

Machine Learning in Cycling

Investigations into Classification, Computer Vision and Language Modelling in Cycling Aerodynamics

First Supervisor: **Dr. Stuart Gibson**

Second Supervisor: **Prof. James Hopker**

UNIVERSITY OF KENT
SCHOOL OF ENGINEERING, MATHEMATICS AND PHYSICS

A THESIS SUBMITTED TO
THE UNIVERSITY OF KENT
IN THE SUBJECT OF PHYSICS
FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY.

By
Callum J. Barnes

December 2025

Acknowledgements

The past three years have been an extraordinary journey, etched with challenges and unforgettable memories - from the late nights spent crafting this thesis to personal highlights like 'running' alongside Olympic and Ironman champions in the awe-inspiring Grand Canyon. This PhD has been a profound undertaking, unlike any other I have experienced.

My deepest gratitude goes first to my PhD supervisor, Dr. Stuart Gibson, for his unwavering guidance and support. His assistance ranged from securing the necessary equipment for my studies to dedicating time for thesis reviews, and even included what I consider the best supervisory meeting ever: a memorable cycle through the Kent Downs, punctuated by chats about bikes. My thanks also go to my Second Supervisor, Professor James Hopker, whose expertise and guidance in the world of cycling and sport proved indispensable throughout my research.

A special thank you is also owed to Eric DeGolier, CEO of Body Rocket Ltd. Without his invaluable support and mentorship, this PhD would not have been nearly as enriching, educational, or rewarding. His involvement has afforded me incredible opportunities, from travelling to the peaks of the Pyrenees, the captivating coast of the Côte d'Azur, and the arid landscapes of Arizona. However, the travel is secondary to the profound reward of being connected to the Body Rocket team and the athletes with whom I have worked; my thanks extends to every one

of them. Specifically, I extend my deepest appreciation to every member of the Body Rocket team who supported my PhD journey: Neil Newell, Anna Cibinel, Len Fox, Jordan Bolland, Conor Doyle, Jason Fitchew, and Matt Hutchings and Body Rocket athletes; Kristian Blummenfelt, Gustav Iden, Lance Childs and Sam Barley - thank you.

I would also like to thank Dr. Spartico Landi for his early-stage assistance, when I frequently sought out the teaching lab to use the soldering iron, borrow Raspberry Pis, and obtain advice, all culminating in the creation of the now-famed ShuffleCam. Further thanks are due to Dr. Chris Fennell for his dedication as a willing pilot athlete, and to Dr. Jake Bowd for his support in the biomechanics lab.

The appreciation, of course, is not confined solely to my professional network. A PhD is deeply personal, and I must express my sincere gratitude to all my fellow PhD candidates in the Ingram building. Thank you for your support, whether it was during a K-bar Friday evening or through our shared bonding over the struggles and triumphs of our doctoral journeys.

Beyond my academic and professional circles, I express my deepest gratitude to my family. To my parents, thank you for your unwavering love, encouragement, and belief in me throughout this journey. Your support has been a constant source of strength during both the most challenging and rewarding moments. To my sister Jasmine, thank you for your humour and for always reminding me of the world beyond the PhD with chats about coffee or a simple bike ride.

Finally, my heartfelt thanks go to my closest friends, Harry and Tom, who have been absolute pillars of sanity throughout this entire process from providing an excuse to travel up to Nottingham for a conference to simply taking a trip to the pub to talk about life. Without this steadfast support in my personal life, this thesis would not have achieved the same calibre.

Abstract

Machine learning has been applied extensively within cycling and sports literature. However, prior to this thesis, there has been a notable lack of research exploring the application of these powerful data analysis techniques within cycling aerodynamics.

This thesis pursues four areas of novel research, investigations into Shuffling, the classification of rider position on the bike, the estimation of a rider's pose on the bike through the use of computer vision and the application of Language Modelling within the domain of cycling. Each area of research is of interest; however, the motivation for these studies was inspired by the Body Rocket system, with the overarching goal of lowering the barrier of entry at the professional level. The Body Rocket system is a novel drag-force measurement device from Body Rocket Ltd (Sussex, UK) and comprises force sensors at the contact points of a bike, enabling direct measurement of a rider's drag force alongside other performance metrics. As a result, the first study aimed to validate one of these performance metrics: a rider's position on the saddle. After validating a rider's position on the saddle, the second study applied machine learning to all Body Rocket data to determine an athlete's full position on the bike. The third study took the second study a step further, moving away from the Body Rocket system as the data acquisition tool and using cameras to investigate continuous identification of a rider's pose on the bike using computer vision. The final study has the potential

to integrate all of these studies, as each alone could require interpretation by a coach or professional. As a result, the final study aimed to investigate whether, using Language Modelling, a domain-specific chatbot for cycling aerodynamics could be created and evaluated. All of these studies demonstrate the potential of these applications and are preliminary works.

This thesis pursues four areas of novel research:

1. The impact of shuffling on cycling aerodynamics and power output using a novel computer vision system.

2. The application of machine learning classifiers to data collected from a novel on-bike drag measurement system.

3. The application and fine-tuning of computer vision pose models in both upright and time trial cycling positions.

4. The use of language modelling as a personal aerodynamics coach.

In the first study, five riders completed a 4 km pursuit effort on bikes set up to their personal preferences and equipped with the Body Rocket drag measurement system. The system also has the potential to measure a rider's position on the saddle. The first aim of the study was to validate this capability using a purpose-built infrared imaging system. Across all riders, the system detected shuffling and showed good positional agreement for most individuals.

The second aim was to determine the effect of shuffling on C_dA and power output, evaluated in a cohort of ten athletes. Results indicate that the impact of shuffling may be rider-specific. Some riders experienced increases in C_dA and reductions in power during each shuffle, while others showed the opposite pattern.

The second study explored the application of machine learning, using data from the Body Rocket system, to predict a cyclist's position. Data was collected in

both the wind tunnel and outdoor settings. The classification task examined three machine learning models: the Gaussian Mixture Model (GMM), Support Vector Machine (SVM) and Gradient Boosted Machine (GBM). Model performance evaluation yielded high test accuracies for GMM (89.54%), SVM (96.00%) and GBM (100%). However, when the GBM model was applied to unseen wind tunnel data, its accuracy decreased from 100% on the training data to 45%. Notably, a separate GBM model trained solely on outdoor data achieved 100% accuracy.

In the third area of study, the capability of the YOLOv8 pose estimation model for determining cyclist joint locations in road bike and time trial positions was investigated, considering variations in apparel and background. Three riders adopted four distinct cycling positions. Fine-tuning for only 5 epochs on a bespoke dataset achieved adequate accuracy for the intended application.

The fourth study of this thesis explored the potential of Language Models (LMs) in the field of cycling aerodynamics. A novel ground truth was created by aggregating the consensus of a cohort of experts in cycling aerodynamics, bike fit, and biomechanics. The investigation tested 9 Large Language Models and 1 Large Reasoning Model. These included 7 open-source models from major developers such as Google, Meta, Microsoft, and Alibaba, alongside closed-source models from OpenAI. Models were evaluated under various conditions: without a system prompt, with a system prompt, with Retrieval Augmented Generation (RAG) using an enthusiast-level knowledge base, and with RAG using a more technical knowledge base. The top performer was OpenAI's GPT-4o, achieving an average score of $(90 \pm 0.41)\%$. The highest-scoring open-source model was Alibaba's Qwen2.5:32b, which scored an average of $(88.73 \pm 0.29)\%$ when used with a system prompt and the technical knowledge base. The findings indicate that it is feasible to develop an LM capable of performing at a level comparable to a human expert in cycling aerodynamics, bike fit, and biomechanics. Furthermore, this research

proposes an experimental methodology for quantifying the performance improvements an athlete can achieve through the assistance of a domain-specific Large Language Model.

Foreword

Publications arising from this thesis

Barnes, C., Hopker, J. and Gibson, S. (2023). To shuffle or not to shuffle. *Journal of Science and Cycling*, 12(2), pp. 28–30.

Barnes, C., Hopker, J. and Gibson, S. (2024). Artificial intelligence in cycling–live positional tracking using on-bike aerodynamic sensors. *Journal of Science and Cycling*, 13(2), pp. 41–48.

Barnes, C., Hopker, J. and Gibson, S. (2025). Applications of language modelling for a cycling aerodynamics' coach. *Journal of Science and Cycling*, 14(2), pp. 25–25.

Publications relating to this thesis

Hopker, J. G., Barnes, C., Fennell, C. R. and Gibson, S. (2025). Validity and reliability of an on-bike sensor system for the determination of aerodynamic drag in cycling. *European Journal of Sport Science, 25(12), Article. e70098.*

Contents

Acknowledgements **ii**

Abstract **iv**

Foreword **viii**

Contents **ix**

List of Tables **xii**

List of Figures **xiv**

List of Algorithms **xxvii**

1 Introduction **1**

2 Background **7**

2.1 The Physics of Cycling 7

2.2 Aerodynamic Testing within Cycling 24

2.3 Machine Learning within Cycling 37

2.4 Machine Learning and Artificial Intelligence 41

3 Shuffling **49**

3.1 Introduction to Shuffling 49

3.2 System Design and Image Processing Pipeline 51

3.3 Body Rocket System Validation 67

3.4 Results 71

3.5 Discussion 74

3.6 Conclusion 83

4 Positional Classification using the Body Rocket System **86**

4.1 Introduction 86

4.2 Theory 90

4.2.1 Gaussian Mixture Model 90

4.2.2 Support Vector Machines 91

4.2.3 Gradient Boosted Machine 93

4.3 Methods 99

4.4 Results 102

4.5 Discussion 114

4.6 Conclusions 118

5 Pose Estimation in Cycling **120**

5.1 Introduction 120

5.2 Theory 122

5.3 Materials and Methods 129
5.4 Results 134
5.5 Discussion 144
5.6 Conclusion 154

6 Application of Large Language Modelling for a Personal Aerodynamics' Coach 157

6.1 Introduction 158
6.2 Theory 161
6.3 Materials and Methods 171
6.4 Results 176
6.5 Discussion 183
6.6 Conclusions 186

7 Conclusion 188

7.1 Thesis Aim and Context 188
7.2 Summary of Key Findings 189
7.3 Future Work and Closing Statement 192

Bibliography 197

List of Tables

1 Drag Coefficients (Cd) for Common Shapes 18

2 Table of errors comparing Hough Circles and Template Matching 66

3 Table of errors containing the results for each riders Body Rocket system data and the Infrared camera. 69

4 Table of average powers and C_dA for each rider, alongside the average difference when a rider shuffles and the number of shuffles for each rider. 73

5 Train and test accuracy for the four different GMM covariance types investigated in this study (spherical, diagonal, tied and full) on post-normalisation and PCA data. 106

6 Train and test accuracy for the four different GMM covariance types investigated in this study (spherical, diagonal, tied and full) on unnormalised data without a PCA. 106

7 Train and test accuracy after modal moving average filtering. . . . 107

8 Table showing the training and test scores for the four different SVM kernel types investigated in this study (LinearSVC, SVC (linear), SVC (RBF) and SVC (poly)). 109

9 the four different SVM kernel types investigated in this study (LinearSVC, SVC (linear), SVC (RBF) and SVC (poly)) with a 6-second modal moving average applied. 110

10 Average estimation errors for the out-of-the-box unfine-tuned YOLOv8 pose estimation model. 136

11 Table showing the average estimation errors for the fine-tuned YOLOv8 pose estimation model over 5-epochs. 140

12 Table of results showing all of the results obtained from all the different Language Models tested in this study, with each respective configuration, number of iterations, standard deviation in the mark, standard error of the mark and coefficient of variation. 222

List of Figures

1.1 An image showing a bike equipped with the Body Rocket system. With red circles indicating the location of the sensors and arrows showing an example direction, which can be measured, specifically F_x. 2

1.2 Two images showing the markers placed on the posterior of the rider (left) and the mounting location of the infrared camera used to validate the Body Rocket system's ability to determine the position of the rider on the saddle. 3

1.3 Diagram showing an example of vertical force vectors on the saddle specifically (left) and across all contact points on the bike (right). 4

1.4 Diagram showing the position of a rider on the bike and the vertical forces applied at the contact points (left) and a kinematic model applied to a rider on the bike with their joint locations identified (right). 5

1.5 An example diagram showing outputs from three of the studies in this thesis and showing how information from them could be interpreted by a Large Language Model expert and provide protocols and insight. 6

2.1 [REDACTED] 8

2.2 [REDACTED] 11

2.3 [REDACTED] 12

2.4 A diagram showing the gravitational component of a cyclist on a slope. 13

2.5 Image of a cyclist in the University of Southampton wind tunnel. 16

2.6 Three different cycling helmets, (left) the MET drone wide body (MET-Helmets 2025), (Middle) HJC Adwatt (HJC Helmets 2025) and (right) POC Omne Beacon (POC Sports 2025) 19

2.7 An image of a rider wearing a skinsuit with a focus on a specific design element of the skinsuit. 21

2.8 [REDACTED] 22

2.9 [REDACTED] 25

2.10 [REDACTED] 28

2.11 [REDACTED] 32

2.12 Two images. Left showing the Body Rocket pedals, handle bar and saddle drag force sensors and airspeed sensor and right showing the Body Rocket sensors mounted to a bike with the drag force sensors highlighted in red circles. 34

2.13 Free body diagram showing the bike frame forces $F_{x,\text{bike}}$ and $F_{y,\text{bike}}$, the weight mg, and the inertial term ma. 35

2.14 Plot showing an example of a logistic regression plot. The red data points represent the class 0 and the blue represent the class 1. The green line shows the decision boundary. 42

2.15 An example of a Receiver Operator Characteristic curve with a number of examples. 43

2.16 Diagram of a computational perceptron showing from left to right: the input, weights, summation, activation function and output. . . 44

2.17 **Multilayer Perceptron (MLP)** architecture illustrating input, three hidden layers, and output layers with connections representing forward propagation. 45

3.1 Diagram of the Body Rocket load cell showing the respective forces and moments it is subject to when a rider is on the saddle. 51

3.2 Image of an early Raspberry Pi Model B 53

3.3 Image of a Raspberry Pi zero 2 W 53

3.4 Exploded diagram of the final iteration of the infrared camera. This design and all previous designs were developed by the principal investigator. 55

3.5 [REDACTED] 55

3.6 Image output from the infrared camera showing three retro-reflective markers in a line and adhered to the rear of a rider. 56

3.7 Plots showing how from an original image a Canny edge can be applied. Circles of various sizes can be drawn on the edge and where the peaks are identified the centre of the circles are found. . 57

3.8 Output image from camera with the Hough circle method applied alongside the steps to achieve the output image. 58

3.9 Output image from camera with the template matching method applied. 62

3.10 Diagram of the infrared camera with the blue shape representing the field of view of the camera. And the variables β_1, β, x_c and d showing the angle subtended by the retro reflective marker, the field of view of the infrared camera, the distance between the marker and the camera and the diameter of the ball, respectively. 63

3.11 Diagram of the calibration/validation set-up depicting the infrared camera, a ruler and the retro-reflective marker placed at different intervals away from the camera. 64

3.12 Plot showing the distance markers were from the camera, using the Hough circle method. The green trace shows the raw reading from the hough circle estimation, the red trace is with a 5 second rolling average applied and the '+' signs are the distance checkpoints. . . 65

3.13 Plot showing the distance markers were from the camera, using the template matching method. The green trace shows the raw reading from the template matching estimation, the red trace is with a 5 second rolling average applied and the '+' signs are the distance checkpoints. 65

3.14 Diagram of a bike equipped with the infrared camera and the red circles representing the locations of the Body Rocket sensors. . . . 68

3.15 Images showing the two versions of the infrared camera. 69

3.16 Plot showing the output from the Body Rocket system (cyan trace) and the output from the infrared camera (red trace). 70

3.17 Rider 2 saddle position (Red trace) plotted with power (black trace) and identified shuffles (blue). 71

3.18 Rider 2 saddle position (Red trace) plotted with C_dA (black trace) and identified shuffles (blue). 71

3.19 Rider 3 saddle position (Red trace) plotted with power (black trace) and identified shuffles (blue). 72

3.20 Rider 3 saddle position (Red trace) plotted with C_dA (black trace) and identified shuffles (blue). 72

3.21 Rider 5 saddle position (Red trace) plotted with power (black trace) and identified shuffles (blue). 72

3.22 Rider 5 saddle position (Red trace) plotted with C_dA (black trace) and identified shuffles (blue). 72

3.23 Relationship between each rider's average shuffle rate and average change in power during shuffling. Each point represents one rider, with rider identity indicated by colour in the legend. Error bars show the uncertainty in the average power change. The red line shows the fitted linear trend. 78

3.24 Relationship between each rider's average shuffle size and average change in power during shuffling. Each point represents one rider, with rider identity indicated by colour in the legend. Error bars show the uncertainty in the average power change and the average shuffle size. The red line shows the fitted linear trend. 79

3.25 Relationship between each rider's average shuffle rate and average change in C_dA during shuffling. Each point represents one rider, with rider identity indicated by colour in the legend. Error bars show the uncertainty in average C_dA change. The red line shows the fitted linear trend. 81

3.26 Relationship between each rider's average shuffle size and average change in C_dA during shuffling. Each point represents one rider, with rider identity indicated by colour in the legend. Error bars show the uncertainty in average power change and average shuffle size. The red line shows the fitted linear trend. 82

4.1 An image of a Body Rocket-equipped bicycle with red arrows indicating the forces and moments that can be measured on the system. 87

4.2 An example of edge detection being used in a wind tunnel environment. 89

4.3 Simple example of how a linear Support Vector Machine works with regard to the classification of data. 93

4.4 Example plot showing log loss against boosting iterations for a training set (orange trace) and testing set (yellow trace) with early stopping (red dashed trace) at 175 boosting iterations. 96

4.5 Illustration of 5-fold cross-validation with an additional independent holdout dataset reserved for final model evaluation. 97

4.6 Two correlation matrices, before (left) and after (right) the dataset has undergone a principal component analysis. 98

4.7 Images from WT1 of a rider adopting four distinct cycling positions. From left to right the positions are: baseline, head-down, head-up and hands on the aeropads. 99

4.8 Plots showing the weight distributions of the different positions on the contact points of the bike. 103

4.9 Plots showing the weight distributions at the different wind speeds on the contact points of the bike. 104

4.10 Correlation matrix showing the 8 selected features for model training before a PCA is applied. 105

4.11 Correlation matrix showing the 8 selected features for model training after a PCA has been applied. 105

4.12 Four plots of the Body Rocket positional data with the four different GMM covariance types investigated in this study (spherical, diagonal, tied and full). Plotted as position across time with a 6-second modal moving average applied to the prediction data (red trace), the raw prediction data (blue trace) and the ground truth (green trace). 107

4.13 A plot showing the four vector plane outputs from the Support Vector Machine, with the pale green regions indicating the decision boundaries. The triangles represent the data used for SVM training, and the circles represent the data used for testing. Purple represents position 1, blue represents position 2, green represents position 3 and yellow represents position 4. 108

4.14 Four plots of the Body Rocket positional data with the four different SVM kernel types investigated in this study (LinearSVC, SVC (linear), SVC (RBF) and SVC (poly)). Plotted as position across time with a 6-second modal moving average applied to the prediction data (red trace), the raw prediction data (blue trace) and the ground truth (green trace). 109

4.15 Rider position in WT1 during the test, with the rider transitioning between positions 1 and 4 on three occasions. The ground truth position is shown in green, the raw output from the trained Gradient Boosted Machine in blue, and the GBM output after applying a 6-second moving modal average is shown in red. 111

4.16 Rider position in WT2 during the test, where the rider transitions between positions 1 and 4 on two occasions. The ground truth position is shown in green, the raw output from the trained Gradient Boosted Machine in blue, and the GBM output after applying a 6-second moving modal average is shown in red. 111

4.17 Prediction data from the Gradient Boosted Machine trained on data collected from Rider 1 during OD riding, where the rider transitions between positions 1 and 4 over a $\approx$ 450-second period. The true position is shown in green, the raw GBM output in blue, and the GBM output after applying a 6-second moving modal average is shown in red. 112

4.18 Prediction data from a Gradient Boosted Machine trained on data collected from Rider 1 during OD riding, where the rider moves between positions 1 and 4 over a $\approx$ 500 second interval. The true position is shown in green, the raw GBM output in blue, and the GBM output after applying a 6-second moving modal average is shown in red. 113

4.19 Table of performance metrics (log loss, mean squared error, classification success and number of trees) from the models produced in WT1 and WT2 and from the OD velodrome data 113

5.1 [REDACTED] 123

5.2 An example of a YOLO detection model detecting a bicycle (the pink bounding box) and a person (the red bounding box) in an image.125

5.3 [REDACTED] 126

5.4 COCO human pose keypoints shown in frontal view and rider's right side profile as used in this study. 126

5.5 An example of IoU for YOLOv8 128

5.6 An image of the experimental setup with one of the research participants. In this image, the participant has markers placed in the same locations as those identified by the pose estimation and aligned with the COCO standard. 129

5.7 The three different cycling helmets in this study, (left) the MET drone wide body (MET-Helmets 2025), (Middle) HJC Adwatt (HJC Helmets 2025) and (right) POC Omne Beacon (POC Sports 2025) 130

5.8 The four road bike positions undertaken in this study 132

5.9 The four time trial positions undertaken in this study 132

5.10 The unfine-tuned YOLOv8 outputs for road bike positions 1 to 4 (red trace) overlaid with the ground truth (cyan trace). 134

5.11 Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the unfine-tuned YOLOv8 model. 134

5.12 The unfine-tuned YOLOv8 outputs for time trial positions 1 to 4 (red trace) overlaid with the ground truth (cyan trace). 135

5.13 Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the unfine-tuned YOLOv8 model in the time trial position sequence. 136

5.14 Rider 3 in the four road bike positions with the ground truth (cyan trace) and the YOLOv8 output fine-tuned over 5-epochs (red trace).137

5.15 Plot showing the Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the fine-tuned YOLOv8 model 138

5.16 The 5-epoch fine-tuned YOLOv8 outputs for the time trial positions 1 through to 4 (the red trace) overlaid with the ground truth (cyan trace) 139

5.17 Plot showing the Mean Per joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the fine-tuned YOLOv8 model 140

5.18 The 10-epoch fine-tuned YOLOv8 outputs for the road bike positions 1 through to 4 (the red trace) overlaid with the ground truth (cyan trace) 141

5.19 Plot showing the Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the fine-tuned YOLOv8 model over 10-epochs 142

5.20 The 10-epoch fine-tuned YOLOv8 outputs for the time trial positions 1 through to 4 (the red trace) overlaid with the ground truth (cyan trace) 143

5.21 Plot showing the Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the fine-tuned YOLOv8 model over 10-epochs 144

5.22 Plot showing the YOLOv8 calculated hip angle (red trace) compared to the motion capture ground truth (blue trace). For Rider 3 performing the road bike position sequence. 144

5.23 Plot showing the YOLOv8 calculated hip angle (red trace) compared to the motion capture ground truth (blue trace) for the first 6 seconds of the rider's session. 145

5.24 Plot showing the YOLOv8 (red trace) calculated hip angle compared to the motion capture ground truth (blue trace) for 4 seconds with the participant in position 2. 146

5.25 Four images of rider 3 at different positions in the pedal stroke, 0 degrees, 90 degrees, 180 degrees and 270 degrees. On each plot, the ground truth is plotted (cyan trace) alongside the YOLOv8 output (red trace). 147

5.26 Plot showing the YOLOv8 calculated hip angle (red trace) compared to the motion capture ground truth (blue trace). For Rider 3 performing the time trial bike position sequence. 148

5.27 Plot showing the YOLOv8 (red trace) calculated hip angle compared to the motion capture ground truth (blue trace) Over the first 6 seconds of Rider 3 performing the time trial bike position sequence. 149

5.28 Plot showing the fine-tuned YOLOv8 (Red trace) calculated hip angle compared to the motion capture ground truth (blue trace). For Rider 3 performing the time trial bike position sequence. . . . 150

5.29 Plot showing the fine-tuned YOLOv8 (Red trace) calculated hip angle compared to the motion capture ground truth (blue trace). This is over the first 6 seconds of Rider 3 performing the time trial bike position sequence. 151

5.30 Plot showing the fine-tuned YOLOv8 calculated hip angle (red trace) compared to the motion capture ground truth (blue trace). For Rider 3 performing the time trial bike position sequence. . . . 151

5.31 Plot showing the YOLOv8 calculated hip angle compared to the motion capture ground truth. For Rider 3 performing the time trial bike position sequence. 153

5.32 Plot showing the YOLOv8 calculated hip angle compared to the motion capture ground truth. For Rider 3 performing the time trial bike position sequence. 153

6.1 Tokenization of the sentence “The fluffy blue llama was riding a bike”. Each word is treated as a separate token. 162

6.2 Simplified LLM processing pipeline: tokenization, token and positional embeddings, repeated self-attention and MLP layers, and final output logits. 169

6.3 Visual representation of the prompt and response process used for each large language model to evaluate the knowledge of each model 174

6.4 Overview of the Retrieval-Augmented Generation (RAG) Pipeline with Model Details. 175

6.5 Bar chart comparing the performance of two llama models, llama3.1:8b and llama3.2:3b on the cycling aerodynamics assessment. 177

6.6 Bar chart comparing the performance of two Gemma2 models, Gemma2:9b and Gemma2:27b. 178

6.7 Bar chart showing the performance of the Microsoft Phi4:14b model against the cycling aerodynamics MCQ sheet. 179

6.8 Bar chart showing the performance of the Qwen2.5:7b and Qwen2.5:32b model from Alibaba against the cycling aerodynamics MCQ sheet 180

6.9 Bar chart showing the performance of three OpenAI models with and without a system prompt and their respective performance against the Cycling aerodynamics MCQ sheet 181

6.10 Bar chart of the best performing models in this study 182

6.11 Bar chart showing the performance of all the experts in the study. 182

6.12 Bar chart showing the performance of all the experts in the study. 183

List of Algorithms

1 Find three maxima with masking 61

2 Gradient Boosting Algorithm (J. H. Friedman) 94

3 Radius Optimization via Template Matching 221

Chapter 1

Introduction

Performance cycling can be summed up as the cutting edge of technology, where marginal gains are persistently pursued. This thesis aims to investigate how the boundary of entry into performance cycling can be lowered and provide new performance insights through the use of a novel aerodynamic measurement device, machine learning and artificial intelligence. The tools and studies presented in this thesis are novel preliminary works; however, they all act as proofs of concept, which can be built upon.

The research carried out in this thesis was in collaboration with Body Rocket Ltd (Sussex, UK). Body Rocket has developed an aerodynamic drag measurement system which allows for the direct measurement of a rider's drag, therefore allowing a rider to test different aerodynamic configurations and determine what is most aerodynamically optimal. The system is comprised of sensors on the handlebars, saddle and pedals. These are force-based sensors and are the foundation of this direct force drag measurement system. As they are designed to have a high level of sensitivity, they also measure other metrics alongside drag force, such as vertical force, saddle and handlebar moments and power. This can be visualised in Figure 1.1.



Figure 1.1: An image showing a bike equipped with the Body Rocket system. With red circles indicating the location of the sensors and arrows showing an example direction, which can be measured, specifically F_x.

Shuffling on the saddle is a phenomenon which is often seen in a negative light. Most fitters aim to reduce the amount one shuffles on the saddle due to comfort; however, there is little research which supports the idea that it reduces one's performance. The first study of this thesis utilises the Body Rocket system's ability to measure one's position on the saddle and therefore when a shuffle occurs. As a result, this study investigated the impact of shuffling on C_dA and power. Due to a lack of validation of the Body Rocket's system's ability to determine the position of a rider on the saddle, a validation device was created, an infrared camera-based system. The infrared camera-based system was comprised of a Raspberry Pi, an infrared camera and an infrared LED flash ring.



Figure 1.2: Two images showing the markers placed on the posterior of the rider (left) and the mounting location of the infrared camera used to validate the Body Rocket system's ability to determine the position of the rider on the saddle.

This flash ring was designed to be in a similar location to the camera (mounted radially to the lens), allowing for infrared light to reflect off the retro-reflective markers placed on the posterior of the riders sitting in a time trial position. Figure 1.2 shows how the infrared camera was mounted to the bike, and where the retroreflective markers were placed on the rider. With the size of the markers known alongside the specifications of the infrared camera, it is then possible to calculate the distance of the rider's posterior from the camera, enabling a direct comparison with the Body Rocket system's output. This study supports the core aim of this thesis, as it provides insight into a performance metric with little literature surrounding it whilst validating a novel device's ability to measure it.

As a natural progression from focusing on only the position of a rider on the saddle through only one of the Body Rocket sensors, the second study of this thesis looks to investigate the application of machine learning with data from all of the sensors of the Body Rocket system. More specifically, moving from validating its ability to determine the position of a rider on the saddle to determining whether it is possible to predict the position of a rider on the bike through the application of machine learning classifiers.

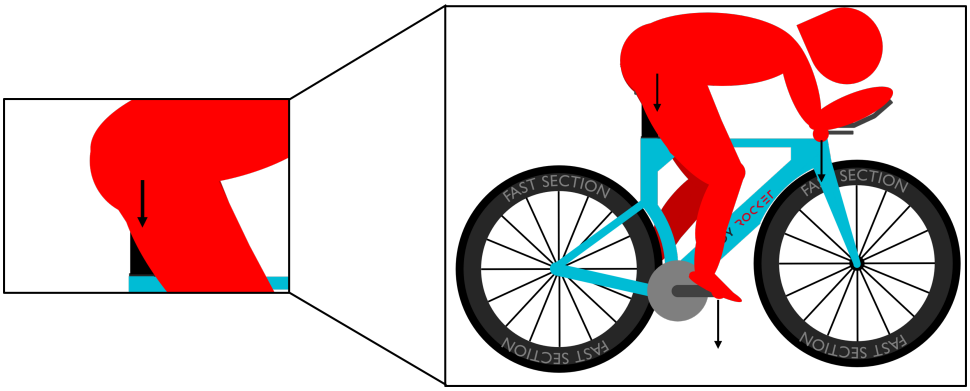


Figure 1.3: Diagram showing an example of vertical force vectors on the saddle specifically (left) and across all contact points on the bike (right).

Figure 1.3 shows an example of the vertical forces on the saddle, which help for the determination of the position of a rider on the saddle and the force vectors across the contact points of the bike. As would be expected, when a rider changes their position on the bike, e.g., shifting more weight to the saddle, the vertical force measured at the saddle would increase.

This study used data from a wind tunnel setting and data from outdoor conditions. With the data collected, three different machine learning classifiers were applied to the collected Body Rocket data: the Gaussian Mixture Model, Support Vector Machine and Gradient Boosted Machine. Each model was chosen as they were simple and is known to show clear human-readable motivations for the actions taken to classify the rider position. In other words, the models chosen were explainable. Not only were the models chosen explainable, they were also relatively simple in architecture, allowing for low-latency applications whilst live on the bike. If it were possible for these to be live, then it would allow for a rider to have live feedback whilst they are training or racing, allowing them to adopt the most appropriate position for a certain part of the course or race.

Following the application of machine learning classifiers to predict the position of a rider on the bike, the natural progression was to investigate the position of

a rider on the bike in a more continuous fashion, such as their joint angles and locations. A visual example of how this progresses from the classification study can be seen in Figure 1.4.

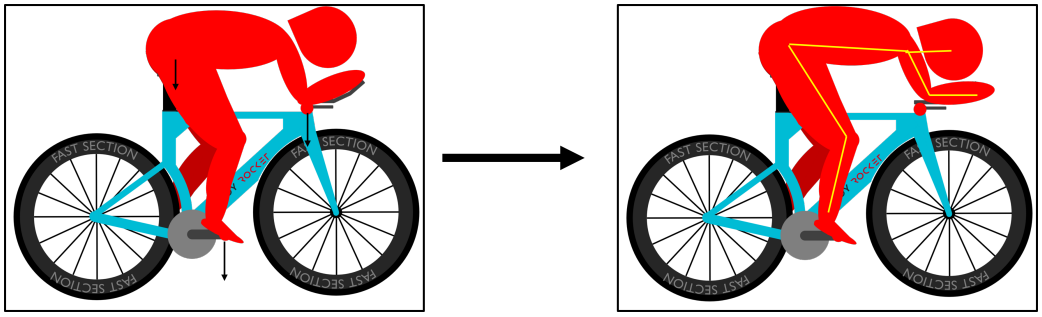


Figure 1.4: Diagram showing the position of a rider on the bike and the vertical forces applied at the contact points (left) and a kinematic model applied to a rider on the bike with their joint locations identified (right).

This study departed from the application of the Body Rocket system and moved towards the application of computer vision. The computer vision model used in this study was YOLOv8 and was applied out of the box and after fine-tuning using data captured in the motion capture studio. The aim of this study was to determine whether positional knowledge can be made accessible through the use of hardware that most users will have access to. In this case, the data was acquired on a laptop camera.

The final study of this thesis took a step back and realised how a number of these studies are great examples of technologies that can be used and made accessible; however, there is a limiting factor in terms of how to interpret that knowledge. In the case of real-world applications with professional-level individuals, they would seek the assistance of a human expert to help them analyse the data and prescribe what tests to perform next. This study poses the question: Can an AI Large Language Model (LLM) be domain-specific in this area of performance cycling and aerodynamics and perform to the level of a human expert? A visualisation of this can be seen in Figure 1.5.

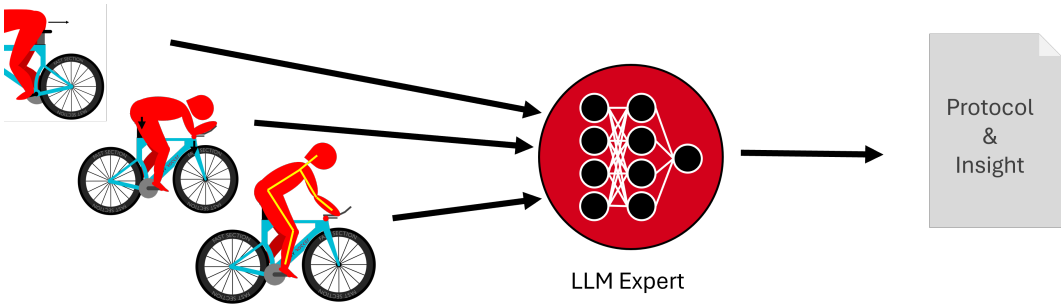


Figure 1.5: An example diagram showing outputs from three of the studies in this thesis and showing how information from them could be interpreted by a Large Language Model expert and provide protocols and insight.

This study recruited a cohort of experts from academia and industry to help produce a ground truth dataset. With this data, it was then possible to comparatively evaluate the performance of the LLMs against the human expert outputs.

All the studies within this thesis demonstrate, through these proof-of-concept works, that through the application of machine learning and artificial intelligence, paired with a novel aerodynamic measurement device, there is potential to lower the barrier of entry into professional-level sport and provide new performance insights.

Chapter 2

Background

Cycling performance can be assessed by breaking down the fundamental physics of cycling. This chapter will aim to show the physics of cycling and where some cycling losses are present, alongside a focus on cycling aerodynamics, measurement techniques and background on artificial intelligence and machine learning.

2.1 The Physics of Cycling

The bicycle is a versatile tool, whether it's for collecting groceries, commuting to work, or pushing the boundaries of human performance. Cycling itself is an interesting case study of how marginal gains, science, and research come together to push the human body faster than ever before. Within performance cycling, the key to performance is speed; the fastest rider wins the race. Although this statement is correct, this doesn't necessarily mean the fastest rider is the strongest physiologically. Cycling can be modelled as a relatively simple equation of motion (Martin et al. 1998):

$$m \cdot \frac{dv}{dt} = \frac{P_{\text{crank}} \cdot \eta}{v} - mgC_{rr}\cos\theta - mg\sin\theta - \frac{1}{2}\rho C_d A v^2 \tag{1}$$

The model in Equation (1) can be broken down into four components: drivetrain resistance, rolling resistance, gravity, and aerodynamic drag.

Drivetrain Efficiency

The term in Equation (1) which accounts for the drivetrain resistance is:

$$\frac{P_{\text{crank}} \cdot \eta}{v} \tag{2}$$

From Equation (2), it can be seen that this term accounts for the crank power P_{crank}, the speed v, and a coefficient of drivetrain efficiency η. Drive-train efficiency can be measured in a lab, where repeatable, reliable results can be obtained.



Figure 2.1: [REDACTED]

At the centre of drivetrain efficiency is friction, which needs to be overcome at the sprockets and rollers, as well as internally within the chain. Friction is defined as a force which opposes motion when two surfaces are in contact, it is caused by microscopic features at the contact point of the two surfaces called asperities,

alongside this in the case of static friction Van-der-Waal forces need to be overcome. Friction itself can be broken down into two components: static friction and dynamic friction. When in motion, the chain, pins, and rollers must overcome dynamic friction. Applying lubricants can greatly reduce this resistance.

With rigs such as in Figure 2.1 they work through measuring the input power and the output power. The difference between the input and output powers can then be taken as the drivetrain efficiency. There are numerous variables in the bike drivetrain system that can be tested for efficiency; these include chain pitch, the difference in chainring sizes, chain line, and other drivetrain metrics.

The external chain drive is standard on most bicycles due to its ease of maintenance and efficiency. A traditional drive train with a half-inch link separation has generally been found to have an efficiency of 95% - 98.5% (Lukes, Hart and Haake 2012). Studies such as those from Spicer et al. (2001) investigate the factors that impact drivetrain efficiency and conclude that chain tension and sprocket size are the primary factors which dominate overall chain efficiency. Traditional chains are constructed from pins, rollers, inner plates, and outer plates. All of these parts slide against one another, creating sliding friction. Studies such as Burgess (1998) investigate the impact of different sprocket sizes on chain efficiency and conclude that a larger sprockets have a higher efficiency than smaller sprockets. Chain tension is another factor which impacts the efficiency of the drivetrain system. Spicer (2013) investigated the impact of chain tension on drivetrain efficiency and concluded that as chain tension increases, power loss decreases.

The drivetrain standard on a bicycle has remained relatively unchanged for the last 100 years, since the invention of the safety bicycle and the later adoption of the ANSI 40 chain standard. This standard has become ubiquitous on most modern bicycles, from mountain bikes to road bikes and track racing bikes. There

have been a number of studies performed to further increase this efficiency, especially for drive train systems with multiple gears. This includes research into the best lubrication techniques (Kozlov, Egorov and Belogusev 2017), the impact of sprocket size (Barnaby, Yon and Burgess 2020) and the application of different drive train mechanisms such as shaft drives, belt drives or chains with a different pitch (Wragge-Morley et al. 2024).

Rolling Resistance

The term in Equation (1) which represents the rolling resistance is:

$$mgC_{rr}\cos\theta \tag{3}$$

As shown in Equation (3), several factors affect the rolling resistance a cyclist experiences in the wild. These include the mass, m, gravity g, the slope θ and the coefficient of rolling resistance C_{rr}. The majority of studies which focus on reducing the magnitude of Equation (3) focus on C_{rr}. C_{rr} can change with respect to tyre width, tyre compound, tyre pressure, tyre durometer, alongside other qualities of the tyre.

Tyres themselves are made of rubber. This is due to the rubber's ability to conform to the ground and therefore generate traction. This is important so that the bike can be propelled, and the amount of traction will depend on the tyre, the surface, and therefore the coefficient of static friction. The application of rubber has drawbacks, as it is not perfectly elastic and undergoes hysteresis. Hysteresis occurs when long-chain rubber molecules move relative to each other, generating heat.

When a tyre is rotating there is a contact patch on the ground, this contact patch changes size depending on the rubber used, the thread density, tyre pressure and

load on the tyre. At the front of the contact patch, the rubber and tyre are under compression, and at the rear of the contact patch, the rubber is performing expansion. Therefore, during rotation, the tyre undergoes a cycle of compression and expansion, and, following hysteresis, energy is lost as rolling resistance.

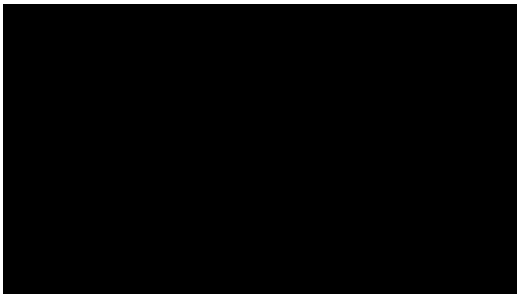


Figure 2.2: [REDACTED]

Equipment such as that shown in Figure 2.2 is commonly used to determine a tyre's rolling resistance. This example in-particular from the Silverstone sports engineering hub measures the difference in required power to spin a drum with and without a tyre in contact with it (Silverstone Sports Engineering Hub 2022). There have been a number of studies and methods used to determine rolling resistance, such as Grappe et al. (1999), which investigated the impact of different tyre pressures and system weight. In their study, the research was conducted in the field, where a rider was recruited to complete a coasting deceleration course. The tyre pressure was then incrementally changed, and the deceleration was measured. They also added another test to their study that used the deceleration method; however, they varied the tyre load by changing the rider's weight. In this study C_{rr} dropped by 62% from the lowest tested pressure and the highest tested pressure. In their research, they also found an increase in C_{rr} of 11.4% when an additional 15kg was added to the rider. Other methods also exist to determine tyre rolling resistance, such as the method presented by Henchoz et al. (2010). This method used a treadmill at an incline and a dynamometer attached to the bike. With the bike at an incline, a component of weight would be measured by the dynamometer;

however, with the treadmill moving, a component of rolling resistance would also be measured. In their study, this method was shown to be repeatable, as it detected the tyre type and inflation pressures.

Other Efficiency Measurement Methods

It is noteworthy that there exist additional systems which do not isolate the drivetrain and tyres but rather assess the bicycle in a more holistic manner. Among such systems is the Silverstone Sports Engineering Hub pedalling efficiency rig (Hub 2025b).



Figure 2.3: [REDACTED]

This system secures a bicycle to a large drum, employing sensors at the pedals and on the drum to measure the discrepancies between them, thereby providing an indication of the system's overall efficiency, including tyre and chain performance. This approach arises from the understanding that testing tyres and chains in isolation does not fully reflect the resistive forces cyclists encounter on the road (Dowd, Cavanaugh and Mansson 2024).

Gravity

Gravity is another factor that affects a cyclist's motion. The term in Equation (1) which represents gravity is:

$$mg \sin \theta \tag{4}$$

If the slope is negative, then gravity will aid the motion of the rider, and if the slope is positive, it will hinder the motion of the rider. From Equation (1), $mg \sin \theta$ is the part of the equation which determines the contribution of the force due to gravity. As shown in this part of the equation, another factor that affects a rider's speed is the combined weight of the bike and rider, mg.

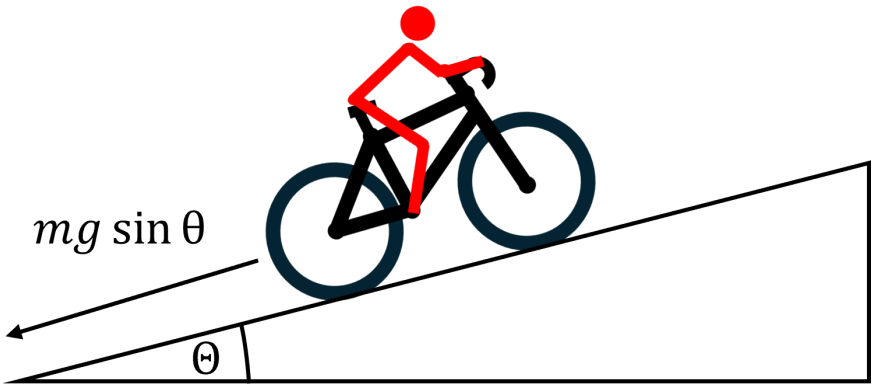


Figure 2.4: A diagram showing the gravitational component of a cyclist on a slope.

As mass is the only controllable factor in a rider's weight, there are ongoing efforts within the industry and research literature to pursue lighter-weight bikes, equipment, and riders. Until the late 1990s, most competition road bikes were made of steel; these bikes were handcrafted but often heavy. As technology developed, materials such as 6061 aluminium were used for their strength and weight, and later carbon fibre dominated the pro peloton. These advancements in materials

lead to lighter bikes and tube profiles that can be formed for better aerodynamic efficiency.

Aerodynamics

The term in Equation (1) which represents the aerodynamic drag of a bike rider system is:

$$\frac{1}{2}\rho C_d A v^2 \tag{5}$$

Equation (5) will often appear as the equation describing the drag on a rider. C_d is the coefficient of aerodynamic drag, A is the frontal area, ρ is the air density and v is the airspeed. In performance cycling, the two main variables minimised are C_d and A, and in most studies, the aim is to reduce their overall product $C_d A$.

Fluid Flow

Understanding the complexities of fluid flow is essential to understanding how air resistance affects a cyclist's performance on the road. A fluid is any substance that deforms continuously under shear stress and adapts to the shape of its container, and air falls into this category.

An important quantity for characterising fluid behaviour is the Reynolds number.

$$\mathrm{Re} = \frac{\rho L U}{\mu} \tag{6}$$

As shown in Equation (6), the Reynolds number is defined in terms of the fluid density ρ, a characteristic length scale L, the flow speed U, and the dynamic viscosity μ. First introduced by Osborne Reynolds in 1883 (Reynolds 1883), this dimensionless quantity characterises how a fluid interacts with a body and

indicates whether the flow will be laminar or turbulent.

In internal flows, such as pipes, transition commonly occurs around Re $\approx$ 2300, whereas for external bluff bodies, such as a cyclist, turbulence develops at much lower values due to early flow separation (White 2011). For bluff bodies relevant to cycling, such as the rider's legs, arms, helmet, and bicycle tubing, separated flow regions form large low-pressure wakes that dominate aerodynamic drag. A classical example is flow around a cylinder, where increasing Reynolds number can trigger a drag crisis which is a sudden reduction in pressure drag associated with delayed flow separation caused by transition to a turbulent boundary layer. Similar mechanisms are exploited in cycling equipment design, where surface textures, trip features, and component shaping can intentionally promote controlled transition to reduce wake size and overall drag (Crouch et al. 2017). Broadly, higher Reynolds numbers correspond to more turbulent behaviour. Reynolds number is widely used in the literature to enable scale modelling of cycling aerodynamics, allowing experimental data to be matched reliably to full-scale conditions. For a cyclist treated as a bluff body, sufficiently high Reynolds numbers lead to boundary-layer separation and the formation of a large turbulent wake, which, in turn, increases the aerodynamic drag experienced during motion. Eddies can also form within this separated-flow region, further contributing to unsteady aerodynamic forces.

Drag

The greatest force affecting a rider's performance on a bike has been the air they are riding through. Many advances have been made in optimising the bike the rider is on, alongside the wheels attached to it; however, the least aerodynamic component of the bike-rider system is the rider. A rider can take up to 90% of the aerodynamic drag of a bike rider system (Kyle and Burke 1984) and athletes

and teams go to great lengths to find those aerodynamically optimal gains, such as the wind tunnel as seen in Figure 2.5

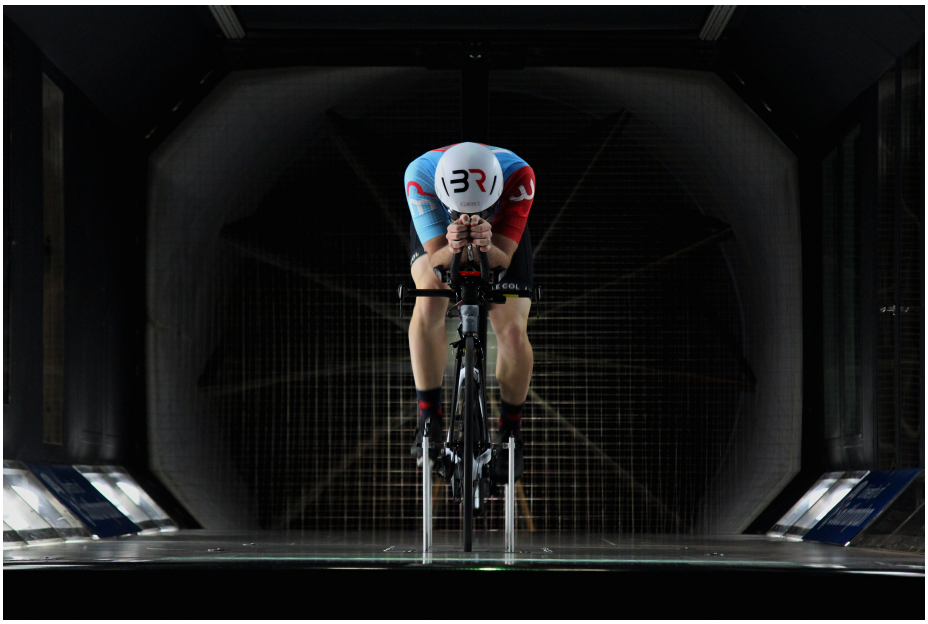


Figure 2.5: Image of a cyclist in the University of Southampton wind tunnel.

Drag is the resistive force exerted by a fluid on an object moving through it, acting opposite to the direction of motion. Lift, which can also appear as downforce when directed toward the ground, acts perpendicular to the path of travel. Many engineered systems, including aircraft, motor vehicles, and hydrofoils, rely on controlling lift or downforce, but reducing drag is almost always a key design priority. For an aircraft, reducing drag reduces fuel consumption and operating costs. For a parachute, increasing drag ensures the user reaches a safe terminal velocity. For cyclists, reducing drag improves efficiency and speed. When the surrounding medium is air, this resistive force is known as aerodynamic drag. In liquids or other fluids, the corresponding force is referred to as hydrodynamic drag (Alam, Chowdhury and Moria 2019). Drag can be broken down into two main components: viscous drag and pressure drag. Viscous drag is the drag caused by a fluid due to the sheer stress of the layers of fluid past the boundary layer of the

object. Pressure drag is the resultant force from the pressure distribution around an object in the direction of interest. For drag, the direction of interest is against the direction of travel (Sudin et al. 2014).

In general, both pressure stresses and shear stresses are the primary contributors to the overall drag force; however, pressure drag is dominant. As a result, the drag equation for a cyclist can take the form of Equation (7)

$$F_d = \frac{1}{2}\rho v^2 C_d A \tag{7}$$

Where F_d is the drag force A is the a reference area dependent on how C_d was determined. For bluff body's like a cyclist A generally is the frontal area. v is the free-stream velocity through the fluid ρ is the fluid density and C_d is the coefficient of aerodynamic drag. From the formula, it is clear that if C_d decreases, the drag force on the object decreases. Similarly, if the frontal area of the object decreases, the drag force decreases. Different objects have different C_d values due to the amount of flow separation they experience, such as bluff- and streamlined-bodies. Examples of the C_d's of different objects follow:

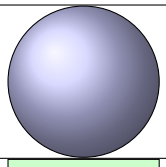
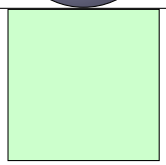
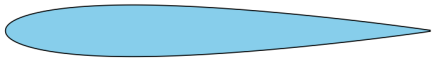
Shape	Diagram	Drag Coefficient (Cd)
Sphere		0.47 (Asai and Hong 2021)
Cube		1.05 (Holmes et al. 2004)
Airfoil (NACA 0012)		0.006 (Abbott et al. 2012)
Bike + rider		≈ 0.88 (University 2025)

Table 1: Drag Coefficients (Cd) for Common Shapes

For a cyclist, the drag coefficient C_d is often treated as approximately constant over a given posture, though in reality it varies with Reynolds number. At low Reynolds numbers, the flow around a body is predominantly laminar, with minimal separation, so viscous (skin-friction) drag is the primary contributor to total drag. As the Reynolds number increases, flow separation begins, and both pressure and viscous drag contribute. With further increases, vortex shedding develops, and the wake becomes increasingly turbulent. Once a large, turbulent wake forms behind the body, pressure drag becomes the dominant drag force for bluff bodies such as cyclists.

The Impact of Apparel and Position on Aerodynamic Drag

There are many ways to reduce a cyclist's aerodynamic drag, whether through components on the bike, the rider's position, or the clothing they wear. At race

speeds, the rider themselves is the largest contributor of aerodynamic drag, contributing up to 64-82% of the drag of the bike rider system (Malizia and Blocken 2021).

Helmets

Helmets are essential for a rider's safety while riding. A helmet which meets the relevant safety standards, such as EN 1078 (BSE 2012), will significantly decrease the chances of a severe injury to the head if an accident were to occur (Olivier and Creighton 2016). As a result, they are compulsory for most competitive cycling events, including those governed by the UCI (Union Cycliste Internationale UCI), International Olympic Committee (Committee 2024) and world triathlon (World Triathlon 2024), to name a few. This leads to an abundance of research and development into making helmets as aerodynamic as possible whilst ensuring they meet the relevant safety standards.



Figure 2.6: Three different cycling helmets, (left) the MET drone wide body (MET-Helmets 2025), (Middle) HJC Adwatt (HJC Helmets 2025) and (right) POC Omne Beacon (POC Sports 2025)

Helmets, such as those seen in Figure 2.6, come in different shapes and sizes. In Figure 2.6, there are three helmets: two with an aerodynamic focus and one with a road bike/endurance focus, prioritising ventilation over reducing aerodynamic drag. As shown in Figure 2.6, the two aerodynamically designed helmets have visors and tails. This deliberate design aids with flow attachment following similar

design principles to airfoils (Forte et al. 2020). Studies such as those performed by Blair, Sidelko and Estivalet (2009) used a wind tunnel at various yaw angles and concluded that time trial (TT) helmets provide significant drag reduction when compared to a standard road bike helmet. The drag reductions presented in the work from Blair, Sidelko and Estivalet (2009) show a drag reduction when the helmets are correctly positioned. As can be noted from Figure 2.6 helmets such as the MET drone have a long tail, therefore the question arises, if a rider were to have the helmet positioned improperly, such as looking down or sitting in a far too upright position could the tail have a detrimental affect on performance? Studies such as those performed by Chabroux, Caroline and Favier (2012) investigated the impact of visors and helmet positions on aerodynamic drag. It was concluded in this study that variations in the TT posture, alongside saddle position, and the inclusion of a helmet visor can increase the aerodynamic efficiency of the rider.

Studies such as Brownlie et al. (2010) look into different cycling helmets and their effect on aerodynamic drag. Specifically, helmets without a face shield and with sunglasses. Underwood et al. (2015) investigated the effect that helmet position and ventilation have on aerodynamic drag. In this investigation, it showed that ventilated helmets had a negligible impact on aerodynamic drag when compared to the positioning of the helmet relative to the wind.

Clothing

Clothing is clearly an essential part of cycling, whether it be for moisture-wicking abilities, the number and size of pockets for energy gels or specific trip points needed to manage flow attachment and separation purposely.



Figure 2.7: An image of a rider wearing a skinsuit with a focus on a specific design element of the skinsuit.

Figure 2.7 displays a rider in a skinsuit, with a magnified view of a particular element of the suit. The lines visible on the arm have purposefully designed ridges that control boundary-layer separation. This design helps to control the airflow around the rider and reduce drag. Studies such as (Hong and Asai 2021) have found that these skinsuit features can reduce the drag on a cylindrical body such as an arm by up to 8%. It can also be seen in Figure 2.7 that the suit is tightly fitting; this is because clothing which is tightly fitting sees a significant reduction in drag as it effectively reduces the frontal area of the cyclist when compared to baggier casual clothing. Studies such as those by Brownlie et al. (2009) used tubes and limbs wrapped in skinsuit material with surface features that managed the airflow wake. Wind tunnel measurements showed that this approach could lower C_d by more than 40% in some cases. Underwood and Jermy (2011) investigated the impact different fabrics have on the aerodynamic drag experienced by a rider using them. During this investigation, they observed that these fabrics show little change in drag when aligned with the flow direction. When the material is oriented

axially to the flow, though, the rougher fabrics exhibit a substantial drop in drag at higher speeds.

Position

The position a rider adopts can strongly influence performance. The literature debates whether a more aerodynamic posture affects lower limb kinematics, with some studies supporting this idea (Jongerius et al. 2022) and others finding no clear effect (Williams et al. 2025). This line of research is valuable because a low front end can deliver major aerodynamic gains. An example of position experimentation taken to the limit is seen with Graeme Obree, known as the Flying Scotsman. During his 1993 hour record attempt, he used the praying mantis position (left Figure 2.8), which the UCI later banned, leading him to switch to the Superman position (right Figure 2.8). Between these efforts, Chris Boardman set the hour record in a more conventional posture. Due to UCI regulations, most studies now focus on traditional setups, comparing TT positions and standard road bike positions. Some studies suggest the difference in drag between an upright position and drops can be between 15-20% and for TT positions 30-35%. However, it is important to note modern TT positions can vary in drag as well (Crouch et al. 2017).



Figure 2.8: [REDACTED]

As established by (Crouch et al. 2017) there can be substantial gains to be found when adopting specific positions, up to 35% differences between TT positions and upright positions. This thesis will focus on cycling position as a key area of interest as, not only can it contribute 64-82% of the drag of a bike rider system (Malizia and Blocken 2021) but is also the drag measured by the Body Rocket system. This thesis will investigate whether the position of a rider has an impact on their power output and C_dA, specifically the impact of quick movements/adjustments on the saddle (also known as shuffling). This thesis will also investigate whether discrete positions on the bike can be predicted using machine learning and data from the Body Rocket system. Further more cycling position on the bike will be further analysed in a continuos manner focussing on the ability to measure joint angles and joint locations through the use of computer vision techniques. Finally the last study of this thesis will look to apply Large Language Modelling (LLM) to the domain of cycling aerodynamics to produce a LLM expert which could have the potential to tie all of these studies together and provide protocols and insight to the average cyclist without the need for human experts.

Other Areas of Aerodynamic Drag Reduction

The methods of reducing aerodynamic drag which have been presented impact the aerodynamics of an individual, however, another area where drag can be reduced is in the peloton. In a peloton cyclists will generally arrange themselves in certain patterns which leads to differences in aerodynamic drag depending on where you are situated. On the edges of the peloton riders are more exposed to the wind and therefore experience more aerodynamic drag, whereas riders within the peloton will generally experience a lower aerodynamic drag. These formations are not unique to cycling and are seen frequently in nature, such as in the flight patterns of geese and in schools of fish. Research by Blocken et al. (2018) has shown that

riding within a peloton can cut a cyclist's drag by up to 50%, and even riders positioned at the edges of the group benefit from reduced drag. This study in particular used both CFD and wind tunnel modelling.

2.2 Aerodynamic Testing within Cycling

Wind Tunnel

Wind tunnels have long been the foundation for aerodynamic testing within cycling. They offer a controlled setting in which cyclists can be tested by directing airflow over the rider at precise speeds. On the floor of the tunnel there exists a force balance, the rider and bike is then fixed to this balance and the drag force they produce is measured. Wind tunnels are consistently used within other fields from motor sport, civil engineering and maritime engineering. Although wind tunnels are controlled testing environments they do have their disadvantages as they are so controlled they do not exactly replicate real world testing conditions. As a result other techniques have since been on the rise. Studies such as those carried out by García-López et al. (2008) used the wind tunnel as their measurement device for optimising the aerodynamics of a cohort of cyclists and found the reliability of the wind tunnel to be high with '($r > 0.96$, coefficient of variation $<$ 2%)'.Other studies, such as those by Underwood et al. (2011), used a wind tunnel and identified a relationship between torso angle and aerodynamic drag, showing that drag decreases as a rider's torso angle becomes lower. Wind tunnels have also been central to skinsuit development, as seen in Brownlie et al. (2009), where they were used to design and optimise a time trial suit for the Tour de France and the Olympics. Further work on fabric behaviour includes Zheng et al. (2024), which examined longitudinal grooves on a rider's arms and reported a 7 percent reduction in drag in wind tunnel testing.



Figure 2.9: [REDACTED]

Wind tunnels have long been regarded as the reference standard for aerodynamic assessment due to their high levels of repeatability and measurement precision. Studies such as those carried out by García-López et al. (2008) used the wind tunnel as their measurement device for optimising the aerodynamics of a cohort of cyclists and found the reliability of the wind tunnel to be high with "($r > 0.96$, coefficient of variation $< 2\%$)". Crouch et al. (2021) Investigated out-of-seat road bike positions and found the largest variation of 5%; however, do consider that for time trial positions, repeatability can be <0.5%. This highlights the highly repeatable nature of controlled aerodynamic testing environments; however, it also shows that they can still be susceptible to substantial rider movement. This can lead to the conclusion that methods where data is acquired over a longer period of time could provide more accurate aerodynamic data, especially if it were collected in the field.

Using equation Equation (7) the drag force is directly measured from the wind tunnel balance, the air density and air speed is measured therefore the C_dA can be determined. Rearranged the equation takes the form:

$$C_dA = \frac{2F_d}{\rho v^2} \tag{8}$$

When paired with a frontal area measurement, it is then possible to determine the C_d of the rider.

Computational Fluid Dynamics

Computational Fluid Dynamics (CFD) is widely used to analyse the behaviour of fluid flows around complex geometries. In a CFD simulation, the object of interest is placed within a computational domain representing the surrounding flow, and the governing equations are solved numerically. These governing equations are the three dimensional compressible Navier Stokes equations Navier (1821), which describe the conservation of mass, momentum, and energy.

Several modelling approaches fall under the umbrella of CFD. At the highest level of fidelity is Direct Numerical Simulation (DNS), which resolves the full Navier Stokes equations without any turbulence modelling. Although DNS offers the most accurate representation of the flow, it is extremely demanding computationally and is therefore restricted to simple geometries or low Reynolds number studies (Hassan 2010). Large Eddy Simulation (LES) reduces the computational cost by explicitly resolving only the large, energy-containing turbulent structures while modelling the smaller subgrid-scale motions. A more affordable approach is Reynolds Averaged Navier Stokes (RANS), in which the Navier Stokes equations are averaged over time, and turbulence effects are represented with closure models such as the $k-\epsilon$ or $k-\omega$ formulations. RANS methods are widely used in engineering because they provide a practical balance between accuracy and computational efficiency.

Defraeye et al. (2010) compared computational fluid dynamics (CFD) predictions obtained using Reynolds-averaged Navier–Stokes (RANS) and large-eddy simulation (LES) models against wind-tunnel measurements. Their results indicated that LES exhibited closer agreement with the experimental data than RANS, with

discrepancies of approximately 7% and 11%, respectively. Similarly, van Druenen and Blocken (2024) reported good agreement between CFD and wind-tunnel experiments using both scale-adaptive simulations and steady RANS with either the Transition SST (T-SST) or the SST k–ω turbulence models. A RANS SST $k-\omega$ simulation has also been compared to practical results by Walker et al. (2024) where it was concluded that this type of CFD simulation could adequately predict C_d and Re dependency via a desktop PC. In addition, Blocken et al. (2019) examined aerodynamic time-trial positions and validated a RANS SST k–ω model against wind-tunnel measurements. Taylor et al. (2024) evaluated the applicability of LES for cycling aerodynamics and found that LES underperformed relative to wind-tunnel results; however, the under-prediction of approximately 3% was considered acceptable for practical applications in cycling aerodynamics.

CFD has also been employed and validated to derive new insights in cycling aerodynamics. For example, Blocken et al. (2018) investigated the influence of peloton formation on aerodynamic drag. Their study used CFD to quantify the drag experienced by riders within a peloton and demonstrated satisfactory agreement with wind-tunnel measurements. Furthermore, CFD has been applied to assess the effect of leg position on cycling aerodynamics (Griffith et al. 2014).



Figure 2.10: [REDACTED]

The studies presented here show that CFD can provide robust predictions of a rider's aerodynamic drag when appropriate assumptions are made. This is supported by validation studies such as Taylor et al. (2024) and Defraeye et al. (2010) providing errors within the range of 3-11% compared to wind tunnel results. This supports the idea that CFD is indeed consistent and reliable. It is important to note however that the CFD configuration does impact the robustness of results as they can be influenced by the turbulence model selected, mesh resolution of the model and boundary condition specification. LES and DNS generally is higher fidelity and therefore more accurate, however, comes at a computational cost. In comparison RANS-based methods are generally considered to be more appropriate for cycling applications acting as an appropriate trade-off between robustness and computational cost.

Coast/run down Method

Wind tunnels are a high quality testing environment as they offer repeatable results in consistent wind conditions, however, as they are so controlled there can be

challenges simulating real world conditions. As a result infield testing techniques have seen a rise in popularity with one of them being the coastdown/run down technique. In this approach, a rider accelerates to a target speed, stops pedalling, and allows the bike to decelerate naturally. Since the deceleration profile is governed by rolling resistance, gradient, and aerodynamic drag, knowing the rolling resistance and gradient allows the aerodynamic drag to be calculated. Repeating the test several times improves reliability and reduces noise from environmental factors (Debraux et al. 2011). Studies that have investigated the coast down technique include Baldissera (2017). In this study Baldissera (2017) investigated the application of speed dependent polynomials to help determine the rolling resistance. Other studies have also investigated the deceleration method and found it to be accurate such as Candau et al. (1999). Other studies also have investigated novel approaches to the same problem such as Tengattini and Bigazzi (2018). The coast-down method usually uses the equation of motion of a cyclist as described in Equation (1). Therefore, if the C_{rr}, μ and all other factors are known it it then possible to determine C_dA.

Virtual Elevation

The power to propel a rider can be broken down into 4 main component parts; the power needed to overcome rolling resistance, the power for changing elevation the power to accelerate and the power to overcome air resistance. Theoretically, the power required to overcome rolling resistance is only a function of speed, weight, and a rolling resistance coefficient C_{rr}. The power required to overcome the change in elevation can be described with the production all of these components of power can be determined easily. The rolling resistance coefficient can be looked up on the manufacturer's specifications. The power to change elevation requires the slope of the terrain being ridden across; the power needed for acceleration can be

determined with an odometer, and the power needed to overcome these can be measured with a power meter. The power to overcome the aerodynamic drag can then be isolated and with air speed the C_dA can be determined. Other studies have looked into modelling in other forms such as Danek, Slawinski and Stanoev (2020), however, the main physical principals remain the same.

$$W = W_{rr} + W_{PE} + W_{KE} + W_{\text{Aero}} \tag{9}$$

$$W = C_{rr}\, m\, g\, v + s\, m\, g\, v + m\, a\, v + \frac{1}{2} C_d A\, \rho\, v_{\text{air}}^2\, v \tag{10}$$

Rearranging the above equation for C_dA produces:

$$C_dA = \frac{2(W - (C_{rr}\, m\, g\, v + s\, m\, g\, v + m\, a\, v))}{\rho\, v_{air}^2 v} \tag{11}$$

From this formula, if it is assumed that the air is static, then the air speed will roughly be equal to the ground speed, which leads to the ability to determine C_dA by using only a power meter. If the air speed is measured, it can be substituted into this equation to obtain C_dA.

A pitot is a device that measures the differential pressure between a dynamic pressure port that is in line with the direction of travel or direction of the wind and a static port that is situated tangentially to the direction of travel Cho et al. (2011). From Marciotto (2016) the speed of a fluid in a pitot tube can be described by the Bernoulli's principle:

$$P_a + \frac{\rho\, V_a^2}{2} + \rho g z_a = P_b + \frac{\rho\, V_b^2}{2} + \rho g z_b \tag{12}$$

Where P_a and P_b are the pressures at different points a and b, $\rho g z_b$ and $\rho g z_b$ are

the potential energies at the points a and b and $\rho V_a^2/2$, $\rho V_b^2/2$ are the kinetic energy terms at point a and b respectively and ρ is the fluid density. If a is taken to be the pitot dynamic end and b is taken to be the pitot static end then some simplifications can be made. As the pitot tube is on the same level the potential energy terms can be removed as $z_a = z_b$. Additionally in the kinetic energy term the velocity at point b (V_b) can be taken to be zero as this is the pitot static end. Following these assumptions the Bernoulli equation takes the form:

$$P_a + \frac{\rho\, V_a^2}{2} = P_b \tag{13}$$

If the difference between P_s and P_b are taken to be ΔP then when rearranged for V_a the air speed can be determined.

$$V_a = \sqrt{\frac{2\,\Delta P}{\rho}} \tag{14}$$

Once air speed has been determined, it can be substituted into Equation (11) to obtain $C_d A$.

Building on a physical modelling framework similar to that employed in the coast-down method, the energy balance/virtual elevation method allows the cyclist to continue pedalling during testing. To implement this approach, the bicycle must be equipped with power-meter pedals, and key parameters must be quantified, including the coefficient of rolling resistance, drivetrain efficiency, and road gradient or inclination changes. These additions render the model more representative of the real-world conditions typically encountered by cyclists in the wild. Commercial on-bike aerodynamic systems such as Notio Konect (Notio Cycling Inc. 2025) and AeroPod (Hamann 2018) employ Pitot tubes to measure the cyclist's airspeed. As demonstrated by van Erp, Galis and Kordi (2021), the Notio Konect device is capable of detecting changes in aerodynamic drag on the order of 1.2%. Other

investigations have implemented the virtual elevation method in a velodrome setting and compared the resulting aerodynamic coefficients with those obtained in a wind tunnel, for example Garcia-Lopez et al. (2014). Similarly, Dyer and Disley (2018) examined the validity of the virtual elevation method in para-cycling, benchmarking the results against wind tunnel measurements and concluding that the method provides sufficient accuracy. Additional research has evaluated the reliability and validity of the Notio device itself Valenzuela et al. (2020).



Figure 2.11: [REDACTED]

Overall, the virtual elevation method can be considered the state of the art for infield testing as it allows for riders and coaches to determine aerodynamic drag in the field with relatively simple equipment. However, the method is not without its limitations. As established in the background outlining the physics of cycling, rolling resistance and drivetrain efficiency are two metrics that can affect the efficiency of the bike-rider system. If correctly measured, this does provide an accurate solution for the virtual elevation method; however, practically it has

its limitations. Many of the virtual elevation models assume consistent rolling resistance and drivetrain losses; however, this is not representative of the actual conditions faced out on the road. There are methods to determine the rolling resistance of a tyre; however, for live and accurate drag force readings, both qualities of the tyre and road surface are needed. If a rider were to ride over a rough surface whilst their tyres were losing pressure, this would cause an increase in rolling resistance. If this were not accounted for in the virtual elevation calculations, this would lead to a false increase in drag readings. If there were a system that was able to directly measure drag using the same principles as the wind tunnel, a much more accurate calculation of drag force could be determined. This is known as Direct Force Drag Measurement (DFDM) and is implemented in the Body Rocket system.

Direct Force Drag Measurement

In field Direct Force Drag Measurement (DFDM) is a novel take on measuring aerodynamic drag as it takes the same principles of a wind tunnel and puts them on a bike. This is through the use of sensors on the handlebar, saddle and pedals, these sensors are similar to the force balance found at the base of a wind tunnel, allowing for a direct measurement of a cyclists aerodynamic drag. Studies such as DeGolier, Cibinel and Barnes (2024) has compared the measurements of one such DFDM system, the Body Rocket system showing it agrees within 2.3% with wind tunnel measurements. Other studies, such as Hopker et al. (2025), have validated Body Rocket system measurements with different-sized discs on rods. This study found a small coefficient of variation of 1.67% and a minimum measurable change of $0.002m^2$. This smallest measurable change of $0.002m^2$ can be considered to be the measurement uncertainty. Additionally, although not required for DFDM the uncertainty in the power meter stated by Body Rocket Ltd is $\pm 0.1\%$, (Body

Rocket 2026). Additionally other studies such as Barnes, Hopker and Gibson (2023) and Barnes, Hopker and Gibson (2024) demonstrate the applications of the Body Rocket system outside of its use as a purely aerodynamic testing device and are pieces of research which will be presented in this thesis.



Figure 2.12: Two images. Left showing the Body Rocket pedals, handle bar and saddle drag force sensors and airspeed sensor and right showing the Body Rocket sensors mounted to a bike with the drag force sensors highlighted in red circles.

The Body Rocket system is an in-field aerodynamic measurement device which mounts four force sensors (load cells) to the handlebar, saddle and pedals of a bike alongside an airspeed sensor. With these force measurements it is then possible to directly measure the drag force of a rider on a bike and when paired with air speed it is then possible to determine the C_dA.

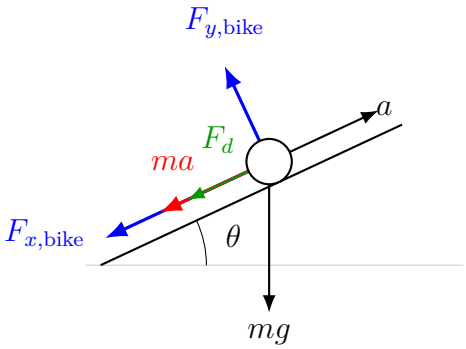


Figure 2.13: Free body diagram showing the bike frame forces $F_{x,\text{bike}}$ and $F_{y,\text{bike}}$, the weight mg, and the inertial term ma.

As the Body Rocket system utilises sensors at the contact points of a bike, isolating the rider from the bike, the rider can be modelled as shown in Figure 2.13. Figure 2.13 shows the rider modelled as a point-mass accelerating up an incline. If we take m as the mass of the rider, θ as the slope angle, a as the riders acceleration and $F_{x,\text{bike}}$ and $F_{y,\text{bike}}$ as the bike frame forces, drag force can be resolved.

Using Figure 2.13, $F_{y,\text{bike}}$ and $F_{x,\text{bike}}$ can be written as:

$$F_{x,\text{bike}} = ma + F_d + mg\sin\theta \tag{15}$$

$$F_{y,\text{bike}} = mg\cos\theta \tag{16}$$

Mass, m, can easily be determined by rearranging Equation (16):

$$m = \frac{F_{y,\text{bike}}}{g\cos\theta} \tag{17}$$

And if θ is known, F_d can therefore be determined:

$$F_d = F_{x,\text{bike}} - ma - mg\sin\theta \tag{18}$$

Substituting Equation (17):

$$F_d = F_{x,\text{bike}} - \left(\frac{F_{y,\text{bike}}}{g\cos\theta}\right) a - \left(\frac{F_{y,\text{bike}}}{g\cos\theta}\right) g\sin\theta \tag{19}$$

Rearranging:

$$F_d = F_{x,\text{bike}} - \left(\frac{F_{y,\text{bike}}}{g\cos\theta}\right) (a - g\sin\theta) \tag{20}$$

As can be seen in Equation (20) drag force can therefore be directly measured at any point in time without the need for an accurate power measurement or knowledge of any of the other resistive forces such as rolling resistance and drivetrain efficiency. For completeness, if v_{air} is known C_dA can be determined.

$$C_dA = \frac{2\left(F_{x,\text{bike}} - \left(\frac{F_{y,\text{bike}}}{g\cos\theta}\right)(a - g\sin\theta)\right)}{\rho_{air} v_{air}^2} \tag{21}$$

As established, there are several methods for determining a cyclist's aerodynamic drag; however, each has its benefits and drawbacks. The wind tunnel is an environment which can produce extremely repeatable results; however, it is not representative of real-world testing conditions and can be expensive to use repeatedly. Computational Fluid Dynamics (CFD) can be even more repeatable than wind tunnel testing; however, it can also be cost-prohibitive, as it requires time and high-performance computing equipment. The virtual elevation method and coast-down methods are more accessible than wind tunnel testing and CFD; however, they require consistent conditions or the ability to monitor metrics such as drivetrain and rolling resistance losses. As a result of the benefits and drawbacks of the previously mentioned methods, the system utilised in this thesis is Direct Force Drag Measurement provided by Body Rocket Ltd. As the Body Rocket system only measures the rider's drag force, this thesis focuses on studies

investigating cycling position with machine learning and artificial intelligence to generate new cycling performance insights and serve as tools to lower the barrier to entry at the professional level of the sport.

2.3 Machine Learning within Cycling

What is Machine Learning

Machine learning is a branch of computing, specifically artificial intelligence, in which data is used to train a model. In machine learning, it can be broadly broken down into inputs and outputs. That can be a data stream or an array, such as an image. This thesis investigates two main techniques: a traditional classification task and computer vision. Classification is a machine learning technique where an input is given to a model and a discrete output is produced. In machine learning, there are many methods for obtaining classifier outputs; these include logistic regression, Gaussian Mixture Models, Support Vector Machines, Decision Trees, and Neural Networks. A simple example of a classification problem is the probability that someone will purchase a waterproof jersey. In this classification problem, training data could be collected. This data could include the time of year, the type of riding someone does, their experience level, how active they are on social media, and who they follow. These pieces of data would be called features or predictors, and they are used to predict the output. The output in this case would be a simple binary 0 or 1, where 0 would indicate that they did not purchase a waterproof jersey and 1 would indicate that they have purchased a jersey. To train a model, all these features and outputs would be collected and used within training.

Computer Vision is a branch of machine learning that takes images and obtains information from these images. Early computer vision models were based on Neural

Network architectures, and some were used to determine letters from handwriting. Other applications include facial construction.

What is Artificial Intelligence

Artificial Intelligence, and more specifically generative artificial intelligence, has seen a rise in popularity in recent years, not only within the computer science fraternity but also in the general population, with notable examples providing generative AI products such as OpenAI, Google, Meta, and Claude, to name a few. All of these models and products can be linked back to the neural network architecture; however, there are certain differences in how they learn, how they are deployed, and how they are used.

There are a number of different types of Generative AI which currently exist (at the time of writing), and whilst this thesis will only investigate one branch of generative AI, Language Modelling, it is worth acknowledging the existence of these other techniques. As addressed in this study, Language Modelling is a type of generative Artificial Intelligence based on transformers (Vaswani 2017). These models take text input and, based on that input, predict characters to output. There have been numerous examples of these, such as AI chatbots (OpenAI 2023; Google 2024; Anthropic 2024), email summarisers (Microsoft 2025; Google 2025) and coding agents (GitHub and OpenAI 2021; OpenAI 2025). These chatbots have seen a cultural shift in the way in which people work and study, and for that reason, they were investigated in Chapter 5.

Applications of Artificial Intelligence and Machine Learning in cycling

Classification Studies

Classification is a useful tool and, as discussed in Section 2.3, can be used for many different applications in a number of fields. Examples where classification has been used include Kholkine et al. (2021), where the XGBoost (Chen and Guestrin 2016) was used to predict the top 10 finishers in professional road racing. The results of this study showed that the classification model outperformed fans and was evaluated on the spring classics race series. Other studies involving classification include those focusing on health aspects of an athlete, such as Ramos et al. (2020), where machine learning was used to extract individual fatigue descriptors from electromyographic (EMG) and heart rate variability (HRV) data. In this case the classifiers were whether an athlete was accumulating fatigue or not accumulating fatigue a non responder or responder. This study demonstrated a 87-93% classification accuracy. Other studies investigating machine learning, specifically Neural Networks, include Grau-Escolano, Bassolas and Vicens (2024), where machine learning techniques, in this case Neural Networks, were used to predict when bicycles in Barcelona's bike-sharing system would require maintenance. The models were shown to be effective and helped the city reduce the costs of maintenance and downtime of the bikes when in need of maintenance.

Computer Vision in Cycling

Computer vision within cycling has been researched on many occasions. Examples of this include studies such as those performed by Li et al. (2025). This study investigated the application of computer vision techniques to detect close pass detection. Other studies, such as Ibrahim (2024), also look into the applications

of computer vision for cycling safety. In particular, a developed computer vision framework was used to automate near-miss video analysis and detect various risk factors. Along with the safety aspect of cycling, another study performed by Desai, Etemad and Greenspan (2025) used computer vision to detect potentially dangerous motor vehicle-cyclist interactions for applications in automated cars. Alongside road safety, computer vision has also been applied in applications such as bike theft detection (ripa et al. 2024). In this study, YOLOv5 (Jocher et al. 2020) was used to detect potential theft attempts, and DeepSORT (Wojke, Bewley and Paulus 2017) was applied for object detection and tracking.

Whilst these studies outline the applications of computer vision outside of performance cycling, the main goal of this thesis is to investigate performance cycling and the applications of machine learning to help lower the barrier of entry to performance metrics within the sport.

There are many studies which have researched the applications of artificial intelligence and machine learning within cycling and sport. Notably within recent years. Examples of this include Ahmadi et al. (2024) where neural networks were applied to predict radial and Mediolateral pedal forces using cadence, crank angle, power and rider attributes. Mediolateral forces are defined as the forces acting on the body in the horizontal plane. Additional studies have investigated the application of recurrent neural networks.

In other studies (Kakavand et al. 2025), marker-less motion capture has been investigated through the use of Opencap (Uhlrich et al. 2023a). Kakavand et al. (2025) found that Opencap had potential, however, does need further refinement.

2.4 Machine Learning and Artificial Intelligence

Machine Learning

Machine learning is a subset of computer science in which a machine is given data and relationships are drawn between different pieces of data or learned. Machine Learning has many applications in real-world scenarios, from recommendation algorithms to self-checkout machines and AI chatbots. For some background on the topic and to better understand the underlying studies compiled within this thesis, there are a few concepts to be explained.

Classification

Classification is the process of taking data and, with that data, predicting an outcome in a discrete manner. To build this out further, here is a simple example of how the mechanism for this model could work: logistic regression. A logistic regressor is one of the simplest machine learning models, where a sigmoidal function is used, and when the predictors are applied to the model, the outcome is simply where the output lies on the logistic regression line.



Figure 2.14: Plot showing an example of a logistic regression plot. The red data points represent the class 0 and the blue represent the class 1. The green line shows the decision boundary.

In Figure 2.14 is a very simple example of how a logistic regression algorithm works. Simply, it takes the feature inputs, and if the data points are beyond the decision boundary, one of the two predictions is made.

When training these models, evaluating their accuracy is important for when they get deployed and applied in the wild. One way to determine the accuracy of a classifier is via the Receiver Operator Characteristic (ROC) and the Area Under the Curve (AUC). These metrics are an easy way to look at the accuracy of these models and work by evaluating the number of false positives, false negatives, true negatives and true positives, and this is done via a confusion matrix.

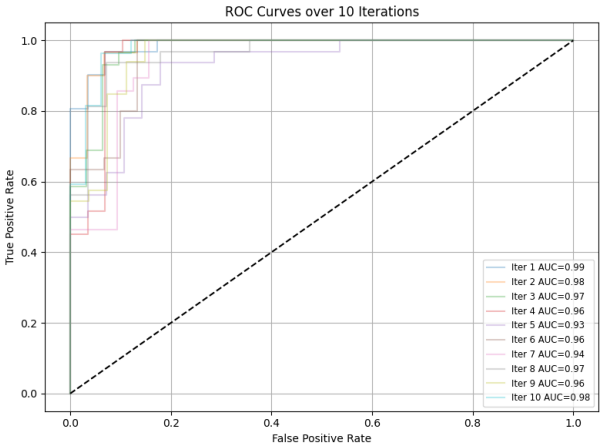


Figure 2.15: An example of a Receiver Operator Characteristic curve with a number of examples.

As shown in Figure 2.15, the true positive rate is plotted against the false positive rate. If the area under the curve is equal to one, then it is a perfect classifier; if it is equal to a half, then it is equivalent to a random guess.

The application of machine learning classifiers will be investigated within Chapter 4 where different types of machine learning classifiers; the Gaussian Mixture Model, Support Vector Machine and Gradient Boosted Machine will be trained on Body Rocket system to predict one of four positions a rider assumes on the bike.

Convolutional Neural Network

A Convolutional Neural Network is one of the cornerstones of modern machine learning and artificial intelligence. These models are inspired by human biological neurons, and in 1943, neuroscientists Warren McCulloch and Walter Pitts created the first computational neuron based on the human one (McCulloch and Pitts 1943). This led to it becoming a fundamental component of neural networks as we know them today, the perceptron.

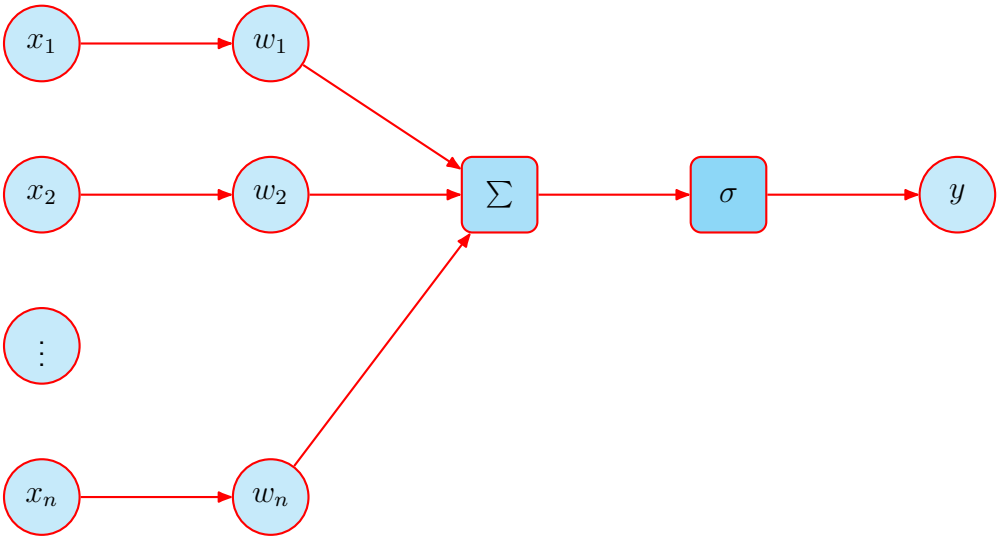


Figure 2.16: Diagram of a computational perceptron showing from left to right: the input, weights, summation, activation function and output.

Mathematically it can be described as:

$$y = \sigma\left(w_0 + \sum_{i=1}^{n} w_i x_i\right) = \begin{cases} 1 & \text{if } w_0 + \sum_{i=1^n} w_i x_i \geq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{22}$$

As can be seen in Figure 2.16 and Equation (22) the perceptron from the input values of x_0 to x_n and multiply them by their respective weights w_0 though to w_n. These are then summated, passed through an activation function σ and the output is provided.

These single perceptrons are effective with linear data, however, come into limitations when the data is not linear (Du et al. 2022). As a result the multi layer perceptron has come into popular use to tackle the limitations of the single layer perceptron on non-linear data.

The Multilayer Perceptron (MLP) is constructed of multiple layers and non-linear activation functions. These multilayer perceptrons are composed of three layers:

the input layer, the hidden layer, and the output layer. The Input layer represents the values of the predictors/features, the hidden layers are applied over various nodes, and the output layer is the prediction. As an input is processed through the different nodes to produce an output, this is known as forward propagation.

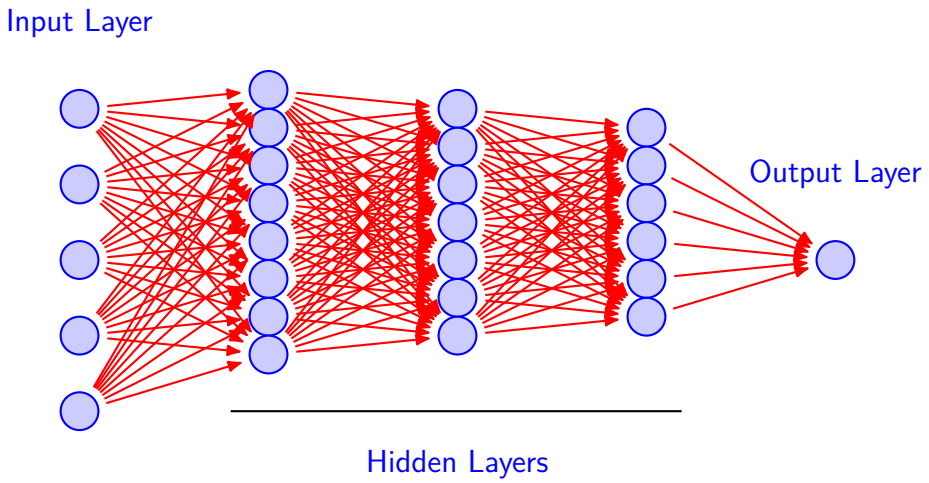


Figure 2.17: **Multilayer Perceptron (MLP)** architecture illustrating input, three hidden layers, and output layers with connections representing forward propagation.

Figure 2.17 shows a Multilayer Perceptron, which is also referred to as a fully connected feed-forward network (Karrach and Pivarčiová 2023). In the same manner as Figure 2.16 the predictors x_0 to x_n are multiplied by weights w_0 to w_n and an activation function σ. This value is determined for each node in the hidden layer and can be expressed as:

$$z_k^{(1)} = \sigma_1(w_{0,k}^{(1)} + w_{1,k}^{(1)}x_1 + +w_{2,k}^{(1)}x_2) \tag{23}$$

From this Equation (23) can take vector notations of x_0 to x_n and the weights w_0 to w_n. As a result z_k can be expressed as:

$$z_k^{(1)} = \sigma_1(\vec{w}_k^{(1)} \cdot \vec{x}) \tag{24}$$

As this procedure is repeated over all layers, the matrix notation for the entire multilayer perceptron can be expressed as:

$$z^{(1)} = \sigma\left(\left(W^{(1)}\right)^T \vec{x} + \vec{b}^{(1)}\right) \tag{25}$$

In this case $W^{(1)} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}$, $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$,$\vec{b}^{(1)} = \begin{bmatrix} w_{0,1} \\ w_{0,2} \\ w_{0,3} \end{bmatrix}$

Network Training

Ensuring that a Network is appropriately trained is pivotal to achieving as accurate predictions as possible. In the simplest form, we can express our CNN as:

$$\hat{y} = \hat{y}(\vec{x}, \vec{w}) \tag{26}$$

Where in Equation (26) $\vec{x}$ is the input vector, $\vec{w}$ are the weights, and $\hat{y}$ is the prediction. Prior to training, the vector weights $\vec{w}$ are random; however, it is during training that these weights are determined.

If we take a dataset D it can be expressed as the following:

$$D = \{(\vec{x_1}, \vec{y_1}), (\vec{x_2}, \vec{y_2}), ..., (\vec{x_n}, \vec{y_n})\} \tag{27}$$

Where $\vec{x}$ and $\vec{y} \in \mathbb{R}$. For regression tasks such as finding the locations of landmarks to generate human pose estimation the mean squared error can be used to determine the error and can be expressed as:

$$\text{MSE} = \frac{1}{n}\sum_{k=1}^{n}||\vec{y}_k - \hat{y}(\vec{x}_k, \vec{w})||^2 \tag{28}$$

In this case $||\cdot||$ represents the Euclidian norm which takes the form:

$$||y||^2 = \sum_{k=1}^{d}(y(k))^2 \tag{29}$$

With this method of determining the error, it is now possible to take a loss function $E(\vec{w})$ and, by finding the minimum of the loss function via its gradient, determine the most accurate values for the weights. This can be expressed in the form:

$$\nabla E = \left(..., \frac{\partial E}{\partial w_{ij}}, ...\right) \tag{30}$$

With the ability to find the gradient of the loss function, the gradient descent procedure takes the form:

$$\vec{w}_{n+1} = \vec{w}_n - \mu \nabla E(\vec{w}_n) \tag{31}$$

During the training procedure, it can be stated that one epoch has been run if all the network weights have been updated n times (Where n is the number of entries in the dataset).

Large Language Models (LLMs) are a class of deep learning models, primarily based on the Transformer architecture Vaswani (2017), that have demonstrated emergent capabilities in natural language understanding and generation when scaled to billions of parameters. These models are trained using self-supervised objectives over massive text corpora, with performance following predictable scaling laws until phase shifts introduce qualitative changes in behaviour Kaplan et al. (2020); Wei et al. (2022). Core theoretical insights into LLMs include their ability

to perform in-context learning Brown et al. (2020), generalise across tasks with minimal supervision, and develop latent representations aligned with linguistic and semantic structures.

Through the combination of machine learning and artificial intelligence techniques with data collected using the Body Rocket system, a number of novel research areas can be explored within performance cycling. The following chapters investigate several of these areas, including the impact of shuffling on cycling aerodynamics and power output, the application of multi-class classification models for predicting rider position using Body Rocket data, the evaluation and fine-tuning of computer vision pose estimation models for cycling analysis, and the development of a domain-specific large language model for cycling aerodynamics coaching.

Chapter 3

Shuffling

As established, the Body Rocket system can directly measure the drag force on a rider using sensors at the bike's contact points. Given the precision required to measure drag force and the added ability to measure moments at the handlebar and saddle, this chapter focuses on a unique capability of the system and investigates how one moves on the saddle (shuffles). Using power data and C_dA data, this study asks whether shuffling hinders performance and whether bike fitters and coaches should focus more time on optimising or minimising it. Alongside this, a validation method to determine a rider's position on the saddle is implemented and compared with the Body Rocket system's output.

3.1 Introduction to Shuffling

The issue of a rider shuffling on the saddle is a subject of ongoing debate within the cycling community. However, there is a lack of empirical evidence to either support or contradict the prevailing assumption that such shuffling may adversely affect a rider's aerodynamic drag or power output. Given that the Body Rocket system measures both vertical and horizontal forces, as well as the moments on the

saddle, the question thus arises: can the Body Rocket system accurately identify a rider's position on the saddle? Accordingly, this study has two primary objectives:

- To quantify the accuracy of the Body Rocket system's determination of saddle position via an additional saddle position monitoring mechanism.
- To subsequently assess whether shuffling is beneficial, detrimental, or dependent on the individual kinematics of the rider.

Research investigations related to shuffling predominantly centre around optimising rider positioning to enhance comfort (Brooke and Broadbent 2020). From a biomechanical standpoint, forward movement on the saddle results in a decrease in effective saddle height, consequently affecting lower limb kinematics and diminishing cycling efficiency (Ferrer-Roca et al. 2014), as well as reducing the resultant pedalling force and force effectiveness (Bini, Hume and Crofta 2011). It is thus postulated that cycling power output may be diminished in scenarios where riders exhibit high rates of shuffling. In addition, there is a paucity of research investigating the impact of shuffling on a rider's coefficient of aerodynamic drag area (C_dA), and therefore, its impact is largely unknown.

In previous studies, the effects of trunk acceleration on saddle position and drag coefficient have been investigated (Evans et al. 2021a). During their study, a relationship was found between higher magnitude trunk accelerations and an increase in C_d. Other studies indicate that trunk position has an impact on muscle recruitment, specifically McDonald, Holliday and Swart (2022) researched the impact of the effective seat-tube angle on saddle pressure indexes. In this research, it was demonstrated that the effective seat tube angle causes recruitment of different muscles in different positions; however, no conclusive results were found as to what angle was the most optimal for power output. Jongerius et al. (2022) researched the biomechanics of the force application in different cycling positions. In their study, it was shown that a more forward hip position provided better

mechanical effectiveness. From the results obtained in these studies, it can be expected that in this investigation, an increase in C_dA is to be observed with each shuffle, consistent with the conclusions of Evans et al. (2021a). With respect to the rider's power output, it can be expected that there will be more variability in how power output is impacted by shuffling, whether that be an increase or a decrease in power.

Despite the paucity of peer-reviewed research on shuffling, the topic has attracted notable attention from both industry practitioners and enthusiasts. Given the gap in the literature regarding in-field testing of shuffling, this research aims to provide insight into the impact of shuffling on C_dA and power, alongside validating the method the Body Rocket system implements to determine the position of a rider on the saddle.

3.2 System Design and Image Processing Pipeline

The Body Rocket system measures horizontal forces, vertical forces and moments. Therefore, the position of the rider on the saddle can be determined. As can be seen in Figure 3.1.

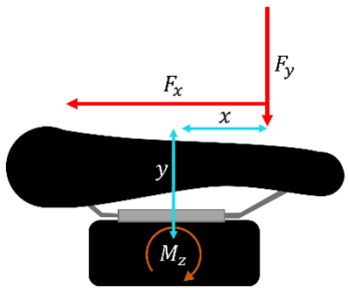


Figure 3.1: Diagram of the Body Rocket load cell showing the respective forces and moments it is subject to when a rider is on the saddle.

The saddle moments and forces can be expressed in Equation (32) to determine the rider's position on the saddle:

$$x = \frac{M_z - yF_x}{F_y} \tag{32}$$

Equation (32) with the Body Rocket load cell is a way to determine the position of a rider on the saddle; however, it does not account for pelvic rotation, and it is unknown whether this will be a limiting factor when a rider is in a time trial position. As a result, a ground truth is needed for the position of the rider on the saddle. If an appropriate device were able to determine the position of a rider on the saddle, or, for the purposes of this study, identify when a rider shuffles, it would be possible to compare it to the Body Rocket output and get an indication of its accuracy.

Methods of Monitoring Saddle Position

Numerous methodologies exist for monitoring a rider's position on the saddle, although not with the intention of measuring shuffling. Certain technologies employ pressure mapping covers to assess the rider's position alongside various other metrics (gebiomized 2025; novel.de 2025; fitkitsystems 2025). Nevertheless, these systems are typically mounted on top of the saddle and are predominantly utilised in static riding scenarios. Conversely, there are static systems that do not necessitate any interface between the rider and the saddle, such as motion capture systems like Qualisys (2025) and Retül (2025). These systems can employ a variety of techniques to track the position of a rider's joints by utilising retro-reflective markers and inertial measurement units.

Given that these systems employ an interface situated between the rider and the saddle or are intended solely for static use cases, it was concluded that the optimal course of action would be to engineer a novel device adhering to four principal criteria: it must be lightweight, minimally disruptive to the rider's airflow,

cost-effective, and incorporate redundancies or multiple methods to detect rider movements. In alignment with these criteria, the Raspberry Pi was selected as the foundational component of this device, initially utilising the **Raspberry Pi 3 Model B** (Foundation 2024a), and subsequently the **Raspberry Pi Zero 2 W** (Foundation 2024b).

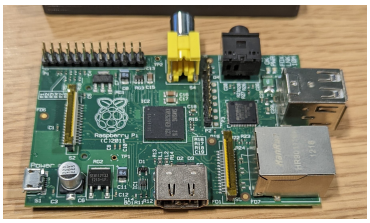


Figure 3.2: Image of an early Raspberry Pi Model B

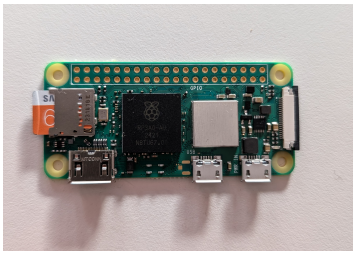


Figure 3.3: Image of a Raspberry Pi zero 2 W

Both of these devices are single-board computers used for a variety of tasks by educators and engineers (Jolles 2021). With a Linux-based operating system and GPIO pins, this allows for adaptability with a multitude of different peripherals. After choosing the base for this computing ability, the next step was deciding the peripherals to use. By this stage, two solutions were considered, one in the form of an ultrasonic proximity sensor (Parallax 2025) and the other using a camera connected to the Pi. An ultrasonic proximity sensor works through the use of an emitter and a receiver, the emitter emits ultrasonic sound, the sound then bounces off an object and from the time taken for the sound to return the distance can be determined. The best place to position this device would be on the rear of the bike; however, the drawbacks would be that the speed of sound would have to be accurately known, and it lacked any form of redundancy. As a result, it was decided that an optical solution was to be implemented. This would involve the use of a camera connected to the Pi monitoring the rear of the rider on the saddle. The camera footage would act as a visual cue for when the rider shuffles on the saddle; however, the main goal would be to implement some computer vision

to quantify the position of the rider on the saddle more in line with the results obtained from the Body Rocket system. To determine the position of the rider on the saddle, inspiration was taken from the motion capture systems, which use retro reflective markers and infrared cameras to determine the locations of the joints on a subject (Qualisys 2025). With the camera mounted at the rear of the bike and the markers on the posterior of the rider, to follow the same design principles as a traditional motion capture system, the output from the camera would be white markers on a black background. Design considerations and the image processing pipeline follow.

The Design of the Infrared Camera

The infrared camera was based on a **Raspberry Pi Zero 2w**, **Raspberry Pi infrared camera**, **Pi Juice battery** , **LISIPAROI infrared flash ring** and Perspex 962 to act as a visible light filter. This can be seen in Figure 3.4. The infrared camera operated using retro-reflective markers on the posterior of the rider. Along with the infrared flash ring, this produced a high contrast image comprising of the white markers and a black background. This acts as a easy foundation to apply simple computer vision techniques and as the size of the markers are proportional to the distance from the camera a ground truth can be determined.

Given the high cost associated with professional-grade visible light filters, Poly Methyl Acrylate (PMMA, also known as Perspex 962) was employed to mitigate expenses associated with the infrared camera. This material was selected due to its capacity to absorb visible light while permitting infrared transmittance.

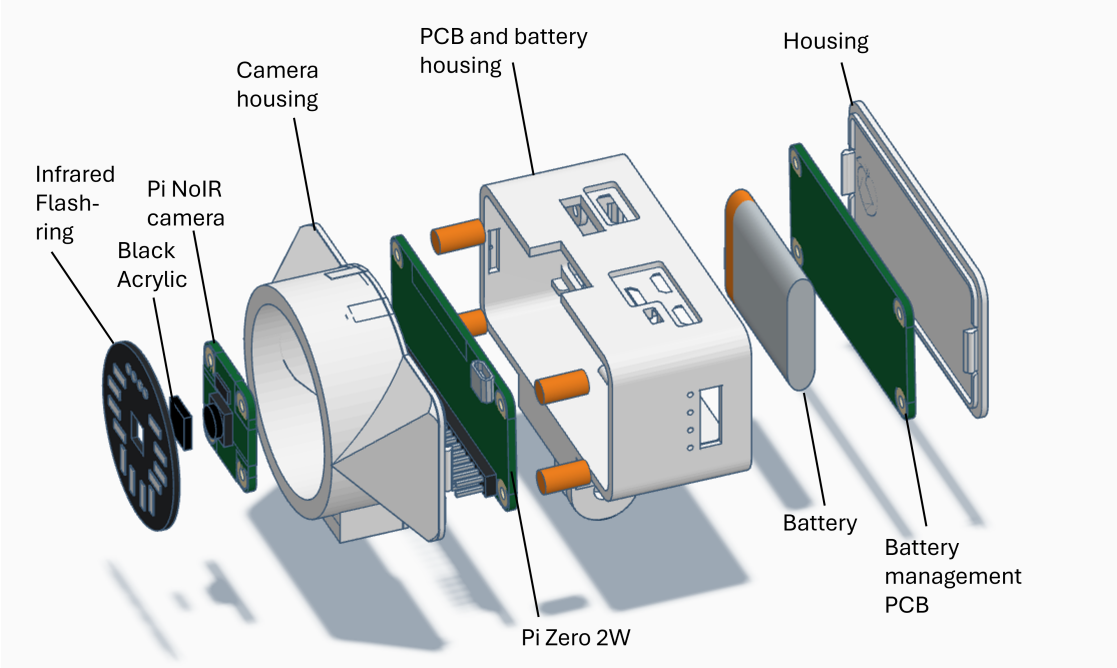


Figure 3.4: Exploded diagram of the final iteration of the infrared camera. This design and all previous designs were developed by the principal investigator.



Figure 3.5: [REDACTED]

As can be seen in Figure 3.5 PMMA does not transmit large amounts of visible light, however, does transmit near IR at a wavelength of 800nm. It has also seen applications in other areas where visible light is not desired yet infrared is (Madhav et al. 2022). When assembled, the camera has a piece of 3mm PMMA

adhered to the IR LED flash ring and pressed against the Pi camera lens.

Post-processing and Marker Detection

Once the data is collected, they take the form of a video with three white markers in the frame moving around the frame and changing in size. With this video data, the goal is to identify where the markers are, determine the size of the markers and therefore the distance from the camera.



Figure 3.6: Image output from the infrared camera showing three retro-reflective markers in a line and adhered to the rear of a rider.

To determine the size and location of the markers, two main techniques were investigated: Hough Circles and template matching. These techniques were specifically investigated due to their explainability. Additionally, the computer vision techniques used and investigated were not state-of-the-art due to the purposeful design of the infrared camera. As the infrared camera produced frames with black backgrounds and white circles, simpler and more explainable techniques were investigated.

Hough Circles

Hough circles work by performing an edge detection, such as a Canny edge detection. Once this is achieved, outlines of circles are produced. With an outline, the equation of a circle can be implemented:

$$r^2 = (x - a)^2 + (y - b)^2 \tag{33}$$

Where r is the radius, a is the distance between the origin and the y-axis b is the distance between the origin and the x-axis and x and y are the cartesian coordinates of the circle. With this a circle can be drawn from every point of the detected edge and the centre of the circle can be found where there is a peak.

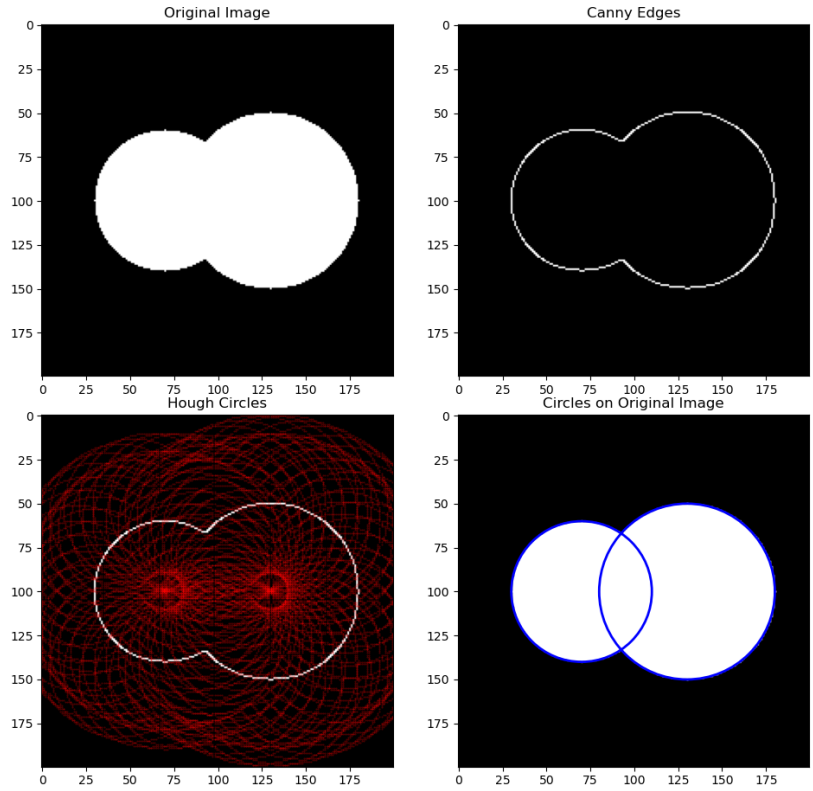


Figure 3.7: Plots showing how from an original image a Canny edge can be applied. Circles of various sizes can be drawn on the edge and where the peaks are identified the centre of the circles are found.

In Figure 3.7, two initial circles are shown. These two circles overlap and are of different sizes. When the Canny edge detection is applied, unsurprisingly, the edges of the circles are found. Once the edges are found, using Equation (33), circles of various sizes are drawn from the edge. When the size of the drawn circle matches the size of the original circle, there will be a clear peak at the origin of the original circle; therefore, the size and location of the circle are identified (Atherton and Kerbyson 1993).

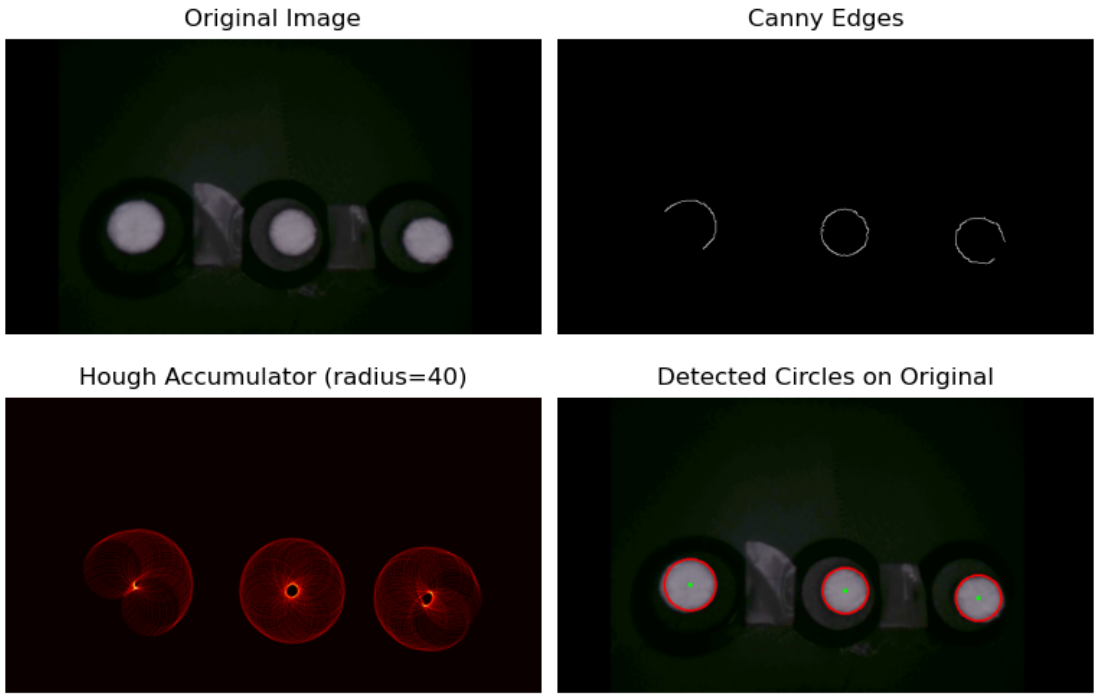


Figure 3.8: Output image from camera with the Hough circle method applied alongside the steps to achieve the output image.

Figure 3.8 shows the Hough circle output on an example image from a rider wearing the markers; from visual inspection, this method appears to perform well.

Template Matching

This study implemented a template matching method, taking inspiration from studies such as Yoo and Han (2009), however, the algorithms and methods were bespoke for this study.

Template matching is a technique where a template is created which is a close match to the target object in an image. In template matching a template (also referred as kernel) is convolved across an image sequentially and by convention from left to right across the image. Once the convolution takes place, if the objects are similar, then the score of the output image will have a peak in the centre of the target. With this peak, the template can then be overlaid, and the kernel can then incrementally increase in size. As the kernel increases in size, the sum of the pixels can be calculated, and when the summation reaches a maximum, then the template can be considered a match. The technique of template matching can be broken down into two main steps: identifying the centroid of the object, and then identifying the size.

With the template matching technique, initially a trapezoid kernel from **Astropy** (The Astropy Developers 2025) was used. The idea behind this being that the lit-up markers would look like the kernel with a gradient of brightness around the marker, and, although true a template matching the marker more exactly was found to be more effective.

The resultant footage from the camera comprised three white circles and a black background. As for each frame, the colour of the markers is unneeded information; the images were converted from the Red Green and Blue format (RGB) to grey, and once converted to greyscale, they were normalised between 0 and 1. The kernel template can then be convolved across the image, where a plot with three distinct peaks is produced.

Mathematically, if we take an image X:

$$X = \{X_R, X_G, X_B\} \tag{34}$$

where $X_R(x, y), X_G(x, y), X_B(x, y)$ are the red, green, and blue channel intensities at pixel (x, y). X is converted to greyscale.

$$X_g(x, y) = 0.299\, X_R(x, y) + 0.587\, X_G(x, y) + 0.114\, X_B(x, y) \tag{35}$$

Then normalized, where $X_{\min}$ and $X_{\max}$ are defined as:

$$X_{\min} = \min_{x,y} X_g(x, y), \quad X_{\max} = \max_{x,y} X_g(x, y) \tag{36}$$

The normalisation then follows as:

$$X_n(x, y) = \frac{X_g(x, y) - X_{\min}}{X_{\max} - X_{\min}} \tag{37}$$

With the image normalised, a kernel K, similar in size and look to the marker, is convolved across the image to produce $Y(i, j)$:

$$Y(i, j) = \sum_{m=-a}^{a} \sum_{n=-b}^{b} K(m, n)\, X(i - m, j - n) \tag{38}$$

$$Y = X * K \tag{39}$$

Initialize

$$X^{(1)} = X.$$

Algorithm 1 Find three maxima with masking

1: Initialize $X^{(1)} \leftarrow X$, $\mathcal{I} \leftarrow \varnothing$
2: **for** $k = 1$ to 3 **do**
3: $(p_k, q_k) \leftarrow \arg\max_{(i,j)} X^{(k)}(i, j)$
4: $\mathcal{I} \leftarrow \mathcal{I} \cup \{(p_k, q_k)\}$
5: Update $X^{(k+1)}(i, j)$ as:

$$X^{(k+1)}(i,j) = \begin{cases} 0, & \text{if } |i - p_k| < r_k \text{ and } |j - q_k| < r_k, \\ X^{(k)}(i,j), & \text{otherwise.} \end{cases}$$

6: **end for**
7: **return** $\mathcal{I}$ =0

After three iterations, the output is

$$\mathcal{I} = \{(p_1, q_1), (p_2, q_2), (p_3, q_3)\}. \tag{40}$$

With the centres of the peaks detected the kernel template can be placed at the centre point for each marker and a sizing procedure can begin.

For each location i the radius is found by computing the radius which corresponds to the largest score $S(i, r)$. A visual example of the image processing steps can be seen in Figure 3.9.

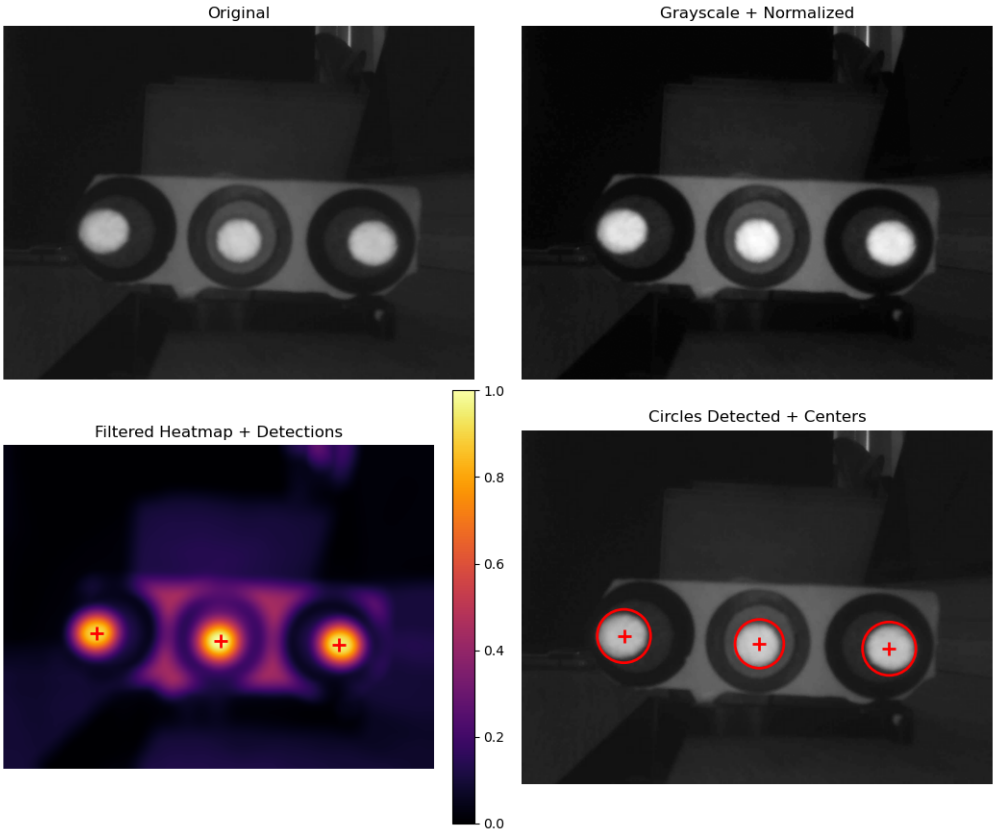


Figure 3.9: Output image from camera with the template matching method applied.

From Figure 3.9, an outline of the steps can be seen. These include the raw image, a grey scale and normalization application, the template convolution and the circle sizing and scoring.

Determining Distance from Marker Diameter

With the size of the markers determined finding the distance of them to the camera is relatively straight forward.

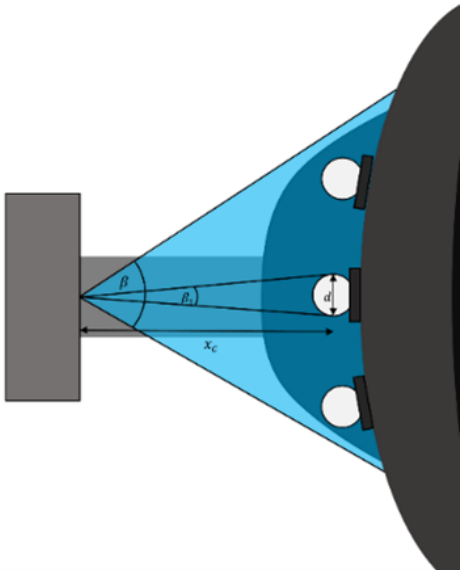


Figure 3.10: Diagram of the infrared camera with the blue shape representing the field of view of the camera. And the variables β_1, β, x_c and d showing the angle subtended by the retro reflective marker, the field of view of the infrared camera, the distance between the marker and the camera and the diameter of the ball, respectively.

The rider's position relative to the camera was determined by the camera having a known field of view β and horizontal pixel resolution of N. Using these variables, a conversion factor can be derived to convert the pixel diameter of the marker d to the angle subtended by the marker β_1. Using basic trigonometry, the distance from the marker can therefore be expressed as:

$$x_c = \frac{d}{2} tan(\beta_1) \tag{41}$$

Implementing the conversion factor for β_1 the final formula to determine x_c takes the form:

$$x_c = \frac{d}{2} tan\left(p_d \frac{N}{\beta}\right) \tag{42}$$

After determining x_c to get x_c as in Figure 3.10 the difference between the camera to the saddle mount is calculated.

Technique Comparison

To evaluate the performance of Hough circles compared to template matching, the setup in Figure 3.11 was constructed. In Figure 3.11, the infrared camera was placed facing a ruler and retro-reflective marker.

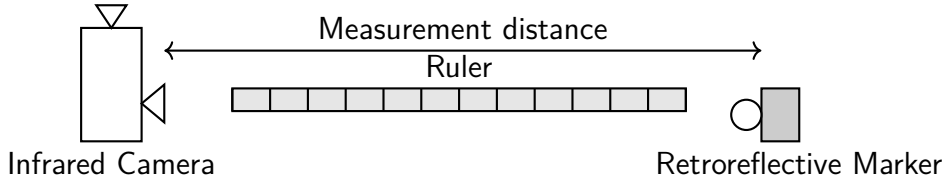


Figure 3.11: Diagram of the calibration/validation set-up depicting the infrared camera, a ruler and the retro-reflective marker placed at different intervals away from the camera.

The marker was sequentially positioned at 1 cm increments over the range 9 cm to 18 cm, covering the device's functional measurement range. For each position, measurements were recorded continuously for multiple seconds to capture repeatability. The procedure was repeated three times to assess both intra-trial variability and measurement stability over repeated trials.

Validation Results

The results from the Hough Circle and template matching method follow:

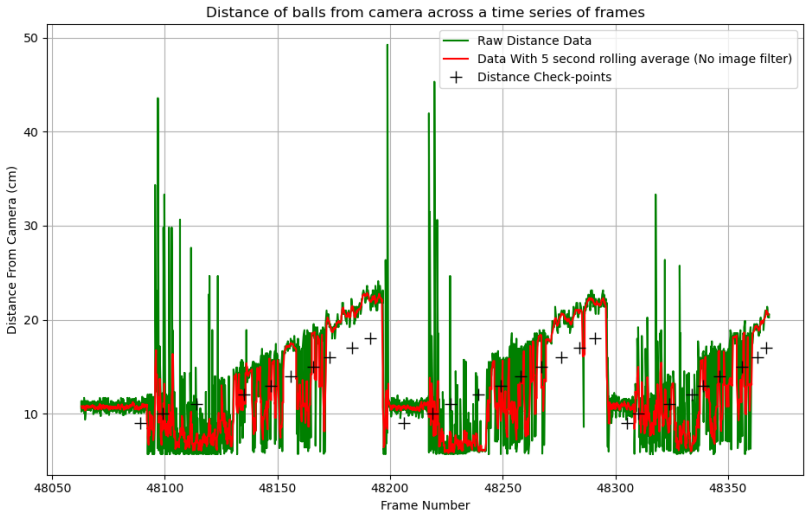


Figure 3.12: Plot showing the distance markers were from the camera, using the Hough circle method. The green trace shows the raw reading from the hough circle estimation, the red trace is with a 5 second rolling average applied and the '+' signs are the distance checkpoints.

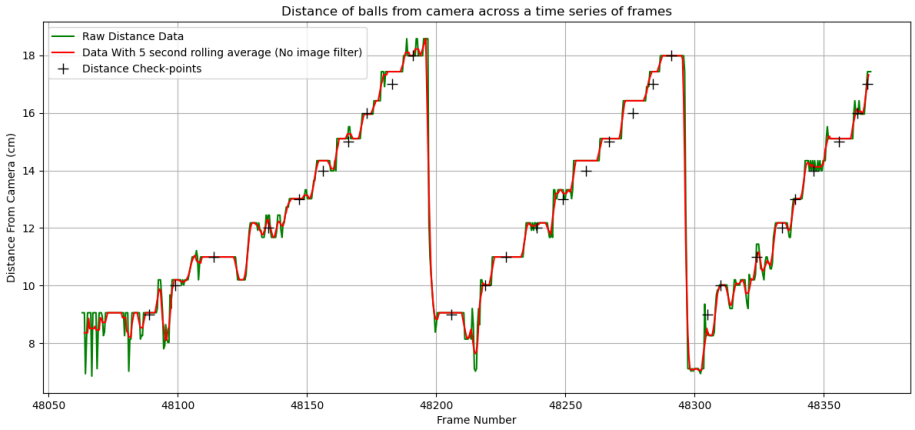


Figure 3.13: Plot showing the distance markers were from the camera, using the template matching method. The green trace shows the raw reading from the template matching estimation, the red trace is with a 5 second rolling average applied and the '+' signs are the distance checkpoints.

From the results obtained in Figure 3.13 and Figure 3.12 it is clear that, whilst

Error metric	Template Matching	Hough Circles
RMSE (cm)	0.30	4.10
t (bias)	8.023	4.340
Slope	1.0293	1.3984
Intercept (cm)	-0.25	-4.40
R^2	0.9885	0.4617
Shapiro W	0.8966	0.8363

Table 2: Table of errors comparing Hough Circles and Template Matching

not perfect, the template matching method was more accurate then Hough circles.

Performance of Hough Circles

The Hough circles exhibited moderate accuracy, with an overall RMSE of 4.10 cm and limited linear agreement with reference measurements ($R^2 = 0.4617$). A statistically significant positive bias was detected ($t = 4.340$, $p < 0.0001$), indicating a consistent overestimation of distances. The regression slope (1.3984) revealed a substantial proportional error, with measured changes approximately 39.8% greater than true changes, while the intercept (-4.40 cm) reflected a notable constant offset. The combination of low R^2 and large proportional bias suggests reduced precision and scaling accuracy across the measurement range. Shapiro–Wilk analysis confirmed that the error distribution deviated significantly from normality ($W = 0.8363$, $p < 0.0001$), implying skewness or the presence of outliers. These results indicate that both multiplicative (scaling) and additive (offset) calibration adjustments are required to improve accuracy, alongside potential filtering or noise reduction to enhance precision.

Performance of Template Matching

The template matching demonstrated high accuracy, with an overall RMSE of 0.30 cm and strong linear agreement with reference distances ($R^2 = 0.9885$). However, a significant positive bias was detected ($t = 8.023$, $p < 0.0001$), corresponding to a slight overestimation in measured distances. The regression slope (1.0293) indicates a small proportional error, while the intercept (-0.25 cm) reflects a negligible constant offset. Error distribution analysis via the Shapiro–Wilk test indicated non-normality ($W = 0.8966$, $p < 0.0001$), suggesting minor deviations from Gaussian assumptions. A simple multiplicative calibration factor could eliminate the proportional bias without degrading precision.

3.3 Body Rocket System Validation

With the template matching method evaluated to be of a suitable capability, the testing was conducted with athletes on Body Rocket-equipped bikes.

Body Rocket Validation Method

Subsequent to obtaining institutional ethical clearance, we enlisted ten experienced track cyclists to participate in a four-kilometre cycling time trial conducted on an indoor velodrome, Newport, Wales. Initially, equipped with the infrared camera, five participants were directed to complete a 4km pursuit effort; however, they were not informed that the study's primary focus was on saddle repositioning.

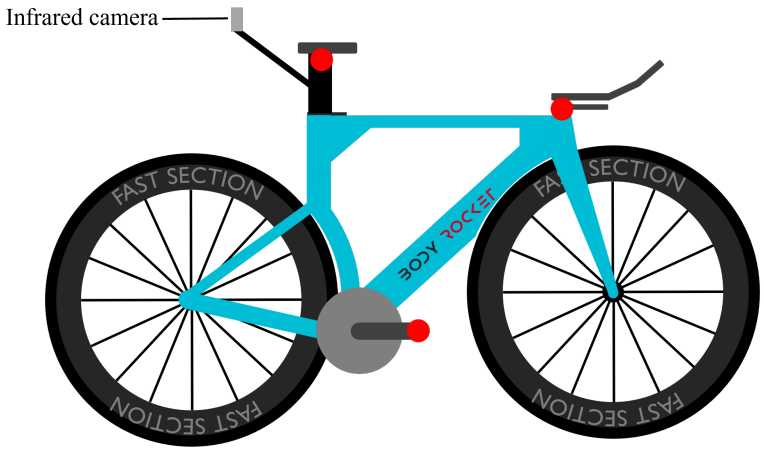


Figure 3.14: Diagram of a bike equipped with the infrared camera and the red circles representing the locations of the Body Rocket sensors.

After the initial five, after system validation, another five were enlisted to also complete a 4km pursuit effort, without the infrared camera. The bicycles were adjusted to the preferred specifications of each rider to ensure optimal comfort. Riders were advised to verify the setup from a subjective standpoint. During the time trial, data on saddle position, rider's coefficient of aerodynamic drag (C_dA), and cycling power output (measured via pedal force) were captured concurrently utilising the Body Rocket system. Prior to the trials, participants engaged in a warm-up session, executing consistent laps at a sustainable power output.

To reduce the error in measuring the separation in the markers, the three 15mm diameter markers were placed on a backing piece of double-sided tape and separated by 25mm. The infrared camera recorded images at a frame rate of 5Hz. To define a shuffle, it was determined to be a sustained downward movement in saddle position that meets three criteria. First, the rider must slide continuously for at least 1.5 seconds, with no upward movements during that interval. Second, the total drop in saddle position during this slide must exceed 6mm. Third, once the main slide ends, the rider must continue drifting downward for roughly another 2 seconds. A shuffle is therefore a prolonged, uninterrupted downward shift of more

than 6 mm, followed by a short period in which the rider continues to move lower on the saddle.

The thresholds used to identify a shuffle were selected using a heuristic approach based on the observed characteristics of the dataset. A displacement of 6mm was chosen as it exceeded the measurement RMSE of 3mm and far exceeded the changes observed during steady pedalling, therefore reducing the chance of false positive detections. Following a similar ethos the minimum duration of 1.5 s was selected as was sufficiently long enough to exceed short-term fluctuations in the signal while still short enough to detect shuffles effectively as they occur.

Body Rocket Validation Results



Figure 3.15: Images showing the two versions of the infrared camera.

Dataset	R^2	Bias	RMSE (mm)	MAPE (%)	SD(diff)	Median AE
Rider 1	0.8291	-2.510	5.103	9.546	4.445	3.277
Rider 2	0.0663	3.893	11.001	31.946	10.293	6.872
Rider 3	0.8746	0.409	1.891	3.672	1.847	1.268
Rider 4	0.5061	-0.197	3.207	2.634	3.202	2.014
Rider 5	0.4885	-4.637	7.876	7.514	6.369	4.827

Table 3: Table of errors containing the results for each riders Body Rocket system data and the Infrared camera.

From Table 3 generally the Body Rocket system performed well when compared to the infrared camera output. Given the resolution of the camera, it was found

that the camera's signal was a lot noisier than with the Body Rocket system. It must be noted, however, that for Rider 2, the data collected from the infrared camera did suffer severe data drops due to the camera frame capture and infrared flash ring being out of sync. This led to some images not illuminating the retro-reflective markers, leading to no markers being detected. During the testing this issue was addressed for subsequent riders and an effort was made to recover the data from Rider 2, however, in the results obtained is it is clear they are not consistent with the data collected from the other riders. Interestingly, in all cases (besides Rider 2), upon visual inspection, it can be seen that the Body Rocket system does indeed accurately detect the position of a rider on the saddle and, most importantly, correctly detects shuffles, and they are not associated with other positional changes such as pelvic rotation.

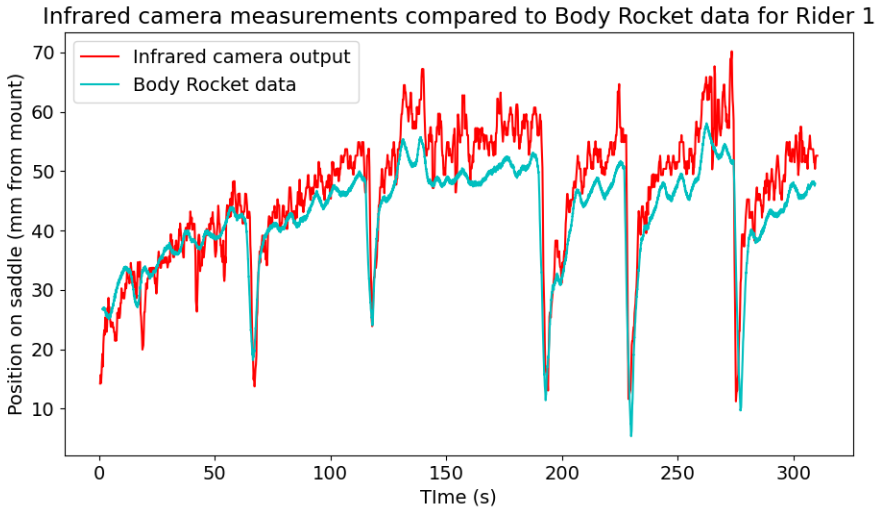


Figure 3.16: Plot showing the output from the Body Rocket system (cyan trace) and the output from the infrared camera (red trace).

Analysis of synchronous data acquired during 4km time trials indicates that the Body Rocket System exhibits a systematic underestimation of saddle displacement, relative to measurements obtained via infrared camera, by approximately 5mm. Intriguingly, the magnitude of this discrepancy appears to be positively

correlated with the duration of the displacement event; larger magnitude displacements are associated with a slightly greater divergence between the two measurement systems. Discrepancies in recorded displacement duration between the two systems may be attributable to their disparate methodologies for quantifying saddle movement. The infrared camera system tracks the positional data of spherical markers affixed to the posterior aspect of the rider, whereas the Body Rocket system quantifies forces and moments exerted upon the saddle. It is hypothesised that subtle variations in pelvic tilt may induce shifts in the rider's centre of mass relative to the saddle, thereby influencing the observed positional data.

3.4 Results

With the Body Rocket system's ability to determine the position of a rider on the saddle, analysis of the results can be continued. The mean power output for riders during the 4km time trial was $(355.24 \pm 0.61)W$. The power output profile and saddle position of riders during the time trial is shown in Figure 3. Identified saddle shuffles are shown by the blue line, with the duration of the movement depicted by the grey shaded areas. As can be seen in Figure 3, all the riders have different rates of shuffle.

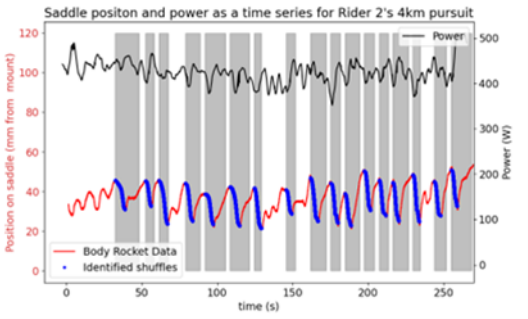


Figure 3.17: Rider 2 saddle position (Red trace) plotted with power (black trace) and identified shuffles (blue).

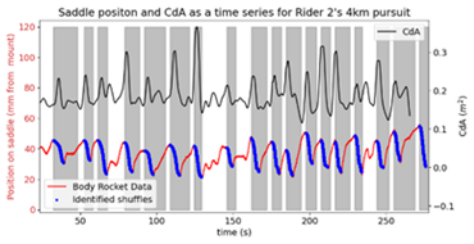


Figure 3.18: Rider 2 saddle position (Red trace) plotted with C_dA (black trace) and identified shuffles (blue).



Figure 3.19: Rider 3 saddle position (Red trace) plotted with power (black trace) and identified shuffles (blue).

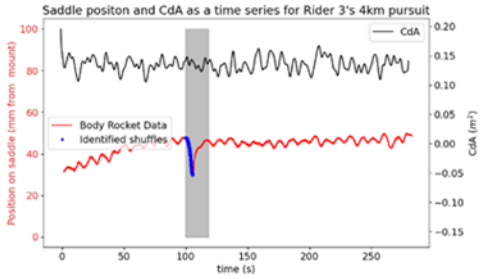


Figure 3.20: Rider 3 saddle position (Red trace) plotted with C_dA (black trace) and identified shuffles (blue).

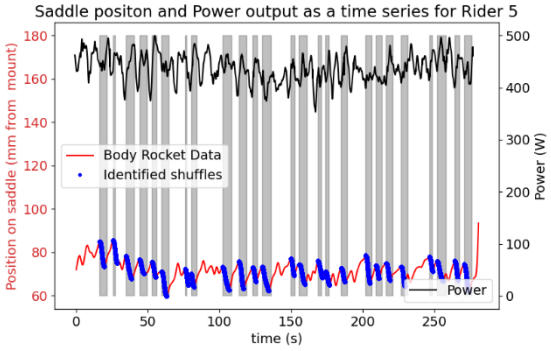


Figure 3.21: Rider 5 saddle position (Red trace) plotted with power (black trace) and identified shuffles (blue).

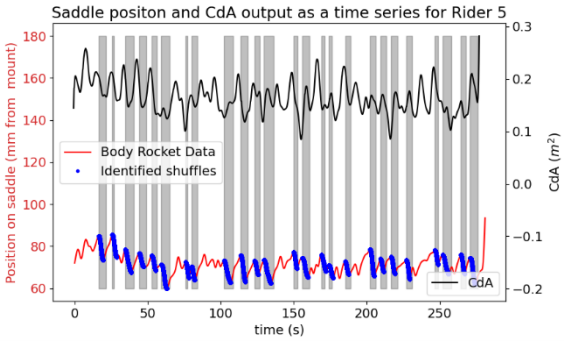


Figure 3.22: Rider 5 saddle position (Red trace) plotted with C_dA (black trace) and identified shuffles (blue).

In Table 4, the changes in C_dA and power output can be seen, with the respective impacts on power for the changes in C_dA and the number of shuffles.

Rider No	Avg Power(W)	Avg C_dA (m^2)	Avg ΔP (W)	Avg ΔC_dA (m^2)	No. Shuffles
Rider 1	374.67 ± 0.18	0.213 ± 0.005	-0.22 ± 8.51	-0.021 ± 0.007	5
Rider 2	427.73 ± 0.16	0.219 ± 0.005	-11.87 ± 4.52	0.047 ± 0.01	19
Rider 3	371.61 ± 0.21	0.159 ± 0.004	12.42 ± 0.21	0.004 ± 0.004	1
Rider 4	364.68 ± 0.24	0.140 ± 0.004	20.48 ± 0.19	-0.024 ± 0.005	2
Rider 5	437.83 ± 0.15	0.177 ± 0.005	-3.815 ± 6.19	0.003 ± 0.008	9
Rider 6	285.65 ± 0.85	0.171 ± 0.001	0.690 ± 2.82	-0.010 ± 0.003	16
Rider 7	265.84 ± 0.90	0.250 ± 0.004	-9.30 ± 16.50	-0.006 ± 0.014	6
Rider 8	307.79 ± 0.99	0.159 ± 0.001	-13.29 ± 6.21	0.0002 ± 0.003	14
Rider 9	356.99 ± 1.30	0.165 ± 0.006	2.93 ± 4.96	-0.021 ± 0.01	23
Rider 10	359.61 ± 1.11	0.218 ± 0.006	4.930 ± 7.07	-0.002 ± 0.01	9
Avg Rider	355.24 ± 0.609	0.187 ± 0.004	0.296 ± 5.72	-0.032 ± 0.0073	10

Table 4: Table of average powers and C_dA for each rider, alongside the average difference when a rider shuffles and the number of shuffles for each rider.

3.5 Discussion

The data in Table 4 shows the Rider number, alongside their average power for the effort, their average C_dA, the average change in power for each shuffle, the average change in C_dA for each shuffle and the number of shuffles. The average power from the riders range from $(265.84 \pm 0.90)W$ to $(437.83 \pm 0.15)W$ and the average C_dA's range from $(0.159 \pm 0.001)m^2$ to $(0.250 \pm 0.004)m^2$. Interestingly, there is also a large range in the number of shuffles, from one rider only experiencing 1 shuffle to another rider experiencing 23 shuffles.

To evaluate whether the change in power (ΔP) and C_dA (ΔC_dA) was significant, the data is presented with the standard error for (ΔP) and C_dA (ΔC_dA) across the change measured for each shuffle. Overall, the results are mixed; they show that half (5) of the participants saw a reduction in power, and the other half (5) saw an increase. This can also be said for C_dA, where 4 of the riders saw an increase in C_dA with the remaining 6 seeing a decrease; however, with the standard error associated with both the power measurements and C_dA measurements, some of the results are within the noise.

The mean power output of the riders was $(355.24 \pm 0.609)W$, accompanied by an average C_dA of $(0.187 \pm 0.004)m^2$. The mean power change associated with a shuffle was $(0.296 \pm 5.72)W$, while the mean C_dA change was $(-0.032 \pm 0.007)m^2$. With the substantial error associated with the ΔP and a standard deviation in ΔP of $10.67W$ and a standard deviation of $0.020m^2$ for ΔC_dA, this suggests substantial variability. This leads to the conclusion that the effect of shuffling on power and C_dA is subject to individual differences.

When looking at the individual data, notable cases do arise, such as Rider 2. In the case of Rider 2 they shuffled 19 times, the second highest number when compared to the other riders and as can be seen in Figure 3.18 there is, on average, an associated increase in C_dA of $(0.047 \pm 0.01)m^2$, this increase in C_dA is also the

largest when compared to the other riders and falls outside of the standard error of the C_dA measurement and outside the error measurement for each ΔC_dA. It is also noteworthy to mention that Rider 2 also sees a substantial reduction in power, $(-11.87 \pm 4.52)W$ when compared to the losses which the other riders, who did lose power, experienced. With these results, it can be concluded that shuffling, for this rider, does have a negative impact on their performance as their C_dA increases and power decreases.

Conversely, a rider who saw an increase in their power and a decrease in their C_dA was Rider 4. Rider 4 experienced 2 shuffles, and for each shuffle, there was a slight decrease in C_dA; this is also seen in the average difference data, where the average difference in C_dA was $(-0.024 \pm 0.004)m^2$, the second largest change in C_dA behind Rider 2. Rider 4 also saw an increase in power, the largest increase of $(20.48 \pm 0.19)W$. Both of these differences are outside of the standard error for both the C_dA measurement and the power measurement, and show that for this rider, shuffling was in fact useful for them. In Table 4, it must also be mentioned that there are a number of riders where some of the measured ΔP and ΔC_dA values are smaller than their associated standard errors. This may indicate that for these riders, where the measurement is within the noise, shuffling has a negligible impact on both their C_dA and power. As a result to further continue these studies it'd be beneficial to recruit a mixed cohort of riders both male and female and have them carry-out repeat results over the course of a number of days and weeks to further quantify the impact of shuffling on their performance and determine whether, as this study has demonstrated if shuffling has a positive, negative or negligible impact on ones performance.

Uncertainty considerations

When interpreting the results presented in this study, potential sources of error and measurement uncertainty should be considered. The uncertainty values presented in these results are the standard error of the mean (SEM) the mean; this was selected as it is an indicator of variation in the riders' C_dA and Power. As established in Chapter 2, the Body Rocket system has a measurement uncertainty in C_dA of $0.002m^2$; as a result, any results where the SEM is smaller than the measurement uncertainty should be taken with caution. In this study, the only riders which have a SEM smaller than the known measurement uncertainty is Rider 8 and Rider 6, however, these are only in the average C_dA measurements and not in the average change in C_dA (ΔC_dA). Therefore, analysis of the ΔC_dA and respective SEM can be interpreted with greater confidence. The associated measurement error of the power meter as claimed by Body Rocket Ltd is $\pm 0.1\%$ (Body Rocket 2026). All the results presented in this study show an SEM larger than the measurement error from Body Rocket Ltd; therefore, the results in power can be interpreted with confidence.

Impact on Power

With respect to the impact on power output, the variation may be attributable to factors relating to bike set-up or the manner in which a rider shuffles. The manner of adjustment could influence this outcome, as some cyclists might apply force to the handlebars to reposition themselves towards the rear of the saddle, whereas others might exert greater pressure on the pedals, thereby reducing their weight on the saddle. Both methodologies would result in an adjustment of position; however, should a cyclist reduce their weight on the saddle, they would need to offset this by increasing the applied force on the pedals. Conversely, exerting force on the handlebars would necessitate a reduction in leg power to facilitate

repositioning on the saddle.

Saddle setback and crank length are factors that impact the functional range of the hip flexion and ultimately influence rider comfort. Discomfort in the time trial position is common because of the constraints that the need to maintain an aerodynamic position has on the rider, and the small contact points between rider and bicycle (i.e. handlebars, saddle and pedals), with the trunk bent forwards to allow the rider to contact each point simultaneously. This tucked position, combined with riding for a prolonged period of time at high power outputs, leads to frequent discomfort (Polanco, Suárez and Muñoz 2017), and likely increases the desire to change posture on the bicycle. Therefore, even though riders in this study utilised their preferred cycling position, discomfort was experienced from maintaining the position for a period of time under high levels of physiological stress. Indeed, results from previous studies have suggested that even though more aggressive aerodynamic positions might provide an aerodynamic advantage, it does not necessarily lead to an improvement in cycling performance due to reduced cycling efficiency and power output production capability (Fintelman et al. 2015), as well as leading to adverse effects on the comfort of the rider.

Whilst evaluating how shuffling impacts power, the question arises as to whether specific shuffle characteristics, namely shuffle rate and shuffle size, influence the magnitude of this effect. Consequently, Figure 3.23 presents the average change in power against each rider's average shuffle rate, while Figure 3.24 presents the average change in power against average shuffle size.

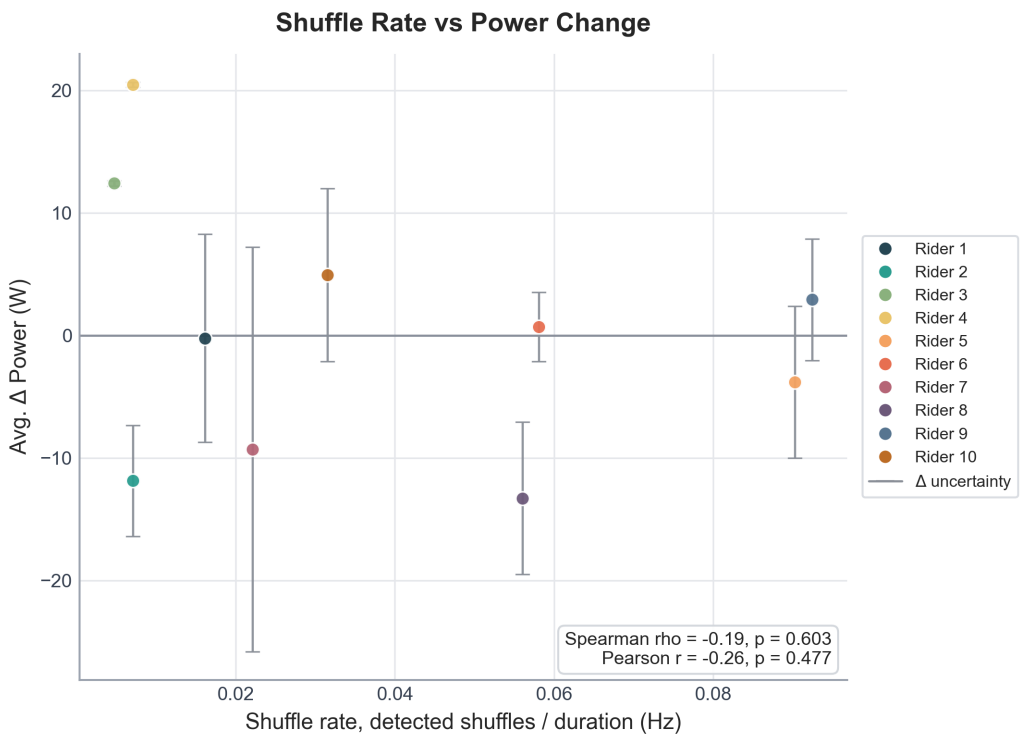


Figure 3.23: Relationship between each rider's average shuffle rate and average change in power during shuffling. Each point represents one rider, with rider identity indicated by colour in the legend. Error bars show the uncertainty in the average power change. The red line shows the fitted linear trend.

Visually in Figure 3.23 there is variation in the shuffle rate and average change in power, Δ power, for each rider. With the significant error bars presented in Δ power and variation with the data collected, it can be concluded that there is no clear correlation between one's shuffle rate and its impact on their power output as a result of a shuffle.

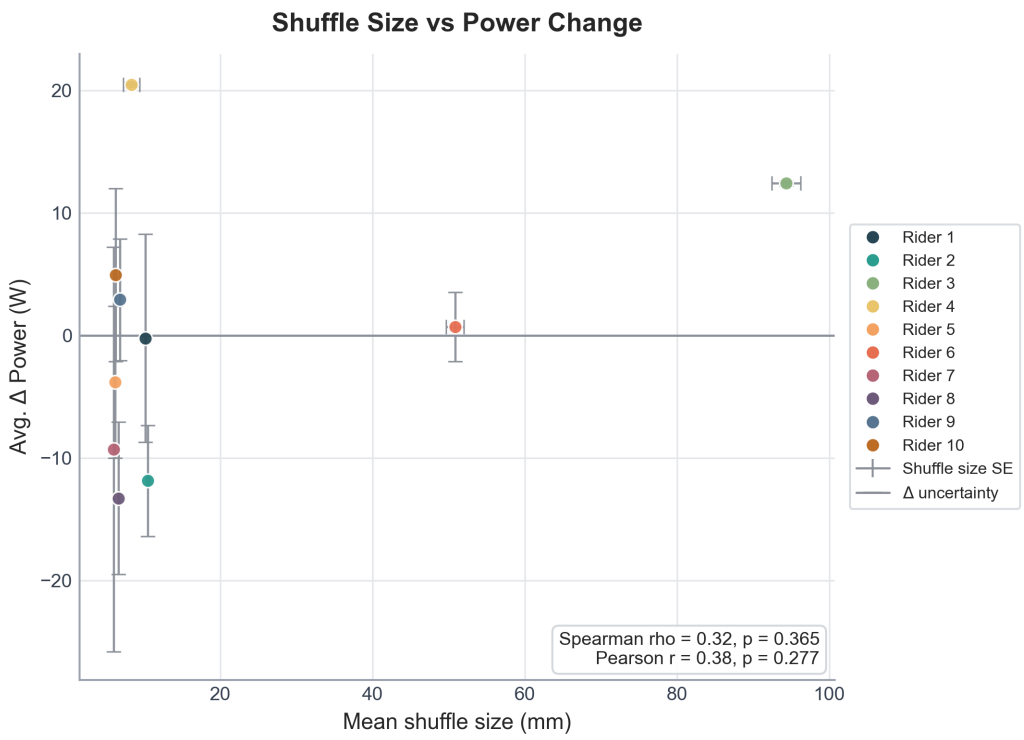


Figure 3.24: Relationship between each rider's average shuffle size and average change in power during shuffling. Each point represents one rider, with rider identity indicated by colour in the legend. Error bars show the uncertainty in the average power change and the average shuffle size. The red line shows the fitted linear trend.

Figure 3.24 presents the average change in power with the average shuffle size for each rider, with error bars representing the SEM for both the average change in power and shuffle size. These results show that each athlete is consistent in the size of their shuffles, producing repeatable, unique shuffles for each rider, however, there is no clear trend indicating that there is a relationship between shuffle size and its influence on the average change in power.

Impact on CdA

Overall, in the same vein as the impact on power, the variability in the impact of shuffling on C_dA may be due to the way in which a rider shuffles. The impact on drag of the tilt of a helmet can be seen from Alam et al. (2014), where they were testing the drag of different helmets at differing tilt angles. In this study, they found an increase in drag when a rider's head is tilted upward; this could be a possible reason for the spikes in drag. When some riders shuffle, they possibly move their head up. The reason for the decrease in drag for some riders may be due to the shape of the rider (Crouch et al. 2017) or due to the way the rider shuffles, possibly flattening the back, therefore allowing for a smoother transition for the air around the rider. For future research, specifically looking into the impact of shuffling on aerodynamics, could be to place inertial measurement units on the joints of the rider. With a 3-dimensional scan of the rider, this will allow for a model to be made and the way in which the riders shuffle to be identified. In tandem with the Body Rocket system, computational fluid dynamics (CFD) could be used to verify the changes in C_dA with a 3-dimensional dynamic model in the simulation. Additionally, a venue such as Catesby tunnel (Catesby Tunnel 2025) could be used, which is a more controlled environment than a velodrome and does not have corners, which will have an impact on the aerodynamics or tailwind effects due to riding the same loop. The motivation for using a different venue would be so that the CFD simulation could be compared fairly to the experimental results. To quantify the impact of shuffling the velodrome, however, is a better venue to quantify what the impact is on performance, as it is a setting where competitions take place.

From the results obtained there is a possibility that latent variables are causing the apparent rider dependences, because of this, it may be valuable to investigate possible latent variables such as the saddle set-back, saddle height, saddle angle

etc. A simple investigation could be carried out where the same rider rides a 4km pursuit effort, where they ride a bike in different configurations. Alternatively, another study could be carried out where shuffles are induced with a low-friction interface or by changing other aspects of the bike fit.

In the same vein as the analysis on power, other interesting metrics to evaluate are the impact of shuffle rate and shuffle size on C_dA. These can be seen in Figure 3.25 and Figure 3.26.

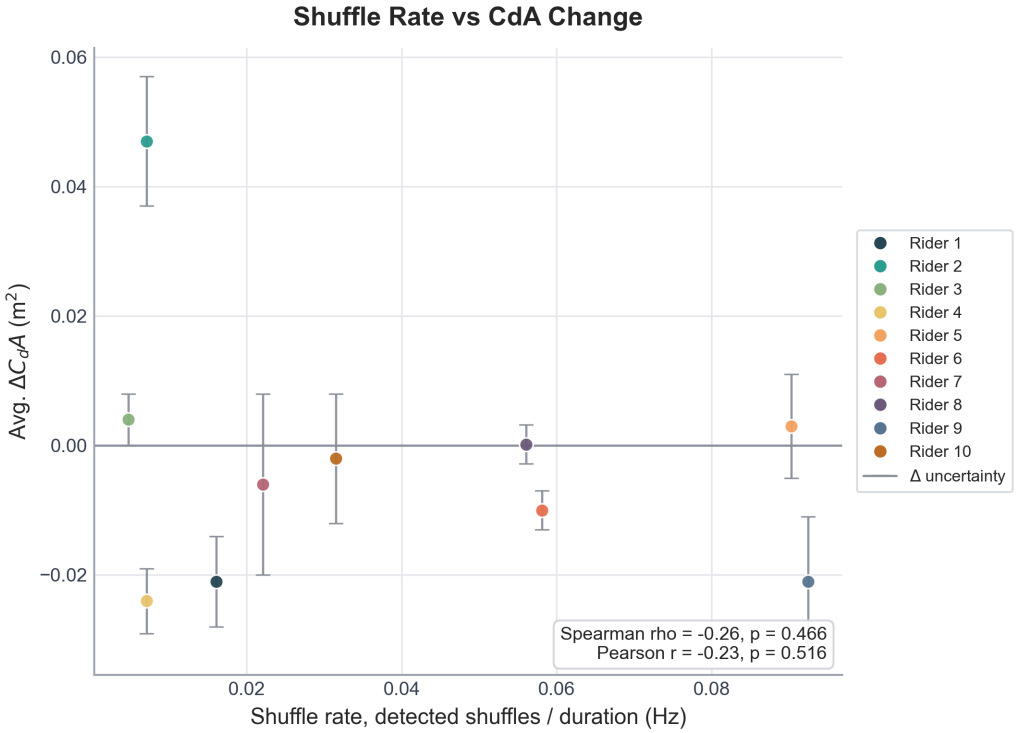


Figure 3.25: Relationship between each rider's average shuffle rate and average change in C_dA during shuffling. Each point represents one rider, with rider identity indicated by colour in the legend. Error bars show the uncertainty in average C_dA change. The red line shows the fitted linear trend.

Figure 3.25 presents the average change in C_dA against each rider's average shuffle rate. Visually, there is not a strong correlation between one's shuffle rate and the impact on C_dA. One outlier can be seen in Rider 2 with a significant increase in

C_dA and low shuffle rate; however, with both Rider 3 and Rider 4 also showing a low shuffle rate, and not a substantial increase in C_dA, it cannot be concluded that the shuffle rate impacts the change in C_dA.

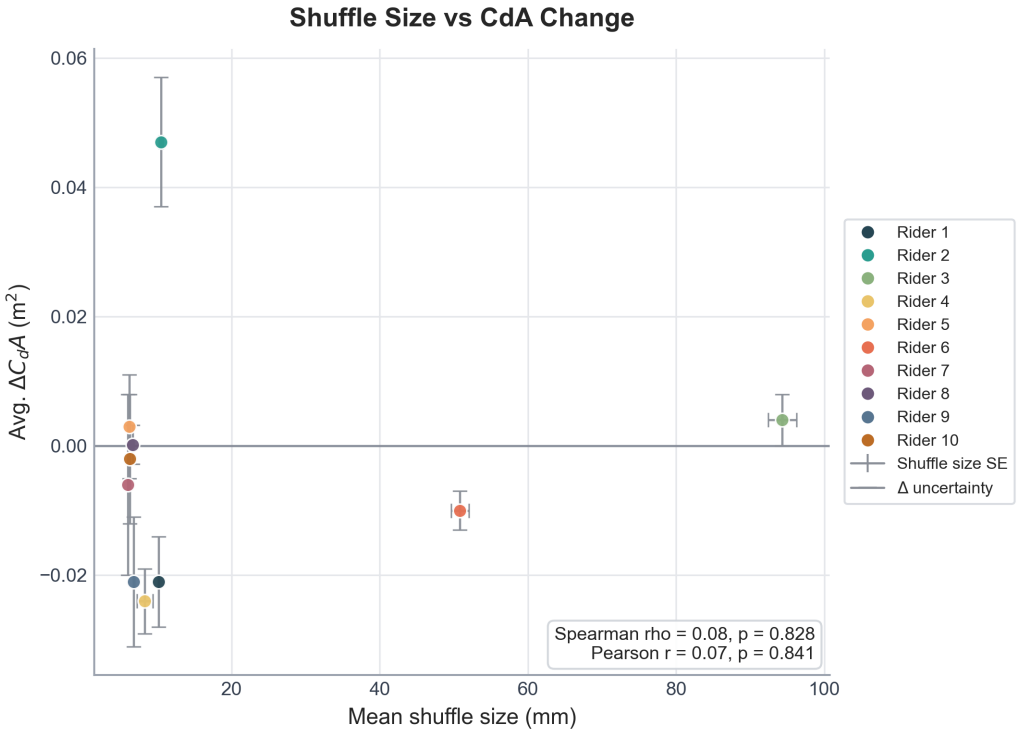


Figure 3.26: Relationship between each rider's average shuffle size and average change in C_dA during shuffling. Each point represents one rider, with rider identity indicated by colour in the legend. Error bars show the uncertainty in average power change and average shuffle size. The red line shows the fitted linear trend.

Figure 3.26 presents the average change in C_dA with the average shuffle size for each rider, with error bars representing the SEM for both the average change in C_dA and shuffle size. As with the analysis of power, each rider is consistent with the size of the shuffles and is repeatable; however, there is no clear relationship between shuffle size and change in C_dA.

3.6 Conclusion

The results from this study demonstrate good agreement between data recorded on the infrared camera and the Body Rocket system. Our data demonstrated that all participants exhibited saddle shuffles during the 4 km time trial effort, although there were clear between-rider differences in both frequency and magnitude. Moreover, the impact of shuffling on cycling power output appears to be rider dependent, with some riders experiencing larger reductions in power output for the same shuffle magnitude compared to others and a few experiencing an increase. This is further supported when investigating the nature of the shuffles in both their size and frequency. Whilst the majority of riders had average shuffles on the magnitude of 6-10mm and two outliers with shuffle sizes for Rider 3 and Rider 4 being (94.33 $\pm$ 1.89)mm and (50.84 $\pm$ 1.19)mm respectively, the impact on C_dA and power did not follow any particular trend. This was also found when investigating shuffle frequency. Although shuffle frequency varied more then the average shuffle size for each rider no trend was found relating shuffle frequency to any affect on either performance metric. This further supports that there is another rider-specific factor which determines the impact of shuffling on their performance. With this, it may be concluded that shuffling should be reduced as much as possible in professional cycling; however, as this study is the first study where the impact of shuffling has been quantified in field with respects to its impact on power and C_dA via a DFDM system, there are areas of further research which can be explored alongside optimisations of this study. As it became apparent that latent variables are present that impact the shuffling of a rider, to further verify the results obtained a study could be constructed that induces shuffles for the same rider, this may be creating a low friction interface between the rider and their saddle or changing certain dimensions of the bike such as the crank length, the saddle height, offset etc. A further study could be to isolate the impact of

a high-power output to see if a higher output power induces more shuffling in a rider and therefore has a negative impact on their performance due to shuffling. The way in which a rider shuffles may impact the effect on C_dA, however, it may also be due to the shape of the rider. To investigate this further, IMUs could be used to monitor the position of all of the points of a rider, with a venue such as Catesby tunnel and CFD, the reason why these changes happen could be investigated. Additionally, the question arises, as the Body Rocket system measures the forces and moments on the contact points of the bike, would it be possible to know the position of a rider on the bike, replacing the need for IMUs to monitor the entire position of the rider?

Key point summary

- During this study, a novel optical device (with computer vision capabilities) was created to monitor the position of a rider on the saddle.
- With this device validated and calibrated, it was used to successfully validate the Body Rocket system and its ability to locate the position of a rider on the saddle.
- Within this study, the impact of shuffling was examined, and it became clear that some riders experience a marked drop in performance, while others improve. This suggests that shuffling itself is not inherently harmful. Instead, the manner in which a rider shuffles is likely the key factor influencing any reduction in performance.

Chapter 4

Positional Classification using the Body Rocket System

As established in Chapter 3, it is possible for the Body Rocket system to track the position of a rider on the saddle. This can be seen as a useful tool as it allows for the rider to know how well they are seated on the saddle and how much they shuffle. As a natural progression from tracking one's position on the saddle using a single Body Rocket sensor, this study poses the question: "Can all of the Body Rocket sensors combined determine the position of a rider on the bike as a whole?" If this were possible, it'd provide another layer of insight into rider testing and training, allowing riders to train for certain positions that are better suited to specific sections of race courses, for example.

4.1 Introduction

Optimising rider position on the bike is a balance between aerodynamic efficiency and kinematic efficiency and is a key factor in cycling performance (Crouch et al. 2017). Reducing a rider's C_dA during competitive events such as time trials

could substantially enhance their likelihood of success, particularly when real-time data is available to facilitate adjustments, as evidenced in Peeters, Garimella and Verwulgen (2020), where a live video feed coupled with a vibrating device was used to offer real-time positional feedback. Previous research has illustrated the potential of inertial measurement units to provide feedback to cyclists during time trials, assisting them in maintaining optimal positioning on the bike (Winter et al. 2023).

As previously explored and discussed in Chapter 2, the on-bike sensor system (Body Rocket Ltd, Sussex, UK) directly measures the drag force of a rider through the use of sensors at the contact points of the bike. These contact points measure not only aerodynamic drag but also weight and moments, as shown in Figure 4.1.



Figure 4.1: An image of a Body Rocket-equipped bicycle with red arrows indicating the forces and moments that can be measured on the system.

The aim of this study is to determine whether these measurements can be used to predict the position of a rider on the bike. This investigation consists of three parts:

- Verify whether machine learning techniques can identify rider positions from a wind tunnel dataset and determine which model is most suitable for this task.
- Evaluate the accuracy of the selected model using the full wind tunnel testing dataset.
- Assess whether a model can be trained on open-road cycling scenarios.

All data acquisition methods were approved by the University of Kent School of Sport and Exercise Science ethical approval committee.

If it were possible to classify a rider's position on the bike, this could be utilised as an effective training tool. This is because the optimal position in certain race scenarios may differ, for example, on inclines where a more upright position may be optimal, or on flat sections where a more aerodynamic position may be optimal. As a result, if it were possible to identify when a rider is in these positions throughout a training session, they could train to these positions, moving from a paradigm of one position fits all to fitting different positions to different conditions. If these models were deployed live, it would be possible to track a rider's position on the bike and, with knowledge of the conditions and trained positions, advise on the best position to take to optimise their performance on the course.

Positional Identification Methods

Methods do exist to give an indication of a rider's position on the bike, and these come in the form of inertial measurement units (IMUs), optical sensors and computer vision solutions. Some sensor systems, such as Aerosensor's Aerobody (Aerosensor 2025), use optical techniques to determine the location of a rider's torso. In a wind tunnel setting, other techniques can be used, such as edge detection. Since the rider is on a static plate, cameras can be mounted around

them. With edge detection, it is then possible for the rider to see an outline of themselves, ensuring they maintain the same position during testing (Hub 2025a). This is useful as it allows consistent results to be obtained; however, it lacks in-field transferability.



Figure 4.2: An example of edge detection being used in a wind tunnel environment.

Furthermore, IMUs have been shown to work in outdoor conditions, such as in Evans et al. (2021b), where accelerometers were used to determine trunk acceleration. For applications in real-world conditions, all of these solutions have limitations. The wind tunnel creates a highly controlled environment that does not reflect outdoor conditions. The Aerobody does not monitor a rider's saddle position and only tracks the head and torso, and IMUs must be attached to the rider, which could disrupt airflow. If machine learning techniques can be used to predict the rider's position on the bike from Body Rocket system data, this would address all of these issues and could offer a viable solution for real-world testing.

Numerous studies in the literature have explored classification challenges using cycling data. For instance, de Smit et al. (2024) employed a convolutional long short-term memory neural network to predict cyclist manoeuvres using IMUs.

Studies outside of cycling but still within sport have investigated the application of classification techniques to predict foot strike patterns from a load-measuring insole (Moore et al. 2020). The classification accuracy was found to be greater than 90% for all models tested.

4.2 Theory

For this investigation, three multi-class classifiers were examined: the Gaussian Mixture Model (GMM), Support Vector Machine (SVM) and Gradient Boosted Machine (GBM). The motivation for selecting these models was their explainability. Explainability is important in this application because more explainable models are likely to run at lower latencies, meaning that if this were applied in field for a live use case, it would be able to quickly predict the position of a rider on the bike. Additionally, if a model is more explainable, the results can be trusted more, as it becomes possible to trace why certain predictions were made.

4.2.1 Gaussian Mixture Model

The Gaussian Mixture Model (GMM) is a model widely used in governmental economics (Andronicеanu, Kinnunen and Georgescu 2020), environmental economics (Salamzadeh et al. 2022) and environmental condition prediction (Kim et al. 2021), to name a few. Due to this wide range of applications and the stipulation that the forces on the saddle could be modelled as a superposition of Gaussian curves, it was thought that it could be a simple and explainable model for this dataset. A GMM is a model that takes a weighted summation of M component Gaussian densities (Reynolds et al. 2009).

$$p(x|\gamma) = \sum_{i=1}^{M} w_i \, g(x|\mu_i, \Sigma_k) \tag{43}$$

In the case of Equation (43), x is N-dimensional data, where w_i is the weight for each Gaussian in the superposition and $g(x|\mu_i, \Sigma_k)$ represents each component Gaussian density. Each Gaussian density $g(x|\mu_i, \Sigma_k)$ can be expressed as:

$$g(x|\mu_i, \Sigma_k) = \frac{1}{(2\pi)^{N/2}|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_k^{-1}(x-\mu_i)\right\} \tag{44}$$

In Equation (43) and Equation (44), Σ_k is the covariance matrix. The GMM is a very straightforward model to train. First, the number of classes is identified, which determines the number of clusters. After this, the distance between the data and the initialised Gaussian distributions is calculated. This allows each data point to have partial classification depending on its distance from each distribution. The Gaussian distributions are then fitted to the data using the mean, variance in each dimension and the covariance matrix (either full, tied, diagonal or spherical). This process is iterated until the weights and distributions converge.

The importance of the different covariance matrices comes down to trade-offs between flexibility and simplicity. The full and tied methods are generally more complex and are likely to perform better when correlations exist in the data, whereas the diagonal and spherical models are likely to perform better on uncorrelated data. However, they are simpler and therefore more explainable.

4.2.2 Support Vector Machines

Support Vector Machines (SVMs) are widely used in various fields, from medicine (Guido et al. 2024) to financial forecasting (Kim 2003) and text categorisation (Joachims 1998). The foundation of SVMs is Support Vector Classifiers (SVC). An

SVC works by separating clusters of data by a line, plane or hyperplane, depending on the dimensionality of the data. When the data is not linearly separable in the original feature space, SVMs employ kernel functions to implicitly map the data to a higher-dimensional space. SVMs can be split into two major categories: Linear SVM and Nonlinear SVM. Within this study, the viability of four different SVMs was evaluated to see whether they were a suitable candidate for the classification task at hand: two Linear SVMs, a Polynomial SVM and a Radial Basis Function (RBF) SVM.

A linear SVM is based on the principle of constructing a linear decision boundary, typically expressed as the hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$. When the weight vector $\mathbf{w}$ and feature vector $\mathbf{x}$ are represented in $\mathbb{R}^N$, this formulation naturally extends from a simple line in two dimensions to a hyperplane in an N-dimensional feature space. In practice, a linear SVM seeks the hyperplane that maximises the margin between classes, thereby improving generalisation performance. Geometrically, this results in the separation of data points by the optimal hyperplane in N-dimensional space, enabling effective classification even in high-dimensional settings (Noble 2006). All the SVMs used in this study were from scikit-learn. The two linear models; 'LinearSVC' (scikit-learn developers 2025a) and 'SVC (linear)' (scikit-learn developers 2025b) follow similar principles, however, they are separable in how they work. 'LinearSVC' works without the use of a linear kernel and uses the squared hinge loss function, as opposed to the hinge loss function.

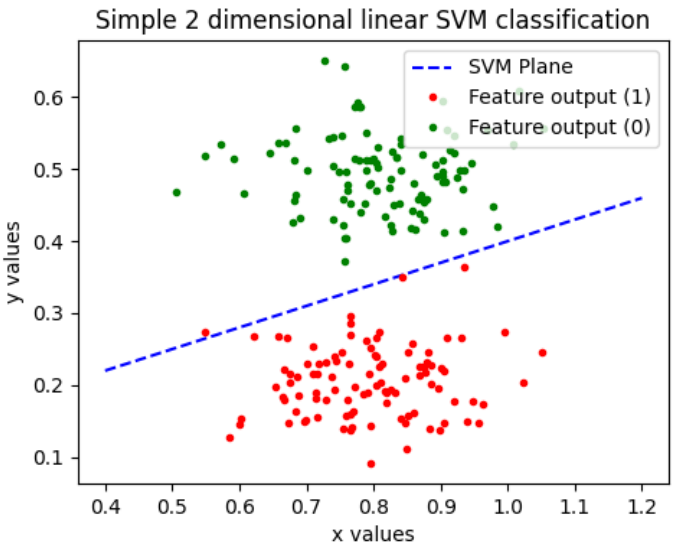


Figure 4.3: Simple example of how a linear Support Vector Machine works with regard to the classification of data.

C is a variable when training a SVM which can be altered. If data is well clustered then a high C will increase classification accuracy, however, if the data is not well clustered C may want to be reduced to increase the likelihood of correct classifications. Along the same principle as the Linear Support Vector Machine, a Polynomial SVM will use a polynomial function to separate the different data classes. For the case of the RBF, it is defined as (Suykens and Vandewalle 1999):

$$K(x, x') = \exp\Big(-\gamma \left\|x - x'\right\|_2^2 / \sigma^2\Big) \tag{45}$$

Where x and x' are input vectors, γ is a width parameter and σ is a scaling term.

4.2.3 Gradient Boosted Machine

The Gradient Boosted Machine (GBM) is an ensemble learning method. Ensemble learning is where a number of weak learners are combined to create a model

which has a greater predictive performance than one of those learners alone. Examples of ensemble learning methods include bagging and boosting. The bagging method constructs multiple base learners and trains them in parallel. A popular example of a bagging ensemble method is the Random Forest method. Alternatively, boosting doesn't just combine weak learners; it boosts the performance of those weak learners to increase the performance of the entire ensemble model.

GBMs are a decision tree-based approach to both classification and regression problems. In the case of GBM's or any tree based method a tree is simply a rule that splits data into smaller groups, similar to a splitting flow chart. Leaves, therefore, are the terminal nodes in a sequence of trees.

From the work of Friedman (2001), they work by taking a data set which can be expressed by the variable y. Trees are constructed, and the weak learners are boosted with respect to a loss function $L(y, \rho)$ where ρ can be expressed as:

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m)) \tag{46}$$

With,

$$\mathbf{a}_m = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^{N} [-g_m(\mathbf{x}_i - \beta h\ (\mathbf{x}_i; \mathbf{a}_m))] \tag{47}$$

It is trained via the following algorithm (Friedman 2001):

Algorithm 2 Gradient Boosting Algorithm (J. H. Friedman)

1: $F_0(\mathbf{x}) = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, \rho)$
2: **for** $m = 1$ to M **do**
3: $\quad \tilde{y}_i = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, \quad i = 1, \ldots, N$
4: $\quad \mathbf{a}_m = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^{N} [\tilde{y}_i - \beta h(\mathbf{x}_i; \mathbf{a})]^2$
5: $\quad \rho_m = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m))$
6: $\quad F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a}_m)$
7: **end for**=0

Early Stopping and Logarithmic Loss

To improve generalisation and avoid over-fitting, GBMs are often trained using early stopping in conjunction with a validation metric such as log loss.

For classification tasks, the performance of the model is commonly evaluated using the logarithmic loss (or log loss), which measures the divergence between the predicted probabilities and the actual labels. In the binary case, it is defined as:

$$\text{log loss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right], \tag{48}$$

where $y_i \in \{0, 1\}$ denotes the true label, and p_i is the predicted probability that $y_i = 1$. Lower log loss values indicate better-calibrated probabilistic predictions. In this study, log loss is also used as an indicator of predictive performance and will have associated dimensionless units of Positional Log Loss Units (PLLU) (Seto et al. 2022).

For a multi-class classifier, the log loss formula takes the form:

$$\text{log loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\,\log\left(\hat{p}_{i,c}\right) \tag{49}$$

During training, the model's performance on a separate validation set is monitored at each boosting iteration. If the validation log loss fails to improve after a fixed number of rounds, known as the early stopping rounds, training is halted. This mechanism serves as a form of regularisation, helping to prevent overfitting by terminating training when the model begins to fit noise in the training data. The final model is selected from the iteration that yielded the best validation performance.

Early stopping is especially valuable when the number of boosting rounds (i.e., trees) is large or when the learning rate is small, as it allows for automatic selection

of the optimal model complexity.

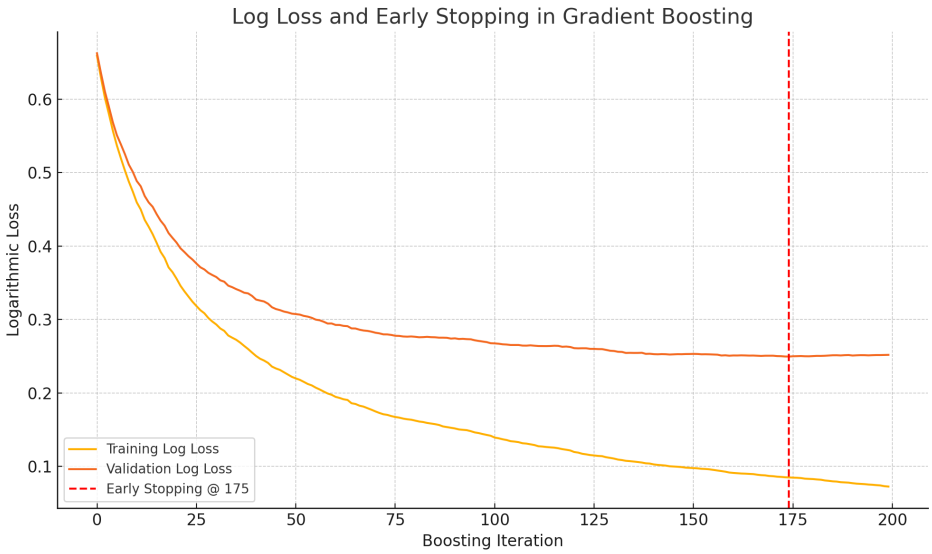


Figure 4.4: Example plot showing log loss against boosting iterations for a training set (orange trace) and testing set (yellow trace) with early stopping (red dashed trace) at 175 boosting iterations.

The plot Figure 4.4 illustrates how the log loss evolves over boosting iterations for both the training and validation sets. The training log loss continuously decreases as the model fits the data more closely. However, the validation log loss reaches a minimum and then begins to increase, indicating the onset of overfitting. The red dashed line marks the early stopping point, selected as the iteration where the validation log loss is minimised. Stopping training at this point prevents the model from overfitting to the training data and improves generalisation.

Cross validation

During the training of the Gradient Boosted Machine, cross-validation was implemented. Cross-validation is a statistical method used in model training to increase generalisability and performance. During cross-validation, the dataset is

partitioned into multiple subsets, also known as folds. The model is then iteratively trained on a portion of the data and validated against the remaining subset. This process reduces the risk of overfitting and is generally considered to be superior when compared to just a single test train split. This allows for parameter optimisation to be performed, which can then be compared to the truly unseen test (also known as holdout) data (Allgaier and Pryss 2024). A visualisation of cross-validation can be seen in Figure 4.5.

Fold	**S1**	**S2**	**S3**	**S4**	**S5**	**Holdout Set**
Fold 1	Test	Train	Train	Train	Train	Unseen
Fold 2	Train	Test	Train	Train	Train	Unseen
Fold 3	Train	Train	Test	Train	Train	Unseen
Fold 4	Train	Train	Train	Test	Train	Unseen
Fold 5	Train	Train	Train	Train	Test	Unseen

Train	Training subset
Test	Validation/testing subset
Unseen	Independent holdout dataset not used during cross-validation

Figure 4.5: Illustration of 5-fold cross-validation with an additional independent holdout dataset reserved for final model evaluation.

Data Pre-processing

The first step in the data preprocessing is removing outliers outside 3 times the standard deviation and normalising the data between 0 and 1. For many models that take high-dimensional data, strong correlations (multicollinearity) amongst predictors can reduce generalisation performance and cause overfitting (Shrestha 2020). Expectedly, for models which assume feature independence, such as Naïve Bayes, correlations can deteriorate classification performance (Blanquero et al.

2021; Zaidi et al. 2013). Including many correlated or redundant predictors encourages overly complex models that may fit noise rather than signal, thereby degrading predictive performance (Blanquero et al. 2021).

One method that can be used to decorrelate features is a principal component analysis (PCA). A PCA is a data analysis method that reduces complex high-dimensional data (Shlens 2014) to a smaller number of dimensions whilst retaining as much of the variance associated with the original dataset as possible. To visualise correlations in a given dataset, a correlation matrix can be implemented. A correlation matrix is a visualisation technique that performs a Pearson's correlation between each feature of a data-set Equation (50). Once the data is visualised, an idea of how correlated the data is can be determined before and after the PCA.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2} \cdot \sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{50}$$

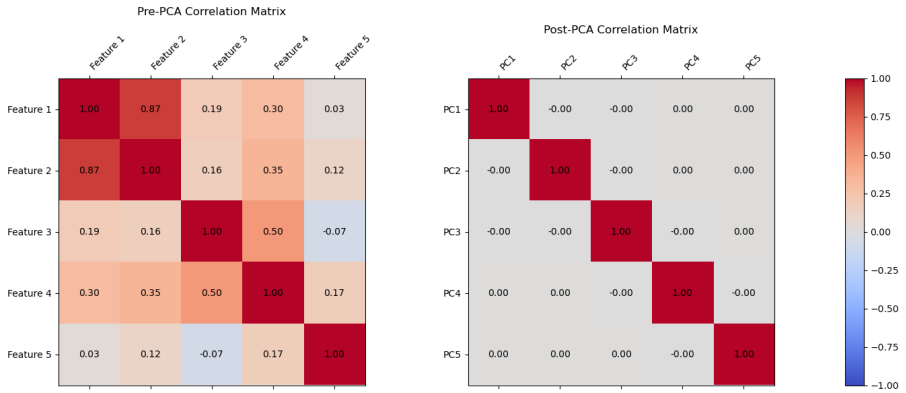


Figure 4.6: Two correlation matrices, before (left) and after (right) the dataset has undergone a principal component analysis.

4.3 Methods

Four datasets were provided by Body Rocket Ltd., originating from controlled tests performed in two different wind tunnel facilities: the University of Southampton's WT1 (Hampshire, UK) and Flanders' Bike Valley's WT2 (Beringen, Belgium). Across all four datasets, a single rider adopted four consistent positions (head down, baseline, head up, and hands on the aero-pads), as detailed in Figure 4.7. Continuous video monitoring within the wind tunnel during each run verified the rider's maintenance of the specified position (Figure 4.7).

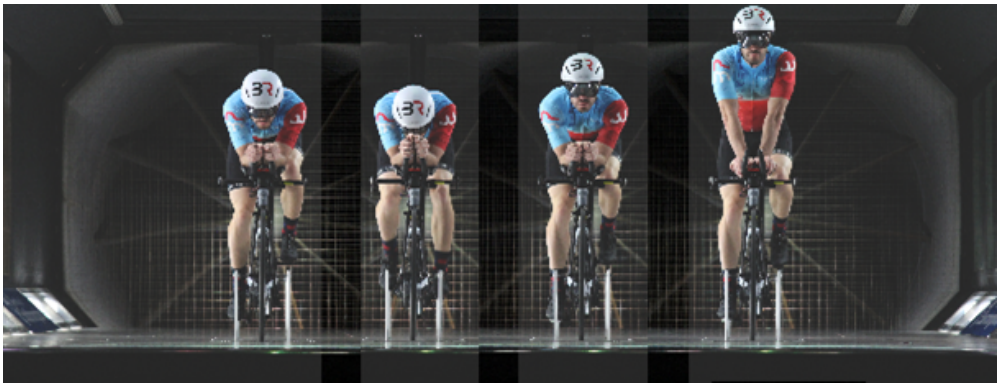


Figure 4.7: Images from WT1 of a rider adopting four distinct cycling positions. From left to right the positions are: baseline, head-down, head-up and hands on the aeropads.

For WT1, wind speeds of $15.2ms^{-1}$, $13.2ms^{-1}$, and $11.2ms^{-1}$ were investigated. For WT2, two wind speeds, $16.2ms^{-1}$ and $12.8ms^{-1}$, were analysed. Initially, the data pre-processing was performed. Subsequently, a multi-class Gaussian Mixture Model (GMM), Support Vector Machine (SVM) and Gradient Boosted Machine (GBM) were trained on the cleaned data to classify the four distinct cycling positions. To obtain a quick indication of model viability, GMM and SVM models were trained and tested on one dataset collected from WT1. The datasets were partitioned, with 80% allocated for training and 20% reserved for subsequent blind testing of the model on this initial dataset.

For the GBM, the 80% used for training also underwent 5-fold cross-validation on the training set. Within the training dataset the 64 channel data from the Body Rocket system was distilled down into 8 usable features. This is because there are a number of confounding variables within the raw Body Rocket dataset which may lead to the models being trained on features which are related to rider position but not causally related. Exampled of features in the dataset which could lead to these misclassification's include power and airspeed. It is likely, in general, that power will be reduced for more compact positions, however, there are circumstantial variables not captured within the raw Body Rocket data which may influence a riders power output under different conditions. The same can be said for airspeed where the rider could be riding in gustier conditions then the training set and as a result misclassification's could occur.

Following the analysis of the archival wind tunnel data provided by Body Rocket and following institutional ethical approval, two male cyclists were selected to participate in trials on an outdoor velodrome (Preston Park, Brighton, Sussex, UK), where they maintained the four positions: Baseline, Head-up, Head-down, and hands over the aero extensions. Both cyclists completed four laps in each position using a bicycle outfitted with the Body Rocket Ltd (Sussex, UK) system, which was adjusted to correspond with their personal preferences. As Preston Park velodrome has a clear line of sight during each run, the riders were monitored to ensure they were maintaining the target position.

Statistical Analysis

To determine the accuracy of the GMM, SVM and GBM, classification success was implemented and took the form:

$$CS = \left(1 - \frac{F_p + F_n}{N}\right) \times 100\% \tag{51}$$

where CS is the classification success, F_p is the number of false positives, F_n is the number of false negatives, and N is the total number of samples.

Post-processing involved the application of a 6-second modal moving average filter to mitigate noise within the raw prediction data. By definition, the modal moving average took the form:

$$\hat{p}_i = \text{mode}(p_{i-h}, \ldots, p_{i+h}). \tag{52}$$

For Equation (52) $\hat{p}_i$ is the filtered position data, p is the raw position data and $h = \lfloor w/2 \rfloor$ is half the window size w. To visually represent the models' accuracy, data was plotted as time versus position, enabling the categorisation of the four distinct positions.

Further Statistical Analysis for the Gradient Boosted Machine

Two additional distinct metrics were employed to rigorously evaluate the accuracy of the GBM, these were log loss as defined in Equation (49) and Mean Squared Error.

During the testing phase with blind data, normalisation was performed relative to the statistical properties of the training dataset. Additionally, for the GBM, the accuracy was determined on both the training portion of the data and the test portion of the dataset as the modal moving average averages across both the training and testing portions.

Given that the GBM is a tree-based ensemble model, the number of leaves within its constituent trees is a critical hyperparameter. An excessively high number of leaves can lead to model over-fitting, while an insufficient number of leaves may

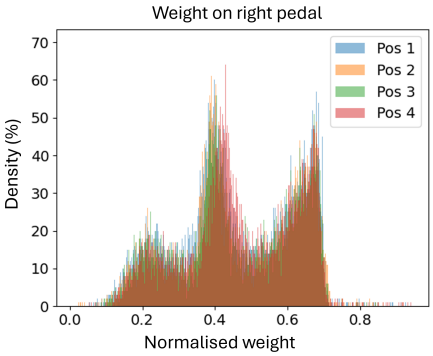
result in under-fitting. During model construction, early stopping rounds were implemented. This mechanism allows the model to continue building beyond a pre-specified number of trees. If the log loss on a validation set continues to decrease, the number of trees (and consequently, leaves) is increased. Conversely, if the log loss begins to increase, the model reverts to the configuration that yielded the lowest log loss, thereby preventing over-fitting.

4.4 Results

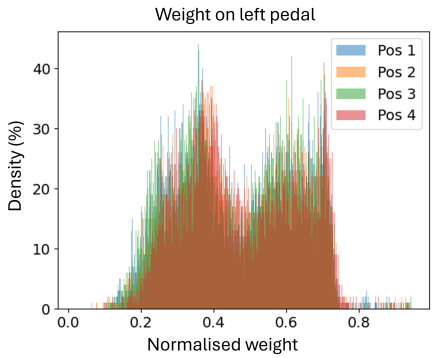
Before committing to training a GBM on all the data within this study, the simpler GMMs and SVMs were trained on one of the WT1 datasets. The motivation for this was to quickly evaluate whether it would be worthwhile to train the SVM or GMM on all of the data, since if they were to perform poorly on a single dataset, it is unlikely that they would perform well on subsequent datasets, especially when they are likely to be noisier for the OD datasets.

Normalised Histograms

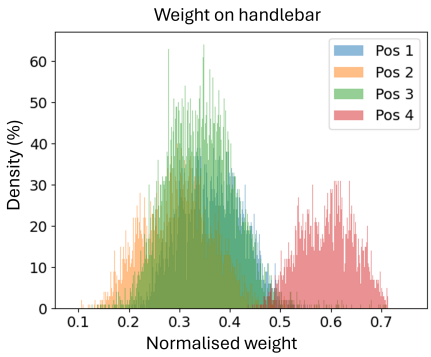
The construction of normalised histograms for various features facilitates the visualisation of class distributions corresponding to each feature. This approach enables the identification of features that effectively differentiate between classes, thereby enhancing their utility for classification purposes. normalisation ensures the comparison is based on probabilities rather than raw counts, thereby maintaining fairness regardless of variations in class sizes.



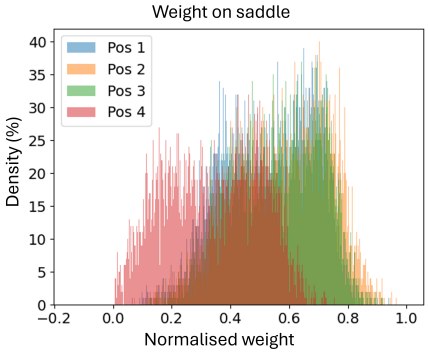
(a) Plot of weight on right pedal



(b) Plot of weight on left pedal



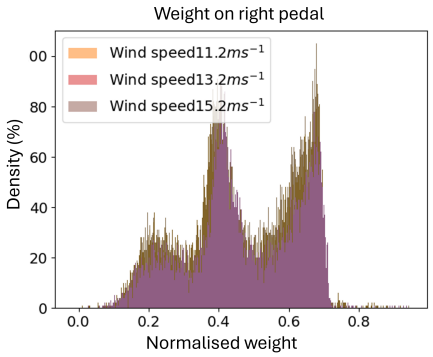
(c) Plot of weight on the handlebar



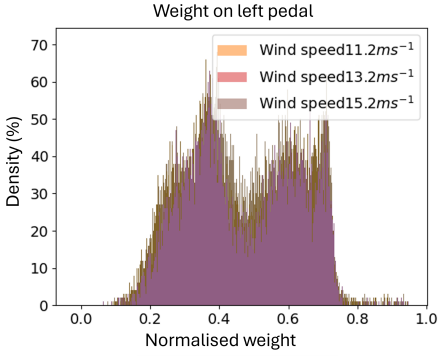
(d) Plot of weight on the saddle

Figure 4.8: Plots showing the weight distributions of the different positions on the contact points of the bike.

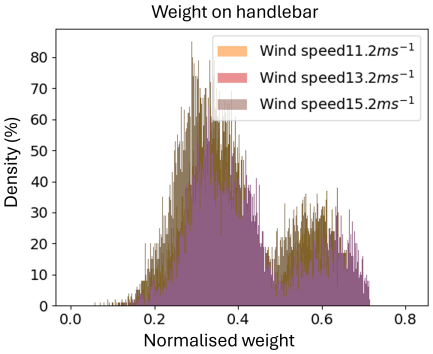
The plots in Figure 4.9 show the normalised weights of the rider on the contact points of the bike. In the figure legends, each position can be identified via red, green, orange and blue for positions 1, 2, 3, and 4, respectively. As can be seen, the weight on the pedals is similar across all positions, and there are some clear differences in the weight distribution when comparing to the handlebar and the saddle.



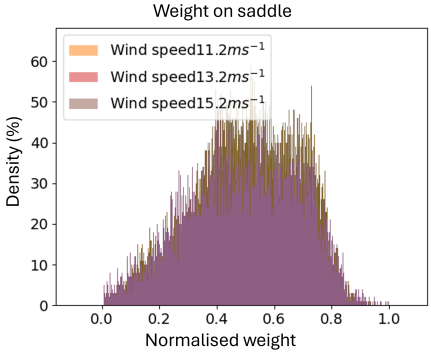
(a) Plot of weight on right pedal



(b) Plot of weight on left pedal



(c) Plot of weight on the handlebar



(d) Plot of weight on the saddle

Figure 4.9: Plots showing the weight distributions at the different wind speeds on the contact points of the bike.

Correlation Matrices

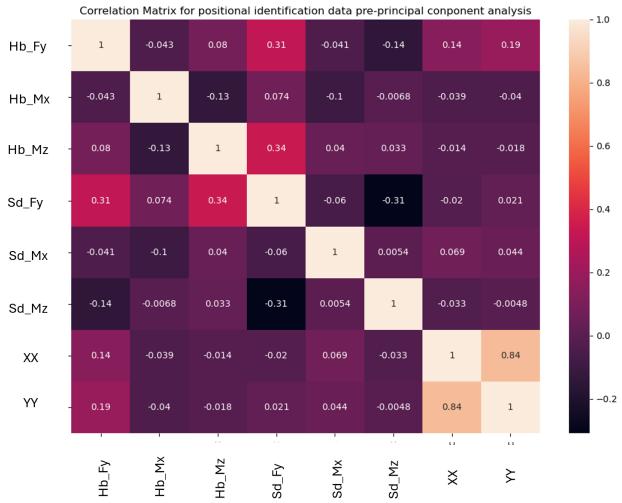


Figure 4.10: Correlation matrix showing the 8 selected features for model training before a PCA is applied.

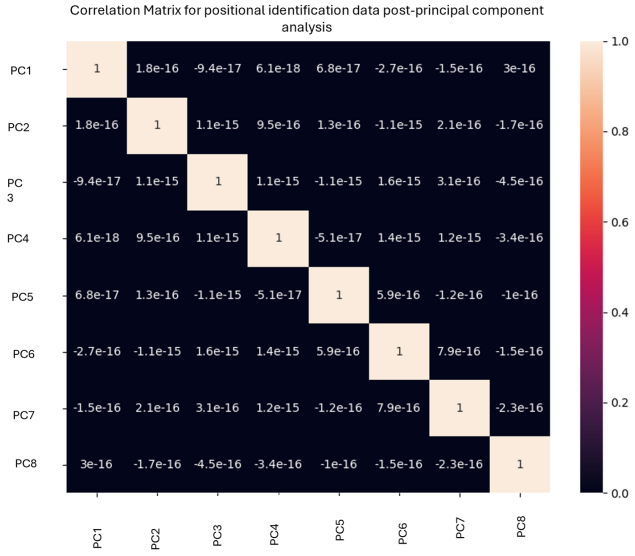


Figure 4.11: Correlation matrix showing the 8 selected features for model training after a PCA has been applied.

Figures 4.10 and 4.11 shows the correlation matrix on one of the wind tunnel datasets pre and post PCA.

Gaussian Mixture Model

Following the outlined method of 3-sigma clipping and normalisation, all four covariance types were tested with a GMM. The results are shown in Table 5.

GMM Type	Train Accuracy (%)	Test Accuracy (%)
Spherical	30.64	8.56
Diag	24.82	42.99
Tied	21.95	45.37
Full	23.21	32.67

Table 5: Train and test accuracy for the four different GMM covariance types investigated in this study (spherical, diagonal, tied and full) on post-normalisation and PCA data.

Because the results from the post-PCA data were so low, the GMM was also trained and tested on data without PCA or data normalisation.

On unnormalised data from the University of Southampton wind tunnel without a PCA, the training accuracy for the spherical was 25.2% with a test accuracy of 25.0%, for diagonal it was 25.4% with a test accuracy of 25.9%. Tied was 76.8% with a test accuracy of 76.6%, and full had a training accuracy of 26.5% with a test accuracy of 25.6%

GMM Type	Training Accuracy (%)	Test Accuracy (%)
Spherical	25.2	25.0
Diagonal	25.4	25.9
Tied	76.8	76.6
Full	26.5	25.6

Table 6: Train and test accuracy for the four different GMM covariance types investigated in this study (spherical, diagonal, tied and full) on un-normalised data without a PCA.

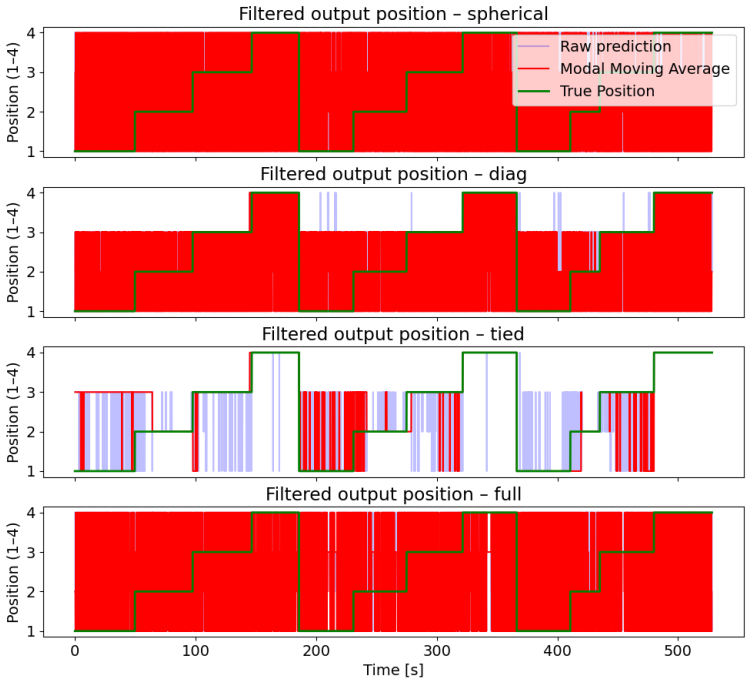


Figure 4.12: Four plots of the Body Rocket positional data with the four different GMM covariance types investigated in this study (spherical, diagonal, tied and full). Plotted as position across time with a 6-second modal moving average applied to the prediction data (red trace), the raw prediction data (blue trace) and the ground truth (green trace).

Figure 4.12 shows the true position (green trace), raw prediction (blue trace) and the prediction with a 6-second modal moving average applied (red trace).

GMM Type	Train Accuracy (%)	Test Accuracy (%)
Spherical	27.13	22.82
Diag	22.43	43.50
Tied	76.75	89.54
Full	22.97	37.86

Table 7: Train and test accuracy after modal moving average filtering.

Table 7 shows the table of accuracy for the different GMM types.

Support Vector Machine

Figure 4.13 shows the four different types of SVM initially tested in this study to qualify which model to use for the remainder of the investigation. The models tested in this portion of the study were the Linear SVC, the SVC with a Radial Basis Function and a polynomial SVC to the 3rd degree.

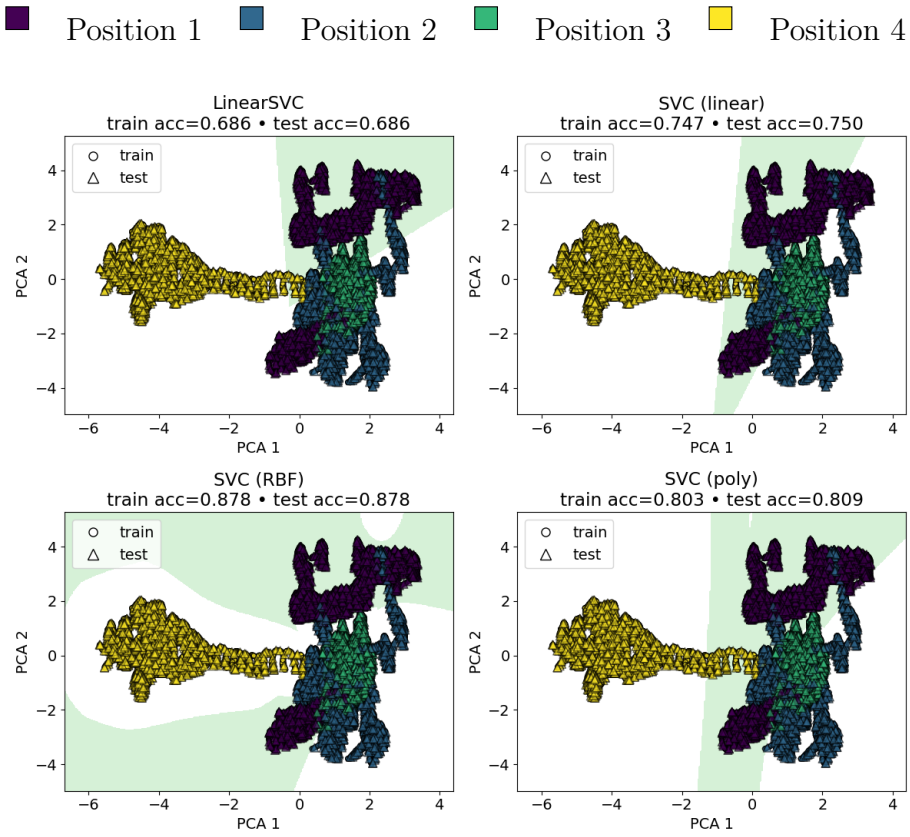


Figure 4.13: A plot showing the four vector plane outputs from the Support Vector Machine, with the pale green regions indicating the decision boundaries. The triangles represent the data used for SVM training, and the circles represent the data used for testing. Purple represents position 1, blue represents position 2, green represents position 3 and yellow represents position 4.

Table 8 shows the raw results from the Support Vector Machine.

SVM Type	Train Accuracy (%)	Test Accuracy (%)
LinearSVC	68.64	68.61
SVC (linear)	74.66	75.03
SVC (RBF)	87.84	87.78
SVC (poly)	80.33	80.89

Table 8: Table showing the training and test scores for the four different SVM kernel types investigated in this study (LinearSVC, SVC (linear), SVC (RBF) and SVC (poly)).

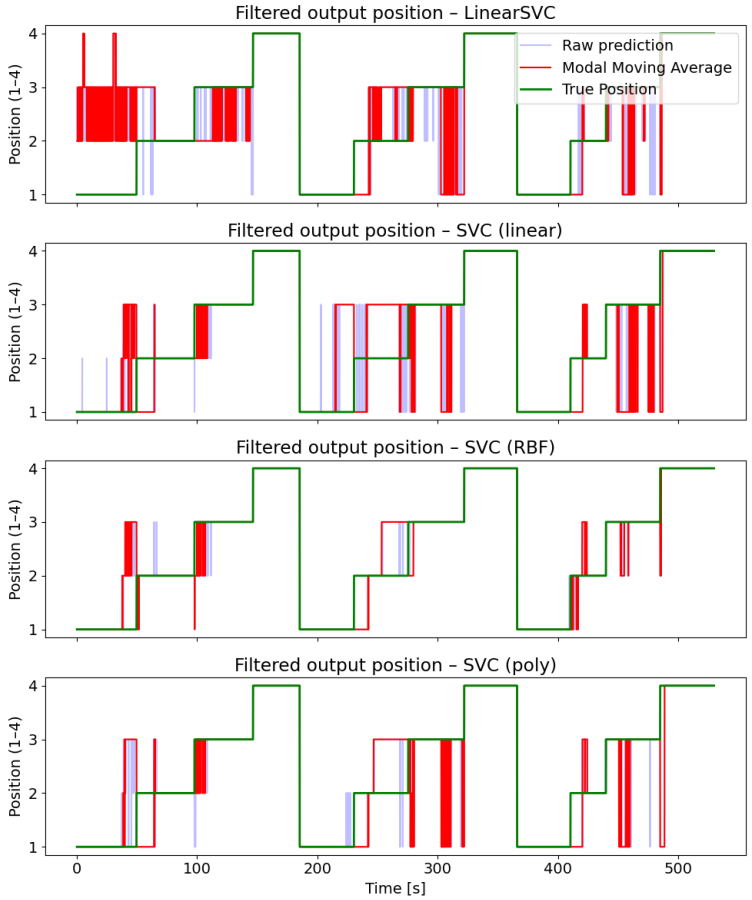


Figure 4.14: Four plots of the Body Rocket positional data with the four different SVM kernel types investigated in this study (LinearSVC, SVC (linear), SVC (RBF) and SVC (poly)). Plotted as position across time with a 6-second modal moving average applied to the prediction data (red trace), the raw prediction data (blue trace) and the ground truth (green trace).

Table 9 shows the raw results from the Support Vector Machine with the temporal

modal moving average.

SVM Type	Train Accuracy (%)	Test Accuracy (%)
LinearSVC	63.48	92.11
SVC (linear)	72.38	83.75
SVC (RBF)	85.95	96.00
SVC (poly)	78.40	89.60

Table 9: the four different SVM kernel types investigated in this study (LinearSVC, SVC (linear), SVC (RBF) and SVC (poly)) with a 6-second modal moving average applied.

Gradient Boosted Machine

The following results are presented in graphs alongside an overview of all the results in Figure 4.19.

Wind Tunnel Data

The GBM constructed for WT1 used 75 trees and 15 stopping rounds. The model exhibited a low error, evidenced by a log loss of 0.00035PLLU and a mean squared error of 0.00129PMSE. When a modal average filter was applied to the positional data over time, the classification accuracy reached 100% as can be seen in Figure 4.15 as the modal moving average (the red trace) is exactly the same as the true position (the green trace).

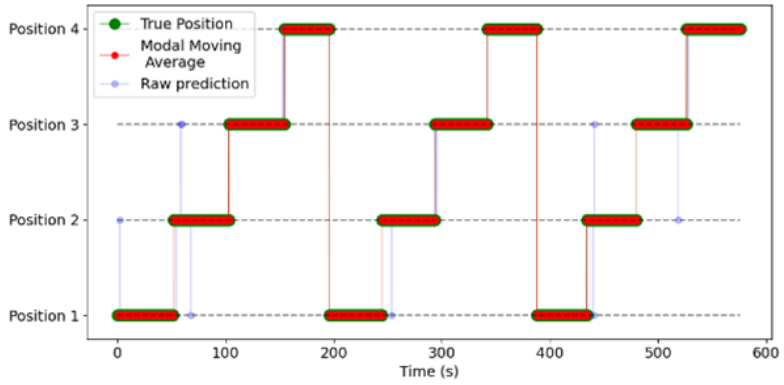


Figure 4.15: Rider position in WT1 during the test, with the rider transitioning between positions 1 and 4 on three occasions. The ground truth position is shown in green, the raw output from the trained Gradient Boosted Machine in blue, and the GBM output after applying a 6-second moving modal average is shown in red.

The WT2 dataset resulted in an 85-leaf (GBM) model that achieved a log loss of 0.00080PLLU and a mean-squared error (MSE) of 0.00022PMSE. This performance yielded a 100% classification accuracy when the data was filtered by time (see Figure 4.16). Notably, the rider in WT2 transitioned from position 1 to position 4 on two occasions, but over a significantly longer duration (2000 seconds) compared to WT1 (600 seconds).

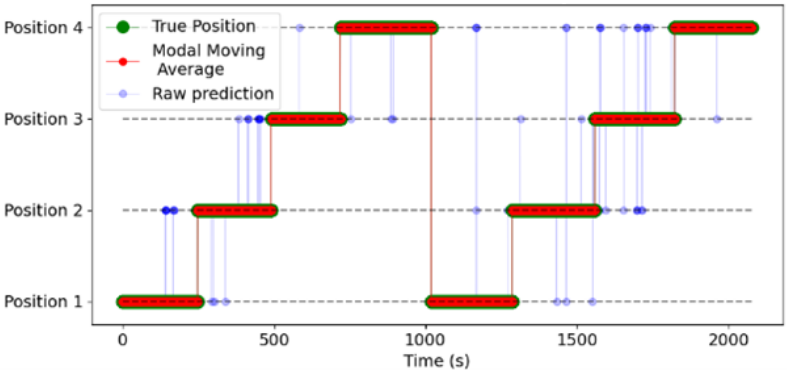


Figure 4.16: Rider position in WT2 during the test, where the rider transitions between positions 1 and 4 on two occasions. The ground truth position is shown in green, the raw output from the trained Gradient Boosted Machine in blue, and the GBM output after applying a 6-second moving modal average is shown in red.

Blind Data

A blind dataset was generated from both WT1 and WT2. Using the data gathered from WT1, the model achieved a log loss of 4.29312 PLLU, a corresponding MSE of 0.53055 PMSE, and a classification accuracy of 45% on the blind dataset. Conversely, with the WT2 dataset, the model demonstrated a log loss of 6.08763 PLLU, an MSE of 0.68907 PMSE, and a classification accuracy of 29%.

Outdoor Data

A 100 leaf model was generated for Rider 1, yielding a log loss of 0.01498PLLU, an MSE of 0.00377PMSE, and a classification success rate of 99.63%. The model's performance is visually represented in Figure 4.17. While Figure 4.17 shows some misclassifications, applying the modal moving average results in an accuracy of 100%.

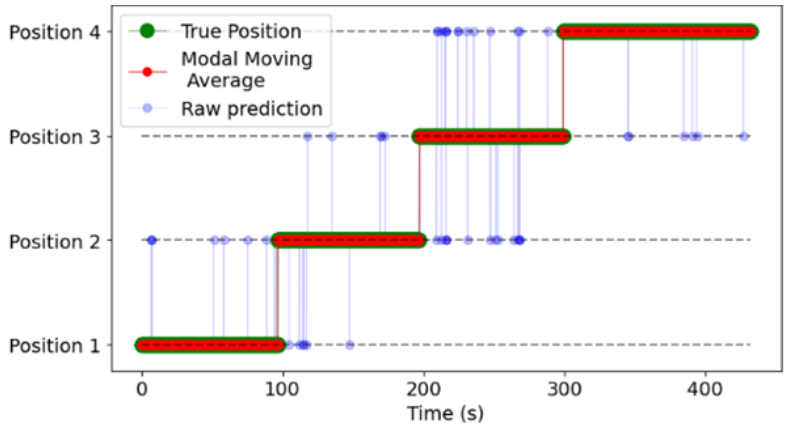


Figure 4.17: Prediction data from the Gradient Boosted Machine trained on data collected from Rider 1 during OD riding, where the rider transitions between positions 1 and 4 over a ≈ 450-second period. The true position is shown in green, the raw GBM output in blue, and the GBM output after applying a 6-second moving modal average is shown in red.

For Rider 2, a 100 leaf model was generated, yielding a log loss of 0.02163PLLU

and a corresponding MSE of 0.00606PMSE, with a classification success rate of 99.33% (see Figure 4.19). The model's performance is visually represented in Figure 4.18. Consistent with the outcomes for Rider 1, the results here indicate a 100% accuracy once the modal moving average is applied.

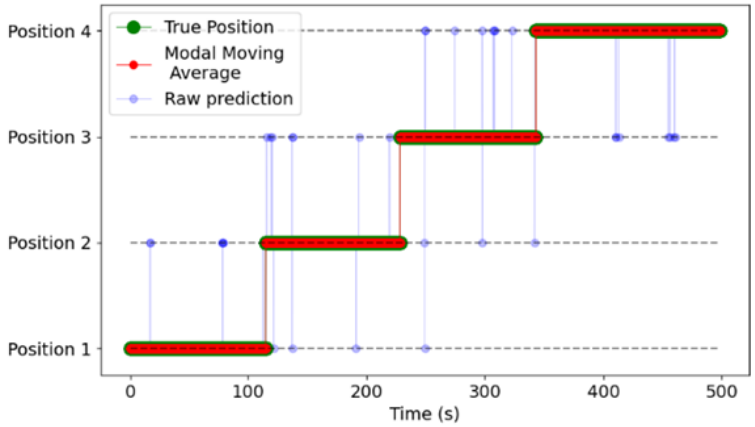


Figure 4.18: Prediction data from a Gradient Boosted Machine trained on data collected from Rider 1 during OD riding, where the rider moves between positions 1 and 4 over a $\approx$ 500 second interval. The true position is shown in green, the raw GBM output in blue, and the GBM output after applying a 6-second moving modal average is shown in red.

Model and dataset	log loss (PLLU)	MSE (PMSE)	CS (%)	Number of trees
WT1 (Train & Test)	0.000350	0.001290	99.96	75
WT1 (Blind)	4.293120	0.530550	45.03	75
WT2 (Train & Test)	0.000800	0.000220	99.97	85
WT2 (Blind)	6.087630	0.689070	28.48	85
OD– Rider 1	0.014980	0.003770	99.63	100
OD – Rider 2	0.021630	0.006060	99.33	100

Figure 4.19: Table of performance metrics (log loss, mean squared error, classification success and number of trees) from the models produced in WT1 and WT2 and from the OD velodrome data

4.5 Discussion

In Figure 4.9, the normalised weight on the pedals, handlebar and saddle is plotted from the data collected in WT1. As can be seen from the plots, the pedals do not appear to provide good class separation, whereas the handlebar and saddle do. This is a somewhat expected result. There is not much evidence to suggest that changing the way in which you position on the bike will change the amount of weight you put on the pedals. Studies such as (Bini et al. 2010) have researched the impact of pedalling cadence and its effects on joint mechanical work, where pedal force effectiveness was used. In their study, they found differences in pedal effectiveness between cyclists and triathletes with different seating locations. This finding is novel, as there have been studies where the angle of peak force application does change when the rider is in a different position. Interestingly, in Figure 4.9(a) and in Figure 4.9(b), it can be seen that there are some distinct peaks in the histograms. To understand the reason for the peaks, the weight distribution data was plotted at different wind speeds, as can be seen in Figure 4.8. From the results in Figure 4.8 it can be seen that wind speed does not contribute towards these distinct peaks. Additionally, Figure 4.9(a) and Figure 4.9(b) show different weight distribution characteristics. Figure 4.9 (a) shows three distinct peaks whereas Figure 4.9 (b) shows roughly two. Potentially, this could be due to a pelvic imbalance at the saddle, and inappropriate saddle height or a potential injury this rider has. Potentially, any pelvic imbalances could have been determined through the use of a motion capture system and a professional bike fitter or biomechanist to look over the athlete. This would not be a necessity for this study, as the positional classification should be independent of a ride's bike fit; however, it does show that the Body Rocket system could also provide insight into bike fit and rider kinematics. In Figure 4.9 (c) and Figure 4.9 (d), it is clear that there are distinct peaks for each position. Position 4 is clearly separate from the

other distributions in Figure 4.9 (c), potentially due to how upright the position is, leading to more force on the saddle and less on the handlebar. Unsurprisingly, for both Figure 4.9 (c) and Figure 4.9 (d), position 2 is rather distinct, which can be seen in Figure 4.7; however, positions 2 and 3 are very similar, leading to the results where the weight distribution on the saddle and handlebar is also similar.

Model Selection

From Figures 4.12 and 4.13, the results from the GMM and the SVM can be seen. From Figure 4.12 plots are shown for the classification via the spherical, diagonal, tied and full methods when trained and tested on data from WT1. The Spherical, diagonal and full covariance methods both show low classification accuracies from 25.0%-25.9%. An explanation for this poor performance could be broken down into what the different types of GMM covariance assume. A spherical coverance assumes equal variance with no feature correlations. Diagonal covariance also depends on equal variance, which may determine this result. Interestingly, the poor performance of the full covariance could be due to a lack of data, and the data may not be in a traditional Gaussian distribution form. Conversely, the tied method shows good classification accuracy of 76.8% on the training set and 76.6% on the testing set. This may be due to the tied covariance method having more data per covariance, which allows for the same covariance matrix to be shared. This is good accuracy; however, it could be improved. In Figure 4.12, the performance of the GMM can be seen across time, with the accuracy of the post modal moving average can be seen in Table 7. In line with the other results obtained, the modal moving average showed a negligible performance increase for the spherical, diagonal and full covariance types; however, for the tied covariance, the predictive performance increased for the testing portion of the wind tunnel dataset. In Figure 4.13 a visualisation of the SVM results can be seen. From the results presented

in Figure 4.13, it is clear that there is a distinct cluster representing position 4. This distinct cluster representing position 4 is likely due to the upright nature of this position. From Table 8 it is also clear that the predictive performance of the different SVM types is significantly greater than the results obtained from the GMM. With the MMA implemented, the performance of all SVMs is significantly increased, as can be seen in Table 9. The best performing SVM type was the RBF with a classification accuracy of 96% with the modal moving average applied. Following the results obtained, the Gradient Boosted Machine was investigated to see whether, with a little more complexity, the predictive performance of a classifier on the Body Rocket data could be increased.

The analysis of results from WT1 and WT2 demonstrates the capacity of the Gradient Boosted Machine to accurately distinguish the discrete rider positions investigated in this study (baseline, head-up, head-down and hands on the aero pads) as seen in Figure 4.7. The outdoor positions in this study, in particular, are representative of positions, as the hands on the aeropads position was replaced with hands over the aero extensions. Upon evaluation, the models derived from wind tunnel data exhibit comparable accuracies: WT1 achieves a classification success rate of 99.957%, while WT2 exhibits a rate of 99.969%. The application of a modal moving average to this data yields further enhancements in predictions, as depicted in Figure 4.15 and Figure 4.16, where the output prediction achieves a 100% classification success in identifying rider position. The result of the modal moving average improving predictions is expected, as the positions in this study were assumed for fixed periods of time and were not randomly chosen. As a result, it was a fair assumption to assume that the rider would be in one position or another, so a modal moving average was an appropriate choice. Both models maintain analogous levels of complexity, with WT1 being constructed with 75 trees and WT2 with 85 trees.

Notably, this investigation reveals considerable intra-dataset variability. Of particular interest is the finding that the model developed from WT1 demonstrates superior overall performance Figure 4.19. This discrepancy in results may be attributed to the design of the wind tunnel; specifically, WT1 utilised a rigidly fixed platform, whereas WT2 permitted some movement during pedalling. Furthermore, the dataset from WT2 was collected prior to the subsequent developments of the Body Rocket system, potentially influencing the variability in results.

Irrespective of the WT dataset employed, it is probable that the models exhibit overfitting or other latent variables in the data. To address this challenge, an increase in data is necessary, coupled with the prospect of further pre-processing and additional feature selection prior to model training. This should involve the careful selection of highly correlated force and moment data while excluding less relevant data. Moreover, feature engineering (Nargesian et al. 2017), combining features that show a high correlation with the output to generate a novel set of features, may further enhance model performance.

In this investigation, the outdoor data presents compelling evidence, with reliability demonstrated between the two riders in the classification of position, achieving 100% accuracy following the application of a modal moving average. Prior to this application, the model appears noisier than those derived from the wind tunnel, most likely due to environmental factors associated with outdoor cycling. As illustrated in Figures 4.17 and 4.18, data from Rider 1 displays greater stability with fewer misclassifications of position compared to Rider 2. This observation is evident in the raw outputs (depicted as blue traces) in the plots, and is further substantiated by Figure 4.17, which indicates Rider 1's classification success at 99.63% compared to Rider 2's 99.32%.

4.6 Conclusions

Within this study, it can be concluded that certain Machine Learning methods have the potential to be used in conjunction with Body Rocket data to predict the position of a rider on a bike to a high degree of accuracy, following the methodology in this study. As a result, this increases the potential for riders to be able to effectively and independently train without the need to visit wind tunnels and the need to use IMUs and other bespoke pieces of equipment to monitor their position on the bike. As the Body Rocket system is designed to work live on the bike, this also provides the potential for riders to accurately know and track their position on the bike both live and after a ride. Although this study predominantly addresses four discrete positions, future research could explore the determination of a continuous spectrum of positions that may delineate aspects such as head, torso, and hip angles. In a wind tunnel setting, less complicated GBMs achieve an output of 100% classification accuracy when a modal moving average is applied. However, application of these models to a blind dataset results in lower accuracy. The outcomes from the outdoor testing indicate 100% classification accuracy after the application of a modal moving average, though these models are more complex, which is likely attributed to the inherent variability in outdoor conditions.

Future advancements may integrate pose estimation computer vision to track anatomical landmarks in real-time using cameras. This continuous data on joint angles and body orientations, combined with force and moment insights from the Body Rocket system, can enhance understanding of rider aerodynamics and optimise a continuous range of riding positions. This approach offers a detailed, dynamic assessment of a rider's aerodynamic profile, surpassing discrete classifications.

Key point summary

- Four different covariance types of the Gaussian Mixture Model were applied to data collected from the Wind Tunnel of a rider in 4 discrete positions.
- Four different types of Support Vector Machine were applied to the Body Rocket data to predict the position of a rider on the bike in the wind tunnel.
- The Gradient Boosted Machine was successfully applied to the wind tunnel data, leading to a 100% classification accuracy.
- The Gradient Boosted Machine was applied to the Outdoor data, also leading to 100% classification accuracy once a modal moving average was applied.

Chapter 5

Pose Estimation in Cycling

5.1 Introduction

The previous chapter established that the Body Rocket system has the potential to predict a rider's position on the bike. This is a useful tool as it could allow riders to train in multiple different positions, which are optimal for specific sections of race courses, rather than using one position for an entire course; however, this tool only works on discrete positions. This chapter investigates whether computer vision can be used to get a more continuous identification of a rider's position. This would act as an accessible tool allowing for anyone with a camera to get information on their fit rather than being limited by needing a motion capture studio or similar expensive equipment.

- To evaluate the accuracy of YOLOv8 for road bike and time trial positions.
- To determine whether fine-tuning can improve model accuracy.

During aerodynamic tests, it is essential that only one variable of the test is changed at a time. This is so that the changes that are being measured can easily

be mapped to a specific difference in the testing configuration. As a result, during any testing, it is key to ensure that the athlete is maintaining a consistent position whilst on the bike, as any changes in position will impact the measurements made and could lead to false conclusions. This has been achieved previously in the wind tunnel through the use of image processing techniques such as edge detection. Within outdoor testing, inertial measurement units and optical techniques have been used to ensure position can be maintained. As mentioned previously in Chapter 4, the Body Rocket system also shows promise in its ability to predict the position of a rider on the bike.

With advancements in computer vision and human pose estimation, there has been a spike in interest within the sports engineering fraternity. Some studies have been undertaken to evaluate their accuracy (Fukushima et al. 2024) and have shown potential in pose estimation when compared to a ground truth obtained by 3D motion capture system. Within running, specifically gait analysis, pose models such as Openpose (Cao et al. 2017) have been compared to inertial measurement unit data to determine the accuracy of these pose models (D'Antonio et al. 2021). The study concluded that these pose models may be accurate for more qualitative systems, but quantitatively, the performance of the pose models was not accurate enough to make meaningful conclusions from the collected data. Additional sports where pose estimation models have been applied include rowing (Szűcs and Tamás 2018), where it was found that the models and adaptations made to the models were to a level which surpassed the state of the art. Within football (Pinheiro et al. 2022), pose models have been used successfully for anticipating the movement of a player when taking a penalty.

The state of the art in determining joint locations on a person is through the use of a motion capture system. These systems can vary in design, some being based on inertial measurement units, which have found notable accuracy in both short and long duration applications (Obradović and Stančin 2023; Goreham, MacLean and

Ladouceur 2022). Other systems are based on the application of infrared cameras and retroreflective markers, which are generally considered the most accurate for 3D motion capture applications.

Within cycling specifically, there have been a number of studies looking into the applications of computer vision as a marker-less solution to study pedalling dynamics and posture. At all levels of cycling, from beginners to elite athletes, ensuring a proper bike fit is pivotal to ensure that you are not going to cause undue stress on your body, and at the elite level, it can be tuned for maximal power output and muscle engagement alongside aerodynamic efficiency. Some studies have investigated the application of computer vision with sensors on the pedals and tracking to tackle problems such as occlusion (Serrancoli et al. 2020). Other studies within cycling aerodynamics have investigated the principle of predicting aerodynamic efficiency through the use of computer vision pose models from Openpose (Garimella et al. 2020). In the sagittal plane, there has been recorded success in determining the accuracy of OpenCap (Uhlrich et al. 2023b), another open-source pose estimation model. The results from this study showed comparable results to the ground truth data and have comparable precision when measuring the 'hip (flexion/extension), knee (flexion/extension), and ankle (dorsiflexion/plantarflexion) joints.' (Kakavand et al. 2024). Throughout the literature, it is clear that there is a lot of research covering the application of Openpose within many areas of sport, including cycling; however, there is limited research determining the accuracy of the pose estimation of YOLO within cycling and pose models as a whole in tucked time trial positions.

5.2 Theory

Pose estimation models have been a developing area of research due to the clear applications in many fields such as gymnastics (Kitajima et al. 2023), bike fitting

(Zhang and Zhao 2024) and robotics (Rodrigues et al. 2022). Due to this, the early pose estimation models started with simple architectures and eventually developed into the models used today.



Figure 5.1: [REDACTED]

Previous Pose Modelling Techniques

The pictorial structure model (PSM) helped lay the groundwork for pose estimation within the field of computer vision, using spring-like constraints to determine the location of body joints (Belagiannis et al. 2014). These early models were an effective start in the field; however, their accuracy was greatly reduced when occlusions and more complex poses were tested. Around the same time, Deformable Part Models (DPMs) were introduced, which modelled each body part individually, but they also struggled with complex articulations. Both approaches relied heavily on handcrafted features, which limited their scalability and generalisation.

With the advent of deep learning, successive pose modelling techniques began to emerge. Most of these models were based on Convolutional Neural Networks (CNNs) and reformulated pose estimation as a regression problem (Zheng et al. 2023). This transition marked a paradigm shift, moving from rule-based, manually engineered representations to end-to-end learning pipelines. This led to models which started to have increasing accuracy; however, this increase in capability

came at the cost of the computing resources required. To address these limitations, real-time detection frameworks such as the YOLO family of models were introduced, offering both speed and accuracy, making them highly suitable for practical pose estimation systems.

You Only Look Once

You Only Look Once (YOLO) is a type of computer vision model introduced by Redmon et al. (2016). Its original purpose was to act as an object classifier where the model only looks at an image once, then has the ability to identify what objects are in the image. Unlike classifier-based approaches, YOLO is trained with the use of a loss function that corresponds to detection performance. The original purpose of the YOLO models was to perform object detection and localisation. Object detection is the process of identifying what an object is in an image (Liu et al. 2020). Object localisation, on the other hand, determines a bounding box of the detected image, therefore, providing an idea of the location of an object in an image.

In general, an object detection model can be described as a function:

$$\hat{y} = f_\theta(x) = \{(p_i, b_i)\}_{i=1}^{N} \tag{53}$$

where x is the input image, N is the number of detected objects, $p_i \in [0,1]^C$ represents the class probabilities for C object classes and $b_i = (x_i, y_i, w_i, h_i)$ represents the bounding box parameters (centre coordinates and width/height).

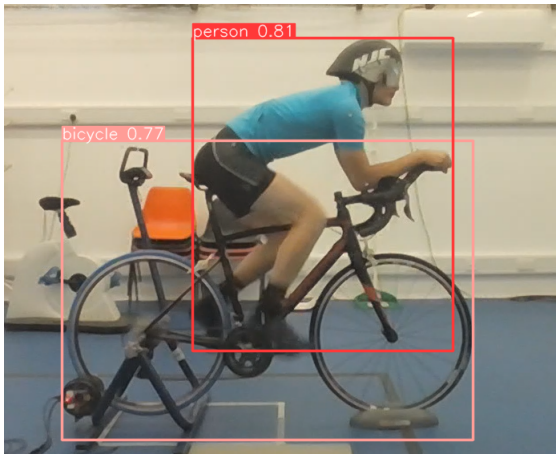


Figure 5.2: An example of a YOLO detection model detecting a bicycle (the pink bounding box) and a person (the red bounding box) in an image.

The unique part of a YOLO model is the fact that it breaks up an image into a grid cell. This allows for there to be any number of objects in the image. The benefits of YOLO are that it has low latency and has seen numerous applications in traffic monitoring and management (Flores-Calero et al. 2024), facial composition (Qi et al. 2022), and sports (Dedgaonkar et al. 2025), to name a few.

YOLO for Human Pose Estimation

With human pose estimation, an object detection and localisation is performed; however, there is another part to YOLO's processing — once a person is identified in a frame, their body joints need to be detected. Unlike models such as Openpose, pose estimation with YOLO does not use heatmaps. To determine the joint locations for a person is fairly straightforward, as it takes each joint to be a key-point, and much like object detection, it is taken to be a key-point detection problem.



Figure 5.3: [REDACTED]

As can be seen in Figure 5.3, the structure of YOLO can be broken down into three main chunks: the backbone, the neck and the prediction head. The backbone is responsible for extracting meaningful features from the input. The neck connects the backbone to the head, performing feature fusion, and the head generates the model's outputs, the bounding boxes and the pose estimation (Yaseen 2024). The backbone is constructed of convolutional layers and C2f's. C2f's are a new addition specifically for YOLOv8. C2f helps with enhancing how feature maps are processed, shared and refined between layers, essentially providing a better image for the YOLOv8 model to understand. The human pose models with YOLO use the Common Objects in Context (COCO) dataset (Lin et al. 2014) with the key-point format as follows (Ultralytics 2025):

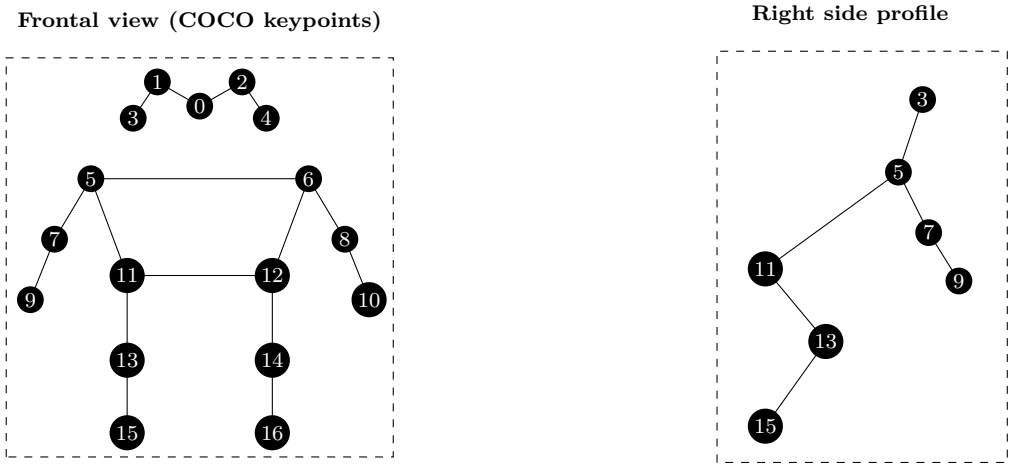


Figure 5.4: COCO human pose keypoints shown in frontal view and rider's right side profile as used in this study.

Training a YOLO Model

As with most model training, initially a model is given a ground truth dataset where patterns can be learnt, leading to accurate predictions on data inputted outside of the training set. To train YOLO, back-propagation is implemented. In this process, the internal weights are adjusted with respect to the training data. During training, to get an indication of the accuracy of the prediction,s Intersection over Union (IoU) is used alongside Mean Average Precision (mAP). IoU works by comparing the bounding boxes of the prediction and the ground truth (determined via manual labelling). Mathematically, it is defined by:

$$\text{IoU} = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \tag{54}$$

Specifically for the training of YOLOv8 Complete IoU (CIoU) is used:

$$\text{CIoU} = \text{IoU} - \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} - \alpha v \tag{55}$$

Where:

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2, \quad \alpha = \frac{v}{(1-\text{IoU}) + v} \tag{56}$$

Producing the loss function:

$$L_{\text{CIoU}} = 1 - \text{CIoU} \tag{57}$$

Visually it can be represented as in Figure 5.5

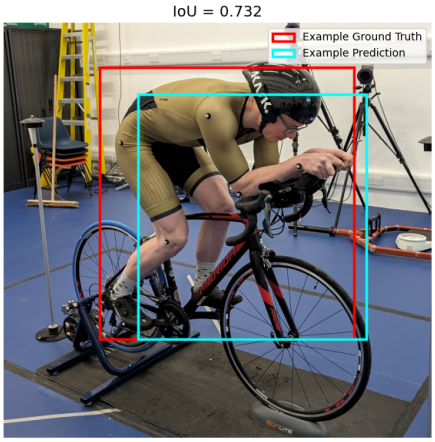


Figure 5.5: An example of IoU for YOLOv8

Mean Average Precision (mAP) is another metric used in YOLO model training. This accuracy metric uses the ground truth classification data and the prediction data across an entire data set to get a mean average accuracy.

Fine-Tuning YOLOv8

For out-of-the-box pose models such as YOLOv8, it can be preferable to fine-tune a model as opposed to training the model from scratch. This presents several advantages, and in the case of this study, was determined to be the best option. As discussed, training a model can be very computationally intensive, requiring substantial computing power to train on a large dataset. For tasks that fall far outside the intended use of the model or the data it was trained on, training a new model is usually the best option. In this case, however, the pose models are applied to the side profile of a rider, which is a configuration already present in typical pose-estimation training sets. As a result, the models can be fine-tuned effectively using task-specific datasets.

5.3 Materials and Methods

This study focuses on the position of a rider on a bike in the sagittal plane. To make it possible to determine the torso angle of an athlete, together with all other possible metrics that may be used. The landmarks used for the computer vision models were the right hip, right shoulder, right elbow, right ear, nose, right knee and right ankle.

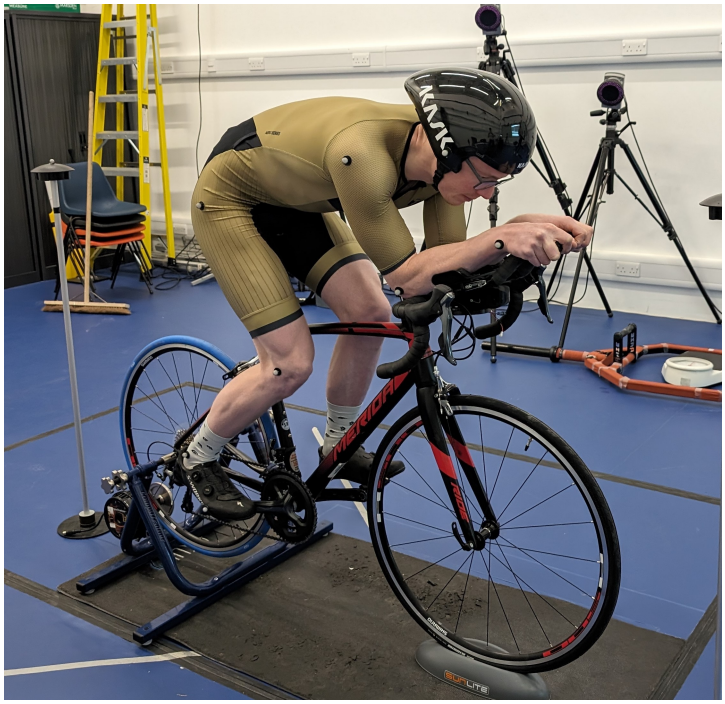


Figure 5.6: An image of the experimental setup with one of the research participants. In this image, the participant has markers placed in the same locations as those identified by the pose estimation and aligned with the COCO standard.

Through institutional ethical approval, a cohort of three athletes was recruited to collect data and produce 42 unique datasets across various configurations. These configurations included four discrete positions on a road bike and a time trial bike setup, as shown in Figures 5.8 and 5.9. Each position on each bike setup was tested with three different jerseys, three different helmets, and a variable background. The variable background in particular was a white background (the

background of the motion capture studio and banners which included images of people, to test the limits of the pose models pose identification). The purpose of this was to test the computer vision models not on inter-person differences but on configuration differences, including jerseys of a similar colour to the image background, banners in the background featuring people, and helmets that could create differences compared with the datasets on which the models were trained. The different helmets tested were road bike helmets and aero-TT helmets in black and white, again to evaluate them against the pose models.



Figure 5.7: The three different cycling helmets in this study, (left) the MET drone wide body (MET-Helmets 2025), (Middle) HJC Adwatt (HJC Helmets 2025) and (right) POC Omne Beacon (POC Sports 2025)

With the Qualisys motion capture system (Qualisys 2025), a laptop camera (HP Inc. 2025) was used to record the side profile of the athletes on the bikes. Once the data was collected, the motion capture data was saved as 3D spatial coordinates, and the video files were stored in the appropriate secure locations. During data acquisition, two separate human operators started the recordings of the motion capture system and the laptop video feed. This led to a difference in start time between the motion capture data and the video data, requiring a solution to synchronise the datasets. Additionally, to appropriately scale the data, the coordinates of the motion capture points had to be scaled to match the video feed. Both scaling and synchronisation were performed manually. The scaler offsets were determined by manually marking the upper body landmarks and using a least squares approach to determine the linear offset and scaling factor.

The coordinates of the landmarks can be defined in vector notation as follows:

$$\mathbf{P}_x = \begin{bmatrix} x_1^{\text{pose}} \\ x_2^{\text{pose}} \\ \vdots \\ x_5^{\text{pose}} \end{bmatrix}, \quad \mathbf{P}_y = \begin{bmatrix} z_1^{\text{pose}} \\ z_2^{\text{pose}} \\ \vdots \\ z_5^{\text{pose}} \end{bmatrix}, \quad \mathbf{G}_x = \begin{bmatrix} x_1^{\text{gt}} \\ x_2^{\text{gt}} \\ \vdots \\ x_5^{\text{gt}} \end{bmatrix}, \quad \mathbf{G}_y = \begin{bmatrix} y_1^{\text{gt}} \\ y_2^{\text{gt}} \\ \vdots \\ y_5^{\text{gt}} \end{bmatrix}$$

Assuming a linear relationship between the ground truth and the motion capture output:

$$x^{\text{gt}} = n_x \cdot x^{\text{pose}} + \text{offset}_x$$
$$y^{\text{gt}} = n_y \cdot z^{\text{pose}} + \text{offset}_y$$

This can be written in matrix form as:

$$\mathbf{G}_x = A_x \cdot \begin{bmatrix} n_x \\ \text{offset}_x \end{bmatrix}, \quad A_x = \begin{bmatrix} x_1^{\text{pose}} & 1 \\ x_2^{\text{pose}} & 1 \\ \vdots & \vdots \\ x_5^{\text{pose}} & 1 \end{bmatrix}, \quad \mathbf{G}_y = A_y \cdot \begin{bmatrix} n_y \\ \text{offset}_y \end{bmatrix}, \quad A_y = \begin{bmatrix} z_1^{\text{pose}} & 1 \\ z_2^{\text{pose}} & 1 \\ \vdots & \vdots \\ z_5^{\text{pose}} & 1 \end{bmatrix}$$

Both the x and y scaling factors can be obtained by the following:

$$\begin{bmatrix} n_x \\ \text{offset}_x \end{bmatrix} = \arg\min_{\mathbf{w}} \|A_x \mathbf{w} - \mathbf{G}_x\|^2, \quad \begin{bmatrix} n_y \\ \text{offset}_y \end{bmatrix} = \arg\min_{\mathbf{w}} \|A_y \mathbf{w} - \mathbf{G}_y\|^2$$

This provides the linear offset and scaling factor.

$$x^{\text{gt}} = n_x \cdot x^{\text{pose}} + \text{offset}_x$$

$$y^{\text{gt}} = n_y \cdot z^{\text{pose}} + \text{offset}_y$$

Within this investigation, three models were tested: YOLOv8, YOLOv8 fine-tuned over 5 epochs, and YOLOv8 fine-tuned over 10 epochs.



(a) Road bike Position 1 (b) Road bike Position 2 (c) Road bike Position 3 (d) Road bike Position 4

Figure 5.8: The four road bike positions undertaken in this study



(a) Time trial Position 1 (b) Time trial Position 2 (c) Time trial Position 3 (d) Time trial Position 4

Figure 5.9: The four time trial positions undertaken in this study

Whilst analysing the data, it was noticed that the downstroke of the pedal was identified more easily than the upstroke. This may be due to the composition of the background, making it harder to segment the ankle and knee. One potential source of error was determining when to start the motion capture and the laptop video feed.

Error analysis

To determine the accuracy of the pose estimation model, three error metrics were employed: Object Key-point Similarity (OKS), Mean per Joint Position Error (MPJPE) and Percentage of Correct Keypoints (PCK).

$$\text{OKS} = \frac{\sum_{i=1}^{N} \exp\left(\frac{-d_i^2}{2S^2\sigma_i^2}\right) \delta(v_i > 0)}{\sum_{i=1}^{N} \delta(v_i > 0)} \tag{58}$$

Alongside the Object Keypoint Similarity, the Mean Squared Error can be used to determine the accuracy of the model in comparison with the ground truth, as presented in studies such as (Maji et al. 2022). The MPJPE:

$$\text{MPJPE} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{J}_i^{\text{pred}} - \mathbf{J}_i^{\text{gt}} \right\|_2 \tag{59}$$

In Equation (59) N is the number of samples, $\mathbf{J}_i^{\text{pred}}$ is the predicted position $\mathbf{J}_i^{\text{gt}}$ is the ground truth and $||.||_2$ is the Euclidean norm. This has seen applications in a number of studies (No and Lee 2025; Zhang et al. 2020; Yang et al. 2024).

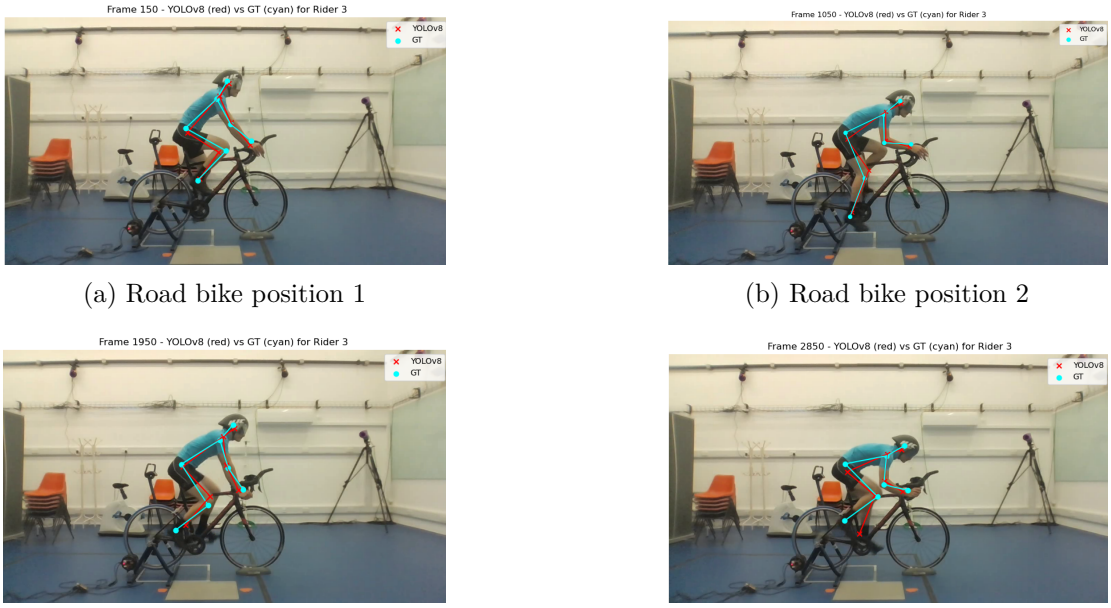
And the Percentage of Correct Keypoints

$$\text{PCK}(\alpha) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(\left\| \mathbf{J}_i^{\text{pred}} - \mathbf{J}_i^{\text{gt}} \right\|_2 < \alpha \cdot s \right) \tag{60}$$

5.4 Results

Performance of YOLOv8

Figure 5.10 shows an example of a participant assuming one of the four road bike positions, with the ground truth output (cyan trace) and the unfine-tuned YOLOv8 output (red trace).



(a) Road bike position 1

(b) Road bike position 2

(c) Road bike position 3

(d) Road bike position 4

Figure 5.10: The unfine-tuned YOLOv8 outputs for road bike positions 1 to 4 (red trace) overlaid with the ground truth (cyan trace).

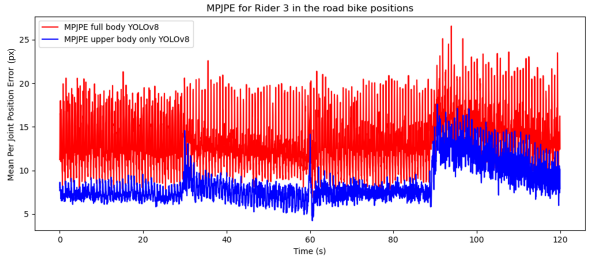
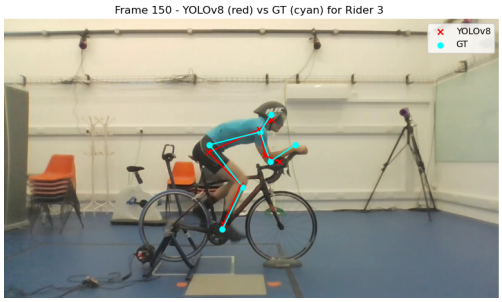


Figure 5.11: Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the unfine-tuned YOLOv8 model.

Figure 5.11 shows the MPJPE across time for Rider 3 in the four road bike positions. The red trace shows the full-body MPJPE and the blue trace shows the MPJPE in the upper body only.

Time Trial Positions

Figure 5.12 shows an example of a participant assuming one of the four time trial positions, with the ground truth output (cyan trace) and the unfine-tuned YOLOv8 output (red trace).



(a) Time trial position 1



(b) Time trial position 2



(c) Time trial position 3



(d) Time trial position 4

Figure 5.12: The unfine-tuned YOLOv8 outputs for time trial positions 1 to 4 (red trace) overlaid with the ground truth (cyan trace).

Figure 5.13 shows the MPJPE across time for Rider 3 in the four time trial positions. The red trace shows the full-body MPJPE, and the blue trace shows the MPJPE in the upper body only.

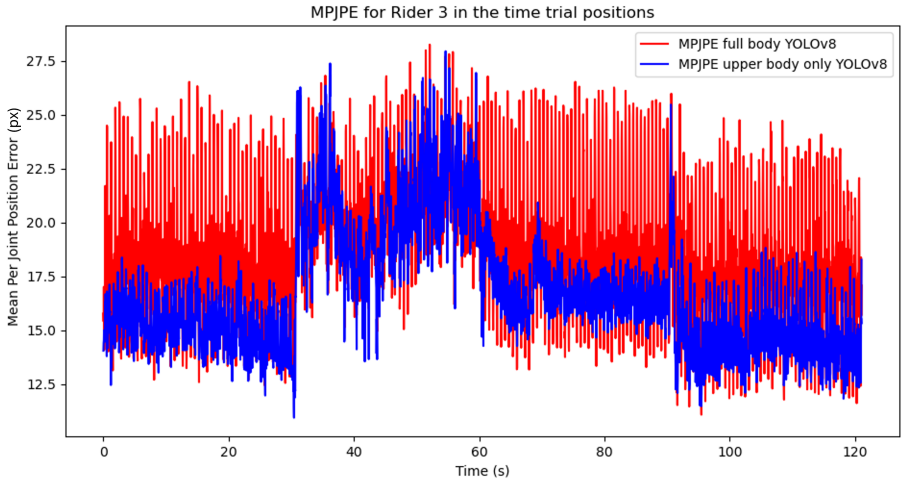


Figure 5.13: Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the unfine-tuned YOLOv8 model in the time trial position sequence.

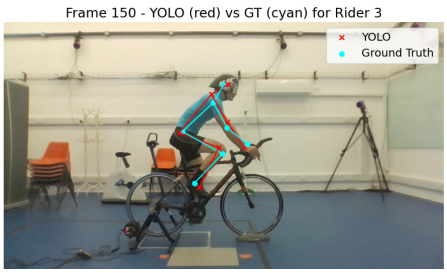
Table 10 shows the errors obtained for YOLOv8 across all positions and set-ups.

Error metric	Value
full_MPJPE_mean (px)	16.46
full_PCK_mean	0.99
full_OKS_mean	0.96
full_MSE_mean (px^2)	790.84
upper_MPJPE_mean (px)	12.92
upper_PCK_mean	1.00
upper_OKS_mean	0.98
upper_MSE_mean (px^2)	905.11
elbow_angle_error_mean (°)	10.84
shoulder_angle_error_mean (°)	8.15
hip_angle_error_mean (°)	6.23
knee_angle_error_mean (°)	12.04

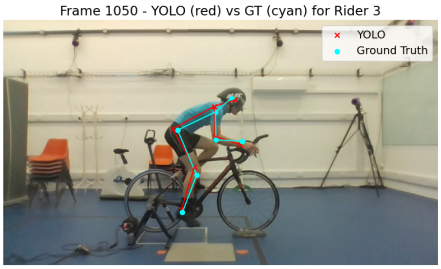
Table 10: Average estimation errors for the out-of-the-box unfine-tuned YOLOv8 pose estimation model.

Fine-tuned YOLOv8 Pose Model over 5-epochs

Figure 5.14 shows an example of a participant assuming one of the four road bike positions, with the ground truth output (cyan trace) and the YOLOv8 output fine-tuned over 5-epochs (red trace).



(a) Road bike position 1



(b) Road bike position 2



(c) Road bike position 3



(d) Road bike position 4

Figure 5.14: Rider 3 in the four road bike positions with the ground truth (cyan trace) and the YOLOv8 output fine-tuned over 5-epochs (red trace).

Figure 5.15 shows the MPJPE across time for Rider 3 in the 4 time trial bike positions for one test. The red trace shows the full body MPJPE, and the blue trace shows the error in the upper body only.

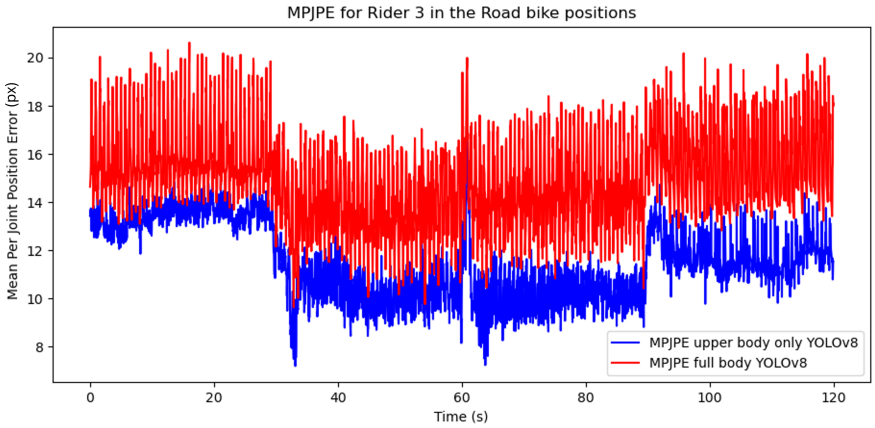
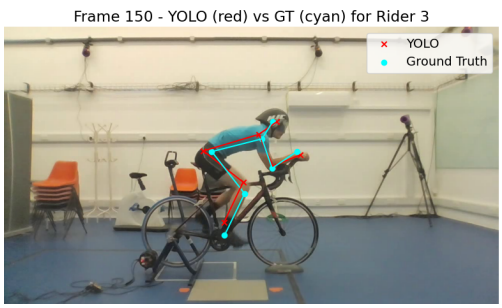


Figure 5.15: Plot showing the Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the fine-tuned YOLOv8 model

Time Trial Positions

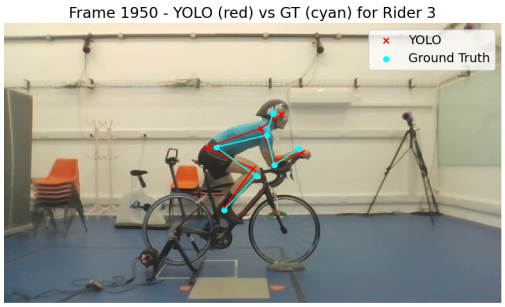
Figure 5.16 shows an example of one of the participants assuming one of the four road bike positions alongside the ground truth output (red trace) and the YOLOv8 output fine-tuned over 5-epochs (cyan).



(a) Road bike Position 1



(b) Road bike Position 2



(c) Road bike Position 3



(d) Road bike Position 4

Figure 5.16: The 5-epoch fine-tuned YOLOv8 outputs for the time trial positions 1 through to 4 (the red trace) overlaid with the ground truth (cyan trace)

Figure 5.17 shows the MPJPE across time for Rider 3 in the 4 time trial bike positions for one test. The red trace shows the full body MPJPE and the blue trace shows the error in the upper body only.

Error title	Error Value
full_MPJPE_mean (px)	14.97
full_PCK_mean	1.00
full_OKS_mean	0.97
full_MSE_mean (px^2)	287.14
upper_MPJPE_mean (px)	10.93
upper_PCK_mean	1.00
upper_OKS_mean	0.98
upper_MSE_mean (px^2)	146.31
elbow_angle_error_mean (°)	16.71
shoulder_angle_error_mean (°)	3.40
hip_angle_error_mean (°)	7.51
knee_angle_error_mean (°)	6.19

Table 11: Table showing the average estimation errors for the fine-tuned YOLOv8 pose estimation model over 5-epochs.

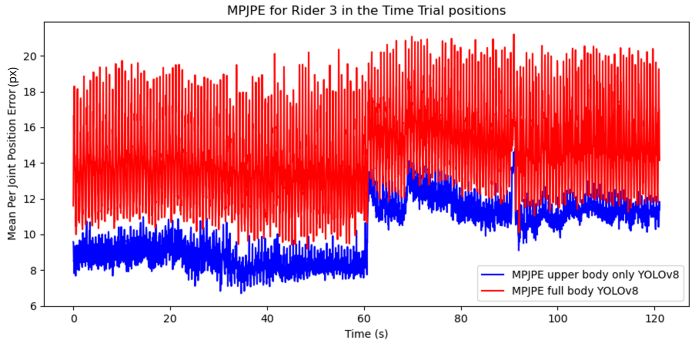
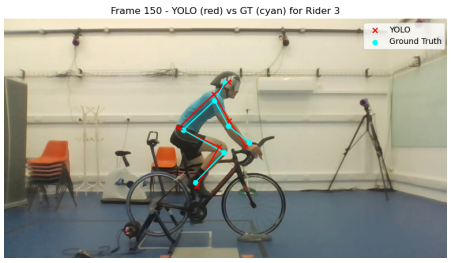


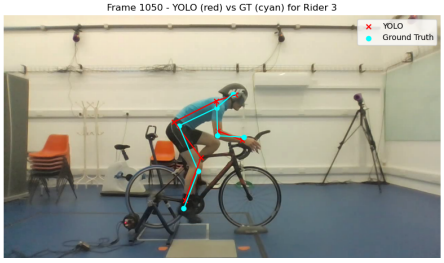
Figure 5.17: Plot showing the Mean Per joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the fine-tuned YOLOv8 model

Fine-tuned YOLOv8 Pose Model over 10-epochs

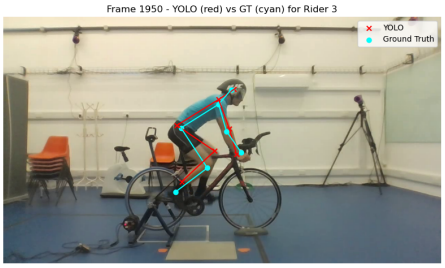
Figure 5.18 shows an example of one of the participants assuming one of the four road bike positions alongside the ground truth output (red trace) and the YOLOv8 output fine-tuned over 10-epochs (cyan).



(a) Road bike Position 1



(b) Road bike Position 2



(c) Road bike Position 3



(d) Road bike Position 4

Figure 5.18: The 10-epoch fine-tuned YOLOv8 outputs for the road bike positions 1 through to 4 (the red trace) overlaid with the ground truth (cyan trace)

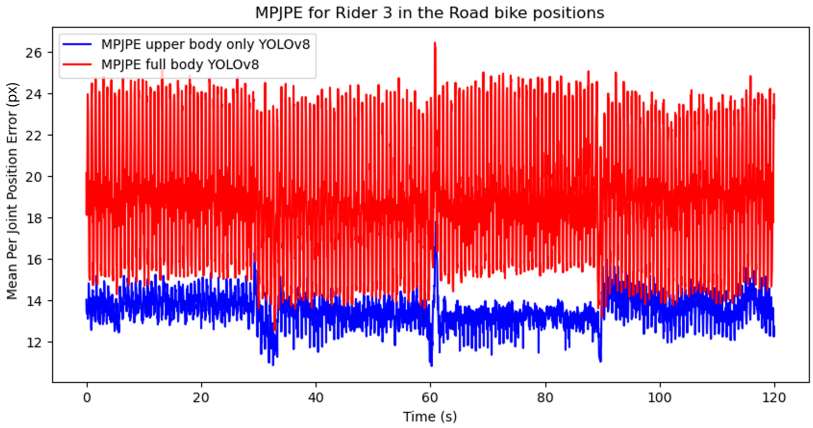


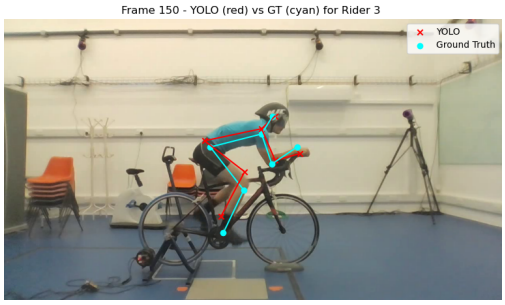
Figure 5.19: Plot showing the Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the fine-tuned YOLOv8 model over 10-epochs

Figure 5.19 shows the MPJPE across time for Rider 3 in the 4 road bike positions for one test. The red trace shows the full body MPJPE, and the blue trace shows the error in the upper body only.

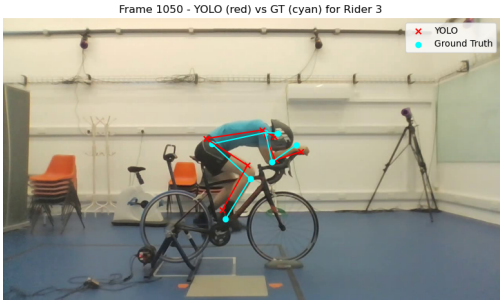
Time Trial Positions

Figure 5.20 shows an example of one of the participants assuming one of the four time trial bike positions, alongside the ground truth output (red trace) and the YOLOv8 output fine-tuned over 10-epochs (cyan).

Figure 5.21 shows the MPJPE across time for Rider 3 in the 4 road bike positions for one test. The red trace shows the full body MPJPE, and the blue trace shows the error in the upper body only.



(a) Road bike Position 1



(b) Road bike Position 2



(c) Road bike Position 3



(d) Road bike Position 4

Figure 5.20: The 10-epoch fine-tuned YOLOv8 outputs for the time trial positions 1 through to 4 (the red trace) overlaid with the ground truth (cyan trace)

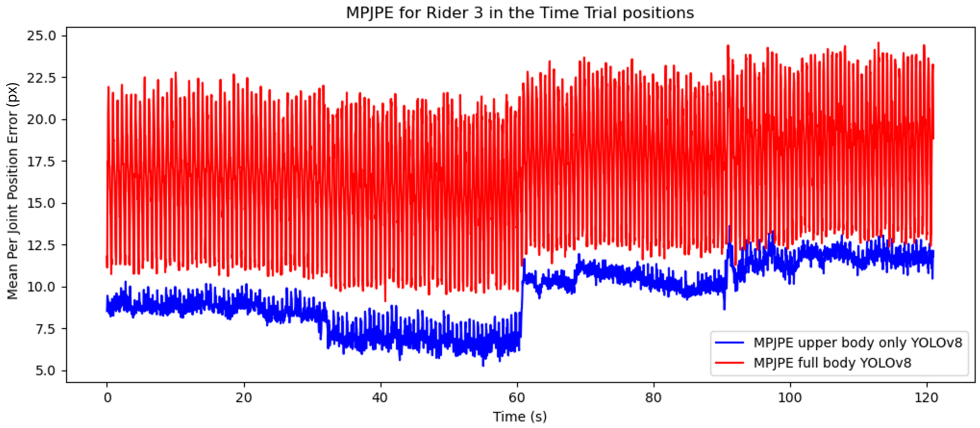


Figure 5.21: Plot showing the Mean Per Joint Position Error for the upper body joints (blue trace) and the full body joints (red trace) for the fine-tuned YOLOv8 model over 10-epochs

5.5 Discussion

Pose Detection for Road Bike Set-up

From Figure 5.22 the ground truth hip angle is plotted against time with the YOLOv8 estimate.



Figure 5.22: Plot showing the YOLOv8 calculated hip angle (red trace) compared to the motion capture ground truth (blue trace). For Rider 3 performing the road bike position sequence.

From Figure 5.22 it can be seen, due to the four distinct positions, there are clear steps in the data where the rider changes position. The larger hip angle sections are evidence of the rider sitting in the upright position, whereas the smaller hip angles are where the rider is in the more aggressive aero tucked position. For the fourth position, it is clear that this position produces the lowest hip angle on average, and this is logical, as it is where the rider is on the drop bars with tucked elbows.

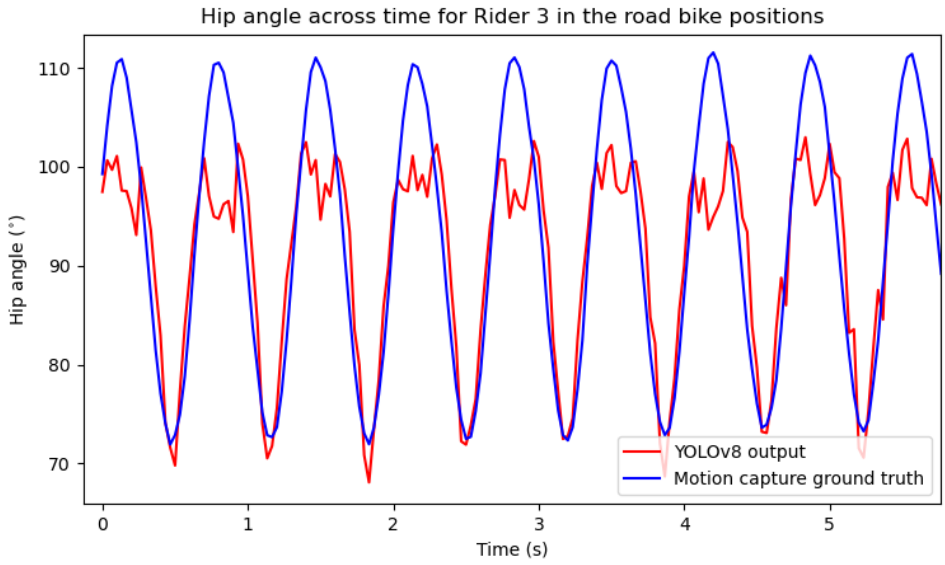


Figure 5.23: Plot showing the YOLOv8 calculated hip angle (red trace) compared to the motion capture ground truth (blue trace) for the first 6 seconds of the rider's session.

Figure 5.23 presents the first 6 seconds of the rider on the bike. This is in the less aggressive, more upright position. It can be seen that the ground truth (the blue trace) shows a clear sinusoidal pattern. In comparison, the YOLOv8 output is a lot less regular; the sinusoidal pattern is present, but the peaks of this sinusoidal pattern appear to be truncated. From 30 seconds onwards, however, when the rider is in the first aggressive position, the peaks are not truncated (Figure 5.24).

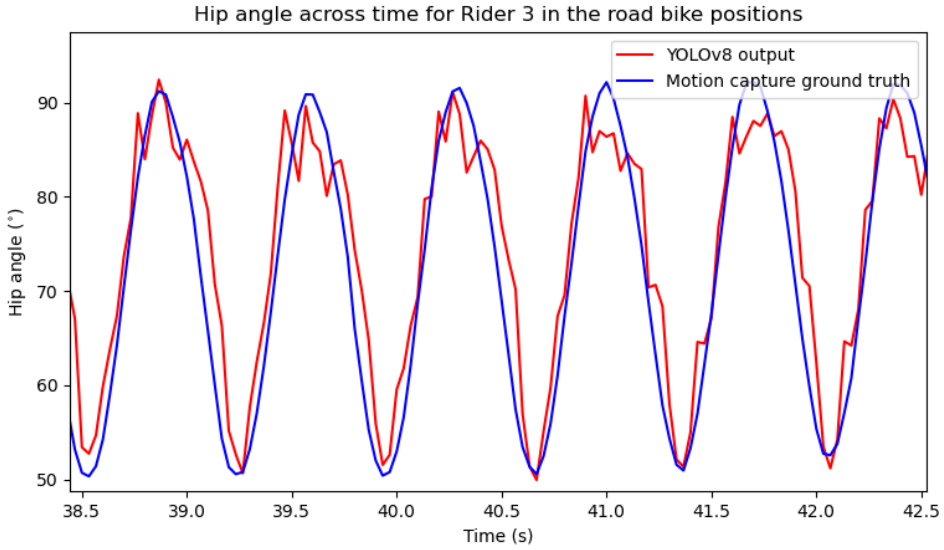
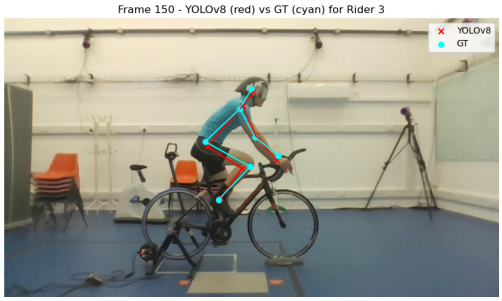
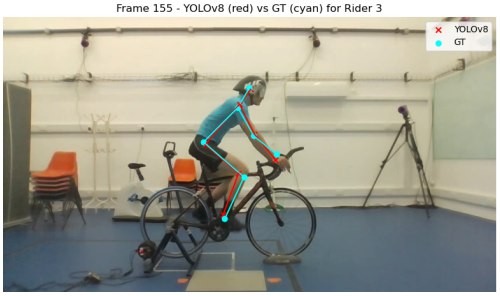


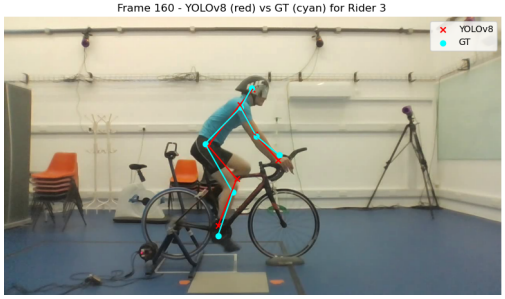
Figure 5.24: Plot showing the YOLOv8 (red trace) calculated hip angle compared to the motion capture ground truth (blue trace) for 4 seconds with the participant in position 2.



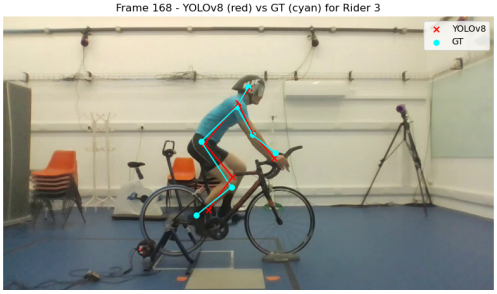
(a) Leg position at 0 degrees



(b) Leg position at 90 degrees



(c) Leg position at 180 degrees



(d) Leg position at 270 degrees

Figure 5.25: Four images of rider 3 at different positions in the pedal stroke, 0 degrees, 90 degrees, 180 degrees and 270 degrees. On each plot, the ground truth is plotted (cyan trace) alongside the YOLOv8 output (red trace).

Figure 5.25 presents the cyclist in four different locations in the pedal stroke. It can be seen visually that when the rider is at the bottom of the stroke, the pose estimation model struggles to accurately determine the location of the ankle. This is also supported by Figure 5.23, where the peaks are truncated at the higher angle values. These observations indicate that the YOLOv8 pose model has difficulty identifying the ankle and knee locations when the cyclist is at the bottom of the pedal stroke and in a more upright position. Similar findings appear in Figure 5.11.

For all of these plots, the error metric is shown across time, and the errors for both the whole body and torso only are extracted. Visual inspection suggests that the model identifies the upper body more accurately than the legs, and this conclusion is reinforced by the results in Figure 5.23. As shown in Figure 5.22, the four distinct positions are clearly visible in Figure 5.11. Examining the full

body data, positions 1 and 3 show greater variation in error. This may be due to the rider's arms being straight rather than bent in these positions.

When comparing the upper body only error with the full body error in Figure 5.11, position 4 shows a much larger joint identification error. This may be an artefact of the more aggressive posture, where the elbows occlude the knee.

Pose Detection for Time Trial Set-up

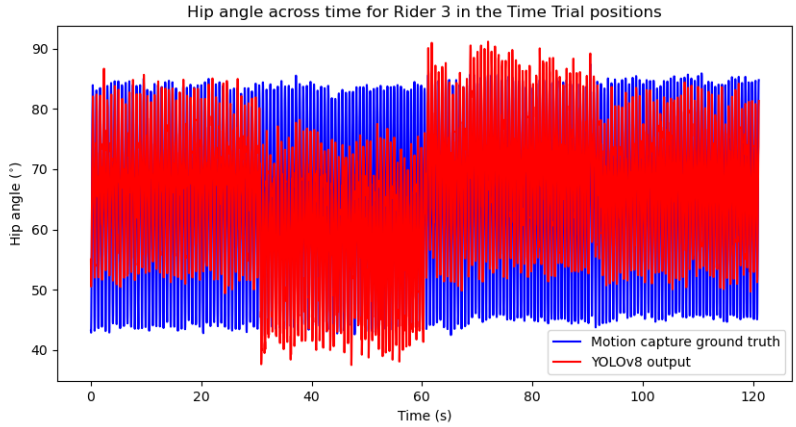


Figure 5.26: Plot showing the YOLOv8 calculated hip angle (red trace) compared to the motion capture ground truth (blue trace). For Rider 3 performing the time trial bike position sequence.

From the motion capture trace in Figure 5.26, it can be seen that there is very little hip angle variation in the time trial positions. This is expected, since the position changes involve only the head and arm placement, affecting only the upper body. The YOLOv8 output contradicts this observation, as it shows varying hip angles for each position and also presents a smaller hip angle range compared with the motion capture output. This difference in range is further supported by the first six seconds of the rider in the time trial position, shown in Figure 5.27, where the hip angle is in phase with the motion capture data, although the output is noticeably noisier.

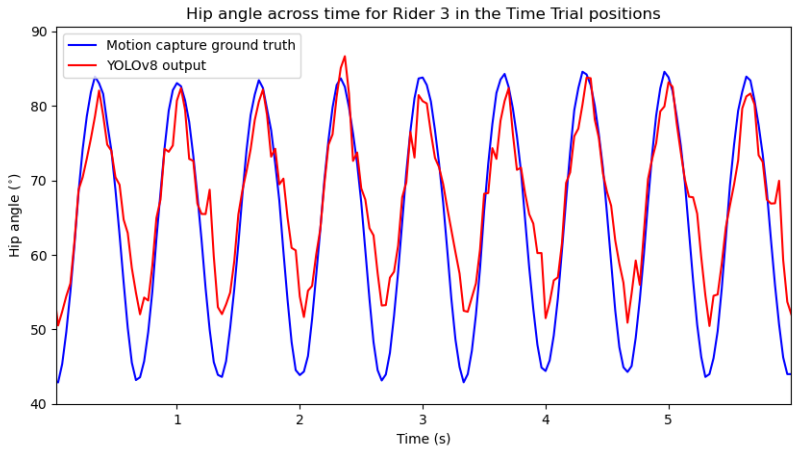


Figure 5.27: Plot showing the YOLOv8 (red trace) calculated hip angle compared to the motion capture ground truth (blue trace) Over the first 6 seconds of Rider 3 performing the time trial bike position sequence.

Figure 5.13 shows the MPJPE. It can be seen that the accuracy of YOLOv8 in the time trial positions is lower than when it is applied to the road bike positions (Figure 5.11). These results suggest that YOLOv8 performs better in more upright positions, which are closer to the human poses it was trained on, compared with viewing a rider primarily in the sagittal plane. It can also be noted from Figure 5.13 that performance is greatly reduced for position 2. This is a head-down posture, and the more compact body shape may have contributed to the poorer performance. Additionally, the results indicate that the identification of upper body joints is more accurate than that of the full body, consistent with the findings from Figure 5.11 in the road bike positions.

5-epoch fine-tuned Model

A YOLOv8 model was fine-tuned using data from Rider1 and Rider2, with Rider3 used as the validation set.

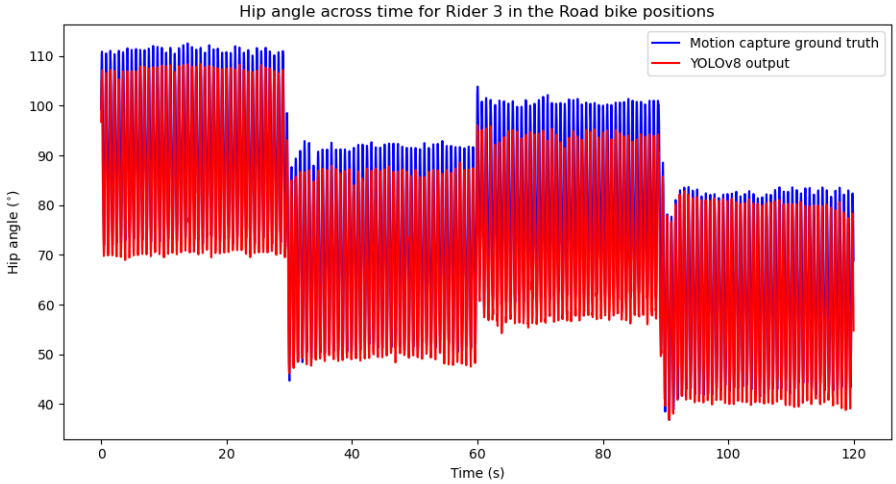


Figure 5.28: Plot showing the fine-tuned YOLOv8 (Red trace) calculated hip angle compared to the motion capture ground truth (blue trace). For Rider 3 performing the time trial bike position sequence.

To determine the accuracy of these models, the models were fine-tuned in increments of 5-epochs. Figure 5.28 shows the 5-epoch model, fine-tuned on two riders and tested on a rider who was not visible during fine-tuning. In comparison with Figure 5.22, the output is noticeably smoother. This is further supported by Figure 5.29, where the output is clearly smoother, although there appears to be a constant offset that can be easily adjusted.

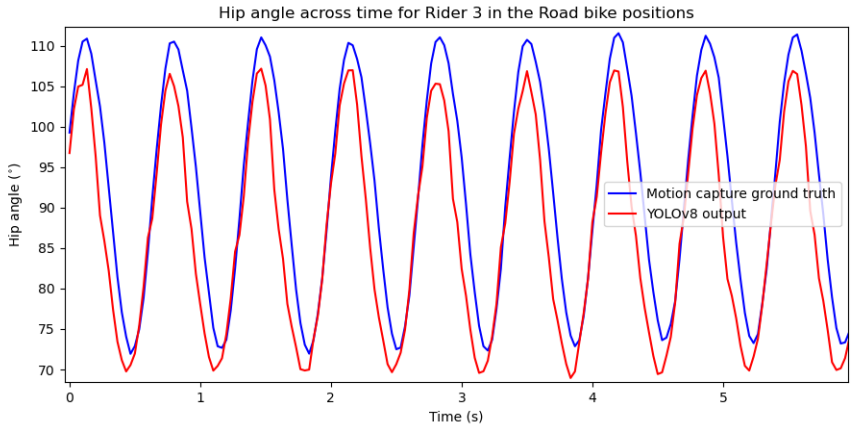


Figure 5.29: Plot showing the fine-tuned YOLOv8 (Red trace) calculated hip angle compared to the motion capture ground truth (blue trace). This is over the first 6 seconds of Rider 3 performing the time trial bike position sequence.

Figure 5.15 shows the same errors plotted for the out-of-the-box (ootb) YOLOv8 model, and it can be seen that the fine-tuned model performs worse in several areas compared with the ootb model. It also produces a number of misidentifications that are not captured in these error metrics. This may be due to intra-test differences in marker placement. The mean squared error in Figure 5.15 also has a smaller range than that of Figure 5.11.

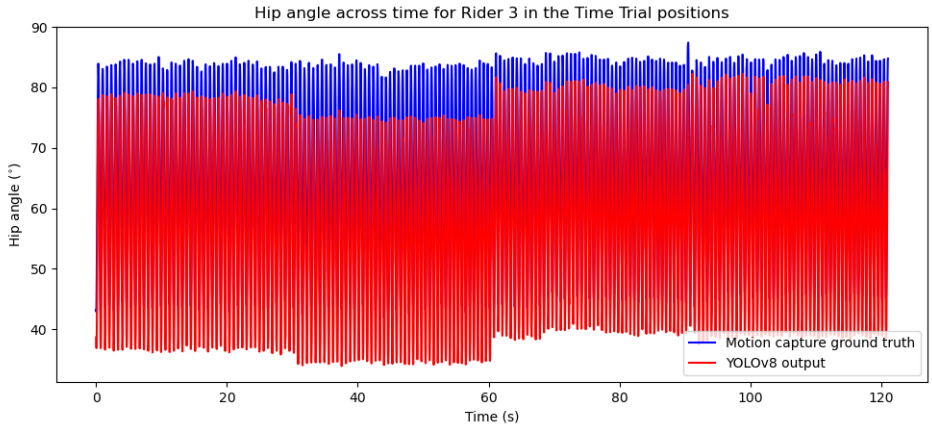


Figure 5.30: Plot showing the fine-tuned YOLOv8 calculated hip angle (red trace) compared to the motion capture ground truth (blue trace). For Rider 3 performing the time trial bike position sequence.

Figure 5.30 shows the ground truth hip angle alongside the 5-epoch fine-tuned YOLO output. As can be seen, when compared with Figure 5.27, the fine-tuned YOLO model is more robust to changes in the rider's head and arm position, although it is not entirely immune and still shows some differences in hip angle when the rider changes their head position. Additionally, much like Figure 5.28, the estimated hip angle appears to exhibit a constant offset from the ground truth. When looking at Figure 5.17 and comparing it with the ootb model, it can be noted that the model performs slightly better than the ootb model, with an MSE of 287.14 px^2 compared with the ootb model's 790.84 px^2 (Table 11). As the models were fine-tuned on the raw image data with the markers on the riders, there was speculation as to whether the fine-tuning process would learn the locations of the markers. This was not the case, within this study, due to the resolution of the laptop camera; the markers were not visible in either the fine-tuning data or the test data. As a result, it can be concluded that the fine-tuned models did not fine-tune themselves to the markers adhered to the rider, but other features that make up the rider within image.

10-epoch fine-tuned Model

In Figure 5.31 the estimated hip angle is plotted with the ground truth for Rider 3 in the road bike positions.

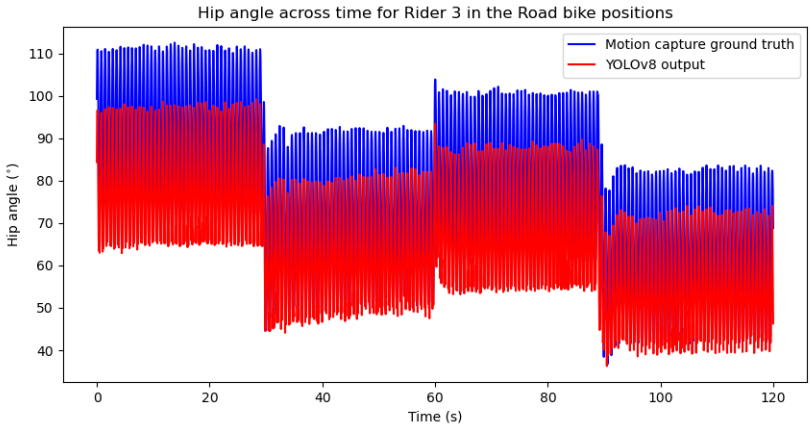


Figure 5.31: Plot showing the YOLOv8 calculated hip angle compared to the motion capture ground truth. For Rider 3 performing the time trial bike position sequence.

Here it can be seen that, as in Figure 5.28, the hip angle has a constant offset. However, as shown in Figure 5.32, there is also a periodic error where the estimated hip angle increases. Since this error is periodic, it may be possible to correct it with a simple function fitting approach, although it could also be an artefact of overfitting to the training data.

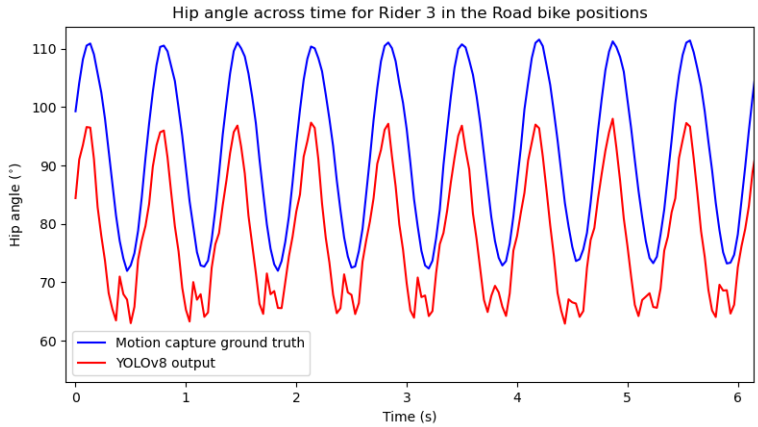


Figure 5.32: Plot showing the YOLOv8 calculated hip angle compared to the motion capture ground truth. For Rider 3 performing the time trial bike position sequence.

It can be seen from Figure 5.19 that the errors follow the same trend as in the other results, where the upper body has a smaller error than the full body, likely due to inaccuracies in identifying the ankle and knee. In this case, the MPJPE is greater than that of the ootb model, possibly due to overfitting.

5.6 Conclusion

As shown in the results of this study, an ootb YOLOv8 model is not suitable for accurate pose estimation for cyclists in the sagittal plane. It was observed that metrics such as the hip angle are influenced by upper body posture, with more upright seated positions causing an underestimation of the hip angle. It was also noted that in more tucked positions, the YOLOv8 pose estimation model lost accuracy, for example, in the drop bar tucked position on the road bike, in the head down position on the TT bike, and more generally across all time trial positions. These findings align with previous research, which reported that YOLOv8 performs worse in more compact or tucked body positions, likely due to occlusions and the absence of such poses in the original training set, similar to the results reported by Huang et al. (2019).

To counteract the errors observed in the more tucked positions, an approach was taken to fine-tune the ootb model on two of the riders and test it on the third rider in the time trial positions. It was found that fine-tuning over 5 epochs was sufficient to increase the accuracy of the pose estimation model. However, as fine-tuning progressed to 10 epochs, clear artefacts of overfitting emerged.

The 5-epoch model produced hip angle traces similar to the ground truth, with a small constant offset. In the results collected in this study, the fine-tuned model showed a lower OKS than the unfine-tuned model. However, it produced less noise than the ootb model when examining the full body results. This trend was

also reflected in both the MSE and MPJPE metrics. When comparing the TT positions, the hip angle estimation was influenced by upper body movement to a lesser degree than in the ootb YOLO model, resulting in hip angle estimates that were closer to the ground truth.

The results from this study demonstrate that it is possible to generate a cycling-specific pose estimation model and, through fine-tuning, improve accuracy in more compact positions such as time trial postures. Fine-tuning also improved the identification of the ankle and knee, leading to more accurate hip angle estimation. To further improve performance, it may be useful to increase the cohort size and expand the variety of clothing worn by riders. Introducing a wider range of background environments could also be beneficial. Beyond the scope of this study, an additional area of interest would be investigating the impact of occlusions and improving the detection of the left leg joints would also be beneficial.

Building upon this foundational work in cycling pose estimation, future research could explore the development of a domain-specific Large Language Model (LLM) tailored for cycling aerodynamics. Such an LLM, trained on aerodynamic studies, computational fluid dynamics (CFD) simulations, wind tunnel data, and athlete performance metrics, could provide insights into optimising rider position, equipment selection, and overall aerodynamic efficiency. This would represent a significant progression toward a more holistic understanding of cycling performance by integrating visual data with theoretical and empirical knowledge.

Key Point Summary

- A dataset of 3 riders in 42 unique configurations was collected in a state-of-the-art motion capture studio to evaluate against computer vision pose models.
- An out-of-the-box (ootb) YOLOv8 pose model was tested and compared against the ground truth. It was concluded that an ootb YOLOv8 model is not accurate enough to determine the joint locations of riders in the sagittal plane in tucked road bike and time trial positions.
- A YOLOv8 model was fine-tuned on data collected from two of the three riders and tested on the data of the third rider. The fine-tuned model over 5 epochs showed good joint location identification; however, when trained over 10 epochs, it showed signs of overfitting.

Chapter 6

Application of Large Language Modelling for a Personal Aerodynamics' Coach

Throughout this thesis, several tools have been developed. These tools are useful; however, they can be limited by the person using the tools and interpreting the results. Usually, an expert would fill this gap, such as a bike fitter, coach or data analyst. Their role would be to look through the data and use the tools. Unfortunately, access to these experts can require a lot of capital or already be a full-time professional. This leaves a gap for those who are physically fit enough to compete with the professionals, yet do not have access to human experts to chase marginal gains. With the rise of Large Language Models (LLMs) this chapter investigates whether an LLM expert could perform to the same level as a human expert.

6.1 Introduction

Accessing expert knowledge on aerodynamics can be a challenge due to limited resources and expertise. For many athletes, this presents a barrier to entry, making it difficult for them to participate at a competitive level when compared to athletes with substantial financial backing and sponsors. As established, Body Rocket Ltd has created a system of sensors which directly measure the drag force of a rider on the bike, therefore helping alleviate the need for wind tunnel visits like professional teams and athletes do. Throughout this thesis, shuffling and positional identification with the support of the Body Rocket system has been investigated alongside the application of pose estimation models. Each of these studies alone investigate an area of interest within cycling and can be seen as useful tools, however, to make the best use of these tools a human expert could provide some insight. In this case, a human expert will cost money and may also be as inaccessible as wind tunnels. This study has one main aim:

- To investigate the application of Language Models and determine whether they have the capability of performing as well as a human expert.

With inspiration from the other chapters in this thesis, examples of how an LLM cycling chatbot expert can be applied include Using statistical techniques to analyse post-ride data, for example, examining how often a rider shuffles and determining its impact on their performance. This analysis could be done by the LLM chatbot, and afterwards, protocols and suggestions could be provided to aid in the performance of the cyclist.

For positional classification or pose estimation, a domain-specific cycling chatbot could be used similarly. The position across a ride or the joint locations and continuos position of a rider could be submitted to the LLM chatbot and analysis could be performed on that data to evaluate whether a position is sustainable for

a period of time or make recommendations on changing the riders position.

In a wind tunnel setting, the human expert (a professional aerodynamicist or bike fitter) will set up an initial testing protocol, analyse the data, and from this guide the athlete on how to get more aerodynamic. Considering this traditional approach and the aim to investigate the application of Language Models as a substitute for human experts, this study will leverage the advent of Large Language Models (LLMs) and Generative Artificial Intelligence (Generative AI). Specifically, the study aims to create a local LLM chatbot where the user's data is not passed to a third party, kept secure, ensures that the model remains domain-specific, and ensures future reliability.

The recent rise of large language models (LLMs) in language modelling, exemplified by systems like ChatGPT (OpenAI 2023), is significant. These AI-based models, which are fundamentally built upon transformer-based neural network architectures (Brown et al. 2020; Vaswani 2017), offer a promising solution for overcoming the typical scarcity of expert knowledge in fields such as cycling aerodynamics by enabling effective data analysis and knowledge dissemination. An LLM operates by first converting textual input into a high-dimensional vector representation. It then iteratively predicts the subsequent vector by analysing the magnitude and directional relationships of all preceding dimensional vectors. This predicted vector is then translated back into textual output, continuing until a complete response is formed. LLMs are utilised for a variety of functions, including automated writing suggestions, summarising large amounts of text, and acting as conversational agents, as seen with OpenAI's ChatGPT (OpenAI 2023) and Google's Gmail summarisation (Google 2025). The transformer architecture relies heavily on multi-layered matrix multiplication. The number of weights within these matrices determines an LLM's parameter count. A widely accepted theory suggests that increasing the parameter count generally leads to improved model performance (Lee et al. 2021a).

Models such as ChatGPT (OpenAI 2023) and Gemini (Google 2024) are examples of general models, meaning they were trained on a wide range of material with no specific area of interest, or as will be referred to in this chapter, domain. To give a Language Model a domain focus, there are a number of techniques which can be implemented. Three of which include: training a model from scratch on material only in the domain of interest, fine-tuning a model on domain-specific data and Retrieval Augmented Generation (RAG). Both training from scratch and fine-tuning can require an extensive computational effort. RAG, on the other hand, does not require the same amount of computational effort.

RAG is a methodology wherein domain-specific knowledge is indexed and stored within a vector database. Upon request for information, a retriever component fetches relevant information from this vector store, subsequently incorporating it into the generated response. This approach has been demonstrated to significantly mitigate the incidence of "hallucinations" in LLM outputs (J. Li et al., 2024).

The primary aim of this study is to investigate the efficacy of a variety of open-source LLMs with and without RAG in providing accurate knowledge in response to queries concerning cycling aerodynamics. Secondly, this study endeavours to assess the comparative performance of these open-source LLMs, integrated with a basic RAG pipeline (as depicted in Figure 6.4), against commercially available models such as ChatGPT.

Language Model Applications

With recent advancements in Language Modelling, there is now an abundance of research in the field and its applications. With general commercial models such as ChatGPT (OpenAI 2023), grok (xAI 2025) and Gemini (Google 2024), useful for general tasks, there are many applications of language modelling both within sport and across other fields. Within cycling, more specifically Janssens, Bogaert

and Verstockt (2025) used models from OpenAI, specifically ChatGPT 3.5 and 4o to determine where riders are positioned in the race and retrieve it in a structured manner. In this study, ChatGPT 4o outperformed GPT3.5. One of the challenges of a domain-specific LLM is how to verify whether the responses it makes are accurate and true within the domain. One study from Xia et al. (2024) set out to create a SportsQA benchmark where the questions in this benchmark are within the domain of sports. This included information on 35 different sports and presented 70,592 Multiple Choice Questions (MCQs). In this study Llama2 (Touvron et al. 2023) PaLM (Chowdhery et al. 2023), ChatGPT3.5 and ChatGPT4 was tested. ChatGPT4 consistently outperforms all the other models. Other works have investigated the quality of LLMs within sports commentary (Cook and Karakuş 2024). Their work employed a llama-2-7b that was trained to produce 'timely and accurate insights into the game'. Outside of the domain of sport and cycling, there have been advancements in other fields such as medicine and clinical decision-making. Kresevic et al. (2024) used RAG in conjunction with OpenAI's GPT-4 Turbo and hospital clinical guidelines, which presents a level of trust in LLMs and their potential to contribute to positive life-changing outcomes. Also within the domain of medicine Ke et al. (2025) applied RAG to 10 LLMs to assess the generalizability of assessing medical fitness. Within Ke et al. (2025) study GPT3.5, GPT4, GPT4o, Gemini, Llama2, and Llama3, Claude were tested and the best performing model was GPT4.

6.2 Theory

Large Language Models (LLMs) are transformer-based models which take text inputs and predict the next words from the words prior (Brown et al. 2020). In machine learning a transformer is a deep learning architecture introduced by Vaswani (2017) and an explanation of their inner working are to follow.

With a Large Language Model as a chatbot, it can be broken down into a simple, input and output, where text is input into the model and with respect to that input text, the next word is predicted and so forth. For the model to 'understand' the user's input, it needs to be tokenised.

Tokenization

Tokenization breaks down a string of text into chunks, which allows for them to be assigned a numerical equivalence (embedded) and is essential for the LLM to be able to interpret the user's input (Webster and Kit 1992) , in the case for this study the user input would be a query or statement regarding cycling aerodynamics from an athlete requiring assistance. Within Natural Language Processing methods of tokenization vary, this include word tokenization (Yang 2024), character/byte level tokenization and sub-word tokenization (Schmidt et al. 2024). Each type of tokenization has it's benefits and drawbacks, if there are set token limits this can be a drawback for byte level encoding, however, word tokenization also has a limitation when the desired output is on the character level, such as spelling a word backwards.

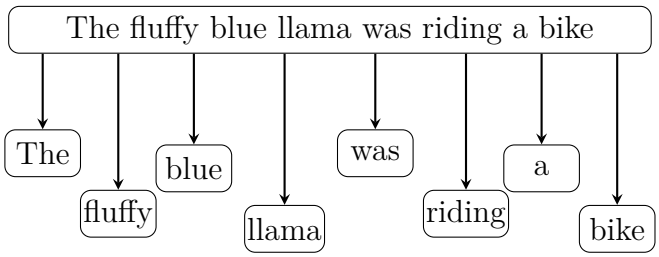


Figure 6.1: Tokenization of the sentence "The fluffy blue llama was riding a bike". Each word is treated as a separate token.

Embedding

Embedding in natural language processing and large language models is the process of converting tokens of text into a numerical representation, which also provides semantic information. In LLMs, tokens are represented in N-dimensional vectors. Each token will have an associated vector which can be represented in an n-dimensional space. The direction of this vector can then be representative of its semantic meaning (Wang et al. 2023). With vector embedding, it is also possible to apply these to retrieval tasks such as Retrieval Augmented Generation (Lewis et al. 2020). When building an LLM, all of the vector values for each token are set to a random number, and it is during the training process where the embedding takes place, assigning a value to the vector representation of a token (Mikolov 2013).

$$E_m = \begin{matrix} \text{Word1} \\ \begin{bmatrix} v_{11} \\ v_{21} \\ \vdots \\ v_{m1} \end{bmatrix} \end{matrix} \quad \begin{matrix} \text{Word2} \\ \begin{bmatrix} v_{12} \\ v_{22} \\ \vdots \\ v_{m2} \end{bmatrix} \end{matrix} \quad \begin{matrix} \text{Word3} \\ \begin{bmatrix} v_{13} \\ v_{23} \\ \vdots \\ v_{m3} \end{bmatrix} \end{matrix} \quad \cdots \quad \begin{matrix} \text{Word}_{n+2} \\ \begin{bmatrix} v_{1(n+2)} \\ v_{2(n+2)} \\ \vdots \\ v_{m(n+2)} \end{bmatrix} \end{matrix} \tag{61}$$

Another key component of an LLM is not only the token embeddings but also the position embedding of those tokens. This component of an LLM is very important to ensure that when the attention mechanism is applied, the sentences produced are 'understood' to a better extent. Consider these two examples:

"The fluffy blue llama was riding a bike" (62)

And:

"The fluffy llama was riding a blue bike" (63)

Both the first sentence and the second sentence contain the same words; however, they are inherently different in their meaning due to the positions of the words within the two sentences. As a result, positional embeddings are required to help the model understand token order. Rotary Position Embedding (RoPE) is one such technique, introduced to replace fixed or learned positional embeddings. RoPE works by applying a rotation matrix to each token embedding in a way that encodes its position. This allows for better extrapolation to longer sequences and preserves relative position information more naturally in attention mechanisms. In 2 dimensions, RoPE takes the form (Su et al. 2024):

$$f_{\{q,k\}}(x_m, m) = \begin{pmatrix} \cos(m\theta) & -\sin(m\theta) \\ \sin(m\theta) & \cos(m\theta) \end{pmatrix} \begin{pmatrix} W^{(11)}_{\{q,k\}} & W^{(12)}_{\{q,k\}} \\ W^{(21)}_{\{q,k\}} & W^{(22)}_{\{q,k\}} \end{pmatrix} \begin{pmatrix} x^{(1)}_m \\ x^{(2)}_m \end{pmatrix} \tag{64}$$

For the general case and optimising for computational efficiency RoPE takes the form:

$$R^d_{\Theta,m} x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \cos(m\theta_1) \\ \cos(m\theta_1) \\ \cos(m\theta_2) \\ \cos(m\theta_2) \\ \vdots \\ \cos\Big(m\theta_{d/2}\Big) \\ \cos\Big(m\theta_{d/2}\Big) \end{pmatrix} + \begin{pmatrix} -x_2 \\ x_1 \\ -x_4 \\ x_3 \\ \vdots \\ -x_d \\ x_{d-1} \end{pmatrix} \otimes \begin{pmatrix} \sin(m\theta_1) \\ \sin(m\theta_1) \\ \sin(m\theta_2) \\ \sin(m\theta_2) \\ \vdots \\ \sin\Big(m\theta_{d/2}\Big) \\ \sin\Big(m\theta_{d/2}\Big) \end{pmatrix}. \tag{65}$$

Self-attention

Once words are embedded, part of their meaning is now encoded; however, there are other steps in the process that are required to ensure context-specific meaning is understood. Using the previous example to show tokenization:

"The fluffy blue llama was riding a bike" (66)

In this example each one of these words could be tokenized and embedded, however, there are context dependent words within this sentence. Take 'llama', normally llamas are white, grey, brown or black (Anello, Daverio and Di Rocco 2022) i.e. not blue. If a standalone embedding was used, the adjectives that describe the llama would not have an impact on the models 'understanding. This is where attention can be applied. Attention takes the vector values of each word in matrix E_m in Equation (61) and changes the vector values with respect to the other words in the sentence.

Before attention is computed, the model forms three learned linear projections of the embeddings. Each token's embedding is multiplied by separate weight matrices to produce the queries Q, keys K and values V. The query represents what a token seeks to attend to, the key represents what information that token exposes to others, and the value contains the information that will be combined once relevance has been established. Attention weights arise from comparing each query with all keys, and these weights determine how strongly each value contributes to the updated token representation. To achieve this, a softmax is applied to the data (Vaswani 2017).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{67}$$

Where Q is the query matrix, K is the key matrix, V is the value matrix, and d_k is the dimensionality of the key query vectors.

Softmax

The softmax function converts raw similarity scores, such as logits produced by a language model, into a probability distribution. It operates by exponentiating each score and normalising by the sum of the exponentiated scores, ensuring that all values are non-negative and sum to one. This allows the model to express its internal scoring as probabilities over possible next tokens or classes (Bengio et al. 2003).

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{68}$$

The matrix E_m then is a collection of vectors where the vector values are different to the straightforward embedded vectors as they are impacted by the other words within the sentence.

Multilayer Perceptron

The Multilayer Perceptron (MLP) is a key component of transformer-based language models and contributes to how these models appear to encode and retrieve factual associations during pre-training. Conceptually, the MLP consists of a sequence of linear transformations separated by a nonlinear activation. Although simple in structure, these layers allow the model to construct higher-level abstractions from contextualised token embeddings. Consider the adapted example sentence:

"The fluffy blue llama was riding the bike in the yellow jersey this means" (69)

In the context of cycling, the phrase "yellow jersey" implies that the rider is the current race leader (Cohen et al. 2021). Understanding how a language model retrieves this association provides insight into the function of the MLP. As we have established before reaching the MLP, each token is embedded and modified through attention, producing contextual vectors that can be interpreted as directions within a high-dimensional semantic space. For example, the embedding for "yellow jersey" will carry components aligned with both "yellow" and "jersey," along with additional context-dependent structure.

The MLP, also referred to as the Feed Forward Network (FFN), refines these contextual representations. It consists of a dimensional expansion, a nonlinear activation and a projection back to the model dimension:

$$\mathrm{FFN}(x) = \mathrm{Act}(xW_1 + b_1)W_2 + b_2. \tag{70}$$

The first matrix W_1 expands the dimensionality of the token representation, enabling the model to express a richer set of intermediate features. Rows of W_1 behave as learned feature detectors that respond to particular semantic or contextual patterns in the input. The weights of the MLP are tuned during pretraining, and the resulting transformations encode statistical regularities present in the data. If the internal representation associated with "yellow jersey" strongly activates a feature that correlates with "race leader," the dot products inside the MLP amplify that relationship.

The activation function then shapes these responses by introducing nonlinearity. This ensures that only the relevant semantic features trigger the correct association, while other similar but unrelated phrases (such as "green jersey" or "yellow banana") do not. Common activation functions include:

ReLU (Rectified Linear Unit) is defined as:

$$\text{ReLU}(x) = \max(0,\, x) \tag{71}$$

GeLU (Gaussian Error Linear Unit) is defined as:

$$\text{GeLU}(x) = \frac{1}{2}\, x \left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right) \tag{72}$$

SwiGLU (Switchable Gated Linear Unit) is defined as:

$$\text{SwiGLU}(x) = x^2 \cdot \sigma(x) = x^2 \left(\frac{1}{1 + e^{-x}}\right) \tag{73}$$

This process is iterated over many times, in the case for llama 8b, 70b and 405b it is 32, 80 and 126 respectively. After the final layer, the model produces a final hidden vector for the next token. A softmax is then applied to convert this vector into a probability distribution over the vocabulary, selecting the most likely token. The procedure continues token by token until the context limit is reached or a stopping condition is met.

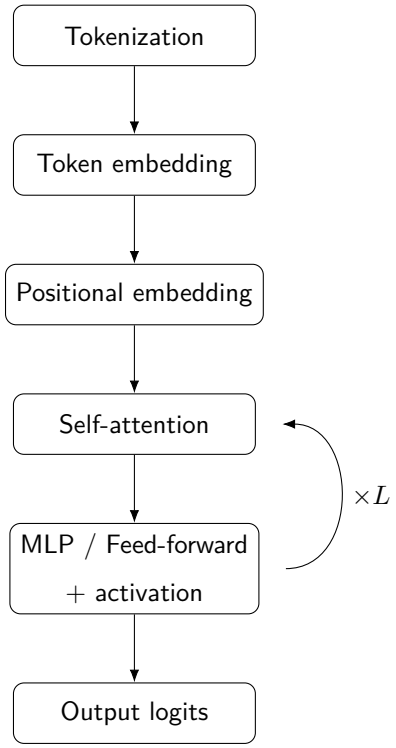


Figure 6.2: Simplified LLM processing pipeline: tokenization, token and positional embeddings, repeated self-attention and MLP layers, and final output logits.

Domain Specificity

Most widely available and high-performance large language models (LLMs) are trained as general-purpose systems such as ChatGPT (OpenAI 2023), grok (xAI 2025) and Gemini (Google 2024). While this makes them versatile across a broad range of tasks, it presents challenges when the model needs to operate effectively within a specific domain, such as law, medicine, engineering or cycling aerodynamics. Generalised LLMs may lack the necessary depth of knowledge or produce inaccurate or overly generic responses when confronted with specialised queries. The system prompt is a useful tool in making Language Models domain-specific as they allow for models to have a pre-defined area to focus on, alongside allowing the model to 'understand' how it should respond (Neumann et al. 2025). However,

there are additional methods which can be used to improve the performance of a language model within a given domain.

To adapt an LLM for domain-specific applications, two primary strategies are commonly employed: Retrieval-Augmented Generation (RAG) (Lewis et al. 2020) and fine-tuning (Lu, Luu and Buehler 2025). RAG enables the model to access external domain-specific data during inference, ensuring responses are grounded in relevant knowledge. Fine-tuning, on the other hand, involves training the model further on a curated corpus from the target domain, embedding domain expertise directly into the model's parameters. Both methods offer distinct advantages and can be used independently or in combination, depending on the application requirements and constraints.

Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) is a hybrid approach that combines retrieval-based methods with generative language models to improve factual accuracy and reduce hallucinations in language model outputs. Rather than relying solely on the parametric memory of the model, RAG enables dynamic incorporation of external knowledge at inference time. Within this study this would be information relating to cycling aerodynamics. This is achieved through two main components: a retriever and a generator.

Cosine similarity is a measure used to evaluate the similarity between two vectors, making it highly suitable for comparing document embeddings in retrieval-based applications (Steck, Ekanadham and Kallus 2024). It is computed as:

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{74}$$

where A and B are the embedding vectors of the query and the document, $A \cdot B$

is the dot product of the two vectors, $\|A\|$ and $\|B\|$ are the Euclidean norms (magnitudes) of the vectors.

The retriever module searches a pre-indexed external knowledge base often composed of documents, structured data, or previous user interactions—based on the input query. It returns the top-k most relevant passages, which are then concatenated or otherwise integrated into the prompt that is passed to the generative model. This allows the model to generate responses that are grounded in real-world information, even if that information was not part of its original training data.

RAG was introduced by Lewis et al. (2020), where the authors demonstrated its application using a dense retriever (DPR) combined with a sequence-to-sequence transformer model. It has since been put into practice in various domains, such as customer support (Xu et al. 2024), scientific research (Lozano et al. 2023), and domain-specific assistants (Benfenati et al. 2024).

6.3 Materials and Methods

This study aims to produce a rigorous comparison between different Language Models and Language Model architectures. To achieve this, a new benchmark is needed. This benchmark comprises of 50 multiple-choice questions which are within the area of bike fit, aerodynamics and physiology. To establish the accuracy of this benchmark, experts from academia and industry were recruited to answer these questions and establish the answers which are correct. Alongside the 50 questions asked of the experts, the question sheet also comprised 13 questions around the Body Rocket system. In the evaluation process, the most common answer selected by the industry experts was selected as this indicated that the answer is correct. This can then be compared to the response from the LLM. To evaluate

the model the stages were broken down into different steps, starting from a simple model, a model with a system prompt, a model with RAG. It would be expected that as the number of parameters increases, the inclusion of system prompts and the implementation of RAG, the performance of the LLMs will increase, i.e. an increasing accuracy for each step. Following institutional ethical approval from the University of Kent, a cohort of 13 subject matter experts (SMEs) was recruited for this study. These SMEs possessed extensive experience in domains critical to cycling performance, including coaching, bike fitting, and aerodynamics. The primary objective of their involvement was to validate the 50-question MCQ sheet produced in this study to evaluate the capabilities of the LLMs. An initial online multiple-choice questionnaire (MCQ), comprising 50 questions, was developed using a combination of generative artificial intelligence techniques. This questionnaire underwent an iterative refinement process with the assistance of the recruited SMEs to ensure its scientific rigour and appropriate level of challenge. The final MCQ encompassed a broad range of topics, including but not limited to bike fit principles, human biomechanics, exercise physiology, and applied aerodynamics in cycling. This validated MCQ was subsequently administered to the broader cohort of experts to establish a "ground truth" comparator for the LLMs' performance.

Language Model Evaluation

Upon completion of the expert-evaluated questionnaires, a collective expert consensus was established for each question. This consensus served as the benchmark against which the performance of various LLMs was quantified. The LLMs were hosted on a high-performance server equipped with an NVIDIA Tesla T4

GPU (Nvidia, 2025). Both open-source and closed-source models were evaluated for their knowledge retrieval capabilities. The open-source models tested included: `llama3.2:3b` and `llama3.1:8b` (Grattafiori et al., 2024), `Gemma2:9b` and `Gemma2:27b` (Team et al., 2024), `Phi4:14b` (Abdin et al., 2024), and `Qwen2.5:7b` and `Qwen2.5:32b` (Qwen et al., 2025). The closed-source models evaluated were ChatGPT 3.5 Turbo, ChatGPT-4o, and the Large Reasoning Model o1-mini. The selection of these models was purposefully constrained to smaller, computationally efficient architectures.

Each LLM was tested against the same set of 50 questions provided to the human experts, in addition to 13 supplementary questions specifically tailored to Body Rocket's proprietary knowledge domain. The 13 additional questions were designed to assess the LLMs' ability to process and synthesise information related to Body Rocket's unique data acquisition methodologies and analytical frameworks for aerodynamic optimisation, which are not widely disseminated in public knowledge bases. To ensure a representative evaluation of each model's capabilities, the assessment process was conducted over 10 iterations for each question.

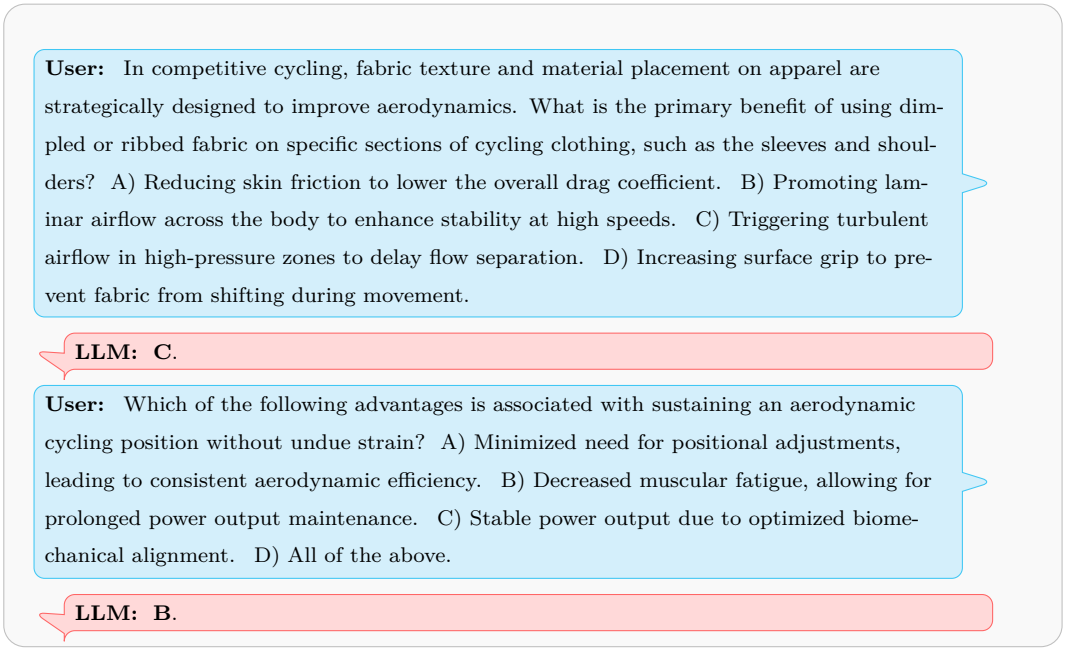


Figure 6.3: Visual representation of the prompt and response process used for each large language model to evaluate the knowledge of each model

Each model from each respective source was tested in 4 different configurations: one without a system prompt, one with a system prompt, and one with a general knowledge base covering the knowledge of an enthusiastic cyclist. The final configuration involved the use of material covering that of the questionnaire, alongside material not covered in the questionnaire, akin to a student receiving course material and being examined on it.

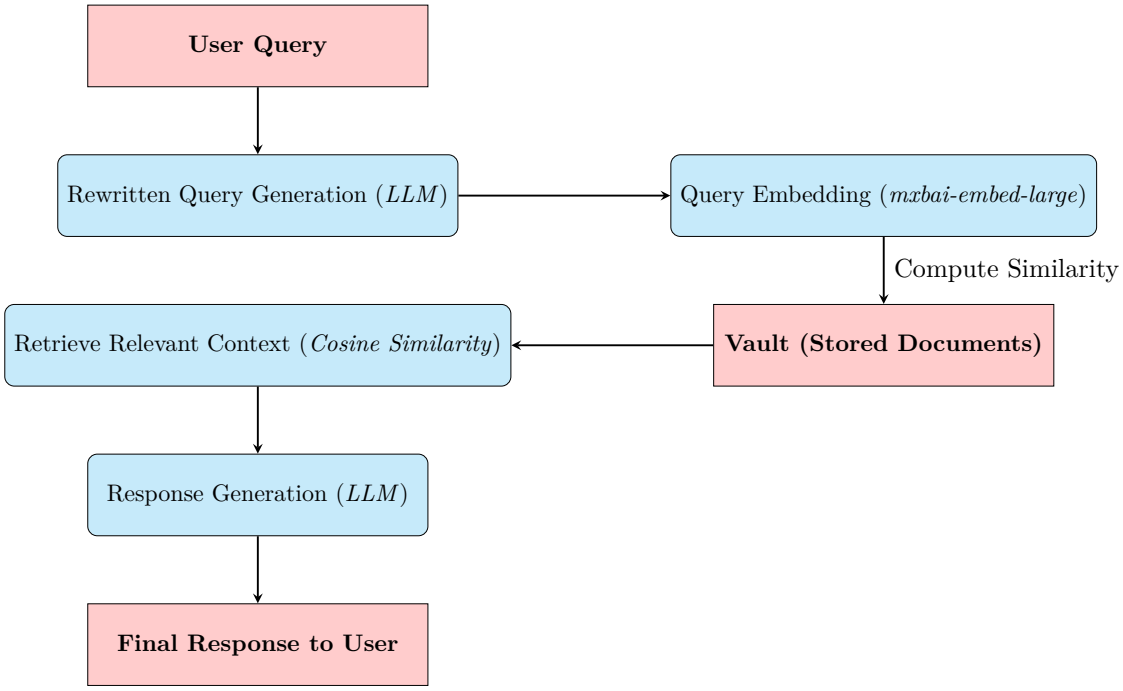


Figure 6.4: Overview of the Retrieval-Augmented Generation (RAG) Pipeline with Model Details.

Figure 6.4 shows the RAG pipeline implemented in this study based upon a public git repository (AllAboutAI-YT 2025). This pipeline re-writes the users query, embeds it with the mxbai-embed-large embedding model (Lee et al. 2024) computes the similarity with the stored knowledge, retrieves the relevant context, and then generates a response from the LLM providing a final response with the user.

Statistical Analysis

Given that each model and its respective configurations underwent 10 iterations, it was possible to compute both Shapiro–Wilk p-values (SHAPIRO and WILK 1965) and Coefficients of Variation (CV) (Vangel 1996) for each configuration. The 10-repetition structure provides a robust basis for statistical inference. The Shapiro–Wilk p-values assess the normality of the results, indicating whether the

distribution of the repeated measurements is consistent with a normal distribution. Meanwhile, the Coefficients of Variation quantify the relative variability and precision of each model configuration, allowing for a comparative evaluation of their stability and reliability across iterations.

6.4 Results

Llama Models

In Figure 6.5 it can be observed that a discernible trend exists whereby models with increased parameter counts exhibit superior performance. Specifically, the llama3.2:3b model achieved a mean performance metric of $(\mathbf{76.35 \pm 1.07})\,\%$ with an associated p-value of 0.594082 and a coefficient of variation of 4.43. In contrast, the optimally performing llama3.1:8b model yielded a mean performance metric of $(\mathbf{78.10 \pm 0.62})\,\%$ with a p-value of 0.39 and a coefficient of variation of 2.50. Both models exhibit p-values > 0.05, indicating they are likely to come from a normal distribution, and the llama3.1:8b model, with a smaller CV of 2.5, indicates more consistent responses.

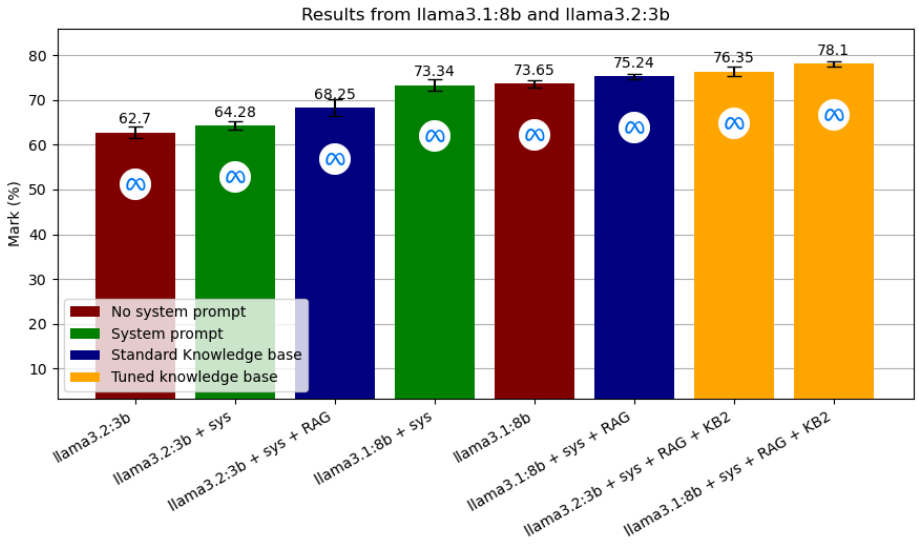


Figure 6.5: Bar chart comparing the performance of two llama models, llama3.1:8b and llama3.2:3b on the cycling aerodynamics assessment.

Gemma2 Models

Similarly, Figure 6.6 illustrates a positive correlation between model scale and predictive accuracy, particularly when evaluated against collective expert consensus. This finding aligns with general trends observed in the scaling of large language models, where increased parameter counts often lead to enhanced performance due to greater capacity for learning complex representations. Specifically, the optimal configuration of the Gemma2:9b model achieved a mean accuracy of $(\mathbf{73.81 \pm 0.79})\,\%$ with an associated p-value of 0.85 and a coefficient of variation of 3.39. In contrast, the superior Gemma2:27b model demonstrated a notably higher mean accuracy of $(\mathbf{87.30 \pm 0.53})\,\%$, accompanied by a p-value of 0.07 and a coefficient of variation of 1.92. The lower p-value of the Gemma2:27b indicates the data is less normal than the data collected from the Gemma2:9b model; however, given that it is still above 0.05, it can still be considered normal.

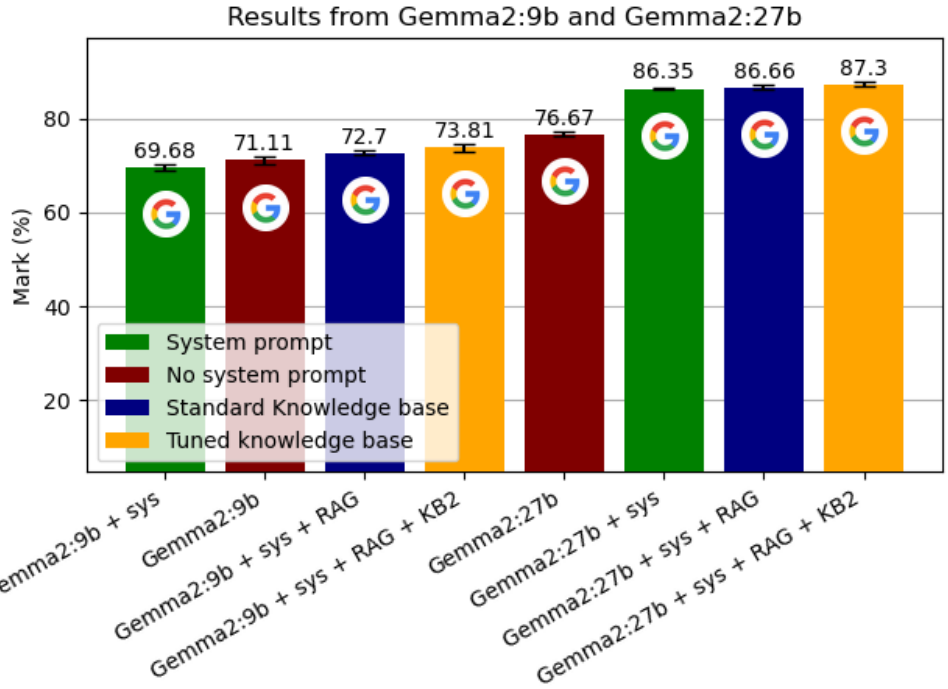


Figure 6.6: Bar chart comparing the performance of two Gemma2 models, Gemma2:9b and Gemma2:27b.

Phi4 – 14b

Figure 6.7 shows the performance of the Phi4:14b model here the configuration with the highest accuracy has an average mark of $(\mathbf{85.24 \pm 0.53})\,\%$ a p-value of 0.052 and a coefficient of variation of 1.97. With a p-value$>$ 0.05, it does indicate the data is normal; however, it is one of the smaller p-values obtained in this study.

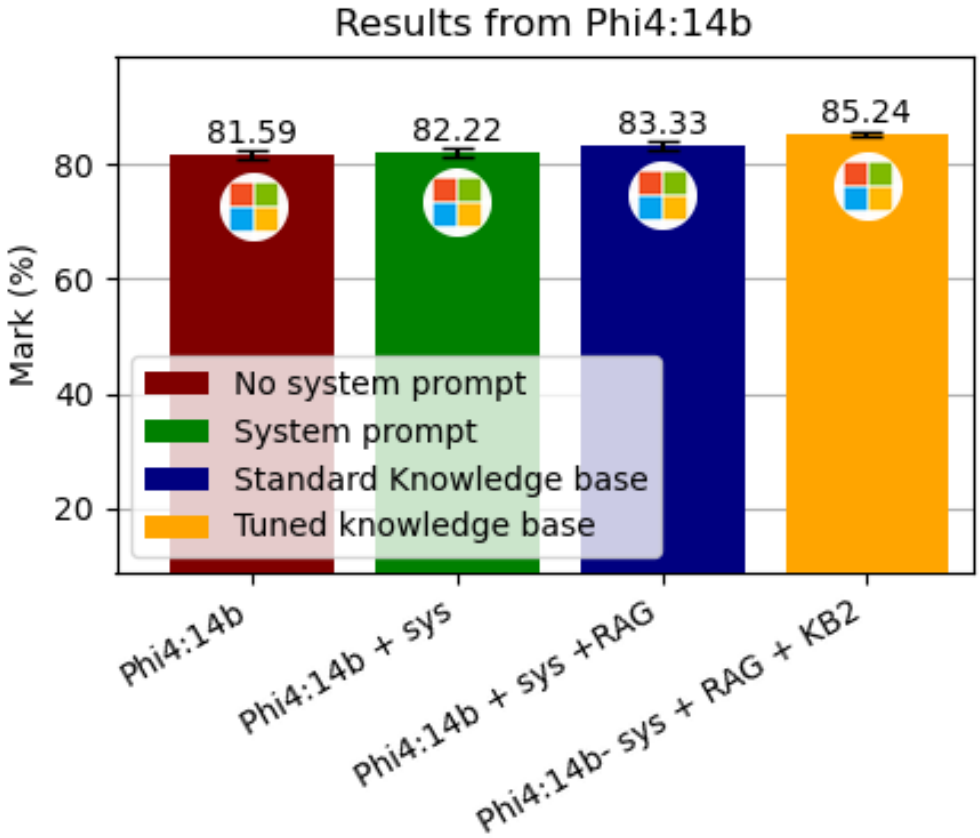


Figure 6.7: Bar chart showing the performance of the Microsoft Phi4:14b model against the cycling aerodynamics MCQ sheet.

Qwen Models

The performance difference between Qwen2.5:32b and Qwen2.5:7b, shown in Figure 6.9, highlights the combined influence of model size and a tuned knowledge base. The 32 billion parameter Qwen2.5:32b model clearly surpasses its 7 billion parameter counterpart, achieving an average score of $(88.73 \pm 0.29)\%$ compared with $(82.7 \pm 0.55)\%$. The Shapiro–Wilk p-values (0.004 vs. 0.01) indicate that the score distributions for both models deviate from normality, although this does not reflect performance quality. More importantly, the substantially lower coefficient of variation for Qwen2.5:32b (1.02 vs. 8.82) demonstrates its greater consistency

and reduced variability. Although both models employ similar techniques, the larger parameter count and refined knowledge base with Qwen2.5:32b contribute to its superior accuracy and stability. These findings suggest that for tasks requiring higher precision and robustness, larger models with optimised knowledge bases, such as Qwen2.5:32b, offer a clear advantage.

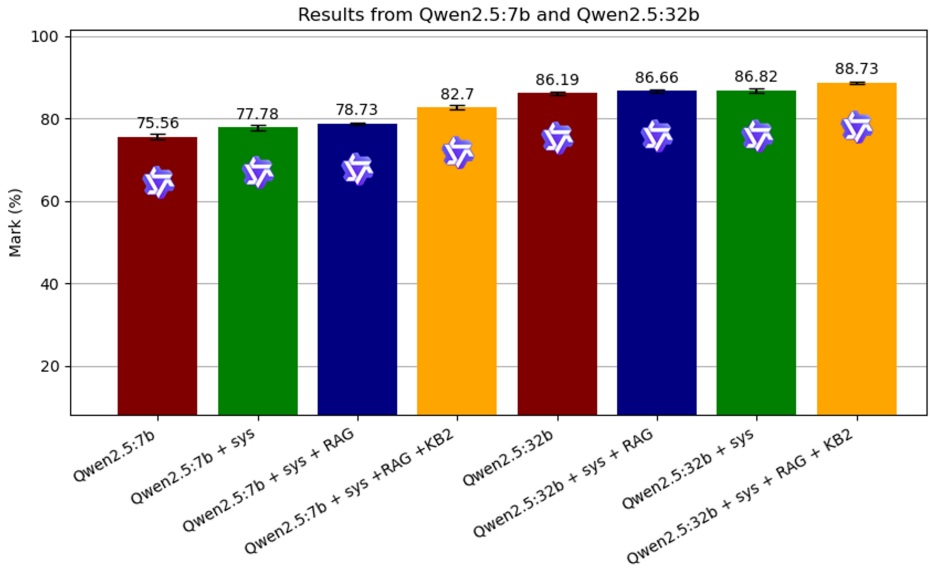


Figure 6.8: Bar chart showing the performance of the Qwen2.5:7b and Qwen2.5:32b model from Alibaba against the cycling aerodynamics MCQ sheet

OpenAI Models

As can be seen in Figure 7, the best OpenAI model is ChatGPT 4o with a system prompt providing an average mark of $(\mathbf{90} \pm \mathbf{0.41})\%$ a p-value of 0.04 and a coefficient of variation of 1.45.

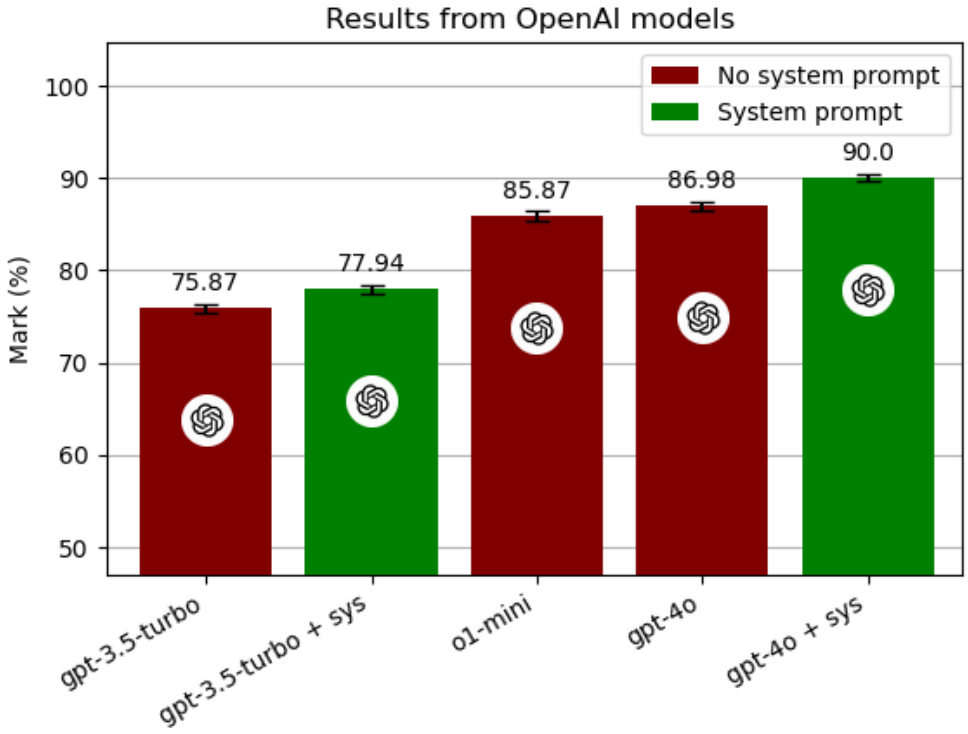


Figure 6.9: Bar chart showing the performance of three OpenAI models with and without a system prompt and their respective performance against the Cycling aerodynamics MCQ sheet

The Best Performing Models

From Figure 6.10 the best performing models in this study can be observed. Notably, the optimal performing model in this investigation is ChatGPT, achieving a score of $(90.00 \pm 0.41)\%$. This is closely followed by Qwen2.5:32b, which registered a performance of $(88.73 \pm 0.29)\%$. This outcome indicates that the superior model evaluated in this study is a closed-source model. However, given that the leading open-source model trails by less than 1.5% in performance, their capabilities can be considered statistically comparable within the context of this study.

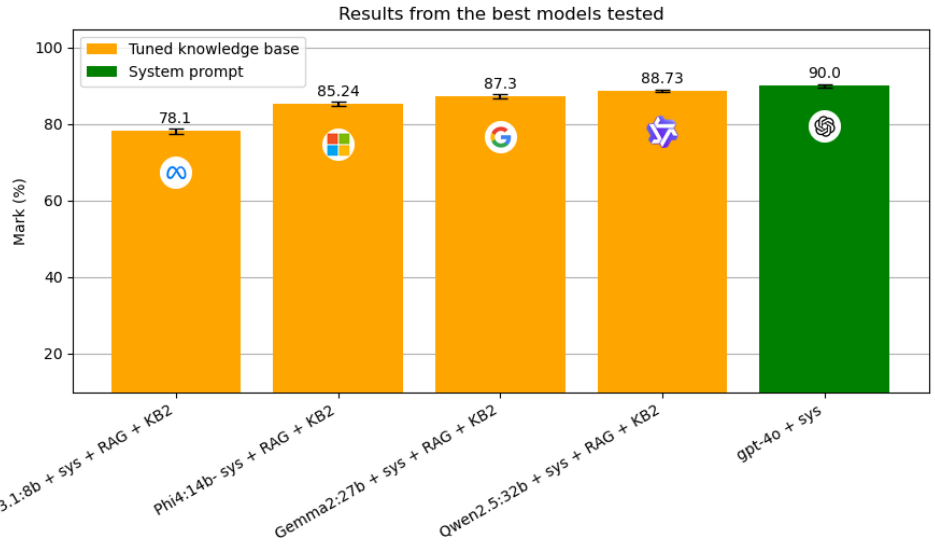


Figure 6.10: Bar chart of the best performing models in this study

Expert Performance

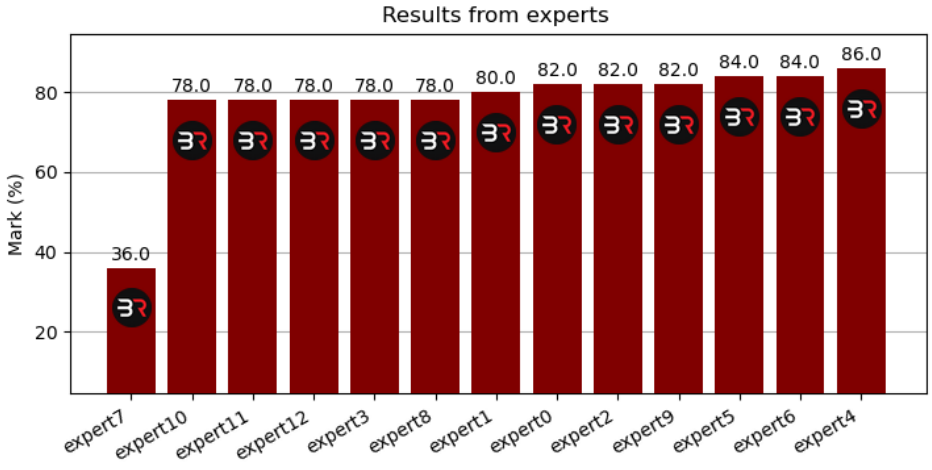


Figure 6.11: Bar chart showing the performance of all the experts in the study.

In Figure 6.11, the range of marks achieved by the experts spanned 8 marks, excluding the identified outlier exhibiting a score of 36%. The mean expert score is $(\mathbf{80.83 \pm 0.83})$ %, with the highest individual expert score reaching 86%. This data suggests that the assessment questions possessed an appropriate level of difficulty for domain experts. As a comparison, the top-performing LLM was evaluated using the same 50 questions as the expert panel.

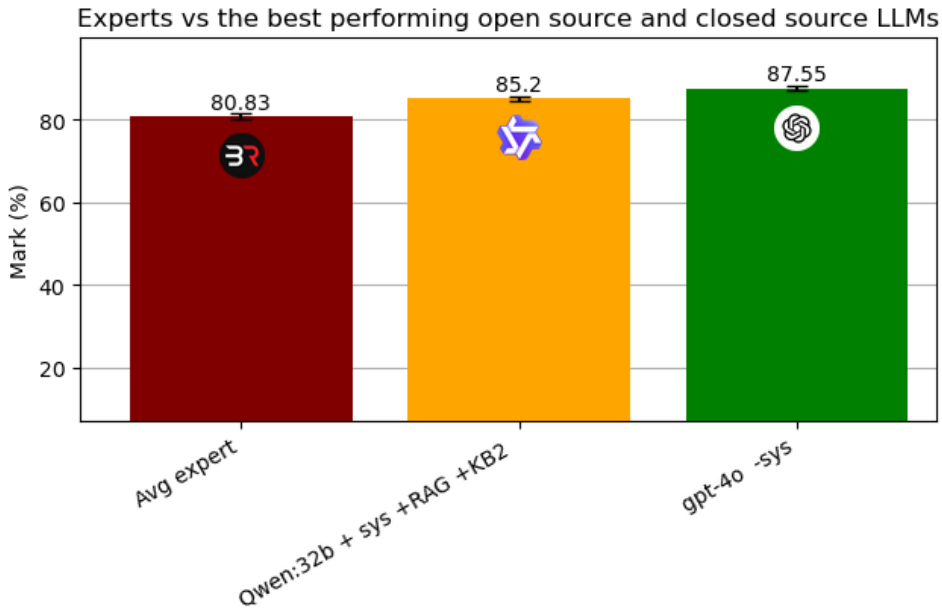


Figure 6.12: Bar chart showing the performance of all the experts in the study.

6.5 Discussion

The results obtained in Figures 6.5 to 6.7 and 6.11 all show assortments of language models with their respective configurations; Without a system prompt, with a system prompt, with a limited RAG knowledge base and with a more in-depth RAG knowledge base. As a general trend, the results are somewhat expected. For the majority of the models, as the number of parameters increased, the mark obtained also increased, supported by studies such as those carried out by Lee et al.

(2021b). Differences then started to appear as the different system configurations were added. Also, as a system prompt was added, the performance of the models generally did increase, with the exception of **llama3.1:8b** and **Gemma2:9b**. These results were not expected. Without a system prompt **llama3.1:8b** obtained a mark of $(73.65 \pm 0.79)\%$ whereas with a system prompt it obtained a mark of $(73.34 \pm 1.27)\%$, alongside this the marks awarded for **llama3.1:8b** showed greater variation as can be seen in the associated error which was generally not the case for the other models. As the marks from **llama3.1:8b** had such a large standard deviation in comparison to those without a system prompt, it can be concluded that the results are within the noise. **Gemma2:9b** also showed a reduction in performance when a system prompt was added, again, this is contrary to what was hypothesized and in this case with a mark of $(69.68 \pm 0.65)\%$ with the system prompt and $(71.11 \pm 0.85)\%$ without a system prompt this is comfortably outside of the variation of the marks obtained for both configurations.

Another unexpected result obtained was the fact that some models with the limited RAG document store performed worse than with a system prompt alone. Interestingly, this was with the **Qwen:32b** model with an average mark of $(86.66 \pm 0.35)\%$ with RAG compared to a mark of $(86.82 \pm 0.53)\%$ without RAG and just a system prompt. The RAG document store clearly reduced variation in the answers, keeping this model configuration more focused; however, it still led to a worse performance than the model configuration without the document store. This could be because the model was focused on some false positive responses, leading to the low variation and reduction in performance. Some studies such as one produced by Yu et al. (2024) explores the degradation of Language Model outputs when RAG is implemented in the response pipeline.

All of the models with the enhanced document store performed better than all of the other configurations. This is an expected result, and a notable result is that obtained from **llama3.2:3b**. This model shows a clear increase in performance

when compared to its counterparts in other configurations. The variation of marks is greater than that of the **llama3.1:8b** model; however, it is an interesting result given the fact that this small model clearly outperforms models more than double its parameter size. There is a trend within the Language Modelling fraternity to optimise smaller models, with some studies, such as Borgeaud et al. (2022), showing compatible performance between small models when compared to larger ones after implementing a retrieval-based method in their data pipeline.

Figure 6.12 illustrates that the best performing Large Language Models (LLMs) consistently outperform domain experts in knowledge retrieval, as assessed by a common set of 50 questions. While the LLMs demonstrated superior performance in this study, it is plausible that this advantage is partly attributable to their inherent, general-purpose question-answering capabilities, rather than exclusively to their domain-specific knowledge and comprehension. This represents a potential confound in the present work, although its estimated effect is considered weak. An improvement to this methodological approach would involve augmenting the number of questions and transitioning towards a more open-ended question format, as opposed to the current closed multiple-choice structure. Such a modification would necessitate the application of advanced comparison techniques, such as BLEU and ROUGE scores, which have been previously employed in relevant studies (Chatoui and Ata 2021).

The findings of this study reveal a trend where model performance is correlated to both the number of parameters within the model architecture and the quality of the provided dataset or knowledge base. This investigation demonstrates the feasibility of developing a high-performance LLM through the creation of a unique Multiple Choice Question (MCQ) sheet specifically tailored to the domain of aerodynamics in cycling. However, the reported results could be further enhanced by increasing the cohort of recruited experts and expanding the number of questions within the MCQ sheet. Future research endeavours could explore the

application of lightweight fine-tuning techniques, such as Low-Rank Adaptation (LoRA) (Qin et al. 2024), which could potentially enable an open-source model to achieve superior accuracy with minimal computational resource expenditure.

As these models cultivate a more profound understanding of domain fundamentals, the integration of their outputs with traditional statistical methodologies (e.g., standard deviation, coefficient of variation) could facilitate a more comprehensive interpretation and deeper understanding of aerodynamic performance. For instance, in scenarios where collected aerodynamic data exhibits substantial variability in the coefficient of drag area (CdA) over a testing session, concurrently with a high frequency of saddle shuffles, the LLM could analyse these concomitant trends and provide insights linking the two phenomena, building upon previous research indicating a potential impact of shuffling on performance (Barnes, Hopker and Gibson 2023). For subsequent investigations, a domain-specific LLM, akin to those examined in this study, could be instrumental in a detailed inquiry into the aerodynamic optimisation of cycling body position, equipment, and clothing, both with and without the direct assistance of the LLM as an expert consultant.

6.6 Conclusions

This study demonstrates that both open-source and proprietary LLMs possess the capability to retrieve information within the domain of cycling aerodynamics at a proficiency level comparable to, and in this study, superior to, that of a human expert, particularly when integrated with Retrieval Augmented Generation (RAG). However, it is acknowledged that certain features pertaining to the question style may confer a marginal advantage to the LLM. Furthermore, the observed performance parity between open-source and closed-source LLMs in this investigation suggests that the trade-off for employing open-source LLMs in an in-house system is only marginal.

Key Point Summary

- A novel ground truth dataset was created for cycling aerodynamics with the support of a cohort of 13 industry and academic experts.
- Applying techniques that require minimal computational resources, such as retrieval augmented generation (RAG), was shown to perform at, and in this study, outperform the human experts.
- This study found that Qwen2.5:32b was the best performing open source language model (after being enhanced with RAG), and the best model overall was ChatGPT 4o. Both of these models outperformed the average human expert in the study.

Chapter 7

Conclusion

7.1 Thesis Aim and Context

Throughout this thesis, machine learning, computer vision, and artificial intelligence techniques were applied within the context of cycling and cycling aerodynamics. The overarching goal was to show that these techniques can provide insight into how riders behave on the bike, while also lowering the barrier to accessing professional-level individualised feedback. This thesis demonstrates that such approaches are effective in cycling and cycling aerodynamics, which suggests that they may be applicable across many other fields as well.

The first chapter addressed this goal by investigating the impact of shuffling during cycling and by validating how the Body Rocket system determines a rider's position on the saddle. The second chapter also used the Body Rocket system to monitor rider position, enabling users to understand their position at any moment and train to maintain aerodynamic positions for optimal performance. Chapter three moved away from the Body Rocket system and showed that a fine-tuned YOLOv8 model can accurately determine joint locations of a rider in the sagittal plane and tucked positions. This result indicates that riders and bike fitters may

eventually move beyond physical measurement systems and adopt more remote or automated methods for bike fitting. The final study in this thesis has the potential to tie all of these elements together as it investigates the application of Language Modelling to create a cycling aerodynamics artificial intelligence expert.

7.2 Summary of Key Findings

Chapter 3 - Shuffling

This chapter addressed a commonly discussed phenomenon within the cycling community, particularly during high-intensity efforts, known as shuffling. Ten experienced cyclists were recruited to ride in an indoor velodrome and complete a 4 km pursuit effort. Each bike was set up to the rider's preferred dimensions and equipped with the Body Rocket system.

To validate how the Body Rocket system determines a rider's position on the saddle, a bespoke computer vision-based infrared camera was developed to monitor the rear of the rider during the pursuit effort. Following development, validation and calibration, the camera was deemed sufficiently accurate and was paired with the Body Rocket system for five riders. The system's measurements were considered reliable after comparing Body Rocket data with the camera outputs for four of these riders; however, for one rider, severe data loss occurred, which could not be recovered because the camera system was operating out of sync.

Once validated, a further five riders completed the same 4 km pursuit protocol. The study found that the impact of shuffling on a rider's C_dA and power output was negligible for most participants. However, some riders experienced clear improvements in C_dA and power during periods of shuffling, while others showed the opposite trend. This indicates that shuffling is likely technique-dependent and

affects riders differently.

By validating the Body Rocket system's method for identifying saddle position, this study creates opportunities for future research that uses the system to further investigate the shuffling phenomenon.

Chapter 4 - Positional Classification

This chapter took inspiration from Chapter 2. Instead of asking the question 'Is it possible to determine the position of a rider on the saddle through the use of the Body Rocket system?', this chapter asks the extended question 'Can you determine the position of someone as a whole on the bike, through the use of the Body Rocket system?'. The logic of this question is that the Body Rocket system is a suite of sensors at the contact points of the bike, and therefore, if you can take the weight distribution data alongside all the moments on the sensors, will it be possible to classify the position of a rider on the bike if some basic machine learning techniques are applied? This study first aimed to determine what model could be suitable for this task, then asked the question of how the best model performs on data collected from the Wind Tunnel and, by extension, how the model performs on data collected in outdoor conditions. This study investigated the Gaussian Mixture Model (GMM), Support Vector Machine (SVM) and Gradient Boosted Machine (GBM) as candidates for this classification task. Four different covariance types were tested with the GMM, four different types of SVM were tested, and a single type of GBM was tested. The GMM showed promising performance, and the SVM further showed an increase in performance; however, the best performing model was the GBM with 100% accuracy, classifying the four distinct positions of the rider on the bike in both the Wind Tunnel and outdoor conditions after a Modal Moving Average was applied. This study shows promise to be of significant impact within the cycling fraternity. As if the position of a rider on a bike can

be known without the use of additional sensors adhered to a rider on the bike, they will then be free to wear whatever clothing they wish and train to specific aerodynamic positions.

Chapter 5 - Pose Estimation in Cycling

This chapter built on the work from Chapter 2 on positional classification by aiming to obtain a continuous representation of a rider's position on the bike through computer vision techniques. The existing literature contains limited research on the capabilities of YOLO v8 and pose estimation models for riders in time trial positions, and there is also little work examining the effect of fine-tuning YOLO v8 to improve performance in this context.

Three athletes were recruited to ride a static bike in a state-of-the-art motion capture studio. Each rider used three different helmets and three different clothing setups, and they were instructed to maintain eight positions in total, four in road cycling postures and four in time trial postures.

This study found that the out-of-the-box YOLOv8 model was not suitable for determining joint locations in the sagittal plane, particularly when the ankle reached the bottom of the pedal stroke. It also found that fine-tuning over 10 epochs led to signs of overfitting. In contrast, fine-tuning over 5 epochs yielded a substantial improvement relative to the baseline model and generalised better than the 10-epoch model.

These results, similar to those in Chapter 2, show that computer vision pose models, specifically a fine-tuned YOLOv8, can provide accurate joint location data for both road and time trial cycling without requiring physical sensors or markers. Not only is this application novel for time trial positions, but it also offers greater freedom and flexibility for riders and practitioners, enabling possibilities such as remote assistance, rapid feedback, and even the potential for individuals

to perform their own bike fits at home.

Chapter 6 - Language Modelling in Cycling

Research Chapters 3-5 were primarily concerned with detecting on-bike position. The final chapter, however, focuses on providing individualised feedback relating to position adjustments alongside other areas of cycling and cycling aerodynamics, with the overarching goal to increase the accessibility of expert-level knowledge within this area. This chapter takes advantage of one of the newest and fastest-paced areas of research at the time of writing, language modelling. This study aimed to evaluate whether, with relatively few resources, if a local open source language model could perform as well as a human expert and to compare its performance to commercially available models. After recruiting 13 subject matter experts in cycling aerodynamics, bike fit and physiology, a ground truth question and answer sheet was created. This provided the capability to evaluate the performance of all the models and the human experts. This study found that the average human expert obtained a score of $(80.83 \pm 0.83)\%$ whereas the best open source model was Qwen2.5:32b with a score of $(85.2 \pm 0.51)\%$ (after being enhanced with Retrieval Augmented Generation), and the best model overall was ChatGPT 4o with a score of $(87.55 \pm 0.57)\%$. This study shows a real shift in the accessibility of expert-level data, especially in tandem with the performance of the open-source models, and shows that this technology can be accessible for all.

7.3 Future Work and Closing Statement

From monitoring shuffling on the saddle, identifying rider position on the bike, both in outdoor and static conditions, alongside the applications of a Large Language Model coach, there are a number of areas of future work where these proof-of-concept studies can be applied in case studies and real-world applications.

Future Shuffling Studies

Within the investigation of shuffling conducted in this thesis, it was concluded that the effect of shuffling on a rider's power output and C_dA was rider-dependent. Future work could explore this dependency further by investigating intra-rider variability through changes to different aspects of the bike setup, since the configurations used in this study reflected each rider's personal preferences.

Potential future studies could examine adjustments such as saddle setback, saddle height, or reductions in friction at the rider-saddle interface. Reducing friction on the saddle could be achieved through the application of a lubricant to the saddle surface. A study with a cohort of similar size to that presented in this thesis, combined with multiple repetitions for each rider and configuration, such as lubricated saddle surfaces or larger saddle angles, could provide compelling results. Such work would help determine whether shuffling is primarily rider-dependent or whether elements of biomechanical fit contribute to rider performance while shuffling.

Future Positional Classification Studies

In the positional classification chapter, it was established that, using the Body Rocket system and machine learning techniques, the position of a rider on the bike can be determined. As a result, further investigations could include applying this technology in a live setting to ensure that the rider maintains the correct

position. As the models used in this study were specifically intended to be low-latency, data could be provided to an athlete live, and an investigation could be carried out where the rider is provided live feedback on their position and how they are holding it, and their performance (C_dA gains or losses) can be compared between feedback and no feedback.

This would act as a case study of how this technology can be used in real-world applications as well as being a research investigation into whether continuous positional feedback can benefit a rider's performance.

Future Pose Estimation Studies

Within the pose estimation chapter, it was established that it was possible to determine the position of a rider on the bike, more specifically, their joint locations. To further the applications of this research, the new fine-tuned YOLO models could be used within bike fitting applications where a bike fitter is working remotely, and their client is in a different location from the bike fitting studio.

Further studies to aid the development of these models could include comparing the YOLOv8 fine-tuned model with other commercial offerings. The motion capture studio used in this study was state-of-the-art and can still be used as the ground truth, further studies could compare both the YOLOv8 fine-tuned model and a commercial solution, which is used in real-world bike fitting applications. This would help evaluate whether the fine-tuned YOLOv8 model is truly accurate enough for bike fitters to be using and potentially prove that it is a viable alternative.

Future Language Modelling Studies

With the knowledge retrieval of these domain-specific models evaluated in this thesis, further areas for investigation could include a case study in which these models are used in the field and compared to a human expert. Examples include an athlete with access to this model asking questions and obtaining advice prior to starting a ride, such as requesting a suitable testing protocol. With a protocol provided, the athlete could carry out the tests, the LLM could look at the results from the testing and recommend further tests. The potential C_dA gains can then be directly compared to a human expert.

There are a number of logical next steps that can be followed as a result of the findings obtained within this thesis. Further work inspired by shuffling would include increasing the number of participants and the selection criteria, allowing for both male and female cyclists. Additionally, studies can now be constructed with the Body Rocket system where no validation measure will be needed. As a result, the Body Rocket system could be used in a study where the contact points of the bike are iteratively adjusted, therefore investigating the impact of shuffling for the same rider in different configurations. Additional applications and work could use the Body Rocket system itself as a bike fitting tool (at the time of writing, it is available).

Future work with the second study is also clear. Similar to the first study, a larger cohort could be recruited, alongside a greater variety of positions that could be investigated to test the resolution of these positional classification capabilities. Additionally, the work from the second study and the third study could be combined. Now it has been established that YOLOv8 fine-tuned to 5 epochs is sufficiently accurate, it will be easier to collect data on the joint locations of a rider on the bike and potentially conduct research combining this joint data with the data from the Body Rocket system.

With respect to the final study, low-intensity fine-tuning methods like Low Rank Adaptation could be implemented on an open-source model to see if performance is greatly improved. This study could also be extended with additional models that are released, and if paired with simple statistical measures such as standard deviation and mean, then one of these models could perform some basic data analysis, allowing for better prescribed aerodynamic tests.

The research in this thesis yields novel findings in cycling aerodynamics through machine learning, from insights into the impact of shuffling on cycling performance to the creation of a language model–based chatbot that outperforms human experts. Ultimately, it democratises high-level aerodynamic insight for performance cycling.

Bibliography

(2012). Helmets for pedal cyclists and for users of skateboards and roller skates – BS en 1078:2012 + a1:2012. Accessed: 2025-11-14.

Abbott, I. H. et al. (2012). *Theory of wing sections: including a summary of airfoil data*. Courier Corporation.

Aerosensor (2025). Aerobody. `https://aerosensor.tech/products/aerobody`, accessed: 2025-01-17.

Ahmadi, R. et al. (2024). A machine learning approach for predicting pedaling force profile in cycling. *Sensors*, 24(19).

Alam, F., Chowdhury, H. and Moria, H. (2019). A review on aerodynamics and hydrodynamics in sports. *Energy Procedia*, 160, pp. 798–805.

Alam, F., Chowdhury, H., Wei, H. Z., Mustary, I. and Zimmer, G. (2014). Aerodynamics of ribbed bicycle racing helmets. *Procedia Engineering*, 72, pp. 691–696, the Engineering of Sport 10.

AllAboutAI-YT (2025). easy-local-rag: Supereasy 100% local rag with ollama + email rag. `https://github.com/AllAboutAI-YT/easy-local-rag`, accessed: 2025-12-02.

Allgaier, J. and Pryss, R. (2024). Cross-validation visualized: a narrative guide

to advanced methods. *Machine learning and knowledge extraction*, 6(2), pp. 1378–1388.

Androniceanu, A., Kinnunen, J. and Georgescu, I. (2020). E-government clusters in the eu based on the gaussian mixture models. *Administratie si Management Public*, (35), pp. 6–20.

Anello, M., Daverio, M. S. and Di Rocco, F. (2022). Genetics of coat color and fiber production traits in llamas and alpacas. *Animal Frontiers*, 12(4), pp. 78–86.

Anthropic (2024). Claude: Ai language model. `https://claude.ai`, accessed: 2025-07-24.

Asai, T. and Hong, S. (2021). Aerodynamics of the newly approved football for the english premier league 2020–21 season. *Scientific Reports*, 11(1), p. 9578.

Atherton, T. J. and Kerbyson, D. J. (1993). The coherent circle hough transform. In *BMVC*, pp. 1–10.

Baldissera, P. (2017). Proposal of a coast-down model including speed-dependent coefficients for the retarding forces. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, 231(2), pp. 154–163, `https://doi.org/10.1177/1754337116658587`.

Barnaby, G. C., Yon, J. and Burgess, S. (2020). Sprocket size optimisation for derailleur racing bicycles. *Journal of Science and Cycling*.

Barnes, C., Hopker, J. and Gibson, S. (2023). To shuffle or not to shuffle. *Journal of Science and Cycling*, 12(2), pp. 28–30.

Barnes, C., Hopker, J. and Gibson, S. (2024). Artificial intelligence in cycling–live positional tracking using on-bike aerodynamic sensors. *Journal of Science and Cycling*, 13(2), pp. 41–48.

Belagiannis, V. et al. (2014). 3d pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1669–1676.

Benfenati, D., De Filippis, G. M., Rinaldi, A. M., Russo, C. and Tommasino, C. (2024). A retrieval-augmented generation application for question-answering in nutrigenetics domain. *Procedia Computer Science*, 246, pp. 586–595.

Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb), pp. 1137–1155.

Bini, R. R., Hume, P. A. and Crofta, J. L. (2011). Effects of saddle height on pedal force effectiveness. *Procedia engineering*, 13, pp. 51–55.

Bini, R. R. et al. (2010). Pedaling cadence effects on joint mechanical work during cycling. *Isokinetics and Exercise Science*, 18(1), pp. 7–13, `https://doi.org/10.3233/IES-2010-0361`.

Blair, K., Sidelko, S. and Estivalet, M. (2009). *Aerodynamic Performance Of Cycling Time Trial Helmets (P76)*. pp. 371–377.

Blanquero, R., Carrizosa, E., Ramírez-Cobo, P. and Sillero-Denamiel, M. R. (2021). Variable selection for naïve bayes classification. *Computers & Operations Research*, 135, p. 105456.

Blocken, B. et al. (2018). Aerodynamic drag in cycling pelotons: New insights by cfd simulation and wind tunnel testing. *Journal of Wind Engineering and Industrial Aerodynamics*, 179, pp. 319–337.

Blocken, B., van Druenen, T., Toparlar, Y. and Andrianne, T. (2019). Cfd analysis of an exceptional cyclist sprint position. *Sports Engineering*, 22(1), p. 10.

Body Rocket (2026). Shop our products | buy your power meter. `https://www.bodyrocket.cc/product/power-meter`, accessed: 2026-06-01.

Borgeaud, S. et al. (2022). Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, PMLR, pp. 2206–2240.

Brooke, A. and Broadbent, B. (2020). Is the saddle always to blame? a discussion of potential causative factors of saddle discomfort. *Journal of Pelvic, Obstetric and Gynaecological Physiotherapy*.

Brown, T. et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, pp. 1877–1901.

Brownlie, L. et al. (2009). Streamlining the time trial apparel of cyclists: the nike swift spin project. *Sports technology*, 2(1-2), pp. 53–60.

Brownlie, L. et al. (2010). The wind-averaged aerodynamic drag of competitive time trial cycling helmets. *Procedia Engineering*, 2(2), pp. 2419–2424, the Engineering of Sport 8 - Engineering Emotion.

Burgess, S. (1998). Improving cycling performance with large sprockets. *Sports Engineering*, 1(2), pp. 107–113.

Candau, R. B. et al. (1999). Simplified deceleration method for assessment of resistive forces in cycling. *Medicine and science in sports and exercise*, 31, pp. 1441–1447.

Cao, Z., Simon, T., Wei, S.-E. and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7291–7299.

Catesby Tunnel (2025). Catesby tunnel: Full-scale aerodynamic and vehicle testing facility. Information page for the Catesby vehicle testing tunnel in Northamptonshire, UK.

Chabroux, V., Caroline, B. and Favier, D. (2012). Aerodynamics of cyclist posture, bicycle and helmet characteristics in time trial stage. *Journal of applied biomechanics*, 28, pp. 317–23.

Chatoui, H. and Ata, O. (2021). Automated evaluation of the virtual assistant in bleu and rouge scores. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–6.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Cho, A., Kim, J., Lee, S. and Kee, C. (2011). Wind estimation and airspeed calibration using a uav with a single-antenna gps receiver and pitot tube. *IEEE transactions on aerospace and electronic systems*, 47(1), pp. 109–117.

Chowdhery, A. et al. (2023). Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240), pp. 1–113.

Cohen, C., Brunet, E., Roy, J. and Clanet, C. (2021). Physics of road cycling and the three jerseys problem. *Journal of Fluid Mechanics*, 914, p. A38.

Committee, U. U. . O. O. (2024). *Guidelines Regarding Authorised Identifications — Cycling, Paris 2024 Olympic Games*. Union Cycliste Internationale, accessed: 14-11-2025.

Cook, A. and Karakuş, O. (2024). Llm-commentator: Novel fine-tuning strategies of large language models for automatic commentary generation using football event data. *Knowledge-Based Systems*, 300, p. 112219.

Crouch, T. et al. (2021). A wind-tunnel case study: Increasing road cycling velocity by adopting an aerodynamically improved sprint position. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, 235(4), pp. 301–309.

Crouch, T. N., Burton, D., LaBry, Z. A. and Blair, K. B. (2017). Riding against the wind: a review of competition cycling aerodynamics. *Sports Engineering*, 20(2), pp. 81–110.

Danek, T., Slawinski, M. A. and Stanoev, T. (2020). On modelling bicycle power-meter measurements: Part i. estimating effects of air, rolling and drivetrain resistance. `2005.04229`.

de Smit, G., Yeleshetty, D., Havinga, P. J. and Huang, Y. (2024). Predicting turn maneuvers of cyclists using bicycle-mounted imu with cnn-lstm. In *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, IEEE, pp. 587–592.

Debraux, P., Grappe, F., Manolova, A. V. and Bertucci, W. (2011). Aerodynamic drag in cycling: methods of assessment. *Sports biomechanics*, 10(3), pp. 197–218.

Dedgaonkar, S. et al. (2025). Real-time football analysis system using yolo and opencv. In *2025 International Conference on Intelligent Systems and Computational Networks (ICISCN)*, IEEE, pp. 1–6.

Defraeye, T., Blocken, B., Koninckx, E., Hespel, P. and Carmeliet, J. (2010). Aerodynamic study of different cyclist positions: Cfd analysis and full-scale wind-tunnel tests. *Journal of biomechanics*, 43(7), pp. 1262–1268.

DeGolier, E., Cibinel, A. and Barnes, C. (2024). Validation of body rocket on-bike

wind-tunnel technology: Drag measurement accuracy and aerodynamic gains sensitivity. *Journal of Science and Cycling*, 13(2), pp. 22–28.

Desai, N. P., Etemad, A. and Greenspan, M. (2025). Cyclecrash: A dataset of bicycle collision videos for collision prediction and analysis. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, pp. 6688–6698.

Dowd, T., Cavanaugh, P. and Mansson, J.-A. (2024). A novel model for bicycle drivetrain efficiency.

Du, K.-L., Leung, C.-S., Mow, W. H. and Swamy, M. N. S. (2022). Perceptron: Learning, generalization, model selection, fault tolerance, and role in the deep learning era. *Mathematics*, 10(24).

Dyer, B. and Disley, B. X. (2018). Validation of the virtual elevation field test method when assessing the aerodynamics of para-cyclists with a uni-lateral trans-tibial amputation. *Disability and rehabilitation: Assistive technology*, 13(2), pp. 107–111.

D'Antonio, E., Taborri, J., Mileti, I., Rossi, S. and Patané, F. (2021). Validation of a 3d markerless system for gait analysis based on openpose and two rgb webcams. *IEEE Sensors Journal*, 21(15), pp. 17064–17075.

Evans, S., James, D., Rowlands, D. and Lee, J. B. (2021a). Exploring the effects of trunk acceleration on saddle position and the drag coefficient.

Evans, S. A., James, D. A., Rowlands, D. and Lee, J. B. (2021b). Evaluation of accelerometer-derived data in the context of cycling cadence and saddle height changes in triathlon. *Sensors*, 21(3).

Ferrer-Roca, V. et al. (2014). Acute effects of small changes in bicycle saddle

height on gross efficiency and lower limb kinematics. *The Journal of Strength & Conditioning Research*, 28(3), pp. 784–791.

Fintelman, D., Sterling, M., Hemida, H. and Li, F. (2015). The effect of time trial cycling position on physiological and aerodynamic variables. *Journal of sports sciences*, 33(16), pp. 1730–1737.

fitkitsystems (2025). Velometrik smartcover bt saddle pressure mapping system. `https://fitkitsystems.com/product/smartcover-bt-saddle-pressure-mapping/`, accessed: 2025-06-09.

Flores-Calero, M. et al. (2024). Traffic sign detection and recognition using yolo object detection algorithm: A systematic review. *Mathematics*, 12(2).

Forte, P. et al. (2020). Analysis of a normal and aero helmet on an elite cyclist in the dropped position. *AIMS Biophys*, 7(1), pp. 54–64.

Foundation, R. P. (2024a). Raspberry pi 3 model b. Accessed: 2024-11-28.

Foundation, R. P. (2024b). Raspberry pi zero 2 w. Accessed: 2024-11-28.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232.

Fukushima, T., Blauberger, P., Guedes Russomanno, T. and Lames, M. (2024). The potential of human pose estimation for motion capture in sports: a validation study. 27(1).

Garcia-Lopez, J., Ogueta-Alday, A., Larrazabal, J. and Rodríguez-Marroyo, J. (2014). The use of velodrome tests to evaluate aerodynamic drag in professional cyclists. *International journal of sports medicine*, 35(05), pp. 451–455.

García-López, J. et al. (2008). Reference values and improvement of aerodynamic drag in professional cyclists. *Journal of Sports Sciences*, 26(3), pp. 277–286, pMID: 17943597, `https://doi.org/10.1080/02640410701501697`.

Garimella, R. et al. (2020). Estimating cycling aerodynamic performance using anthropometric measures. *Applied Sciences*, 10(23).

gebiomized (2025). Saddle pressure mapping. `https://gebiomized.de/en/products/measurement/gp-bike/`, accessed: 2025-06-09.

GitHub and OpenAI (2021). Github copilot: your ai pair programmer. GitHub, aI-powered code completion assistant integrated into IDEs like VS Code, JetBrains, Neovim.

Google (2024). Gemini: Ai language model. `https://gemini.google.com`, accessed: 2025-07-24.

Google (2025). New gemini summary cards now available in the gmail app on android and ios devices. Google Workspace Updates Blog, gemini adds "Summarize this email" and auto-generated summary cards at the top of extended email threads.

Goreham, J. A., MacLean, K. F. and Ladouceur, M. (2022). The validation of a low-cost inertial measurement unit system to quantify simple and complex upper-limb joint angles. *Journal of Biomechanics*, 134, p. 111000.

Grappe, F. et al. (1999). Influence of tyre pressure and vertical load on coefficient of rolling resistance and simulated cycling performance. *Ergonomics*, October 1, pp. 1361–1371.

Grau-Escolano, J., Bassolas, A. and Vicens, J. (2024). Cycling into the workshop: e-bike and m-bike mobility patterns for predictive maintenance in barcelona's bike-sharing system. *EPJ Data Science*, 13(1), p. 48.

Griffith, M. D. et al. (2014). Computational fluid dynamics study of the effect of leg position on cyclist aerodynamic drag. *Journal of Fluids Engineering*, 136(10), p. 101105.

Guido, R., Ferrisi, S., Lofaro, D. and Conforti, D. (2024). An overview on the advancements of support vector machine models in healthcare applications: a review. *Information*, 15(4), p. 235.

Hamann, J. (2018). Aeropod: Power and aerodynamic cda measurement for cyclists. `https://www.kickstarter.com/projects/powerpod/aeropod-power-and-aerodynamic-cda-measurement-for`, kickstarter campaign by Velocomp.

Hassan, Y. A. (2010). Computational fluid dynamics for natural circulation flows. In *ICTP School on Advanced Reactor Systems*, Trieste, Italy, lecture notes.

Henchoz, Y., Crivelli, G., Borrani, F. and Millet, G. P. (2010). A new method to measure rolling resistance in treadmill cycling. *Journal of Sports Sciences*, 28(10), pp. 1043–1046, pMID: 20686997, `https://doi.org/10.1080/02640414.2010.498483`.

HJC Helmets (2025). Hjc adwatt helmet – product page.

Holmes, J. et al. (2004). Aerodynamic forces and moments on cubes and flat plates, with applications to wind-borne debris. In *Summary Papers of the 5th International Colloquium on Bluff Body Aerodynamics and Applications*, pp. 103–106.

Hong, S. and Asai, T. (2021). Aerodynamics of cycling skinsuits focused on the surface shape of the arms. *Applied Sciences*, 11(5).

Hopker, J. G., Barnes, C., Fennell, C. R. and Gibson, S. (2025). Validity and reliability of an on-bike sensor system for the determination of aerodynamic drag in cycling. *European Journal of Sport Science*, 25(12), p. e70098.

HP Inc. (2025). Hp zbook power 15.6 inch g10 a mobile workstation pc specifications. https://support.hp.com/hk-en/document/ish_8428016-8428060-16, accessed: 2025-05-20.

Huang, Y., Sun, B., Kan, H., Zhuang, J. and Qin, Z. (2019). Followmeup sports: New benchmark for 2d human keypoint recognition. 1911.08344.

Hub, S. S. E. (2025a). The cycling wind tunnel's features. urlhttps://silverstonesportshub.co.uk/the-features-of-the-cycling-wind-tunnel/, accessed: 2025-12-02.

Hub, S. S. E. (2025b). Pedalling efficiency rig: Mechanical efficiency optimisation. Accessed: 2025-11-06.

Ibrahim, M. (2024). Computer vision and statistical insights into cycling near miss dynamics. *Scientific Reports*, 14(1), p. 21151.

Janssens, B., Bogaert, M. and Verstockt, S. (2025). Large language models on race commentary: Towards granular data in cycling analytics. In U. Brefeld, J. Davis, J. Van Haaren and A. Zimmermann, eds., *Machine Learning and Data Mining for Sports Analytics*, Cham: Springer Nature Switzerland, pp. 14–25.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, Springer, pp. 137–142.

Jocher, G. et al. (2020). ultralytics/yolov5: v3. 0. *Zenodo*.

Jolles, J. W. (2021). Broad-scale applications of the raspberry pi: A review and guide for biologists. *Methods in Ecology and Evolution*, 12(9), pp. 1562–1579.

Jongerius, N., Wainwright, B., Walker, J. and Bissas, A. (2022). The biomechanics of maintaining effective force application across cycling positions. *Journal of Biomechanics*, 138, p. 111103.

Kakavand, R., Ahmadi, R., Parsaei, A., Edwards, W. B. and Komeili, A. (2024). Opencap markerless motion capture estimation of lower extremity kinematics and dynamics in cycling. *arXiv preprint arXiv:240903766*.

Kakavand, R., Ahmadi, R., Parsaei, A., Brent Edwards, W. and Komeili, A. (2025). Comparison of kinematics and kinetics between opencap and a marker-based motion capture system in cycling. *Computers in Biology and Medicine*, 192, p. 110295.

Kaplan, J. et al. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:200108361*.

Karrach, L. and Pivarčiová, E. (2023). Using different types of artificial neural networks to classify 2D matrix codes and their rotations-a comparative study. *J Imaging*, 9(9).

Ke, Y. H. et al. (2025). Retrieval augmented generation for 10 large language models and its generalizability in assessing medical fitness. *npj Digital Medicine*, 8(1), p. 187.

Kholkine, L. et al. (2021). A learn-to-rank approach for predicting road cycling race outcomes. *Frontiers in sports and active living*, 3, p. 714107.

Kim, J. E., Yu, J., Ryu, J.-H., Lee, J.-H. and Kim, T.-W. (2021). Assessment of regional drought vulnerability and risk using principal component analysis and a gaussian mixture model. *Natural Hazards*, 109(1), pp. 707–724.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2), pp. 307–319.

Kitajima, E., Sato, T., Kurata, K. and Miyata, R. (2023). Automatic feature selection for performing unit 2 of vault in wheel gymnastics. *Plos one*, 18(6), p. e0287095.

Kozlov, K., Egorov, A. and Belogusev, V. (2017). Experimental evaluation of chain transmissions lubricants quality using a new method based on additional inertia moment use. *Procedia Engineering*, 206, pp. 617–623, international Conference on Industrial Engineering, ICIE 2017.

Kresevic, S. et al. (2024). Optimization of hepatological clinical guidelines interpretation by large language models: a retrieval augmented generation-based framework. *npj Digital Medicine*, 7(1), p. 102.

Kyle, C. R. and Burke, E. (1984). Improving the racing bicycle. *Mechanical engineering*, 106(9), pp. 34–45.

Lee, C. et al. (2021a). On the redundancy in the rank of neural network parameters and its controllability. *Applied Sciences*, 11(2), p. 725.

Lee, C. et al. (2021b). On the redundancy in the rank of neural network parameters and its controllability. *Applied Sciences*, 11(2).

Lee, S., Shakir, A., Koenig, D. and Lipp, J. (2024). Open source strikes bread – new fluffy embedding model. Accessed: 2025-12-02.

Lewis, P. et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33, pp. 9459–9474.

Li, M. et al. (2025). A benchmark for cycling close pass detection from video streams. *Transportation Research Part C: Emerging Technologies*, 174, p. 105112.

Lin, T.-Y. et al. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, Springer, pp. 740–755.

Liu, L. et al. (2020). Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2), pp. 261–318.

Lozano, A., Fleming, S. L., Chiang, C.-C. and Shah, N. (2023). Clinfo. ai: An open-source retrieval-augmented large language model system for answering medical questions using scientific literature. In *Pacific Symposium on Biocomputing 2024*, World Scientific, pp. 8–23.

Lu, W., Luu, R. K. and Buehler, M. J. (2025). Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities. *npj Computational Materials*, 11(1), p. 84.

Lukes, R., Hart, J. and Haake, S. (2012). An analytical model for track cycling. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, 226(2), pp. 143–151, https://doi.org/10.1177/1754337111433242.

Madhav, M. S. et al. (2022). The dome: a virtual reality apparatus for freely locomoting rodents. *Journal of neuroscience methods*, 368, p. 109336.

Maji, D., Nagori, S., Mathew, M. and Poddar, D. (2022). Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. 2204.06806.

Malizia, F. and Blocken, B. (2021). Cyclist aerodynamics through time: Better, faster, stronger. *Journal of Wind Engineering and Industrial Aerodynamics*, 214, p. 104673.

Marciotto, E. R. (2016). Classic bernoulli's principle derivation and its working hypotheses. *Physics Education*, 51(4), p. 045005.

Martin, J. C., Milliken, D. L., Cobb, J. E., McFadden, K. L. and Coggan, A. R. (1998). Validation of a mathematical model for road cycling power. *Journal of Applied Biomechanics*, 14(3), pp. 276 – 291.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), pp. 115–133.

McDonald, R., Holliday, W. and Swart, J. (2022). Muscle recruitment patterns and saddle pressure indexes with alterations in effective seat tube angle. *Sports Medicine and Health Science*, 4(1), pp. 29–37.

MET-Helmets (2025). Met drone wide body ii helmet.

Microsoft (2025). Get a summary of a conversation. Microsoft Support, "Summary by Copilot" appears at the top of the message thread; click it to get brief bullet-point summaries of the conversation.

Mikolov, T. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:13013781*.

Moore, S. R. et al. (2020). Foot strike angle prediction and pattern classification using loadsoltm wearable sensors: A comparison of machine learning techniques. *Sensors*, 20(23), p. 6737.

Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B. and Turaga, D. S. (2017). Learning feature engineering for classification. In *Ijcai*, vol. 17, pp. 2529–2535.

Navier, C. (1821). Sur les lois des mouvement des fluides, en ayant egard a l'adhesion des molecules. In *Annales de Chimie et de Physique*, vol. 19, Lavoisier Paris, France, p. 1821.

Neumann, A., Kirsten, E., Zafar, M. B. and Singh, J. (2025). Position is power: System prompts as a mechanism of bias in large language models (llms). In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pp. 573–598.

No, C. and Lee, M. (2025). Global and local feature communications with transformers for 3D human pose estimation. *Scientific Reports*, 15(1), p. 6624.

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, 24(12), pp. 1565–1567.

Notio Cycling Inc. (2025). Notio: Real-time aerodynamic measurement for cyclists. `https://notio.ai/`, accessed online.

novel.de (2025). Pressure mapping on bicycle saddle. `https://novel.de/products/pliance/bicycle-saddle-pressure-mapping/`, accessed: 2025-06-09.

Obradović, S. and Stančin, S. (2023). Inertial measurement units' reliability for measuring knee joint angle during road cycling. *Electronics*, 12(3).

Olivier, J. and Creighton, P. (2016). Bicycle injuries and helmet use: a systematic review and meta-analysis. *International Journal of Epidemiology*, 46(1), pp. 278–292, `https://academic.oup.com/ije/article-pdf/46/1/278/13733257/dyw153.pdf`.

OpenAI (2023). ChatGPT: Large language model (mar 14 version). `https://chat.openai.com`, accessed: 2025-07-24.

OpenAI (2025). Codex: an agentic coding assistant. OpenAI Blog, cloud-based coding agent (codex-1) that can write, debug, run tests, make pull requests.

Parallax (2025). Ping))) ultrasonic distance sensor (28015). `https://docs.rs-online.com/ca69/0900766b81382974.pdf`, accessed: 2025-06-09.

Peeters, T., Garimella, R. and Verwulgen, S. (2020). An indoor training bike to provide real-time feedback on the aerodynamic cycling position using frontal area calculations. *Proceedings of the 3DBODY TECH*.

Pinheiro, G. d. S., Jin, X., Costa, V. T. D. and Lames, M. (2022). Body pose estimation integrated with notational analysis: A new approach to analyze penalty kicks strategy in elite football. *Front Sports Act Living*, 4, p. 818556.

POC Sports (2025). Poc omne air mips – hydrogen white / fluorescent orange.

Polanco, A. P., Suárez, D. R. and Muñoz, L. E. (2017). Effect of body posture on comfort during cycling. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 58158, American Society of Mechanical Engineers, p. V003T01A012.

Qi, D., Tan, W., Yao, Q. and Liu, J. (2022). Yolo5face: Why reinventing a face detector. In *European Conference on Computer Vision*, Springer, pp. 228–244.

Qin, H. et al. (2024). Accurate lora-finetuning quantization of llms via information retention. *arXiv preprint arXiv:240205445*.

Qualisys (2025). *Qualisys Track Manager (QTM)*. `https://www.qualisys.com/software/qualisys-track-manager/`.

Ramos, G. et al. (2020). Fatigue evaluation through machine learning and a global fatigue descriptor. *Journal of healthcare engineering*, 2020(1), p. 6484129.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788.

Retül (2025). Bike fit | product matching. `https://www.retul.com/`, accessed: 2025-08-16.

Reynolds, D. A. et al. (2009). Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663).

Reynolds, O. (1883). An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Proceedings of the Royal Society of London*, 35(224-226), pp. 84–99.

ripa, T. S. et al. (2024). Bicycle theft detection by using yolov5s and deepsort algorithm. In *2024 International Conference on Recent Progresses in Science, Engineering and Technology (ICRPSET)*, pp. 1–5.

Rodrigues, I. R. et al. (2022). A framework for robotic arm pose estimation and movement prediction based on deep and extreme learning models. *arXiv preprint arXiv:220513994*.

Salamzadeh, A., Ebrahimi, P., Soleimani, M. and Fekete-Farkas, M. (2022). Grocery apps and consumer purchase behavior: application of gaussian mixture model and multi-layer perceptron algorithm. *Journal of Risk and Financial Management*, 15(10), p. 424.

Schmidt, C. W. et al. (2024). Tokenization is more than compression. *arXiv preprint arXiv:240218376*.

scikit-learn developers (2025a). *sklearn.svm.LinearSVC — scikit-learn 1.7.2 documentation*. Accessed: 2025-12-05.

scikit-learn developers (2025b). *sklearn.svm.SVC — scikit-learn 1.7.2 documentation*. Accessed: 2025-12-05.

Serrancoli, G. et al. (2020). Marker-less monitoring protocol to analyze biomechanical joint metrics during pedaling. *IEEE access*, 8, p. 122782–122790.

Seto, H. et al. (2022). Gradient boosting decision tree becomes more reliable than logistic regression in predicting probability for diabetes with big data. *Scientific reports*, 12(1), p. 15889.

SHAPIRO, S. S. and WILK, M. B. (1965). An analysis of variance test for normality (complete samples)†. *Biometrika*, 52(3-4), pp. 591–611, https://academic.oup.com/biomet/article-pdf/52/3-4/591/962907/52-3-4-591.pdf.

Shlens, J. (2014). A tutorial on principal component analysis. 1404.1100.

Shrestha, N. (2020). Detecting multicollinearity in regression analysis. *American Journal of Applied Mathematics and Statistics*, 8, pp. 39–42.

Silverstone Sports Engineering Hub (2022). Rolling resistance testing. https://silverstonesportshub.co.uk/2022/12/22/rolling-resistance/, accessed 17-08-2025.

Spicer, J. B. (2013). Effects of the nonlinear elastic behavior of bicycle chain on transmission efficiency. *Journal of applied mechanics*, 80(2), p. 021005.

Spicer, J. B. et al. (2001). Effects of frictional loss on bicycle chain drive efficiency. *J Mech Des*, 123(4), pp. 598–605.

Steck, H., Ekanadham, C. and Kallus, N. (2024). Is cosine-similarity of embeddings really about similarity? In *Companion Proceedings of the ACM Web Conference 2024*, pp. 887–890.

Su, J. et al. (2024). Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568, p. 127063.

Sudin, M. N., Abdullah, M. A., Shamsuddin, S. A., Ramli, F. R. and Tahir, M. M. (2014). Review of research on vehicles aerodynamic drag reduction methods. *International Journal of Mechanical and Mechatronics Engineering*, 14(2), pp. 35–47.

Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3), pp. 293–300.

Szűcs, G. and Tamás, B. (2018). Body part extraction and pose estimation method in rowing videos. *Journal of computing and information technology*, 26(1), pp. 29–43.

Taylor, M., Butcher, D., Crickmore, C. and Walker, A. D. (2024). Practical computational fluid dynamic predictions of a cyclist in a time trial position. *Sports Engineering*, 27(2), p. 34.

Tengattini, S. and Bigazzi, A. (2018). Validation of an outdoor coast-down test to measure bicycle resistance parameters. *Journal of Transportation Engineering, Part A: Systems*, 144(7), p. 04018031.

The Astropy Developers (2025). TrapezoidDisk2DKernel — astropy v7.2.0. `https://docs.astropy.org/en/stable/api/astropy.convolution.TrapezoidDisk2DKernel.html`, accessed: 2025-12-07.

Touvron, H. et al. (2023). Llama 2: Open foundation and fine-tuned chat models. `https://www.llama.com/llama2/`, accessed: 2025-11-17.

Uhlrich, S. D. et al. (2023a). Opencap: Human movement dynamics from smartphone videos. *PLoS computational biology*, 19(10), p. e1011462.

Uhlrich, S. D. et al. (2023b). OpenCap: Human movement dynamics from smartphone videos. *PLoS Comput Biol*, 19(10), p. e1011462.

Ultralytics (2025). Pose estimation — ultralytics yolo docs. `https://docs.ultralytics.com/tasks/pose/`, accessed: 2025-12-06.

Underwood, L. and Jermy, M. (2011). Fabric testing for cycling skinsuits. *Procedia Engineering*, 13, pp. 350–356, 5th Asia-Pacific Congress on Sports Technology (APCST).

Underwood, L., Schumacher, J., Burette-Pommay, J. and Jermy, M. (2011). Aerodynamic drag and biomechanical power of a track cyclist as a function of shoulder and torso angles. *Sports Engineering*, 14(2), pp. 147–154.

Underwood, L., Jermy, M., Eloi, P. and Cornillon, G. (2015). Helmet position, ventilation holes and drag in cycling. *Sports Engineering*, 18(4), pp. 241–248.

Union Cycliste Internationale(UCI) (2025). *Clarification Guide of the UCI Technical Regulation: As from 01.02.2025*. Union Cycliste Internationale, accessed: 14-11-2025.

University, P. (2025). Mechanics of the cyclist. Accessed: 2025-07-25.

Valenzuela, P. L. et al. (2020). Validity of a novel device for real-time analysis of cyclists' drag area. *Journal of Science and Medicine in Sport*, 23(4), pp. 421–425.

van Druenen, T. and Blocken, B. (2024). Cfd simulations of cyclist aerodynamics: Impact of computational parameters. *Journal of Wind Engineering and Industrial Aerodynamics*, 249, p. 105714.

van Erp, T., Galis, G. and Kordi, M. (2021). Validity of bicycle mounted pitot tubes for real-time analysis of cyclists' drag area for outdoor testing. *Journal of Science and Cycling*, 10(3), pp. 17–24.

Vangel, M. G. (1996). Confidence intervals for a normal coefficient of variation. *The American Statistician*, 50(1), pp. 21–26.

Vaswani, A. (2017). Attention is all you need. *arXiv preprint arXiv:170603762*.

Walker, A. D., Butcher, D., Crickmore, C. and Taylor, M. (2024). Practical cfd predictions of a cyclist in a time trial position.

Wang, L. et al. (2023). Improving text embeddings with large language models. *arXiv preprint arXiv:240100368*.

Webster, J. J. and Kit, C. (1992). Tokenization as the initial phase in NLP. In *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*.

Wei, J. et al. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:220607682*.

White, F. (2011). *Fluid Mechanics*. McGraw-Hill series in mechanical engineering, McGraw Hill.

Williams, D. et al. (2025). Effect of handlebar height on lower-limb biomechanics and upper body kinematics in elite pursuit cyclists. *ISBS Proceedings Archive*, 43(1), p. 84.

Winter, L., Bellenger, C., Grimshaw, P. and Crowther, R. G. (2023). Analysis of movement variability in cycling: An exploratory study. *Sensors*, 23(10), p. 4972.

Wojke, N., Bewley, A. and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, IEEE, pp. 3645–3649.

World Triathlon (2024). *World Triathlon Competition Rules 2024*. World Triathlon, accessed: 14-11-2025.

Wragge-Morley, R., Barnaby, G., Burgess, S. and Hicks, B. (2024). Big sprockets and small-pitch chains: Advancements in transmissions for olympic track cycling. *Journal of Science and Cycling*, 13(2), pp. 116–117.

xAI (2025). Grok: Ai assistant by xai. `https://grok.com/`, accessed: 2025-11-17.

Xia, H. et al. (2024). SportQA: A benchmark for sports understanding in large language models. In K. Duh, H. Gomez and S. Bethard, eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Mexico City, Mexico: Association for Computational Linguistics, pp. 5061–5081.

Xu, Z. et al. (2024). Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 2905–2909.

Yang, C. et al. (2024). Egoposeformer: A simple baseline for stereo egocentric 3d human pose estimation. `2403.18080`.

Yang, J. (2024). Rethinking tokenization: Crafting better tokenizers for large language models. *International Journal of Chinese Linguistics*, 11(1), pp. 94–109.

Yaseen, M. (2024). What is yolov9: An in-depth exploration of the internal features of the next-generation object detector. *arXiv preprint arXiv:240907813*.

Yoo, J.-C. and Han, T. H. (2009). Fast normalized cross-correlation. *Circuits, systems and signal processing*, 28(6), pp. 819–843.

Yu, W. et al. (2024). Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 14672–14685.

Zaidi, N. A., Cerquides, J., Carman, M. J. and Webb, G. I. (2013). Alleviating naive bayes attribute independence assumption by attribute weighting. *The Journal of Machine Learning Research*, 14(1), pp. 1947–1988.

Zhang, Y. and Zhao, N. (2024). Intelligent bike fitting system design based on openpose. In *Proceedings of the 2024 4th International Conference on Big Data, Artificial Intelligence and Risk Management*, pp. 638–643.

Zhang, Z., Wang, C., Qiu, W., Qin, W. and Zeng, W. (2020). Adafuse: Adaptive multiview fusion for accurate human pose estimation in the wild. *International Journal of Computer Vision*, 129(3), p. 703–718.

Zheng, C. et al. (2023). Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, 56(1), pp. 1–37.

Zheng, C., Zhou, P., Mao, J. and Zhang, X. (2024). Experimental investigations on the effects of longitudinal groove fabric for drag reductions on a track cyclist. *Experimental Thermal and Fluid Science*, 156, p. 111223.

Appendix

Template matching algorithm from Chapter 3

Algorithm 3 Radius Optimization via Template Matching

Require: $Y \in \mathbb{R}^{H \times W}$ (input image), K (template image), $\mathcal{I} = \{\ell_1, \ldots, \ell_L\}$ (list of locations), $\mathcal{R} = \{r_0, r_1, \ldots, r_n\}$ (radii to test)
Ensure: $\mathcal{S}$: set of tuples (Score, Radius, Index)
Initialize $\mathcal{S} \leftarrow \varnothing$
$i \leftarrow 0$
for each $\ell_i = (y_i, x_i) \in \mathcal{I}$ **do**
  for each $r \in \mathcal{R}$ **do**
    Resize K to $K_r \in \mathbb{R}^{(2r)\times(2r)}$
    Extract $M_{i,r}$ from M centered at (y_i, x_i) of size $(2r) \times (2r)$
    Normalize:

$$\hat{M}_{i,r} = \frac{M_{i,r}}{\max M_{i,r}} \cdot 255 - 127, \quad \hat{K}_r = \frac{K_r}{\max R_r} \cdot 255 - 127$$

    Compute score:

$$S(i,r) = \frac{1}{(2r)^2} \sum_{u=0}^{2r-1} \sum_{v=0}^{2r-1} \hat{M}_{i,r}(u,v)\ \hat{K}_r(u,v)$$

    Append $(S(i,r), r, i)$ to $\mathcal{S}$
  end for
  $i \leftarrow i + 1$
end for
return $\mathcal{S}$ =0

Table of Marks from Chapter 6

Variant	Condition	N	mean	std	sem	cv_%
Gemma2:27b	base	10	76.6670	1.309003	0.413943	1.707388
Gemma2:27b	sys	10	86.3460	0.821072	0.259646	0.950910
Gemma2:27b	sys+RAG	10	86.6640	1.340806	0.424000	1.547131
Gemma2:27b	sys+RAG+KB2	10	87.3010	1.673901	0.529334	1.917390
Gemma2:9b	base	10	71.1120	2.677660	0.846750	3.765412
Gemma2:9b	sys	10	69.6820	2.044189	0.646429	2.933596
Gemma2:9b	sys+RAG	10	72.7000	1.801789	0.569776	2.478390
Gemma2:9b	sys+RAG+KB2	10	73.8100	2.509444	0.793556	3.399870
Phi4:14b	base	10	81.5880	2.388062	0.755172	2.926977
Phi4:14b	sys	10	82.2230	2.456190	0.776715	2.987229
Phi4:14b	sys+RAG	10	83.3330	2.506281	0.792556	3.007549
Phi4:14b	sys+RAG+KB2	10	85.2360	1.680067	0.531284	1.971077
Qwen2.5:32b	base	10	86.1890	1.305496	0.412834	1.514690
Qwen2.5:32b	sys	10	86.8250	1.680802	0.531516	1.935850
Qwen2.5:32b	sys+RAG	10	86.6640	1.111737	0.351562	1.282813
Qwen2.5:32b	sys+RAG+KB2	10	88.7310	0.902557	0.285414	1.017184
Qwen2.5:7b	base	10	75.5570	1.861780	0.588746	2.464073
Qwen2.5:7b	sys	10	77.7790	2.246244	0.710325	2.887982
Qwen2.5:7b	sys+RAG	10	78.7340	0.821072	0.259646	1.042844
Qwen2.5:7b	sys+RAG+KB2	10	82.6990	1.745766	0.552060	2.110988
llama3.1:8b	base	10	73.6510	2.504194	0.791896	3.400082
llama3.1:8b	sys	10	73.3350	4.016093	1.270000	5.476366
llama3.1:8b	sys+RAG	10	75.2360	2.142222	0.677430	2.847337
llama3.1:8b	sys+RAG+KB2	10	78.0970	1.952918	0.617567	2.500632
llama3.2:3b	base	10	62.6980	4.114934	1.301256	6.563103
llama3.2:3b	sys	10	64.2840	2.824745	0.893263	4.394164
llama3.2:3b	sys+RAG	10	68.2530	5.796731	1.833087	8.493005
llama3.2:3b	sys+RAG+KB2	10	76.3490	3.383261	1.069881	4.431310

Table 12: Table of results showing all of the results obtained from all the different Language Models tested in this study, with each respective configuration, number of iterations, standard deviation in the mark, standard error of the mark and coefficient of variation.