[1].

Therefore, by investigating CCR, this chapter addresses RQ2 and provides valuable and novel contributions to the esports ML domain.

[1] `https://github.com/ChitaAPC/Dota2CCR`

3.2 Methodology

This section describes the steps taken to collect and process the data used. As explained in Chapter 1, the methodology proposed in this thesis - and consequently this chapter - focuses on the Dota 2 title, as a popular esport game with a wealth of academic research, and large player-base/audience.

3.2.1 Data Collection

In order to conduct this study, two distinct datasets were collected. Firstly patch data was collected, which consists of data about game design parameters for characters, including abilities and attributes from patches 7.27 to 7.33. Secondly, a match history dataset was compiled which consists of the character selection data of all competitive matches for these patches, as well as the duration of the match, the number of kills on each side and the result of the match.

Both datasets were collected through the OpenDota API[2] and its associated public table of constants[3]. OpenDota is a free platform that offers in-depth statistics and a break down of *Dota 2* public matches, including those in professional tournaments and events. This platform has been commonly used by other works in the literature for data collection [59, 29, 30].

As OpenDota's table of constants is stored in a Git repository, it is possible to access older versions of the tables, referring to previous game patches, through the history of individual files or folders. The first dataset was compiled by parsing through three JSON files retrieved from this repository

[2] `https://docs.opendota.com/` (accessed 13-05-2025)

[3] `https://github.com/odota/dotaconstants/tree/master/build` (accessed 13-05-2025)

("hero_abilities.json", "abilities.json" and "heroes.json").

The second dataset was compiled through OpenDota's SQL query feature available on their platform. The data collected included:

- MatchID - a unique identifier for the match
- Patch - the patch in which the match was played in (such as '7.27')
- Duration - The duration of the match in seconds
- KillsR - The number of kills obtained by the Radiant team
- KillsD - The number of kills obtained by the Dire team
- HeroX - The hero played by player X, where X is a number from 0 to 9. Players 0-4 correspond to the Radiant team, and 5-9 correspond to the Dire Team
- RadiantWin - A binary variable containing 1 if Radiant won the game and 0 if Radiant lost (note it is not possible for a match to end in a draw in *Dota 2*)

Only professional and premium matches were collected from patches *7.27* to *7.33*, which included data from *Jun 2020* to *May 2023*. Games which did not conclude in a natural state (for example if a player has abandoned, or if there had been a server error) were not collected. This lead to a total of *61,254* matches from the 7 different patches. Note that, at the time of conducting the study, the latest patch (7.33) was still active. Therefore fewer matches were available, as demonstrated in Table 3.1.

Table 3.1: Matches distribution per patch

Patch	*7.27*	*7.28*	*7.29*	*7.30*
Matches	9,822	4,666	5,492	9,807
Patch	*7.31*	*7.32*	*7.33*	Total
Matches	13,476	16,915	2,235	61,254

3.2.2 Data Processing

In order to perform clustering on the game design data, the JSON files retrieved from the table of constants were processed. Firstly "hero_abilities.json" contains the name of each individual character and their associated abilities. This file was parsed to compile a list of the relevant character abilities, which allowed abilities from non-playable entities, such as neutral characters, to be discarded. Using the name of each individual ability as a key, "abilities.json" was then parsed to generate a CSV file containing the properties of every character ability for each of the relevant patches. This was achieved by compiling a script that normalise and standardise the way data is represented.

Several inconsistencies were detected in the way data was stored by the game between patches and abilities. For example, an ability with the property of "movement_speed_slow" of *30* and another ability of "movement_speed_bonus" of *-30* both have the same effect of reduce the movement speed of the target by *30* units. Through manually comparing the properties names and values and cross referencing to the game's wiki, a script that standardised the properties was compiled. While this approach is subject to errors, gathering data is limited by the way in which data is made available by the game's publishers. However, extracting data using this methodology allows for more in-depth analysis than previous techniques which generally ignore what abilities do and their associated properties [29, 70]. Once all of the properties for every ability was compiled, info on the characters attributes - as described in the literature [174] - was extracted from the “heroes.json” file and amended to the relevant abilities for the heroes. Thus, each entry in the newly compiled CSV file contained data for each ability for each character as well as data about the character itself, which contextualises its use in the game. This process has been illustrated in Figure 3.1 and the CSV files have been made available in the resources listed in the Introduction section for future use.

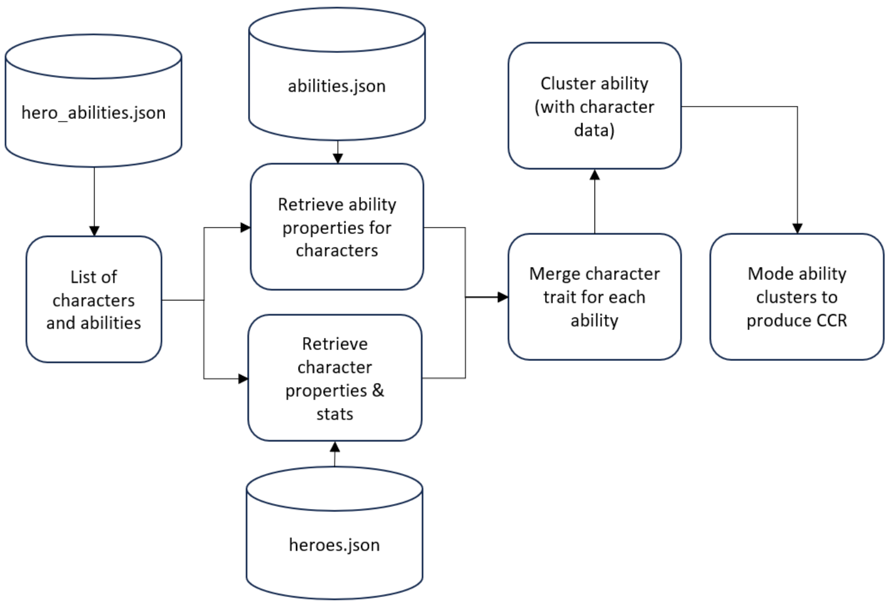


Figure 3.1: A depiction of how data was processed to produce CCR

3.2.3 Cluster Analysis

Once all of the character abilities were processed and the CSV files were compiled it was possible to utilise clustering algorithms on the data to produce CCR. As this study aims to provide longevity to machine learning models in esports analytics and support the addition of new unseen data, K-Means [62] was utilised as it can be readily used without altering the labelling or number of clusters once the centroids are found. This means that once the centroids are found, the model would not need to be retrained with unseen patches. Instead, the data from new patches could simply be predicted with the existing centroids, therefore clusters do not change when adding new data. This approach would then contribute to reducing the impact of breaking changes, as the value of K would not need to change unless the new data leads to a new concentration of clusters (i.e. an impactful change). In this case analysis of the cluster distribution - such as that described in this section - would reveal the need for new clustered. However, data from several patches (7.27 to 7.31) are used to generate the clusters to reduce the risk of impactful changes being present. Note that data from patches 7.32 and 7.33 were excluded from cluster process completely, as those are used as test datasets in the later stages.

In this study, K-Means was used and the "Sum of the Squared Error" (SSE) was calculated to evaluate the split of the data and identify an optimum value for K [62]. Figure 3.2 outlines the elbow plot generated with different values for K. As the figure outlines, the elbow lies roughly between *K=40* and *K=75*. The exact value for K was then selected using the highest "Silhouette Score" [62] within this range, which provided ***K=68***.

Once clustering was performed with the ability data, the labels obtained could then be used to represent characters. As one character contains multiple abilities, the cluster for each ability was encoding into a one-hot-encoded format. This means that any single ability was then represented as a K-

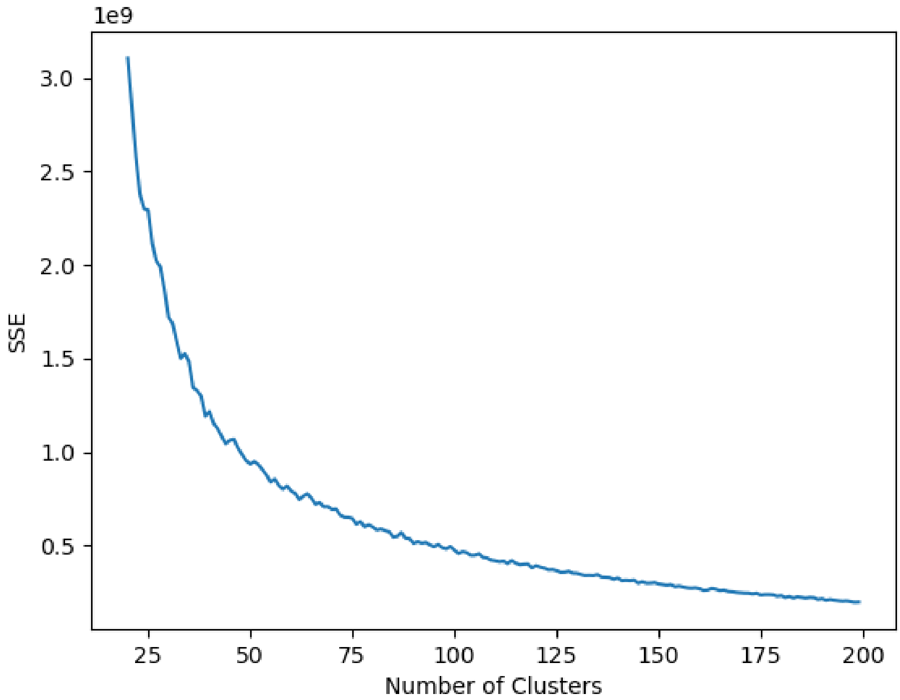


Figure 3.2: The Elbow Plot for the K-Means clustering of the character's abilities and traits together

long vectors of zeroes, except for the i's dimension, where i is the cluster number for that particular ability. Subsequently, one character can then be represented by adding all of the K-long vectors of their abilities. This allows one character to be represented as a K-long vector, containing the modes for each of their ability clusters as depicted in Figure 3.1. Unlike the One-Hot encoding of Character IDs, however, CCR would not be subjected to breaking changes when new characters are introduced, as the value for K would remain the same.

For demonstrations purpose, this approach can be applied to a fictional game, where K=3. In this fictional game "Character X" has three abilities, of clusters 0, 0, and 2 respectively. This means that Character X can be represented as the addition of the vectors (1, 0, 0); (1, 0, 0) and (0, 0, 1). Through this technique, Character X can then be represented as a single K-dimensional vector of (2, 0, 1) - where the mode of *cluster 0* is two, *cluster 1* is zero and *cluster 2* is one. Conversely, a more complex character with 5 abilities of clusters 0, 0, 1, 1 and 2 respectively could be represented as the single vector (2, 2, 1). This simplistic, yet powerful approach allows for complex patterns to emerge from the data while still maintaining the meaning of each cluster as identified by the data. Furthermore, it maintains the size of the vector needed to represent any characters to K, even when new abilities or new characters are introduced by future patches.

3.2.4 Predictive Neural Network

Three NN are used to evaluate CCR. These networks are performing a kills prediction task. This simulates potential use-case within future research, with two baselines. The case-study of predicting the number of kills at the end of a match was selected, as this is a under-explored area in the literature with some of the potential highlighted by existing research as discussed in Chapter 2. This is also a similar use-case as to predicting the winner of a game, which contains a wealth of knowledge [174, 59, 97, 161]. However, as the inputs for the models are highly controlled to enable validation, including the final match duration (which is not known until the game is ended), a comparison of performance between these NN and other predictive models would not be suitable. For this reason, the kills prediction use-case was selected with a range of controls through the three distinct approaches, which allows for a more representative comparison to focus on the evaluation of CCR.

Firstly (NN1) was trained, which used only the match duration and no additional information. This provided a baseline for comparison, as all models include the match duration. This was included as the literature suggests that duration bears a significant impact in the outcome of the game, both in terms of result and in terms of score [174]. Secondly (NN2) was trained which contained both match duration and character selection data. The character selection data was represented by encoding the Character IDs into a one-hot encoding as a standard technique in the literature [147, 141, 22]. Lastly, (NN3) was trained using only the duration and the character selection data as represented by CCR. In order to train this model, whole line-ups - rather than individual characters - were used. This was achieved by replicating the steps for transforming an ability vector into a character vector as depicted in the previous section. The individual character vectors for a team's selection is added together collapsing it into one K-long vector. This means that vectors for abilities, characters and team compositions can all be represented in the same number of dimensions, which expresses the modes for each of the clusters.

All matches for patches *7.32 & 7.33* were excluded from the training set to be used as special test comparisons. Patch *7.32* introduced a brand new character, *Muerta*[4]. Patch *7.33* had several major changes to the game, including the way in which characters primary attributes are handled. In this patch, a new type of primary attribute was created that makes no attribute the primary. This means that - where before, characters were classed as one

[4]`https://www.dota2.com/hero/muerta` (accessed 13-05-2025)

of "*Strength*", "*Agility*" and "*Intelligence*", as of the this patch characters can also be "*Universal*"[5], which means no attribute is their primary attribute. The data collected for the two patches serve as a illustration of breaking and impactful changes. Additionally, the remaining match data (patch *7.27* to *7.31*) were split using a (64/16/20)% to be used as training, test and validation datasets. Table 3.2 outlines the number of matches per each of the different data splits.

Table 3.2: Number of matches per split used for training and evaluating the neural networks from the total *61,254* matches in the dataset

Train	**Validation**	**Test**	**Test** (7.32)	**Test** (7.33)
27,171	*6,793*	*8,491*	*16,564*	*2,235*

As an additional control, all networks were trained using the same architecture with the exception of the input layer, as described above. The networks consist of 6 hidden layers with (*1024, 512, 128, 64, 32, 8*) neurons respectively, with the "Sigmoid" activation function. These models predict the number of kills for each team simultaneously (i.e. two outputs neurons, one for each team). The model was trained for 100 epochs, which was sufficient to outline the trends, detailed in the Sections 3.3 and 3.4. The networks were trained with an *Adam* optimizer and learning rate was set to *1e-4*. No batch, dropout layers or regularizer were used. Figure 3.3 outlines the train and validations loss for each of the three networks per epoch. Further work to this architecture could produce more reliable results, however a comprehensive study on this use-case is beyond the scope of this thesis.

The performance of the predictions were then compared. This includes

[5] `https://dota2.fandom.com/wiki/Universal` (accessed 13-05-2025)

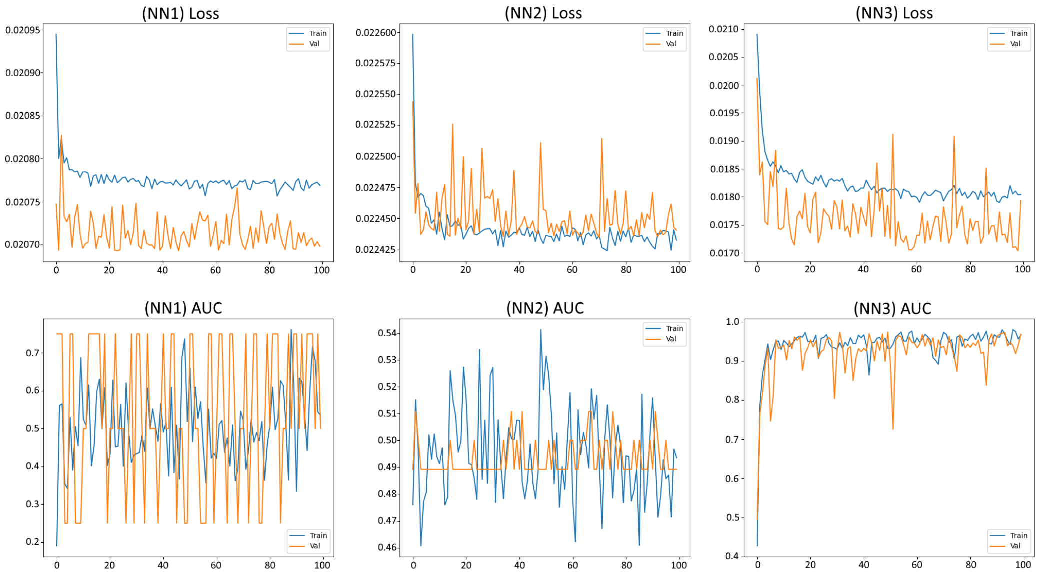


Figure 3.3: Training and validation loss and AUC graph per epoch for each neural network.

the training, validation and test performances for patches from 7.27 to 7.31, which outlines the performance within representative data. Additionally, the performance of predictions for games in patch 7.32 and 7.33, which introduced the new character and new character types were also compared. In order to make those predictions possible for (NN2), the maximum value used for the Character ID encoded included the Character ID for *Muerta*. This was made with the sole purpose of ensuring the architecture supports the additional dimension. It is important to note, however, that this dimension would - necessarily - always have zero in the input vector for all training data, as the character was not present in those patches. Therefore this approach would not be suitable for actually training models with real world applications and it is limited to exactly one additional character (thus, any future characters released would not be supported by the (NN2)'s architecture, incurring breaking changes into the model).

3.3 Results

The results obtained for all networks can be found in Table 3.3. The test dataset contains unseen matches from patches 7.27 to 7.31 - which are the same patches used in the training dataset. Conversely, all matches from Patch 7.32 and 7.33 have been entirely excluded from the training dataset. Patch 7.32 introduced a new character, while Patch 7.33 (the currently active patch as of time of the study) introduced some significant changes to the game design, such as a new character type, as previously discussed.

Table 3.3: AUC of the kills prediction for each dataset

NN	**Test**	**Test** (7.32)	**Test** (7.33)
(NN1)	0.50	0.50	0.50
(NN2)	0.50	0.49	0.49
(NN3)	0.85	0.85	0.86

Furthermore, both (NN1) and (NN2) typically produced the same prediction value for both outputs. This means that for most matches, the prediction for the number of Radiant kills was exactly equal to the prediction for the number of Dire kills. Between the approximately 27 thousand matches in all test datasets (Test, Patch 7.32 & 7.33), (NN1) only produced different predictions for 351 matches while (NN2) produced different prediction 403 of the matches. On those cases (for both networks), the difference in kills only varied by one between teams. This was not the case for (NN3), which only predicted the same number of kills in approximately 3 thousand matches (11.1% of the dataset).

3.4 Discussion

As outlined by Figure 3.3, all three models had a sharp decrease on the loss followed by a plateau within the first 25 epochs. This suggests that all three models reached their convergent point within a few epochs. It is also noteworthy that the train and validation losses did not deviate significantly at any stage for any of the three neural networks. This suggests that none of the models overfit to the training data. Table 3.3 also suggests that there was no overfitting in any of the models, as the overall performance of each network was consistent from that observed with its respective training/validation accuracy at the 100th epoch.

Furthermore, both (NN1) and (NN2) produced very similar results throughout all test datasets. Thus, it is conjectured that (NN2) was trained in a way to put very little significance for the line-up vectors, instead having more significant weights for the duration feature. This brings to question the use of line-up vectors as input features, particularly as it has been shown to be a consistently utilised input feature for esport ML models in the MOBA literature [51]. This is also evidenced when comparing similar performance in the literature. For example, the model proposed by Hodge et al. [59] does not include any character selection data and is performs similarly to the model proposed by Summerville et al. [161] - which does include character selection data. Another example can be observed when comparing the comparatively lower accuracy obtained by Semenov et al. [147] which only uses character IDs, to the work proposed by Viggiato and Bezemer [174] which includes additional contextual information.

Moreover, both (NN1) and (NN2) produced a relatively flat training AUC graph, despite the drop in loss at the early epochs. This may indicate that the models trained to produce a consistent prediction that minimizes the loss without detecting any meaningful patterns. On the other hand, (NN3)'s AUC graph produces a sharp rise in AUC, which is consistent with the drop in the loss. In addition to this, it significantly outperformed both baseline models (NN1 & NN2), pointing to the network detecting meaningful patterns in input data.

When comparing the performance of the three networks across the Test, Patches 7.32 and 7.33 datasets, it is clear that (NN3) produced consistent results despite the introduction of the new character and the change in character primary attributes observed in the two new patches. No observable difference could be detected in (NN1), which is expected, as it was not trained with any character information. Although there was a small drop in performance for (NN2), this could fall within the margin of error, particularly as the predicted values for this model resembles that of (NN1). Similarly, while there was a small increase in accuracy for (NN3) in Patch 7.33, that could also fall within the margin of error, particularly because there are fewer matches in that patch as the current live patch at the time of the study. However, as (NN3) maintained a high degree of accuracy through all patches, that is indicative that the CCR may hold significant underlying principles of the game that is able to contribute to the prediction.

It is important to note, however, that significant changes to the core game design - beyond that of the patches present in this study - could impact the cluster themselves and change the density distribution of the clusters. In this case new clustering may be required, which would similarly cause breaking changes to (NN3) - as the number of clusters or the meaning of a individual cluster may change. Thus, while this approach is a step towards more robust machine learning models, it is not completely devoid of risks of future impacts. Nevertheless, it reduces the risk of breaking changes, which provide improvements over the existing methodology (One-Hot encoding of Character IDs). Additionally, it would be possible to identify the need for new clusters as soon as new patches are released. This can be done by comparing the data distribution of CCR of a new patch to previous patches, which can be achieved using the same metrics described in Section 3.2.3. Therefore, by investigating the SSE for the clustered data of a new patch, and comparing it to previous patches, it would be possible to identify impactful changes from day 1, without the need to way for telemetry data to be generated as players interact with the new patch.

Lastly, the advantages observed in this study (maintaining high performance, supporting the addition of any number of characters) suggest that the CCR can increase the longevity of models developed in future research. Additionally, the results presented here suggests that this approach can more easily represent underlying patterns in the data, as observed by the differences in results between (NN2) and (NN3).

Thus the methodology discussed in this chapter serves as a way for ML models to sustain performance and functionality beyond initial game patches. CCR itself is an example of how ML can be used to enrich datasets, enabling sustained performances of future models. Furthermore, other features beyond character selections can also be incorporated with a similar methodology described in this chapter. For example, in-game items are another consistently utilised feature across the ML esport literature [51]. The same methodology proposed for character selection can be readily applied to items to create a version of Clustered Item Representation (CIR). Thus, this chapter investigated and addressed RQ2, as it provided a novel methodology for representing input features used in ML models that can contribute with performance beyond that of the original data used for training even across multiple patches.

4 Applying and Visualising Complex Models in Esport Broadcast Coverage

4.1 Introduction

The last chapter proposed a methodology for patch-aware data representation which allows ML models to perform patch-agnostic analysis, which explored RQ2. This chapter investigates how ML models can be used alongside the live broadcast of esports in order to address RQ3.

This chapter aims to extends previous work in the literature [22] in modelling the "ward" mechanic in Dota 2 and investigates how to produce and deploy a visualisation tool for this metric. This tool was later utilised in the 2021 *Promod Esports* coverage of the "Roshan Rumble" grand finals (25th July 2021), a UK based Dota 2 tournament[1] that took place with a in-person audience and streamed live in the Twitch platform.

4.1.1 Complex Models in Esport Broadcast Coverage

As discussed in Chapters 1 & 2, esports as a form of digital media is one of the larger markets for digital entertainment [113]. As a result, esport has seen a wealth of investments, including an abundance of academic interest [128,

[1] https://liquipedia.net/dota2/Roshan_Rumble (accessed 13-05-2025)

106, 104]. One example is in the field of understanding the game environment through modelling [29, 146, 30]. Another area which has been investigated in the existing literature is audience focused visualisation [82, 10, 18].

Both areas provide indisputable contributions and insights, and generally relate and rely on each other [82, 168]. However, a detailed exploration on the steps necessary to bridge the gap between complex theoretical models to visualisation applications has not been fully investigated.

4.1.2 The WARDS Metric

As described in the Chapter 1, Dota 2 is an imperfect information game, i.e. players do not have access to all game state data. Due to the Fog of War (FoW) mechanic, teams can only gain information about a region of the playing field if a friendly entity is present in the area. Wards are in-game items that can be placed by players on the map. Once placed, a ward becomes a friendly entity, providing "vision" in the area, enabling information gathering as demonstrated in Figure 1.3.

This chapter first discusses the replication of the Wards Aggregate Record Derived Score (WARDS), first introduced in the academic literature [22], in which the authors have created a metric for quantifying the impact of a ward in a *Dota 2* match. The authors have provided a general formula for calculating WARDS, which relays on several in-game metrics and events. This allows for more in-depth analysis of the mechanic, which is generally considered an important aspect of the game that directly impacts the outcome [22, 168].

It is important to note that this chapter builds on previously published work, to which the author of this thesis is also the author of the replicated work. However, the previously published work [22] was carried out entirely outside of the commencement of this PhD thesis. Therefore, while the replication of this work provides a foundation for establishing the research domain and generating additional contributions, the original WARDS metric itself is not considered part of the contributions of this thesis. Refer to the Declaration Chapter for the full list of declared work which is considered part of the contribution of this thesis.

4.1.3 Chapter Contribution

The contribution of this chapter is twofold. Firstly, a case study on visualising and deploying the WARDS [22] metric is presented. This case study was deployed into a real-world tournament, enabling the collection of ecological data, as well as highlighting practical considerations when dealing with the esport broadcast field. Secondly, this chapter compiles the key findings, reflecting on the data collected in the case-study, to formulate a set of design considerations, aimed at enabling the application of complex models into esport broadcast contexts for audience experience.

4.2 The WARDS Visualisation Case-Study

4.2.1 Identifying Stakeholder Needs

In order to design this visualisation, it was important to understand the demands and requirements for such a tool. This includes the steps needed for it to be integrated into the funding strategy of the tournaments, the steps required to be incorporated into the broadcasting infrastructure as well as understanding what was needed to ensure the tool was usable and useful. To ensure all of these criteria were successfully identified and ultimately met, three interviews were carried out. For brevity and ease of parsing,the three interviews will be referred to as the Stakeholder Interview, Commentator Interview and Post-Prototype Interview respectively.

The first interview was conducted with senior stakeholders at *Promod Esports*, including members of the management team and production team. This was an unstructured interview to initially outline the possible collaborations between the research team and the tournament coverage. The second interview conducted included the two commentators who would be working in the delivery of the tournament. This interview consisted of a series of structured questions aimed at identifying usability requirements. Particular focus on how the collaboration could benefit the commentators existing strategies during coverage. The interview also concluded with an opened discussion section to give commentators an opportunity to discuss any additional thoughts, comments or requirements that might have been missed in the structured part of the interview. Lastly, another interview was conducted with senior stakeholder post prototype development. This was the last opportunity to make changes and ensure the visualisation system could be integrated into the broadcast.

Stakeholder Interview

This interview was conducted in a video call, which served as the starting meeting between the research team and the tournament organisers. The primary aim of this initial meeting is to identify the relevant criteria by the senior stakeholders within the organisation. This was an important step, as any tools would only be integrated into the broadcasting system if it aligned with the needs of the tournament.

Three main points were identified within this explorative meeting. The organisation was seeking a novel feature to include in the post-game analysis panel. Their intent was to include unique commentary as an opportunity to gain and retain viewers, particularly during the post-game analysis panel, which typically proceeds the advertisements that are run in-between games. Furthermore, the use of such tool could serve as an opportunity for future sponsorships, where brands could financially support the tournament, and in return have their branding integrated into the tool. Lastly, a clear production requirement was identified. The tool needed to be secure and complete, which means that integrating it into the streaming pipeline had to be accomplished with minimum changes to the existing infrastructure.

Commentator Interview

In addition to the first interview conducted with senior stakeholders, it was also important to understand the needs of the commentators. To this end, a dedicated interview with the two commentators working in the tournament finals were conducted, also through the use of a video call. The commentators were Jared "Nomad" Bajina [2] and Robson "TeaGuvnor" Merritt [3].

This interview focuses on the narrative needs of these broadcasters, particularly focused on the existing WARDS metric, and how it can be integrated within their storytelling narratives for the post-game panel. The commenta-

[2] `https://liquipedia.net/dota2/Nomad` (accessed 13-05-2025)
[3] `https://liquipedia.net/dota2/TeaGuvnor` (accessed 13-05-2025)

tors emphasised the need for the tool to be contextual and constructive. This means that it needs to be clearly and quickly understood, as they have very limited time to construct narratives, with enough details about the metric to allow for narrative hooks to be drawn without the need for post-hoc interpretations. Lastly, it was also clear that the tool needed to be optional to their broadcast. While both broadcasters were keen to explore the warding narrative the tool could provide, they also wanted to ensure that they were not forced to do so, if the match had other significant moments that needed to be discussed instead.

Post-Prototype Interview

This interview was conducted with the same senior stakeholder present in the first interview. Unlike the first interview, however, this was conducted after a working prototype had been developed. This allowed for any additional changes to be made prior to the tournament coverage as well as served as an opportunity for the tool to gain final internal approval by the stakeholders.

Therefore, in this interview, the final workflow was devised and approved ahead of the tournament. The design and development process is discussed in this chapter in Section 4.2.2, however, this section provides a brief overview of the way in which the tool was integrated into the streaming pipeline, as discussed in this interview.

In order to allow for data retrieval, the ML code replicating the WARDS metric would be operated by an external device (managed by the research team). This machine would be running a client of the game (Dota 2) and be invited to the local lobby of the match, alongside both the commentators and the players. By having access to the local lobby, it was then possible to retrieve the data needed in real-time by running a Dota 2 console command ('instant_replay') that allows the client to record a readable replay file in real time. The data extracted from the game is used to run the model, generating the WARDS needed for all wards in the match. The calculated WARDS were then used by a custom built visualisation tool (also running in the research team computer) that generated the WARDS visualisation as an image. This image would then be uploaded by the researcher to an internal file-store server accessible to production team. The streaming software handling the broadcast was set up to automatically mirror the latest image in the file-store into an internal scene. This scene was visible by the commentators, who could then choose to integrate that with the broadcast (which would also make the image visible to the audience).

4.2.2 Development & Deployment

Replicating the WARDS Metric

As the original work had been designed for an older iteration of Dota 2 (i.e. patches), certain features of game design had changed since the publication. Therefore when working to replicate the original work, all constants and in-game values were validated with the most up-to-date versions of the game.

While braking changes (as defined in Chapter 3) were identified for the predictive NN depicted in the original work [22], the WARDS metric itself did not architecturally suffer from any of the patches released since the work had been originally published. Furthermore, as the WARDS metric was only needed at the end of the game, the predictive NN was not needed, thus was not replicated or adapted. This is because, in the original work, the authors described both a quantifiable model that provides a data-driven evaluation of the WARDS as well as a predictive model that would predict the final value of a WARDS as soon as it was first placed into the game. The final score of a WARDS can only be calculated after it expires, which means that in a live (in-game) coverage, a predictive network could provide additional storytelling insights. However, for the post-game panel, all of the wards could be calculated directly, therefore their WARDS value did not need to be predicted in this instance.

By utilising the replay files recorded live, a custom tool that replicates the steps depicted in the original work could be developed. This was done following the instructions depicted in the original work and enabled continued integration into a visualisation tool. The final WARDS function could be calculated using Equation 4.1. In this derived equation, K is the number of consequence kills, T is the total ward duration, $\mathrm{D_c}$ is the cumulative time of enemies detection, $\mathrm{D_i}$ is the number of items detected, $\mathrm{D_l}$ is the number of levels detected, O is the "optimality" score and P is the penalty score. Additional details on how the parameters are calculated can be found in the original WARDS work [22].

$$WARDS = K * (T + D_c + D_i + D_l) * (O - P) \tag{4.1}$$

Visualising the WARDS Metric

Once the WARDS was replicated and could be readily calculated for a given ward, a visualisation tool was then designed. As this was integrated into the

"Roshan Rumble" coverage, it was important that the visualisation followed the same branding patterns as the rest of the production. This would enable - as discussed in the first meeting with senior stakeholders - the tournament to use the tool as a possible sponsorship opportunity. Additionally, this graphic provides insight into Dota 2, which also has its own branding that is recognisable by audiences. Therefore, merging these branding patterns could assist in creating a tool that is useful for audiences and broadcasters alike.

In this case-study, the graphic was going to be utilised to cover the entire screen for a short duration during the post-game analysis while broadcasters provide commentary. For this reason, an appropriate game background image was used, which allowed audiences to find familiarity with the graphic [10, 11]. Furthermore, the Dota 2 UI utilise a series of dark greens panels for backgrounds, and the "Roshan Rumble" branding uses stretched and damaged edges for panels and background. Therefore, a main panel containing the information was added which takes advantage of both branding patterns as displayed in Figure 4.1. In addition, "Roshan Rumble" branding utilises orange and beige, which were used to provide accents and details. Lastly, as the WARDS metric is closely linked to a map location, it is therefore important to ensure the spatial information is conveyed [1]. Thus, the game's artistic minimap was used to allow for a location to be highlighted.

As shown in the literature, a small understandable amount of data can provide significant insight and drive narratives [10]. Additionally, as the WARDS function quantifies an aspect of the game into a numeric value, this can be used to effectively convey the information related to the ward. However, the values obtained by the WARDS metric are not easily linked to

Figure 4.1: A demonstration of how branding was used to guide the initial development of the WARDS visualisation tool

any other in-game metric and may not be quickly digested by audiences or broadcasters. This would be in conflict with the commentator needs for the tool to be contextual and constructive.

To address that, two threshold values for max and min WARDS were selected. These were chosen by calculating hypothetical maximum and minimum scores for the WARDS value. However, while hypothetically possible, those values are infeasible in nature. Therefore the thresholds were scaled by 25% - i.e. multiplied by *0.75* (maximum) and *1.25* (minimum). These were then used to calculate a percentage score for the WARDS, allowing any ward to be represented in a intuitive scale from *0* to *100%*. It is important to note, however, that as the threshold values were scaled to arbitrary amounts, it would be possible for a ward to score outside of the thresholds. Therefore additional checks were put in place to ensure WARDS could not go below

0% or above 100%.

In addition to the WARDS value itself, some additional insight into the metric could be beneficial, to aid the commentator to build narrative [82]. For this reason certain aspects of the formula were also included, as well as the player who placed the ward and the time and location in which it was placed. This provides the commentator with spatio-temporal and ownership information to easily identify the ward, as well as the context required to understand the assigned value. This contextual and constructive information was then presented as succinct bullet points, to allow commentators to build their narratives without the need to post-hoc interpretations of the model.

Those were added following the same branding patterns described and are shown in Figure 4.2 As demonstrated by the figure, the location of the ward (spatio information) is depicted in the allocated minimap as a cross. The times of ward placement and destruction (temporal information) are denoted in text underneath the name of the player who has placed them. Lastly, the WARDS value itself is displayed in a highlighted circle, providing insight into the ward performance at-a-glance. Additionally, to further enhance the "at-a-glance" readability of the WARDS metric, the radial panel containing the percentage value is also highlighted by the given WARDS amount. Thus, in this example 95% of the circular panel is outlined in bright orange and 5% in darker orange, matching the WARDS value for the hypothetical ward depicted.

At-a-glance information is proven to be an effective narrative strategy [10], which allows for commentators to quickly parse the main WARDS metric information. The additional context can then assist in storytelling [82], by

Figure 4.2: An example of the main WARDS panel depicting the WARDS, and relevant spatio-temporal information about the ward.

providing the necessary context to understand how the metric was calculated. Separating the two assets between a panel and contextual information given on the side assists the commentator to parse what is needed without cognitively overloading them while they are constructing the narrative, which is a cognitively demanding task [91, 73]. Similarly, the information being presented are primarily answering *what* questions, providing what the score for the ward was, and what features lead to the score. This was also designed to ensure the commentator can build their narrative, be it by addressing *why* this is relevant or building other forms of *what* narrative, within the constraints of the live commentary of the post-game panel of the match [142]. This design of separating the at-a-glance panel and the contextual information, aims to direct the user attentions primarily to the panel with additional information being digested auxiliary. Thus, as depicted in Figure 4.3, the main (at-a-glance) panel is placed on the left and additional information on the right with some padding in-between the two.

Figure 4.3: An example depiction of the full WARDS graphic, displaying the at-a-glance panel and the contextual information

Lastly, in order to deploy the graphic, a tool that parses through a live recorded replay file was developed. When the game reaches the end state, the ward with the highest score is used to generate the graphic depicted as an image in Figure 4.3 and uploaded into the organisation's shared file-store. This image was then visible to the commentator, who may choose to integrate it into their narrative or not. This provides full control over the use to the commentator, allowing them to choose not to include the graphic or only include the at-a-glance portion of the graphic if needed, depending on narrative or production needs.

4.2.3 Utilisation & Evaluation

The WARDS visualisation tool was deployed in the “Roshan Rumble” grand finals, which included a total of 5 games between “The Boys” vs “Hidden Pool Pride”. Although the Twitch channel that held the tournament has been discontinued (therefore subsequent on-demand recording of the live tournament is no longer accessible) observational data of the narrative provided by the broadcasters was collected during the deployment and is used to evaluate this case-study.

Game 1

The WARDS graphic was not utilised in the first game of the series. Instead the commentators opted to focus the post-game analysis on a specific team-fight that has happened towards the end of the game. This was possible because broadcasters had control over the use of the graphic and could choose to conduct the post-game panel at their discretion. The generated graphic for Game 1 is provided for reference in Appendix C.

Game 2

The second game of the grand final series was the first time the WARDS graphic was introduced to the audience. In this 40 minutes game, the team “The Boys” lost to “Hidden Pool Pride”, which was reflected in the post-game analysis commentary. The graphic generated by for this game is available in Figure 4.4.

When introducing the new graphics, the broadcasters highlighted the

Figure 4.4: The generated WARDS graphic for the "Roshan Rumble" Grand Final Game 2

WARDS value of 84% and explained that it was related to the impact of the ward in the game. Following from this brief explanations, the broadcasters then highlighted how despite loosing the game, the ward was placed by "The Boys". The ward itself was then identified by the commentators, providing the audience with some additional context in relation to the game more broadly. For example, the commentators first identified the ward:

> *Broadcaster 1: Oh... this [the ward] is for when they had that good Roshan team fight.*

Which was then followed by the importance of the ward, allowing them to build the subsequent narrative of the post-game panel:

> *Broadcaster 2: Yeah, that was their best chance back in the game, but it did not work out.*

Their post game analysis then continued outlining several events after that ward, which lead to “The Boys” loosing the game.

Game 3

Game 3 of the series was once again won by team “Hidden Pool Pride” in approximately 40 minutes. In the post-game analysis, the commentators focused on how contested the “early game” was, which refers to the large number of encounters that happened within earlier stages of the game. As shown in Figure 4.5 the WARDS graphic supported that narrative, as the temporal information shows that the most impactful ward was in place between approximately 10 and 16 minutes and is located covering a common entrance to one of the lanes. While it is unclear if the broadcasters built their narrative around the graphic, or if the graphic supported their understanding of the game, it is clear that the WARDS tool was integrated within the narrative delivery.

Game 4

The forth match of the series was won by “The Boys”, who won the game in approximately 25 minutes. This is considered to be a fast game. It was also the shortest duration of the five match series. The post game commentary identified how the vision mechanic for this match was highly contested, with most wards being destroyed quickly by the enemy teams. This was also reflected by the lowest of the WARDS seen in the series at 56% as depicted in Figure 4.6.

Broadcaster 1: The vision game was hard, both teams dewarding

Figure 4.5: The generated WARDS graphic for the "Roshan Rumble" Grand Final Game 3

> *[destroying the enemy wards] constantly!*
>
> *Broadcaster 2: But Overlom's ward on the top lane really secured the safe lane for the Dire ["The Boys"]. It [the ward] stopped, I think, 3 rotations [when a player moves from another lane to force an encounter] from the mid lane.*

Additionally, the commentators praised the location of the ward, as it provided strategic defensive insight to protect the players on the top lane, emphasizing the relatively high number of kills to low detection time.

Game 5

The last game of the series was won by "Hidden Pool Pride", securing a tournament victory. This ward scored the highest in the series, with a 99% WARDS value. The match lasted for 32 minutes 30 seconds, although the

Figure 4.6: The generated WARDS graphic for the "Roshan Rumble" Grand Final Game 4

top score used in the graphic was placed at the early stages of the game (between 2 and 8 minutes, approximately), as depicted in Figure 4.7.

When presenting the graphic to the audiences, the broadcaster focused on the additional information denoting four enemy heroes had been killed within 30 seconds of being detected.

> *Broadcaster 1: This just shows how DNC [one of the players of "Hidden Pool Pride"] just spent the whole early game feeding [a term used when you are killed repeatedly, which consequently provides a lot of resources to the enemy]*

The post-game analysis then drew comparisons between the top lane (where the ward was placed) and the other two lanes, and how "The Boys" were more successful on the top lane but "Hidden Pool Pride" were generally more successful in the other two lanes, which lead them to a ultimate victory

Figure 4.7: The generated WARDS graphic for the "Roshan Rumble" Grand Final Game 5

later in the game.

4.3 Discussion

Commentators had complete control over the use of the WARDS graphic. As Game 1 had a recent narrative hook, the post-game panel opted to drive the narrative towards the team-fight rather than vision. This provided the commentators with greater storytelling flexibility. In contrast, the other games (particularly 3 and 4) had the overall narrative centred and complemented by the graphic tool, despite not being externally prompted to utilise the tool. This suggests that commentators choose to utilise the tool directly.

Thus, it is important to reflect on the successful integration of this system. Firstly, the design language employed in the visualisation tool had to match the tournament branding. This was a crucial part to successfully applying the model into the broadcast, as tournament organisers are bound by their funding requirements. Incorporating such needs into the visualisation system can ensure that ML models, particularly if they are audience facing, can be integrated into the main broadcast channels directly, reaching a wider population. Secondly, generating a solution with minimum technical requirements can also aid in the successful integration of such systems. Furthermore, understanding the commentator's narrative needs can also ensure that the tool is useful for building and enhancing narratives. Particularly in relation to the utilisation of such tools, as commentators have many challenges and considerations during storytelling.

To aid in future work in the domain, these reflections have been condensed into a set of design considerations. These have been formulated as questions, allowing future ML visualisation tools to be designed with the findings of this case-study. This can be achieved by systematically addressing each one of the questions listed during the design process of visualisation toos.

- **Branding & recognition**: Are there any branding and design patterns that need to be matched, including title specific and platform branding?

- **Understandability**: Can the model provide understandable insights without the need of prior knowledge of the model?

- **At-a-glance**: What are the key parts of the model that can be understood at-a-glance?

- **Spatio-temporal information**: Are there any spatio-temporal information that needs to be depicted with animations, text or images?

- **Additional context**: Is there any fields or parameters that can provide additional information if displayed beyond at-a-glance content?

- **Controllable and delivery**: Who would have control over display and delivery, and how best to integrate it with the broadcast?

These design considerations focus solely on the design of ML visualisation tools for live esport broadcast and how they can be integrated into the ecosystem. This crucial aspect addresses RQ3, explaining how application systems can be used alongside esports broadcast. The findings outlined in the case-study provides useful insights into the development phase of a pre-existing model. The design considerations suggested in this chapter can then allow for this to be replicated in future developments.

The next chapter then utilises the knowledge gained in this case-study as well as the methodology described in Chapter 3 to design a purpose built ML model with the intent to enhance esport broadcast narratives. This seeks to address RQ4, concluding the investigation of the four sub-RQs. Consequently, this leads to the main RQ of how to apply ML to enhance esport broadcast narratives to be addressed.

5

How Could They Win? An Exploration of Win Condition for Esports Narratives in Dota 2

5.1 Introduction

In the previous chapter, design considerations for visualisation tools were proposed, in relation to integrating and applying complex models in the live coverage of esport broadcast. This chapter focuses on the design of ML models, following the same principles outlined in Chapter 3, ensuring the model can be used across multiple patches, and Chapter 4, ensuring the visualisation and utilisation of the model can be integrated with the live broadcasting. By studying the design and architecture of the ML model and how it impacts the broadcast narrative, this chapter seeks to address RQ4.

Specifically, this chapter proposes a novel variation to the win prediction task. In this approach a novel win condition model is presented, reflecting on the usability and impact on storytelling.

5.1.1 Problems with Win Prediction

As outlined in the literature discussed in Chapter 2, win prediction is a prevalent topic of research across the esport literature [59, 147, 97, 58, 68, 77, 154, 180, 190, 189, 66, 71]. However, despite the wide-spread focus in the topic, the accuracy dilemma outlined in the literature [59], the usability issues of how audiences interact with the models [179] and the disparity between what the model is expected to do and how it is evaluated in regards to accuracy and performance limit the impact such models can have on esport narratives.

The contents of esport broadcast coverage reflects these issues, where commentators generally do not integrate win prediction models in the narratives [59, 10], despite its widespread implementation in games. As an example Figure 5.1 depicts an example of a win prediction graph that is available for spectators of Dota 2.

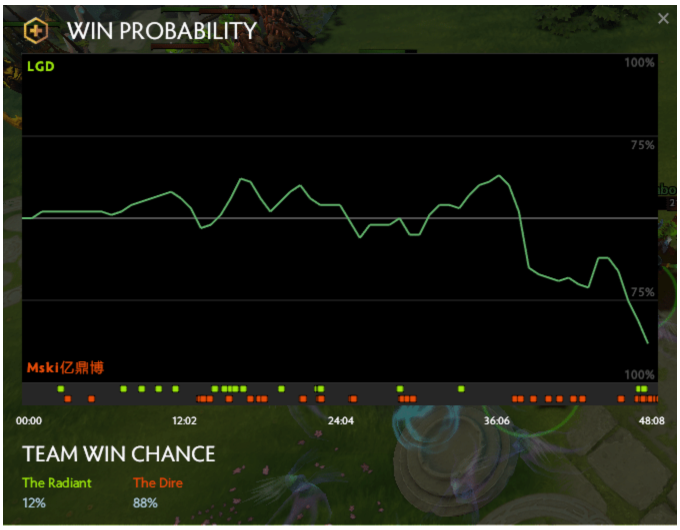


Figure 5.1: An example of the Dota 2's Win Prediction graph as available within the in-game client

As depicted in Figure 5.1 and discussed in Chapter 2, the win prediction available in the game is presented as a continuous prediction that updates over time, reflecting events that happen in a match. Models used for predictions are commonly black-box systems - i.e. it is not usually feasible to ascertain what caused a particular prediction to be generated. As a result, broadcasters and audiences are generally required to use post-hoc interpretations of the results to build narratives. However, predictions may not always match user expectation, which can lead to a jarring process and generally limit the utilisation of such tools in an ecological context [191].

Furthermore, while more interpretable models exist [191], the competitive and complex nature of the titles typically lead to sharp shifts in output predictions as depicted in Figure 5.1, providing fewer insights into potential future states, instead providing a description of the current and past states of the match. This is particularly problematic as the general user expectation of these models - within both academia and audiences [59, 82, 179] - is that the model is predicting the final outcome of the match at all times. However, as discussed in Chapter 2, this may not be a true reflection of the model's output. This can consequently lead to either less trust in the prediction system [82] or, in the case of more knowledgeable viewers, lead to limited insights that were not already known by the users directly [179].

To address this issue, this chapter explores an alternative approach to address the win prediction task, proposing a narrative focused solution that predicts win condition, rather than the outcome.

5.1.2 Defining Win Condition

This chapter investigates the ways in which *Win Condition* can be defined, including the features associated with win condition and their relationships. This section provides a general definition of the term, and how it contrasts to a *win prediction*.

As explored in Chapter 2, win prediction is a well studied subject within esports, with several machine learning models being proposed in the existing literature [59, 174, 191]. Those models output is interpreted as a prediction of who will win a match, given the current state of the game. As explored in this chapter, this has several implications to storytelling, including potential detrimental effects including in audience trust and suspense building [59, 179].

By contrast, this chapter proposes *Win Condition*, which is designed to more seamlessly integrate with existing esport narratives. This is achieved by first reversing the flow of data from typical win prediction models. Rather than predicting who will win the game, win condition produces a prediction of *how* a team may win the game. Therefore, while the output of a typical win prediction system is a percentage chance for a team to win a game, the output of a win condition system is the game state that is estimated to be needed in order for a team to win a game.

Additionally, win prediction models typically utilise a wide range of features to produce predictions (i.e. to represent the game state as an input). These features are selected with the intent to maximise model performance and accuracy, and may not always correlate to user needs (see Chapter 2). By contrast, *Win Condition* is designed for storytelling, with the intent of being readily integrated into existing narratives. For this reason, the features used as output of the win condition system (i.e. to represent the game state needed to win) were selected from existing narrative, ensuring the system matches the broadcaster's user needs (refer to Section 5.3.1).

By using an exploration system, similar to other works proposed in the literature [191], this chapter provides an interpretable output for win condition. Furthermore, ML has been shown to be a reliable way to investigate and analyse different features of esports [174], while data-driven stories have been known to enhance the general audience experience [82, 10]. Additionally, complex causal relationships between variables can be modelled through structured causal models (SCM) [122], which allow for a greater understanding and representation of complex underlying relationships and correlations. For this reason, the *Win Condition* system proposed by this chapter first investigates common narratives (see Section 5.3.1), then models these features in a SCM (see Section 5.3.2) and then evaluates it through the use of machine learning (see Section 5.4).

5.1.3 Chapter Contribution

This chapter investigates a different approach at prediction. By focusing on the narrative and storytelling applications within Dota 2, this chapter proposes a win condition model. This is achieved by first investigating what broadcasters and esport analysts use as narrative points to communicate their expectations of the game (particularly at early stages or pre-game phases). By analysing existing broadcast content, features that are perceived to have predictive significance can be identified. As extracted from existing broadcast content, these features are already commonly used in the ecosystem, and thus have significant impact into driving narratives. Their relationships are then explored, in which a model for representing win condition in terms of what is used in narratives is proposed. Then, the proposed model is evaluated through a controlled comparison between machine learning models, in which the causal relationship between variables is investigated and the way in which they can be predicted and utilised is outlined. Lastly, this chapter explores some of the ways in which win condition can be incorporated into the broadcast, and how it can be used to enhance existing narratives.

Thus the contribution of this chapter is twofold. Firstly, the storytelling features which are commonly used in the esport broadcast ecosystem are identified. This is compared to the existing literature as well as empirically evaluated through the use of ML models. Secondly, a narrative focused win condition system is proposed. This system is designed to enhance existing narratives in such a way that it can be more readily applied to the esport broadcast ecosystem when compared to existing win prediction tools and knowledge. In doing so, this chapter addresses RQ4, depicting how ML can be integrated into an application system to enhance esport broadcast narratives.

5.2 Methodology

In order to create a system that is ecologically consistent with the domain, it is first important to understand the main topics of existing narrative, when related to commentators making predictions of outcomes. Therefore, a study of existing esport content is performed, in which the coverage of six distinct matches are analysed. The study focused on the draft-phase section of the matches, which is the phase in which characters are selected by both teams. During this phase, the main match has not yet began, therefore the commentary provided is highly speculative when compared to the descriptive - play-by-play and colour - commentary that is predominant during the game-phase. Therefore, broadcasters tend to focus their narrative on their own predictions and understanding of how the game may be played by both teams and what that could mean in relation to the outcome. This can provide an insight into the features and aspects of the game that broadcasters consider important towards winning, and more crucially, what they believe a team must achieve in order to emerge victorious.

Three games from two tournaments (for a total of six matches) were selected. Firstly, the lower bracket finals from the 2023 ESL One Berlin[1] was selected. The ESL One Berlin is a major Dota 2 tournament, with large coverage both in person and through live broadcast. For brevity, the three 2023 ESL Berlin Lower Bracket Finals games will be referred to as ESL Game 1 through 3 respectively. Secondly, to ensure a diversity of content is explored, the use of three games from a amateur tournament were selected. Despite being played by amateur players, the games were commentated by two professional broadcasters (Ted “Pyrion-Flax” Forsyth[2] and Jake “SirActionSlacks” Kanner[3]) who have both also produced content in the 2023 ESL One Berlin and other major Dota 2 tournaments. The games used were part of “Pyrion-Flax” weekly in-house tournament, in which members of Pyrion-

[1] `https://liquipedia.net/dota2/ESL_One/Berlin_Major/2023` (accessed 13-05-2025)

[2] `https://liquipedia.net/dota2/PyrionFlax` (accessed 13-05-2025)

[3] `https://liquipedia.net/dota2/SirActionSlacks` (accessed 13-05-2025)

Flax's streaming community take part in teams in a league format. The three games utilised took place in the 22nd August 2023. For brevity, the three games from this tournament will be referred to as PF Games 1 through 3 respectively.

The three ESL games provides insight of narratives while in a more formal context, as those are professional coverage of real-world tournaments. By contrast, the three PF games demonstrates a more casual coverage in a less formal setting. By incorporating these two setting into the analysis the samples provided can be representative of these two common environments that exist within the wider esport domain.

Audio and video footage for the draft phase of all games were extracted from the Twitch platform, and then transcribed using Descript[4]. Nvivo 14 [5] was used to perform a Content Analysis (CA) [84] in the transcribed data, in order to identify recurring topics in broadcast in relation to win prediction and the conditions for winning.

Data language (i.e. codes) were generated after familiarity with the data was gained. Codes are presented in Table 5.1 and discussed in detail in Section 5.3.1. Data was encoded post-familiarity after codes had been formalised. Coding was done in two sessions and manually assigned by the researcher. One additional validation session was also performed post-encoding. This was done with the original encoder and one additional encoder, to ensure consistency and data validity.

Once an understanding of the common narrative patterns can be established, it is important to model and formulate the causal relationships be-

[4]https://www.descript.com/

[5]https://lumivero.com/product/nvivo-14/

tween the topics being depicted and the outcome of the game. Structural Causal Model (SCM) have been shown to be an effective tool to determine and express causal relationships [122]. In turn, an exploration of causal relationships can also aid in explainable relationships, particularly when exploring win predictions [60]. Thus this chapter utilises SCM to identify the causal relationship between what is perceived by commentators to be important features in order to advice on the feature engineering for training and evaluating ML models to storytelling. Such ML models are then utilised in this study to formulate win condition narratives as a data-driven tool for enhanced esport coverage.

5.3 Features of Win Condition

To identify the features that constitute win condition, this study first performs a content analysis on data collected from real-world tournaments and environments. Then the resulting data is analysed and a SCM is proposed to identify the relationships between variables. Lastly, ML models are trained using the proposed relationships, serving as a data-driven method to evaluate the relationship of perceived features and allow for a win condition system to be established.

5.3.1 Content Analysis

Through the analysis of the draft phase of the games, it is clear that the narrative presented by broadcasters contains a series of speculations on what will happen in the game and why. Certain codes can be extracted when the narrative of the broadcast is analysed. Commentary tends to focus on similar aspects of the game to guide their predictions and explanations. The general codes have been outlined in Table 5.1.

Table 5.1: CA codes and occurrences

Main Code	Sub Code	Professional	Amateur	Total
Performance	Combined	59	21	80
Performance	Kills	31	12	43
Performance	Timings	20	4	24
Performance	Comfort picks	8	5	13
Meta	Combined	37	25	61
Meta	In meta	29	15	44
Meta	Not in meta	8	10	17

Kills

The code for Kills refers to the instances where broadcasters focus on the predicted score or the ability to secure or deny kills from the enemy team.

For example, the following extracted quote was observed in ESL Game 1:

> *My big question is are you gonna get through the wall of tide [referring to Tidehunder, a hero that has been selected and relating it back to a previous comment where they described it as a "wall" because of its survivability] like this hero is really gonna determine the game*

In this case we observe the narrative driving the attention to the high survivability of one of the heroes. The narrative implies that the ability of the hero to remain alive is a key aspect of the character, while putting pressure in the enemy team to kill them. In other words, the casters are outlining how a successful outcome for either team will be connected to whether or not the hero is killed during the game.

Another example extracted from the broadcast of PF Game 2:

> *How does he get anywhere near an enemy? My boy [referring to one of the players] gets chained up [referring to abilities which stops characters from being able to move temporarily] at any time, he's dead, it's over, there's nothing he can do.*

In this example the narrative expresses how a character is particularly vulnerable to being killed if their movement is disabled. This suggests that the commentators identified this character survivability as a relevant factor for a teams success, forming a narrative where they expect the character to be killed repeatedly, which reduces the perceived chance for that team to win the match.

Timings

This code contains content which refers to match duration and other timing related subjects. In other words, it includes references to time-dependent features, which can generally be encapsulated by the time of the match (i.e. duration).

One example extracted from PF Game 1:

> *Monkey King [character name], uh, hard counter to Axe [character name] in the laning phase [refers to the early stages of the game], but towards the end of the game can... Flip on its head.*

In this example, the casters are outlining how the team that has picked the character "Axe" has an advantage in the earlier stages of the game, as Axe performs particularly well against "Monkey King", being able to restrain the hero's capabilities at the early stages. However, as the game progresses the advantage shifts and "Monkey King" is no longer at a disadvantage. This puts great perceived importance on the timing of the game, which can determine the outcome.

Another example of casters emphasising time as a key factor in determining the outcome can be observed in this extract from ESL Game 3:

> *it's not like they [team 1] can't win late, but I definitely do favour them [team 2] in the later portion of the mid game more so.*

In this example, the casters are outlining how the team composition from team 2 is stronger the longer the game takes to finish. This means that team 1 would be expected to win the game faster, while team 2 would be perceived as having an advantage in later stages. Thus, if the match has a short duration, team 1 would be expected to win, while otherwise a longer match duration would benefit team 2. Another example extracted from the same game can be seen below, where the casters continue with the early and late advantage narrative:

> *Liquid [team 1] definitely, effectively saying, guys, we're gonna be here in 25 minutes, and 9Pandas [team 2] need to weather the storm and buy enough time.*

Comfort Pick

Comfort picks refer to instances where the narrative draws attention to a player's experience with a particular character. In this case, casters outlined how playing characters in which players have demonstrated particular abilities can be strongly beneficial to their perceived chance of winning.

The following example was extracted from ESL Game 1:

> *They're playing to their teammates strengths. Roger [player name] on this Chen [character name] or Enchantress [character name] has looked... he's the best Chen or Enchantress at the tournament.*

In this quote it can be observed how the broadcaster was outlining how Roger had previously demonstrated high degrees of proficiency while playing both Chen and Enchantress (two characters can be played similarly and are able to fulfil similar roles in a team). Additionally, the caster has outlined them as "the best" at both characters in the entire tournament, which demonstrates playing this character would increase their perceived expectation of winning the game. A similar pattern can be observed in the following extract (PF Game 2), where a hero ban is explained to be due to a comfort pick denial:

> *Science Whiz Ben [player name], uh, an OD [abbreviation of character name] specialist, which is why OD's been banned,*

In Meta

This extracts refer to casters outlining how changes in game design have disproportionately improved certain characters or items. This can be observed in the extract from PF Game 3:

Witch Doctor [character name] is probably, uh, in the best support at the moment

Another example extracted from ESL Game 2:

Monkey King. Uh, this is a hero that has been a bit overlooked, I think, in this meta. I think he's looking insanely strong,

In both examples we can observe how casters perceive certain heroes to be good in a given game iteration. It can be noted that the narrative outlined by them puts an expectation of victory if teams play those characters.

Not In Meta

While certain changes to the game may benefit some characters, others may be disproportionately weakened. This code contains narrative in which casters outline aspects of the game they perceive to have a negative impact in the expectation of winning the match due to patch changes. This can be seen in the example from PF Game 1:

whatever lane Marci [character name] is in is lost. Now that hero is an absolute dog [bad].

Another example of such narratives was extracted from Game 3 of the same tournament:

Sand King [character name], a hero that used to be good last week, um, now, no longer

In this particular example, the caster used the fact that this particular character had been changed recently to build their narrative.

Similar narratives can also be observed in professional games coverage, such as the two extracts from ESL Game 2:

> *Wisp [character name] has fallen out of priority a little bit for teams at this tournament.*
>
> *...*
>
> *been a pretty loosey hero overall, Nyx [character name].*

Content Analysis Synthesis

Overall, the content analysis shows that broadcasters rely on four main features to make predictions.

- Player past experience with a character
- Team composition and how that reflects with the current patch (i.e. the meta)
- The kills potential of a team
- The ideal timings for a team

As outlined in the literature, the features utilised by broadcasters have also been proven to be reliable factors for predicting the outcome, in particular player past performances, match duration and kills differentials [174]. Furthermore, game updates are designed to alter the balance of the game. This can significantly change the viability of a character [81] and the way in which they are played [166]. Therefore, comparing the team composition not only to each other, but also to the current patch can lead to a better understanding of the context in the game [166]. Thus, the narrative points raised by casters generally align with features used to produce, or are known to impact, predictions in the ML literature.

5.3.2 Structural Causal Model

Generally, win prediction models use some form of data to represent the game, (or game state) to predict the winner. This can be represented in a SCM at its most abstract form depicted in Equation 5.1. However, the proposed model in this study attempts to predict the game state needed to win the game. Thus, a win condition model can be broadly represented in the SCM depicted in Equation 5.2.

$$Winner := f_{Winner}(State) \tag{5.1}$$

$$State := f_{State}(Winner) \tag{5.2}$$

Content analysis of the ecosystem outlined the key factors used by broadcasters to ascertain the state. Therefore the variable "State" can be broken down into player historic performance & preferences (*Player*), team composition (*Lineup*), *Kills* and match duration (*Duration*). However, those variables have complex causal relationships between them, so in order to produce a model, it is important to identify and attempt to model this structure.

Using these features, it is then possible to break down the win condition causal relationship as follows:

$$\begin{aligned} Duration &:= f_{Duration}(Winner, Lineup, Kills) \\ Kills &:= f_{Kills}(Winner, Lineup, Duration) \\ Lineup &:= f_{Lineup}(Player) \\ Player &\in U \end{aligned} \tag{5.3}$$

Firstly, Kills and Duration are the only variables that change once a game starts. Therefore, for the purpose of describing a condition to be reached in game, those are the main variables that will be used to describe the state goal. However, both Kills and Duration have a direct causal relationship with Player and Lineup, as the specific values for these features may have an impact in the number of kills present in the game and the overall match duration. Furthermore Lineup is assumed to be a function of Player, as players pick their characters and they are presumed to know their experiences and preferences. Lastly, although a relevant feature, Player is identified as an unobserved variable (U). This is a limitation of this study as measuring each individual player is beyond the scope of this chapter. Certain aspects of the literature have represented this feature by identifying their past performances with the character [174], however this approach is limited to specific players and the general approach proposed by this study is not player specific.

As outlined by the full breakdown of the space, modelling it into a single SCM graph is infeasible, due to the complex and cyclic relationship observed between Kills and Duration. Thus, a simplification of the space can be proposed, in which 2 models are explored, depicted in Equation 5.4 as well as in the SCM graph displayed in Figure 5.2.

$$\begin{aligned} Model\ (1):\quad & \mathbb{E}_{Kills}[\mathbb{E}[Duration|Winner, Lineup]] \\ & or \\ Model\ (2):\quad & \mathbb{E}_{Duration}[\mathbb{E}[Kills|Winner, Lineup]] \end{aligned} \tag{5.4}$$

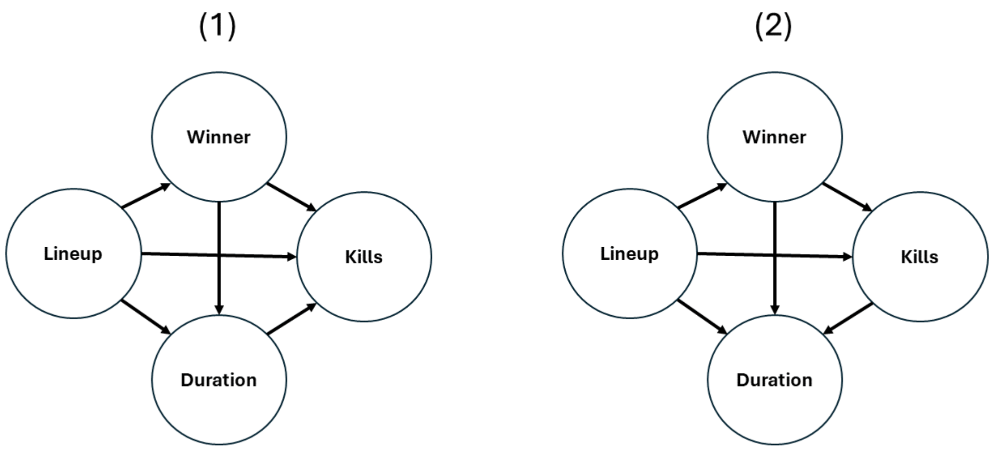


Figure 5.2: A SCM representation of the proposed simplification of the space

As both proposed models are simplification of the space, some aspects of the relationship is lost in the abstraction. However, the relationship may display a stronger asymmetry in one of the models, given additional confounding variables which may not be trivially understood. In this case, a ML model may perform more accurately and more consistently given one of the two input features as modeled by the simplification proposed. This is explored by training a set of ML models - in this case as predictive neural networks (NN), as depicted in Equation 5.5.

$$\begin{aligned} &Model\ (1) \Rightarrow \\ &NN_1 : (Winner, Lineup) \mapsto Duration; \\ &\underline{NN_2 : (Winner, Lineup, Duration) \mapsto Kills} \\ &Model\ (2) \Rightarrow \\ &NN_3 : (Winner, Lineup) \mapsto Kills; \\ &NN_4 : (Winner, Lineup, Kills) \mapsto Duration \end{aligned} \tag{5.5}$$

By comparing the averaged performance of both models, it may be possible to identify a stronger asymmetry between variables. This can lead to an exploration of the space, which can be integrated into the broadcast of esport content. In this exploration, rather than attempting to predict the *Winner*, a depiction of both possible outcomes is displayed, with the predicted necessary *State* for both teams being identified, such that $\mathbb{E}[State|Winner = 1, Lineup]$ and $\mathbb{E}[State|Winner = 0, Lineup]$. Where *State* is defined by the resulting features as identified by the best performing series of models (1) or (2) based on the combined output by their respective NNs.

5.4 Modeling Win Condition

Once the main narrative features are identified, ML models can be used to represent and evaluate win condition. In order to train such models, appropriate data must be collected.

Firstly, to account for patch changes and patch specific metas, the use of a patch-aware representation of heroes is utilised. As described in Chapter 3, by leveraging game design data through patch notes, it is possible to perform patch agnostic analysis by changing the way in which characters are represented to account for their capabilities, rather than other forms of encoding (such as character IDs). For this reason, CCR was used to represent the lineup feature.

Data from all professional games for Dota 2 for which CCR was available at the time of the study was collected in order to train and evaluate ML models. This was collected through the Open Dota platform, in the form of SQL queries, which can be extracted free of charge using the explorer feature[6]. Data collected for those games included:

- IDs of character selected (later encoded into the CCR equivalent [123])
- Patch in which the match took place (used to translate character IDs into CCR)
- Total number of kills obtained by the Radiant team (i.e. team 1)
- Total number of kills obtained by the Dire team (i.e. team 2)
- Match duration in seconds
- is team 1 winner (represented as a binary variable - note that it is not possible to draw in *Dota 2*).

These features can then be related to the states defined in Section 5.3.2, where: The Lineup feature is represented by the CCR of the selected characters. Since the CCR of a character is representative of their intrinsic capabilities for a given patch, a machine learning model can leverage the feature

[6] `https://www.opendota.com/explorer`

to identify relationships and mechanics that are connected to what a player may perceive as the meta. Furthermore, in the same way that $Player \in U$ is assumed, it is also assumed that professional players will generally select characters they identify as being in the meta. Thus, the data being used to train a neural network would be curated by the players themselves to reflect the meta, and only the interactions of capabilities would need to be detected by a ML model. This is facilitated by the way in which CCR is designed, and by the patch-aware values it provides. The remaining variables (*Winner*, *Duration* and *Kills*) are represented by their respective values directly, where $Winner{=}1$ when Radian (team 1) wins, and $Winner{=}0$ when Dire (team 2) wins, *Kills* is the total number of kills acquired by each team and *Duration* is the duration of a match in seconds.

A total of *59,019* games were collected for this study from 6 distinct patches (7.27 to 7.32). Only games which reached an end state were included, to ensure no games with server errors or other technical issues were included.

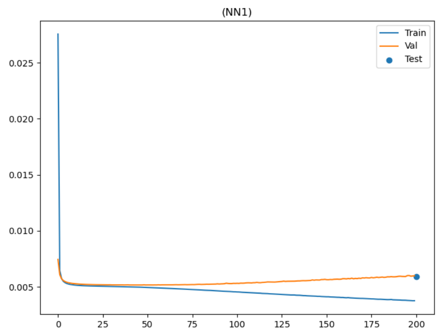
In order to train and evaluate the models proposed, four neural network models were trained (NN1-4) as described in Section 5.3.2. NN1 utilised the *Lineup* (encoded with CCR) and the game outcome (*Winner*) as input features to predict the final duration of the game. NN2 utilised the same features as well as the match duration in seconds to predict the final score for each team. These two NN combined represent Model (1), which assumes a stronger asymmetry between *Duration* and *Kills* (i.e *Duration* is easier to predict without *Kills* than *Kills* are to predict without *Duration*).

Conversely, NN3 utilised the outcome and *Lineup* to predict the total number of kills for both teams, while NN4 utilises both variables as well as the number of kills to predict the duration. This constitutes Model (2), which assumes a stronger asymmetry between *Kills* and *Duration*. Comparing the performance of the four NNs allows for an investigation of the two proposed simplifications for the structure causal models.

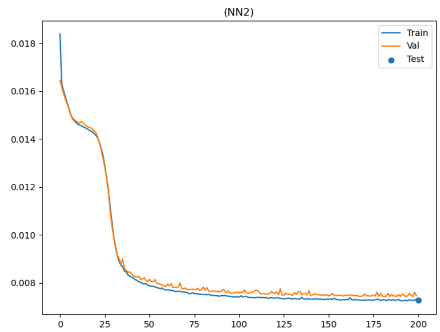
All four models were trained using the same architecture, consisting of 4 hidden layers of [128, 64, 32, 8] neurons each. ML models were trained for 200 epochs with a batch size of 512 using an Adam optimizer. The dataset was divided using a 60/20/20% split between training, test and validation datasets respectively, with the loss graphs depicting the training process available in Figure 5.3.

As demonstrated in Figure 5.3, the training and validation graphs for NN3 indicate that the model started to overfit to the data rapidly, with a noticeable deviation between the training and validation graphs by the 50$^{\text{th}}$ epoch. Similarly, graphs for NN1 and 4 both demonstrate that the networks were affected by overfitting with a noticeable deviation by the 100$^{\text{th}}$ epoch. While Figure 5.3b does not demonstrate any apparent overfitting, the overall performance of the network had plateaued rapidly by the 50$^{\text{th}}$ epoch.

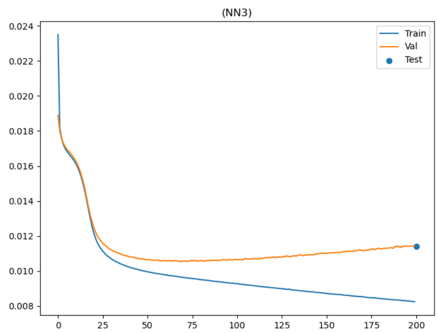
Furthermore, it can be noted that both models that predict the duration (NN1 and 4) outperformed the models attempting to predict the number of kills. The test loss of NN1 can be observed as approximately 6e-3 and NN4's at 4e-3. In contrast, the loss observed in the test dataset for NN3 is approximately 1e-2, where even when provided the duration of a match, NN2's test loss was observed at approximately 7e-3. While the differences in performance are likely due to the properties of the features, it does suggest that a stronger asymmetry between *Duration* and *Kills* is observed (albeit as



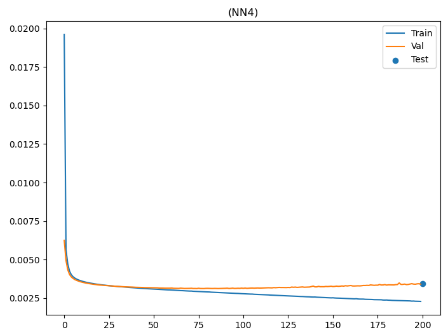
(a) NN1 Loss: Predicting duration without kills



(b) NN2 Loss: Predicting kills with duration



(c) NN3 Loss: Predicting kills without duration



(d) NN4 Loss: Predicting duration with kills

Figure 5.3: Graphs demonstrating the training, test and validation losses for all 4 NNs individually

a consequence of the prediction itself). This is also supported by the values of the training losses themselves when compared between all four NN, as demonstrated in Figure 5.4, where all losses for training were overlaid for ease of parsing. Furthermore, Table 5.2 depicts the training, validation and test results observed by the 200th training epoch.

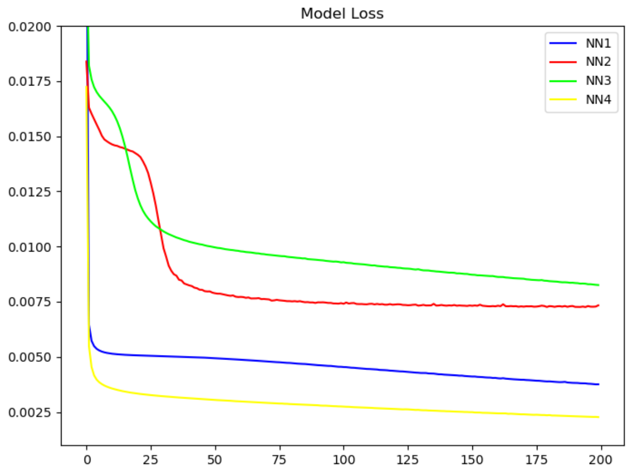


Figure 5.4: Loss graphs observed while training all 4 NN combined into one graph

Table 5.2: Train, Validation and Test loss values (rounded to 4 decimal values) of all 4 NN as of the 200th training epoch.

NN	Training	Validation	Test
NN1	0.0038	0.0059	0.0059
NN2	0.0073	0.0074	0.0073
NN3	0.0083	0.0115	0.0114
NN4	0.0023	0.0036	0.0035

5.5 Discussion

This chapter proposes the use of narrative focused ML models to produce a win condition system that is designed to be applied to existing broadcast content present in the esport ecosystem. The aim of the win condition model is to provide context to how each team could win the match, rather than attempting to predict who is most likely to win. Win condition as a conceptual narrative feature has been shown to be present in existing coverage of esport as shown in the analysis of ecological esport content (see Section 5.3.1). When analysing the content of esport commentary in the ecosystem, speculations of game states and outcomes have emerged as common patterns by professional commentators. By extracting and condensing the features that are already in common place when producing commentary, the ML approach of defining and predicting win condition can be more readily applied to the broadcast framework seen in the domain.

Furthermore, the observed features present in the coverage of esport content is in agreement to what is observed in the literature as reliable features for predicting the outcome. Firstly, the current game iterations and changes to the meta are consistent topics of narrative. Similarly it has been shown that the meta has a significant impact on the win prediction models, as well as relevant features to understand and represent the domain [51, 24]. Secondly, broadcasters consistently refer to the different stages of the game, stating their expected temporal advantages to different teams. This is understood as a reliable feature for predicting the winner, where the outcome of matches that are exceedingly short or long can generally be predicted more reliably [174].

Therefore, the features of importance for winning a game, as perceived by commentators, were identified in the SCM in Section 5.3.2. As observed in Figure 5.4, match duration as an output generally produces more accurate results than the total number of kills in a match. This is demonstrated as both NN1 and NN4 produce more accurate predictions than NN2 and NN3. This was observed despite NN2 utilising an additional feature (match duration), which includes knowledge that is only available at the end of the match. In contrast, NN1 does not utilise the knowledge of kills as an input, which contributes to demonstrating this phenomenon. Therefore, as demonstrated by the experiment, Duration seems to be more reliant on *Lineup*, given the *Winner*, than *Kills*. Conversely, *Kills* is more dependent on Duration, and cannot be as easily ascertained by ML models using only the *Winner* and the *Lineup*. This demonstrates an asymmetry in the causal relationships between the variables, where Duration is understood to impact the value of *Kills* more heavily than the inverse.

On the other hand, *Duration* as an input feature of *Kills* prediction seem to have a noticeable impact in the performance of the model, as observed when comparing Figures 5.3b and 5.3c. As shown in the differences in training and validation graphs, NN3 is observed to overfit rapidly, with a negative gradient trend line for the training graph and a positive gradient trend in the validation graph by the 50^{th} epoch. Conversely, NN2 displayed a relatively stable graph by the end of the training period, with no significant deviation between training and validation trends. This suggests that match duration as an additional feature helps prevent the effect of overfitting, thus allowing the NN to more easily generalise to unseen data.

It is also noteworthy that the models utilised in this study were designed using the same architecture. No hyper-parametrization was performed to fine-tune the training process or reduce the effects of overfitting or improve the performance. Therefore, the final obtained models could be improved with continued work in the design and training of the models. However, the results observed by the four NN can be interpreted as a controlled experiment, in which the SCM models proposed in Section 5.3.2 can be compared and contrasted.

When evaluating Model (1) and Model (2) it is clear that the State needed for a team to win can be more reliably represented in terms of Model (1). While Model (2)'s NN4 has been seen to produce the most reliable predictions, NN3 is also seen to produce the least reliable results of the four NNs, which creates a large disparity between performance of the predictions of the features. The effects of this disparity is worsen, when considering the use-case would require the output of the least reliable prediction (NN3) being used as an input feature to run NN4, which would impact the reliability of the resulting prediction, since the input data itself may not be reliable. Additionally, despite no attempts to mitigate the effects of overfitting, it is clear that NN2 has been affected the least by this phenomenon and thus can more easily generalise to unseen data. Furthermore, the differences in performance between NN1 and NN4 are amplified by the overfitting observed in NN1. This is expected, as no attempts were made to reduce the effects, therefore, given fine-tuning it may be possible to produce more generalisable predictions of the match duration with results that more closely aligns with what is observed in NN4. While the same argument can be made for NN3, the overall performance of the kills prediction still outlines a stronger asymmetry towards predicting duration, which suggests that Model (1) may be more reliable at predicting and depicting the State for data-driven narratives.

Thus, this chapter proposes a model for driving the prediction of win condition which aligns with Model (1) as suggested in Section 5.3.2. The win condition proposed has been depicted in Figure 5.5 where the use of NN1 and NN2 have been introduced to a dataflow diagram depicting how the State (as defined by *Kills* and *Duration*) could be generated to enhance existing storytelling narratives. In this figure, *Winner** represents one of $Winner = 1$ or $Winner = 0$, where in a real world use-case, both values are utilised to produce two distinct results, allowing broadcasters to compare their narratives with the predicted State depicted by Model (1).

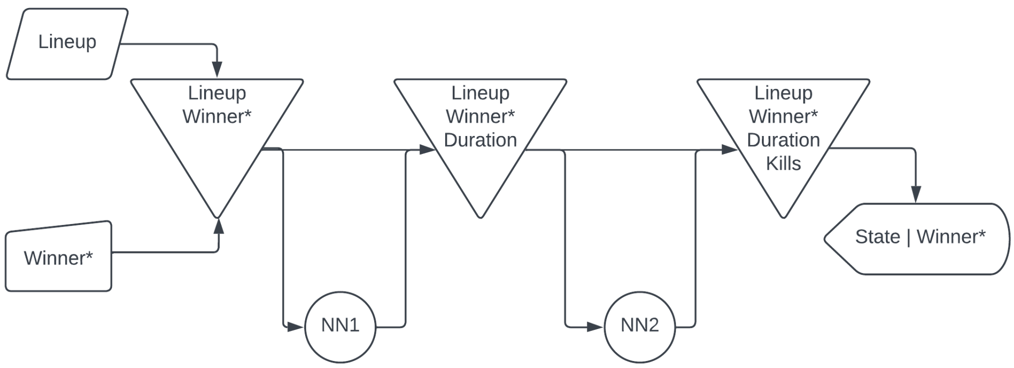


Figure 5.5: A dataflow diagram depicting how a prediction that depicts the win condition can be generated in a live scenario.

As the proposed model does not give insights into the most likely outcome, but rather depicts a State to be achieved for a team, it is purely a depiction of goals. This can provide greater narrative opportunities, and particularly as the feature engineering of the model was performed through the features that are already utilised in the existing ecosystem. By leveraging common talking points, and producing results that match commentaries, this model aims to enhance the experience without suffering from the dilemma present in general win prediction models, which can detract from the excitements and user needs of audiences and broadcasters.

5.5.1 Utilising Win Condition

In order to aid the understanding of how Win Condition, as defined by this chapter, can be integrated into a *Dota 2* esport broadcast, some example use-cases can be discussed. By reflecting and adopting the design considerations discussed in Chapter 4, a visualisation system for win condition is proposed and discussed in this section.

Firstly, in relation *branding & recognition*, this tool is not designed for a particular tournament or organisation. Therefore, the visualisation tool will follow Dota 2's branding to ensure a recognisable interfaces that is intuitive for broadcasters to use. Secondly, to ensure the tool follows the consideration for *controllable and delivery* in design, the interface provides full control of the system to the user, allowing them to integrate the ML insights into their narrative at their discretion.

Thus in order to follow both considerations, the interface for selecting the lineup of characters manually can be seen in Figure 5.6. As demonstrated in this figure, a user is able to click on the recognisable portraits for each of the characters to add it to the next available lineup. Users are similarly able to click on the hero's portrait in the team's lineup to remove them, which allows for other heroes to be added instead.

Furthermore, when a character is selected, its portrait in the original

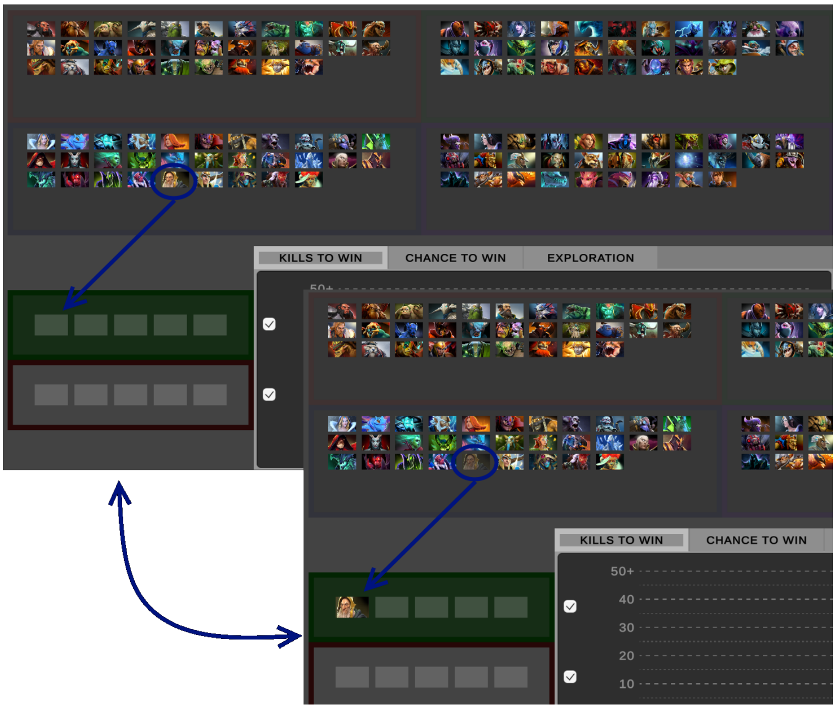


Figure 5.6: A demonstration of how a team lineup can be selected to be used within the visualisation tool, showing the changes before and after selecting a hero to be used in a team's lineup

character selection screen is greyed out, to demonstrate that it has already been used. This simulates the character selection screen behaviour in the game itself, which players are presented with. Similarly, characters are split into their four primary attributes, 'str', 'agi', 'int' and 'universal'. The order in which characters are sorted as well as the order in which the attribute bins appear also follow the same patterns as in Dota 2's character selection screen.

For comparison, Dota 2's character selection screen can be seen in Figure 5.7. Note that within Dota there are additional information being displayed that is not reflected in the tool. For example, by playing with a character in several matches, a player can increase their (out of game) level with the character. This is a distinct level system to what is experienced in the game, i.e. in game level always starts at 1 and is levelled up in the match, with any progress lost between games, while the out of game level increases between matches but bears no impact in an actual match and is purely for tracking. Therefore, the out of game level is a reflection of how often a player chooses to play a character and thus is not included in the visualisation tool, as it is not tied to any particular player.

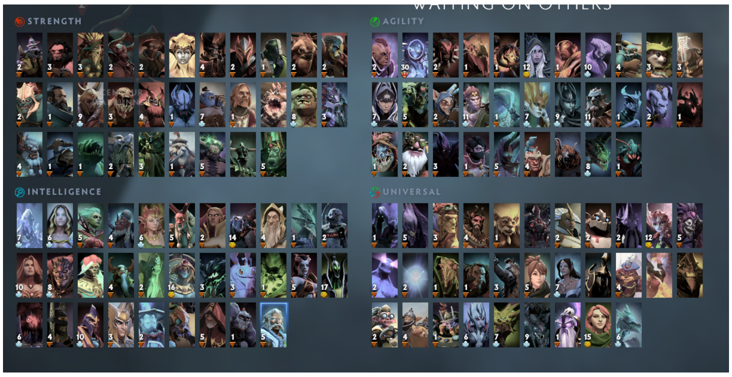


Figure 5.7: The official Dota 2 character selection screen

After the character selection is input into the tool, the win condition system described in Figure 5.5 can be executed. NN1 would produce two predictions for the duration (one for each team as a winner), which is then used as input for NN2, which would also generate two predictions (one for each team as a winner). This could then be presented textually. As an example, data from ESL Game 3 was used in the proposed Win Condition system. For comparison, the actual final result of this game concluded with a Radiant [team 1] victory at 21 minutes, 36 seconds with a kill score of 23 to 7 in favour of Radiant. However, in order to increase human readability, keeping it in-line with the *At-a-glance* design consideration, all Duration predictions (i.e. the output for NN1) were round to the nearest 5 minutes.

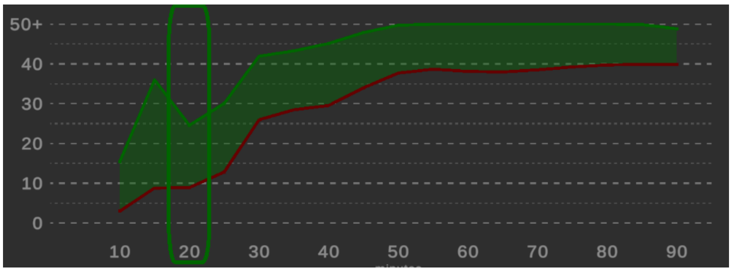
This step will undoubtedly have an impact in the kills prediction system (i.e. NN2), which is a trade off between performance and user-needs. The generated outputs could then be presented textually in the following way:

- Radiant: Duration 20 minutes - Kills 25 to 9.
- Dire: Duration 50 minutes - Kill 53 to 42.

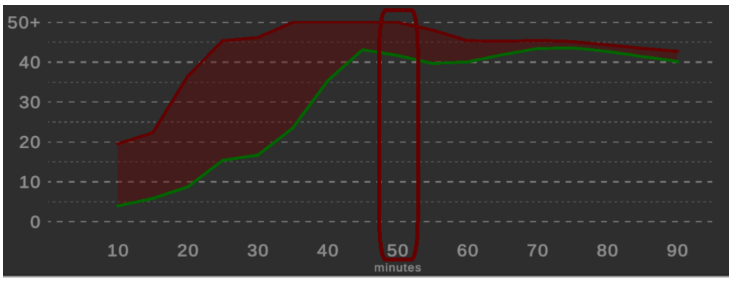
This intuitive presentation provides the minimum amount of information describing the system output. This use-case would require minimal amount of user-training and integration with existing broadcasters, which ensures the system follows the *Understandability* principle. Analysts would simply be presented with textual information and they may choose to integrate it with their existing narrative. However, this use-case provides minimal exploration. One possible alternative that may require some training by broadcasters is to integrate the results of the networks with an explorative graph. Figure 5.8 depicts how two graphs could be generated following an exploration. In this case, NN2 is run multiple times to predict the kills condition needed for each team at a wide range of match durations. The duration condition (as predicted by NN1) is also drawn, to showcase the primary win condition identified by the system.

In this proposed visualisation, each duration contains two lines connected together by a semi-transparent shading. The green line indicates the predicted amount of kills for Radiant, while the red line indicates the predicted amount of kills for the Dire (output of NN2). Graphs 5.8a and 5.8b are read independent of each other and refer to a particular team's win condition - i.e. $Winner{=}1$ for 5.8a and $Winner{=}0$ for 5.8a. In this graph, a wide difference between the two predicted lines (such as what is observed in Figure 5.8b at

30 minutes) indicates that the difference in kills between the two teams must be large for a team to achieve their win condition. Conversely, a shallow difference between the two predicted lines (Figure 5.8b at 80 minutes) suggests that a team does not require a significant amount of kills more than the enemy team.



(a) Kills predicted per duration (NN2) for Radiant [team 1] as a winner.



(b) Kills predicted per duration (NN2) for Dire [team 2] as a winner.

Figure 5.8: A kills prediction graph, outlining the win condition needed (as predicted by NN2) for each team to win at any given time. Lines in green outline the Radiant [team 1] predicted score, while lines in red outline the Dire [team 2] predicted score. The expected duration predicted by (NN1) has also been highlighted with a green/red box for each team.

These explorations graphs and the textual information would then be presented to broadcasters. The use of the textual information would provide *At-a-glance* benefits, while the exploration graphs provides *Additional context*, describing other possible win condition states beyond what is predicted by the win condition system.

5.5.2 Win Condition in Broadcasting

All of the example visualisations and insights provided have been generated from the models proposed (NN1 and 2) using ESL Game 3's lineup. Therefore, it is possible to discuss how they could be implemented alongside the narratives of the game. As explored in Section 5.3.1, the following two quotes were extracted from the game's draft phase:

> *it's not like they [team 1] can't win late, but I definitely do favour them [team 2] in the later portion of the mid game more so*
>
> *...*
>
> *Liquid [team 1] definitely, effectively saying, guys, we're gonna be here in 25 minutes, and 9Pandas [team 2] need to weather the storm and buy enough time*

In this case, the most simplistic textual insights could enhance both narrative, as it affirms the broadcaster assessment that Radiant [team 1] is more favourable to win at earlier stages. The graphs depicted on Figure 5.8a could also depict how Dire [team 2] would need to perform well at the earlier stages of the game to prevent Radiant [team 1] from reaching their win condition, being able to comment on the amount of kills needed by the Dire to "weather the storm and buy enough time".

Therefore, it is clear that the win condition system described in this chapter can be integrated into esport broadcast and that it can have a positive impact in the broadcast narrative. By utilising CCR as described in Chapter 3, the win condition system can reliably model the context of the patch, allowing it to perform patch agnostic analysis. Similarly, by designing a visualisation system that follows the design considerations outlined in Chapter 4, the win condition system can be readily applied into esport broadcasts. Additionally, the ML model explored in this chapter has been designed by using the narrative features already employed by commentators. This enables the tool to be integrated into existing narratives, enhancing the storytelling provided by broadcasters. For this reason, this chapter has addressed RQ4, demonstrating how ML can be integrated into an application system to enhance esport broadcast narratives.

The next chapter reflects on the previous chapters to formulate a systematic methodological framework. This framework aims at providing a practical set of guidelines to aid ML practitioners to apply ML to enhance esport broadcast narratives.

6 Methodological Framework & Guidelines

6.1 Introduction

The previous chapters explored the current state of knowledge in the field (Chapter 2), addressing RQ1 and establishing the need to investigate RQ2, RQ3, and RQ4. This enquiry resulted in three distinct case studies, each addressing one of these questions, as detailed in Chapters 3, 4, and 5, respectively.

This chapter consolidates the contributions of this thesis into a systematic methodological framework. The framework provides a structured approach to designing, developing, implementing, and evaluating ML solutions aimed at enhancing esports broadcasting narratives.

While this framework is primarily designed for ML practitioners, other stakeholders in the esports broadcasting ecosystem may also benefit from the findings outlined in this chapter. ML practitioners, defined as users with expertise in standard ML methodologies and technologies, can use this framework to create solutions that are more seamlessly integrated into esports broadcasts. By following these guidelines, practitioners can ensure their models are not limited in scope or application, enabling them to enhance broadcasting narratives across the esports domain more effectively.

This framework aims to provide practical guidelines for future work in the esports domain. The findings are derived from the academic contributions of this thesis and the broader state-of-the-art (SOTA) literature. However, the framework itself is not intended as a direct academic contribution but rather as a set of actionable methodological guidelines rooted in the research presented here.

6.2 Design and ML Principles

This framework draws on both standard ML processes and traditional design and User Experience (UX) principles. Instead of adhering strictly to conventional ML practices or design methodologies, it integrates elements of both disciplines. It emphasises end-to-end solutions, addressing the entire pipeline for creating ML models, from design and training to integration into esports broadcasts and evaluation within a broadcasting context.

User-Centred Design (UCD) serves as a key principle, focusing on solutions that address user needs and core problems [50]. UCD follows an iterative cycle of design, implementation, and evaluation, as illustrated in Figure 6.1. A solution is first designed based on identified user requirements, implemented as a prototype, and subsequently evaluated to inform further redesign and iteration in the cycle.

While terms like “design”, “implementation”, and “evaluation” are shared between ML and UCD, their meanings differ significantly in the two fields. A framework that blends traditional UX principles (such as UCD) with ML methodologies must carefully distinguish these terms. For clarity, this framework separates these disciplines into two domains: ML Space and Design

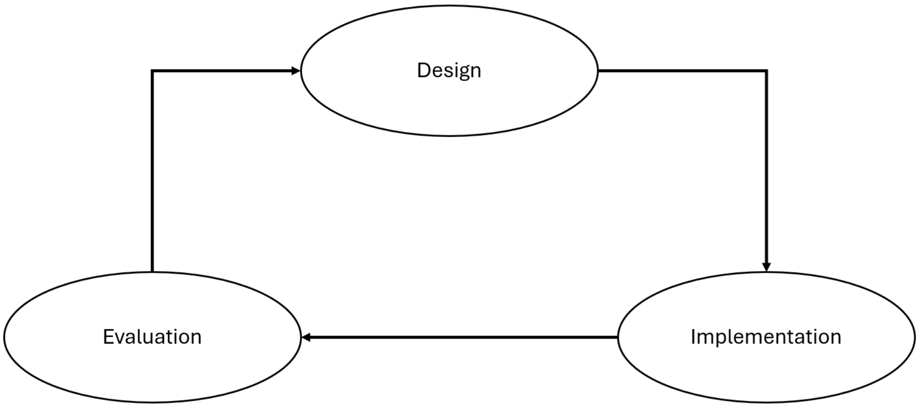


Figure 6.1: The design, implementation and evaluation cycle for User-Centred Design

Space.

In *ML Space*, the term “design” typically refers to model architecture design. This encompasses feature engineering, data cleaning, and structuring the model’s layers and connections, such as deciding on recurrent versus feed-forward architectures. Following this, the “implementation” step refers to deploying the trained model into its intended use-case, which usually occurs only after rigorous evaluation. “Evaluation” in ML Space focuses on model performance, often measured through quantifiable metrics like accuracy, which reflect how well the model performs its task against a known ground truth.

In *Design Space*, the term “design” pertains to the visual and functional aspects of the solution. This may involve crafting UI elements, designing the timing and flow of interactions, or addressing usability. “Implementation” refers to any version of the solution that allows user interaction, enabling data collection for evaluation. In this context, “evaluation” assesses usability and how effectively the solution meets user needs, which may involve quantitative metrics, such as task completion time, or qualitative insights from user feedback.

Figure 6.2 illustrates the differences between the UCD cycle and the ML pipeline.

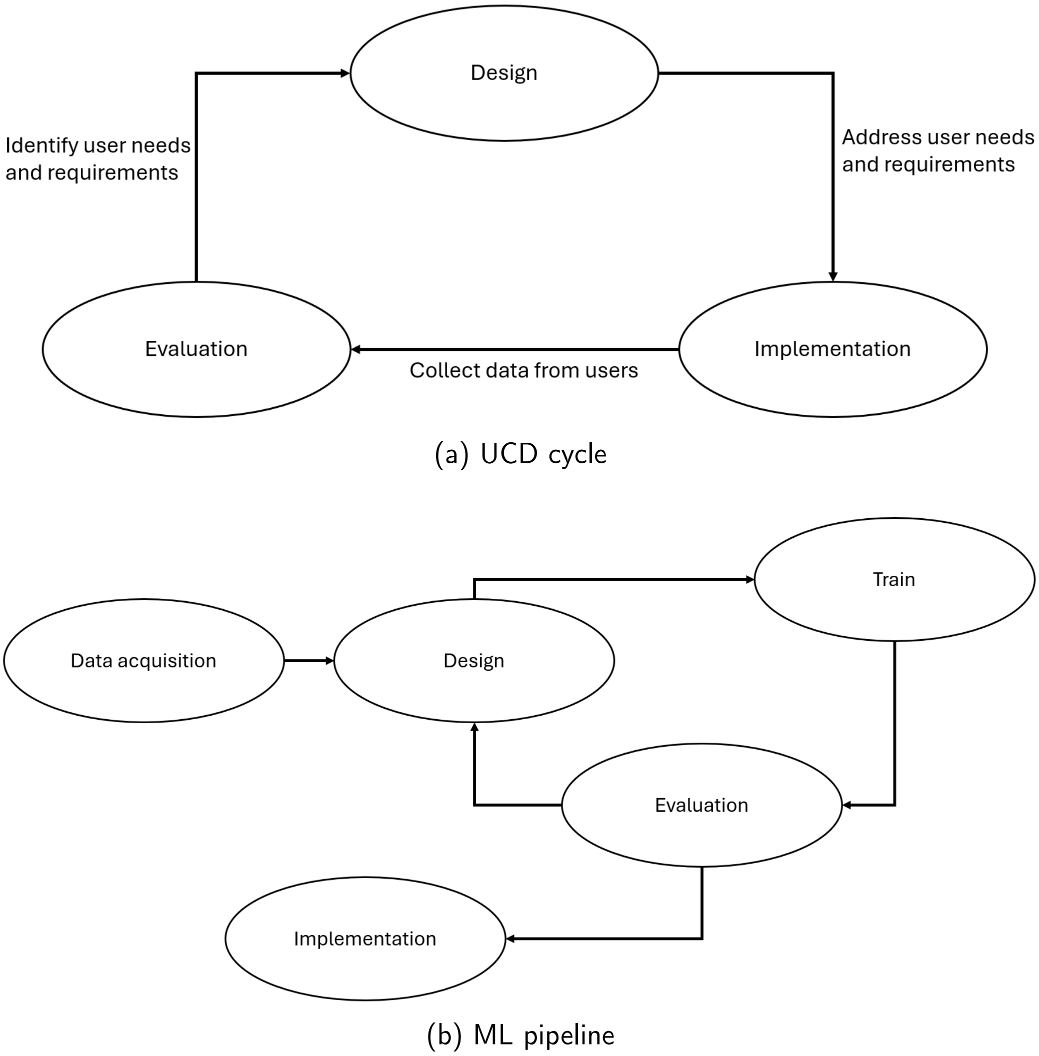


Figure 6.2: A depiction of the UCD cycle and the ML pipeline

As shown in Figure 6.2b, traditional ML development is guided primarily by data availability, with tasks defined by the data at hand. By contrast, UCD principles focus on user needs and requirements, iterating solutions to address these directly. This thesis demonstrates how combining these methodologies can enhance traditional ML practices.

For instance, Chapter 3 introduces a novel data representation methodology that improves model functionality and longevity by addressing real-world limitations. Chapter 5 explores user-centred model design, starting with user needs and requirements to guide data exploration. Similarly, Chapter 4 highlights how visualisation tools can be designed to integrate seamlessly into the ecosystem while addressing overlooked user requirements. The implementation process described in Chapter 5 further demonstrates how UCD and Design Space principles can enrich the ML pipeline.

6.3 Framework

The framework proposed in this chapter comprises a sequence of 10 steps. Figure 6.3 illustrates the flow of these steps, incorporating and preserving the key findings outlined in this thesis. The steps are as follows:

1. Problem Definition
2. Data Access
3. Explorative Analysis
4. Model Architecture
5. Model Training
6. Model Evaluation
7. Model Visualisation
8. Broadcast Visualisation

9. Broadcast Implementation
10. Broadcast Evaluation

As shown in Figure 6.3, the framework is organised into three cyclical stages. The first is the *Problem* cycle, which addresses problem definition and the identification of solution requirements, laying the foundation for subsequent steps. The second is the *Model* cycle, which focuses on the development of the model, encompassing training and various forms of evaluation to ensure its effectiveness. Finally, the third is the *Broadcast*, which cycle emphasises the implementation and integration of the solution within the esports broadcast domain.

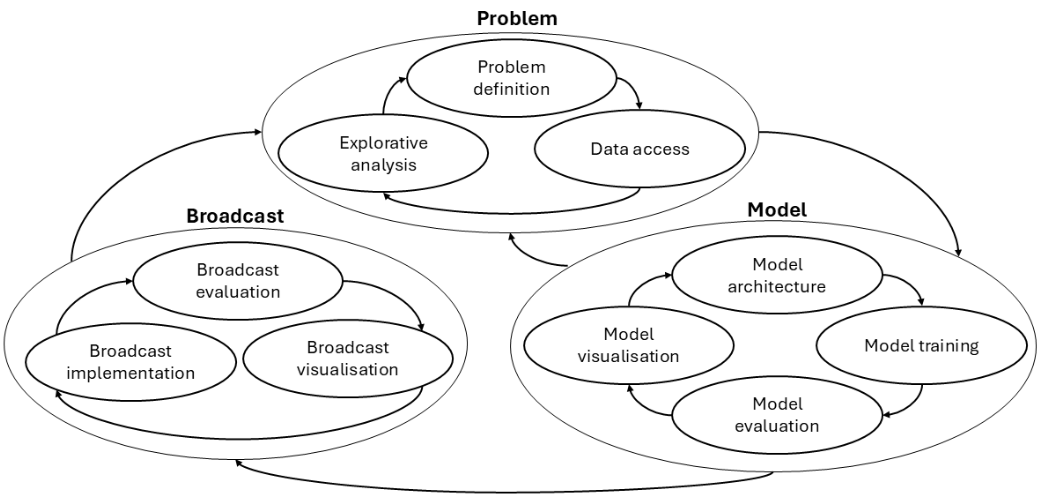


Figure 6.3: The framework diagram for applying ML to enhance esport broadcast narratives

It is important to note that the diagram provides a simplified representation of the process for clarity. In practice, certain stages may occur in parallel or follow a different sequence, depending on the specific use case or the unique requirements of the ML solution.

While some flexibility in execution is expected, the framework provides general guidance for applying ML to enhance esports broadcast narratives. This section describes each of the 10 steps in the order listed, from (1) problem definition to (10) broadcast evaluation, outlining their purpose and how they can be utilised effectively.

6.3.1 Problem Definition

In the initial phase of this framework, the primary goal is to comprehensively define and understand the problem that the ML solution seeks to address. Crucially, the problem should reflect a real-world issue. This could involve research-specific challenges, such as the difficulty of classifying player roles [29], or practical issues like the frequent obsolescence of ML models due to new game patches, as discussed in Chapter 3. Alternatively, the problem might relate to broader ecological challenges, such as the difficulty of following an esport title due to unclear indicators of match progression or winners [59], which can hinder new viewers from understanding the flow of the game. Identifying how the problem manifests and contextualising it within its ecological framework can provide valuable insights and considerations for the solution.

In addition to defining the problem, it is equally important to identify the key stakeholders involved. Esports broadcasting operates as a multi-stakeholder system, with diverse needs, demands, and expertise across various groups. Stakeholders may include, for example:

- ML Practitioner & Data Scientist - Individuals who analyse data and generate insights, often focusing on technical and analytical aspects of the solution.

- Content Producer - Those involved in the production side of the broadcast, such as observers (digital camera operators), referees, or the technical team managing the live stream.

- Content Creator - The broadcasters and commentators who engage the audience by delivering narratives and providing live storytelling during matches.

- Sponsors - Financial backers of tournaments, events, teams, or players, who support esports initiatives in exchange for brand exposure.

- Professional Player - The competitors featured in the broadcast, whose focus lies on high-level performance and gameplay.

- Audience - The viewers who consume esports broadcasts, with diverse motivations ranging from entertainment to learning gameplay strategies.

This list is not exhaustive but illustrates some of the primary stakeholder groups within the esports ecosystem. Identifying the relevant stakeholders for the specific problem is essential for understanding the varying user needs that must be addressed. Analysing these needs and determining how the problem impacts each group can help ensure that the architecture of the ML solution aligns with the real-world challenges they face. Ultimately, this alignment enables the development of solutions that are both effective and relevant to the broader esports context.

6.3.2 Data Access

In addition to understanding the problem, it is crucial to evaluate the data associated with the problem and determine how it can be retrieved. Even the most effective model would fail if it relies on data that cannot be accessed in a representative use-case, rendering the solution inapplicable and unable to address the underlying problem.

For instance, if the model is intended for live deployment, it is essential to identify which features can be retrieved in real-time. Additionally, some features may be available live but only at low frequencies. For example, a game API might provide in-game information at second-intervals, offering periodic snapshots of the game state but no data for events occurring between these intervals, such as the case with Dota 2's GSI [10]. Such data limitations can significantly influence how the solution is designed.

To address these challenges, it may be necessary to explore multiple data sources or modalities. Considering alternative or supplementary data inputs can help mitigate constraints and ensure the model remains effective in its intended use-case.

6.3.3 Explorative Analysis

With a clear understanding of the problem and the related data, the next step is to explore potential solutions and define the specific task for the ML model. This stage involves identifying the role the model will play in addressing the real-world problem. For instance, if the issue is that team-fights are being missed in broadcast coverage, the task for the model could be to predict team-fight locations.

Exploring and analysing potential solutions requires a clear understanding of the model's intended use. Figure 6.4 provides an example of how models can be classified along two axes: "entertain" and "inform". The X-axis represents the extent to which a model is intended to provide entertainment value, while the Y-axis reflects its focus on delivering informative insights about the match. The diagram includes examples from the literature, as well as the win condition model described in Chapter 5. It should be noted that the values assigned to each model are illustrative and based on the researchers' experience; they do not necessarily represent precise measurements. However, this visualisation can serve as a useful conceptual tool for understanding the intent and use-case of a model.

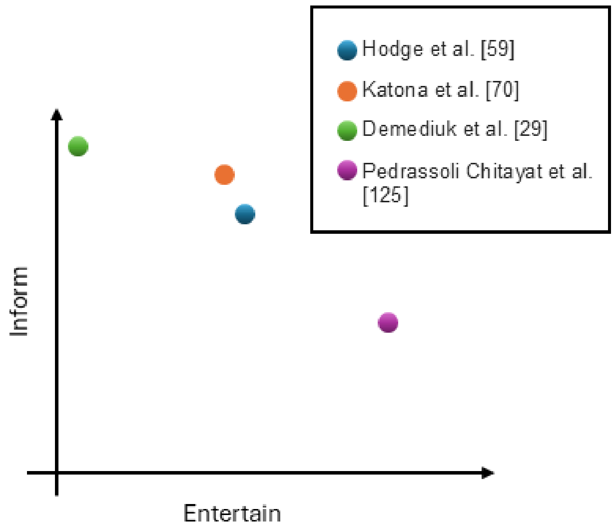


Figure 6.4: Example models classified along the inform and entertain axes

For example, the role classification model described by Demediuk et al. [29] has a high value on the inform axis and a low value on the entertain axis. This is because the model is designed to provide meaningful information, aiding future models in abstracting the environment, without a direct intention to entertain audiences.

By contrast, the win condition model discussed in Chapter 5 (Pedrassoli Chitayat et al. [125]) is designed to integrate with speculative narratives, focusing more on entertainment. While it provides data-driven insights, its explorative nature prioritises enhancing narratives over accurately predicting future game states, particularly when compared to the win prediction model by Hodge et al. [59].

Understanding the intent of the model's task is essential for defining appropriate metrics and constraints. For models designed to inform, accuracy and truthfulness must take precedence. This often involves well-defined metrics, as seen in the role detection work by Demediuk et al. [29], where roles in Dota 2 are clearly understood. In contrast, models intended to entertain may prioritise storytelling objectives, striking a balance between narrative enhancement and factual accuracy.

This balance is explored in Chapter 5, where the limitations of traditional win prediction models are identified (Section 5.1) and data is analysed (Section 5.3.1) to address user needs. The analysis highlights the narrative features that the solution must accommodate, providing clarity on how the task should align with storytelling requirements.

During explorative analysis, it may be helpful to consult relevant stakeholders, even if they are not direct users of the solution. For example, while designing the WARDS visualisation tool (Chapter 4), senior stakeholders at the tournament organising company were interviewed. Although they did not interact with the final tool, their feedback introduced critical requirements, such as branding considerations.

Prototyping - whether functional or not - can also aid this process. Sketching a prototype can reveal functional requirements and help stakeholders articulate user needs without requiring a fully trained or implemented model. This ensures that the final model is fit for purpose.

At this stage, it may be necessary to revisit earlier steps, such as refining the problem definition (Section 6.3.1). This iterative process, as exemplified in Chapter 5, allows for further analysis, such as exploring important features through Structured Causal Models (Section 5.3.2).

This iterative approach also facilitates solution design from a Design Space perspective. For instance, Chapter 4 describes stakeholder meetings that identified key requirements, culminating in a post-prototype meeting to evaluate a mock solution. Although the model in this case had been pre-trained as it was based on existing work, the solution's design could still be refined independently of the model's functionality at this stage.

Alternatively, if the problem and data are well understood, it may be possible to proceed to the *Model* cycle in the framework, which begins with model architecture as described in Section 6.3.4.

6.3.4 Model Architecture

The model architecture should be designed to reflect the task and problem defined in the earlier steps of this framework. It is essential that the model aligns with the accessible data, relying only on representative datasets that can be retrieved in the intended use-case.

Standard methodologies for designing ML models should be followed, with the complexity of the architecture tailored to the specific task. While more complex architectures may be required for certain problems, these typically necessitate larger datasets and longer training times. Therefore, it is crucial to ensure the model architecture accounts for the available time frames, dataset frequencies, and the quantity of data.

In some cases, such as the example discussed in Chapter 5, the optimal architecture for a given problem may be unclear. When this occurs, an comparison of different architectures (see Section 5.4) may be necessary to identify the most suitable approach.

Additionally, the model architecture should consider the intended functionality of the solution. For instance, if the solution is expected to provide long-term utility, it may be necessary to design the model to remain robust across future iterations of the esport title. This could involve adopting strategies such as the CCR methodology discussed in Chapter 3.

6.3.5 Model Training

The training stage of the ML pipeline is essential to ensure the model can learn from the data and produce meaningful, data-driven outputs. As discussed in the previous step, the training process should align with the overall time frame of the solution, balancing model complexity and performance against the available computational resources and deadlines.

It is recommended to follow conventional best practices for ML training, which include selecting appropriate loss functions, optimisers, and evaluation metrics suited to the problem domain. Additionally, the choice of model architecture will significantly influence the training process, dictating factors such as the amount of data required, the need for hyperparameter tuning, and the expected training time.

Ultimately, the training stage serves as a foundational step in the overall ML pipeline, and while it may not require novel approaches, careful adherence to established practices is critical for achieving reliable results.

6.3.6 Model Evaluation

Similar to the training phase, model evaluation is a crucial step in the ML pipeline. It provides the first indication of the model's performance, which ultimately influences how well it addresses the problem defined in earlier stages of this framework.

It is essential to ensure that the evaluation reflects the model's intended functionality. For instance, as discussed in Chapter 2, win prediction models are often presented to audiences as indicators of which team is currently winning. However, these models are typically evaluated by comparing their predictions against the final match result. This approach can be problematic, as the final winner may not always represent the team that was winning at every stage of the match, as noted in Chapter 5. Therefore, it is important to design models that can be accurately evaluated against their intended tasks, ensuring they meet the specific use-case requirements.

Once an appropriate evaluation metric is defined, conventional methodologies should be followed. For example, testing the model with unseen data is a standard practice to ensure that it generalises well to new contexts and datasets.

6.3.7 Model Visualisation

In addition to numerical evaluation, internal visualisation tools can help assess how the model performs against its intended task. This may involve conventional practices such as plotting training and validation learning curves, as demonstrated in Chapter 5 (see Figure 5.3).

Beyond these conventions, creating simple visualisations of the model's output within its intended context can highlight potential issues early. For example, the hypothetical visualisation in Chapter 5 illustrates how the model's output might be used in an esports broadcast. Such visualisations enable researchers to qualitatively evaluate the model's performance within a real-world use-case, ensuring its outputs align with environmental expectations.

At this stage, the visualisation should serve as a prototype focused on understanding the model's behaviour in context. This could involve creating an approximation of the final broadcast visualisation or depicting an aspect of the model that is more easily visualised. The goal is to validate that the model performs as expected, particularly in relation to the problem defined earlier in the framework.

If the model's performance does not meet expectations, it may be necessary to revisit the Model Architecture step (Section 6.3.4) for refinement. Similarly, revisiting earlier steps (Section 6.3.1) may help further refine the problem, data, or task definition.

Conversely, if the model achieves satisfactory performance, the process can move to the final cycle of the framework (*Broadcast)*, starting with Broadcast Visualisation (Section 6.3.8).

6.3.8 Broadcast Visualisation

Once the model is ready for deployment, the next step is to design the visualisation system for use in the broadcast. This may involve directly displaying the model's output or creating a more complex visualisation system. For instance, the win condition system in Chapter 5 includes both textual output and graphical representations of the models, while the WARDS graphic in Section 4 integrates multiple features to provide additional narrative context.

This step focuses on creating the user-facing interface that integrates the trained model with the design developed during Explorative Analysis (Section 6.3.3). While implementing the design, it is important to validate it against the considerations outlined in Chapter 4.

Technical constraints may require deviations from the original design, but all identified user requirements (Section 6.3.1) must still be met. By the end of this step, the solution should function effectively within a controlled environment. This is also an opportunity to engage stakeholders to validate the design and usability of the solution.

6.3.9 Broadcast Implementation

The Broadcast Implementation step involves integrating the solution into the esports broadcast. For audience-facing solutions, this may include embedding the visualisation within the broadcast feed so it is visible during live matches.

This phase also involves ensuring the solution functions within the operational environment. For instance, specific setup requirements may be necessary for data collection or real-time analysis, as described in Chapter 4. Collaboration with stakeholders, such as tournament organisers, may be essential to facilitate this integration.

User training is another critical aspect of implementation, ensuring that those using the solution can operate it effectively. Additionally, data collection during implementation - whether observational or utilisation data - forms the basis for the Broadcast Evaluation phase (Section 6.3.10).

A test implementation in a controlled environment can help identify and resolve issues before full deployment. This allows for an additional cycle of refinement across the Broadcast Visualisation, Broadcast Implementation, and Broadcast Evaluation steps.

6.3.10 Broadcast Evaluation

The final step involves evaluating the implemented solution within the broadcast environment. This evaluation focuses on determining whether the solution meets the user needs identified in the *Problem* cycle of the framework.

General evaluation metrics should also be assessed, such as whether the solution was successfully integrated into the broadcast and whether users could utilise it as intended. Critically, the evaluation should reflect on the solution's impact on the broadcast narrative. For example, observational data collected during the deployment of the WARDS visualisation (Chapter 4) enabled reflections on how the tool influenced the narrative.

Evaluation methods may include post-event user interviews, which can be qualitatively analysed to inform how the solution addresses the problem. Likert-scale questionnaires [110], or other quantitative approaches such as usage data [82, 126] or eye-tracking [18] may also be used evaluate the solution. Pairing these insights with reflections on narrative impact can reveal both the benefits and shortcomings of the solution and its impact in the esport broadcast narrative.

If significant user needs remain unmet, this may necessitate revisiting earlier steps, such as the Problem Definition phase (Section 6.3.1). It is not advisable to revise model architecture (Section 6.3.4) without first reassessing the problem and associated data to ensure alignment with user requirements.

6.4 Framework Conclusion

This chapter presents a novel methodological framework designed to help ML practitioners integrate ML solutions into esports broadcasts. The framework provides actionable guidelines for applying ML to enhance broadcast narratives, consolidating findings from Chapters 3, 4, 5, and the broader literature (Chapter 2).

By centralising the contributions of this thesis, the framework offers a practical methodology that can be readily applied in future ML model development for esports broadcasts. It ensures that ML solutions are not only effective but also seamlessly integrated into real-world contexts, enhancing the quality and impact of esports narratives.

7 Conclusion & Future Work

This thesis investigates how ML can be applied to enhance esports broadcast narratives. In doing so, it contributes to the field of ML for esports, enabling future solutions to achieve a deeper and more meaningful impact on real-world broadcasts. By enhancing esports audience experiences, these contributions aim to bridge the gap between academic advancements and practical applications in esports broadcasting.

To this end, this chapter reflects on the research goals outlined in Chapter 1 and evaluates how the research questions were addressed.

7.1 Research Goals

This section revisits the research goals presented in Section 1 and reflects on how each research question was addressed throughout this thesis. Each research question is explored individually to provide a comprehensive overview of the findings and contributions.

7.1.1 Research Question 1

This research question asks "how is ML currently being used within the esport broadcast domain for storytelling and broadcasting narratives?" To address this, a comprehensive literature review was conducted in Chapter 2.

The review focuses on how ML techniques are being utilised alongside esports broadcasts. It highlights that data-driven storytelling plays a critical role in esports broadcasts, particularly as a tool for commentators, who serve as pivotal figures in the broadcast ecosystem. The literature underscores that commentators not only face significant challenges in crafting immersive and engaging narratives but also benefit greatly from purposefully designed tools to support their role. The positive reception of data-driven storytelling tools among commentators and broadcasters further emphasises the importance of developing such solutions.

A major limitation identified in the esports ML literature is that models rely on specific versions of their respective esports titles. Esports games frequently undergo balance changes through patches, which significantly alter gameplay. These updates aim to maintain fairness, introduce new content, and prevent the game from becoming repetitive. However, such changes can impact the performance of previously trained ML models, potentially rendering them ineffective or obsolete. Despite this being a well-known challenge, there is little to no reflection in the literature on how to address it. This gap informed the formulation of RQ2, which focuses on addressing the issue of model longevity.

Another gap identified is the limited utilisation of proposed models within live esports broadcasts. The majority of works in the literature focus on model performance and accuracy, proposing novel techniques for model architecture or training. While these contributions are valuable, their lack of practical application restricts their impact to primarily theoretical and methodological advancements. Few studies offer reflections on the practical considerations necessary to implement these models in real-world broadcasts. This lack of focus on real-world applications highlights the need to address RQ3, which seeks to bridge this gap by exploring how ML models can be integrated into live esports broadcasts.

The final gap pertains to how models are designed and evaluated. The literature's consistent emphasis on performance and accuracy often neglects usability and the meaningful application of model outputs. For example, win prediction models are commonly evaluated based on accuracy relative to the final match result, even though this metric may not align with user needs or expectations. As outlined in Chapter 2, some studies suggest that users do not find these results meaningful or impactful, contrasting with the needs of commentators. These findings highlight the importance of developing tools that utilise more sophisticated data-driven storytelling techniques to support narrative creation. The lack of literature investigating how to design and evaluate models that meaningfully impact broadcast narratives informed the development of RQ4.

In summary, the literature review in Chapter 2 reveals a clear need for advanced data-driven storytelling solutions that can be integrated into live esports broadcasts. It identifies three major gaps that drove the formulation of RQ2, RQ3, and RQ4. The first gap relates to ensuring model longevity (RQ2). The second addresses the usability of models within live broadcasts (RQ3). The third focuses on making models impactful for broadcast narratives (RQ4).

7.1.2 Research Question 2

The literature review in Chapter 2 outlined that a significant challenge faced by esports ML models is the impact of game patches on pre-trained models. To address this issue, RQ2 asks "how can ML models sustain performance and functionality beyond initial game patches?" This question is explored in Chapter 3, which introduces a novel methodology leveraging game design parameters to enable patch-aware representations of input features.

Specifically, Chapter 3 proposes the CCR methodology, which utilises game patch notes as input features for a clustering model. This clustering approach is then used to represent character selection data in ML models, replacing the conventional use of character IDs. By incorporating patch contexts into feature sets, this thesis demonstrates that patch-aware inputs enable models to perform analyses that are agnostic to game patches.

Additionally, the methodology proposed in Chapter 3 has been made freely available, supporting future research in the field. The study further demonstrates that CCR can detect the impact of changes introduced by new patches immediately, offering a significant improvement over traditional methods like character IDs. This capability allows ML practitioners to measure a model's potential reliability in a new game context, even before sufficient match data is available.

In conclusion, Chapter 3 addresses RQ2 by presenting a solution that can be integrated into future ML models, extending their performance and functionality beyond the patches they were originally trained on.

7.1.3 Research Question 3

A key limitation of the existing academic literature identified in Chapter 2 is that most ML models proposed in the literature are not incorporated into esports broadcasts. This restricts their contributions to theoretical advancements and prevents their practical application in real-world scenarios. As a result, the constraints and considerations necessary for integrating these models into broadcasts have not been explored. RQ3 addresses this gap by asking "How can ML models be integrated with the live coverage of esports?".

To explore this question, Chapter 4 presents a case study involving the application of an ML model during a Dota 2 tournament. This study provides ecological and empirically-derived insights into designing visualisation systems that allow ML models to be incorporated into esports broadcasts. By reflecting on the needs and requirements of various stakeholders, the study identifies critical design principles for integrating such systems, presented through a set of design considerations.

These principles include ensuring models cater to broadcasters' needs, such as branding and audience recognition, and adhering to understandability principles to make visualisations meaningful to viewers. The study also emphasises the importance of spatio-temporal information, which helps convey complex insights, as well as at-a-glance and contextual information, which together enhance usability and storytelling impact. Additionally, the study highlights the need for controllability and delivery mechanisms, ensuring that broadcasters can integrate tools seamlessly into existing infrastructure without disrupting their workflows or narrative strategies.

By adopting the strategies and design considerations proposed in Chapter 4, future ML visualisation systems can be more readily integrated into esports broadcasts. The case study also demonstrates strategies for understanding the needs of often-overlooked stakeholders, contributing to the development of practical solutions. In doing so, Chapter 4 addresses RQ3 and provides actionable insights for incorporating ML models into live esports broadcasts.

7.1.4 Research Question 4

The final major limitation identified in the literature pertains to how ML models are designed, presented, and evaluated. Existing approaches often limit the usability of models by failing to align their architecture and evaluation with the tasks they are intended to address. To address this issue, RQ4 asks "How can ML models enhance esport broadcast narratives?"

This question is explored in Chapter 5, which redefines the win prediction task to align with the needs of storytelling and narrative creation. The study begins by analysing how narratives are constructed within the esports ecosystem. A content analysis of commentary from professional esports broadcasters identifies key narrative features and the ways in which predictions are integrated into storytelling.

Based on these findings, Chapter 5 introduces the win condition system, an exploratory tool powered by two ML models specifically designed to enhance existing narratives. This system provides data-driven insights while supporting speculative narratives, aligning with the storytelling strategies already employed by broadcasters.

By investigating existing storytelling practices and integrating them into the design of the win condition system, Chapter 5 demonstrates how ML models can enhance esports broadcast narratives. Through this contribution, the chapter addresses RQ4 and highlights the potential for ML-driven tools to enrich storytelling in esports broadcasts.

7.1.5 Main Research Question

As described in Chapter 1, this thesis aims to address the main research question: "How can ML be applied to enhance esport broadcast narratives?" To achieve this, RQ1 was proposed to understand the current state of the art of the relevant academic fields. Through the findings identified in the SOTA, RQ2, RQ3, and RQ4 were identified as essential steps to address the main research question.

Consequently, a three-step approach was designed, supported by three distinct case studies detailed in this thesis. The win condition system described in Chapter 5 synthesises findings from all case studies. By integrating the CCR methodology proposed in Chapter 3 and the design considerations outlined in Chapter 4, the win condition system serves as a meaningful, usable, and effective tool for broadcast integration. This system builds upon and enhances the existing narratives provided by commentators, offering practical and speculative reflections on its integration and utilisation within storytelling contexts.

Therefore, this thesis addresses the main research question by exploring how ML models can be designed for seamless integration into live broadcasts. These models aim to enhance esports narratives as commentator tools, providing long-term usability even beyond the original game patch.

The contributions of this thesis are further consolidated into a systematic methodological framework, presented in Chapter 6. This framework offers practical guidance for incorporating the findings of this thesis into future ML work within the esports broadcast domain.

7.2 Limitations & Future Work

The work presented in this thesis seeks to assist ML practitioners in designing and developing models that positively impact esports broadcast narratives. However, esports is a vast domain encompassing a wide variety of game titles from numerous genres. While this work aims to be generalisable to the broader esports domain, the case studies focused on Dota 2. Consequently, the direct findings may be more readily transferable to research involving Dota 2 or similar games, such as League of Legends.

It is important to note that, while not essential, generalisability is ultimately a step towards maximizing the impact of a contribution. If a contribution can be generalised to a different use-case or environment, then it can impact a wider population than a solution that cannot be generalised. Of course, in many cases there exists a trade-off between generalisability and specialisation. Highly specialised solutions may be better at addressing their specific problem over a more generalisable solution. Therefore, developing non-generalisable solutions may be unavoidable.

However, it is also important to recognise that specialised solutions may also provide generalisable insights, even if the solution can not be entirely generalised. For example, CCR described in Chapter 3 is a highly specialised solution that may not be directly generalised to other titles. However, the insights that were obtained while generating CCR can be generalised to similar games. This depicts how generalisability my be viewed as a layered component, with some aspects of a solution being more generalisable than others.

Additionally, the resources and methodologies described in this thesis were developed to address the specific needs of the case studies and investigations in this thesis. While the general contributions address the relevant research questions, further development of, for example, alternative patch-aware representations could enable future work to better capture the complexities of evolving game environments. This would help mitigate the effects of impactful or disruptive game updates as described in Chapter 3.

Additionally, the investigation in Chapter 4 primarily considers the needs of tournament organisers, production teams, and broadcasters. While this provides valuable insights into often-overlooked aspects of esports broadcasting, a deeper exploration of other stakeholders could reveal additional considerations. For instance, the needs of camera operators (observers) and referees were not examined, yet their roles may provide important perspectives for the broader ecosystem.

Similarly, while Chapter 5 focuses on narrative impact, a more direct exploration of audience experiences could offer further insights. User studies evaluating audience reception to tools like win condition visualisations - particularly those involving audience-facing features - could provide data to help future work better align with viewer expectations and further enhance engagement.

Moreover, this thesis has strived to deliver transparency and reproducibility in relation to model selection and evaluation, as depicted in Chapters 3, 4 and 5. It is important to reflect on the work presented to improve on transparency and reproducibility. For example, the use of random seeds or additional visualisations (such as error or silhouette score graphs) could facilitate continued research to reproduce the findings more precisely. As continued research is done in the subject, it is important to strive to transparently report on the work to foster collaboration, and enabled more impactful contributions to the wider domain.

Another important limitation that should be mentioned is in relation to sample sizes. Particularly in relation to the analyses described in Chapters 4 and 5. While the analysis described was sufficient to address the posed research questions, larger sample sizes could provide further refinements of the findings of the thesis. This could have an impact of both generalisability and robustness of the impact in the domain.

Finally, the systematic framework presented in Chapter 6 condenses the findings of this thesis into practical guidelines. This framework is designed to guide future ML research in esports, enabling sustained, positive impacts on broadcast narratives. However, continued iterations of the framework, incorporating feedback from diverse stakeholders and case studies in other esports titles, could strengthen its applicability and relevance across the field.

7.3 Closing Remarks

This thesis explored the application of ML techniques in esports broadcasting, with a particular focus on their impact on commentary and narrative construction. Three major gaps were identified in the literature:

1. The challenge of sustaining model performance in dynamic environments due to frequent game patches.

2. The lack of exploration into integrating ML models into live broadcast coverage.

3. The limited consideration of narrative impact in the design and evaluation of ML models.

To address these limitations, this thesis presented a series of case studies that offer meaningful findings for the continued development of ML models in esports broadcasting. Chapter 5 demonstrated how the findings from Chapters 3 and 4, as well as the broader literature, can be incorporated into ML solutions to enhance narrative impact.

The contributions of this thesis are consolidated into a systematic methodological framework (Chapter 6) that provides practical guidance for designing, implementing, and evaluating ML solutions. This framework aims to enable the seamless integration of ML models into the esports broadcast ecosystem, enhancing narrative construction and audience engagement.

By bridging gaps in the literature and providing actionable methodologies, this thesis contributes to advancing the field of ML in esports broadcasting, offering a foundation for future research and practical applications.

Appendices

A

Term Definitions

A.1 General Esports and Gaming Terms

Esports

Instances when a Game (title) is played competitively.

Esport Broadcast

When Esports are recorded and streamed to audiences. This is typically done live, however Video on-demand (VOD) versions are also available.

NPC

Non-playing character or non-player controlled. Refers to in-game characters which the player can interact with, but not directly play as.

Game (title)

Refers to the specific game title. For example, Dota 2 is an esport game.

Game (series)

Refers to the set of Matches played in one series of match-ups between players. For example, the final of a tournament is one game that may be played in up to 5 matches in a Bo5 format.

Match

A match-up between one set of players. A Game (series) may be composed of one or more matches, where each match is played independently of the others. These are played until a conclusion is reached i.e. one team/player wins or a draw is reached. Note that a Game (title) may not allow for draws, for example, Dota 2 does not allows for draws.

Bo5

Best of 5. Refers to a specific format for a Game (series). In this format, up to 5 Matches are played, with one team winning the overall Game (series) when they win 3 of the 5 Matches in the series.

Bo3

Best of 3. Refers to a specific format for a Game (series). In this format, up to 3 Matches are played, with one team winning the overall Game (series) when they win 2 of the 3 Matches in the series.

Patch

Refers to the changes in the Game (title) released by game developers, typically with the intent to alter the balance of the game. Patches can release new content, remove old content or change existing content to different values, such as altering the movement speed of a character. The patch can also refer to the given instance of the game (i.e. the current values of the game which are available until the next game patch is released).

Meta

The community identified convention on how to play the game in the given Patch. This typically refers to what the community believes to be the most optimum way of playing the game, often used as a qualifier for some aspect of the game, such as saying a particular character is "not in the meta" to mean that the character is not considered to be viable in the current patch.

MOBA

Multiplayer Online Battle Arena. This is a popular genre of Esports Game (title) where two teams compete in a match to destroy the opponent base while defending their own. MOBAs typically includes NPC units (called Creeps in Dota 2) that will follow predetermined paths attacking any enemies that they encounter.

Dota 2

A popular Esports' Game (title) of the MOBA genre.

League of Legends

A popular Esports' Game (title) of the MOBA genre.

FPS

First Person Shooter. This is a popular genre of Esports Game (title), where the player plays from a first person perspective (i.e. seeing what the playing character would see, as opposed to seeing behind the playing character or with a top-down view of the playing field). FPSs typically involves guns, such as firearms, though some Game (title) may replace conventional guns with magical variations.

Valorant

A popular Esports' Game (title) of the FPS genre.

Overwatch 2

A popular Esports' Game (title) of the FPS genre.

CS

Counter Strike. A popular Esports' franchise of the FPS genre, with several popular titles.

CS:GO

Counter Strike Global Offence. A popular Esports' Game (title) of the CS franchise.

CS2

Counter Strike 2. A popular Esports' Game (title) of the CS franchise.

A.2 Esports Broadcasting Terms

Broadcaster

A general term referring to individuals who are involved in the broadcasting of esport. Often (but not limited to) used to describe the Audience facing roles, such as the Commentator.

Caster

Short form of broadcaster. Caster most commonly refers to the Commentator in particular, as opposed to the other Broadcaster roles.

Commentator

Used to describe the Broadcaster who provide commentary during the live broadcast of Esports.

Observer

Used to describe the Broadcaster that operates the virtual camera used to displays the in-game footage of the esport match.

Production team

The Broadcasters who are in charge of joining the multiple sources of media into the final broadcasting content that is delivered to Audiences.

Audience

The individuals who are consuming the Esports media content being broadcast.

Spectator

Another term used for the Audience.

Viewer

Another term used for the Audience.

Viewership

Broad term used for Audiences in aggregate. Typically refers to the number of viewers for a particular broadcast.

Spectatorship

Another term used for Viewership.

A.3 Dota 2 Specific Terms

Radiant

The green team in Dota 2. commonly, though not necessarily, referred to as Team 1.

Dire

The red team in Dota 2. commonly, though not necessarily, referred to as Team 2.

Creep

NPC characters which spawn periodically in the three Lanes moving from their team's base towards the enemy base. These units will attack any enemy units they encounter in their path, are typically week and is the primary source of gold and experience for players, particularly in the very early stages of a match.

Lane

The path which Creeps will follow. There are three lanes in Dota 2, top, mid and bot lanes, which refer to top middle and bottom lanes respectively.

Hero

A term used to refer to the playing characters within Dota 2.

Attribute

Refers to in-game modifiers that can alter a Hero or Ability. This includes Primary Attributes as well as more broad attributes, such as health or movement speed.

Primary Attributes

The main Attribute that most heavily impact a specific character. Primary attributes can be one of Strength (str), Agility (agi), Intelligence (int) or Universal (uni).

Gold

Resources that can be accumulated in each Match. This resource is used to purchase in-game items that can be used throughout the Match.

Exp

Experience. Refers to a resource gathered on each match that enables the characters to level up. This provides improvements to the overall Attributes of a Hero.

Ability

An in-game ability or skill that can be used by players. Each Hero has their own unique set of abilities. Abilities typically have an effect in the game, either by impacting the player character directly, or a target character (such as dealing damage or healing a unit).

Cast

Refers to utilising an Ability.

Pick

The act of selecting a playing character to be played in a Match. Pick can also refer to the character selection itself such as in “this is a good pick”.

Ban

Used to refer to when a character is marked as un-pickable. This means that neither team is able to Pick this Hero to be played within a particular Match.

Draft

Sometimes referred to as the Draft phase or the Pick/Ban phase. The character selection phase that takes place at the start of every match.

Vision

Refers to area of the playing field that the Dota 2 player has access to the state off. Areas in Fog or FoW where the player does not have access to the state off are considered not in vision.

Fog

Term used to describe the areas of the playing field that a team does not have access to the game state off. This can be more broadly referred to as areas without Vision.

FoW

Fog of War. Another term used for Fog.

Wards

In-game items that provides temporary Vision to a team.

Roshan

A large in-game unit that is hostile to all other units. Successfully defeating Roshan awards the entire team with a large reward of Gold and Exp as well as unique items that cannot be gained through any other means. This makes Roshan a strategically important aspect of Dota 2.

B

Search Strategy and Exclusion Criteria

B.1 Education & Serious Games

- Gamification
- Education
- Educational Games
- Serious Games
- E-learning

B.2 Representation & Wellbeing

- Female
- Gender
- Male
- Accessibility
- Autism
- Economic and Social Effects

- Health
- Mental Health
- Physical Health

B.3 Monetisation

- Marketing
- Non-fungible Tokens
- NFTs
- Advertising
- Blockchain
- Commerce
- Gambling
- Gamblification

C

Unused Generated Graphics

Figure C.1: The generated WARDS graphic for the "Roshan Rumble" Grand Final Game 1

D

What Are You Looking At? Team Fight Prediction Through Player Camera

Abstract

Esport is a large and still growing industry with vast audiences. Multiplayer Online Battle Arenas (MOBAs), a sub-genre of esports, possess a very complex environment, which often leads to experts missing important coverage while broadcasting live competitions. One common game event that holds significant importance for broadcasting is referred to as a team fight engagement. Professional player's own knowledge and understanding of the game may provide a solution to this problem. This paper suggests a model that predicts and detects ongoing team fights in a live scenario. This approach outlines a novel technique of deriving representations of a complex game environment by relying on player knowledge. This is done by analysing the positions of the in-game characters and their associated cameras, utilising this data to train a neural network. The proposed model is able to both assist in the production of live esport coverage as well as provide a live, expert-derived, analysis of the game without the need of relying on outside sources.

D.1 Introduction

The esport industry has been rapidly growing and generating vast interest, both through large audiences [169] as well as economically [43]. For this reason, academic interest has also developed to meet the demands of this growing industry [137]. One popular branch of esports is the field of Multiplayer Online Battle Arena (MOBA), including titles such as Dota 2 [36] and League of Legends [109]. Esport tournaments for these titles are commonly organised with community-funded prize pools (either partly or entirely) that reach over 34 million dollars - as of the Dota 2 International 2019 [28]. Due to this active and engaged community, broadcast entities are often looking for different ways to enhance the audience experience to maintain and increase their viewership numbers [82].

Tournament organisers play a fundamental role in directing audiences focus during the match through their streaming and similar delivery platforms. If engagements, such as team fights are not timely detected, important parts of the game can be missed by their audience. Due to the overwhelming accessibility of esport, and indeed regular sports, through several streaming and broadcasting options [63], broadcasters are driven to provide as complete and comprehensive coverage of important game events as possible. However, esports are characterised by their fast pace which can present a challenge even for experts [10] in the field. This can be observed during quick events, such as team fights that typically start abruptly. Thus, a key challenge identified by researchers and the industry is to losslessly transmit the highly complex flow of information from the game to the audience.

This paper provides a novel methodology in the identification and prediction of game events that are prevalent in the broadcast of MOBAs. Specifically, ways to identify and predict team fights, which are in-game events where both teams engage in a confrontation using their abilities and resources in an attempt to gain the advantage in the match while penalising the opposing team. As identified in the literature, these events are established as some of the most enthralling moments for audiences [82], as they are typically very active and can often change the course of the game [70]. However, despite their importance, team fights can often be entirely or partly missed by audiences or tournament broadcasters due to the complexity of the game and the fast-paced tendencies of engagements.

By utilising player input, the work presented in this paper attempts to predict team fight engagements. The suggested model is designed to utilise players' knowledge and understanding of the game to address this problem and predict team fights shortly prior to their start. This is achieved by studying the camera position of players at any given time to train a neural network to identify patterns that can be used to predict and detect ongoing team fights.

Camera locations have been selected as they represent the information that a player can visually derive from the environment at any given time. As this is closely related to player vision [22], meaningful information about the player's intent could be inferred. Player vision is particularly relevant for understanding player's informed decision making. Patterns in player behaviour could be observed and used to make game events predictions. By studying the effects of player vision and how it may connect to team fights, this paper proposes a model that attempts to understand this complex environment by relying on the player's understanding of the game.

This paper shows that, despite the complexity of MOBA games, it is possible to successfully predict player engagements such as team fights, by only considering the player character and camera position. The model is designed to interpret player intent which, as observed in the literature, can be driven by vision and game-state data. The hypothesis proposed by this paper is that when a team is planning to start a fight, their cameras should converge together before their characters move into the position[1].

This paper focuses on Dota 2 as a domain in order to study this hypothesis. The game description, with an emphasis on core game mechanics and relevant game-specific terminology is provided in Section D.2. Section D.3 reviews the literature about event prediction in the esport domain. A description of the employed methodology, including data acquisition and the training process, is presented in Section D.4. Section D.5 displays the outline of the performance, for test and train data sets, as well as the ecosystem data, representing a real use case scenario. The analysis of the obtained results, with a comparison to their respective in-game events, is shown in Section D.6. Section D.7 revisits the proposed hypothesis and evaluates it in regards to the observed results. Finally, Section D.8 considers potential avenues of research and proposes several paths of extending this methodology for future projects.

[1]In MOBAs players can move their cameras independently of their characters

D.2 Dota 2

Dota 2 is a top-down perspective game with a diagonally symmetric map. In this game, two teams (Radiant and Dire) of five players each attempt to attack and destroy the opponent's base. Players choose from a wide number of characters (heroes), each with their own unique set of abilities and skills, allowing for different roles to be taken.

The two bases are connected through three lanes (Top, Middle and Bottom), each containing buildings (towers) that attack the opponents, dealing a large amount of damage when in close proximity. The map, showing the lanes and both bases, can be seen in Figure D.1.

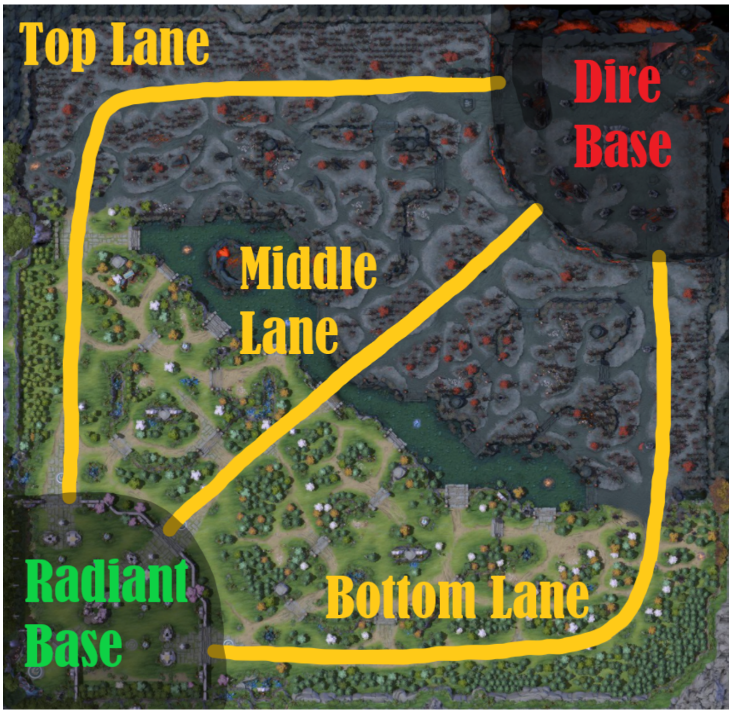


Figure D.1: Dota 2 map with marked lanes and bases.

In order to reach the enemy's base, teams must destroy each of the towers in order, which requires a large amount of in-game resources and teamwork to achieve. For this reason, teams often engage in confrontations with each other, which are referred to as engagements. Large engagements are further categorised as team fights, which provide opportunities for inflicting a severe detriment to your opponent while providing the winning team with a large number of in-game resources. The game is won once the main building in the opponent's base is destroyed. Due to the large commitment required to achieve this, teams often engage in large scale team fights to penalise the opponent and progress towards that goal.

D.3 Related Work

Through the esport literature, many attempts to identify and predict different events and aspects within MOBAs have been explored. A common focus in the literature is on the game outcome prediction, where authors model the game environment to discern the game state in order to predict the winner of the match. This is done throughout several stages of the game. Some authors have suggested models that predict the game prior to match start [77], while others have used game state data to predict the outcome during live matches [59]. The authors outline the potential of utilising machine learning techniques, such as logistic regression and random forests, to achieve varying degrees of accuracy. Their results also highlight the difference in performance that can be achieved depending on the time period of the prediction, where shorter intervals typically achieve higher accuracy and reliability compared to the ones looking further into the future.

Furthermore, predicting game events is not limited to their outcome. The use of game-state data is often employed in making predictions. Some authors have utilised this technique [70] to determine the danger level encountered by a player at any given time, performing a death prediction. This was achieved by employing a deep neural network with a vast amount of labelled data to train a model. The resulting network is able to make short term predictions on how likely an in-game character is to die within the next five seconds, displaying the capability of making data-derived predictions that are reliable.

Other authors have attempted to use data to identify aspects of the game which are not easily ascertained, such as player roles [29]. By utilising clustering techniques, and historic performance from professional players, the authors were able to identify the role that an individual takes in the team. This allows for a more in-depth understanding of the game state. This study also showcases the potential of utilising player decision making to detect and classify performance. This is particularly reliable for professional esport tournament data, where player performance is of a high degree of proficiency due to the competitive nature of those tournaments [10].

Similarly, some authors have used game state data to define events such as engagements, also known as encounters [146]. By analysing the capability of each character, the authors were able to determine a minimum distance between characters that allows for an encounter to happen. By observing the utilisation of the character's active abilities as well as the transfer of damage or healing, the authors formally defined an encounter. Team fights - which is the focus of this paper - is a type of encounter that was later labelled by the game developers through the OpenDota platform [2] within Dota 2 games. A formal description of what the game developers define as a team fight is not

[2] `https://www.opendota.com/`

available. However, it can be inferred that it involves an encounter in which more than two characters die within a set amount of time. Those labels can be retrieved through the OpenDota platform although, as described in Section D.6, those labels have inconsistent standards for start and end time.

Lastly, some authors have identified the importance of players acquiring information [22] - which is referred to as *player vision*. By studying the amount of vision available to the team, due to the imperfect information aspects of the game, the authors were able to observe a direct link between vision and in-game advantage. This highlights the importance of information available to a player and how it impacts their strategy and decision making.

As noted in the literature, modelling the game state is difficult, but it can be achieved through several means of interpreting and acquiring information about the game. For this reason, this paper proposes the utilisation of player knowledge to direct a model to simplify the complex game state and make predictions. Player camera positions can be used to infer player vision, which has been shown to be connected to decision making [22].

D.4 Methodology

D.4.1 Data

In this study, a total of *1,457* professional Dota 2 matches were gathered from the game patch *7.27* using the OpenDota API. This data set consisted of team fight labels as well as match replay files. Using the Clarity Analyzer library [155] - a free Java library for reading Dota 2 replay files - camera and in-game character position data were extracted from those files at one-second intervals. This data was then encoded into four heatmaps, one for each of the following:

- Radiant players position.
- Radiant cameras position.
- Dire players position.
- Dire cameras position.

The heatmaps were generated by aggregating the data from five consecutive snapshots, matching five seconds of game time. A single snapshot contains information about each player's camera position and their in-game character position. The five-second interval was chosen for the purpose of capturing enough information about camera position while reducing the impact of the short term flicking of the camera, moving to a random location due to various events, which players occasionally perform in the game. Combining a series of snapshots provides an indication of where the heroes in a single team were, as well as their associated player vision - i.e. what they were looking at.

The generated heatmaps split the Dota 2 map into cells. The original range of the coordinates spanned between $[-8472, 9198]$ on the x-axis and $[-8579, 8845]$ on the y-axis. In order to balance between maintaining the precision of the input data, with keeping the input to the neural network as small as possible the dimensions of the heatmap was set to 50×50. Each cell on the heatmap representing 300×350 pixels of the Dota 2 map. The 4 heatmaps were encoded in one of the 4 image channels. The upper limit value of 255 corresponds to the maximum amount of convergence of data (i.e. all players or camera positions converged to the same pixel area in the world map for the entire 5-second interval), while a lower limit value of 0 representing an empty area.

This information makes the input for the neural network, and it is used to predict if a team fight is going to happen. In order to ensure that the model was predicting future team fights instead of only detecting currently ongoing ones, an additional 3 seconds were added to the prediction time labels. The selected delay would give sufficient time to the broadcaster to focus their attention on the incoming event. Those 3 seconds were not used as a part of the heatmap. To reflect what is expected of a prediction using live data, if the snapshots were taken for the in-game time of 1-5 seconds, then the prediction is registered at the start of second 9 without using the snapshots for seconds 6-8. The information stored in each heatmap, delay and prediction time can be seen in D.2.

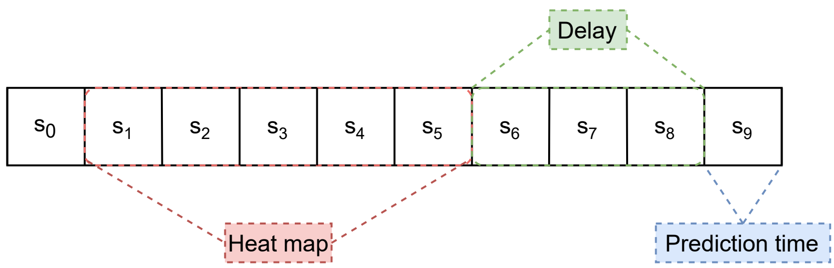


Figure D.2: Heat map generation. S_i denotes information gathered in a single snapshot.

For each of the 1457 professional matches, team fight start and end times are collected timestamps data through the OpenDota platform. The data was processed into a form matching the previously acquired snapshots. For each second of the match, a label was created. That label represented whether there was a team fight at that particular time or not, with the addition of the three seconds delay. This suggests that the training labels values were

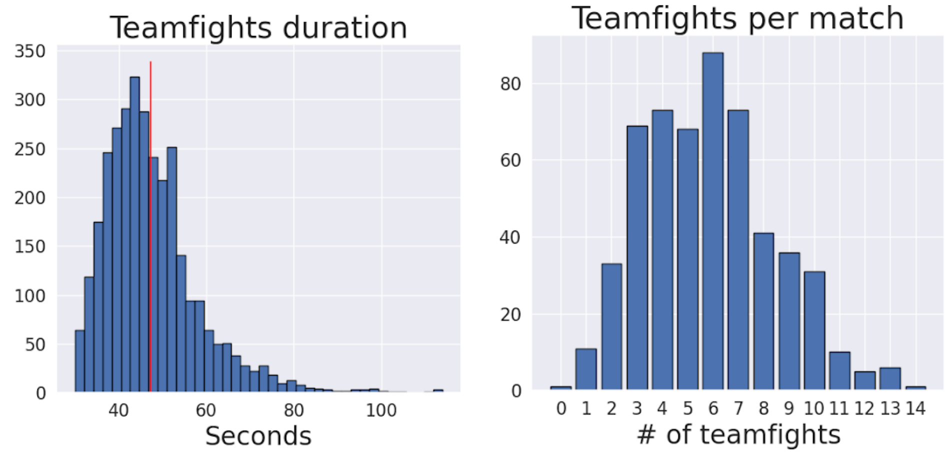


Figure D.3: Team fight duration and the number of team fights per game.

set to true for any snapshot that contained a team fight. It is important to note that those labels contained inconsistencies with the formal engagement definition as defined by the literature, as described in Section D.6.

In a typical Dota 2 game, on average, team fights account for only 12% of the total game length. This causes a notable disparity in the amount of team and non-team fight labels. Figure D.3 displays the aggregated data for the labels retrieved from OpenDota. In order to address the much lower frequency of team fights, the data set was artificially balanced. This was done by selecting all of the labels for each individual team fight. An equal amount of non-team fight data was also included per team fight, half of this was retrieved prior to the team fight start, and the other half post the conclusion of the fight. As Figure D.3 outlines, on average there are six team fights per game with each of them lasting about 48 seconds in duration. After applying this balancing filter, in total there were approximately $800,000$ data points of balanced data.

Furthermore, symmetries were explored to augment the training data.

The training data was inverted to generate a greater variance in the heatmaps. Due to the Dota 2 map being mostly symmetrical, inverting the X, Y, or both axes of the heatmap provided additional variety in the training data. This step was done to reduce over-fitting.

D.4.2 Training procedure

The data was split $80\% - 20\%$ for training and testing purposes. Each data point contained all four heatmaps, and a label, marking if there was a team fight three seconds into the future. Because of the nature of the heatmaps used as inputs, a two-part network was employed to allow for feature extraction. The first part consists of a convolutional network, and the second part is represented by a deep sequential network. Different architectures were trained and their performance compared. These variations included:

- Different non-linearities.
- Adjusting the parameters for the convolutional layers.
- Batch normalization.
- Adaptive max pooling.
- Different amount of linear layers.
- Multiple dropout rates.

Multiple changes have been enacted. The sequential module of the network was initially expanded from 1 to 4 layers, containing [512, 256, 128, 1] neurons per layer, and subsequently increased to 8 containing [2048, 1024, 768, 512, 256, 128, 64, 1] neurons in each layer. The dropout rate was also

Table D.1: Explored network parameters.

Non-Linearity	ReLU, ReLU + inplace, Leaky ReLU
Number of filters	4, 32, 64
Batch normalization	No normalization, 4, 32, 64
Max pooling	[2, 2], [5, 5], Adaptive Max Pool
Number of linear layers	1, 4, 8
Dropout rate	0.0, 0.1, 0.2, 0.4, 0.5

included to treat the over-fitting issue. Dropout rates of 0.1, 0.2, 0.4 and 0.5 after each layer have been tested. Leaky ReLU non-linearity has also been introduced instead of the standard ReLU with in-place enabled. Different values for Batch normalization ranging from 4 to 64 have been tested, and adaptive max pooling introduced although it was not adopted for the final model. The summary of the different parameters used for the final model is presented in Table D.1.

All of the architectures showed similar performance, with the biggest difference manifesting with the increase in the number of linear layers. The final architecture employed two convolutional layers, connected by Leaky ReLU activation functions [96]. Parameters for the number of filters in the convolutional layers was set to 32 and 64 respectively, kernel size to 3, and stride to 1. The network architecture can be seen in Figure D.4. Batch normalization was done after each layer. Max pooling with kernel size 2, stride value of 2, was added after both layers as well. Lastly, the output of the convolutional module was flattened and fed into a series of fully connected linear layers.

The sequential part of the network consists of 8 linear layers, including the final output layer, with Leaky ReLU as the selected activation function, all connected by dropout layers in between. The final layer was the classification layer with one output, representing the network prediction. During the training procedure, the network achieved the best performance with a heavy dropout rate of 0.5. The full network architecture, including the sizes for each of the layers, can be seen in Figure D.4. The model was trained with the Adam optimizer [76]. The initial learning rate was set to $1e^{-5}$ with the weight decay of $1e^{-4}$. Binary cross-entropy loss was selected for the training process.

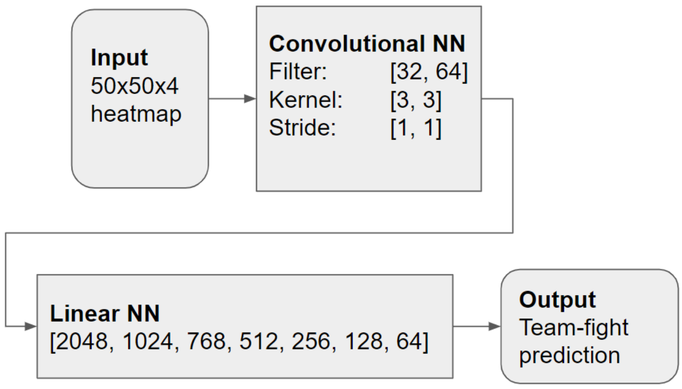


Figure D.4: Final network architecture.

D.5 Results

In this section, the train and validation accuracy is used to report on the model's performance. The result of the training process was a model that reached the accuracy of 84% on both train and validation set. Figure D.5 displays the entire training performance, including training and validation for both accuracy and loss. However, this accuracy was achieved on an artificially balanced and augmented data set. For this reason, additional testing was needed.

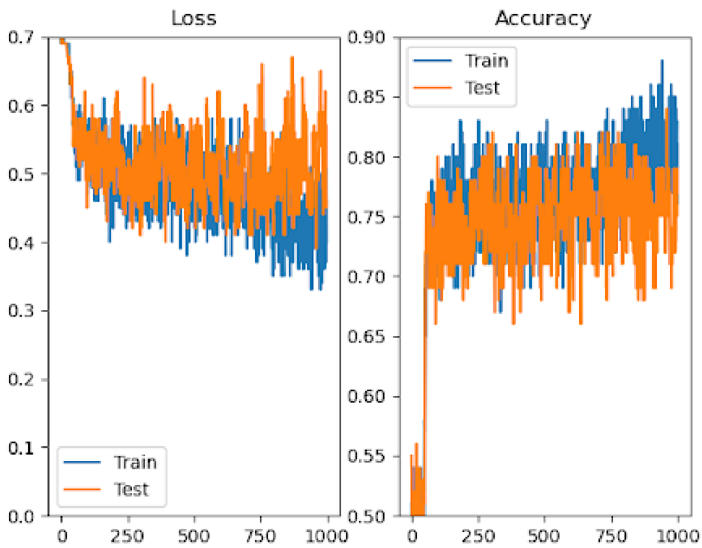


Figure D.5: Training loss and accuracy

In order to test the model further, a previously unseen match is used to analyse the performance in an actual use case environment. This method best simulates the performance during its use in the ecosystem, reproducing the behaviour of the model as in a real Dota 2 game.

Figure D.6 reports an example of the results obtained for a concrete match *ID:5492227432*[3]. In this Figure, the 'Time to next team fight' graph displays the amount of time until the next labelled team fight. The following graphs indicate the desired (in blue) and achieved (in orange) behaviours of the prediction model. The Blue line represents obtained labels, while the orange represents the model's outputs, with the corresponding confidence thresholds. If the output has reached or surpassed the confidence threshold, the orange line is set to high. Otherwise, it is set to low. A total match

[3]A replay of the match can be obtained the through OpenDota website at: `https://www.opendota.com/matches/5492227432`

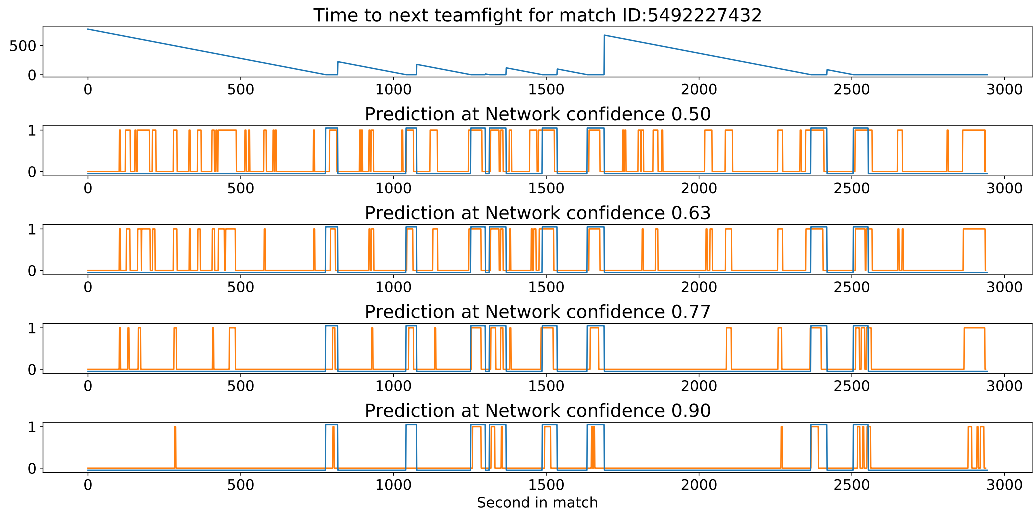


Figure D.6: Testing on an entire match with different confidence thresholds.

between the orange and the blue line would indicate a perfect prediction model. For demonstration purposes, four different levels of the prediction confidence threshold of the network classifier are used, ranging from 0.5 to 0.9.

Additionally, Figure D.7 summarises the classification performance of the model using confusion matrices for the same confidence thresholds as displayed in Figure D.6. Using the data from the confusion matrices, f1 scores were calculated. Base f1 score, for confidence threshold of 0.50, was 0.49, at 0.63 it increased to 0.51. The maximum f1 score of 0.55 was reached at 0.77, and it dropped to 0.50 at the threshold of 0.90.

D.6 Discussion

In this study, a model, and more importantly, a new methodology for predicting and detecting ongoing team fights are proposed. Using only in-game

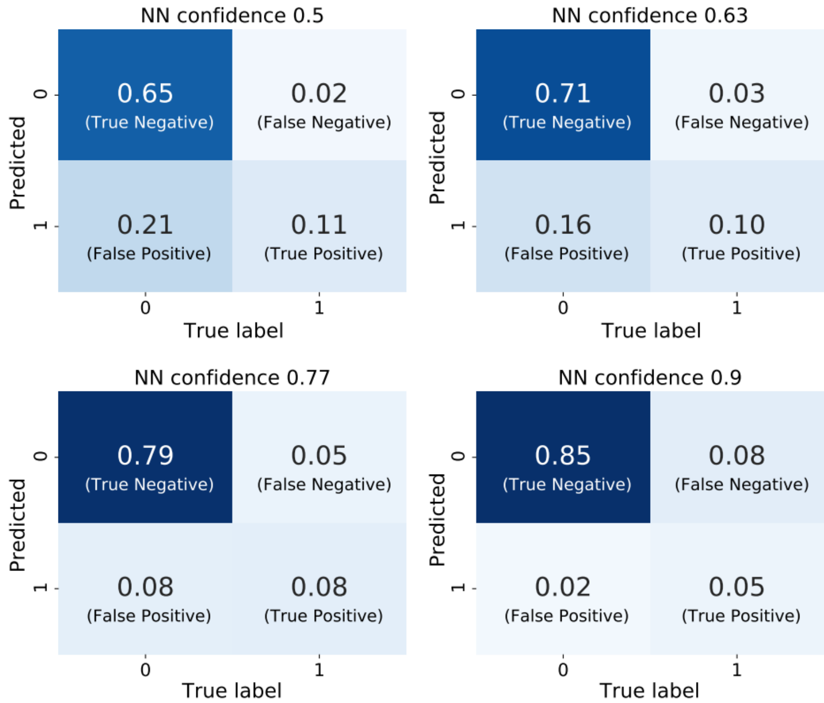


Figure D.7: Confusion matrices for different confidence thresholds.

character and camera positions the proposed model was able to reach similar levels of accuracy as encountered in the literature for other similar event predictions [22, 70]. However, it is evident that the high amount of false-positives has impacted the performance of the neural network, as they are the main reason for the f1 scores not reaching similar values.

A parameter search done on the confidence threshold of the model offers a way of reducing the number of false-positives. Modifying the threshold amount from 0.5 to a different value, showed improvement in the performance of the model. Despite visible improvement, the obtained f1 score remained lower than observed accuracy. This indicated a possible issue with the labels. As a formal definition of the labelled team fights is not available, manual evaluation was required to determine the cause for the low f1 scores.

Table D.2: False-positive predictions and the corresponding in game events.

Game Time	Description
6:20	Two engagements at the same time at last for 15 seconds. 6 characters present in the engagement in the top lane, 4 characters on the bottom one. Two heroes die, one in each engagement.
20:35	A small engagement that starts a few seconds before (20:28) turns into a bigger one after several characters join the fight. 5 heroes die.
36:18	All 10 characters present. Relatively quick engagement on the Radiant side of the map, two Dire characters die.
46:29	Last engagement in the game. Starts with a chase, and ends with the Dire team winning the game. The duration of the fight was 58 seconds, all of the characters on the Radiant side die.

Upon closer examination of the events happening in a match, it was discovered that the large majority of false-positive predictions (91.4%) correspond to an engagement in the game as defined in the literature [146]. The analysis of the positive prediction was done for the match presented in Figure D.6 and a sample of the analysis and their corresponding game events is presented in Table D.2. All of the analysis was done with the confidence threshold of 0.77, with the exact criteria for an engagement taken from the existing literature.

The full description of all positive predictions for the entire match and their corresponding game events can be seen in Appendix D.9. Most of the cases had all of the preconditions needed to be classified as an engagement. The only differentiating factor that can be inferred from observing the game events was that the number of characters that died during the engagement did not reach the estimated threshold of three and thus the event did not get labelled as a team fight. Additionally, an edge case could be observed where fights that meet the presumed team fight criteria were not labelled as such in the OpenDota data due to it happening just before the game ends. Furthermore, it can be noted that in many instances the OpenDota labels were set to true either too early (i.e. several seconds prior to the engagement start) or too late (i.e. several seconds into the engagement, even when heroes have already been killed). This inconsistency with the labels could be a major factor that negatively impacted the performance of the model. In many of these cases, the proposed model was able to provide more precise labelling for the start and end of the conflict based on the literature definition of engagements, despite being trained on these inconsistent labels. This is further evidence of the potential of the methodology which uses and benefits from player expertise.

Due to the fact that the input data consists only of in-game character and camera positions, the model does not receive any information about character state, and is not able to estimate how many players would die in an encounter, and thus make a distinction between an encounter and a team fight. However, the small quantity of false-negative predictions suggests that even without detecting the difference between team fights and encounters, using player and camera position can be a reliable way of predicting and detecting both team fights and similar encounters. This supports the assumption that the camera position reflects aspects of player knowledge as well as their strategy. The proposed network can detect patterns in this data and make relevant predictions without the need to explicitly model this complex process of human decision making.

D.7 Conclusion

This paper address the predictability of a team fight using a small number of input parameters. In-game character position, in addition to their player cameras, are used to make observations and conclusions to predict and detect those encounters. Using this data to train a two-part neural network, the model was able to achieve 84% accuracy. However, despite the high achieved accuracy, when evaluated on an entire match, the model possessed lower f1 scores. Through further investigation, inconsistencies and potential issues with existing industry labels for team fights were identified. These inconsistencies influenced both the training and the evaluation process. As the obtained predictions outperform the industry labels, they could be used in conjunction with other existing formal definitions in the literature to provide a better definition for the concept of a team fight, allowing for more meaningful and reliable labels for future work in the area.

Moreover, despite the inaccuracies encountered with labelling, it is clear that the model possesses predictive capabilities. This indicates that the employed methodology, which takes advantage of players expertise to derive conclusions is a reliable way of extracting meaningful information. Due to the high complexity of the environment, models that attempt to interpret the game entirely from raw data may be replicating already available human understanding. In some cases, such as player planned encounters, this understanding could be derived from available or otherwise existing data sources.

This paper presents a novel approach for extracting player intent and utilising this data. No studies could be found - at present - that utilise player camera to replicate or derive player intent and decision making. The obtained results provide a clear indication of the potential of this technique.

The model described in this paper could be used in the industry to enhance game coverage for audiences. One promising example is implementing this technique in existing apps that are used in the broadcast of such titles. Another possible application is to make the results available for camera operators in the production of esport events, to assist in directing the focus of the coverage.

D.8 Future Work

The study described in this paper reveals several avenues of research. Considering other game information, in addition to the existing camera and character position, could be relevant in differentiating between encounters and team fights. In a game such as Dota 2, there are many factors effecting the likelihood of a specific team winning an engagement and the number of deaths that are going to happen in a team fight. Character levels, character roles, gold and experience difference between teams, and game time may all have an impact on the decision of whether a team commits to a team fight or not. Adding the information about the character state could allow the neural network to predict deaths, which could make it suitable for distinguishing between specific types of engagements.

Furthermore, the techniques employed in this paper could be used in different models to improve their performance by relying on players expertise to reduce the complexity of the problem space. Detecting and predicting when a player is planning an engagement could be used as a factor for determining the potential winner of an encounter, and indeed, the outcome of the game. This could serve to aid in the win prediction domain.

Finally, some games may include player-made markers, i.e. pings, to aid rapid communication between teammates. In these cases, a similar approach, as described in this paper, could be used to derive other forms of meaningful information through player intent. Looking into the correlation between pings and team fights could provide additional insight into player's expert knowledge. Combined with the camera positions a model could attain more in-depth modelling of player behaviour and take greater advantage of their expertise.

D.9 Complete match analysis

Full analysis of the positive predictions is presented in Table D.3 below. All of the predictions are calculated with a confidence threshold of 0.77 on the match presented in Figure D.6. The criteria for the engagement was taken from [146], and is as follows:

1. At least three heroes in range of each other.
2. Heroes of both teams present.
3. Transaction of damage from one team to the other.

In Table D.3, Event provides a numbered id for each event in chronological order. Game time depicts the in-game time in which the event started. Event descriptions provides a summary of what happened in the event. Network provides a binary output of the prediction of the network (i.e. 1=team-fight predicted, 0 otherwise). Lastly, Label reflects what the data retrieved from OpenDota displays.

Table D.3: Analysis of the positive predictions

Event	Game Time	Event Description	Network	Label
1	0:17	A small engagement on the top lane, Enchantress fires a couple of shots at 2 Dire characters going through the lane.	1	0
2	0:41	Radiant heroes engage in 3v2 on the top lane, Morphling gets caught out of position. First kill of the game.	1	0
3	1:16	Top lane, 3v2 fight, lasts for 8 seconds. Nobody dies.	1	0
4	1:33	Top lane, 1v2, Enchantress doing some damage to the enemy heroes. One character very close to dying.	1	0
5	3:09	Engagement starts on top, 4 seconds later another one starts on bottom. Top engagement ends with one death on the Dire side, no casualties on the bottom.	1	0

6	5:23	Phoenix and Morphling start trading shots at 5:16. Additional characters come to aid both sides at 5:23, Morphling starts running away but gets chased down.	1	0
7	5:51	A small engagement on top, dire characters teleports in and gets engaged on. Engagement finishes in a couple of seconds. No deaths.	1	0
8	6:20	Two engagements at the same time at last for 15 seconds. 6 characters present in the engagement in the top lane, 4 characters on the bottom one. Two heroes die, one in each engagement.	1	0
9	11:39	A big team fight starts. At first, only 3 characters present, others joined during the duration of the fight. The fight lasts for 30 seconds with 3 characters dying.	1	1
10	13:57	No engagement at this moment. There are several heroes close by, but nothing happens.	1	0

11	14:04	A small engagement on top, 2v1. One character dies, short engagement, only lasts for 4 seconds.	1	0
12	15:56	A team fight happening near the top lane. Starts with a small engagement, heroes from both sides join the fight. The fight ends in 20 seconds with 4 deaths.	1	1
13	17:30	Engagement near the bottom lane on the Dire side of the map, Morphling (D) was caught out of position but runs away quickly.	1	0
14	19:24	The biggest team fight of the game. All characters from both sides are present, the fight lasts for 35 seconds, three characters die.	1	1
15	20:35	A small engagement that starts a few seconds before (20:28) turns into a bigger one after several characters join the fight. 5 heroes die.	1	1
16	21:37	A bit mistimed prediction. Just as an engagement finished.	1	0

17	23:19	The radiant team goes into the Dire area near their base, Dire engages, 4 Radiant characters and 1 Dire character dies. The fight lasts for 40 seconds.	1	1
18	26:01	A team fight in the middle of the map. Engagement starts slowly but ends with 7 characters losing their lives. Engagement lasts for 50 seconds.	1	1
19	29:42	Engagement starts at 29:32 and ends at 29:40. The prediction was two seconds late.	1	0
20	32:36	Another small engagement happens before the prediction, the prediction happens two seconds after the engagement ends.	1	0
21	33:28	Engagement near the Dire base. 9 out of 10 characters are present, lasts for 25 seconds.	1	0
22	36:18	All 10 characters present. Relatively quick engagement on the Radiant side of the map, two Dire characters die.	1	0

23	37:51	Probably the game-deciding team fight. Both teams are ready, every character is in the position, and prepared for the fight. The engagement starts at 37:58 and lasts until 38:38. All of the Radiant characters die.	1	1
24	40:33	Team fight in front of the Radiant base. Long engagement, all of the characters present, multiple characters die, revive and come back to the fight. Radiant successfully defend the base.	1	1
25	46:29	Last engagement in the game. Starts with a chase, and ends with the Dire team winning the game. The duration of the fight was 58 seconds, all of the characters on the Radiant side die.	1	0

E

From Passive Viewer to Active Fan: Towards the Design and Large-Scale Evaluation of Interactive Audience Experiences in Esports and Beyond

Abstract

Esports - competitive video games watched by online audiences - are the fastest growing form of mainstream entertainment. Esports coverage is predominantly delivered via online video streaming platforms which include interactive elements. However, there is limited understanding of how audiences engage with such interactive content. This paper presents a large-scale case study of an interactive data-driven streaming extension developed for *Dota 2*, reaching over 300,000 people during the *DreamLeague Season 15 DPC Western Europe* tournament. The extension provides interactive live statistics, analysis and highlights reels of ongoing matches. This paper presents an analysis of audience telemetry collected over the course of the four week tournament, introducing a novel approach to analysing usage data delivered seamlessly in conjunction to a linear broadcast feed. The work presented advances our general understanding of the evolving consumption patterns in

esports, and leverages esports as a lens to understand future challenges and opportunities in interactive viewing across sports and entertainment.

E.1 Introduction and background

Esports are video games played competitively, and attracting over half a billion viewers annually [112, 52, 136, 111, 164, 54, 107, 89, 118, 183]. Esports games come in a variety of genres, including first-person shooters (e.g. *CS:GO*[1], *Overwatch*[2]) and battle arena games (e.g. *Dota 2*[3], *League of Legends*[4]).

The viewing experience in esports is similar to linear coverage in traditional sports, blending live footage of gameplay with audio commentary, and providing pre- and post-game panel shows. Coverage is thus similar to the concept of sports broadcasts, where every viewer receives the same experience. However, the platforms of delivery and patterns of consumption differ from traditional sports [87, 10, 82, 21, 165]. Esports are predominantly broadcast online via streaming platforms such as *Twitch*[5], YouTube Live[6] and *Facebook Gaming*[7]. These platforms providing rich social interaction between viewers [165, 10]. Esport audiences are highly active on the platform's built-in live chat and social media, and increasingly demand more active ways of engaging with live events [82, 21, 16, 157]. Responding to evolving audience needs, streaming platforms have recently introduced the capability to

[1] `https://store.steampowered.com/app/730/CounterStrike_Global_Offensive`
[2] `https://playoverwatch.com/en-us/about`
[3] `https://www.dota2.com/play/`
[4] `https://na.leagueoflegends.com/en/game-info`
[5] `https://www.twitch.tv/`
[6] `https://www.youtube.com/`
[7] `https://www.facebook.com/gaming/`

enrich the linear video stream with interactive elements, such as informative overlays, mini-games or audience voting [108, 31, 48]. This enabled broadcasters to develop interactive interfaces [16] that allow audiences to choose when and how information is displayed to them [21]. Subsequently, previous research has explored the design of interactive spectator experiences in esports, highlighting a range of benefits brought to viewers through interactive elements alongside linear video streams, including the ability to personalise one's viewing experience, discovering additional insights, and facilitating enhanced emotional engagement with the live event [87, 10, 82, 21, 165]. However, despite the identified benefits, there is still little understanding and evidence of how the use of interactive elements practically entwine with existing passive consumption patterns, nor suitable definitions and methodology for analysing active viewing behaviour.

This paper presents the first large-scale naturalistic study of interactive viewing in esports, drawing on detailed telemetry data collected from over *300,000* unique viewers in the context of a large international esports tournament. Collaborating with one of the world's largest esports company (ESL), we designed and developed the *The Dota 2 Twitch Extension*, an interactive twitch extension for *Dota 2*, one of the top three most popular esports worldwide [159, 158]. This extension gives Twitch viewers the ability to display on-demand game statistics, highlight recaps, match (game) status and more.

The contribution in this paper is two-fold. After a review of related work, this paper introduces the *The Dota 2 Twitch Extension*, describing the 2 year iterative design process and the resulting experience for viewers, aiming to bring out design patters, UI challenges and means of integrating with linear video stream. Secondly, this paper presents the results of a longitudinal large-scale audience study of how the extension was used in a real tournament environment, providing new methodology for the analysis of interactive spectator experiences, validating various aspects of the design, and providing a detailed characterisation of observed interaction patterns. Finally, a discussion on the findings, and design implications are outlined. This work seeks to inform the design of interactive viewing more broadly, establishing foundations for design of interactive viewing experiences, and discuss the wider application of the findings to traditional sports and entertainment more broadly.

E.2 Related work

Esports is a rapidly growing field and the associated academic and industrial research domain is diverse, quickly evolving but also somewhat fragmented in the diversity of research available [136, 10, 82]. There is also a related cross-disciplinary body of research surrounding traditional sports and broadcasting, consumer research and behavioural analytics [130, 79, 44, 33, 82]. The case study presented here operates across a number of these lines of investigation, chiefly the interplay between *data, audiences* and *experiences*. Therefore, this section explores a range of previous work available in the literature in a range of domains. Firstly a review of previous work exploring audience needs for esports (and sports) is outlined. Secondly, this paper summarizes some of the existing work in providing advances to data-driven esport audience experiences.

E.2.1 Understanding Esport Audience Needs

Understanding the needs of audiences and their varied knowledge and interests is crucial to be able to deliver engaging experiences. In traditional sports, data-driven content has become more common in recent years, with visualisations of player position and ball tracking data across multiple sports, increasingly relying on GPS-based tracking data and 3D-visualisations [32, 61, 196]. For esports, there is an even greater need to break down the complexity of the gameplay and facilitate insights to the audiences [52, 87]. Indeed, consumer needs research has found that understanding the in-game actions, the skill and decisions of the cyberathletes, is a key motivator for watching because many viewers are also players [52, 87, 184, 150]. This emphasizes the potential for data-driven storytelling to provide engagement points for audiences [52, 10, 82].

Within culture, the esports literature has outlined a clear need for continued research [160, 171]. Previous work within this field has outlined several aspects of consumers that are particularly relevant for esport audiences. This includes a range of features, such as the perception on sponsorship [143], methods of engagement [171, 104] and an evolving trend of consumer needs which outlines the importance of interactive and audience focused experience [83, 133]. At the same time, consumer research in sports and esports have also highlighted the possibility space for interactive content, with data-driven content forming one venue for delivering such content [10, 116, 132, 111].

E.2.2 Data Driven Audience Experiences

Broadcasting and streaming of esports share many similarities with physical sports. Footage is captured through an in-game camera and transmitted in near real-time. In esports, camera position is flexible due to the virtual nature, e.g. viewing through player perspective - such as in CS:GO[8] - or through an isometric top-down spectator view - such as in Dota 2[9]. Previous work has sought to optimise esports interfaces for spectators using data [16, 172, 21, 19, 176]. One of the underlying motivations being to create *'Information Asymmetry'*, in which the audience has more information than players [21]. This helps generate suspense in esports in the same way as in physical sports, where audiences can have a better overview of the game.

A range of products introduce some form of data-driven visualisations or overlays that augment linear broadcasts [156, 45]. Prior work have shown that data-driven content can measurably invoke emotional responses from audiences [10] and improve the range and diversity of storytelling in esports broadcasting, leading Kokkinakis et al. [82] to coin the term **Data-driven Audiences Experience (DAX)**. Most esports content producers, such as *ESL* [45] and *PGL* [156] use solutions that present key statistics to the audience in overlays on top of the main broadcast view. One example by Jonathan "PimpmuckL" Liebig [67], provides a graphics engine for *Dota 2* that adds non-interactive statistical information onto the spectator interface. Similarly Block et al. [10] introduced a system which utilized machine learning to identify top performances during a live match, and generate corresponding audience-facing graphics. Research in esports analytics has also focused

[8] `https://store.steampowered.com/app/730/CounterStrike_Global_Offensive`
[9] `https://www.dota2.com/play/`

on providing augmented maps to illustrate high-level gameplay [176], similar to the visual overlay of tactical formations in e.g. football. However, these solutions are predominantly controlled by broadcasters, and are typically delivered as part of a linear - one size fit all - broadcast feed.

Conversely, live viewing experiences that are interactive are comparatively rare. Examples include Charleer et al. [19], who introduced interactive data-driven dashboard for *League of Legends* and *CS:GO*. These give viewers access to live in-game statistics, including some performance measures which aim to be understandable to a non-expert audience. A range of commercial mobile apps exist and have been investigated in the literature [82]. These apps most commonly give users access to statistics, match schedules and simple match recaps. Similarly, major traditional sports leagues use apps to provide statistics, such as *FIFA, NFL,* and *Soccer* [42, 115, 167, 153]. Furthermore, web-based post-game analysis services also exist, which provide statistics or map-based analyses, e.g. *Dotabuff*[10] and *TrackDota*[11]. While these services provide a more customisable user experience, they are most commonly detached from the main broadcast platform, accessible via a companion app or a separate website. This creates a separation from the linear feed - introducing extra hurdles imposed in the audience - impacting reach, thus reducing the overall benefit associated with the experience they provide.

Despite this clear need for more interactive, audience focused, data driven experiences, no work in literature found at present has evaluated a comprehensive study of audience behaviour within an active broadcasting environment. This study aims at filling this gap by conducting the first large scale

[10] `https://www.dotabuff.com`
[11] `https://www.trackdota.com/`

longitudinal case study of user behavioural data within a seamlessly integrated user experience deployed alongside a linear broadcasting feed. Furthermore, this study provides a design framework and its evaluation - through the implementation of the *Dota 2 Twitch Extension* - aimed at informing and advising the development of further similar features.

E.3 Case Study: The Dota 2 Twitch Extension

The contributions in this paper build on a two year long process of co-design between the authors and one of the world's largest independent esports company (ESL), producing an interactive twitch extension that was deployed at a leading international esports tournament. This section describes the context of the case study, the underlying design process and the resulting *Dota 2 Twitch Extension*. The user evaluation and data analysis will be presented in Section E.4.

E.3.1 Dota 2

Dota 2 is one of the most established and popular esports titles, attracting record price pools and consistent viewership [159, 158]. This title is a competitive fantasy game, in which two teams of five players compete for resources in a large battle arena. Each player controls a unique *hero*, picked during the "draft" phase (directly before the match starts) from a pool of over 100 available heroes. Each hero has unique strengths and abilities. Both teams tactically pick their heroes to create synergies within the team and counter their opponents strengths. Once the match starts, heroes start at their base

- or *ancient*. Each base is heavily guarded by several perimeters of defences. The aim of the game is to defend one's own ancient, while breaking down the opponent's defences and destroy their ancient. Heroes start at a low power level and with minimal gear and need to increase their strength before attacking the enemy base. To do so, both teams collect resources within the battle arena to level up heroes and acquire new gear that gives players additional abilities. Resources are scattered around the arena in form of non-player character called *creeps*, which yield gold and experience when slain by a player. Throughout the process of levelling their heroes, both teams frequently clash, leading to spectacular engagements and team fights [146, 168] that yield gold to the winning team. When a hero dies, it re-spawns at its base after a time penalty, giving the winning team a further advantage in progressing their siege.

Dota 2 provides a rich environment for the experimentation with interactive viewing elements. Gameplay in *Dota 2* is fast-paced and involves frequent occurrence of multiple important events taking place at once. Heroes are usually dispersed across the arena to maximise resource extraction, and clusters of player frequently clash with their opponents. Broadcasters, however, can only capture a small portion of the battle arena at a time, leaving many important actions and events hidden from viewers [168]. As a consequence, commentators and audience members commonly need to draw on live statistics and additional layers of information about player performance to fully understand the unfolding action. *Dota 2* provides a wealth of open data, both live and historical, that lends itself to the creation of interactive spectator experiences. The commercial significance and large viewership, combined with its tactical depth and opportunities to reveal additional layers of information to viewers through it's available data interfaces make *Dota 2* a conducive environment for the design and study of interactive viewing.

E.3.2 Design Process

The case study presented here is the outcome of a two year long innovation project between the authors and ESL, one of the world's largest independent esports company. Building on our prior work in esports [10, 29, 82, 22, 119, 30, 142, 168, 81], the iterative design process included consultations with senior stakeholders at the company and close co-design with fans. Key decision makers and stakeholders were identified by the company leadership. The design process involved over 20 design iterations throughout a 24 month period, including focus groups, surveys and interviews with hundreds of fans. Company stakeholders were consulted through regular focus group sessions, in which designs and prototypes were presented throughout the iterative design process. Participants of the focus groups were identified by the company's leadership team based on their involvement in tournament production. The sessions were shadowed by all researchers, and observer notes were aggregated and translated into design requirements. We also drew on on the outcomes from surveys and interviews with fans from our prior work [10, 82], which were conducted on-site at 5 major international tournaments produced by ESL. The process generated a suite of interactive fan experiences, including an experimental mobile app and various virtual reality experiences. The *Dota 2 Twitch Extension* presented in this paper is the product of the collaboration, reflecting learning from previous design iterations and user evaluations. However, in contrast to the other apps and VR experiences, which were standalone experiences accompanying the main broadcast, the *Dota 2 Twitch Extension* was tightly embedded with the primary live broadcast of the event, delivered via Twitch. For instance, any information displayed on the Twitch extension would occlude the video feed and compete with viewers attention. This created a series of new design challenges, which the research team translated into design guidelines together with senior experts at the company from across broadcast, product and innovation divisions. The design guidelines encompass elements of user experience and usability, as well as aspects of the content of the interactive experience, i.e. what types of additional on-demand information is most suited to be delivered embedded alongside the live video stream. The final set of design principles guiding our design are as follows:

- **Unintrusive**: It is important that the user experience does not distract or obstruct main (linear) coverage permanently. Spectators need to be in full control of showing or hiding interactive content, while maintaining focus in the main coverage content at all times. This is particularly relevant for quick and time sensitive events of high narrative importance.

- **Discoverable and controllable**: Fans need to be able to easily gauge what is offered by the extension, and the mechanism for discovering interactive feature needs to be easy to learn. Users should also be able to personalise content to their specific interests, even if they diverge from the primary commentary.

- **Game status and highlight tracking**: Esport titles are complex and it often hard for fans to judge the state of the game and each player's performance. Interactive content should aim to advice and explain to audiences the key moments as well as core indicators in order to break down the state of the game in a understandable manner. A comprehensive status overview also allows the viewers to better anticipate what will happen, and sets the stage for surprise.

- **Providing context**: Due to the high complexity and varied nature of esports, it can be challenging to fully judge key events. For instance, in *Dota 2* heroes can pick up certain items that dramatically alter the power of a hero. The timing of pick-up - whether it is early or late in the game - is crucial to provide meaning to fans. Interactive overlays provide opportunity to offer such context (for instance, by taking into account historical match data), allowing fans to develop deeper understanding of key events.

- **Facilitate catch-up**: Esports audience commonly tune in late and consume live esports alongside other activities (e.g. playing games) and across different devices (e.g. through mobile devices while on the move). Interactive experience should be utilised to allow fans to catch up after missing parts of the match, provide key highlights and facilitate transition of focus between watching the live action and competing activities.

- **Tournament status**: Tournament standings and schedules are important to fans, but often located on separate website. Interactive overlays allow the provision of such information in close proximity to the actual coverage.

- **Visual integration with tournament style and game UI**: The extension had to conform to the tournament's style guide, including colour palettes, fonts and tournament branding. Similar to visual broadcast graphics, the extension also needed to be visually distinguishable from the game's visual style and user interfaces to clearly separate interactive components from the underlying non-interactive video feed.

The set of identified design principles were then translated into concrete design and working prototypes by the esports company's technical team. The extension was trialled at three international tournaments, with minor refinements being implemented following focus groups with fans. The final design and large-scale deployment are the basis for the case study presented in this paper.

E.3.3 The *Dota 2* Twitch Extension: Description

The *Dota 2* extension broadens the video stream with a set of interactive information panels that can be accessed any time before, during or after live matches. In order to adhere to the the "**Unintrusive**" principle, all panels are initially closed, and the extension presents itself to users as a collapsed column of five navigation button floating in the centre-left area on top of the video feed (Figure E.1). This specific placement location has been selected as it does not interfere with existing elements on the video streams, such as bottom halves or in-game displays. Each button is associated with a separate information panel, which can be opened and closed at any time by the user. This allows the user to take control over what and when information is displayed ("**Controllable**"). When opened, the panel itself has a semi-transparent background to further minimise disruption from the linear broadcast.

When loading the extension, the top-most button - **Live Recap** - presents a tooltip that makes viewers aware of the Live Recap function and marks unread notifications as red dots ("**Discoverable**"). Clicking the button opens the Live Recap panel (Figure E.2). The panel consists of a scroll panel that chronologically lists important highlights from the match, as well as periodically providing match summaries ('This match so far'). The Live Recap is designed to highlight notable events and performances to viewers that are usually invisible to viewers, as well as to allow viewers who join the stream at any point to catch up with the live match ("**Highlight Tracking**", "**Facilitate catch up**"). Furthermore, key moments of the match provide contextual information. For instance, when a key item is picked up, the summary states if this is "fast", "average" or "slow", by comparing timings to historical matches of the current season ("**Providing context**").

The second button opens the Player Performance panel (Figure E.3). On the top, both teams logos and their win probability are shown, which is calculated by taking into account hundreds of key performance indicators (Section E.3.4 will provide details on how data-driven content is generated). In lieu of a reliable "score" (equivalent of goals in football), the win probabilities give viewers a more refined birds-eye view on "**Game status**" and help viewers "**Catch up**" after a break to gauge the overall state of the game. Below the win prediction, the panel lists a head-to-head comparison

Figure E.1: In its initial state, the *Dota 2* Twitch extension provides a column of five yellow buttons on the centre left portion of the game footage. The top button - Live Recap - can present push notifications, informing users of key events during the match.

of team and players performances, highlighting important key performance indicators and showing how well each player performs on a normalised scale from 0% (poor performance) to 100% (top performance), providing crucial context for judging each player's performance. The performance indicator is explained in detail in the previous work by Demediuk et al. [30]. The performance indicator is calculated by taking into account a range of contextual factors (the chosen hero, their role, time of the match) as well as 13 different Key Performance Indicators, such as gold earned or kills scored [30]. For each player, the top three most impactful Key Performance Indicators are listed, giving viewers insights into which aspects of the players' performance determine their performance index.

The third panel the Player HUD - presents in-depth status information about each player (Figure E.4). At the top, the HUD shows the players hero, name and level, alongside primary KPIs, such as the number of kills and the effectiveness in earning gold. Underneath, the HUD visualises how each player develops the talents and skills of their in-game characters as well as

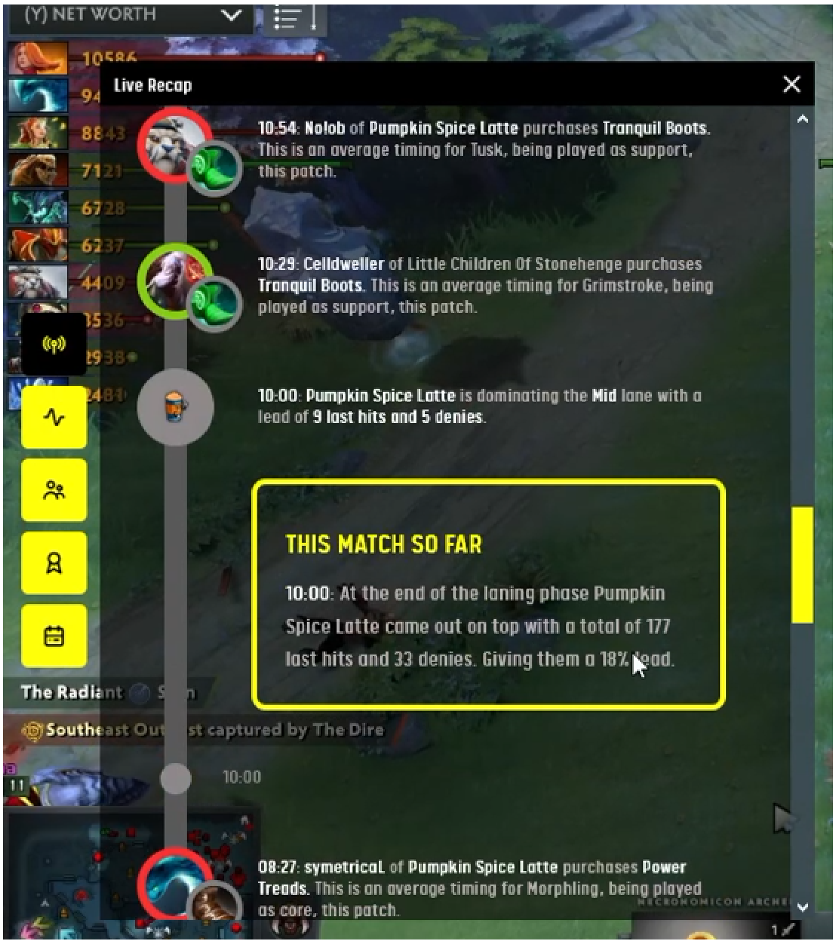


Figure E.2: The live recap provides a chronological list of important highlights and outstanding performances as well as offers periodic match recaps that allow viewers who have tuned in late to catch up on the action.

Figure E.3: The performance tab shows an overall win probability for each team, and lists key performance indicators of each player alongside their contribution to the team's chance of winning.

provides a status of their health and resources (green and blue bars). Lastly, the HUD shows the items each player has in their positions (giving characters important power boosts), as well as their performance in obtaining gold. Some elements in the HUD allow for additional interaction that reveals more layers of information. For instance, moving the mouse over a skill or item provides a tooltip with details about the item. The provided information gives viewers more insights and ability to explore performance beyond what is normally covered in the linear video feed ("**Game status**").

Figure E.4: The player Heads-up-display (HUD) provides detailed status information about each player.

The last two interface panels provide information about standings and schedule of matches, this covers the further information needed which has been highlighted in the "**Tournament status**" requirement. The Standings panel provides a tree view of the group or knock-out stage, depending on the stage of the tournament. If a live match is on, the current match is highlighted visually within the standings. The final panel - schedule - provides information about upcoming matches. Both panels provide convenient access to information otherwise only found disparately, across multiple web-sources and apps. Both standings and schedule are available any time during the tournament (before, during and after matches).

Overall, the *Dota 2 Twitch Extension* creates a rich interactive experience that accompanies the live broadcast. The extension presents highlights to the viewers that might have been missed on the primary coverage, and gives individual viewers a broad range of on-demand content.

E.3.4 Implementation

The extension is developed in HTML and JavaScript, utilising the Twitch API [31] to integrate with the live video stream. The twitch extension receives its content through a dedicated data link with a bespoke data service, which was designed and implemented by the authors (a detailed description of the data analytics and machine learning techniques can be found in [82, 29, 30]). The data service has access to the live match data, which it receives from the tournament operator. The data service compares the live match data to thousands of matches from the current season, and automatically generates highlights, status overviews, analyses of performance and match predictions, which are then broadcast in form of a structured JSON packages to all viewers via Twitch's content delivery network. The extension runs in the browser of each viewer and translates the data packages into the described interactive user experience.

E.4 User Evaluation

The *Dota 2 Twitch extension* extension was used during the *DreamLeague Season 15 DPC Western Europe* for both Upper and Lower division for the period between *Apr 24 2021* to *May 22 2021.* The 4 week event provided an opportunity to study how audiences engaged with the interactive features in an ecologically valid setting. For this purpose, the extension was modified to collect anonymised telemetry data for each individual user. Full ethics approval was granted by the University of York's ethics committee. The aim of the data collection was to characterise general usage patterns of interactive features as well as validate our design principles. Specifically, the study sought to identify how do passive consumption of the linear video stream and active engagement with interactive content entwine depending on various stages of the tournament, and phases of live coverage.

In order to draw observations and comparisons from the data, several steps were followed to allow for data validation and synchronisation, in addition to analysing user activity and defining interactions between user and extension. This section first outlines the steps utilised in pre-processing user data, including steps taken for collecting data, followed by a general analysis of user interactions and how it relates to real world events, and ending on an analysis of user consumption patterns which assist in defining how an interaction behaviour can be observed across users.

E.4.1 User Analytics Capture

In order to collect usage data, the *Dota 2 Twitch extension* utilised the Google Analytic API [47] that allowed for several aspects of user interaction

Table E.1: Dataset feature overview

Feature	Description
User ID	A unique ID that relates to a single user
Time	Human readable time of interaction
Delta	Time difference between two consecutive interactions
Category	A category defining the event, used to identify the event
Label	Additional context for the event when available
Value	The numerical value associated with the event when available
Action	The textual value associated with the event when available
Unix Timestamp	The time in Unix Index (seconds) that the event was performed by the user

to be collected for analysis. Table E.1 contains a breakdown of all of the features present in the usage dataset.

The *Dota 2 Twitch extension* deployed during the *DreamLeague Season 15 DPC Western Europe*, from the period starting **Apr 24 2021** to **May 22 2021**. During this period, the *Dota 2 Twitch extension* covered 60 matches for the upper and lower divisions, which generated data from *306,545* unique users. Data was collected for all users who enabled Twitch extensions. Some browser settings prevent google analytics from collecting data. Packet loss was also commonly observed (see next subsection). Users who did not actively engage with the extension did not generate active log entries. However, the delivery of the push notifications (the little red number flag on the "live recap" button) was recorded for all user, regardless of their activity.

E.4.2 Data Validation and Cleaning

Of the 306,645 total users, *101,729* users (~33%) engaged with the extension directly. The rest received push notifications, but did not otherwise engage with the extension and are therefore excluded from further data analysis. In order to evaluate interactions across the different stages of broadcast, exact timings of the different matches and their phases were collected. The two main phases of a *Dota 2* match consist of the *Draft Phase* (in which teams pick their heroes) and the *Game Phase* (the actual match). This process was automated using the OpenDota API [25]. Additionally, timings of the different phases within a match were determined utilising the Clarity Parser [155] library - a open source Java library for parsing and processing *Dota 2* replay files.

All times were converted into UTC times so the information could be synchronised with the Goggle Analytics data. An artificial delay of five minutes was added to all game times, as live video of the event, as well as the data received by the extension was delayed by that amount by the tournament organiser. This is a common anti-cheat technique in esports broadcast, particularly during the COVID-19 pandemic in which players competed from home.

In order to assess and validate the dataset further, the usage dataset was pruned of errors, such as faults caused by network traffic or other similar technical issues. This was achieved by investigating the sequence of actions performed by any given user, particularly the "Category" of the events. The event categories relate to the type of event, including:

- **OpenWidget**: when a panel has been opened.

- **CloseWidget**: when a panel has been closed.
- **Hover**: when a user hovers their mouse over any relevant UI elements, which reveals additional textual information in the form of tooltips.
- **Scroll**: when the user scrolls through text.
- **Tab Open**: when the user opens an additional window within a panel.
- **RecapNotification**: when the user receives a push notification indicating there is a new live recap story element that can be read.

By identifying possible sequence of events, a simple finite state diagram - as depicted in Figure E.5 - can be reproduced to validate and identify any unexpected behaviour and remove it from the dataset. For example, a user path of "OpenWidget -> Hover -> Scroll -> CloseWidget" would be a valid path, whereas a path of "OpenWidget -> CloseWidget -> CloseWidget" might indicate a problem with the logged data due to two consecutive close events.

Finally, after removing users that did not engage with the extension, and discarding faulty interactions (i.e. that have reached unexpected states) a total of *101,729* unique users were included in the data analysis.

E.4.3 Data analysis

In total the cleaned dataset contains *1,628,017* distinct interactions within the four week period of **Apr 24 2021 - May 22 2021**. Figure E.6 depicts the average distribution of activity events per user. Outliers above the upper quartile range were removed due to a large deviation of values between maximum and remaining values. A series of observations can be made. Given

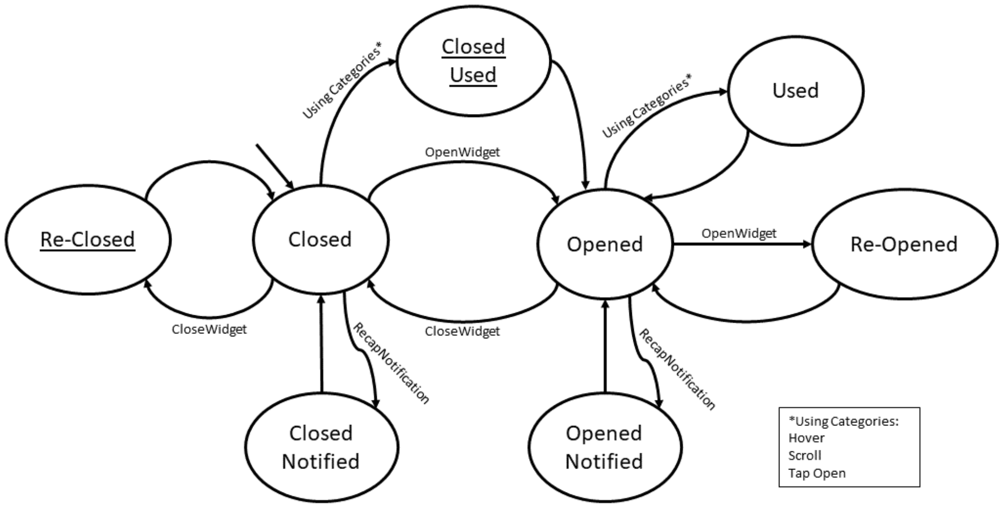


Figure E.5: A graphical representation of the Finite State Machine used to parse through and join user interactions. States indicating a unexpected sequence of actions have been underlined.

that we had removed 'passive' users from the dataset, the majority of log information relates to "active" interactions that require user input - the mean for the "passive" *RecapNotifcication* was comparably low. Secondly, users commonly inquire deeper into the interactive feature than just opening and closing the extension, with a wide range of interactions recorded (e.g. *Hover* and *Tap Open*). Both *Second Level* interactions - switching to new panel when the extension is already open, *Third Level* interaction - switching sub-panels - and *Fourth Level* interactions - inspecting additional information in the sub-panels via "Hover" as well as "scroll" - are commonly observed.

Note that across all users, the average was 8 actions combined. This average number is relatively low, given the data was collected across a four weeks period (but see Commercial Implications in Section E.5). Figure E.7 shows a histogram of number of interactions per user. From the *101,729* users that have engaged with the extension, *15,302* have interacted 20 or

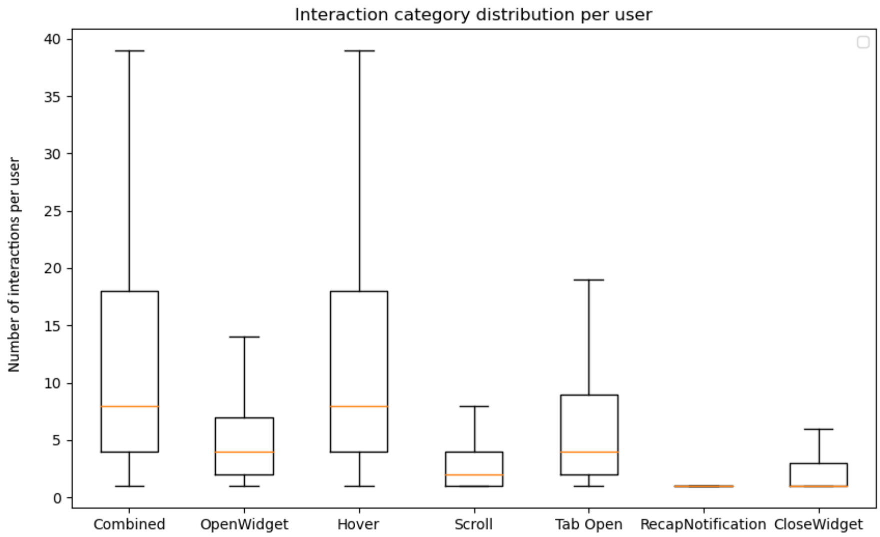


Figure E.6: A box diagram depicting the distribution of user interactions per category - outliers have been removed for display purposes

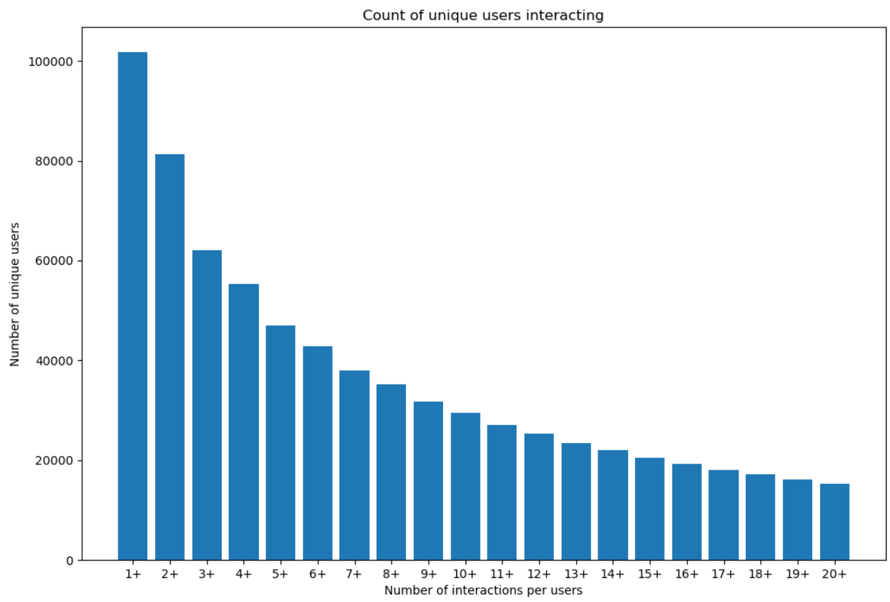


Figure E.7: A histogram that displays the amount of user interactions per users more times. Conversely, a large proportion of the population has interacted fewer than 10 times. Consequently, the majority of interactions are generated by a minority of "power users". This distribution of engagement is well known from other observational studies [9], splitting users into a large portions of "shoppers" (displaying only shallow engagement), and "power users" (show prolonged engagement and fully explore the interactive offerings).

Longitudinal Analysis

We analysed engagement with the extension across different stages of live coverage by synchronising log telemetry data, as well as match and game-phase timings. The majority of interactions (68.5%) occurred during the live match phase. 5.2% of interactions took place during the relatively short draft phases (15 - 20 minutes). The remaining 26.3% of interactions happened in the time period between matches. Note that the extension's full functional capacity was only available during the live match. Between matches and during the draft phase, the extension only provides tournament schedule and standings. However, the data suggests that viewers were aware of the extension during all stages of the tournament, and increased their engagement during the primary game phase. This also provides evidence that the primary data-driven content presented in the extension - providing context and in-depth analysis - provides additional value during the game phase.

Figure E.8 displays user activity for the duration of the final match, during which a peak number of concurrent users can be observed. The example illustrate the slow build-up of engagement with the widget before the match (white background) and during the draft phase (light blue). In both cases, only tournament status and schedule are populated, all other tabs remain inactive until the match starts. During the live match (green background), activity across all types of interactivity increases, with notable peaks and troughs. Usage of the extension builds up over the course of the example game, then drops a few minutes before the end of the match. *Dota 2* involves variable gameplay, creating both quiet periods of no conflict and short bursts of intensive team encounters, which may explain the peaks and troughs in active use. The last minutes of a match usually involves substantial action, as one of the teams invades the enemy base, which may explain the drop-off. Similar patterns could be observed across the other matches. While further analysis is required, the data suggest that the observed patterns of interaction may correlate with the cadence of the match. This is further discussed in Section E.5.

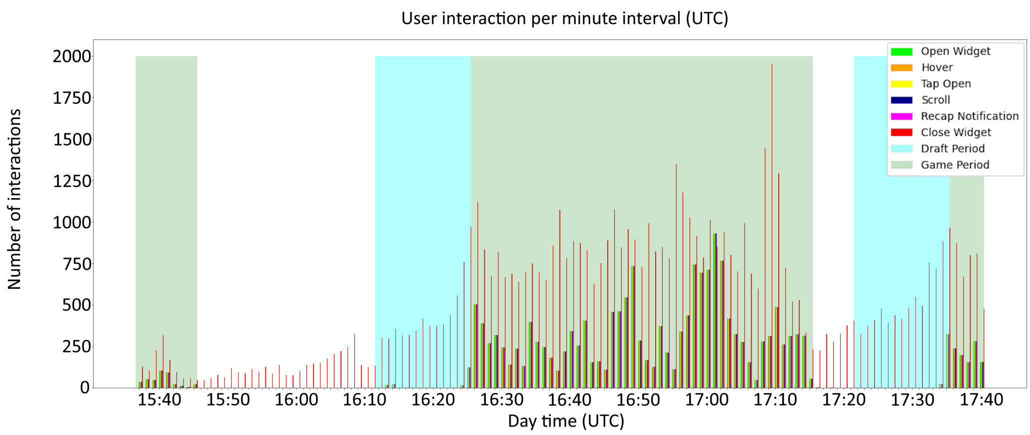


Figure E.8: A demonstration of user activity in one-minute intervals for Upper Division last week final Game 1 with 30 minutes offsets

Usage of the extension also increased over the four week period, peaking during the final weekend (Figure E.9 depicts). This is expected, as it follows general viewership, which typically increases over the course of a tournament.

The authors did not have access of match-level data on total viewership (Twitch only provides aggregate statistics of daily peak viewership), therefore a full comparison was not possible.

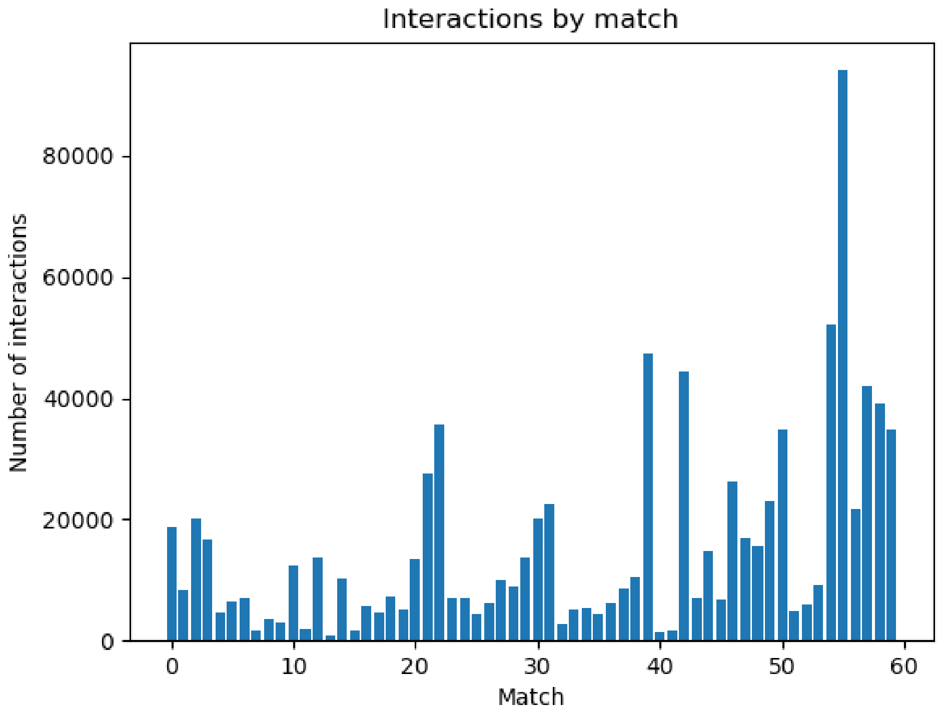


Figure E.9: A histogram displaying the number of users active at all games covered in the series

Active vs. Passive Engagement

Previously described analysis has focused on aggregate numbers of discrete activity entries. This section focuses on analysing sequences of actions performed by individual users, and how phases of "active engagement" can be further qualified. The aim is to provide additional insight of how the viewers' focus switches between active use of the extension and passive consumption of the video stream. We define active engagement as periods of time in which the user is predominantly focused on the extension. For instance, a sequence of actions where the user opens the extension, selects the Heads-up-display, a player and then hovering the mouse over an item to get a tooltip. During this time, it is likely that the user is fully focused on interacting with the extension. In an alternative scenario, a user opens the extension and leaves the live recap open for one minute, then switches to another panel, and subsequently closes the extension. In this scenario, while the extension is visible, the user is likely to split their attention between the extension's information displays and the live video stream.

The challenge of characterising "active engagement" in our telemetry data is in identifying a suitable time threshold that links individual actions into sequences of active use. If time lapsed between two subsequent activity events is below the threshold, it would link them into a period of active use. Multiple subsequent events that all fulfil this criteria form a larger active period. If the time lapsed between two subsequent activity events is above a certain threshold, it is assumed that the engagement is passive, in which the user may not be fully focused on the extension. Literature suggests that user attention retention is unlikely to be maintained over long periods [148]. Particularly as the information being consumed by users is displayed alongside a *Dota 2* game, which increases the amount of overall information displayed on screen [10]. Furthermore, it is known that sports can modulate and impact audience attention retention [3], however the effects of esports has not been tested. Similarly, single interaction performed by the users are expected to retain their attention for a limited, but non-zero length of time.

To determine a suitable threshold, a histogram showing the distribution of time lapsed between any two subsequent events in the dataset (Figure E.10) has been plotted. The first bar shows that around 270,000 log entries occurred less than 1 seconds from each other. Of the *768,309* total interactions, *155,145* (approximately 20%) were separated by more than 30 seconds. The distribution focuses on time scales that are reasonable to consider for an activity threshold.

Figure E.10 provides a visual representation of usage patterns, displaying the number of times that users performed two consecutive actions (X) seconds apart. The data shows that the time between interactions plateaus relatively quickly at approximately 8 seconds, thus user patterns past this threshold are roughly equally distributed. By utilising the Elbow technique [34], a threshold of *5 seconds* can be established, as this the most pronounced curve in the data distribution. This time period can then be used as an approximate duration for which users are actively engaging with the extension. While this is an approximate value, it suggests that a large number of subsequent actions happen within that time period. This could assist further analysing the data and provides a starting point for quantifying user interaction based on engagement patterns.

Using the active threshold outlined, it is then possible to identify all sequences of active usage. This can be defined by a sequence of actions where all log entries succeed each other within 5 second. A sequence can contain a single interaction, if it is isolated from previous and subsequent log entries by more than 5 seconds. Using this method a total of *121,101* sequences

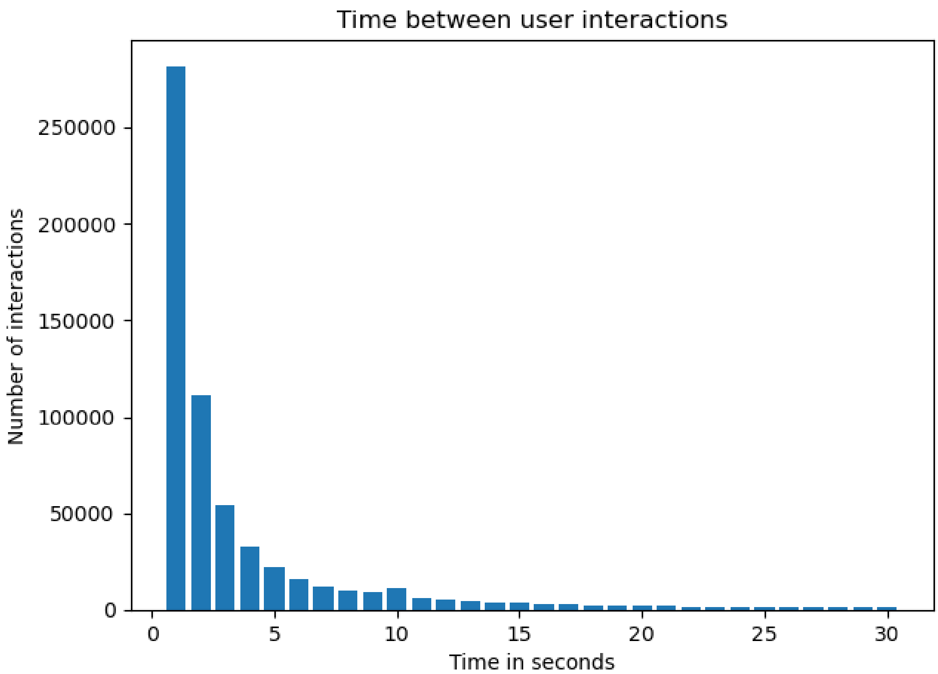


Figure E.10: A histogram displaying the time difference between any two consecutive interactions capped at 30 seconds were identified across all users. Figure E.11 shows a distribution of number of interactions per active sequence. The distribution shows that a large number of sequence only consist of a single isolated action. However, a total of *74,643* sequences (62% of all sequences) contain two or more interactions. The distribution also shows that the number of interactions per sequence is varied.

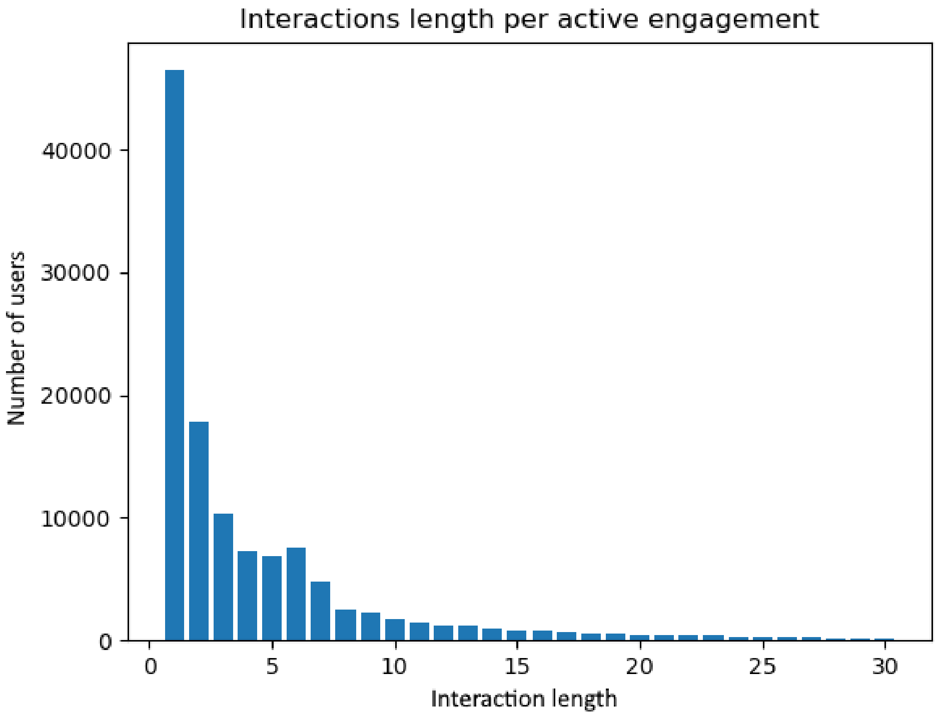


Figure E.11: A histogram depicting the number of interactions for active engagement

Figure E.12 shows the distribution of users based on the percentage of time spent actively engaging with the extension as per our measure. This was achieved by calculating the duration of each chain of interactions, and compared with the overall time the extension was visible. The maximum

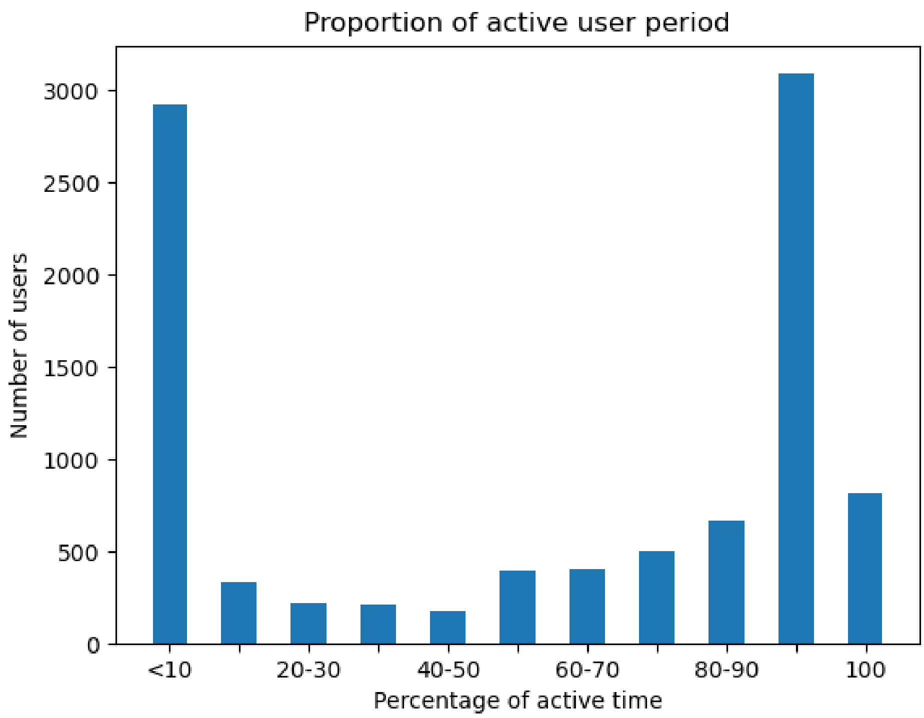


Figure E.12: A histogram displaying the proportion of time users spend actively engaging with the extension in relation to the time the extension remained opened value was calculated by measuring the time difference between the cumulative time between all pairs of "OpenWidget" interaction and the subsequent "CloseWidget". The data shows that a large number of users spent less than 10 percent of their time 'actively' engaging with the extension - as depicted by a large peak in the "<10%" bin. However, a similarly large number of users spent the vast majority of their time (>90%) actively engaged with the extension. This usage pattern seems substantial, given that the extensions competes for attention with the live match. Additionally, the data shows that the rate of occurrence and duration of active usage may correlate with the cadence of gameplay. Figure E.13 displays all active sequence performed by all users within the upper bracket Game 1 final. In this graph, every user is depicted as a thin horizontal row on the *Y-axis*, with active sequences being connected in blue. Longer horizontal lines mean longer active sequences. Short lines or dots represent short bursts of activity. All users are stacked vertically. The X-axis represents time from the start of the draft phase to the end of the match.

As depicted in Figure E.13, the active engagement periods clearly cluster around certain time of the match (see regions that are more intensely blue). Other regions of the figure are mostly white and devoid of active usage, suggests that little interaction occurred across all users during this time of the match. While more in-depth analysis is required, this provides an indication that in-game aspects may affect how the extension is used, and consequently how the audience attention can shift during the match.

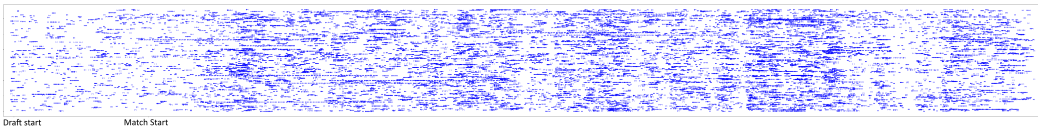


Figure E.13: User activity performed in the Upper Division last week final Game 1 as chained by the Active period threshold of 5 seconds. The X-axis represents chronological time from start of the draft period until the end of the match. The Y-axis represent user activity where each unique user has been assigned a horizontal value in the axis.

E.5 Discussion

The data analysed in this paper draws a detailed picture of how users interacted with the interactive content provided by the *Dota 2 Twitch extension*. The following sub sections discuss the major findings of the analysis, with a specific focus on validating the principles underlying the design of the extension. Finally, this section outlines limitations of the presented study and identifies areas for future work.

E.5.1 Overall Characterisation of Usage

The *Dota 2 Twitch extension* introduced novel interactive content alongside the video stream. Despite not being actively advertised or explained as part of the video coverage, the extension organically attracted a substantial number of viewers. Roughly one third of all viewers captured by the telemetry data actively engaged with the extension. Of those that did engage with the extension, the depth of interaction was varied. 55% of users displayed only "shallow" engagement, dipping in an out of the extension only once and never returning. However, 20% of active viewers can be considered "power users" who displayed active prolonged engagement, and explored the full functional depth of the extension over the course of the tournament. In between shallow and power users is a varied gradient of engagement. However, while this distribution of engagement is not new in the context of prior research (e.g. visitor studies in museums [9]), this study is the first to quantify large-scale adoption of interactive offerings in the context of live esports and sports. While engagement will depend on the specific design of the interactive offering (see Limitations & Future Work), this study observed a high "organic" uptake (without additional advertisement or incentives) and good levels of engagement considering the added experiences compete with action-packed live coverage. Consequently, the data provides clear evidence that interactive content is in demand, and can help conversion of audience from passive consumers to active viewers. The varying levels of engagement and evidence for systematic functional exploration across users suggest that the design was successful in being "**Discoverable and controllable**". However, based on the data, it remains uncertain if every user was able to discover the functionality. The design was purposefully subtle, only showing five buttons in a marginal area of the screen when inactive. Two thirds of viewers never engaged with the extension, suggesting the design was also successful in being "**Unintrusive**". Unintrusiveness and discoverability are at obvious odds with each other. The measured engagement characteristics can likely swayed towards higher adoption rates by deploying animation or active guides to encourage usage. Additionally, explaining and encouraging use of the interactive extension in the main video feed (e.g. through panellists) may result in increase adoption rates.

E.5.2 Contextualising Interaction

The study showed that levels of engagement varied depending on the tournament stage and on the stages of live coverage during each day. Since this extension provides most of its content during live matches, this stage saw the highest usage. Viewers consistently opened tournament schedule and standings throughout the broadcast, validating our "**Tournament status**" design criteria. Within the live match, usage tended to build up as time progressed suggesting that the relevance of the provided information displays increased to viewers. A possible explanation for this observation is that judging performance and strategy in *Dota 2* matches becomes increasingly complex as the match progresses. The extension was purposefully designed to show "**Game status and highlight tracking**" as well as "**Providing Context**". The data-driven insights provided by the extension (such as win prediction and performance indicators) may thus become more valuable as the match enters later stages.

E.5.3 Active vs. Passive Usage

We introduce a mechanism for identifying "active" use, defined as a sequence of interaction happening in short succession, based on a 5 second threshold. This analysis brought out two common forms of interactions across users. By analysing those behavioural patterns, a set of three engagement levels can be determined. In a large number of cases, users engage with the extension in bursts of connected interactions. This can be commonly observed when users open the extension and navigate through several panels utilising features and engaging with the extension in varying ways. This interactions typically occur within short intervals of times - commonly within the highlighted threshold of 5 seconds. During this period, it is expected that the user is actively engaging and consuming the information being displayed. In another common behaviour of interaction, users open a panel that remains open for a long period of time. During this time users perform actions sporadically, with large periods of inactivity in between actions. In this case, information is being displayed and it is expected that the user is passively consuming the information with limited attention, as this is being shared with the linear coverage feed in the broadcast. Lastly, it is important to note that there are periods of time during which the extension is closed completely. This provides an insight into a third behavioural pattern, during which users and not engaging with the extension, neither passively nor actively.

E.5.4 Interaction with In-Game Events

The data included various indicators that usage of the interactive features and events happening in the virtual game worlds correlated. First, a sharp drop in engagement with the interaction in the final minutes of each match can be observed. This stage often entails intense action that audiences follow closely. Secondly, there were also clear peaks and troughs in usage of the extension throughout the live match across all users. These fluctuations in collective usage are likely linked to particular in-game events as well as the general cadence of the match. However, a statistical link could not yet be established and should be explored further in continued work. Identifying key events and measures for such a “cadence” are non-trivial, involving the detailed analysis of replay data. Esports such as *Dota 2* provide rich match recordings for each match, enabling the detailed correlation between audience telemetry and game analytics. This can be subject to future research.

E.5.5 Commercial Implications

It is important to contextualise the findings from this paper within the commercial drivers of the esports ecosystem. Most esports broadcasts are free to watch. The majority of esports revenue comes from sponsorship [111], which have been shown to have had a positive impact on brand awareness and image [39]. Consequently, the addition of new audience experiences, such as the one presented here, is ultimately driven by generating additional revenue. The collected data and presented methodology has various commercial implications. While not all viewers utilise the extension, those viewers who do, engage actively and receive added value through the contents of the extension. This can positively impact the effectiveness of advertising, as more disruptive forms for brand placements (such as rolling unskippable ads) can lead to negative impact on users, less disruptive forms have comparable brand effectiveness while reducing the negative impact to audiences [121, 5]. Sponsorship embedded in interactive experience is thus prone to generate a different depth of engagement with fans and a clearer perception of added value than traditional "passive" sponsorship displays embedded in the linear video feed [6, 74]. Even for the majority of fans who only click a handful of elements in the twitch extension over the course of the four week tournament, the short burst of interaction creates a moment of *focused and measurable attention.* From a sponsorship perspective, this interactive engagement is different to usual product or logo placements in the passive video, of which no metrics of exposure exists. Such 'interaction with brands' - similar to advertising placed within virtual game worlds - was considered valuable to the tournament organisers we worked with and deemed to create new sponsorship offerings that compliment existing inventories. It is also important to note that such interactive engagement lends itself to being personalised to each individual fan, which has been shown to offer similar levels of brand retention while reducing the negative impact to audiences [6]. Furthermore, interactive content generates detailed information about fans. While in this paper we deployed offline analysis of the telemetry data after the tournament, similar techniques could be utilised to collect real-time data about fans. A tournament organiser or brand could leverage such data to know when there is a downtime in the cadence of the match, when fans are active in the extension and what they are interested in (e.g. what player they have selected in the HUD). This could create numerous possibilities for the delivery of tailored advertisement or merchandise. Lastly, interactive extension could be leveraged to deliver premium content to fans and be monetised through subscriptions or micro-transactions.

E.5.6 Limitations & Future Work

The presented study provides the first large-scale analysis of how interactive audience experiences in esports are used in an ecologically valid environment. Many of the presented findings are specific to the design of the *Dota 2 Twitch extension*, and generalisation will require additional study of engagement across different esports titles and applications. The the presented design principles and methodological foundations for our analysis, including the addressed challenges of log analysis and definition of active use seek to inform future design and evaluation of interactive audience experience in esports.

Additionally, as explored in Commercial Implications, the analysis presented was performed post event. A possible extension of this work could be the use of *real-time* telemetry data to further drive fan engagement 'in situ', creating a series of interesting creative and commercial opportunities, as well as generating additional insights into audience behaviour. Combining real-time and historical engagement data from each user alongside generative AI, for instance, could enable much more sophisticated penalisation's an tailored content delivery. This, in turn, could enable better live customisation of data, catering to individual users needs and preference - ultimately generating value for fans and commercial opportunities for content creators. Similarly, analysis of live viewer engagement could also be leveraged to improve other elements of the broadcast, including social media engagement, or post-match analysis. For instance, the highlights treated during the post-match analysis could be informed by spikes in the use of the extension and automatically generate social media short-form content. This could also be further enhanced by continued work to investigate live usage with in-game "cadence" as explored in Interaction with In-Game Events.

Future work needs to also focus on greater customization in appearance and presentation. While the design language of the extension followed the "**Unintrusive**" general patterns, users may choose to customize the look to better suit their needs and preferences. Additionally, the extension also adhered to the overall tournament style guide, dictating several aspects of the look and feel. While this was a limitation in this study, future work in the domain could explore the effects of greater customization for appearance and presentation with continued engagement, in line with the "**Discoverable and controllable**" criteria discussed in this paper. Additional customizability could also have accessibility implications, allowing users to adjust the extension to suit their needs. This greater customization in addition to Real-Time Viewer Engagement Analysis could also serve to provide greater insights into why some users opted out of engaging with the app.

Lastly, it needs to be recognised that the behaviour of esports audiences - particularly of analysis-focused and data-heavy game titles such as *Dota 2* - may not generalise of the wider esports viewership. Future work has to focus on applying the analytical methods proposed here to other interactive audience experiences to generalise our understanding of audience behaviour and associated creative and commercial opportunities beyond this first case study. Similarly, the translation of interactive audience experiences to traditional sports needs to be subject to further study.

E.6 Conclusion

This paper presented the first large-scale case study of esports viewing consumption patterns an interactive, data driven audience experience in esports, used by over *101,729* people during the *2020 DreamLeague Season 15 DPC Western Europe* tournament. By investigating sequences of user interactions, this paper proposes an active period threshold of 5 seconds. This in turn allows for a more refined investigation of how users engage with interactive overlays. By studying user activity in conjunction with the active period threshold proposed in this study, this paper documents methods for detecting behavioural user patterns. Future work will investigate additional details in the potential relationship between user telemetry and in-game events, which may help explain the observed interactions. The contributions raised by this paper can also be used to assist the understanding of future challenges and opportunities in interactive viewing across sports and entertainment.

References

[1] Ana Paula Afonso, Maria Beatriz Carmo, and Tiago Moucho. Comparison of visualization tools for matches analysis of a moba game. In *2019 23rd International Conference information visualisation (IV)*, pages 118–126. IEEE, 2019.

[2] Joseph Ahn, William Collis, and Seth Jenny. The one billion dollar myth: Methods for sizing the massively undervalued esports revenue landscape. *International Journal of Esports*, 1(1), 2020.

[3] Zahra Aminiroshan, Seyed Morteza Azimzadeh, Mahdi Talebpour, and Majid Ghoshuni. The effect of sport events environments on audience attention to advertising using brain activity. *Annals of Applied Sport Science*, 9(1):0–0, 2021.

[4] Roy F Baumeister and Debra G Hutton. Self-presentation theory: Self-construction and audience pleasing. In *Theories of group behavior*, pages 71–87. Springer, 1987.

[5] Daniel Belanche, Carlos Flavián, and Alfredo Pérez-Rueda. Consumer empowerment in interactive advertising and ewom consequences: The pitre model. *Journal of Marketing Communications*, 26(1):1–20, 2020.

[6] Steven Bellman, Robert F Potter, Jennifer A Robinson, and Duane Varan. The effectiveness of various video ad-choice formats. *Journal of Marketing Communications*, 27(6):631–650, 2021.

[7] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[8] Kokten Ulas Birant and Derya Birant. Multi-objective multi-instance learning: A new approach to machine learning for esports. *Entropy*, 25(1):28, 2022.

[9] Florian Block, James Hammerman, Michael Horn, Amy Spiegel, Jonathan Christiansen, Brenda Phillips, Judy Diamond, E Margaret Evans, and Chia Shen. Fluid grouping: Quantifying group engagement around interactive tabletop exhibits in the wild. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 867–876, 2015.

[10] Florian Block, Victoria Hodge, Stephen Hobson, Nick Sephton, Sam Devlin, Marian F Ursu, Anders Drachen, and Peter I Cowling. Narrative bytes: Data-driven content production in esports. In *Proceedings of the 2018 ACM international conference on interactive experiences for TV and online video*, pages 29–41, 2018.

[11] Florian Oliver Block, Marian Ursu, Jonathan David Hook, Ben Kirman, Anders Drachen, Simon Peter Demediuk, Athanasios Kokkinakis, Olu Olarewaju, Daniel Slawson, Alistair Coates, et al. Echo suite of software (showcase brochure), 2020.

[12] Ivan Bonilla, Andrés Chamarro, and Carles Ventura. Psychological skills in esports: Qualitative study of individual and team players. *Aloma*, 40(1):35–41, 2022.

[13] Brian Bowman, Niklas Elmqvist, and TJ Jankun-Kelly. Toward visualization for games: Theory, design space, and patterns. *IEEE transactions on visualization and computer graphics*, 18(11): 1956–1968, 2012.

[14] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[15] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 263–270, 1993.

[16] Christian Carlsson and Alex Pelling. Designing spectator interfaces for competitive video games. Master's thesis, Chalmers University of Technology, 2015.

[17] Francesco Cauteruccio and Yubo Kou. Investigating the emotional experiences in esports spectatorship: The case of league of legends. *Information Processing & Management*, 60(6):103516, 2023.

[18] Sven Charleer, Kathrin Gerling, Francisco Gutiérrez, Hans Cauwenbergh, Bram Luycx, and Katrien Verbert. Real-time dashboards to support esports spectating. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '18, page 59–71, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356244. doi: 10.1145/3242671.3242680. URL `https://doi.org/10.1145/3242671.3242680`.

[19] Sven Charleer, Francisco Gutiérrez, Kathrin Gerling, and Katrien Verbert. Towards an open standard for gameplay metrics. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, CHI PLAY '18, pages 399–406, New York, NY, USA, 2018. ACM.

[20] Kathy Charmaz. Grounded theory. *Qualitative psychology: A practical guide to research methods*, 3:53–84, 2015.

[21] Gifford Cheung and Jeff Huang. Starcraft from the stands: understanding the game spectator. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 763–772, 2011.

[22] Alan Pedrassoli Chitayat, Athanasios Kokkinakis, Sagarika Patra, Simon Demediuk, Justus Robertson, Oluseji Olarewaju, Marian Ursu, Ben Kirmann, Jonathan Hook, Florian Block, et al. Wards: Modelling the worth of vision in moba's. In *Intelligent Computing: Proceedings of the 2020 Computing Conference, Volume 2*, pages 63–81. Springer, 2020.

[23] Wei-Ta Chu and Yung-Chieh Chou. Event detection and highlight detection of broadcasted game videos. In *Proceedings of the 2nd Workshop on Computational Models of Social Interactions: Human-Computer-Media Communication*, pages 1–8, 2015.

[24] Lincoln Magalhães Costa, Anders Drachen, Francisco Carlos Monteiro Souza, and Geraldo Xexéo. Artificial intelligence in moba games: a multivocal literature mapping. *IEEE Transactions on Games*, 2023.

[25] Albert Cui, Howard Chung, and Nicholas Hanson-Holtry. Opendota blog: Faq, 2014. [Online]. Available: `https://blog.opendota.com/2014/08/01/faq` (accessed: 2022-01-20).

[26] David Cumming. Understanding the experience of australian esports spectatorship. In *DiGRA Conference*, 2018.

[27] Adele Cutler and Leo Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994.

[28] cyborgmatt. Dota 2 prize pool tracker, 2022. [Online]. Available: `https://dota2.prizetrac.kr/` (accessed: 2021-03-22).

[29] Simon Demediuk, Peter York, Anders Drachen, James Alfred Walker, and Florian Block. Role identification for accurate analysis in dota 2. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pages 130–138, 2019.

[30] Simon Demediuk, Athanasios Kokkinakis, Moni Sagarika Patra, Justus Robertson, Ben Kirman, Alistair Coates, Alan Chitayat, Jonathan Hook, Isabelle Nolle, Oluseyi Olarewaju, et al. Performance index: A new way to compare players. In *2021 MIT Sloan Sports Analytics Conference. IEEE*, 2021.

[31] Developer Community. Twitch developer documentation, 2020. [Online]. Available: `https://dev.twitch.tv/docs` (accessed: 2022-01-10).

[32] Carlos Dietrich, David Koop, Huy T. Vo, and Cláudio T. Silva. Baseball4D: A tool for baseball game reconstruction & visualization. In *2014 IEEE Conference on Visual Analytics Science and Technology*, VAST, pages 23–32, New York City, 2014. IEEE.

[33] Anders Drachen and Matthias Schubert. Spatial game analytics. In *Game Analytics*, pages 365–402. Springer, 2013.

[34] Minh Quan Duong, Bui Thi Minh Le Hong Lam, Giap Quang Huy Tu, and Nguyen Huu Hieu. Combination of k-mean clustering and elbow technique in mitigating losses of distribution network. *GMSARN International*, pages 153–158, 2019.

[35] Christoph Eggert, Marc Herrlich, Jan Smeddinck, and Rainer Malaka. Classification of player roles in the team-based multi-player game

dota 2. In *Entertainment Computing-ICEC 2015: 14th International Conference, ICEC 2015, Trondheim, Norway, September 29-Ocotober 2, 2015, Proceedings 14*, pages 112–125. Springer, 2015.

[36] Jessica Elam and Nick Taylor. Above the action: The cultural politics of watching dota 2. *Commun Cult Crit*, February 2020. doi: 10.1093/ccc/tcz033. URL `https://academic.oup.com/ccc/article-abstract/doi/10.1093/ccc/tcz033/5730340`.

[37] Jessica Formosa, Nicholas O'donnell, Ella M Horton, Selen Türkay, Regan L Mandryk, Michael Hawks, and Daniel Johnson. Definitions of esports: a systematic review and thematic analysis. *Proceedings of the ACM on Human-Computer Interaction*, 6(CHI PLAY):1–45, 2022.

[38] Gabriel Franco, Marcos Henrique Fonseca Ribeiro, and Giovanni Comarela. Towards an interpretable metric for dota 2 players: An unsupervised learning approach. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 341–346. IEEE, 2019.

[39] Bruno Duarte Abreu Freitas, Ruth Sofia Contreras-Espinosa, and Pedro Álvaro Pereira Correia. Sponsoring esports to improve brand image. *Scientific annals of economics and business*, 67(4):495–515, 2020.

[40] Cheng-Yang Fu, Joon Lee, Mohit Bansal, and Alexander C Berg. Video highlight prediction using audience chat reactions. *arXiv preprint arXiv:1707.08559*, 2017.

[41] Daniel C Funk, Anthony D Pizzo, and Bradley J Baker. esport management: Embracing esport education and research opportunities. *Sport Management Review*, 21(1):7–13, 2018.

[42] Fédération Internationale de Football Association. Fifa - tournaments, football news & live scores, 2021. [Online]. Available: `https://play.google.com/store/apps/details?id=com.fifa.fifaapp.android` (accessed: 2022-01-10).

[43] Joey Gawrysiak, Rick Burton, Seth Jenny, and Dylan Williams. Using esports efficiently to enhance and extend brand Perceptions–A literature review. *Physical Culture and Sport. Studies and Research*, 86(1):1–14, 2020. URL `https://content.sciendo.com/view/journals/pcssr/86/1/article-p1.xml`.

[44] Mehreen Gillani, Muhammad U. Ilyas, Saad Saleh, Jalal S. Alowibdi, Naif Aljohani, and Fahad S. Alotaibi. Post summarization of microblogs of sporting events. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, page 59–68, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN 9781450349147. doi: 10.1145/3041021.3054146. URL https://doi.org/10.1145/3041021.3054146.

[45] ESL Gaming GmbH. About us - ESL gaming GmbH, 2000. [Online]. Available: https://about.eslgaming.com/about-us (accessed: 2022-01-23).

[46] Barbara Göbl, Suzana Jovicic, Natalie Denk, and Simone Kriglstein. Let's play-professional views on barriers and potentials in digital gaming and e-sports. In *Extended Abstracts of the 2021 Annual Symposium on Computer-Human Interaction in Play*, pages 314–319, 2021.

[47] Google LLC. Google analytics, 2005. [Online]. Available: https://www.blog.google/products/marketingplatform/?_ga=2.249156423.1195102475.1579786011-1581655310.1579624784 (accessed: accessed: 2021-12-11).

[48] Google LLC. Youtube live streaming api overview, 2021. [Online]. Available: https://developers.google.com/youtube/v3/live/getting-started (accessed: 2022-01-20).

[49] Stephen Grossberg. Recurrent neural networks. *Scholarpedia*, 8(2): 1888, 2013.

[50] Jan Gulliksen, Bengt Göransson, Inger Boivie, Stefan Blomkvist, Jenny Persson, and Åsa Cajander. Key principles for user-centred systems design. *Behaviour and Information Technology*, 22(6): 397–409, 2003.

[51] Juan Guillermo López Guzmán and Cesar Julio Bustacara Medina. Relevant independent variables on moba video games to train machine learning algorithms. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 171–179, 2021.

[52] Juho Hamari and Max Sjöblom. What is esports and why do people watch it? *Internet Research*, 27(2):211–232, 2017.

[53] Emily Jane Hayday and Holly Collison. Exploring the contested notion of social inclusion and gender inclusivity within esport spaces. *Social Inclusion*, 8(3):197–208, 2020.

[54] Douglas Heaven. Rise and rise of esports. *New Scientist*, 223(2982): 17, August 2014.

[55] Laura Herrewijn and Sven Charleer. Power to the spectator: Towards an enhanced video game stream discovery experience. In *International Conference on Computer-Human Interaction Research and Applications*, pages 349–360. Springer, 2023.

[56] Laura Herrewijn and Sven Charleer. Spectator lf streamer: facilitating better stream discovery through spectator motivations and stream affordances. In *Companion Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pages 135–142, 2023.

[57] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.

[58] Victoria Hodge, Sam Devlin, Nick Sephton, Florian Block, Anders Drachen, and Peter Cowling. Win prediction in esports: Mixed-rank match prediction in multi-player online battle arena games, 2017. URL `https://arxiv.org/abs/1711.06498`.

[59] Victoria J Hodge, Sam Devlin, Nick Sephton, Florian Block, Peter I Cowling, and Anders Drachen. Win prediction in multiplayer esports: Live professional match prediction. *IEEE Transactions on Games*, 13 (4):368–379, 2019.

[60] Md Naimul Hoque and Klaus Mueller. Outcome-explorer: A causality guided interactive visual interface for interpretable algorithmic decision making. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4728–4740, 2022. doi: 10.1109/TVCG.2021.3102051.

[61] Thomas Horky and Philipp Pelka. Data visualisation in sports journalism: Opportunities and challenges of data-driven journalism in german football. *Digital Journalism*, 5(5):587–606, 2017.

[62] Hestry Humaira and R Rasyidah. Determining the appropiate cluster number using elbow method for k-means algorithm. In *Proceedings of*

the 2nd Workshop on Multidisciplinary and Applications (WMA) 2018, 24-25 January 2018, Padang, Indonesia, 2020.

[63] Brett Hutchins, Bo Li, and David Rowe. Over-the-top sport: live streaming services, changing coverage rights markets and the growth of media sport portals. *Media, Culture & Society*, 41:7:975–994, 2019.

[64] Seth E Jenny, Joey Gawrysiak, and Nicolas Besombes. Esports. edu: An inventory and analysis of global higher education esports academic programming and curricula. *International Journal of Esports*, 2021.

[65] Hyeon Jo and Seung-A Shin. Investigating viewer engagement in esports through motivation and attitudes toward metaverse and nfts. *Scientific Reports*, 14(1):19934, 2024.

[66] Filip Johansson and Jesper Wikström. Result prediction by mining replays in dota 2. Master's thesis, Blekinge Institute of Technology, 2015.

[67] Jonathan "PimpmuckL" Liebig. Introducing: Layerth, 2017. [Online]. Available: `https://medium.com/layerth/introducing-layerth-dead7ce11b13` (accessed: 2022-01-20).

[68] Kaushik Kalyanaraman. To win or not to win? a prediction model to determine the outcome of a dota2 match. *Technical Report. Technical report*, 2014.

[69] Seok-Kyu Kang and Jee-Hyong Lee. An e-sports video highlight generator using win-loss probability model. In *Proceedings of the 35th annual ACM symposium on applied computing*, pages 915–922, 2020.

[70] Adam Katona, Ryan Spick, Victoria J Hodge, Simon Demediuk, Florian Block, Anders Drachen, and James Alfred Walker. Time to die: Death prediction in dota 2 using deep learning. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2019.

[71] Cheng Hao Ke, Haozhang Deng, Congda Xu, Jiong Li, Xingyun Gu, Borchuluun Yadamsuren, Diego Klabjan, Rafet Sifa, Anders Drachen, and Simon Demediuk. Dota 2 match prediction through deep learning team fight models. In *2022 IEEE Conference on Games (CoG)*, pages 96–103. IEEE, 2022.

[72] Xiaobo Ke and Christian Wagner. What explains the next level of gaming? an experience-anticipation model for amateur esports participation. *Computers in Human Behavior*, 153:108125, 2024.

[73] Lucas Kempe-Cook, Stephen Tsung-Han Sher, and Norman Makoto Su. Behind the voices: The practice and challenges of esports casters. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[74] Nam Young Kim. The effect of advertising content control on advertising effectiveness in the different forced exposure circumstance. *Journal of Promotion Management*, 24(6):845–862, 2018.

[75] Se Jin Kim and Youngsun Sean Kim. Gender inequality in esport participation: Exploring the social process of women esport consumers. *Journal of Sport Behavior*, 45(4):79–107, 2022.

[76] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[77] Nicholas Kinkade and K Lim. Dota 2 win prediction. *Univ Calif*, 1: 1–13, 2015.

[78] Erica Kleinman, Nikitha Preetham, Zhaoqing Teng, Andy Bryant, and Magy Seif El-Nasr. "what happened here!?" a taxonomy for user interaction with spatio-temporal game data visualization. *Proceedings of the ACM on Human-Computer Interaction*, 5(CHI PLAY):1–27, 2021.

[79] Minsam Ko, Seung-woo Choi, Joonwon Lee, Subin Yang, Uichin Lee, Aviv Segev, and Junehwa Song. Motives for mass interactions in online sports viewing. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, page 329–330, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450327459. doi: 10.1145/2567948.2577340. URL `https://doi.org/10.1145/2567948.2577340`.

[80] Memduh Kocadağ. An esport research: psychological well-being differences of teenagers in terms of several variables. *Psychology Research on Education and Social Sciences*, 1(1):31–39, 2020.

[81] Athanasios Kokkinakis, Peter York, Moni Patra, Justus Robertson, Ben Kirman, Alistair Coates, Alan Pedrassoli Chitayat, Simon Peter Demediuk, Anders Drachen, Jonathan David Hook, et al.

Metagaming and metagames in esports. *International Journal of Esports*, 2021.

[82] Athanasios Vasileios Kokkinakis, Simon Demediuk, Isabelle Nölle, Oluseyi Olarewaju, Sagarika Patra, Justus Robertson, Peter York, Alan Pedrassoli Pedrassoli Chitayat, Alistair Coates, Daniel Slawson, et al. Dax: Data-driven audience experiences in esports. In *ACM International Conference on Interactive Media Experiences*, pages 94–105, 2020.

[83] Simone Kriglstein, Günter Wallner, Sven Charleer, Kathrin Gerling, Pejman Mirza-Babaei, Steven Schirra, and Manfred Tscheligi. Be part of it: Spectator experience in gaming and esports. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2020.

[84] Klaus Krippendorff. *Content analysis: An introduction to its methodology*. Sage publications, 2018.

[85] Haosen Lai, Jiong Yu, Shuoxin Wang, Dawei Zhang, Sixing Wu, and Wei Zhou. Enhancing moba game commentary generation with fine-grained prototype retrieval. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 850–862. Springer, 2023.

[86] Bjarke Alexander Larsen and Elin Carstensdottir. Community, storytelling, and play: Making and breaking rituals in destiny 2. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2024.

[87] Donghun Lee and Linda J Schoenstedt. Comparison of esports and traditional sports consumption motives. *ICHPER-SD Journal Of Research*, 6(2):39–44, 2011.

[88] Pascal Lessel, Maximilian Altmeyer, and Antonio Krüger. Viewers' perception of elements used in game live-streams. In *Proceedings of the 22nd International Academic Mindtrek Conference*, pages 59–68, 2018.

[89] Leo Lewis and Tim Bradshaw. Esports: Is the gaming business ready to come of age? *Financial Times*, November 2017.

[90] Jierui Li, Lemao Liu, Huayang Li, Guanlin Li, Guoping Huang, and Shuming Shi. Evaluating explanation methods for neural machine translation. *arXiv preprint arXiv:2005.01672*, 2020.

[91] Lingyuan Li, Jirassaya Uttarapong, Guo Freeman, and Donghee Yvette Wohn. Spontaneous, yet studious: Esports commentators' live performance and self-presentation practices. *Proceedings of the ACM on Human-Computer Interaction*, 4 (CSCW2):1–25, 2020.

[92] Brian Y Lim and Anind K Dey. Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 195–204, 2009.

[93] Brian Y Lim, Anind K Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2119–2128, 2009.

[94] Wenli Looi, Manmeet Dhaliwal, Reda Alhajj, and Jon Rokne. Recommender system for items in dota 2. *IEEE Transactions on Games*, 11(4):396–404, 2018.

[95] Zijin Luo, Matthew Guzdial, and Mark Riedl. Making cnns for video parsing accessible: event extraction from dota2 gameplay video using transfer, zero-shot, and network pruning. In *Proceedings of the 14th international conference on the foundations of digital games*, pages 1–10, 2019.

[96] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013.

[97] Ilya Makarov, Dmitry Savostyanov, Boris Litvyakov, and Dmitry I Ignatov. Predicting winning team and probabilistic ratings in "dota 2" and "counter-strike: Global offensive" video games. In *Analysis of Images, Social Networks and Texts: 6th International Conference, AIST 2017, Moscow, Russia, July 27–29, 2017, Revised Selected Papers 6*, pages 183–196. Springer, 2018.

[98] DL S Mamoru, AD Panditha, WASS J Perera, and GU Ganegoda. Automated commentary generation based on fps gameplay analysis. In *2022 7th International Conference on Information Technology Research (ICITR)*, pages 1–5. IEEE, 2022.

[99] Valerio Mancini, Tommaso Marazzi, and Alessio Postiglione. Sport as a global business: Esports, crypto, nft and metaverse. *The Market*, 4: 8–144, 2023.

[100] George Margetis, Stavroula Ntoa, Margherita Antona, and Constantine Stephanidis. Human-centered design of artificial intelligence. *Handbook of human factors and ergonomics*, pages 1085–1106, 2021.

[101] George Margetis, Konstantinos C Apostolakis, Stavroula Ntoa, Ioannis Chatzakis, Eirini Sykianaki, Ioannis Markopoulos, and Constantine Stephanidis. Visual summarisations for computer-assisted live color casting and direction in league of legends. In *International Conference on Human-Computer Interaction*, pages 133–153. Springer, 2023.

[102] Stefan Marshall, Paris Mavromoustakos-Blom, and Pieter Spronck. Enabling real-time prediction of in-game deaths through telemetry in counter-strike: Global offensive. In *Proceedings of the 17th international conference on the foundations of digital games*, pages 1–10, 2022.

[103] Philip Z Maymin. Smart kills and worthless deaths: esports analytics for league of legends. *Journal of Quantitative Analysis in Sports*, 17 (1):11–27, 2021.

[104] Brian McCauley, Kieran Tierney, and Dinara Tokbaeva. Shaping a regional offline esports market: understanding how jönköping, the ‘city of dreamhack’, takes url to irl. *International Journal on Media Management*, 22(1):30–48, 2020.

[105] Zixiang Meng, Qiang Gao, Di Guo, Yunlong Li, Bobo Li, Hao Fei, Shengqiong Wu, Fei Li, Chong Teng, and Donghong Ji. Mmlscu: A dataset for multi-modal multi-domain live streaming comment understanding. In *Proceedings of the ACM on Web Conference 2024*, pages 4395–4406, 2024.

[106] Yue Meng-Lewis, Gavin Lewis, Zhi Lin, and Yupei Zhao. Examination of esports fans’ live streaming experiences using an extended expectation-confirmation model: A case study of the king pro league. *International Journal of Human–Computer Interaction*, pages 1–16, 2022.

[107] Andrew Meola. Here are the most popular esport games of 2017 and the companies behind them. *Business Insider*, 2017. URL `http://www.businessinsider.com/top-esports-games-developers-2017-11`. Accessed: 2021-12-18.

[108] Meta. Live video api, 2018. [Online]. Available: `https://developers.facebook.com/docs/live-video-api/getting-started/` (accessed: 2022-01-20).

[109] Marcal Mora-Cantallops and Miguel-Ángel Sicilia. Team efficiency and network structure: The case of professional league of legends. *Social Networks*, 58:105–115, 2019.

[110] Tomoko Nemoto and David Beglar. Likert-scale questionnaires. In *JALT 2013 conference proceedings*, volume 108, pages 1–6, 2014.

[111] Newzoo. Newzoo global esports market report 2019, 2019. [Online]. Available: `https://newzoo.com/insights/trend-reports/newzoo-global-esports-market-report-2019-light-version/` (accessed: 2024-10-27).

[112] Newzoo. Newzoo global esports market report 2020: Light version. *Newzoo, United Kingdom*, March 2021. URL `https://newzoo.com/insights/trend-reports/newzoo-global-esports-market-report-2020-light-version/`. Accessed: 2022-02-09.

[113] Newzoo. Newzoo's global esports & live streaming market report 2022 | free version, 2022. [Online]. Available: `https://newzoo.com/resources/trend-reports/newzoo-global-esports-live-streaming-market-report-2022-free-version` (accessed: 2024-10-27).

[114] Newzoo. Newzoo's global games market report 2023 | may 2024 update, 2024. [Online]. Available: `https://newzoo.com/resources/trend-reports/newzoo-global-games-market-report-2023-free-version` (accessed: 2024-10-27).

[115] NFL Enterprises LLC. Nfl, 2020. [Online]. Available: `https://play.google.com/store/apps/details?id=com.gotv.nflgamecenter.us.lite` (accessed: 2021-12-11.

[116] Nielsen. Esports handbook for brands. *Nielsen, United Kingdom*, 2019. URL `https://www.nielsen.com/wp-content/uploads/sites/3/2019/05/esports-playbook-for-brands-2019.pdf`. Accessed: 2022-01-19.

[117] Chollakorn Nimpattanavong, Pittawat Taveekitworachai, Ibrahim Khan, Thai Van Nguyen, Ruck Thawonmas, Worawat Choensawat, and Kingkarn Sookhanaphibarn. Am i fighting well? fighting game commentary generation with chatgpt. In *Proceedings of the 13th International Conference on Advances in Information Technology*, pages 1–7, 2023.

[118] Sam Nordmark and Jerome Heath. The top 10 highest prize pools in esports, 2019. [Online]. Available: `https://dotesports.com/general/news/biggest-prize-pools-esports-14605` (accessed: 2022-01-21).

[119] Oluseyi Olarewaju, Athanasios V Kokkinakis, Simon Demediuk, Justus Roberstson, Isabelle Nölle, Sagarika Patra, Daniel Slawson, Alan P Chitayat, Alistair Coates, Ben Kirman, et al. Automatic generation of text for match recaps using esport caster commentaries. In *CS & IT Conference Proceedings*, volume 10. CS & IT Conference Proceedings, 2020.

[120] Kieun Park, Hajin Lim, Joonhwan Lee, and Bongwon Suh. Enhancing auto-generated baseball highlights via win probability and bias injection method. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2024.

[121] Max Pashkevich, Sundar Dorai-Raj, Melanie Kellar, and Dan Zigmond. Empowering online advertisements by empowering viewers with the right to choose: the relative effectiveness of skippable video advertisements on youtube. *Journal of advertising research*, 52(4): 451–457, 2012.

[122] Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(4):675–685, 2000. doi: 10.1017/S0266466603004109.

[123] Alan Pedrassoli Chitayat, Florian Block, James Walker, and Anders Drachen. Beyond the meta: Leveraging game design parameters for patch-agnostic esport analitics. In *Proceedings of the AAAI*

Conference on Artificial Intelligence and Interactive Digital Entertainment, pages 116–125, 2023.

[124] Alan Pedrassoli Chitayat, Florian Block, James Alfred Walker, and Anders Drachen. Applying and visualising complex models in esport broadcast coverage. In *Proceedings of the 2024 ACM International Conference on Interactive Media Experiences*, pages 108–116, 2024.

[125] Alan Pedrassoli Chitayat, Florian Oliver Block, James Alfred Walker, and Anders Drachen. How could they win? an exploration of win condition for esports narratives. In *Annual Symposium on Computer-Human Interaction in Play (CHI PLAY 2024)*. ACM, 2024.

[126] Alan Pedrassoli Chitayat, Alistair Coates, Florian Block, Anders Drachen, James Alfred Walker, James Dean, Mark McConachie, and Peter York. From passive viewer to active fan: Towards the design and large-scale evaluation of interactive audience experiences in esports and beyond. In *Proceedings of the 2024 ACM International Conference on Interactive Media Experiences*, pages 94–107, 2024.

[127] Sean Penney, Jonathan Dodge, Andrew Anderson, Claudia Hilderbrand, Logan Simpson, and Margaret Burnett. The shoutcasters, the game enthusiasts, and the ai: Foraging for explanations of real-time strategy players. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 11(1):1–46, 2021.

[128] Elena Petrovskaya and David Zendle. The battle pass: A mixed-methods investigation into a growing type of video game monetisation. *OSF Preprints, Sep*, 2020.

[129] Johannes Pfau and Magy Seif El-Nasr. On video game balancing: Joining player-and data-driven analytics. *ACM Games: Research and Practice*, 2(3):1–30, 2024.

[130] G Pingali, A Opalach, Y Jean, and I Carlbom. Visualization of sports using motion trajectories: providing insights into performance, style, and strategy. In *Proceedings Visualization, 2001. VIS '01.*, pages 75–544, October 2001.

[131] Anthony D Pizzo, Sangwon Na, Doohan Kim, Kostas Alexandris, and Moonsup Hyun. Esports gender diversity: A leisure constraints perspective. *Journal of Leisure Research*, 54(5):602–623, 2023.

[132] PwC. The burgeoning evolution of esports from the fringes to front and center, 2016. [Online]. Available: https://www.pwc.se/sv/pdf-reports/the-burgeoning-evolution-of-esports.pdf (accessed: 2021-12-21).

[133] Tyreal Yizhou Qian, Jerred Junqi Wang, James Jianhui Zhang, and Laura Zhenqiu Lu. It is in the game: dimensions of esports online spectator motivation and development of a scale. *European sport management quarterly*, 20(4):458–479, 2020.

[134] Xin Qian, Rafet Sifa, Xuefei Liu, Shreyashi Ganguly, Borchuluun Yadamsuren, Diego Klabjan, Anders Drachen, and Simon Demediuk. Anomaly detection in player performances in multiplayer online battle arena games. In *Proceedings of the 2022 Australasian Computer Science Week*, ACSW '22, page 23–30, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450396066. doi: 10.1145/3511616.3513095. URL https://doi.org/10.1145/3511616.3513095.

[135] Jana Rambusch, Anna-Sofia Alklind Taylor, and Tarja Susi. A pre-study on spectatorship in esports. In *Spectating Play 13th Annual Game Research Lab Spring Seminar, Tampere, Finland, April 24-25, 2017*, 2017.

[136] Jason G. Reitman, Maria J. Anderson-Coto, Minerva Wu, Je Seok Lee, and Constance Steinkuehler. Esports research: A literature review. *Games and Culture*, 15(1):32–50, April 2019.

[137] Jason G Reitman, Maria J Anderson-Coto, Minerva Wu, Je Seok Lee, and Constance Steinkuehler. Esports research: A literature review. *Games and Culture*, 15(1):32–50, January 2020. ISSN 1555-4120. doi: 10.1177/1555412019840892. URL https://doi.org/10.1177/1555412019840892.

[138] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[139] Frans Rijnders, Günter Wallner, and Regina Bernhaupt. Live feedback for training through real-time data visualizations: A study with league of legends. *Proceedings of the ACM on Human-Computer Interaction*, 6(CHI PLAY):1–23, 2022.

[140] Charles Ringer, Mihalis A Nicolaou, and James Alfred Walker. Autohighlight: Highlight detection in league of legends esports broadcasts via crowd-sourced data. *Machine Learning with Applications*, 9:100338, 2022.

[141] Charles Ringer, Sondess Missaoui, Victoria J Hodge, Alan Pedrassoli Chitayat, Athanasios Kokkinakis, Sagarika Patra, Simon Demediuk, Alvaro Caceres Munoz, Oluseji Olarewaju, Marian Ursu, et al. Time to die 2: Improved in-game death prediction in dota 2. *Machine Learning with Applications*, page 100466, 2023.

[142] Justus Robertson, Athanasios Vasileios Kokkinakis, Jonathan Hook, Ben Kirman, Florian Block, Marian F Ursu, Sagarika Patra, Simon Demediuk, Anders Drachen, and Oluseyi Olarewaju. Wait, but why?: assessing behavior explanation strategies for real-time strategy games. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*, pages 32–42, 2021.

[143] Ryan Rogers, Lee Farquhar, and Jacob Mummert. Audience response to endemic and non-endemic sponsors of esports events. *International Journal of Sports Marketing and Sponsorship*, 2020.

[144] Yoonji Ryu, Hui Hwang, Jaehyun Jeong, Wonseok Jang, Gyemin Lee, and Hyunwoong Pyun. Missed a live match? determinants of league of legends champions korea highlights viewership. *Frontiers in Psychology*, 14:1213600, 2023.

[145] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017.

[146] Matthias Schubert, Anders Drachen, and Tobias Mahlmann. Esports analytics through encounter detection. In *10th MIT Sloan Sports Analytics Conference*, 2016.

[147] Aleksandr Semenov, Peter Romov, Sergey Korolev, Daniil Yashkov, and Kirill Neklyudov. Performance of machine learning algorithms in predicting game outcome from drafts in DOTA 2. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 26–37. Springer, 2016.

[148] Tali Sharot and Elizabeth A Phelps. How arousal modulates memory: Disentangling the effects of attention and retention. *Cognitive, Affective, & Behavioral Neuroscience*, 4(3):294–306, 2004.

[149] Rafet Sifa, Anders Drachen, Florian Block, Spencer Moon, Anisha Dubhashi, Hao Xiao, Zili Li, Diego Klabjan, and Simon Demediuk. Archetypal analysis based anomaly detection for improved storytelling in multiplayer online battle arena games. In *Proceedings of the 2021 Australasian Computer Science Week Multiconference*, pages 1–8, 2021.

[150] Max Sjöblom and Juho Hamari. Why do people watch others play video games? an empirical study on the motivations of twitch users. *Comput. Human Behav.*, 75:985–996, October 2017.

[151] Anton Smerdov, Andrey Somov, Evgeny Burnaev, Bo Zhou, and Paul Lukowicz. Detecting video game player burnout with the use of sensor data and machine learning. *IEEE Internet of Things Journal*, 8(22):16680–16691, 2021.

[152] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[153] SofaScore. Sofascore - live scores, fixtures & standings, 2020. [Online]. Available: `https://play.google.com/store/apps/details?id=com.sofascore.results` (accessed: 2021-12-11).

[154] Kuangyan Song, Tianyi Zhang, and Chao Ma. Predicting the winning side of dota2. *Sl: sn*, 2015.

[155] spheenik. Clarity analyzer. `https://github.com/spheenik/clarity-analyzer`, 2015.

[156] PGL Esports srl. Pgl, 2002. [Online]. Available: `http://www.pglesports.com/` (accessed: 2022-01-20).

[157] Samantha Stahlke, James Robb, and Pejman Mirza-Babaei. The fall of the fourth wall: Designing and evaluating interactive spectator experiences. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)*, 10(1):42–62, 2018.

[158] Statista. Most played games on steam as of december 2021, by average number of concurrent players. *Statista, Germany*, December

2021. URL https://www.statista.com/statistics/1179973/steam-games-peak-concurrent-players/. Accessed: 2022-01-30.

[159] Statista. Leading esports games worldwide in 2021, by cumulative tournament prize pool. *Statista, Germany*, February 2022. URL https://www.statista.com/statistics/501853/leading-esports-games-worldwide-total-prize-pool/. Accessed: 2022-02-09.

[160] Constance Steinkuehler. Esports research: Critical, empirical, and historical studies of competitive videogame play. *Games and Culture*, 15(1):3–8, 2020.

[161] Adam Summerville, Michael Cook, and Ben Steenhuisen. Draft-analysis of the ancients: predicting draft picks in dota 2 using machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 12, pages 100–106, 2016.

[162] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[163] Stefan Szymanski. Sport analytics: Science or alchemy? *Kinesiology Review*, 9(1):57–63, 2020.

[164] Lizzie Tantam. European esports market worth €240 million, with 86 million fans across europe. *Deloitte United Kingdom*, August 2019. URL https://www2.deloitte.com/uk/en/pages/press-releases/articles/european-esports-market-worth-240-million-euros-with-86-million-fans-acr html. Accessed: 2020-12-20.

[165] TL Taylor. *Watch me play: Twitch and the rise of game live streaming*. Princeton University Press, 2018.

[166] Sunny Thaicharoen, Jeremy Gow, and Anders Drachen. An ecosystem framework for the meta in esport games. *Journal of Electronic Gaming and Esports*, 1(1):jege.2022–0045, 2023. doi: 10.1123/jege.2022-0045.

[167] TLS Media AS. Tls football – premier live opta stats 2019/2020, 2020. [Online]. Available: https:

//play.google.com/store/apps/details?id=com.toplivestats (accessed: 2020-01-16).

[168] Marko Tot, Michelangelo Conserva, Alan Pedrassoli Chitayat, Athanasios Kokkinakis, Sagarika Patra, Simon Demediuk, Alvaro Caceres Munoz, Oluseji Olarewaju, Marian Ursu, Ben Kirmann, et al. What are you looking at? team fight prediction through player camera. In *2021 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2021.

[169] Riikka Turtiainen, Usva Friman, and Maria Ruotsalainen. "not only for a celebration of competitive overwatch but also for national pride": Sportificating the overwatch world cup 2016. *Games and Culture*, 15(4):351–371, June 2020. ISSN 1555-4120. doi: 10.1177/1555412018795791. URL https://doi.org/10.1177/1555412018795791.

[170] Karl Tuyls, Shayegan Omidshafiei, Paul Muller, Zhe Wang, Jerome Connor, Daniel Hennes, Ian Graham, William Spearman, Tim Waskett, Dafydd Steel, et al. Game plan: What ai can do for football, and what football can do for ai. *Journal of Artificial Intelligence Research*, 71:41–88, 2021.

[171] Tanja Välisalo and Maria Ruotsalainen. Esports fandom in the age of transmedia: the reception of the overwatch league. *Eracle. Journal of Sport and Social Sciences*, 4(1):11–35, 2021.

[172] J L van Ditmarsch. Video games as a spectator sport. Master's thesis, Utrecht University, 2013. Available at: https://studenttheses.uu.nl/handle/20.500.12932/15595.

[173] Stan JP van Kempen, Erik D Van der Spek, and Günter Wallner. The importance of dashboard elements during esports matches to players, passive-viewers and spectator-players. In *International Conference on Entertainment Computing*, pages 283–295. Springer, 2022.

[174] Markos Viggiato and Cor-Paul Bezemer. Trouncing in dota 2: An investigation of blowout matches. In *Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment*, volume 16, pages 294–300, 2020.

[175] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard

Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575 (7782):350–354, 2019.

[176] Günter Wallne. Enhancing battle maps through flow graphs. In *2019 IEEE Conference on Games (CoG)*, pages 1–4. IEEE, 2019.

[177] Guenter Wallner and Anders Drachen. Visualization of player movement patterns with line integral convolution and alpha shapes. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*, pages 1–10, 2024.

[178] Günter Wallner, Marnix Van Wijland, Regina Bernhaupt, and Simone Kriglstein. What players want: information needs of players on post-game visualizations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2021.

[179] Minglei Wang. The communication effectiveness of ai win prediction applied in esports live streaming: A pilot study. In *Computer-Human Interaction Research and Applications*, pages 315–325, Cham, 2023. Springer Nature Switzerland. doi: 10.1007/978-3-031-49368-3_19.

[180] Weiqi Wang. Predicting multiplayer online battle arena (moba) game outcome based on hero draft data. Master's thesis, Dublin, National College of Ireland, 2016.

[181] Zihan Wang and Naoki Yoshinaga. Commentary generation from data records of multiplayer strategy esports game. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 263–271, 2024.

[182] Ken Watanabe, Naoki Saijo, Sorato Minami, and Makio Kashino. The effects of competitive and interactive play on physiological state in professional esports players. *Heliyon*, 7(4), 2021.

[183] Andrew Webster. Dota 2 tournament the international 2019 surpasses $30 million prize pool, topping fortnite, 2019. [Online]. Available: `https://www.theverge.com/2019/7/22/20703528/dota-2-the-international-9-prize-pool-battle-pass-fortnite` (accessed: 2021-12-11).

[184] Thomas Weiss and Sabrina Schiele. Virtual worlds in competitive contexts: Analyzing esports consumer needs. *Electronic Markets*, 23 (4):307–316, 2013.

[185] Jianxin Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5 (23):495, 2017.

[186] Peter Xenopoulos, João Rulff, and Claudio Silva. Ggviz: Accelerating large-scale esports game analysis. *Proceedings of the ACM on Human-Computer Interaction*, 6(CHI PLAY):1–22, 2022.

[187] Junjie H Xu, Hong Huang, Xiaoling Ling, and Pujana Paliyawan. Toward collaborative game commentating utilizing pre-trained generative language models. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4. IEEE, 2022.

[188] Junjie H Xu, Yu Nakano, Lingrong Kong, and Kojiro Iizuka. Cs-lol: a dataset of viewer comment with scene in e-sports live-streaming. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, pages 422–426, 2023.

[189] Pu Yang, Brent E Harrison, and David L Roberts. Identifying patterns in combat that are predictive of success in moba games. In *FDG*, 2014.

[190] Yifan Yang, Tian Qin, and Yu-Heng Lei. Real-time esports match result prediction. *arXiv preprint arXiv:1701.03162*, 2016.

[191] Zelong Yang, Zhufeng Pan, Yan Wang, Deng Cai, Shuming Shi, Shao-Lun Huang, Wei Bi, and Xiaojiang Liu. Interpretable real-time win prediction for honor of kings—a popular mobile moba esport. *IEEE Transactions on Games*, 14(4):589–597, 2022. doi: 10.1109/TG.2022.3149044.

[192] Zelong Yang, Yan Wang, Piji Li, Shaobin Lin, Shuming Shi, Shao-Lun Huang, and Wei Bi. Predicting events in moba games: Prediction, attribution, and evaluation. *IEEE Transactions on Games*, 15(2):193–201, 2022.

[193] Jhen-Ni Ye, Jian-Hong Ye, Chih-Mei Wang, and Jon-Chao Hong. Development of 5 cs educational value scale for esport games. *International Journal of Technology in Education and Science*, 5(3): 362–374, 2021.

[194] Taiki Yoshida and Tomomi Ogawa. Highlight generation method for game commentary using acoustic features. In *2023 IEEE 3rd International Conference on Software Engineering and Artificial Intelligence (SEAI)*, pages 266–270. IEEE, 2023.

[195] Wen Bo Yu, Nicole A Beres, Raquel Breejon Robinson, Madison Klarkowski, and Pejman Mirza-Babaei. Exploring esports spectator motivations. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–6, 2022.

[196] Xinguo Yu and Dirk Farin. Current and emerging topics in sports video processing. In *International Conference on Multimedia and Expo*, pages 526–529. IEEE, 2005.

[197] Han Yue, Hongfu Liu, and Jian Chen. A gospel for moba game: ranking-preserved hero change prediction in dota 2. *IEEE transactions on games*, 14(2):191–201, 2021.

[198] Lei Zhang, Chenbo Xu, Yihua Gao, Yi Han, Xiaojiang Du, and Zhihong Tian. Improved dota2 lineup recommendation model based on a bidirectional lstm. *Tsinghua Science and Technology*, 25(6): 712–720, 2020.