Representative Features for Game Agnostic Movement Evaluation (ReFGAME):
Extending a Trajectory Analysis Framework from Human Mobility to Video Games

by

Reia Mendell Drucker
B.Sc., University of Central Florida, 2022

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

We acknowledge and respect the Lək̓ʷəŋən (Songhees and Xʷsepsəm / Esquimalt) Peoples on whose territory the university stands, and the Lək̓ʷəŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Representative Features for Game Agnostic Movement Evaluation (ReFGAME):
Extending a Trajectory Analysis Framework from Human Mobility to Video Games

by

Reia Mendell Drucker
B.Sc., University of Central Florida, 2022

Supervisory Committee

Dr. Kevin Stanley, Supervisor
(Department of Computer Science)

Dr. Regan Mandryk, Departmental Member
(Department of Computer Science)

Dr. Brandon Haworth, Departmental Member
(Department of Computer Science)

ABSTRACT

Understanding how players interact with virtual environments underpins the design of video games and the contextualization of player behaviour. This work extends a framework from human mobility research to the analysis of player movement and camera orientation trajectories in video games. Using features that capture spatiotemporal properties of trajectories, this work demonstrates the framework's applicability in a gaming context through supervised and unsupervised machine learning tasks conducted on data from professional Counter-Strike: Global Offensive matches. The framework can be used to distinguish between the gameplay environment and side of play of teams with 93% accuracy when using features derived from player movement and 97% accuracy with the addition of camera orientation derived features. The framework reveals design archetypes between environments and potentially between the roles of players. Findings validate the generalizability of spatial mobility features to a gaming context, and highlight their potential for applications in role identification, environment design, anomaly detection, and cross-game analysis.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

my cats, friends, and family, for their support during my studies.

Dr. Kevin Stanley, for their mentorship, support, and many opportunities to grow.

Canada, for a place to pursue my studies free from discrimination.

However ruined this world has become,
however mired in torment and despair,
life endures.
Births continue.
There is beauty in that, is there not?

DEDICATION

I dedicate this work to those gone hollow.

Chapter 1

Introduction

Virtual environments, particularly those found in video games, can generate log files that record precise information regarding the state of the environment and the actions of individuals within them. These log files provide an extensive and high fidelity data source for quantitative analysis on how players navigate, interact with, and perceive digital spaces [1–4]. The spatial behaviour of player controlled agents, cameras, and input devices contained within these logs reflect player experience and in game interaction and support a wide array of objectives including: identifying play patterns [5], detecting non-human agents [6, 7], and detecting cheaters [8–10].

Existing approaches in player behaviour analysis often rely on bespoke game-specific metrics tailored to particular genres or research objectives, typically in the form of game-specific action frequencies or social activity metrics [4, 11, 12]. While these specialized metrics yield valuable insight into specific dynamics of player behaviour, the lack of standardization and generalizability of these metrics poses a challenge for comparative analysis across titles, genres, or even within the same game following significant updates. Although prior works have analyzed spatial behaviour in games [5,7], there remains a need for a comprehensive and broadly applicable analytical framework suitable for virtual environments.

Patterns of human movement have been extensively studied, forming strong foundations for diverse applications in fields such as epidemiological modelling, urban planning, and economic policy [13–15]. In recent years, the proliferation of GPS-enabled devices has significantly advanced research into human mobility, facilitating the collection of large volumes of high resolution geospatial data [16]. This surge in both the quantity and quality of available data has driven a methodological shift within the field [17], allowing for greater granularity and precision in analyzing and modelling human movement. Many of the assumptions underlying human mobility in the real world, such as the presence of spatial constraints, goal directed movement, and emergent patterns [18, 19], also hold in structured virtual environments. By treating player controlled agents within virtual environments as a spatiotemporal process, metrics that aptly capture properties of

real-world trajectories can be repurposed to provide insight into player behaviour in games.

To validate this approach, I extended a human mobility framework (ReFGeM) [17] through novel features that build upon the core philosophy of the original framework, enhancing the framework's ability to characterize spatiotemporal behaviours while maintaining its generality and interpretability, and applied it to data collected from professional matches of Counter-Strike: Global Offensive (CSGO). I demonstrated the modified framework's capability to capture differences in player behaviour across environments through various machine learning tasks where I classified the map and side of play of a team using the framework's metrics to derive feature vectors from player movement and camera orientation trajectories. I showcased the utility of this approach in distinguishing between the environment and roles of teams, explored an application of the modified framework in distinguishing between within-team roles of players, and discuss the similarities between real-world and virtual movement. I present here a novel method of quantifying spatiotemporal behaviour in games, evidence of the approach's utility, and directions for future research. The key findings and claims of this work are:

1. Approaches to analyzing and modelling human mobility in the real-world using metrics that capture the spatiotemporal properties of trajectories provide utility in virtual environments.

2. Trajectories of player controlled agents and cameras in virtual environments have similar characteristics to real-world movement trajectories and provide meaningful information regarding the environment and roles of players.

3. It is possible to distinguish between the gameplay environment and role of teams in CSGO using features derived from their movement trajectories with high accuracy and using additional features derived from their camera trajectories can improve accuracy further.

4. These features can be used to reveal design archetypes between gameplay environments and potentially the roles of players warranting further investigation into potential applications for role-identification, environment design, anomaly detection, and cross-game analysis.

In highlighting the shared characteristics of virtual and real-world mobility, this work demonstrates that the study of player behaviour in games can draw on established techniques and findings from the human mobility literature to quantify and interpret patterns in virtual environments. In bridging these domains, this work provides a new perspective that can enrich existing approaches to player behaviour analysis and suggests that the insights provided by understanding how games are played can inform our understanding of real-world behaviour as well.

Chapter 2

Background

2.1 CSGO Game Overview

CSGO is tactical first person shooter (FPS) video game where two opposing teams of five players compete. A game of CSGO consists of up to 30 rounds on a specific map, in which the first team to win 16 rounds wins the game. In the event of a tie, additional rounds are played until a two round lead is achieved. Initially, teams play as either the Terrorist (T) or Counter-Terrorist (CT) side, and after 15 rounds the teams switch. A standard game is typically played over two halves of up to 15 rounds each, with each round having a time limit of 1 minute and 55 seconds.

To win a round a team must either eliminate all players on the opposing side or achieve a side specific objective: planting and defending a bomb in one of the two designated sites on a map for the T side, or defusing the bomb in the 35 seconds before it explodes for the CT side. To accomplish these objectives teams must manage in-game money — awarded to players based on their performances in prior rounds — to purchase guns, grenades, and armour [20].

Typical gameplay involves players on the T side exploring a map to find weak points in the CT defence and gain control of the map. Common play patterns on the T side involve taking space, exploring the map for information, and providing utility to teammates in the form of grenades. On the CT side players are typically more passive, aiming to control space by holding and fighting for key positions. Common play patterns on the CT side involve remaining in key locations for extended periods to preserve defensive structure and transitioning to help reinforce key positions. When moving players can choose to run and make noise or walk to not reveal their positions [21].

CSGO is a popular esports title which fosters competition and incentivizes players to adopt and converge upon the most effective strategies and gameplay patterns. The increased incentive to win encourages professional players to specialize into roles within a team, minimize inefficiencies in both their movement and aim, and generally results in gameplay that is more structured and refined compared to casual play due to increased planning and coordination. In professional play

the convention for team structure, and therefore player roles, is roughly organized as follows [22,23]:

- **Rifler/Entry** - Primarily combat focused, responsible for aggressively taking space and securing the first kill in a round.
- **Anchor** - Primarily positioning focused, responsible for passively and patiently holding space to gather information for the team and maintaining defensive structure.
- **Support/Flex** - Primarily utility focused, responsible for managing grenades, following up on plays made by the Rifler/Entry, and filling gaps in team structure.
- **Lurker** - Primarily scouting focused, responsible for methodical individual plays involving exploration to surprise enemies and gathering information for the team.
- **Sniper/AWPer** - Flexible combat focus, responsible for passively controlling large open spaces, aggressively positioning in unexpected areas, and extracting the specialized value the powerful AWP sniper rifle can provide over other weapons.

2.2 Dataset Overview

This work makes use of the Esports Trajectories & Action (ESTA) dataset [24], a publicly available dataset of parsed CSGO game logs originally published to serve as a benchmark for win probability prediction. It consists of an Online subset containing 878 logs from matches in top Online tournaments between January 2021 and May 2022 and a LAN subset containing 680 logs from matches in top LAN tournaments between July 2021 and May 2022. The dataset does not contain the original game action logs, called demofiles, but rather a pre-processed version of the logs stored as JavaScript Object Notation (JSON) files. These JSON files contain spatiotemporal data on player actions such as damages, kills, grenade throws, bomb plants/defuses, flashes, and weapon fires that occurred during a match. They also reduce the resolution of the logs from 64 Hz or 128 Hz to 2 Hz. Only the LAN subset was used in this work to reduce the effect of network related issues in the data and to ensure all matches were played on a server operating at a 128 Hz tick rate.

The 680 games of CSGO used in this work contain data from 18,338 rounds played by 267 different players organized into 57 different teams. The nature of professional tournament play results in players each participating in a different numbers of games, rounds, and occasionally on more than one team due to the trading of players between organizations. The total duration of individual rounds can also vary, with a player's participation in a round depending on how long they stay alive in each round. Games can be played on any of the seven different available maps, with some maps being played more frequently than others. Depending on which side a team starts on some players

may play more rounds on the CT side than on the T side. While variability in the dataset provides many confounding factors for analysis that cannot be fully mitigated, aggregation at the team level, stratification by match, map, and side, and the filtering out of rounds that have insufficient duration to produce numerically stabile trajectory features, help to control for this variability and ensure that meaningful conclusions can still be drawn regarding player movement dynamics. Further details and methodological justifications are provided in Chapter 4.

2.3 Feature Set

2.3.1 Original Feature Set

This work adapts and extends the work of [17] where the authors presented a standardized feature set for differentiating population spatial behaviour using Representative Features of Geospatial Mobility (ReFGeM). There the authors proposed that features or metrics used to distinguish patterns in spatial behaviour should be phenomenologically representative, mathematically rigorous, provide utility in distinguishing sub-populations, and most importantly be generalizable. Here generalizability is the most selective requirement, as many spatial metrics are susceptible to the Modifiable Areal Unit Problem [25] where different discretizations of the space and sampling frequency affect the value computed for the metric. As not all of the features proposed in [17] are relevant in a gaming context, a full background is provided only for those included in this work. The features of trajectories in [17] excluded from this work are the buffered area (BA) and the convex hull comprised of the ten locations with the longest dwell time (CH10). These features both require arbitrary choices for their parameters and the fidelity of game logs removes the need to buffer the area along a trajectory to compensate for sensor related issues present in GPS trajectory data sets.

2.3.2 Multi-scale Entropy Rate Constants

When working with spatial-temporal data, a common approach involves discretizing the space into grid cells with unique identifiers. With this spatial discretization and a sampling rate, it is possible to construct a sequence of symbols corresponding to the grid cells traversed by a given trajectory. Denoting this sequence of symbols as a visit string was a key methodological insight of [26]. The entropy of a string is a measure of its predictability and regularity and can be approximated by the string's compressibility. For a trajectory the same metric can be computed on the corresponding visit string. Formally, for any string S with a unique terminating character the Lempel-Ziv78 (LZ) derived entropy rate H of string S is given by:

$$H = \left(\frac{1}{L} \sum_{i=0}^{L-1} \Lambda_i \right)^{-1} \log L \tag{2.1}$$

where L is the length of string S, i is the index of a symbol in S, and Λ_i is the length of the shortest sub-string starting at position i not yet observed before index i. As $L \to \infty$, the LZ-derived entropy approaches the true entropy of the string.

However, different spatial-temporal resolutions produce different visit strings for the same trajectory, making the entropy implicitly dependent on (d, T). For example, halving the sampling rate could render the visit string *AABB* as *AB* instead. A solution to a visit string's implicit dependence on the spatial-temporal resolution was provided in [27] where a formulation for the explicitly dependent entropy rate of a visit string was given. The formulation is equivalent to the entropy approximated via LZ-compression and is as follows:

$$H(d, T) = \frac{\log L}{d^2 \frac{C_1}{4T^2 L} + \frac{C_2}{4T^2 L} + 2d \frac{C_3}{4T^2 L} + d \frac{C_4}{TL} + \frac{C_5}{TL}} \tag{2.2}$$

where

$$\begin{aligned} C_1 &= \sum_{i=1}^{n} \frac{1}{v_i^{*2}} \quad C_2 = \sum_{i=1}^{n} t_{d_i}^2 \quad C_3 = \sum_{i=1}^{n} \frac{t_{d_i}}{v_i^*} \\ C_4 &= \sum_{i=1}^{n} \frac{1}{v_i^*} \quad C_5 = \sum_{i=1}^{n} t_{d_i} \end{aligned} \tag{2.3}$$

Here v_i^* is the apparent speed across the i-th cell, t_{d_i} is the total dwell time within the i-th cell with side length d, and T is the sampling rate. Note that in the original paper [27] and in [17] v_i^* is incorrectly referred to as the apparent velocity. As v_i^* only captures a magnitude it is more apt to refer to it as apparent speed as [27] do in the appendix for their paper. For notational consistency the same symbols are maintained in the formulation provided here. The constants C_1–C_5 provide a parametrization of a trajectory's entropy at various spatial-temporal resolutions while being resilient to the effects of the different spatiotemporal samplings. Differences in trajectory entropy are captured by these constants and can therefore be used to differentiate between trajectories. The intuition behind this representation is that by modelling the components of a trajectory that involve motion, which produce changes in symbols, the components that involve remaining stationary, which produce repeated symbols, and the ratio of these components, one can approximate the entropy of the resulting visit string produced by a trajectory at a given spatial-temporal resolution. In the context of CSGO, a CT player who is primarily holding space in one area of the map, would have a trajectory with lower entropy than a T player who is exploring various areas of the map to determine where to plant the bomb.

2.3.3 Convex Hull

The convex hull of a set of points X is the unique minimal set of points such that all points in X can be represented as a convex combination of the points comprising the convex hull. Geometrically the convex hull consists of the vertices of the simplest shape that surrounds all points in X. A more formal definition of convex hull is that given some set of points $x \in X$, where $X \subseteq \mathbb{R}^d$ and $\mathbb{R}^d$ is a vector space of dimension d, if n points $u_1, u_2, \ldots, u_n \in U$, where $U \subseteq X$, satisfy the affine combination condition

$$\forall x \in X, \sum_{i=1}^{n} a_i u_i = x \tag{2.4}$$

with $a_1, a_2, \ldots a_n$ such that $\sum_{i=1}^{n} a_i = 1$ and $a_i \geq 0$, then U is a convex set of X. The intersection of all possible convex sets of X is the minimal convex set and is referred to as the convex hull of X.

For use in this work it suffices to have the intuition that the convex hull is analogous to the smallest convex polygon such that a given player's position is always inside the polygon. The convex hull is representative of the space that a player could have occupied, usually summarized by the surface area and/or volume of the convex hull. A player that has explored a larger volume of space will produce a convex hull with a larger surface area and/or volume. In the context of CSGO, a player who primarily holds a key position will have a smaller convex hull compared to a player who frequently changes position or explores the map.

2.3.4 Fractal Dimension

The fractal dimension of a spatial object encodes its spatial complexity and is a generalization of the concept of an object's dimension. The fractal dimension D of a spatial object or set is defined as:

$$D = \frac{\log(N(\epsilon))}{\log(\frac{1}{\epsilon})} \tag{2.5}$$

Where $N(\epsilon)$ is defined to be the number of measuring-objects that can fit inside the original object under a scaling factor of ϵ. As an example, measuring-lines scaled to $\epsilon = \frac{1}{4}$ the size of a line to be measured allow $N(\epsilon) = 4$ measuring-lines to fit within the original line, resulting in a fractal dimension of 1. Similarly, for squares and measuring-squares, $N(\epsilon) = 16$ measuring-squares can fit within a square when scaled to $\epsilon = \frac{1}{4}$ the size of the original square, resulting in a fractal dimension of 2. Sets containing spatial objects with non-integer values for their fractal dimension fill space in a way that falls between the canonical examples of points, lines, planes, and cubes as illustrated in Figure 2.1.

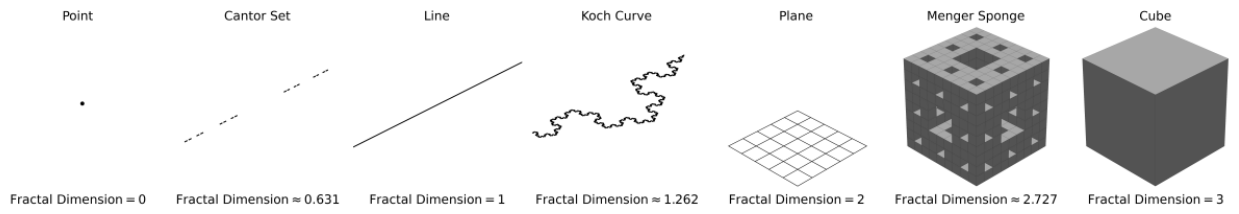

Figure 2.1: Various spatial objects and their corresponding fractal dimensions.

To estimate the fractal dimension of a trajectory, this work employs the canonical box-counting algorithm [28] to compute the box-counting dimension of discrete trajectories.

$$D_{\text{box}} = \lim_{\epsilon \to 0} \frac{\log(N(\epsilon))}{\log(\frac{1}{\epsilon})} \tag{2.6}$$

Where ϵ is the side length of hypercubes subdividing the space and $N(\epsilon)$ is the number of hypercubes containing a point in the set for a specified side length. A regression model is used to compute D_{box} from the results of multiple box counting steps. Empirically estimating the fractal dimension for a trajectory depends on the temporal discretization but will approach the true value as the number of points sampled from the trajectory increases. Player movement trajectories are expected to have fractal dimensions between one and two even though players move through a three dimensional space, as the movement trajectories are well modelled by a curve on a two dimensional manifold due to gameplay in CSGO limiting a player's range of movement vertically. In the context of CSGO, the trajectory of a CT player who is rotating through spawn from one side of the map to the other without the need to pause or check corners is expected to have a lower fractal dimension than a T player who is exploring the map and checking corners. As different maps have different layouts, the fractal dimension of a trajectory should be influenced by map geometry and play patterns that arise from the layout of the map.

2.3.5 Probability Distribution Parameters for Journey and Dwell Durations

A person's trajectory over a period of time of can be split into distinct sections where the person is considered moving or stationary. The duration of these journeys (movement) and dwells (stationary periods) follows the scale invariant power-law distribution in GPS trajectories [14, 18, 26], given by $f(x, \alpha) = \alpha x^{\alpha-1}$ with $x \in [0, 1]$ and $\alpha \in \mathbb{R}^+$. The duration of journeys and dwells follow a decaying power-law distribution, $\alpha \in (0, 1)$, with smaller values of α corresponding to a bias towards shorter durations. A compact representation that captures the power-law behaviour exhibited by the temporal properties of movement trajectories improves upon previous methods that modelled the distribution assuming normality and used high dimensional frequency vectors [7]. While sensitive to the selection of journey and dwell criteria, the lack of noise in game logs allows for idealistic

conditions such as no change in position or zero velocity to be used in journey and dwell segmentation, making α values comparable across trajectories. In the context of CSGO, a player with a large α value for their dwell distribution would be more likely to remain stationary for longer periods, possibly holding a specific part of the map. A player with a smaller α value for their dwell durations would be more likely to dwell for shorter periods, indicating that they could be constantly moving in and out of cover. For journeys larger α values correspond to an increased likelihood of journeys with a longer duration which could be indicative of longer rotation times or aggressive play.

2.4 Machine Learning

2.4.1 Logistic Regression

Logistic regression is a widely used model for classification tasks with a variety of equivalent formulations [29]. It is presented here in the context of multiclass classification with regularization.

Given a labelled dataset $\{(x_i, y_i)\}_{i=1}^N$ where $x_i \in \mathbb{R}^d$ are feature vectors and $y_i \in \mathbb{R}^K$ are class labels represented as probability vectors of length K (using one-hot encoding), the model learns one parameter vector per class, $\theta_1, \ldots, \theta_K$ with $\theta_k \in \mathbb{R}^d$.

For a given sample x_i the predicted probability vector $\hat{p}_i \in \mathbb{R}^K$ has components with values derived from the softmax function

$$\hat{p}_{i,k} = \frac{\exp(\langle \theta_k, x_i \rangle)}{\sum_{j=1}^K \exp(\langle \theta_j, x_i \rangle)}, \quad k = 1, \ldots, K \tag{2.7}$$

and the predicted class label is

$$\hat{y}_i = \arg \max_{k=1,\ldots,K} \hat{p}_{i,k} \tag{2.8}$$

Model parameters are typically learned by minimizing the categorical cross-entropy loss,

$$\ell(y_i, \hat{p}_i) = -\sum_{k=1}^K y_{i,k} \log(\hat{p}_{i,k}) \tag{2.9}$$

where $y_{i,k}$ and $\hat{p}_{i,k}$ denote the k-th components of the true and predicted probability vectors, respectively, although other loss functions can also be used.

To prevent overfitting and improve generalization, a regularization term $r(\theta)$ can be added to penalize large weights, where $\theta = (\theta_1, \ldots, \theta_K)$. Common choices are the L2 penalty $r(\theta) = \frac{1}{2} \sum_{k=1}^K \|\theta_k\|_2^2$ and the L1 penalty $r(\theta) = \sum_{k=1}^K \|\theta_k\|_1$. The resulting regularized loss function to

minimize is

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, \hat{p}_i) + \frac{r(\theta)}{C} \tag{2.10}$$

where $C > 0$ is a regularization parameter, with larger values corresponding to weaker regularization and smaller values corresponding to stronger regularization. Note that this formulation gives equal weight to all samples when minimizing the objective function though in practice each sample can be given it's own weight, typically based on the class distribution of the samples.

2.4.2 Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is an unsupervised clustering algorithm that assumes that data can be represented by a mixture of K Gaussian distributions, with each Gaussian representing a cluster of the data. Each Gaussian component is defined by its mean $\mu_k \in \mathbb{R}^d$, covariance matrix $\Sigma_k \in \mathbb{R}^{d\times d}$, and a mixture weight π_k, which represents the overall proportion of data points belonging to component k across the entire dataset, with $\sum_{k=1}^{K} \pi_k = 1$.

Given a dataset $\{x_i\}_{i=1}^{N}$, the model parameters $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ are estimated using the Expectation Maximization (EM) algorithm, which iterates the following steps until convergence [29,30]:

1. Expectation (E) Step: Compute the responsibilities $r_{i,k}$, which represent the probability that component k generated data point x_i:

$$r_{i,k} = \frac{\pi_k \cdot \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \cdot \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}, \quad k = 1, \ldots, K \tag{2.11}$$

   Ensuring that for each data point x_i that $\sum_{k=1}^{K} r_{i,k} = 1$.

2. Maximization (M) Step: Update the parameters using the responsibilities computed in the (E) step:

$$\mu_k = \frac{\sum_{i=1}^{N} r_{i,k} x_i}{\sum_{i=1}^{N} r_{i,k}} \tag{2.12}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N} r_{i,k}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} r_{i,k}} \tag{2.13}$$

$$\pi_k = \frac{\sum_{i=1}^{N} r_{i,k}}{N} \tag{2.14}$$

The EM algorithm terminates when the model parameters change by less than a specified threshold or when a maximum number of iterations is reached. The clustering can then be examined to try to determine the relationship between samples in the dataset.

2.4.3 Recursive Feature Elimination with Cross Validation

Recursive Feature Elimination with Cross Validation (RFECV) is a feature selection method that evaluates feature importance in classification tasks by combining Recursive Feature Elimination (RFE) with cross validated performance estimation. Given a dataset with N labelled examples, each described by d features and one of K possible class labels, RFE produces a ranking of features according to the their marginal contribution to a model's performance in regards to some performance metric. The performance metric is any real valued measure of predictive quality, such as accuracy, F1 score, or negative log-likelihood. As evaluating every possible subset of features is computationally infeasible, RFE approximates the search by iteratively removing the least informative features according to some importance estimate, retraining the model, and re-estimating the model's performance until only a specified number of features remain or until no features can be eliminated without decreasing performance below some specified threshold [31,32]. When the performance estimate is performed using cross validation this is referred to as RFECV. The resulting feature ranking provides a principled assessment of feature relevance that complements ablation studies and/or the manual inspection of model coefficients.

Chapter 3

Related Literature

3.1 Human Mobility

Human mobility analysis utilizes tools from mathematics, physics, and computer science, to gain insight from GPS enabled devices to model how people move. Key insights from the field include the establishment of the power-law behaviour of human trajectories [14, 18, 26], the rendering of trajectories as visit strings to estimate predictability via the visit string entropy rate [17, 18, 33, 34], and the application of fractal dimension to analyze the spatial complexity of trajectories [17,35,36].

The establishment of the power-law behaviour of human trajectories is supported by the work of [14] in which the circulation of bank notes in the United States was studied as a proxy for human travelling behaviour. This work established that the distribution of travelling distances decays as a power-law, and highlighted their connection to the scale free random walks produced by a Lévy flight or Continuous Time Random Walk (CTRW). The connection was later formalized in [26] where the authors examined mobile phone traces and provided improvements on CTRW models of human mobility through modelling both the power-law exploration step of human trajectories in combination with a preferential return component. The authors of [26] then cemented this power-law behaviour in [18] while also introducing the concept of visit strings and the concept of modelling their predictability using the entropy of the visit strings to establish a possible upper bound on the predictability of human mobility trajectories. Further work on visit string entropy was done in [33] addressing the effects of different spatial-temporal resolutions affecting the visit string entropy, with [34], further improving upon how to explicitly account for this issue. The work of [17] then combined these improvements along with other metrics for analyzing trajectories to provide a generalized trajectory analysis framework called ReFGeM. While the field in general is concerned with human trajectories, fundamentally the techniques employed are utilizable with any spatial-temporal dataset. The related literature on human mobility informs much of this work's methodology as it was primarily motivated by the future work section of [17] where they touched

on this generalizability. The framing of player behaviour analysis as a trajectory analysis problem allows for the exploration of contemporary human mobility approaches to population synthesis, environment perception modelling, and behaviour modelling to be applied to related problems in video games. A comprehensive review of the mobility literature is provided in [37] for further reading.

3.2 Video Games

Identifying sub-populations in video games has a variety of applications such as better understanding player behaviour patterns [38, 39], identifying player roles in team games [12, 40], identifying specific players [5,41], identifying bots [6,7,11], or identifying cheaters [8–10,42]. In approaching each of these specific problems there is a tendency to utilize domain specific metrics due to their discriminative utility. For example, the number of support items bought is a powerful metric in identifying support players in Dota 2 [5], but likely has limited utility in other titles or scenarios. Conversely, traversing an environment is a common feature in many games allowing for metrics that capture characteristics of how players interact with a game's environments to be utilized in examining differences between players, environments, or games. Approaches that examine player inputs so that models may generalize across titles [42] and approaches that focus on spatial metrics of trajectories [7] exist in the literature, but tend to focus on deep-learning models for specific use cases [6, 10], do not utilize all of the metrics presented here [8], or assume mobility distributions to be normally distributed [7]. All of the above approaches rely on the key idea that quantitative measures of player behaviour allow differentiation between classes of player behaviour. The related literature on video games provides context for the objectives and approaches that stand to benefit from this work's results and detail avenues for future work.

3.3 CSGO

This work leverages CSGO data as a test case for using mobility metrics to study player behaviour in games due to its popularity in the literature and as a game generally. Research into CSGO includes papers on player identification [41], player behaviour changes due to game updates [43], optimal economic decisions [44], evaluation of player actions to estimate win probabilities [45,46], and imitation learning approaches for bot creation [47] among others. Research into CSGO is supported by the the Esports Trajectories & Actions dataset (ESTA) [24], a large publicly available dataset of structured game logs which is used in this work. The related literature on CSGO provides context surrounding the subject of this work and evidence of academic interest in the field, though similarities to this work other than the subject of study are limited.

Chapter 4

Experimental Setup

Each game of CSGO generates a log, called a demofile, of the data sent to and from the game host (server) and the clients (players). This logging occurs at a predefined tick rate, typically either 64 Hz or 128 Hz, and can be used to reconstruct the game that was played. For an overview of the CSGO demofile format see [48] and for a detailed overview on the methodology of parsing these demofiles into structured data see [45].

This work focuses on the LAN subset of the ESTA dataset which consists of 680 professional games of CSGO played between July 2021 and May 2022. The ESTA dataset provides JSON files of each game's corresponding demofile parsed at 2 Hz and cleaned to remove disconnects and other issues. Matches played by professional players were chosen for this work to help control for variations in player skill, behaviours that may arise from technical issues such as high latency between the server and players, and due to the consistent temporal resolution of the data, with LAN matches always being played on servers with a 128 Hz tick rate, rather than the 64 Hz tick rate servers occasionally used in online play.

From these files the position, velocity, and camera orientation was extracted for each of the 267 unique players while they are alive in a round. This results in roughly 2.62 million samples of the form $\left(x_p, y_p, z_p, x_{vel}, y_{vel}, z_{vel}, \theta, \phi\right)$ which can be uniquely identified by keys of the form (matchID, player, round, map, tick). An example of the data comprising a trajectory before any processing can be seen in Table 4.1.

Table 4.1: Example Data from ETSA LAN

matchID	**player**	**round**	**map**	**tick**	x	y	z	velocityX	velocityY	velocityZ	viewX	viewY
046cdf91-97ab-4b8b-b19e-4c17ba3aa129	Thomas	1	de_ancient	3767	-336.60	1418.61	22.55	50.37	-232.49	0.00	282.16	0.54
046cdf91-97ab-4b8b-b19e-4c17ba3aa129	Thomas	1	de_ancient	3831	-310.35	1296.99	28.26	52.66	-244.39	0.00	282.16	0.54
046cdf91-97ab-4b8b-b19e-4c17ba3aa129	Thomas	1	de_ancient	3895	-296.14	1174.19	62.85	41.56	-246.52	0.00	289.08	4.59
046cdf91-97ab-4b8b-b19e-4c17ba3aa129	Thomas	1	de_ancient	3959	-266.99	1053.95	67.06	0.00	-241.04	0.00	288.98	4.43
046cdf91-97ab-4b8b-b19e-4c17ba3aa129	Thomas	1	de_ancient	4023	-260.46	967.10	81.08	95.11	-128.51	0.00	319.86	4.63

Each sample is then split into the components: $p = \begin{pmatrix} x_p & y_p & z_p \end{pmatrix}$, $p_{vel} = \begin{pmatrix} x_{vel} & y_{vel} & z_{vel} \end{pmatrix}$,

and $q = \begin{pmatrix} \theta & \phi \end{pmatrix}$ representing a player's position, velocity, and camera orientation respectively. The sequences of vectors $S = \{v_0, v_1, \ldots, v_{n-1}, v_n\}$ corresponding to an individual player's positional trajectory through a map $\mathcal{M} \subset \mathbb{R}^3$ or an individual player's camera orientation through camera orientation space $\mathcal{C} \subset \mathbb{R}^2$ are then examined for the duration of a round and a half of play in a game, where a half consists of the in order concatenation of the sequences representing the first 15 rounds.

For all sequences S, representing either the trajectory of an individual player's movement in a round, movement in a half, camera orientation in a round, and camera orientation in a half, the following features are computed using PYTHON 3.11.11 on a machine running ALMALINUX 9.6 (SAGE MARGAY).

The surface area and volume of the convex hull enclosing all $v \in S$ and the surface area and volume of the convex hull enclosing all $v' \in S'$ where S' consists of all vectors v', where v' is the resulting mapping of v after rescaling all points to lie within the unit cube, to control for map size. These features capture the spatial range of a trajectory and can be interpreted as the absolute and relative amount of space the trajectory covers. The fractal dimension of the set $\{v' \in S'\}$ which captures the spatial complexity of the trajectory. The multi-scale entropy rate constants C_1, C_2, C_3, C_4, and C_5 corresponding to visit strings generated by the sequence S' which capture the spatiotemporal regularity of the trajectory. The values α_j and α_d that parameterize the distributions of journey and dwell durations for the trajectory, which capture the temporal uniformity of moving and stationary periods in a trajectory.

Positional sequences are split into journeys and dwells by identifying intervals of non-zero and zero velocity and for camera sequences dwells are considered to be continuous sequences of identical vectors. This method for identifying journeys and dwells is assumed to be sufficient for positional sequences given the temporal resolution of the data. An increase in sampling frequency or more complex methods for identifying journeys and dwells, such as those of [19], may be needed for other applications.

The convex hull, fractal dimension, multi-scale entropy rate constants, and journey/dwell distribution parameters are computing using SCIPY 1.10.1 and NUMPY 1.25.0 where the size of each map comes from AWPY 1.3.1. In all cases the original resolution of data was used being to within one-hundredth of a game unit called a hammer unit (hu) for position, one-hundredth of a hammer unit per second for velocity (hu/s), and one-hundredth of a degree for camera orientation, with these being recorded every 0.5 seconds. Fractal dimensions computations involve a fit on values provided by discretizations of $\epsilon \in \{1, \frac{1}{10}, \frac{1}{100}\}$ and entropy rate computations involve a fit on values provided by all pairs of (d, T) where $d \in \{1, \frac{1}{5}, \frac{1}{10}, \frac{1}{20}, \frac{1}{50}\}$ and $T \in \{\frac{1}{2}, 1, 2, 4, 8\}$. These discretizations balance for fit quality and computation time over the maximum spatial resolution of the data, and a reasonable time scale for CSGO. Non-linear least squares optimization using the Trust Region

Reflective algorithm is used for entropy fit using SCIPY 1.10.1 and a linear least squares regression using NUMPY 1.25.0 is used for the fractal dimension fit.

To support reproducibility, a repository of all code employed in this work along with a requirements.txt file used to setup a virtual environment with the necessary dependencies can be found at the GitHub link provided in Appendix A.

A snapshot into the statistical behaviour of the features calculated per player over a half can be seen in Figure 4.2 and the population-level journey and dwell distribution of each map and side of play can be seen in Figure 4.1. The notable insights provided by Figure 4.2 are that on the CT side players dwell longer, travel more slowly, over shorter distances, and take less complex paths than on the T side, even though both players are able to move at the same speeds irrespective of side. Figure 4.1 supports these differences in how players on each side tend to move and illustrates that while all maps produce similar player movements, differences in map design do induces differences in player movement. This is most clearly seen in how journey distributions vary between the maps Inferno and Overpass, which have relatively more constrained and open layouts respectively. Visualizations of all maps are provided in Appendix A.

To demonstrate the utility of the features in distinguishing sub-populations in a virtual environment this work presents a collection of multi-class classification tasks where classes consist of 14 (map, side) pairs. There are 7 maps and 2 sides of play and feature vectors consist of averages of the above metrics across all players on a team. An interpretable multivariate logistic regression model is employed for these tasks and its performance in predicting the map and side of play from a the average properties of team's positional and/or camera trajectories in a round and the first half of a match is then examined.

These tasks represent various degrees of difficulty, with halves providing more information than individual rounds, and camera orientation not being explicitly constrained by map geometry. Predicting the map and side of play with varying amounts of data demonstrates feature utility in distinguishing sub-populations among the joint distribution of environment and gameplay objective. Performance in these tasks serves as a baseline for feature utility in differentiating player behaviour using supervised learning.

For all classification tasks this work employed a multivariate logistic regression model with L2 regularization from SCIKIT-LEARN 1.3.2 allowed to train for up to 1000 iterations with balanced class weighting. Logistic regression was chosen as it is a simple classification algorithm which helps to demonstrate the power of the features. All other parameters are the defaults with no hyperparameter tuning conducted so that the results can better serve as a baseline. The non-default parameters ensured convergence and take into account the class imbalance due to some maps being played more frequently than others. For tasks where feature vectors were computed on rounds, only rounds where all players on a team were alive for more than 30 seconds are included to ensure numerical

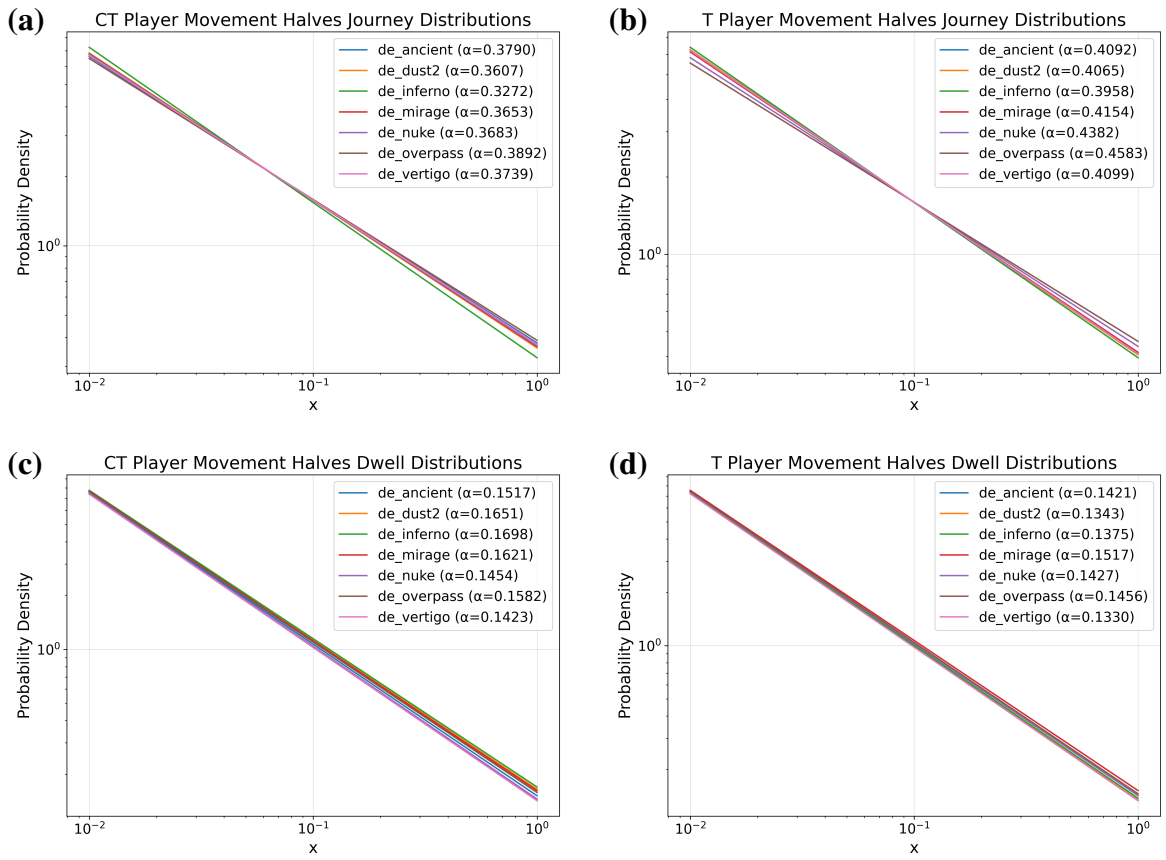

Figure 4.1: Log-log plots showcasing the power-law distributions of journey and dwell durations for player movement trajectories over halves. The alpha value for each map represents the alpha value for a distribution fit on the journey/dwell durations of all players in the dataset for that (map, side). They are the population-level distributions for each (map, side) where **(a)** is CT journeys, **(b)** is T journeys, **(c)** is CT dwells, and **(d)** is T dwells.

stability of the features. For tasks where feature vectors were derived from halves, no rounds were excluded.

For all classification tasks this work uses a training set consisting of 80% of the data while reserving 20% of the data for testing and applies Z-score normalization to the training and testing sets separately following the methodological conventions of the field. In addition to predicting the map and side of play from a team's average positional and/or camera trajectory metrics, ablation studies and a recursive feature elimination with cross validation (RFECV) experiment were conducted to examine feature utility. These tests along with the coefficients of the multivariate logistic regression models provide context on the relative impact of each feature in the classification tasks.

Along with demonstrating feature utility through classification tasks this work includes an application of the features in clustering individual player movement trajectories. Here this work attempts to ascertain an individual player's role or behaviour by examining and interpreting the cluster to

which they are assigned. To accomplish this, this work employs Gaussian Mixture Models (GMMs) SCIKIT-LEARN 1.3.2 with the k-means++ initialization method, and otherwise default parameters. As k-means++ initialization is sensitive to feature scaling all features are min-max normalized to $[0, 1]$ before clustering.

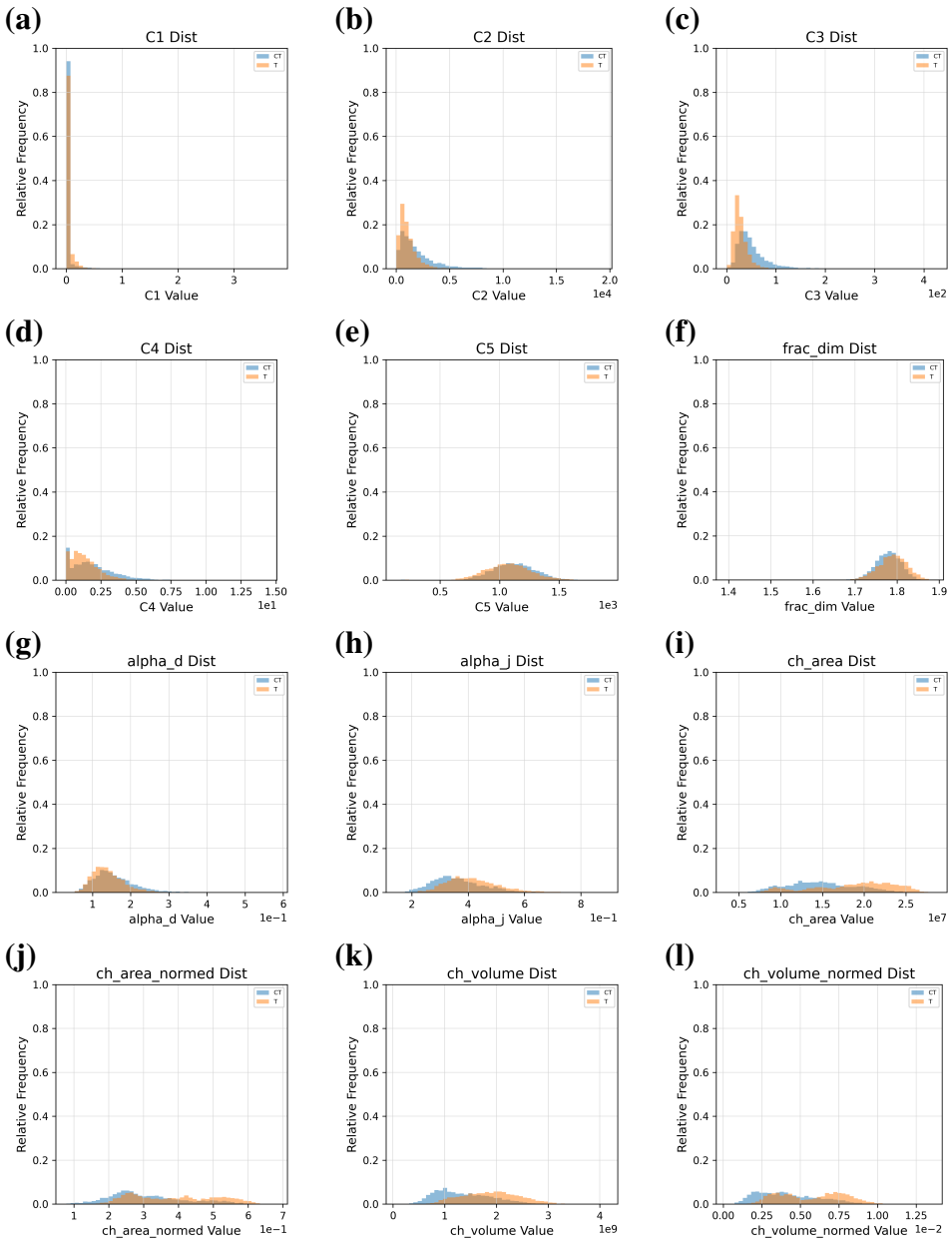

Figure 4.2: Normalized histograms for each feature computed on halves of player movement split by side with CT/T plotted in blue/orange. **(a)-(e)** entropy constants C_1–C_5, **(f)** fractal dimension, **(g)** dwell distribution alpha, **(h)** journey distribution alpha, **(i)** convex hull surface area, **(j)** normalized convex hull surface area, **(k)** convex hull volume, and **(l)** normalized convex hull volume.

4.1 Feature Processing Details

This section serves to more explicitly define and clarify the models and feature vectors used, separate from the source code available at the repository provided in Appendix A, as well as formalize short hand terms used throughout this work. For player movement and camera orientation trajectories the following steps are applied separately for each type of trajectory:

1. Stratify data by (matchID, player, map).
2. Select trajectory length i.e. stratify by round or by half (concatenation of rounds rounds 1-15) producing a trajectory for each of the 10 players (5 per side) for the corresponding time period.
3. For each trajectory compute all features using the methods described above.

The result of these processing steps are features of the following form describing an individual player's movement trajectory:

$$\begin{bmatrix} CH_S \\ CH_V \\ CH'_S \\ CH'_V \\ FD \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ \alpha_j \\ \alpha_d \end{bmatrix} \tag{4.1}$$

where CH_S is the surface area of the convex hull constructed from the original points, CH_V is the volume of the convex hull constructed from the original points, CH'_S is the surface area of the convex hull constructed from the normalized points, CH'_V is the volume of convex hull constructed from the normalized points, FD is the fractal dimension, C_1–C_5 are the multi-scale entropy constants, α_j is the parameter of the journey duration distribution, and α_d is the parameter of the dwell duration distribution.

The camera orientation trajectories produce a similar set of features with a minor difference in the convex hull features due to camera orientation space being two-dimensional rather than three-dimensional:

$$\begin{bmatrix} CH_P \\ CH_A \\ CH'_P \\ CH'_A \\ FD \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ \alpha_j \\ \alpha_d \end{bmatrix} \tag{4.2}$$

where CH_P is the perimeter of the convex hull constructed from the original points, CH_A is the area of the convex hull constructed from the original points, CH'_P is the perimeter of the convex hull constructed from the normalized points, and CH'_A is the area of the convex hull constructed from the normalized points. The other features use the same notation as Equation 4.1.

Player movement and camera orientation trajectories each produce a 12 dimensional feature vector and a 24 dimensional feature vector describing a the properties of a player's movement and camera orientation can be constructed by simply concatenating the two vectors. For the clustering experiments features corresponding to the trajectory of an individual player's movement or camera orientation on a single map, in a single game, over the duration of a round or a half, can be used directly as input to the model. For the classification experiments an additional processing step is conducted where for all players on a team their corresponding trajectory features are averaged. This is done as all five players provide information in the form of a 12 or 24 dimensional feature vector that could be used to predict the (map, side) pair, but permutations in player order and differences in player roles across teams, maps, and matches complicate the prediction task. To control for this, rather than using a 60 or 120 dimensional feature vector where each of the five player's trajectories are concatenated, the average of the five 12 or 24 dimensional features is used. The input to the logistic regression model is therefore a single 12 or 24 dimensional vector produced by computing the arithmetic mean of trajectory properties of all players on a team. For brevity camera orientation and player movement rounds/halves are used when discussing these features as the mean player movement trajectory properties for a team where trajectories are considered over the duration of a round/half of play is quite verbose. Similarly, in line with the prior work of [17] CH is used to refer to the entire collection of convex hull features and fractal dimension is abbreviated as FD. Camera movement, camera orientation, and aiming behaviour are also all used interchangeably.

Chapter 5

Results

The results below provide evidence for the claims outlined in Chapter 1. Section 5.1 provides evidence for the validity of inferring aspects of player behaviour from player movement and camera orientation trajectories and more broadly for using approaches from human mobility to analyze player behaviour video games. This is demonstrated by the performance of multiple logistic regression models in classifying the map and side of play in rounds and halves of CSGO using features derived from player movement and camera orientation trajectories. Section 5.2 provides evidence of the similarity between real world and virtual movement by demonstrating that rankings of feature utility in classification tasks involving virtual movement trajectories exhibit similar trends to those observed in rankings of feature utility in a classification task involving real world movement trajectories. Section 5.1 provides further evidence for the validity of using approaches human mobility to analyze player behaviour in video games by demonstrating that a clustering on features of player movement trajectories reveals design archetypes of the maps these trajectories occurred on and the possible roles of players who produced these trajectories.

5.1 Classification Results

Context on the performance of models is provided by Table 5.1 where the standard performance measures of training accuracy, testing accuracy, and the class weighted test metrics for precision, recall, and F1 scores are presented for the logistic regression models using feature vectors consisting of all of the player movement and/or camera orientation derived features computed on either rounds or halves of play as inputs. For all models the training and testing accuracy are similar demonstrating a lack of underfitting or overfitting. Precision, recall, and F1 scores are all similar as well demonstrating a good quality fit for all models. Confusion matrices for each model are provided in Figure 5.1 to provide further context regarding the types of mistakes each model was making.

As can be observed in Table 5.1 and in Figure 5.1 performance in tasks where feature vectors are derived from sequences representing halves of play is greater than when feature vectors are derived from rounds. Rounds exhibit far more variation than halves and definitionally produce shorter trajectory sequences, leading to higher variation in the computed features. While performance is at worst roughly six times better than the baseline expectation of a uniformly random guess in the case of camera rounds ($0.4257/\frac{1}{14} = 5.96 \approx 6$) the relatively short and variable round-level trajectories make classification challenging. Half-level trajectories are longer and less variable, resulting in improved classification performance.

Table 5.1: Performance metrics for logistic regression models

Dataset	Feature Set	Number of Features	Train Acc	Test Acc	Precision	Recall	F1
Camera Rounds	All	12	0.4190	0.4257	0.4410	0.4257	0.4214
Player Movement Rounds	All	12	0.7420	0.7393	0.7460	0.7393	0.7394
Player Movement & Camera Rounds	All	24	0.9048	0.8791	0.8827	0.8791	0.8799
Camera Halves	All	12	0.7668	0.6875	0.6938	0.6875	0.6837
Player Movement Halves	All	12	0.9558	0.9338	0.9343	0.9338	0.9337
Player Movement & Camera Halves	All	24	**0.9935**	**0.9743**	**0.9791**	**0.9743**	**0.9752**

Performance is greater when features are derived from player movement sequences rather than camera orientation sequences. Player movement on a map is constrained by the map's layout, while camera movement is only constrained by field of view. The convex hull features provide information regarding the size of the map for player movement trajectories and comparatively limited information about a map's verticality for camera orientation trajectories. The 2 Hz sampling rate is a temporal scale better suited to capturing changes in player position compared to camera orientation, particularly in a first person shooter. Despite the 2 Hz sampling rate, performance in identifying the map and side of play from camera orientation trajectories was reasonable, with an accuracy of 68.75%.

When using both movement and camera derived features together, the model was able to predict the map and side of play with 97.43% accuracy, demonstrating feature utility in accurately differentiating sub-populations. While tasks explored in this study serve as conceptual demonstrations due to a limited selection of labels, these initial results suggest the inclusion of these features may benefit other machine learning objectives. As observed in Figure 5.1, models generally made mistakes on maps with similar play patterns and/or layouts and tended to still correctly identify the side of play even when making a mistake on the map. This suggests the metrics still manage to accurately capture information regarding the high-level behaviour arising from the side of play.

While not directly comparable, in relation to the accuracy of 83.12% in [17] where the authors differentiate between datasets of GPS trajectories in various cities and transportation modes using many of the same features, this work's 73.93% accuracy for the player movement in rounds model

and 93.38% for the player movement in halves model is commensurate. Performance here is an indicator that feature utility is similar in both domains.

5.2 Feature Importance

In order to evaluate feature utility this work examines the feature rankings provided by the magnitudes of the coefficients of the logistic regression models and RFECV results for player movement in halves, camera orientation in halves, and compares the results with those of [17] where they provide the ranking [BA, C_5, FD, CH, C_4, C_2, C_1, C_3, CH10] for real-world movement trajectories. Note that the buffer area feature (BA) functions similarly to the convex hull in that it encodes information regarding the spatial range of trajectories.

The values of the coefficients that inform this work's feature ranking can found in Figure 5.2. For brevity only the rankings for halves are discussed here as the feature ranking for rounds is similar with an increased importance of convex hull features, likely due to their numerical stability in the case of the shorter sequence lengths. In all cases the RFECV ranking was in agreement with that of the coefficient magnitude based ranking. Information regarding accessing the full feature rankings, ablation studies, and RFECV rankings can be found in Appendix A.

For player movement halves the ranking of features is [CH, C_3, FD, C_5, C_4, C_2, α_j, C_1, α_d]. For camera orientation halves the ranking of features is [FD, C_3, CH, C_5, C_4, C_2, α_d, α_j, C_1]. For halves using both player movement and camera orientation, the ranking of features is [CH^m, FD^m, FD^c, C_3^m, CH^c, C_4^c, C_2^c, C_3^c, C_4^m, C_5^m, C_5^c, α_j^m, C_2^m, α_j^c, C_1^m, α_d^c, α_d^m, C_1^c], where CH groups all convex hull derived features together, X^m denotes player movement features, and X^c denotes camera orientation features.

There is a general trend of spatial range, spatial complexity, and the entropy constants that capture the ratio of dwell time to apparent speed (C_3) and total dwell time (C_5) along a trajectory being the most significant in differentiating between map and side of play. For camera orientation trajectories the spatial range is less important than in movement. Due to aiming predominantly occurring on the horizontal axis in CSGO and most players making a full rotation across this axis over the course of a round, variation in convex hull values is primarily attributed to vertical adjustments players make. As across all maps professional players tend to aim at head height at all times to be ready for an engagement, the convex hull volume varies minimally between maps with the exception being that when maps employ significant verticality the convex hull is larger. This is likely due to players needing to adjust where they are aiming along the vertical axis more often due to more dramatic changes in elevation on these maps.

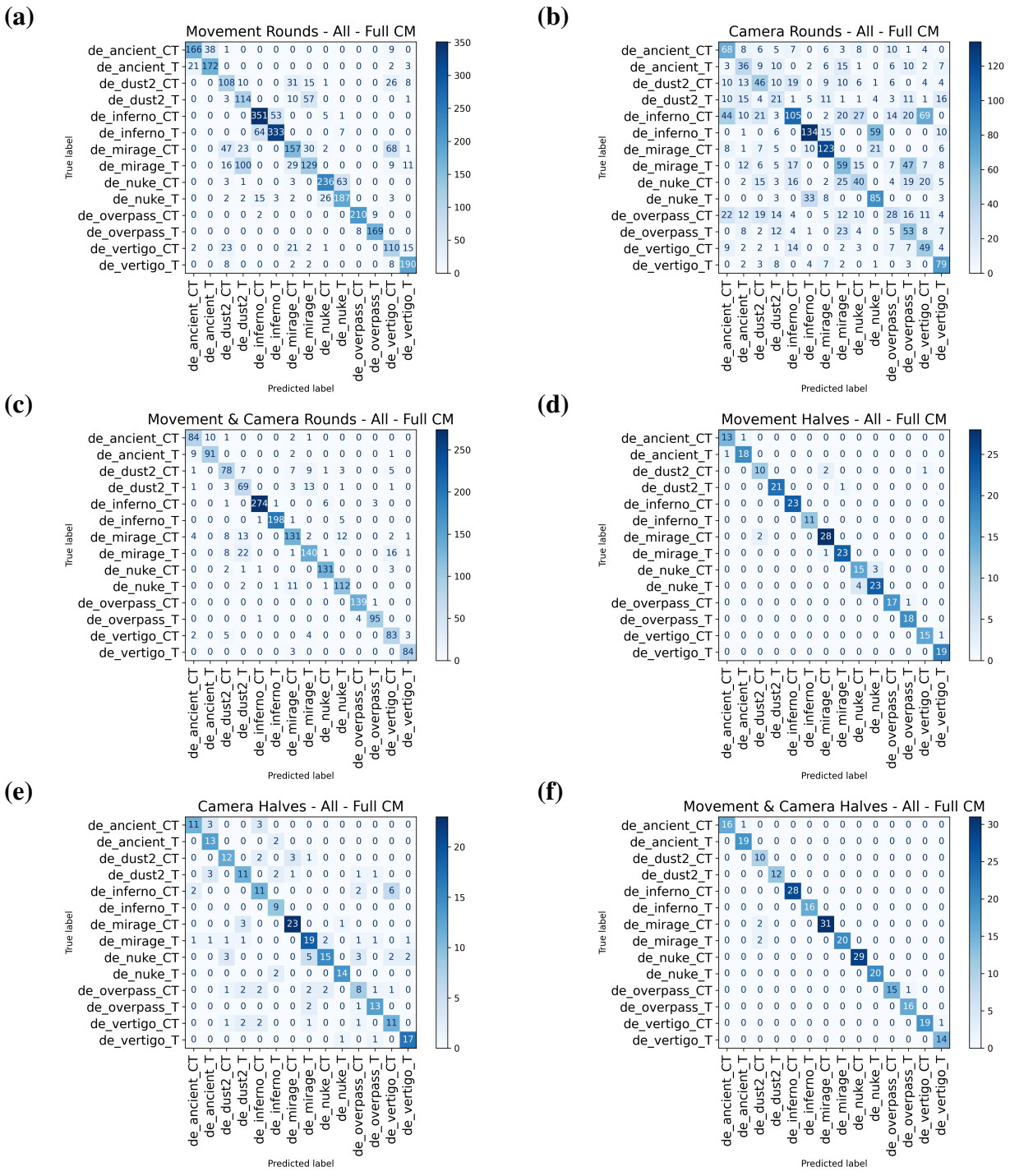

Figure 5.1: Confusion matrices for classifying map and side for a round using **(a)** player movement features over rounds, **(b)** camera orientation features over rounds, **(c)** both player movement and camera orientation features over rounds; and for a half using **(d)** player movement features over halves, **(e)** camera orientation features over halves, and **(f)** both player movement and camera orientation features over halves.

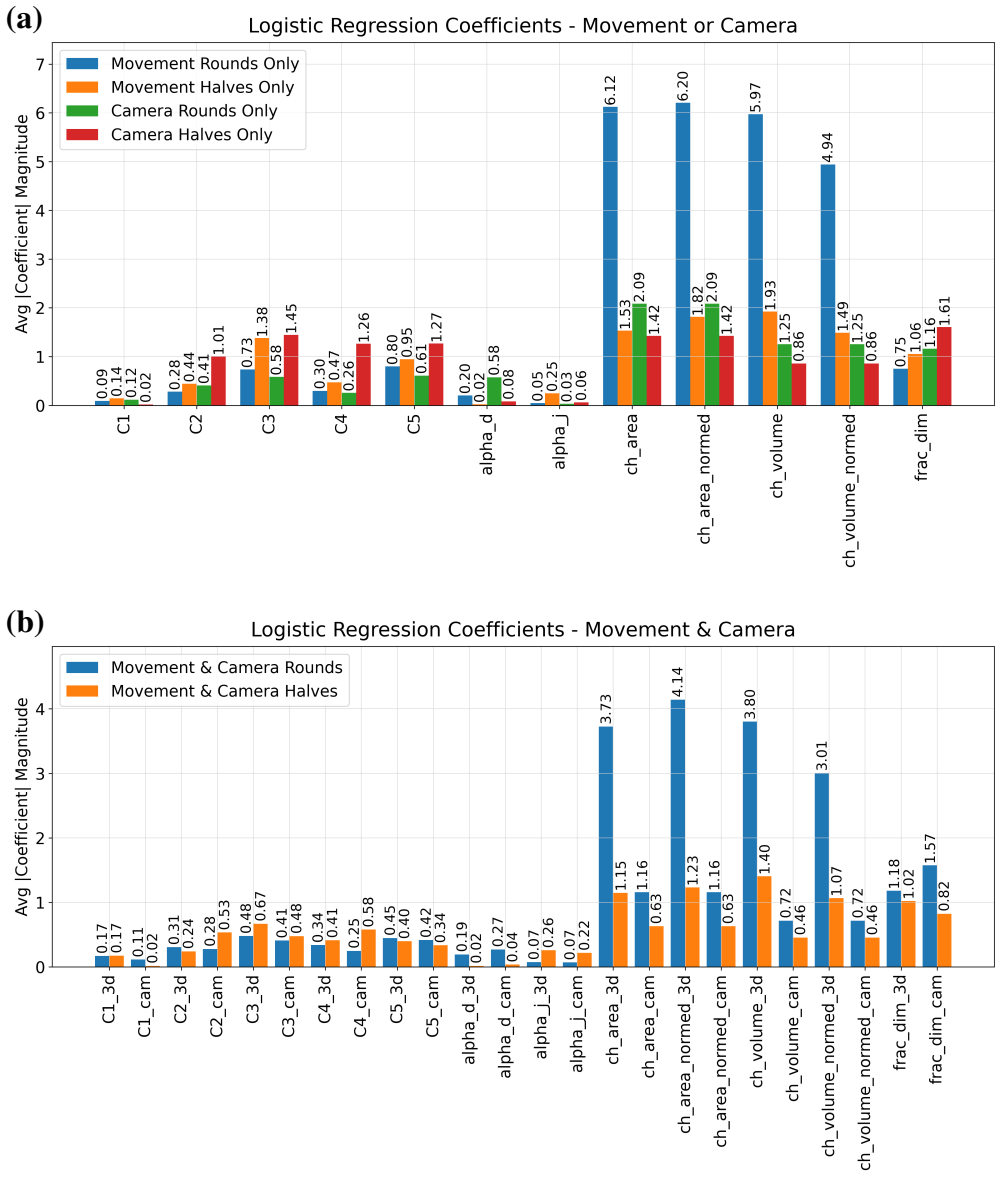

Figure 5.2: Bar plots of the average coefficient magnitudes for each feature for **(a)** player movement and camera orientation models and **(b)** combined player movement and camera orientation models.

5.3 Clustering Results

Gaussian Mixture Models are applied to cluster individual player movement halves, to explore feature utility in capturing role-level distinctions between players for each side of play. To determine the number of clusters, the Akaike Information Criterion (AIC) score for various cluster numbers is computed. AIC score is a tool from information theory defined as

$$\mathrm{AIC} = 2k - 2\ln(\hat{L}) \tag{5.1}$$

where k is the number of estimated parameters in the model and $\hat{L}$ is the maximized value of the likelihood function for the model given the data. Here k represents the number of Gaussians or clusters for each model as the number of parameters scales linearly with the number of Gaussians or clusters in the model. A higher likelihood of the data being represented by a model, and therefore lower AIC score, implies that the data is better modelled by a mixture of k Gaussians. The resulting values indicated diminishing returns at roughly twelve clusters per side, as seen in Figure 5.3.

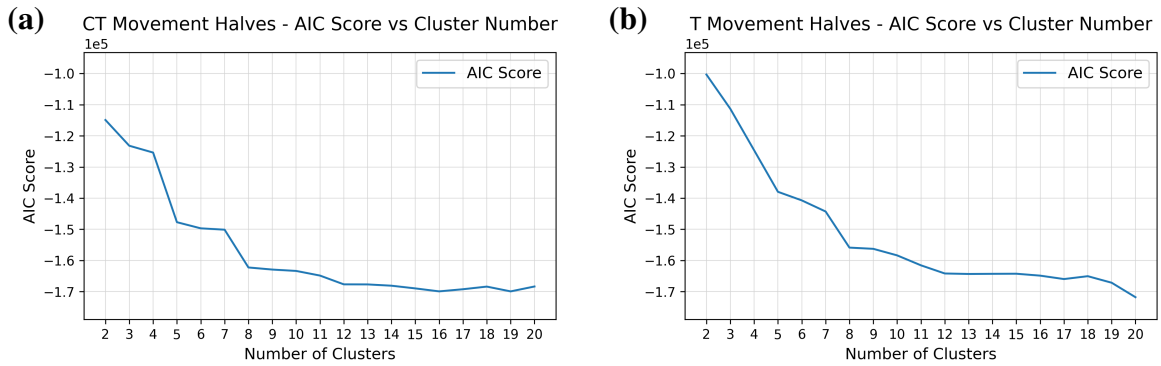

Figure 5.3: **(a)** AIC scores of Gaussian Mixture Models for CT halves of player movement, **(b)** AIC scores for T halves of player movement.

Map distributions within individual clusters diverged from that of the original data, and manual inspection of clusters indicated that clustering was primarily driven by map and secondarily by player behaviour. Clusters that showcase this predominantly map based clustering include the CT clusters 3, 6, and 7, and the T clusters 0, 2, 4 which only contain samples from a single map as seen in Figure 5.4. Secondary clustering behaviour on role is exhibited most cleanly in CT cluster 8 and T cluster 1. These clusters consist of trajectories on maps that are considered to have a stable meta, [49] and contain well known aggressive Rifler players such as Stewie2K, ZywOo, Niko, and b1t [50]. This aligns well with the clusters' lower mean values for C_3 and C_5 and higher fractal dimension in comparison to other clusters. These values are indicative of the faster pace of play that Rifler players exhibit as the combat backbone of teams. The relative sizes of the clusters as seen in Table 5.2 support a predominantly map driven clustering with CT cluster 9 providing strong evidence for the smaller secondary clusters being role driven and non-spurious. While CT cluster 9 is the smallest cluster, the lower fractal dimension and convex hull values are indicative of behaviours commonly associated with the Anchor role present on the more vertical maps of Nuke and Vertigo. On these maps players in the Anchor role tend to stand their ground and make vertical retreats by dropping down to a lower level rather than making the longer lateral retreats that typically occur on other maps.

The observed clustering reflects the dominance of convex hull derived features in distinguishing spatial environments and the structured nature of professional play, where players' adoption of

the most effective tactics available limits variation in player movement. The general composition of the data is represented in the clustering as well, with the largest natural grouping of the data occurring within maps rather than on players. The specific clustering observed is ostensibly due to the effect of the spatial environment on gameplay with the resulting clusters yielding meaningful insight into map design archetypes, designated as Classic/Mid Control (Dust2, Mirage), Choke Heavy/Constrained (Inferno, Ancient), Increased Verticality (Nuke, Vertigo), and Increased Size (Overpass) based on the within cluster map distributions seen in Figure 5.4.

Table 5.2: GMM Movement Halves Cluster Sizes and Unscaled Means

Model	Cluster ID	Samples	ch_area	ch_volume	ch_area_normed	ch_volume_normed	frac_dim	C1	C2	C3	C4	C5	alpha_d	alpha_j
CT (k=12)	0	905	1.622e+07	1.417e+09	0.388	5.235e-03	1.78	3.728e-05	1955	47.2	1.80	1088	0.161	0.363
CT (k=12)	1	71	1.641e+07	1.393e+09	0.294	3.344e-03	1.78	4.505e-01	2225	32.8	2.26	1118	0.158	0.402
CT (k=12)	2	65	1.094e+07	8.618e+08	0.295	3.725e-03	1.78	2.075e-01	1136	31.9	2.67	1001	0.146	0.425
CT (k=12)	3	517	1.298e+07	1.922e+09	0.272	5.829e-03	1.77	6.669e-05	1368	45.9	1.81	1071	0.141	0.365
CT (k=12)	4	61	1.601e+07	1.548e+09	0.389	5.671e-03	1.77	2.164e-04	6519	129	0.805	1258	0.227	0.298
CT (k=12)	5	23	1.658e+07	1.192e+09	0.286	2.702e-03	1.76	1.497e-11	7964	157	2.39	1412	0.225	0.259
CT (k=12)	6	351	1.657e+07	1.812e+09	0.177	2.007e-03	1.80	6.573e-05	1934	48.7	1.97	1202	0.158	0.383
CT (k=12)	7	355	1.484e+07	1.018e+09	0.425	4.935e-03	1.78	1.566e-04	1565	44.9	1.90	1065	0.150	0.379
CT (k=12)	**8**	**120**	**1.503e+07**	**1.506e+09**	**0.350**	**5.242e-03**	**1.78**	**2.116e-01**	**1459**	**28.6**	**1.88**	**1003**	**0.162**	**0.404**
CT (k=12)	**9**	**19**	**5.842e+06**	**4.631e+08**	**0.132**	**1.505e-03**	**1.50**	**2.136e-02**	**155**	**5.79**	**0.186**	**203**	**0.169**	**0.453**
CT (k=12)	10	573	1.636e+07	1.097e+09	0.273	2.364e-03	1.78	1.251e-04	2642	65.5	2.26	1240	0.168	0.329
CT (k=12)	11	334	8.921e+06	8.291e+08	0.222	3.257e-03	1.78	1.900e-04	1476	52.4	2.36	1097	0.141	0.365
T (k=12)	0	86	1.863e+07	1.451e+09	0.534	7.034e-03	1.78	1.038e-01	755	17.7	1.21	949	0.145	0.473
T (k=12)	**1**	**184**	**2.030e+07**	**1.999e+09**	**0.485**	**7.385e-03**	**1.80**	**9.419e-02**	**825**	**15.9**	**0.962**	**959**	**0.144**	**0.463**
T (k=12)	2	593	2.238e+07	1.738e+09	0.373	3.745e-03	1.78	2.965e-02	1110	30.9	1.92	1177	0.136	0.399
T (k=12)	3	92	9.117e+06	9.807e+08	0.223	3.756e-03	1.71	7.291e-02	557	16.7	1.41	817	0.132	0.464
T (k=12)	4	370	2.340e+07	2.605e+09	0.251	2.887e-03	1.82	2.920e-02	949	25.9	1.11	1156	0.145	0.459
T (k=12)	5	807	2.077e+07	2.054e+09	0.496	7.588e-03	1.80	1.138e-05	1153	27.9	1.01	1058	0.144	0.403
T (k=12)	6	425	1.439e+07	2.230e+09	0.302	6.765e-03	1.77	2.184e-05	1018	28	1.27	1023	0.144	0.417
T (k=12)	7	45	1.833e+07	1.895e+09	0.446	7.190e-03	1.79	3.007e-01	3225	40	0.930	1148	0.183	0.367
T (k=12)	8	38	2.045e+07	1.659e+09	0.337	3.518e-03	1.78	4.959e-11	2585	60.4	2.55	1340	0.167	0.334
T (k=12)	9	295	9.584e+06	1.086e+09	0.239	4.264e-03	1.78	5.695e-10	1114	34.9	1.82	1095	0.134	0.398
T (k=12)	10	135	1.429e+07	2.224e+09	0.300	6.748e-03	1.77	1.057e-01	748	16.9	1.17	931	0.136	0.481
T (k=12)	11	327	1.895e+07	1.464e+09	0.543	7.098e-03	1.78	8.921e-14	1207	28.6	1.35	1017	0.141	0.392

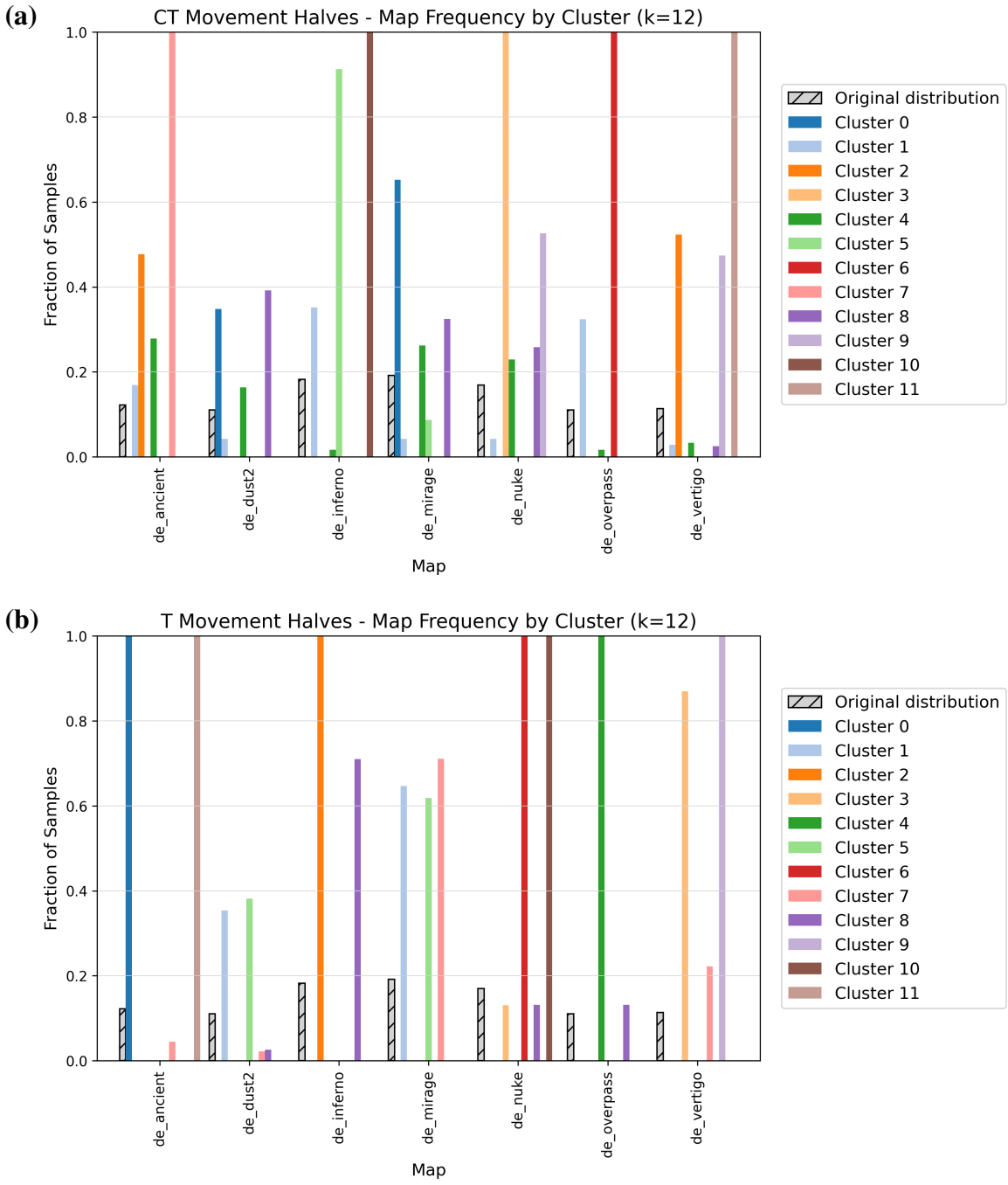

Figure 5.4: Bar graphs representing map distributions for each side's clustering with the original map distribution of the data plotted as a hashed grey bar. Bars can have values in the range [0,1] with extremes indicating that a cluster contains zero samples from a map or that a cluster only contains samples from a single map. **(a)** Map distribution for CT clustering with 12 clusters, **(b)** Map distribution for T clustering with 12 clusters.

5.4 Ancillary Results

5.4.1 Investigating Power-Law Behaviour

To determine whether the durations of journey and dwell distributions for player movement and camera orientation trajectories followed a power-law distribution in the same way real-world human mobility trajectories do the following experiment was conducted. Trajectories were segmented into journeys and dwells following the methods outlined in Chapter 4 and the durations of these journeys and dwells were then grouped per player, per map, per side. Initial visualizations of the histograms indicated that the duration distributions were likely well modelled by either a power-law or exponential distribution. A fit for the continuous probability distributions of exponential and power-law was then computed for each grouping of the data using SCIPY 1.10.1 and evaluated through the negative log-likelihood of the fits for each distribution. As can be seen in Table 5.3 the data was generally better modelled by a power-law distribution in almost all cases. While either approach is technically valid, the evidence for power-law behaviour justified the primary experiments proceeding with modelling journey and dwell distributions under the assumption that they were power-law distributed. This is consistent with observations of real-world spatial behaviour [14, 17, 18, 26, 33, 34].

Table 5.3: Journey/Dwell Distribution Fit Quality Comparisons

Data	Percent Best Fit by Power-Law	Percent Best Fit by Exponential	Avg. % Difference in Fit Quality (Better Fit by Power-Law)	Avg. % Difference in Fit Quality (Better Fit by Exponential)
Camera Rounds Dwells	**100%**	0%	347.67%	-
Camera Rounds Journeys	**100%**	0%	44.51%	-
Camera Halves Dwells	**100%**	0%	321.43%	-
Camera Halves Journeys	**99.88%**	0.12%	4.51%	0.18%
Player Movement Rounds Dwells	**100%**	0%	298.13%	-
Player Movement Rounds Journeys	**100%**	0%	16.20%	-
Player Movement Halves Dwells	**100%**	0%	38.57%	-
Player Movement Halves Journeys	**99.95%**	0.05%	8.61%	2.83%

5.4.2 Player Identification

A natural extension to the primary work would be to examine the possibility of identifying individual players based on their trajectories. To investigate this I conducted a classification experiment with (player, map, side) as labels with features derived from trajectories of player movement halves. Due to the limited amount of data for many players I excluded players who didn't play at least five games from this experiment. In this classification task the logistic regression model was able to achieve scores of 32% and 12% for training and testing accuracy respectively. As illustrated in Figure 5.5, the model performed reasonably well given that the classification task is a few-shot learning problem [29], where more complex models are typically employed. Compared to uniform

random guessing the results were about 50 times better than expected ($0.1214/\frac{1}{422} = 51.23 \approx 50$) but the quantity of data for each class and clear overfitting made any conclusions unreliable. The approach warranted further investigation considering that α_j seemed to provide utility on par with the CH features in this task but concerns regarding the limited dataset precluded further work in this area. A dataset more focused on individual players could potentially resolve this issue and allow for further research.

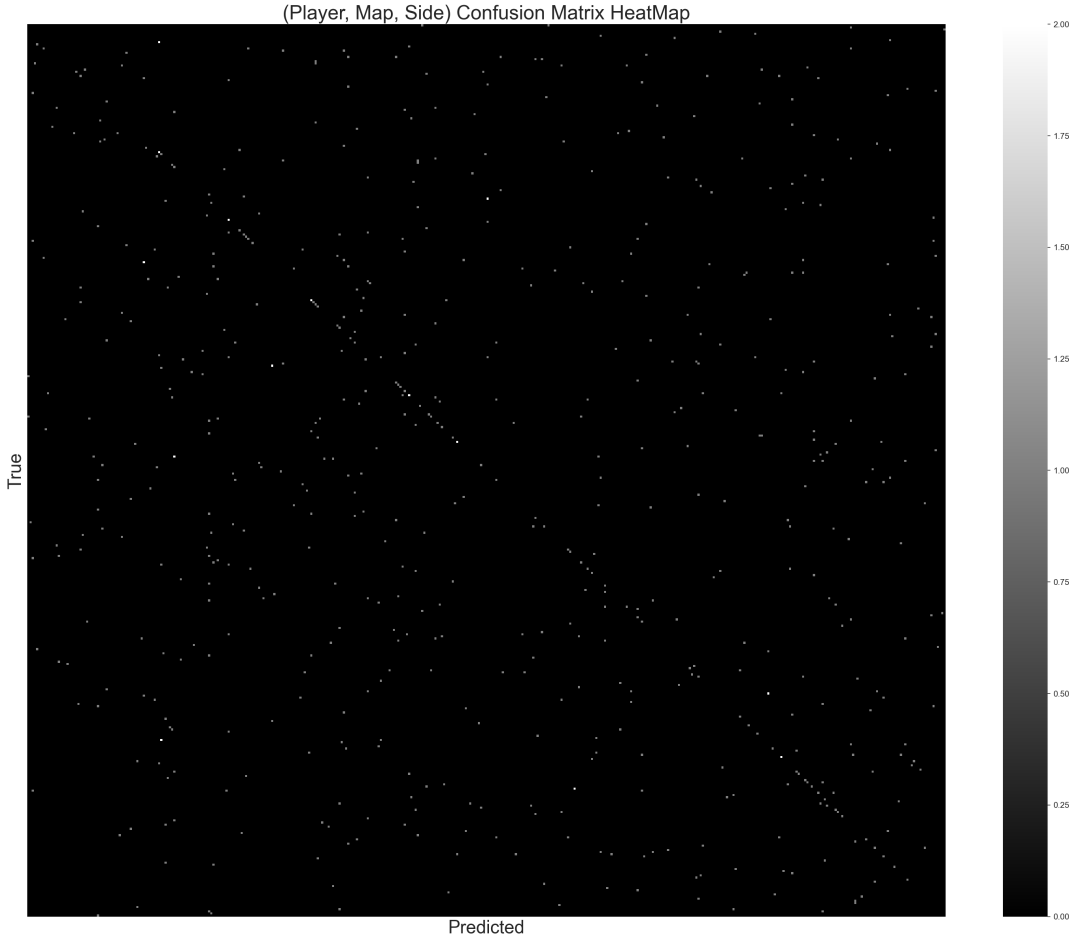

Figure 5.5: Heat map of the test set confusion matrix for the (player, map, side) logistic regression model. The pattern along the main diagonal with seemingly random noise is indicative that the model is performing better than chance.

Chapter 6

Discussion & Conclusion

6.1 Discussion

The results demonstrate the utility of adapting spatial mobility features to the analysis of player behaviour in games. Trajectories of player controlled agents and cameras in virtual environments have similar characteristics to real-world movement and provide meaningful information regarding the environment and roles of players. Journey and dwell duration distributions for player movement and camera orientation in CSGO follow the same characteristic power-law behaviour exhibited by real-world movement [18] and distributions for each map and side of play align with observed gameplay differences in those maps and roles as seen in Figure 4.1. Dwell distributions confirm that CT players have an increased likelihood of dwelling for longer durations, and journey distributions confirm that T players have an increased likelihood of travelling for longer durations. Distributions across maps also align with play patterns induced by map geometry, best exemplified by the constrained environment of Inferno and open nature of Overpass, producing increased likelihoods of shorter and longer journeys respectively.

Given the mostly bimodal nature of movement in CSGO, either walking or running, it is unsurprising that the entropy constant C_3, which captures the ratio between dwell time and apparent speed, was able to differentiate between classes effectively. With a consistent movement speed in CSGO $\frac{d}{T}C_3 \propto C_5$ which results in high degree of correlation between the two features. This relation along with distinct paces of play for the CT and T sides could explain the increased importance of C_3 as well. Correspondingly, less value is placed on C_4 and C_5 in the classification tasks compared to that of real-world movement in [17], where classes had various transportation modes and thus different characteristic apparent speeds and dwelling behaviours. For camera orientation trajectories C_3 can be a signifier of the speed at which players adjust their aim, and the fractal dimension can signify the complexity of these adjustments. In the context of game design, this could indicate the predictability or pacing of an environment, with larger values of C_3 indicating increased peri-

ods of stable or stationary aiming and a lower speed at which adjustments are made. As the fractal dimension is indicative of how spatially complex adjustments are, with more complex adjustments producing increased values, it could be indicative of the complexity of input the game demands of players.

Classification results indicated that the framework can be effectively used to distinguish high-level differences in player behaviour and environment when applied to player movement trajectories. The performance increase when player movement and camera orientation trajectories were used together highlights that player behaviour in FPS games is better represented by movement and aim jointly rather than independently. In other games where the objects of interest may vary, the generalizability of this framework to any spatial object will presumably provide a method for a more comprehensive analysis.

Clustering results indicated that the framework is sufficient in surfacing broad map design archetypes but indicate that distinguishing between low-level difference in player behaviour may require supplementary features or stratification by map type. Results suggest that this work's feature set succeeds in capturing distinctions in environments but that disentangling environmentally driven and role driven properties of player behaviour may require additional data, game specific features, or measures of behavioural consistency across multiple play sessions. Clustering using this framework provides a method to quantify the similarity of play patterns across maps which can aid in producing environment designs either similar to environments already favoured by players or distinctly new experiences. While there seemed to be some indication of role specific behaviour within some clusters, the nature of interpreting clusterings and confirmation bias, precludes further claims based on the results of these experiments.

This work highlights the potential of generalized mobility metrics as components of game-agnostic and interpretable models of player behaviour and advocates for their integration into games research. While the metrics presented are theoretically applicable to any scenario in which an object of interest moves through space over time, this work only provide evidence of their utility in a single game and for relatively simple tasks. Due to a lack of meaningful labels, limited data, and the structured nature of professional play, the framework presented here may not provide meaningful utility in tasks more complex than those demonstrated or for a wider player base. The limited amount of data for camera trajectories in particular limits the impact of this study as changes in camera orientation occur more frequently than every half second. An increased sampling resolution could be required to accurately describe the properties of camera orientation trajectories or capture subtle differences in player aiming behaviour. While this work provides a baseline amount of evidence for the framework's utility, further work is needed to asses impact in practical applications for player behaviour analysis. Overall, this work showcases a novel approach for analyzing player behaviour in games, but is limited in that analysis is constrained to a single title and only to professional play.

While this work provides validity for examining spatial trajectories of player movement and aiming behaviour in the same manner as human trajectories in the real-world, the full scope has not yet been explored. Other tasks should be explored to validate the utility of this approach and further explore the scope of what these metrics can provide for our understanding of player behaviour.

6.2 Future Work

While this work constitutes a novel contribution to the spatial analysis and game analysis literatures, in practice more work is required to deliver value to game developers. In presenting a novel methodological framework for the analysis of games, I hope that future work can focus on integrating these methods into applications that can improve player experiences in games through a better understanding of player behaviour. This is facilitated by this work improving upon the code provided in [17] by parallelizing it, allowing computation of these features on vectors of any dimension instead of just those of dimension two, and removing dependence on libraries not commonly used in data analysis. The generalizability of this framework makes for a broad range of avenues for future work: studying aiming behaviour for performance assessment, detecting bots, cheaters, or smurfs via trajectory-based signatures, comparing player movement across different maps, games, or genres, and simulating mobility models via the Source engine to validate real-world hypotheses. Exploration of these tasks would involve the collection of high resolution labelled data and assessing whether the computational cost of these features warrants their integration into practical applications.

The above tasks all rely on the assumption that differences in player behaviour captured by this framework will allow for a learning algorithm to distinguish between various types of player behaviour. Player behaviour is complex and influenced by various factors, making it difficult to pinpoint exact causes for why a player may be behaving in a certain way. A new account that performs exceptionally well compared to their peers could be a pro player's smurf account, a cheater, or just someone who has played similar titles before and is therefore at higher skill level than expected. To differentiate between these behaviours data needs to be collected, analyzed, and used to make a prediction on the likelihood of each event. The metrics presented here have the benefit of generality, being able to integrated into any system with an objective of this type. Exploring whether their integration will lead to meaningful improvements to these types of systems remains to be seen and thus represents the primary direction for future work. As to behaviours the framework might capture, and therefore assist in the differentiation of, I posit a few theories on how player behaviour may be influenced and captured by these metrics:

- As player skill and familiarity with a game increases, players will make decisions more

quickly and confidently, leading to more efficient movement and less time wasted being stationary. This behaviour may result in trajectories with lower fractal dimensions and dwell durations.

- An aim-bot that consistently corrects player camera orientations in the same manner could lead to more predictable aiming behaviour that could be captured by the multi-scale entropy constants.
- A trigger-bot that automatically fires when an enemy appears on screen could decrease target acquisition time and need for aim correction resulting in decreased fractal dimension and increased dwell durations as players simply wait for enemies to come into view.
- Cheats that provide players information, such as wall-hacks, could allow for more informed decision making resulting in players more confidently moving through spaces they know opponents do not occupy. This could result in longer journeys and increased values of α_j.

6.3 Conclusion

This work presents a framework for analyzing player behaviour in virtual environments through features adapted from human mobility research. Machine learning tasks using these features showcase the framework's utility in providing high level distinctions in gameplay dynamics and environmental structure and highlight the framework's capacity to capture latent properties of virtual environments and player behaviour. While role-level differentiation was limited, it presents an opportunity for future research on disambiguating between players in a game agnostic manner, in that no specific assumptions were made regarding any properties CSGO. Beyond analytics, the generality of this framework positions it for broader application across games and even for cross-domain comparisons with real-world mobility studies. By bridging interpretable mobility metrics with game telemetry, this work contributes a foundation for standardized, comparative approaches to player behaviour analysis. I anticipate that extending this framework to diverse datasets and practical applications such as performance assessment, anomaly detection, and map design evaluation will deepen both our theoretical understanding of mobility and provide practical insights for game development.

Appendix A

Appendix

A.1 Additional Information

All source code required to generate the figures, tables, and models produced by and for this work can be accessed at `https://github.com/ReiaDrucker/ReFGAME_2025` with the dataset used in this work accessible at `https://github.com/pnxenopoulos/esta`. Including all figures and tables for the various ablation studies and other experiments would result in an appendix length of hundreds of pages and therefore digitally archival was performed instead.

A.2 CSGO Maps

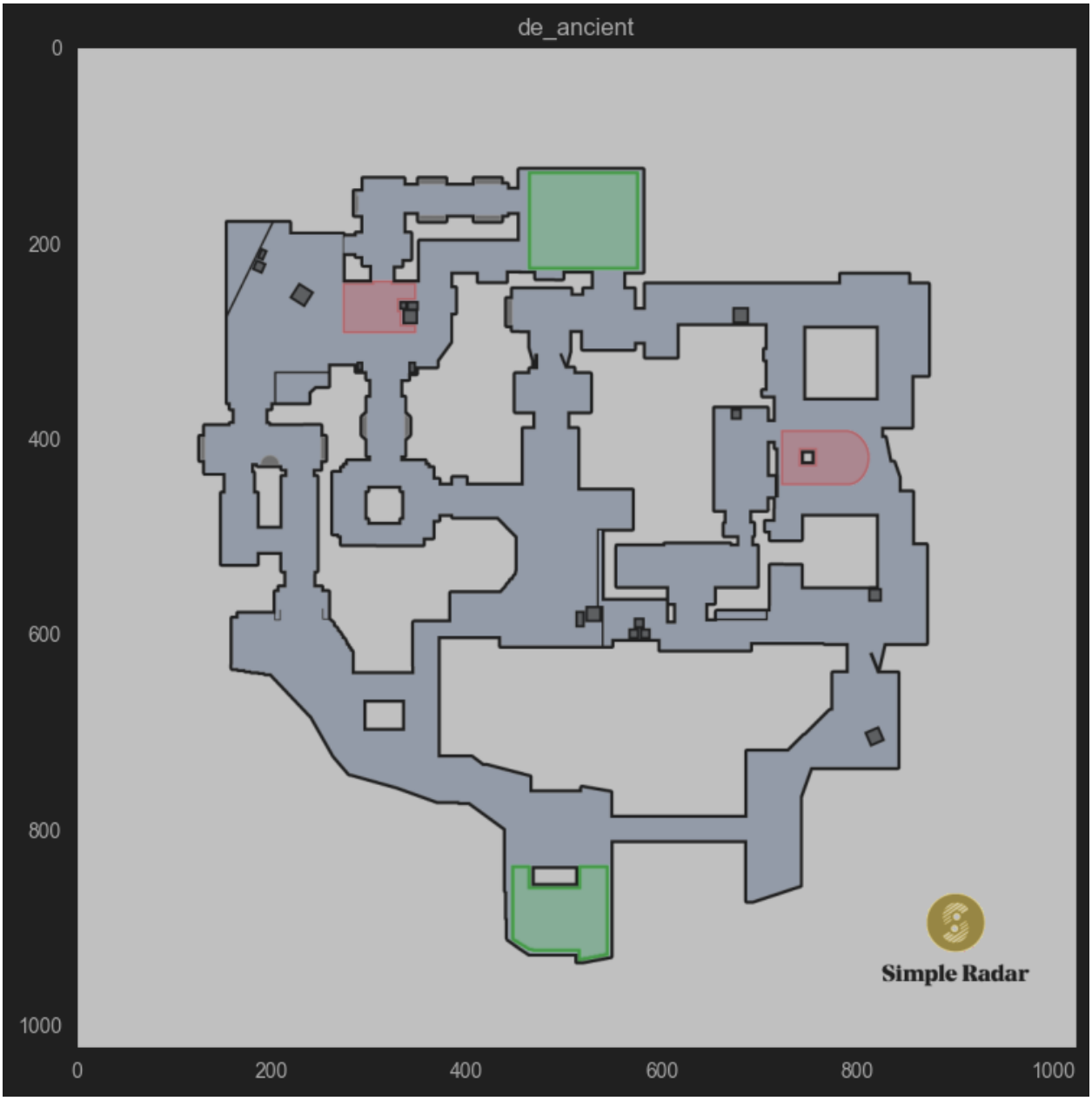

Figure A.1: Mini-map radar view of the map Ancient with spawn locations in green and bomb-sites in red.

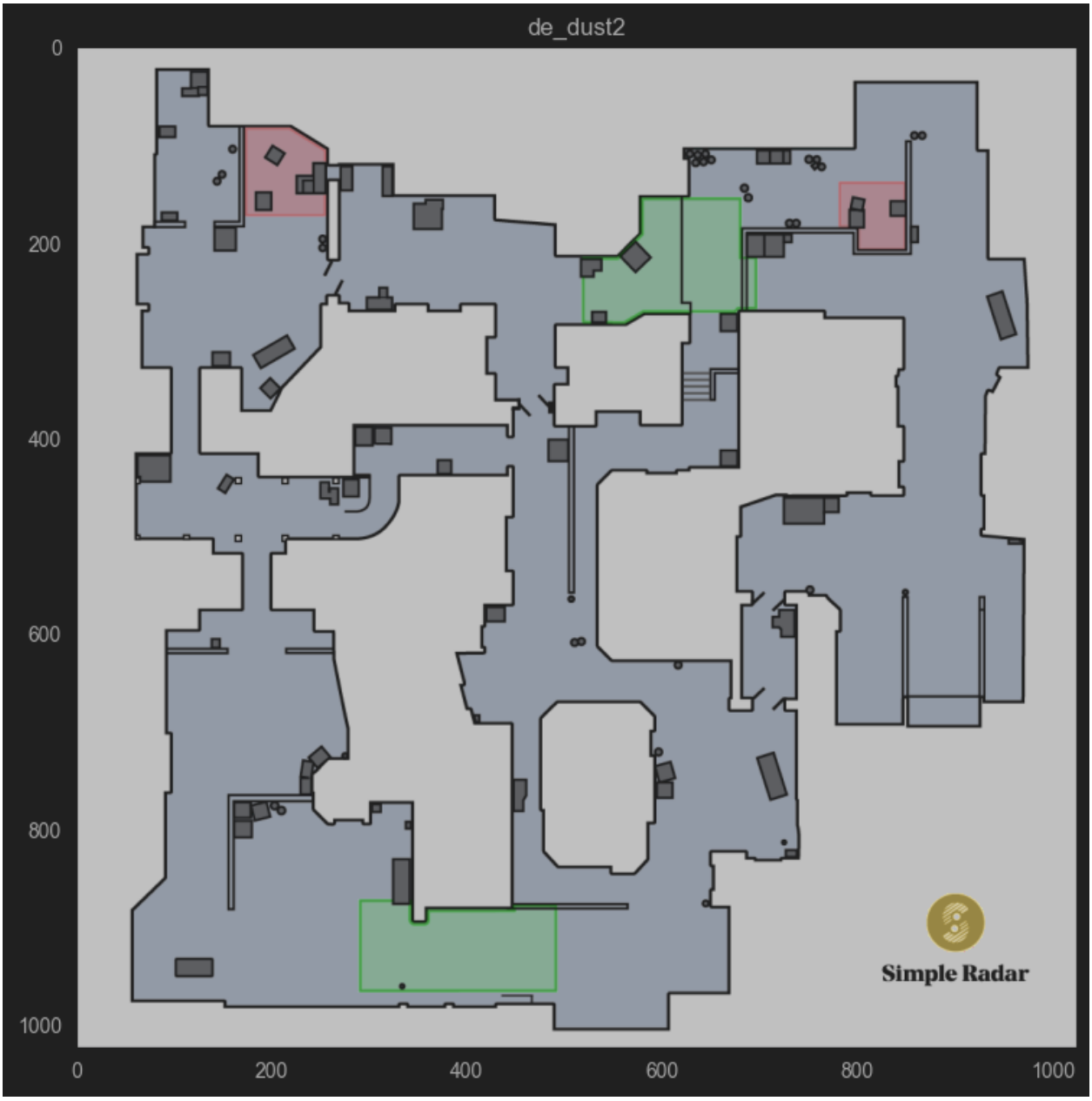

Figure A.2: Mini-map radar view of the map Dust2 with spawn locations in green and bomb-sites in red.

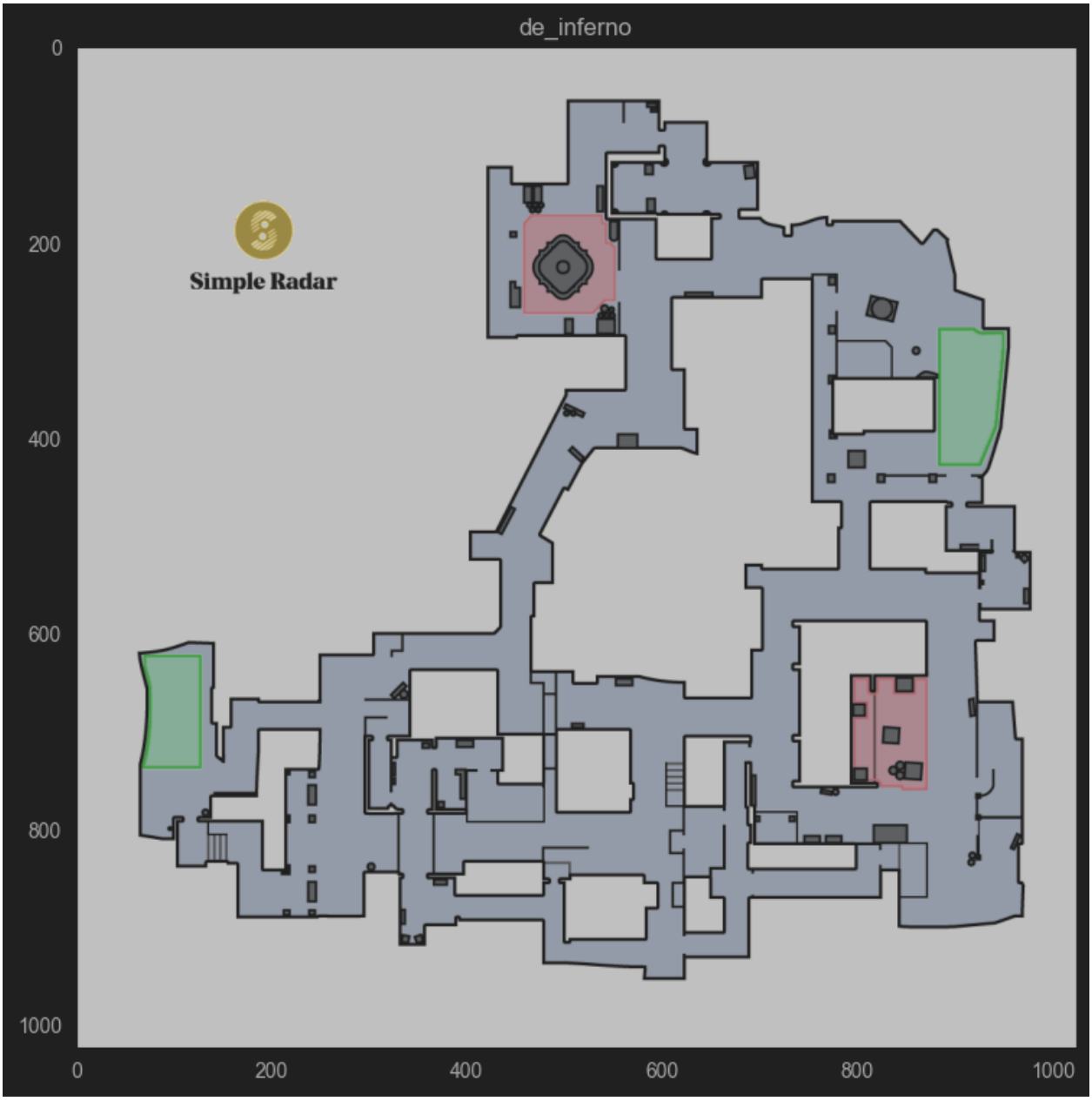

Figure A.3: Mini-map radar view of the map Inferno with spawn locations in green and bomb-sites in red.

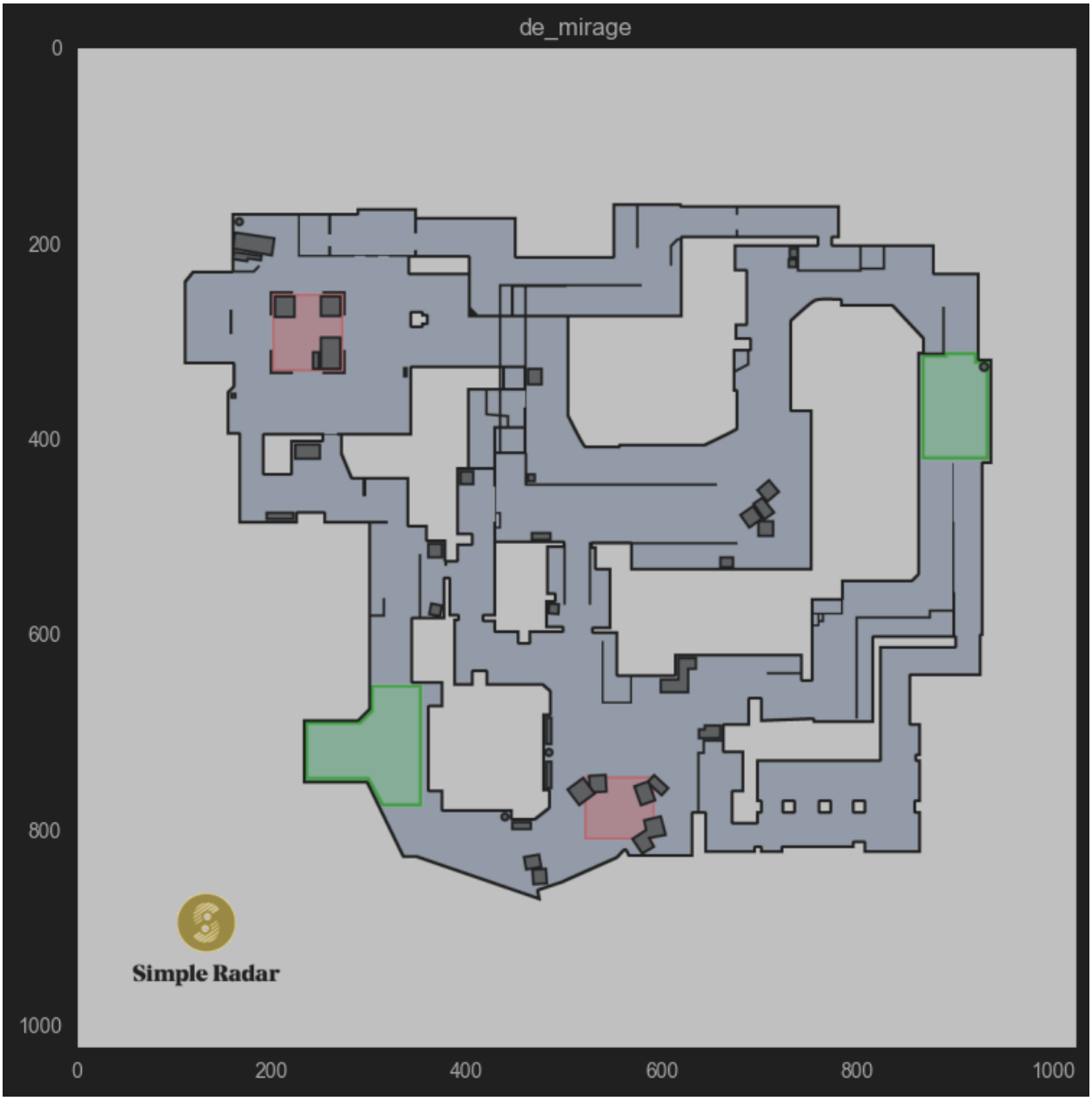

Figure A.4: Mini-map radar view of the map Mirage with spawn locations in green and bomb-sites in red.

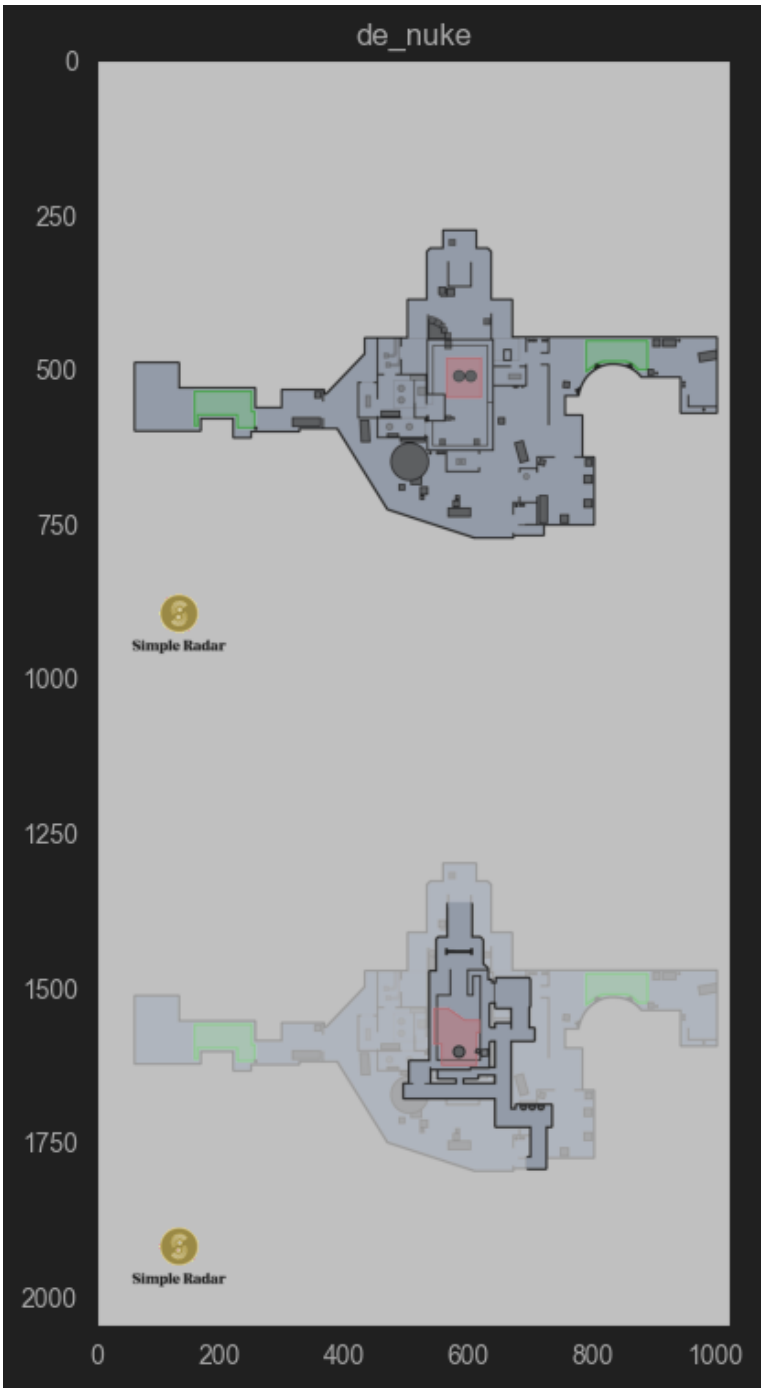

Figure A.5: Mini-map radar view of the map Nuke with spawn locations in green and bomb-sites in red with vertically stacked sections separated for clarity.

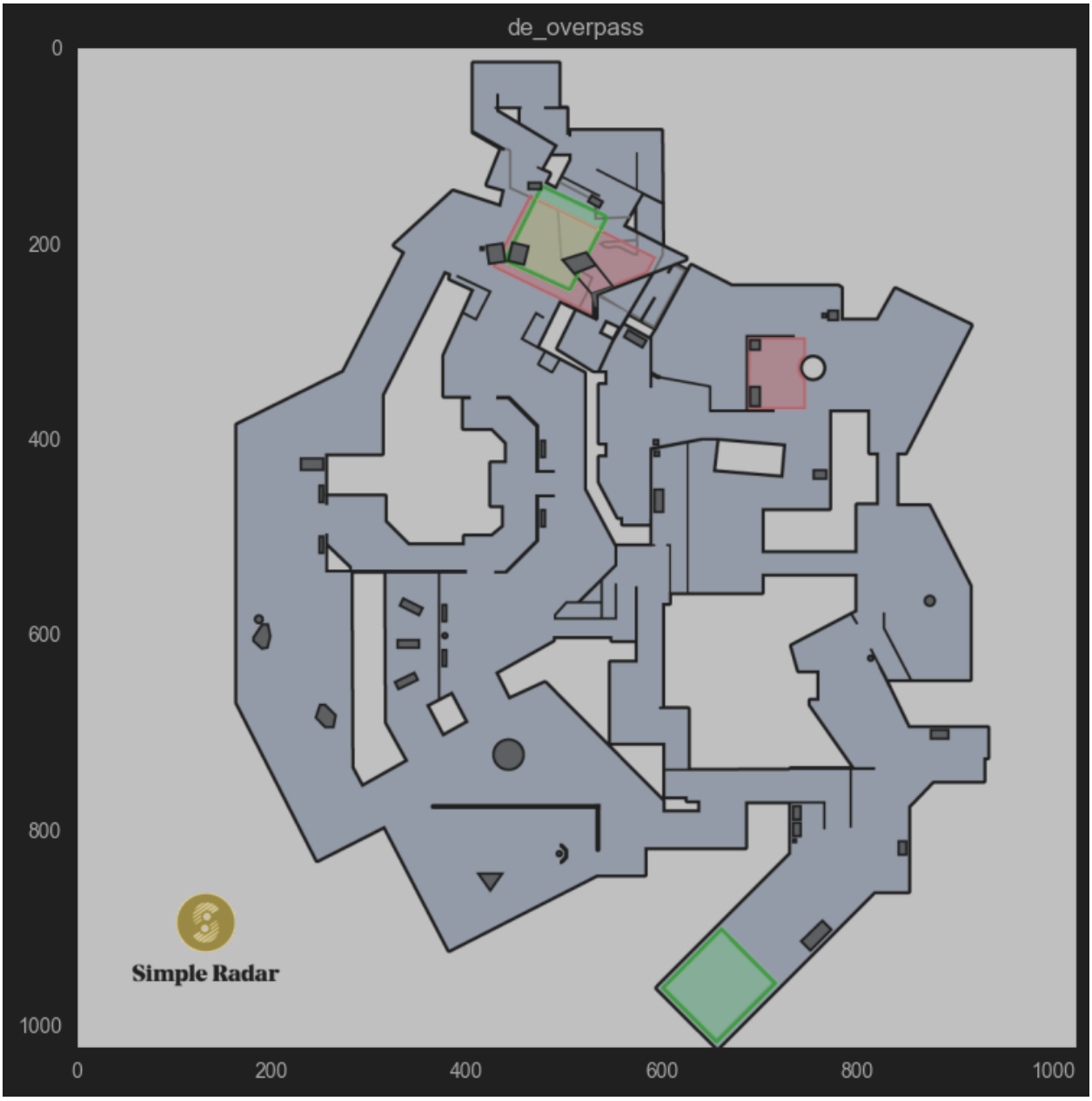

Figure A.6: Mini-map radar view of the map Overpass with spawn locations in green and bombsites in red.

Figure A.7: Mini-map radar view of the map Vertigo with spawn locations in green and bomb-sites in red with vertically stacked sections separated for clarity.

Bibliography

[1] D. J. Gagnon, L. Swanson, and E. Harpstead, "Open Game Data: Defining a Pipeline and Standards for Educational Data Mining and Learning Analytics with Video Game Data," in *2024 IEEE Conference on Games (CoG)*, (Milan, Italy), pp. 1–8, IEEE, Aug. 2024.

[2] S. S. Farooq, J.-W. Baek, and K. Kim, "Interpreting behaviors of mobile game players from in-game data and context logs," in *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, (Tainan, Taiwan), pp. 548–549, IEEE, Aug. 2015.

[3] A. De Gloria, ed., *Games and Learning Alliance: Second International Conference, GALA 2013, Paris, France, October 23-25, 2013, Revised Selected Papers*, vol. 8605 of *Lecture Notes in Computer Science*. Cham: Springer International Publishing, 2014.

[4] V. Chigateri, W. P. Puthran, G. Attigeri, S. Kolekar, and S. Vobugari, "Player Pattern Prediction Using Action Logs of Players," in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, (Bangalore, India), pp. 1–6, IEEE, Oct. 2021.

[5] S. Liu, C. Ballinger, and S. J. Louis, "Player Identification from RTS Game Replays."

[6] M. Tsikerdekis, S. Barret, R. Hansen, M. Klein, J. Orritt, and J. Whitmore, "Efficient Deep Learning Bot Detection in Games Using Time Windows and Long Short-Term Memory (LSTM)," *IEEE Access*, vol. 8, pp. 195763–195771, 2020.

[7] Hsing-Kuo Pao, Kuan-Ta Chen, and Hong-Chung Chang, "Game Bot Detection via Avatar Trajectory Analysis," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, pp. 162–175, Sept. 2010.

[8] S. Yeung, J. Lui, Jiangchuan Liu, and J. Yan, "Detecting cheaters for multiplayer games: theory, design and implementation," in *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.*, vol. 2, (Las Vegas, NV, USA), pp. 1178–1182, IEEE, 2006.

[9] S. Alkhalifa, "Machine Learning and Anti-Cheating in FPS Games," 2016.

[10] M. S. Islam, B. Dong, S. Chandra, L. Khan, and B. M. Thuraisingham, "GCI: A GPU Based Transfer Learning Approach for Detecting Cheats of Computer Game," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020.

[11] D. Kotkov, G. Pandey, and A. Semenov, "Gaming Bot Detection: A Systematic Literature Review," in *Computational Data and Social Networks* (X. Chen, A. Sen, W. W. Li, and M. T. Thai, eds.), vol. 11280, pp. 247–258, Cham: Springer International Publishing, 2018. Series Title: Lecture Notes in Computer Science.

[12] B. Cowley, D. Charles, M. Black, and R. Hickey, "Real-time rule-based classification of player types in computer games," *User Modeling and User-Adapted Interaction*, vol. 23, pp. 489–526, Nov. 2013.

[13] H. Barbosa, M. Barthelemy, G. Ghoshal, C. R. James, M. Lenormand, T. Louail, R. Menezes, J. J. Ramasco, F. Simini, and M. Tomasini, "Human mobility: Models and applications," *Physics Reports*, vol. 734, pp. 1–74, Mar. 2018.

[14] D. Brockmann, L. Hufnagel, and T. Geisel, "The scaling laws of human travel," *Nature*, vol. 439, pp. 462–465, Jan. 2006.

[15] M. Dicke and P. A. Burrough, "Using fractal dimensions for characterizing tortuosity of animal trails," *Physiological Entomology*, vol. 13, pp. 393–398, Dec. 1988.

[16] G. Solmaz and D. Turgut, "A Survey of Human Mobility Models," *IEEE Access*, vol. 7, pp. 125711–125731, 2019.

[17] R. Zhang, K. G. Stanley, D. Fuller, and S. Bell, "Differentiating Population Spatial Behavior Using Representative Features of Geospatial Mobility (ReFGeM)," *ACM Transactions on Spatial Algorithms and Systems*, vol. 6, pp. 1–25, feb 2020.

[18] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of Predictability in Human Mobility," *Science*, vol. 327, pp. 1018–1021, Feb. 2010.

[19] T. K. Rasmussen, J. B. Ingvardson, K. Halldórsdóttir, and O. A. Nielsen, "Improved methods to deduct trip legs and mode from travel surveys using wearable GPS devices: A case study from the Greater Copenhagen area," *Computers, Environment and Urban Systems*, vol. 54, pp. 301–313, Nov. 2015.

[20] C. t. C.-S. Wiki, "Counter strike: Global offensive."

[21] simply_sara, "Steam Community :: Guide :: The Ultimate CS:GO Comprehensive Guide — steamcommunity.com." https://steamcommunity.com/sharedfiles/filedetails/?id=3047206901. [Accessed 15-08-2025].

[22] D. Loudmouth, "All Player Roles in CS:GO – Full overview — cs.money." https://cs.money/blog/esports/all-player-roles-in-csgo/. [Accessed 23-09-2025].

[23] L. Serafico, "Counter-Strike roles explained (and who does them best) — esportsinsider.com." https://esportsinsider.com/counter-strike-roles-explained. [Accessed 23-09-2025].

[24] P. Xenopoulos and C. Silva, "ESTA: An Esports Trajectory and Action Dataset," 2022.

[25] S. Openshaw, *The modifiable areal unit problem*. No. no. 38 in Concepts and techniques in modern geography, Norwick [Norfolk]: Geo Books, 1983.

[26] C. Song, T. Koren, P. Wang, and A.-L. Barabási, "Modelling the scaling properties of human mobility," *Nature Physics*, vol. 6, pp. 818–823, Sept. 2010. Publisher: Springer Science and Business Media LLC.

[27] T. Paul, K. G. Stanley, and N. D. Osgood, "Multiscale entropy rate analysis of complex mobile agents," *Royal Society Open Science*, vol. 5, p. 180488, Oct. 2018.

[28] Caetano Traina Jr., Agma Traina, Leejay Wu, and Christos Faloutsos, "Fast feature selection using fractal dimension," *Journal of Information and Data Management*, vol. 1, pp. 3–16, May 2010.

[29] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[30] M. Patacchiola, "Gaussian mixture models," 2020.

[31] "Recursive feature elimination — scikit-learn.org." https://scikit-learn.org/stable/auto_examples/feature_selection/plot_rfe_digits.html. [Accessed 24-09-2025].

[32] "RFE — scikit-learn.org." https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html. [Accessed 24-09-2025].

[33] M. Lin, W.-J. Hsu, and Z. Q. Lee, "Predictability of individuals' mobility with high-resolution positioning data," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, (Pittsburgh Pennsylvania), pp. 381–390, ACM, Sept. 2012.

[34] R. Zhang, K. G. Stanley, S. Bell, and D. Fuller, "A feature set for spatial behavior characterization," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '18, pp. 512–515, ACM, Nov. 2018.

[35] P. M. Torrens, A. Nara, X. Li, H. Zhu, W. A. Griffin, and S. B. Brown, "An extensible simulation environment and movement metrics for testing walking behavior in agent-based models," *Computers, Environment and Urban Systems*, vol. 36, pp. 1–17, Jan. 2012.

[36] D. Coughlin, J. Strickler, and B. Sanderson, "Swimming and search behaviour in clownfish, Amphiprion perideraion, larvae," *Animal Behaviour*, vol. 44, pp. 427–440, Sept. 1992.

[37] Z. Zhong, H. Zhang, J. Ozaki, Y. Zhou, X. Zhao, D. Dan, and C. Wang, "A comprehensive methodological review of human mobility simulation and modelling: Current trends, challenges, and future directions," *Physica A: Statistical Mechanics and its Applications*, vol. 674, p. 130791, Sept. 2025.

[38] R. Bartle, "HEARTS, CLUBS, DIAMONDS, SPADES: PLAYERS WHO SUIT MUDS."

[39] C. Bateman, R. Lowenhaupt, and L. Nacke, "Player Typology in Theory and Practice," in *Player Typology in Theory and Practice*, Jan. 2011.

[40] C. Eggert, M. Herrlich, J. Smeddinck, and R. Malaka, "Classification of Player Roles in the Team-Based Multi-player Game Dota 2," in *Entertainment Computing - ICEC 2015* (K. Chorianopoulos, M. Divitini, J. Baalsrud Hauge, L. Jaccheri, and R. Malaka, eds.), vol. 9353, pp. 112–125, Cham: Springer International Publishing, 2015. Series Title: Lecture Notes in Computer Science.

[41] F. Zimmer, M. Irvan, M. N. S. Perera, R. Tamponi, R. Kobayashi, and R. Shigetomi Yamaguchi, "Player Behavior Analysis for Predicting Player Identity Within Pairs in Esports Tournaments: A Case Study of Counter-Strike Using Binary Random Forest Classifier," 2025.

[42] J. P. Pinto, A. Pimenta, and P. Novais, "Deep learning and multivariate time series for cheat detection in video games," *Machine Learning*, vol. 110, pp. 3037–3057, Dec. 2021.

[43] N. Elbert, E. Paskovski, N. Stein, and C. M. Flath, "The Layers of Change: Multilevel Adaptations to Exogenous Game Patches in CS:GO," in *2023 IEEE Conference on Games (CoG)*, (Boston, MA, USA), pp. 1–4, IEEE, Aug. 2023.

[44] P. Xenopoulos, B. Coelho, and C. Silva, "Optimal Team Economic Decisions in Counter-Strike," Sept. 2021. arXiv:2109.12990 [cs].

[45] P. Xenopoulos, H. Doraiswamy, and C. Silva, "Valuing Player Actions in Counter-Strike: Global Offensive," in *2020 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2020.

[46] P. Xenopoulos, W. R. Freeman, and C. Silva, "Analyzing the Differences between Professional and Amateur Esports through Win Probability," in *Proceedings of the ACM Web Conference 2022*, (Virtual Event, Lyon France), pp. 3418–3427, ACM, Apr. 2022.

[47] D. Durst, F. Xie, V. Sarukkai, B. Shacklett, I. Frosio, C. Tessler, J. Kim, C. Taylor, G. Bernstein, S. Choudhury, P. Hanrahan, and K. Fatahalian, "Learning to Move Like Professional Counter-Strike Players," 2024.

[48] D. Bednárek, M. Krulis, J. Yaghob, and F. Zavoral, "Data Preprocessing of eSport Game Records - Counter-Strike: Global Offensive," in *Proceedings of the 6th International Conference on Data Science, Technology and Applications*, pp. 269–276, SCITEPRESS - Science and Technology Publications, 2017.

[49] L. Serafico, "Is Dust 2 still competitive? Analysing Counter-Strike's iconic map — esportsinsider.com." `https://esportsinsider.com/counter-strike-dust-2-map-analysis`. [Accessed 26-08-2025].

[50] "Top Riflers in Counter-Strike History: The Greatest Players of All Time. – blix.gg — blix.gg." `https://blix.gg/news/cs-2/the-5-best-riflers-in-csgo-history/`. [Accessed 26-08-2025].