UNIVERSITY OF PIRAEUS
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGIES
DEPARTMENT OF DIGITAL SYSTEMS

Machine Learning methods for Planning Conflict-free Trajectories

Doctoral Thesis

Alevizos Bastas

Piraeus, 2024

UNIVERSITY OF PIRAEUS
DEPARTMENT OF DIGITAL SYSTEMS

Machine Learning methods for Planning Conflict-free Trajectories

Doctoral Thesis Presented
by **Alevizos Bastas**
in Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Piraeus, 2024

Certified by. . . . . 

George Vouros
Professor, University of Piraeus
Supervisor
Member of Examination Committee

Certified by. . . . . 

Orestis Telelis
Assistant Professor, University of Piraeus
Advisor
Member of Examination Committee

Certified by. . . . . 

Konstantinos Blekas
Professor, University of Ioannina
Advisor
Member of Examination Committee

Certified by. . . . . 

Ioannis Theodoridis
Professor, University of Piraeus
Member of Examination Committee

Certified by. . . . . .

Grigorios Tsoumakas
Associate Professor, Aristotle University of Thessaloniki
Member of Examination Committee

Certified by. . . . . .

George Paliouras
Researcher, National Centre for Scientific Research “Demokritos”
Member of Examination Committee

Certified by. . . . . .

Theodoros Giannakopoulos
Researcher, National Centre for Scientific Research “Demokritos”
Member of Examination Committee

Abstract

Safe and efficient transportation, in terms of cost, time and distance covered, in the aviation domain is provided through the Air Traffic Management (ATM) system, which includes all airborne and ground-based operations required to ensure safe and efficient traffic flow. Every year the volume of air traffic increases pushing the ATM system to its limits, requiring it to handle greater complexity and density of traffic.

Different initiatives worldwide, such as NextGen [53] in the US and SESAR [70] in Europe, have been investigating the implementation of automation to enhance the efficiency and cost-effectiveness of the air traffic management (ATM) system. Towards this goal Artificial Intelligence and Machine Learning (AI/ML) methods are considered for providing accurate predictions of flight trajectories and addressing complexity issues while ensuring safety.

During airborne operations safety between aircraft is provided by the Air Traffic Control (ATC) service. According to International Civil Aviation Organisation (ICAO) Annex 11 [55], ATC is "a service provided for the purpose of: (a) preventing collisions: (1) between aircraft, and (2) on the maneuvering area between aircraft and obstructions; and (b) expediting and maintaining an orderly flow of air traffic". This includes imposing certain separation minima between aircraft, detecting conflicts that breach separation minima (loss of separation) between flights and their resolution by appropriate actions.

The provision of safe ATC services determines traffic volume, which must not exceed the airspace's capacities declared. However, capacities should be utilized to the maximum extent due to increased demand and the need for the optimal utilization of resources, without compromising the efficiency and safety of flights. This trade-off introduces challenging issues in the aviation industry, where AI/ML can provide solutions.

The objective of this Ph.D. study is to explore and present state of the art AI/ML algorithms towards planning conflicts-free trajectories in computationally efficient ways, following a methodology combining data-driven and agent-based approaches.

In the context of this study the conflicts-free trajectory planning task is defined to incorporate trajectory prediction and conflicts detection and resolution. While trajectory prediction concerns predicting the spatiotemporal evolution of the aircraft state along a trajectory (also called, trajectory evolution), conflicts detection and resolution concerns the detection of conflicts that breach separation minima (loss of separation) between flights and their resolution by appropriate actions. Therefore, the objective of the conflicts-free trajectory planning task is to predict the evolution of trajectories, regulating flights to avoid loss of separation. While trajectory planning may take place at the pre-tactical phase of operations, methods developed in this study are expected to have a large impact in the tactical phase of operations. Aiming to model stakeholders' decisions to planning conflicts-free trajectories, the major emphasis of this study is to imitate flights' trajectories and air traffic controller's behavior according to demonstrations provided by historical data.

Περίληψη

Οι ασφαλείς και αποτελεσματικές μεταφορές, όσον αφορά το κόστος, το χρόνο και την απόσταση που καλύπτεται, στον τομέα των αερομεταφορών παρέχονται μέσω του συστήματος διαχείρισης εναέριας κυκλοφορίας, το οποίο περιλαμβάνει όλες τις εναέριες και επίγειες λειτουργίες που απαιτούνται για την εξασφάλιση ασφαλούς και αποτελεσματικής ροής της κυκλοφορίας. Κάθε χρόνο ο όγκος της εναέριας κυκλοφορίας αυξάνεται και ωθεί το σύστημα διαχείρισης εναέριας κυκλοφορίας στα όριά του, απαιτώντας από αυτό να διαχειρίζεται μεγαλύτερη πολυπλοκότητα και πιο πυκνή κυκλοφορία.

Διάφορες πρωτοβουλίες παγκοσμίως, όπως η NextGen [53] στις ΗΠΑ και η SESAR [70] στην Ευρώπη, έχουν διερευνήσει την εφαρμογή αυτοματοποίησης για τη βελτίωση της αποτελεσματικότητας και της σχέσης κόστους-αποτελεσματικότητας του συστήματος διαχείρισης της εναέριας κυκλοφορίας. Για την επίτευξη αυτού του στόχου εξετάζονται μέθοδοι τεχνητής νοημοσύνης και μηχανικής μάθησης για την ακριβή πρόβλεψη τροχιών πτήσεων και την αντιμετώπιση ζητημάτων πολυπλοκότητας εξασφαλίζοντας ταυτόχρονα την ασφάλεια των πτήσεων.

Κατά τη διάρκεια των εναέριων επιχειρήσεων η ασφάλεια μεταξύ των αεροσκαφών παρέχεται από την υπηρεσία ελέγχου εναέριας κυκλοφορίας. Σύμφωνα με το διεθνή οργανισμός πολιτικής αεροπορίας, παράρτημα 11 [55], ο έλεγχος εναέριας κυκλοφορίας είναι "υπηρεσία που παρέχεται με σκοπό: α) την πρόληψη συγκρούσεων: 1) μεταξύ αεροσκαφών και 2) στον τερματικό χώρο ελιγμών μεταξύ αεροσκαφών και εμποδίων και β) την επιτάχυνση και διατήρηση ομαλής ροής της εναέριας κυκλοφορίας". Αυτό περιλαμβάνει την επιβολή ορισμένων ορίων ελάχιστης απόστασης μεταξύ αεροσκαφών, τον εντοπισμό συγκρούσεων που παραβιάζουν τα όρια ελάχιστης απόστασης μεταξύ των πτήσεων και την επίλυσή τους με κατάλληλες ενέργειες.

Η παροχή ασφαλών υπηρεσιών ελέγχου εναέριας κυκλοφορίας καθορίζει τον όγκο της κυκλοφορίας, ο οποίος δεν πρέπει να υπερβαίνει τη χωρητικότητα του εναέριου χώρου. Ωστόσο, ο εναέριος χώρος θα πρέπει να αξιοποιείται στο μέγιστο, λόγω της αυξημένης ζήτησης και της ανάγκης για βέλτιστη αξιοποίηση των πόρων, χωρίς όμως να διακυβεύεται η αποτελεσματικότητα και η ασφάλεια των πτήσεων. Αυτός ο συμβιβασμός εισάγει προκλήσεις στον κλάδο των αερομεταφορών, όπου η τεχνητή νοημοσύνη/μηχανική μάθηση μπορούν να δώσουν λύσεις.

Ο στόχος αυτής της διδακτορικής μελέτης είναι να διερευνήσει και να παρουσιάσει μεθόδους αιχμής τεχνητης νοημοσύνης/μηχανικής μάθησης με στόχο την πρόβλεψη τροχιών ελευθέρων συγκρούσεων με υπολογιστικά αποδοτικούς τρόπους, ακολουθώντας μια μεθοδολογία που συνδυάζει προσεγγίσεις που βασίζονται σε δεδομένα και προσεγγίσεις που βασίζονται πράκτορες.

Στο πλαίσιο αυτής της μελέτης, η πρόβλεψη τροχιών ελευθέρων συγκρούσεων περιλαμβάνει την πρόβλεψη τροχιών και την ανίχνευση και επίλυση συγκρούσεων. Ενώ η πρόβλεψη τροχιάς αφορά την πρόβλεψη της χωροχρονικής εξέλιξης της κατάστασης του αεροσκάφους κατά μήκος μιας τροχιάς, η ανίχνευση και η επίλυση συγκρούσεων αφορά την ανίχνευση συγκρούσεων που παραβιάζουν τα όρια ελάχιστης απόστασης μεταξύ πτήσεων και την επίλυσή τους με κατάλληλες ενέργειες. Επομένως, ο στόχος της πρόβλεψης τροχιών ελευθέρων συγκρούσεων είναι η πρόβλεψη της εξέλιξης των τροχιών, μεταβάλλοντας τις πτήσεις ώστε να αποφεύγεται η παραβίαση των ορίων ελάχιστης απόστασης μεταξύ τους. Ενώ ο σχεδιασμός τροχιάς μπορεί να πραγματοποιείται στην προ-τακτική φάση των πτήσεων (πριν την απογείωση), οι μέθοδοι που αναπτύχθηκαν στην παρούσα μελέτη αναμένεται να έχουν μεγάλο αντίκτυπο στην τακτική φάση των πτήσεων (κατά την εναέρα φάση της πτήσης). Με στόχο τη μοντελοποίηση των

αποφάσεων των εμπλεκόμενων μερών (χρήστες του εναέριου χώρου, ελεγκτές εναέριας κυκλοφορίας, κλπ.) για την πρόβλεψη τροχιών ελευθέρων συγκρούσεων, η κύρια έμφαση της παρούσας μελέτης είναι η μίμηση των τροχιών πτήσεων και της συμπεριφοράς των ελεγκτών εναέριας κυκλοφορίας σύμφωνα με επιδείξεις που παρέχονται σε ιστορικά δεδομένα.

Publications

Journal publications:

[1] Bastas, Alevizos, and George Vouros. ”Data-driven prediction of Air Traffic Controllers reactions to resolving conflicts.” Information Sciences 613 (2022): 763-785.

[2] Bastas, Alevizos, and George A. Vouros. ”Data-Driven Modeling of Air Traffic Controllers' Policy to Resolve Conflicts.” Aerospace 10, no. 6 (2023): 557.

[3] Papadopoulos, George, Alevizos Bastas, George A. Vouros, Ian Crook, Natalia Andrienko, Gennady Andrienko, and Jose Manuel Cordero. ”Deep reinforcement learning in service of air traffic controllers to resolve tactical conflicts.” Expert Systems with Applications 236 (2024): 121234.

Conference publications:

[1] Vouros, George, George Papadopoulos, Alevizos Bastas, Jose Manuel Cordero, and Rubén Rodrigez Rodrigez. ”Automating the Resolution of Flight Conflicts: Deep Reinforcement Learning in Service of Air Traffic Controllers.” In PAIS 2022, pp. 72-85. IOS Press, 2022.

[2] Valle, Natividad, M. Florencia Lema, José Manuel Cordero, Enrique Iglesias, Rubén Rodríguez, Gennady Andrienko, Natalia Andrienko et al. ”Transparency & Explainability in higher levels of automation in the ATM domain.” SESAR Innovation days, 2022.

Other publications:

[1] Bastas, Alevizos, Theocharis Kravaris, and George A. Vouros. ”Data driven aircraft trajectory prediction with deep imitation learning.” arXiv preprint arXiv:2005.07960 (2020).

[2] Kravaris, Theocharis, Alevizos Bastas, and George A. Vouros. ”Predicting Aircraft Trajectories via Imitation Learning.”, Adaptive & Learning Agents (ALA) @ AAMAS, 2021.

Acknowledgments

I would like to express my sincere gratitude and appreciation to all those who contributed to this dissertation and supported me throughout this journey.

First and foremost, I would like to express my deepest gratitude to my thesis supervisor Professor George Vouros, for his immense knowledge, and guidance throughout my research endeavor. His unwavering support, patience and mentoring have been instrumental in completing my Ph.D. studies and the lessons I have learned from him, will always accompany me throughout my career.

Furthermore, I would like to express my gratitude to my advisors, Associate Professor Orestis Telelis, and Professor Konstantinos Blekas for sharing their scientific knowledge and experience during my studies. I would also like to thank my Ph.D. thesis committee members Professor Ioannis Theodoridis, Associate Professor Grigorios Tsoumakas, Dr. George Paliouras and Dr. Theodoros Giannakopoulos for devoting time and effort to offer constructive criticism.

Moreover, I would like to extend my gratitude to Dr. Georgios Santipantakis, Mr. Theocharis Kravaris, Mr. George Papadopoulos and all the members of the AI-Lab of the Department of Digital Systems for their invaluable cooperation and support throughout these years.

In addition, I would like to thank Mr. Jose Manuel Cordero Garcia and CRIDA (Centro de Referencia I+D+i ATM) for their support, for always being eager to share their experience and expertise with me and for providing the datasets that this research was based on. Also I would like to sincerely thank SESAR JU and the Engage KTN for supporting and funding this research, sharing their knowledge with me and giving me the chance to be a part of their community and work on this challenging and interesting domain.

Last but not least, I would like to thank my family and friends for believing in me and supporting my decisions all these years.

This research has received funding from the SESAR Joint Undertaking under the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 783287. The opinions expressed herein reflect the authors' view only. Under no circumstances shall the SESAR Joint Undertaking be responsible for any use that may be made of the information contained herein.

Contents

I Introduction and Background knowledge **22**

1 Introduction **23**
1.1 Motivating Problem for Air Traffic Management . . . 23
1.2 Thesis Contributions . . . 24
1.2.1 Data-driven prediction of flight trajectories per Origin-Destination pair. . 25
1.2.2 Data-driven modeling of the ATCOs' behavior in resolving conflicts. . . . 25
1.2.3 Conflicts-free trajectory planning. . . . 26
1.3 Thesis Structure . . . 28

2 Background **29**
2.1 Machine Learning . . . 29
2.1.1 Supervised Learning . . . 29
2.1.1.1 Neural Networks . . . 30
2.1.1.2 Decision Trees . . . 31
2.1.1.3 Random Forest . . . 32
2.1.1.4 Gradient Boosting . . . 32
2.1.2 Unsupervised Learning . . . 32
2.1.2.1 Variational Auto-Encoders . . . 32
2.1.3 Reinforcement Learning . . . 33
2.1.3.1 Markov Decision Process . . . 33
2.1.3.2 Deep Reinforcement Learning . . . 35
2.1.3.2.1 Policy Gradient methods and Trust Region Policy Optimization . 35
2.1.3.2.2 Actor Critic Methods and Generalized Advantage Estimation . . . 38
2.1.4 Imitation Learning . . . 39
2.1.4.1 Behavioral Cloning . . . 39
2.1.4.2 Inverse Reinforcement Learning . . . 39
2.1.4.3 Adversarial Imitation Learning . . . 39
2.1.4.3.1 Directed InfoGAIL . . . 41
2.2 Air Traffic Management . . . 42
2.2.1 Conflict Detection and Resolution . . . 43

II Planning Conflicts-Free Trajectories **45**

3 Data-driven prediction of flight trajectories per origin-destination pair **47**
3.1 Related Work . . . 47
3.2 Data-Driven Aircraft Trajectory Prediction . . . 49
3.2.1 Problem Specification . . . 50
3.3 Predicting aircraft trajectories with IL methods . . . 51

3.3.1 States and Actions . . . 51
3.3.2 GAIL for predicting flight trajectories . . . 51
3.4 Experimental Evaluation . . . 53
3.4.1 Experimental Setting . . . 53
3.4.2 Experimental Results . . . 55
3.5 Conclusions . . . 59

4 Data-driven modeling of the ATCOs' behavior in resolving conflicts 61
4.1 Related Work . . . 62
4.2 Problem Specification . . . 64
4.2.1 Definitions . . . 64
4.2.2 ATCOs' Reaction Prediction Problem Specification . . . 67
4.2.3 Modeling the ATCO policy Problem Specification . . . 68
4.3 Methodology stages . . . 68
4.4 Data sources . . . 69
4.5 Trajectory states . . . 70
4.6 Solving the ATCO's reaction problem . . . 71
4.6.1 Simulating uncertainty in trajectory evolution . . . 71
4.6.2 ATCO modes and resolution actions . . . 73
4.6.3 Learning timely reactions . . . 73
4.7 Solving the ATCOs' policy learning problem . . . 75
4.7.1 ATCO resolution actions . . . 75
4.7.2 Modeling the ATCOs policy . . . 76
4.8 Experimental Evaluation . . . 79
4.8.1 Predicting ATCOs' reactions . . . 80
4.8.1.1 Experimental Setting . . . 81
4.8.1.2 Data sets and Pre-processing . . . 83
4.8.1.3 Evaluation methodology . . . 84
4.8.1.4 Experimental Results . . . 90
4.8.2 Modeling ATCO's policy . . . 95
4.8.2.1 Experimental Setting . . . 95
4.8.2.2 Data sets and preprocessing . . . 96
4.8.2.3 Experimental results . . . 96
4.9 Conclusions . . . 102
4.9.1 ATCOs' reactions . . . 102
4.9.2 ATCOs' policy . . . 103

5 Towards Planning Conflicts-Free Trajectories 105
5.1 Problem specification . . . 105
5.2 A sequential framework for planning conflicts-free trajectories . . . 106
5.3 Models for conflicts-free trajectory planning . . . 109
5.3.1 Features considered by the models . . . 109
5.3.1.1 Trajectory prediction model without considering conflicts . . . 109
5.3.1.2 Models for predicting the ATCO reactions and modeling the ATCO policy . . . 110
5.3.1.3 Models predicting the trajectory's evolution during the execution of a maneuver . . . 110
5.3.2 Methods used . . . 111
5.4 Experimental evaluation . . . 111
5.4.1 Data sets and preprocessing . . . 111
5.4.1.1 TPMwoCC . . . 112

5.4.1.2 TPDT and TPSC . . . 113
5.4.1.3 ARP and RATPr . . . 113
5.4.2 Experimental results . . . 114
5.5 Conclusions . . . 121

III Conclusions **124**

6 Conclusions and future study **125**

A **134**
A.1 Supplementary material on alternative problem formulation and AI/ML methods for learning the ATCOs' policy. . . . 134
A.1.1 Learning the ATCOs policy problem as an IL problem . . . 134
A.1.1.1 Problem specification . . . 134
A.1.1.2 Solving the ATCOs policy problem by imitation . . . 135
A.1.2 Experimental Results . . . 136
A.2 Description of data this study relies on . . . 138
A.2.1 Surveillance data . . . 138
A.2.2 Flight Plan data . . . 139
A.2.3 Sector Configuration Data . . . 139
A.2.4 Weather Data . . . 139
A.2.5 ATCO Events Dataset . . . 139
A.3 Conjugate gradient method . . . 140

List of Figures

2.1 Example of a NN, showing the input, hidden and output layers. 30
2.2 Example of a decision tree, showing the decision nodes corresponding to conditions, based on which leaf nodes corresponding to different partitions of the data space are formed. 31
2.3 The RL scheme modeling interactions between the agent and its environment. . . 33
2.4 Inverse Reinforcement Learning (IRL) training loop. 40
2.5 GAIL architecture. 40

3.1 Along-Track Error (ATE) and Cross-Track Error (CTE) errors w.r.t. the predicted trajectory's points at times t and $t+1$, denoted by $pred_t$ and $pred_{t+1}$, and the actual trajectory's point act_t at time t. $\Delta X = X_p - X_a$ is the difference in the X dimension (longitude) of $pred_t$ (X_p) and act_t (X_a). $\Delta Y = Y_p - Y_a$ is the difference in the Y dimension (latitude) of $pred_t$ (Y_p) and act_t (Y_a). Ψp denotes the bearing of the predicted trajectory (i.e. the angle between the direction of the trajectory and the North). 54
3.2 Specification of the bounding box and of the prediction area for LHR-FCO trajectories. 55
3.3 Example of a predicted (black) vs the corresponding historical (red) trajectory between HEL-LIS 59

4.1 Methodology stages for predicting the ATCOs' reactions and the ATCOs' policy . . 68
4.2 Trajectory points (blue points) associated with a corresponding ATCOs' event. The figure indicates the callsign, the departure (apt_from) and the destination airports (apt_to), the resolution action type (mwm_code), the time (time_annotation) and the sector in which the resolution was issued (sector). The (red) point in the middle of the trajectory depicts the point (with the timestamp 1460660342) associated to the ATCOs' event. 70
4.3 Features enriching points of the ownship's trajectory w.r.t. the aircraft flying a conflicting trajectory T_j. 72
4.4 Modes c^t, categorical actions a^t and continuous actions $\Delta course, \Delta s_v, \Delta s_h, \Delta t$ predicted at each time point t by the encoder and decoder networks given the state s^t and mode c^{t-1}.Figure (a) shows the overall architecture of the method, while Figure (b) shows the architectures of the encoder and decoder in detail. . . 74
4.5 The Neural Network (NN) and Neural Network with attention (NN+att) hyperparameters (**a**), the NN classifier without attention (**b**), and the NN classifier with attention (**c**). $Dense_Q$, $Dense_K$, and $Dense_V$ denote the query, key, and value projections, respectively. 78
4.6 The SA area and the area defined by D_{th} (red rectangular area) w.r.t. the ownship's position (white dot) in the sector-ignorant case. 83

4.7 Score function: Axis x values correspond to x/n, when $n = 5$, and the temporal distance x in seconds is shown at the bottom. Axis y shows the score. 85
4.8 One of the trajectories with the highest difference between the weighted and non-weighted f1-score. "Predicted" shows the modes predicted by the VAE model, whereas "Expert" shows the modes reported in the dataset. X-axis: the sequence number of the trajectory states. Y-axis: the modes. (Blue) Dots denote the mode at each point. (Green) Solid vertical lines at x=0 show the start of the trajectory, while (red) dashed vertical lines at x=100 indicate the point with actual RATP. . . 92

5.1 Outline of the models' combination towards providing conflicts-free trajectories. Given an initial historical state (1), the ARP predicts whether a resolution action will be applied (2). If the ARP predicts the assignment of a resolution action, then RATPr decides the type of the resolution action (3). Finally, the trajectory prediction ensemble is executed (4) controlling the ownship's movement 107
5.2 The trajectory prediction ensemble, exploiting different trajectory prediction models to predict how the ownship's trajectory will evolve. Initially the model set for use is the TPMwoCC. If a resolution action is assigned the selected model changes to CRMP. If no resolution action should be applied the previously selected model is used. 108
5.3 The preprocessing steps for $step_s = 3$. Figure 5.3a shows the original annotated trajectory T. Red points denote $RATP_{aa}$ points and difference in the opacity denotes which points correspond to each subtrajectory T_i^s. These are shown in the disaggregated form in figure 5.3b. Figure 5.3c shows the disaggregated trajectories corresponding to T for the balanced case and figure 5.3d shows which subtrajectory corresponds to T for the non-balanced case. 114
5.4 Number of average trajectory points per trajectory, where the altitude changes over 5 ft for different altitude intervals shown in the X-axis, for historical trajectories and for trajectories predicted by TPMwoCC. 120

List of Tables

3.1 Prediction Errors (in meters) and ETA (in seconds) 56
3.2 Prediction Errors box plots: Numbers below the boxes indicate the medians. . . . 57

4.1 Problem-specific parameters. 66
4.2 Active and passive Loss functions. p_θ denotes the probability output of the model, p the distribution over different labels as revealed by the dataset and K is the number of labels. 79
4.3 Hyperparameters of the decision trees used for the Random Forest (RF) and Gradient Tree Boosting (GTB) algorithms. Descriptions are from scikit-learn. 80
4.4 Hyperparameters used for the Random Forest (RF) algorithm. Descriptions are from scikit-learn. 81
4.5 Hyperparameters of the gradient tree boosting algorithm. Descriptions are from scikit-learn. 82
4.6 Prior distribution of modes (C_0, C_1, C_2) for different subsampling $step$ values computed on the dataset for the sector-ignorant experimental setting. 84
4.7 False Positives (FP) and True Positives (TP) weights based on the score function, given the time point of the prediction t_p and the time point t_a of the closest actual RATP or the time point t_{aa} of the closest actual or annotated RATP. Elements on the diagonal of the table are true positives (TP) and dashes indicate that the weighted measures do not apply between modes and resolution actions. 87
4.8 False Negatives (FN) and True Negatives (TN) weights based on the score function, given the time point of the prediction t_p and the time point t_a of the closest actual RATP or the time point t_{aa} of the closest actual or annotated RATP. Elements on the diagonal of the table are true positives (TP) and dashes indicate that the weighted measures do not apply between modes and resolution actions. . . . 88
4.9 Experimental results of the sector-ignorant case achieved by the VAE and the encoder (Enc). Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes and the resolution actions of ATCOs, for the non-weighted and weighted measures. 90
4.10 Experimental Results of the sector-related case achieved by the VAE and the encoder (Enc). Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes and the resolution actions of ATCOs, for the non-weighted and weighted measures. 93
4.11 p-values computed by applying the Wilcoxon signed rank test on the unweighted average of the f1-scores (weighted and non-weighted) achieved by the VAE and the encoder when predicting the modes of the test set. The samples of the populations tested are the unweighted averages over the modes of the f1-scores (weighted and non-weighted) of the 10 experiments achieved by the VAE and the encoder. . 93

4.12 Number of cases within the 95% confidence interval where the models do not predict a resolution action to any of the annotated or actual RATPs or any point in the time window of 70s near the actual or annotated RATPs. 94
4.13 Scatterplots depicting the probability assigned to each mode by each model (VAE or encoder (Enc)), at every point in 10 trajectories. Column "Setting" denotes the sector-ignorant or sector-related experimental setting. The x-axis shows the sequence number of trajectory states and the y-axis the probability of each mode. Solid (green) vertical lines denote the start of each of the 10 trajectories, while (red) dashed lines denote the actual RATPs. Numbers over the solid (green) lines denote the sequence number of each trajectory. 98
4.14 Experimental results achieved by the NN classifier with an attention mechanism (NN+att) and without attention (NN), in addition to the Random Forest (RF) and the GTB algorithms. Columns report the 95% confidence interval of precision, recall, f1-score, and MCC with regard to the resolution action types of ATCOs. . . 99
4.15 Experimental results achieved by balanced RF, NN with attention SEAL and data augmentation (NN+att+SEAL+augm), and NN with attention active-passive loss and data augmentation (NN+att+AP loss+augm). Columns report the 95% confidence interval of precision, recall, f1-score, and MCC with regard to the resolution action types of ATCOs. 101

5.1 Experimental results, achieved by the VAE and the RF using 5-fold cross validation for the non-balanced case considering samples from all OD pairs. Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes of ATCOs' behavior, for the non-weighted and weighted measures. 116
5.2 Experimental results achieved by the VAE and the RF for the balanced case using 5-fold cross validation considering samples from all OD pairs. Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes of ATCOs' behavior, for the non-weighted and weighted measures. 117
5.3 Experimental results of the sector-ignorant case achieved by the RF on the LPPT-LFPO OD pair. Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes of the ATCOs behavior, for the non-weighted and weighted measures. 118
5.4 The average number of LsOS per trajectory for the historical data, TPMwoCC and TPP. 118
5.5 The average number of LsOS per trajectory that were predicted (Avg Predicted LsOS/ trajectory) for the TPP case, and the average number of LsOS per trajectory that were predicted and the TPP reacted to (Avg Predicted, Reacted LsOS / trajectory). 118
5.6 The average LsOS per trajectory that were consistently predicted (Avg consistently predicted LsOS/trajectory) and the average reactions of the ARP to consistently predicted LsOS per trajectory (Avg reactions to consistently predicted LsOS /trajectory) 119
5.7 Prediction Errors (in meters) 121

A.1 Parameters of the GAIL algorithm. 136
A.2 Experimental results achieved by the IL algorithm GAIL, exploiting an attention mechanism (GAIL+att) and without attention (GAIL). Columns report the 95% confidence interval of precision, recall, f1-score and the MCC w.r.t. resolution action types of ATCOs. 137

Acronyms

AI Artificial Intelligence. 6, 13, 23, 24, 25, 26, 27, 28, 29, 47, 61, 62, 66, 69, 76, 96, 102, 103, 114, 125, 126, 134, 138

AIL Adversarial Imitation Learning. 39

AoR Area of Responsibility. 81, 94, 95

ARP ATCO Reaction Predictor. 13, 15, 17, 107, 108, 112, 113, 115, 116, 117, 118, 119, 121, 122

ASM Airspace Management. 42

ATC Air Traffic Control. 23, 24, 25, 42, 43, 44, 64, 66, 69, 83, 96, 104, 112, 138, 139

ATCO Air Traffic Controller. 25, 26, 27, 29, 42, 61, 62, 63, 64, 67, 68, 69, 72, 73, 75, 76, 77, 79, 83, 85, 94, 96, 100, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 119, 120, 121, 122, 125, 126, 127, 128, 138, 139, 140

ATCOs Air Traffic Controllers. 14, 16, 17, 23, 24, 25, 26, 27, 42, 43, 44, 46, 61, 62, 66, 67, 68, 69, 70, 71, 73, 74, 75, 76, 77, 79, 80, 83, 84, 85, 86, 90, 91, 92, 93, 95, 96, 97, 99, 100, 102, 103, 104, 105, 106, 107, 111, 112, 113, 115, 116, 117, 118, 120, 122, 125, 126, 127, 128, 134, 135, 136, 137, 138

ATE Along-Track Error. 14, 47, 53, 54, 55

ATFCM Air Traffic Flow and Capacity Management. 42

ATM Air Traffic Management. 23, 24, 27, 42, 48

ATON Automated NORVASE Takes. 66, 69, 83, 96, 112, 138

ATS Air Traffic Service. 42, 43

BC Behavioral Cloning. 39, 52, 53, 55

BCN Barcelona. 53, 54, 139

CD&R Conflict Detection and Resolution. 25, 26, 27, 28, 42, 43, 62, 63, 64, 71, 94, 102, 105, 109, 114, 115, 120, 121, 122, 123, 125, 127, 128, 134

CP Crossing Point. 65

CPA Closest Point of Approach. 65, 66, 67, 70, 71, 72, 73, 110

CRMP Conflict Resolution Maneuver Predictor. 15, 107, 108, 111

CTE Cross-Track Error. 14, 47, 53, 54, 55

DRL Deep Reinforcement Learning. 35, 63

DT Decision Tree. 31

DTs Decision Trees. 31, 32, 79

EC Executive Controller. 43

EGKK Gatwick. 84, 96, 112, 138

EHAM Amsterdam. 84, 96, 112, 138

ETA Estimated Time of Arrival. 51, 54, 55, 58

FCO Fiumicio. 53, 54, 55, 58, 139

FIR Flight Information Region. 139

FL Flight Level. 43

FN False Negatives. 16, 86, 88, 89

FP False Positives. 16, 86, 87, 89

GAE Generalized Advantage Estimation. 38, 52

GAIL Generative Adversarial Imitation Learning. 17, 26, 38, 39, 40, 41, 47, 48, 49, 51, 52, 53, 54, 55, 59, 61, 111, 112, 125, 135, 136, 137, 138

GB Gradient Boosting. 32, 79

GTB Gradient Tree Boosting. 16, 17, 61, 79, 80, 97, 99, 100, 102, 103, 136

HCAI Human-Centric AI. 23, 24, 28

HEL Helsinki. 14, 53, 54, 58, 59, 139

ICAO International Civil Aviation Organisation. 6, 24, 42, 43, 138

IFR Instrument Flight Rules. 43

IL Imitation Learning. 13, 17, 25, 26, 27, 39, 47, 48, 49, 50, 51, 58, 59, 61, 104, 112, 125, 134, 135, 136, 137, 138

IRL Inverse Reinforcement Learning. 39, 40

KL Kullback–Leibler. 33, 52

LEMG Malaga. 84, 96, 112, 138

LFPO Paris. 84, 96, 112, 115, 138

LHR London Heathrow. 53, 54, 55, 58, 139

LIS Lisbon. 14, 53, 54, 58, 59, 139

LOS Loss of Separation. 117, 118, 119, 121, 127

LPPT Lisbon. 84, 96, 112, 115, 138

LSGG Geneva. 84, 96, 112, 138

LsOS Losses of Separation. 17, 116, 117, 118, 119, 120, 121, 122, 127

LSTM Long Short-Term Memory. 48

LSZH Zurich. 84, 96, 112, 138

MAD Madrid. 53, 54, 139

MAE Mean Absolute Error. 78, 79

MCC Matthews Correlation Coefficient. 17, 61, 95, 96, 97, 99, 100, 101, 136, 137

MDP Markov Decision Process. 134

MDPs Markov Decision Processes. 33

ML Machine Learning. 6, 13, 24, 25, 26, 27, 28, 29, 39, 44, 47, 61, 66, 69, 76, 96, 102, 103, 109, 114, 121, 125, 126, 134, 138

MSE Mean Square Error. 53, 74

NCE Normalized Cross Entropy. 78, 79

NFL Normalized Focal Loss. 78, 79

NLP Natural Language Processing. 30

NN Neural Network. 14, 17, 30, 31, 32, 35, 39, 48, 63, 76, 77, 78, 97, 99, 100, 101, 102, 103, 136

NNs Neural Networks. 30, 35, 53, 61, 63, 76

NOAA National Oceanic and Atmospheric Administration. 53, 138, 139

NPG Natural Policy Gradient. 37

OD Origin-Destination. 17, 24, 25, 28, 46, 47, 48, 53, 55, 59, 60, 71, 84, 108, 109, 112, 115, 116, 117, 118, 120, 125, 128, 138, 139

PC Planning Controller. 43

RATP Resolution Action Trajectory Point. 15, 16, 84, 85, 87, 88, 91, 92, 96, 100, 112, 113

RATPr Resolution Action Type Predictor. 13, 15, 107, 108, 112, 113, 116

RATPs Resolution Action Trajectory Points. 17, 84, 85, 91, 92, 94, 95, 98

RCE Reverse Cross Entropy. 78, 79

RF Random Forest. 16, 17, 32, 61, 79, 80, 81, 97, 99, 100, 101, 102, 103, 111, 115, 116, 117, 118, 121, 126, 136

RL Reinforcement Learning. 11, 14, 28, 29, 30, 33, 35, 39, 42, 63, 104

RMSE Root Mean Squared Error. 47, 53, 54, 55

SACTA Automated System of Air Traffic Control. 69, 83, 96, 112, 138

SEAL Self-Evolution Average Label. 17, 77, 100, 101, 102

TMA Terminal Maneuvering Area. 25

TN True Negatives. 16, 86, 88, 89

TP True Positives. 16, 86, 87, 88, 89

TPDT Trajectory Predictor, Direct-to Maneuver. 13, 111, 112, 113, 115, 120

TPMwoCC Trajectory Prediction Model without Considering Conflicts. 12, 15, 17, 106, 108, 111, 112, 113, 115, 116, 118, 119, 120, 128

TPP Trajectory Prediction Pipeline. 17, 106, 112, 114, 115, 116, 117, 118, 119, 120, 121

TPSC Trajectory Predictor, Speed Change Maneuver. 13, 111, 112, 113, 115, 120

TRPO Trust Region Policy Optimization. 37, 38, 41, 52, 136

VAE Variational Auto-Encoder. 15, 16, 17, 25, 31, 33, 42, 61, 73, 74, 75, 90, 91, 92, 93, 94, 95, 98, 102, 103, 111, 113, 115, 116, 117, 121, 125, 126

VAEs Variational Auto-Encoders. 32, 33, 73

VE Vertical Error. 47, 53, 54, 55

Part I

Introduction and Background knowledge

Chapter 1

Introduction

Human-Centric AI (HCAI) has received a lot of attention in recent years [25]. According to the HCAI concept autonomous agents and humans will work together as a team, complementing each other towards completing specific tasks. For this to be achieved, agent decisions should be understandable by humans and also AI agents should consider human preferences, human-like expertise and human tolerance in reacting to situations. Thus, AI agents' decisions should be explainable and also agents should somehow model human behavior.

The criticality of considering human preferences, human-like expertise and human tolerance in reacting to situations is more evident in safety critical domains. When safety is key, it is important to not overly exceed human capabilities, guiding the human operators in complex situations that he/she cannot handle. Although actions that lead to such complex situations might be optimal with respect the task at hand, applying them could compromise safety, i.e., the human operator will not be able to ensure safety in case the system crashes, human operators might be more prone to mistakes due to increased stress, etc. On the other hand respecting human preferences and human tolerance can help ensure safety and also increase the trustworthiness to the system, as system recommendations and actions become more self-explanatory and inherently transparent to the human operator.

This study contributes to human behavior modeling and the HCAI concept through the Air Traffic Management (ATM) domain. Specifically, this study a) imitates flight trajectories shaped by multiple stakeholders (mainly airspace users and Air Traffic Controllers (ATCOs)) and thus models their preferences, practices and constraints in an aggregated way, b) learns models of the ATCOs behavior in resolving conflicts between aircraft and c) considers the combination of such models towards a method for planning conflicts-free trajectories.

1.1 Motivating Problem for Air Traffic Management

Europe has a complex airspace, where 30.000 daily flights usually overfly its sky. Therefore, it is one of the airspaces with most activity in the world. While this number is expected to increase significantly in the coming years, ATM needs to handle greater complexity and larger volumes of traffic.

Different initiatives worldwide, such as NextGen [53] in the US and SESAR [70] in Europe, have been investigating the implementation of automation to enhance the efficiency and cost-effectiveness of the ATM system. Air Traffic Control (ATC) in the ATM domain, according to

ICAO Annex 11 [55], is "a service provided for the purpose of: (a) preventing collisions: (1) between aircraft, and (2) on the maneuvering area between aircraft and obstructions; and (b) expediting and maintaining an orderly flow of air traffic". The provision of safe ATC services determines traffic volume, which must not exceed the capacities declared. However, capacities should be utilized to the maximum extent due to increased demand and the need for the optimal utilization of resources, without compromising the efficiency and safety of flights. This trade-off introduces challenging issues in the aviation industry, where Artificial Intelligence and Machine Learning (AI/ML) can provide solutions.

To maintain safety and prevent collisions between aircraft—a consequence of increased traffic—the ATM system imposes specific separation minima constraints between aircraft, both at the horizontal and vertical axes. The ATC service is responsible for maintaining these separation minima by detecting conflicts that breach separation minima between flights and resolving them by appropriate actions.

The main objective of conflicts-free planning of trajectories is to protect the ATC service from overload, enabling ATCOs to deal with complex traffic situations. Given the uncertainties during the planning phase, as well as while executing a plan, reliable planning of conflicts-free trajectories is not that straightforward.

While planning of flight trajectories involves multiple stakeholders (airspace users, air navigation service providers, network manager, airport operators), planning of conflicts-free trajectories also brings the preferences/best practices of ATCOs in performing their duties. Based on the above, this study is motivated to present methods for the planning of conflicts-free trajectories, either at the pre-tactical phase (hours before take off), or at the tactical phase (while airborne) of operations, incorporating into the process preferences/practices and constraints of stakeholders (mainly, air space users and ATCOs), building models that are close to their objectives and their behavior, as these are revealed by historical data on executing flight trajectories and resolving conflicts.

Thus, this study contributes towards HCAI as it aims to model and imitate human behavior and preferences reinforcing human-AI interaction in a safety critical domain. From the ATM point of view, this study contributes towards collaborative decision making (i.e. considering multiple stakeholders) by imitating conflicts-free trajectory planning (i.e. trajectory prediction and conflicts detection and resolution), accounting for complex phenomena due to traffic, increasing predictability via efficient operation plans, reducing buffers and uncertainty as much as possible, and reducing flight inefficiencies due to tactical ATC actions, supporting better planning of operations for airspace users.

1.2 Thesis Contributions

Given that the main objective of this study is to develop AI/ML methods towards planning conflicts-free trajectories, the main contribution is the development of data-driven AI/ML models for (a) the prediction of trajectories, (b) the resolution of conflicts among flights, as well as (c) the combination of such models towards devising a method for planning conflicts-free trajectories.

To achieve this main objective, this study advances the state of the art in three major and challenging topics:

1. Data-driven prediction of flight trajectories per Origin-Destination (OD) pair.

2. Data-driven modeling of the ATCOs' behavior in resolving conflicts.

3. Conflicts-free trajectory planning.

1.2.1 Data-driven prediction of flight trajectories per Origin-Destination pair.

Specifically, this study has formulated the trajectory prediction problem as a data-driven Imitation Learning (IL) problem and developed IL algorithms for learning trajectory prediction models for different OD pairs. The study reports on extensive experimental results regarding the efficacy of these models.

Specifically, major contributions made are as follows:

- The trajectory prediction problem has been formulated as an IL process, where models of trajectories are learned from historical trajectories provided as "expert" demonstrations, considering that these trajectories have been "shaped" by aggregating stakeholders' policies, preferences and objectives.

- State of the art IL methods have been studied, towards learning trajectory models without making any assumption on the form of a cost function, in continuous state-action spaces, with no specific requirements on specifying trajectory constraints (e.g. without requiring information on flight plans), and with minimal data pre-processing requirements.

- Extensive experimental results are provided concerning trajectories between OD airports' pairs with different characteristics, demonstrating the prediction abilities of the method, either at the pre-tactical or at the tactical stage of operations.

1.2.2 Data-driven modeling of the ATCOs' behavior in resolving conflicts.

This study contributes to Conflict Detection and Resolution (CD&R) tasks executed as part of the ATC service, promoting safe, orderly and expeditious flow of air traffic, by modeling ATCOs' behavior in resolving conflicts using data-driven AI/ML techniques. In general, according to the problem specifications made in this study, this implies learning "when" the ATCO will react to resolve a detected conflict, and "how" he/she will react: The first is the ATCO reaction problem specifying "whether" and "when" the ATCO will react , while the second is the problem of learning the ATCO policy, specifying "how" he/she will react in the presence of conflicts. The proposed methodology can be used either at the pre-tactical or at the tactical phase of operations. However, considering the current operations, performing the CD&R task at the tactical phase is more realistic, as it avoids different factors of uncertainty of the pre-tactical phase, such as delays of flights' take off time. Thus, when considering the CD&R task, this study focuses at the tactical phase of operations. Regarding the flight phase when considering the CD&R task, the focus of this study is at the en route phase of flights and does not consider CD&R operations at the Terminal Maneuvering Area (TMA) as the ATCO behavior at the TMA is different.

The specific contributions made towards the ATCO reaction problem are as follows:

- The problem of CD&R has been formulated as an IL problem, aiming to learn ATCO behavior in a hierarchical manner. In so doing, the ATCO reaction prediction problem is formulated.

- A supervised deep learning method employing a Variational Auto-Encoder (VAE) for predicting ATCO reactions has been devised, in the context of a methodology to model ATCOs

behavior;

- A data-driven method for simulating the uncertainty in the evolution of trajectories and for detecting the potential conflicts that may have triggered ATCOs reactions (this is a challenging issue due to inherent data sources limitations), has been proposed;
- A methodology for evaluating data-driven methods to resolve the ATCO reaction problem has been devised, taking into account uncertainties involved in the process;
- The proposed method has been evaluated comparatively with baseline methods towards modeling ATCOs reactions, using real world data.

Regarding the problem of learning the ATCO policy, the contributions made towards this objective are as follows:

1. The problem of learning the ATCO policy is specified as a supervised IL task. Considering specific types of resolution actions that may be applied in the en route phase of flights at the tactical phase of operations, this results in a classification task.
2. Alternative AI/ML methods to learn models of ATCOs' behavior with respect to the formulation proposed are considered. Also a single stage episodic IL method based on the Generative Adversarial Imitation Learning (GAIL) is presented in the Appendix.
3. The proposed AI/ML methods are evaluated using real-world data, addressing the challenges to imitating ATCOs adequately.

Indeed, data-driven techniques for conflict resolution have the potential to reveal and incorporate in the decision-making process the preferred behavior of the various stakeholders, as this information lies implicit in the demonstrated historical data, and is being represented in a machine-crafted model, learned by exploiting the appropriate data sources.

A challenging issue of such a data-driven imitation process, as experienced by this study, is that historical expert samples (i.e. flown trajectories annotated with ATCO resolution actions) do not indicate, together with the resolution actions, the observations perceived by ATCOs before the resolution action, driving the specific action. Such observations include features concerning the evolution of the trajectories perceived/assessed by the ATCO before their "intervention", the features of conflicts assessed, as well as the evolution of conflicts after the instruction of a resolution action. However, historical data sets indicate in the best case the effect of ATCO resolution actions, but neither the potential evolution of the trajectories before the resolution action, nor how trajectories would evolve if the ATCO resolution action had not been applied. This is a challenging issue in the learning process, since imitating the "when" and "how" of the ATCO behavior necessitates recovering the specific state, and the important observations that the ATCO perceived or predicted, driving decisions. This is in contrast to detecting conflicts. This work exploits historical data to assess conflicts that may have occurred, and which caused the ATCOs' reactions. This is a rather challenging issue that is addressed and discussed in this study to a large extent.

1.2.3 Conflicts-free trajectory planning.

This study aims to answer if and to what extent the presented methods for trajectory prediction and CD&R suffice for creating a method for planning conflicts-free trajectories. To do so, this study presents a straightforward way of combining the models for trajectory prediction and CD&R into a unified approach for planning conflicts-free trajectories. It proceeds to evaluate

the resulting method using real world data.

The specific contributions towards this objective are as follows:

- The problem of conflicts-free trajectory planning is specified as a data-driven problem.
- A purely data driven approach, that exploits the trained independent models for trajectory prediction and for modeling the ATCOs' behavior and combines them in a sequential manner is presented.
- The resulting method is evaluated using real world data revealing challenges and problems to be addressed in the future regarding needed data and ways to combine ATCO models towards planning conflicts-free trajectories.

Concluding this section, this study addresses the following challenges:

1. Plan trajectories, considering complex ATM phenomena and operational constraints regarding traffic and conflicts among trajectories.
2. Follow a data-driven approach to learn stakeholders' preferences on the evolution of trajectories and on resolving conflicts: Stakeholders include airspace users (for trajectory prediction) and ATCOs (for CD&R actions).
3. Address optimization in trajectory planning w.r.t. multiple objectives, preferences and constraints of stakeholders involved, as these are demonstrated by historical data.

Overall the contributions that this study makes are as follows:

1. The problem of modeling ATCOs' behavior has been split into two well-defined problems: Modeling ATCOs' reactions on whether and when conflicts' resolution actions should be applied, and modeling ATCOs' policy on how conflicts should be resolved, i.e. what resolution actions should be applied.
2. The problem of trajectory planning (either with or without considering conflicts) has been formulated as an IL problem, based on historical flown trajectories.
3. AI/ML methods have been developed and tested on learning models regarding the evolution of 4D trajectories, using data-driven approaches, i.e. based on historical real-world data.
4. AI/ML methods have been developed and tested on learning models regarding ATCOs' reactions and policy using data-driven approaches, i.e. based on historical real-world data.
5. This study has proposed an elaborated evaluation method for data-driven IL techniques predicting ATCOs reactions, considering the uncertainties involved in the evolution of trajectories, in the assessment of conflicts, and in the reactions of ATCO.
6. Challenging issues due to inherent data limitations have been addressed and thoroughly discussed.
7. This study presents a data driven trajectory planning approach, where models for trajectory prediction and for modeling the ATCOs' behavior are combined in a sequential manner, revealing challenges and problems to be addressed in the future regarding needed data and ways to combine ATCO models towards conflicts-free trajectories.

The above contributions impact HCAI in the following ways:

- This study models preferences and policies of multiple stakeholders (i.e. airspace users and ATCOs), shaping a common artifact (i.e. the flight trajectory) in an aggregated way. This is addressed in contributions 2 and 3.
- It studies the effectiveness of different methods to model preferences and policies of human operators given historical demonstrations recording their decisions, without considering explicitly their actual observations. This is considered in contributions 1, 4 and 6.
- It considers how to evaluate the efficacy of a model that tries to approximate as close as possible human operators' behavior, considering the uncertainty of human behavior. This concerns contribution 5.

1.3 Thesis Structure

This thesis is structured as follows. The first part provides background knowledge on machine learning algorithms used in this thesis and on the CD&R task. Part II presents the methods comprising the main contribution of thesis. Specifically chapter 3 presents advances made on AI/ML methods incorporating Reinforcement Learning (RL) for the prediction of trajectories per OD pair, without explicitly considering conflicts. Chapter 4 presents AI/ML methods for the detection and resolution of conflicts and chapter 5 presents an AI/ML method for planning conflicts-free trajectories combining methods presented in chapters 3 and 4. Finally part III concludes this thesis and provides directions for future work.

Chapter 2

Background

This chapter discusses preliminary knowledge regarding a) the AI/ML methods used in this thesis and b) domain knowledge relative to the problem of conflicts-free trajectory planning.

2.1 Machine Learning

ML is the field of AI, that studies algorithms according to which machines can perform different tasks, without being explicitly programmed to do so using predefined rules or instructions, but instead by observing their decisions and improving them according to a function measuring their performance.

There are three main categories of machine learning algorithms: supervised learning, unsupervised learning and RL. While supervised methods aim to perform specific tasks based on labeled datasets, containing the desired output, unsupervised learning methods discover patterns in unlabeled data. On the other hand RL methods study how autonomous entities called agents, learn to perform complex tasks by interacting with their environment, and improving their decisions based on a reward function that evaluates their actions. This thesis studies mostly supervised and RL methods. The next sections present and discuss the ML methods explored in this thesis.

2.1.1 Supervised Learning

Supervised learning refers to machine learning algorithms that learn to perform specific tasks, such as classification and regression, using labeled datasets. Such datasets contain a) observations, regarding features based on which we aim to predict a target variable, and also b) the desired value of the target variable, called label.

More formally, given a set S_r of feature vectors and the set A of the corresponding labels, supervised learning algorithms learn to predict the target a given the feature vector s_r. a can be a continuous valued number in case of regression tasks, while in classification tasks it is an indicator of the class in which the sample described by the feature vector s_r belongs. Usually in the literature feature vectors and targets are denoted with x and y respectively. In this thesis feature vectors correspond to trajectory states and model outputs to specific actions and thus this notation is followed throughout the whole thesis.

In this thesis supervised learning is mainly used to perform classification tasks for learning "when" the ATCO will react to resolve a detected conflict and "how" he/she will react. Next the

supervised learning algorithms used in this thesis are presented.

2.1.1.1 Neural Networks

Neural Networks (NNs) [8] are function approximators able to model complex non-linear functions, and have been applied with great success in many regression and classification problems, as well as for imitating experts' behavior using behavior cloning [16, 61] and modeling complex agents' policies as will be discussed in section 2.1.3.

NNs are inspired by the human brain. NNs consist of nodes or neurons interconnected in a layered structure that resembles how biological neurons communicate with each other. Figure 2.1 shows an example of a Neural Network (NN), depicting the input, hidden and output layers. In the simplest case, each node computes a weighted sum of its inputs, applies an activation function and passes the output to the next layer. A NN is trained using gradient descent [3, 65, 10], tuning its learnable parameters towards optimizing a loss function based on the training samples provided.

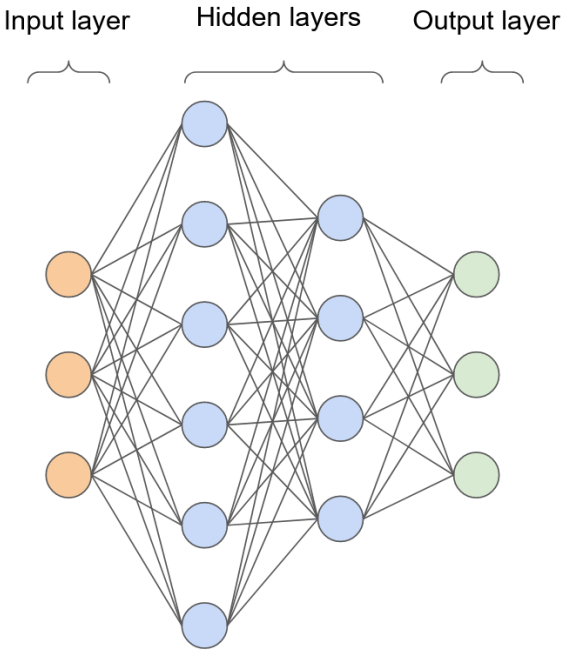


Figure 2.1: Example of a NN, showing the input, hidden and output layers.

Different NN architectures and different types of layers, activation and loss functions are used depending on the task at hand. Recurrent NNs have been applied with great success on time series data, examples include Natural Language Processing (NLP) tasks such as machine translation and audio processing tasks including speech recognition. Convolutional NNs have achieved great results when applied on tasks involving images, for example computer vision tasks including image classification and object detection. Augmenting NN models with attention mechanisms has helped to model interactions between different input components, while also providing explainability. For example in NLP tasks, attention mechanisms can model relations between words capturing the context in which words are used, while in RL tasks attention mechanisms can model interactions between different agents.

NN models can be applied on supervised and unsupervised tasks. For example simple NN models can perform classification using labeled datasets while other NN variants such as Auto-Encoders exploit the embeddings learned by the hidden layers of the model using unlabeled

datasets to perform dimensionality reduction, de-noising or even tackle generative tasks such as image generation.

In this thesis NN models are used mostly in supervised learning tasks and thus are presented under section 2.1.1. An exception is the Variational Auto-Encoder (VAE) model which is presented in section 2.1.2.1.

2.1.1.2 Decision Trees

Decision Trees (DTs) [46] are models with a tree-like structure used for classification and regression.

DTs make predictions by using sequences of rules exploiting the input features. At each step, a rule regarding a specific feature is tested and the answer determines the next rule that will be tested, creating a tree-like structure of rules where rules correspond to decision nodes, shown in Figure 2.2.

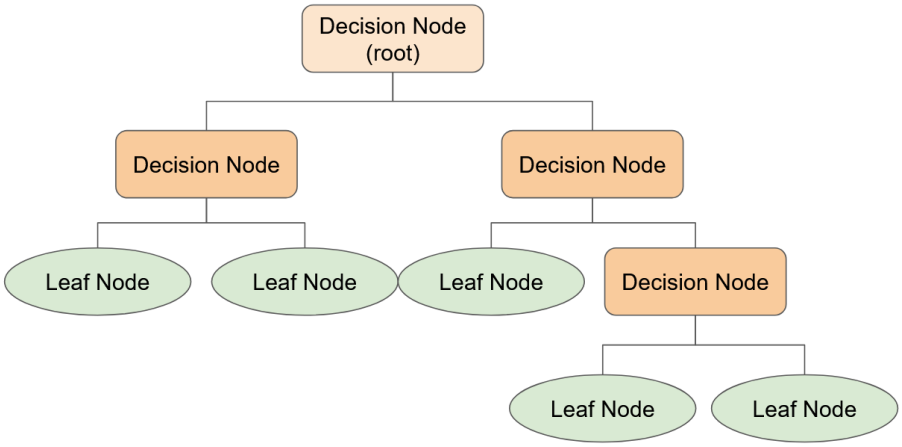


Figure 2.2: Example of a decision tree, showing the decision nodes corresponding to conditions, based on which leaf nodes corresponding to different partitions of the data space are formed.

Rules are inferred based on the training samples. Algorithms for creating DTs must determine the rules that best divide the training instances in separate partitions, corresponding to leaf nodes in Figure 2.2. To do so, splits of the training samples produced by potential rules are assessed by a gain function. The gain can be expressed using different criteria, e.g., Gini in case of classification trees or Mean Squared Error in case of regression trees, and each time the rule with the maximum gain is selected. Samples are split until leaves are pure, containing samples of the same target value, or until leaves contain the minimum number of samples.

To provide predictions using a Decision Tree (DT) given a specific input, first the leaf node at which the input corresponds must be determined by traversing the decision nodes, testing which rules apply to the input features. Then in case of regression trees the prediction can be computed as the mean or the median target value of the training samples in the leaf node. In this thesis DTs are used for classification and DTs provide the probability of each class for a given input. This is predicted by testing which rules apply to the input features, until reaching a leaf node and then calculating the fraction of training samples of the class that corresponds to that leaf.

As discussed the structure of the tree determines how the training instances are divided into different partitions. Thus it determines the effectiveness of the tree in terms of the accuracy of the predictions made. Deep trees with many rules may overfit the training set and fail to generalize properly, whereas small trees may underfit the training set, providing inaccurate predictions.

2.1.1.3 Random Forest

Random Forest (RF) [11] is an ensemble of DTs [46] trained individually on the training set to perform classification or regression tasks. For classification tasks, as are the tasks on which RF is applied in this thesis, the output class is decided either by voting, selecting the class predicted by most trees, or by using an average of the predicted probabilities for each class. In this thesis, the prediction is computed as the average predicted probability for each class of the DTs.

Trees are trained on a subset of the training set built by drawing samples with replacements. This is known as bootstrapping and it results in reducing the variance at the cost of increasing the bias. Another technique used to reduce the variance of a RF model is the random input selection, according to which nodes are split during the construction of the trees using a random subset of the input features. In this thesis both techniques are used.

2.1.1.4 Gradient Boosting

Gradient Boosting (GB) [24] is a machine learning method that constructs an additive model consisting of the weighted sum of multiple base models called base learners. More formally, the model learned using the gradient boosting method is of the form $F(s_r) = \sum_{l=0}^{IT} \beta_l^{GB} h^{GB}(s_r; p_l)$, where s_r is the set of input variables , $h^{GB}(s_r; p_l)$ is the base learner functions with learnable parameters p_l^{GB}, and β_l^{GB} represents learnable expansion coefficients.

GB starts with a simple initial guess for $F_0(s_r)$, usually a constant function, and optimizes the following objective:

$$(\beta_l^{GB}, p_l^{GB}) = \underset{\beta^{GB}, p^{GB}}{\arg\min} \sum_{i=1}^{NoOfSamples} \Psi(a_i, F_{l-1}(s_{r_i}) + \beta^{GB} h^{GB}(s_{r_i}, p^{GB})) \tag{2.1}$$

with $F_l(s_r) = F_{l-1}(s_r) + \beta_l^{GB} h^{GB}(s_r; p_l^{GB})$ for $l = 1, ..., IT$, where Ψ denotes a loss function, a_i the true output value corresponding to s_{r_i}, and IT the number of iterations.

2.1.2 Unsupervised Learning

Unsupervised learning refers to machine learning algorithms that learn patterns from unlabeled data. Common tasks on which unsupervised learning algorithms are applied to, include clustering, feature extraction and also generative tasks where models learn to generate samples from the distribution underlying the available data. Next Variational Auto-Encoders (VAEs), originally designed as generative models are introduced and discussed.

2.1.2.1 Variational Auto-Encoders

Auto-Encoders are NN models trained to reconstruct the input to their output. Internally they can be broken down into two parts: an encoder network and a decoder network. The encoder network, comprising a number of hidden layers, maps the input s_r to an encoding c, which can be denoted as $c = q_\phi(x)$, where ϕ are the parameters of the encoder network. The decoder

network maps the encoding c to a reconstruction of the input $rc = p_\theta(c)$, where θ are the decoder parameters. Auto-encoders do not merely learn to reconstruct the input. They learn an encoded representation c of the input s_r, which retains enough information to allow a reconstruction rc (e.g. for dimensionality reduction or feature learning in [47], [36]). Auto-Encoders have been explored for more than three decades, with recent advances applying auto-encoders to image de-noising [84], anomaly detection [66], information retrieval [48] and generative tasks (i.e. image captioning [62]).

VAEs [44] are a generative variant of auto-encoder models, successfully applied to generative tasks: The encoder of a VAE outputs the parameters of a distribution $q_\phi(c|s_r)$ approximating the true intractable posterior $p(c|s_r)$. The decoder samples c from $q_\phi(c|s_r)$ and outputs a reconstruction rc. Therefore, VAEs learn an approximation $q_\phi(c|s_r)$ of the true intractable posterior $p(c|x)$, represented by the encoder, and a generative model $p_\theta(rc|c)$, represented by the decoder. To do so VAEs maximize the lower bound:

$$L(\theta, \phi; s_r) = -D_{KL}(q_\phi(c|s_r)||p(c)) + \mathbb{E}_{q_\phi(c|s_r)} log(p_\theta(s_r|c)) \tag{2.2}$$

,where $D_{KL}(q_\phi(c|s_r)||p(c))$ denotes the Kullback–Leibler (KL) divergence between the distributions $q_\phi(c|s_r)$ and $p(c)$.

Although originally used for generative tasks, the approximation of the posterior $p(c|s_r)$ learned by VAE models has been proven useful in other ways, as for example done by Directed InfoGAIL [71], in the context of imitation learning, where it represents modes that underlie the demonstrated behavior. This will be further discussed in section 2.1.4.

2.1.3 Reinforcement Learning

RL is the machine learning paradigm, where autonomous entities, called agents, learn to perform specific tasks, by interacting with their environment and observing a reward signal informing them how desirable the outcome of these interactions was.

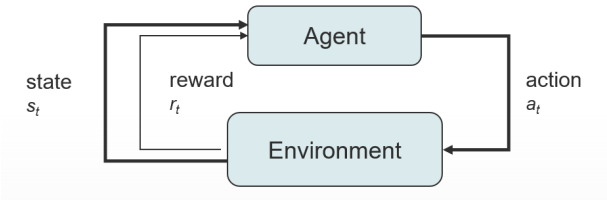


Figure 2.3: The RL scheme modeling interactions between the agent and its environment.

More specifically, as shown in figure 2.3 considering the environment's state s_t at time point t, the agent observes s_t and takes an action a_t. Given the state s_t and the action a_t the environment transitions to the next state s_{t+1}, and returns a reward r_t to the agent. The agent's goal is to learn a policy π that maximizes the expected-cumulative-reward. The policy constitutes the agent's decision making mechanism, modeling the conditional probability distributions of the set of agent's actions over the set of states.

2.1.3.1 Markov Decision Process

Formally RL problems are modeled as Markov Decision Processes (MDPs). MDPs are defined as a tuple $\langle S, A, P, R, \gamma \rangle$ where:

- S is a set of states.

- A is a set of actions.
- $P : S \times A \rightarrow S$ is a transition function defining the probability distribution $P(s_{t+1}|s_t, a_t)$ of reaching state $s_t \in S$ given state $s_{t+1} \in S$ given action $a_t \in A$.
- $R : S \times A \rightarrow \mathbb{R}$ is a reward function returning a real valued number given $s_t \in S$ and $a_t \in A$.
- γ is a discount functor $\in [0, 1]$ weighting future rewards.

As discussed the agent's goal is to maximize its expected-cumulative-reward. More specifically, considering a discount factor γ the agent's goal is to learn a policy π that maximizes the expected-cumulative-discounted-reward starting from any state $s \in S$ and following π. Formally the agent optimizes the following objective:

$$\arg\max_{\pi} \mathbb{E}_{\pi}[\sum_{k=0}^{\inf} \gamma^k R(s_{t+k}, a_{t+k})|s_t = s] \tag{2.3}$$

The expected-cumulative-discounted-reward starting from any state $s \in S$ and following π, $\mathbb{E}_{\pi}[\sum_{k=0}^{\inf} \gamma^k R(s_{t+k}, a_{t+k})|s_t = s]$, is called the value function of state s under policy π, denoted $V_{\pi}(s)$ and $\sum_{k=0}^{\inf} \gamma^k R(s_{t+k}, a_{t+k})$ is the future-cumulative-discounted reward or the return, G. Thus the value function of state s under policy π is defined as:

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t|s_t = s] = \mathbb{E}_{\pi}[\sum_{k=0}^{\inf} \gamma^k R(s_{t+k}, a_{t+k})|s_t = s] \tag{2.4}$$

Similarly to the definition of the state-value function v, the value of action a in state s under policy π, $Q^{\pi}(s, a)$ is defined as the expected return of taking the action a at state s, and thereafter following policy π:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[G_t|s_t = s, a_t = a] = \mathbb{E}_{\pi}[\sum_{k=0}^{\inf} \gamma^k R(s_{t+k}, a_{t+k})|s_t = s, a_t = a] \tag{2.5}$$

$Q^{\pi}(s, a)$ is called the action-value function for policy π.

Optimal policies, policies that maximizes equation 2.3 denoted by π^*, the optimal state-value function denoted by V^* and the optimal action-value function denoted Q^* are defined as:

$$V^*(s) = \max_{\pi} v_{\pi}(s), \ \forall s \in S. \tag{2.6}$$

$$Q^*(s, a) = \max_{\pi} Q_{\pi}(s, a), \ \forall s \in S \text{ and } a \in A. \tag{2.7}$$

Assuming a discrete action space, when Q^* is known an optimal policy can be obtained by assigning a probability of 1 to the action with the maximum Q^* value and a probability of 0 (zero) to other actions. Formally,

$$\pi^*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_{a \in A} Q^*(s,a) \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

Thus many RL methods, e.g. Q-learning, rely on approximating the optimal action values in order to approximate an optimal policy.

2.1.3.2 Deep Reinforcement Learning

Traditional RL methods relied on tabular structures in order to represent functions, such as the action-value function Q and solve RL problems. Such approaches become inapplicable for high dimensional or continuous state spaces (e.g. camera images) as the tabular representation of the functions becomes infeasible.

Deep Reinforcement Learning (DRL) techniques combine RL with deep learning, utilizing deep NNs to approximate useful functions i.e. the policy π or the action-value function Q. By doing so, they can handle high dimensional state-action spaces while requiring less manual feature engineering than traditional RL methods.

2.1.3.2.1 Policy Gradient methods and Trust Region Policy Optimization

In the DRL context policy gradient methods refer to methods that directly optimize the policy NN using gradient descent. More specifically in the DRL context considering an episodic task, having at least one terminal state, and a parametarized policy with parameters θ the agent's goal presented in 2.3 can be written as:

$$\arg\max_{\theta} J(\pi_\theta) = \arg\max_{\theta} \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)] \tag{2.9}$$

where τ denotes state-action trajectories from initial to terminal states sampled from the policy π_θ.

Policy gradient methods solve this problem using gradient ascent on the policy parameters θ using the policy gradient:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{\infty} \nabla_\theta log \pi_\theta(a_t|s_t) \sum_{t'=t}^{\infty} \gamma^{t'-t} R(s_{t'}, a_{t'})] = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{\infty} \nabla_\theta log \pi_\theta(a_t|s_t) \hat{Q}^{\pi_\theta}(s_t, a_t)] \tag{2.10}$$

where $\hat{Q}$ denotes an approximation of the action value function Q.

Policy gradient methods, e.g. reinforce [82], using the policy gradient update of equation 2.10 suffer from high variance in the reward signal and slow convergence. To reduce variance, policy gradient methods subtract a baseline term from the reward [32]. In practice a good baseline is the state-value function V resulting to the following policy gradient update.

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{\infty} \nabla_\theta log\pi_\theta(a_t|s_t)(\hat{Q}^{\pi_\theta}(s_t,a_t) - \hat{V}(s_t)) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{\infty} \nabla_\theta log\pi_\theta(a_t|s_t)\hat{A}^{\pi_\theta}_{DV}(s_t,a_t)] \tag{2.11}$$

with $\hat{A}^{\pi_\theta}_{DV}$ denoting an approximation of the advantage function for policy π_θ.

Other problems of the policy gradient update presented in equation 2.11 are:

- Update is performed at the parameter space not the policy space. Thus, small changes at the parameter space could lead to large changes at the policy space making learning unstable.
- Poor sample efficiency: Presented policy gradient update is on-policy, meaning that for each update samples from the current policy should be gathered making older samples unusable.

Considering Policy Gradient as policy iteration and optimizing the objective $J(\pi_\theta) - J(\pi_{\theta_{old}})$ leads to sample efficient methods that guarantee to improve the policy. Specifically policy improvement can be expressed as maximizing $J(\pi_\theta) - J(\pi_{\theta_{old}})$. As shown in [42]:

$$J(\pi_\theta) - J(\pi_{\theta_{old}}) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t \hat{A}^{\pi_{\theta_{old}}}_{DV}(s_t,a_t)] = \sum_{t=0}^{\infty} \mathbb{E}_{s_t\sim\pi_\theta}[\mathbb{E}_{a_t\sim\pi_\theta}[\gamma^t \hat{A}^{\pi_{\theta_{old}}}_{DV}(s_t,a_t)]] \tag{2.12}$$

Using importance sampling this can be expressed as:

$$J(\pi_\theta) - J(\pi_{\theta_{old}}) = \sum_{t=0}^{\infty} \mathbb{E}_{s_t\sim\pi_\theta}[\mathbb{E}_{a_t\sim\pi_{\theta_{old}}}[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\gamma^t \hat{A}^{\pi_{\theta_{old}}}_{DV}(s_t,a_t)]] \tag{2.13}$$

Bounding the distribution change using the KL divergence D_{KL}, allows sampling states from π_{old} for a small enough bound, leading to the following objective:

$$\begin{aligned} \arg\max_\theta L_{\theta_{old}}(\theta) = \arg\max_\theta \sum_{t=0}^{\infty} \mathbb{E}_{s_t\sim\pi_{\theta_{old}}}[\mathbb{E}_{a_t\sim\pi_{\theta_{old}}}[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\gamma^t \hat{A}^{\pi_{\theta_{old}}}_{DV}(s_t,a_t)]] \\ \text{subject to } D_{KL}(\pi_{\theta_{old}}||\pi_\theta) <= \epsilon \end{aligned} \tag{2.14}$$

This guarantees policy improvement for small enough ϵ and also allows multiple gradient steps using samples from the same policy.

Denoting matrix transposition with the T superscript and using a linear approximation of the objective:

$$\begin{aligned} L_{\theta_{old}}(\theta) = L_{\theta_{old}}(\theta_{old}) + g^T(\theta - \theta_{old}) \\ g = \nabla_\theta L_{\theta_{old}}(\theta)|_{\theta_{old}} \end{aligned} \tag{2.15}$$

and a quadratic approximation of the constraint:

$$
\begin{aligned}
D_{KL} &= \frac{1}{2}(\theta - \theta_{old})^T H(\theta - \theta_{old}) \\
H &= \nabla^2_\theta D_{KL}(\theta || \theta_{old})|_{\theta_{old}}
\end{aligned}
\tag{2.16}
$$

the constraint optimization problem is approximated by:

$$
\begin{aligned}
\theta &= \arg\max_\theta g^T(\theta - \theta_{old}) \\
\text{such that } &\frac{1}{2}(\theta - \theta_{old})^T H(\theta - \theta_{old}) <= \epsilon
\end{aligned}
\tag{2.17}
$$

Solving this constraint optimization problem results to the Natural Policy Gradient (NPG) update:

$$
\theta = \theta_{old} + \sqrt{\frac{2\epsilon}{g^T_{old} H^{-1}_{old} g_{old}}} H^{-1}_{old} g_{old} \tag{2.18}
$$

Using the conjugate gradient method to compute $H^{-1}g$ resulted to the truncated NPG method [20]. The conjugate gradient method used in this study is reported in section A.3 of the appendix.

The Trust Region Policy Optimization (TRPO) method solves the constraint optimization problem presented in equation 2.14 by using a linear approximation of the objective and a quadratic approximation of the constraint to compute the gradient direction, also applying line search in that direction to ensure improvement of the optimization objective while satisfying the constraint. This amends the following two problems of the NPG methods, showing significant performance improvements on large problems:

- For some iteration, ϵ might be too large, allowing big changes in the policy space hindering performance.
- Because NPG methods use a quadratic approximation for D_{KL}, the D_{KL} constraint might be violated.

Algorithms 1, 2 present the linesearch and the TRPO algorithms respectively.

Algorithm 1: Linesearch for TRPO

Compute policy step $\sqrt{\frac{2*\epsilon}{\chi^T H \chi}}\chi$ with $\chi = H^{-1}g$
for *j=0,1,...,K* **do**
    Compute proposed update $\theta = \theta_{old} + a^j \sqrt{\frac{2*\epsilon}{\chi^T H \chi}}\chi$
    if $L_{\theta_{old}}(\theta) >= 0$ *and* $D_{KL}(\pi_{\theta_{old}} || \pi_\theta) <= \epsilon$ **then**
        accept the update and set $\theta = \theta_{old} + \mu^j \sqrt{\frac{2*\epsilon}{\chi^T H \chi}}\chi$
        break

Algorithm 2: TRPO algorithm

while *iteration = 0,1,...,* **do**
- Run policy for T timesteps or N trajectories
- Estimate advantage function at all timesteps
- Compute policy gradient g
- Use the conjugate gradient method to compute $\chi = H^{-1}g$
- Update the policy parameters by backtracking linesearch with
- $\theta = \theta_{old} + \mu^j \sqrt{\frac{2*\epsilon}{\chi^T H \chi}} \chi$
- where $j \in 0, 1, ..., B_M$ is the smallest value improving the policy and satisfying the D_{KL} constraint, μ is the backtracking coefficient and B_M is the maximum number of backtracking steps.

In this thesis TRPO is used as part of the Generative Adversarial Imitation Learning (GAIL) algorithm presented in 2.1.4.3 to update the agent's policy.

2.1.3.2.2 Actor Critic Methods and Generalized Advantage Estimation

Actor critic methods use policy gradient to update the policy network representing the actor but also include a critic network that informs the actor about the quality of each actions in terms of the return G. Specifically the critic network approximates the state-value function based on which the advantage of function, used in the policy update, can be computed. Specifically, the advantage is defined as

$$A^{\pi}_{DV}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t) \tag{2.19}$$

Which can be approximated by

$$\hat{A}^{\pi}_{DV}(s_t, a_t) = R(s_t, a_t) + \hat{V}^{\pi}(s_{t+1}) - \hat{V}^{\pi}(s_t) \tag{2.20}$$

The benefit here is that a critic which learns the V function is enough to estimate the advantage. This approach results in low variance, but introduces a bias. A way to use $\hat{V^{\pi}}$ without introducing bias is to subtract it from the reward sum as a baseline

$$\mathbb{E}_{\tau \sim \pi}[(\sum_{t'=t}^{\infty} \gamma^{t'-t} R(s_{t'}, a_{t'})) - \hat{V}_{\pi}(s_t)] \tag{2.21}$$

This method is unbiased, but has higher variance than 2.20. Generalized Advantage Estimation (GAE) is a method introduced in [69], which provides a trade-off between 2.20 and 2.21, controlled by a hyperparameter λ.

$$\hat{A}^{\pi}_{GAE} = \sum_{t=t'}^{t=T} (\gamma\lambda)^{t'} \delta_t \tag{2.22}$$

where

$$\delta_t = R(s_t, a_t) + \gamma \hat{V}_{\pi}(s_{t+1}) - \hat{V}_{\pi}(s_t)$$

The two notable cases of this formula are obtained by setting $\lambda = 0$ and $\lambda = 1$, where 2.22 becomes equal to 2.20 and 2.21 respectively.

In this thesis GAE is used to estimate the advantage used in the TRPO update.

2.1.4 Imitation Learning

IL aims at learning policies that mimic expert behavior from demonstrations. Although close to RL, in IL the agent does not have access to a handcrafted explicit reward modeling the desired task, but to a set of examples demonstrating the desired behavior. This is a benefit in many real world problems where handcrafting a reward function constitutes a challenging task and there is access to expert demonstrations. There are three main approaches to imitation learning: a) Behavioral Cloning (BC), b)Inverse Reinforcement Learning (IRL) and c) Adversarial Imitation Learning (AIL).

2.1.4.1 Behavioral Cloning

BC aims to learn a model of the expert policy by maximizing the likelihood of the model parameters given the expert demonstrations using supervised ML methods. In BC, learning is based only on the set of expert samples and thus it is likely that the agent will perform poorly when encountering states that are not close to those demonstrated. This is more evident when dealing with state-action trajectories, as small errors in the agent's actions accumulate, leading the agent to states out of the distribution of demonstrated states.

DAgger, tries to fix the problem of the distribution mismatch, by collecting new expert data as needed. Specifically it involves human experts in the training loop to propose actions on states visited by the agent's policy. But, such interactive access to a human expert is time consuming and usually infeasible.

2.1.4.2 Inverse Reinforcement Learning

IRL approaches IL by approximating the reward function that is implicitly described through the expert demonstrations. By doing so the expert's goals and objectives are revealed through the reward function and the agent can learn the expert behavior by using RL methods. The ability of IRL methods to retrieve the expert's reward function has demonstrated the following benefits according to recent studies: a) the RL agent can further optimize the learned task using the retrieved reward function, leading to superior performance compared to the demonstrated behavior, b) the agent can learn a policy that maximizes the retrieved reward according to its own capabilities and morphology which could be different from the expert's.

Early IRL methods such as [1] assume linear w.r.t. the state's features reward models and used RL to fully optimize the agent's policy at every update of the reward model making them extremely expensive computationally. Specifically at each iteration, early IRL methods perform the following two steps: a) find a reward function such that the expert performs better than any of the agent's policies by the biggest possible margin according to a distance function b) find a policy that performs optimally w.r.t. that reward function. Recent IRL approaches such as [23] have raised the linearity assumption, learning complex reward functions represented by NN models and also do not need to train the policy until convergence at each update, increasing their computational efficiency. Figure 2.4 shows the training procedure of recent IRL methods.

2.1.4.3 Adversarial Imitation Learning

In [37] the authors formulated IRL as a min-max game between the policy and the reward models. In the proposed method called Generative Adversarial Imitation Learning (GAIL) the agent's policy tries to maximize the reward received from the reward model, while the reward model tries to distinguish state-action samples produced by the policy from state-action samples from the expert demonstrations. This is analogous to generative adversarial networks [31]

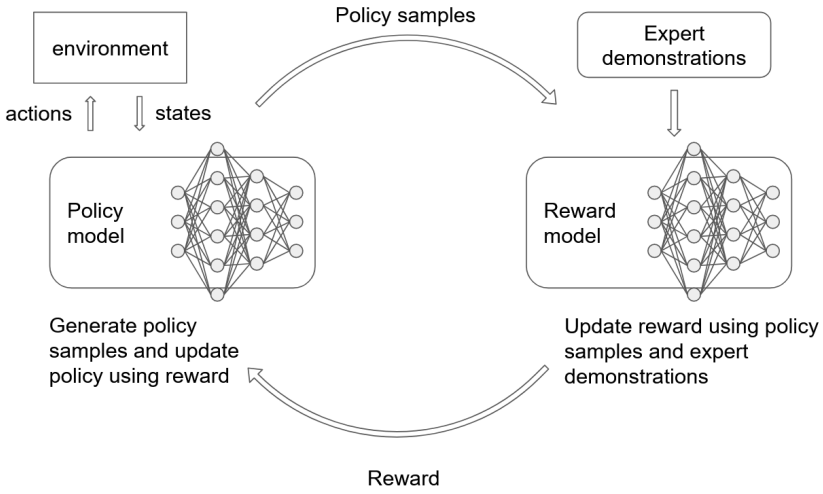


Figure 2.4: Inverse Reinforcement Learning (IRL) training loop.

with the policy model playing the role of the generator and the reward model playing the role of the discriminator.

Formally, GAIL optimizes the following objective:

$$\min_{\theta} \max_{w} \hat{\mathbb{E}}_{\pi_\theta}[log(D_w(s,a))] + \hat{\mathbb{E}}_{\pi_E}[log(1 - D_w(s,a))] - \lambda_H(H(\pi_\theta)) \tag{2.23}$$

where θ denotes the agent policy's weights, w denotes the weights of the reward model, π_θ the agent's policy, π_E the expert's policy, λ_H a regularization term and $H(\pi_\theta)$ the entropy of the agent's policy. As seen in equation 2.23, inspired by maximum entropy IRL, GAIL maximizes the agent policy's entropy preferring policies that are as "uncommitted" as possible. This preference solves the ambiguity introduced by the fact that there many policies that correspond to the demonstrated behavior and some of them could show arbitrary preference, i.e., unrelated to the imitation objective, for specific state action trajectories.

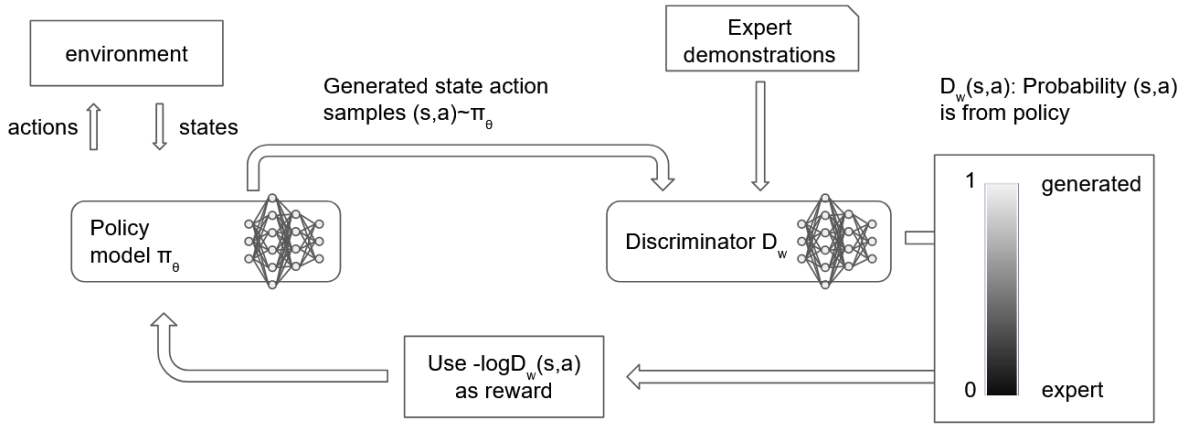


Figure 2.5: GAIL architecture.

In practice GAIL alternates between a reward update step maximizing:

$$\hat{\mathbb{E}}_{\tau_i}[log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[log(1 - D_w(s,a))] \tag{2.24}$$

with respect to w and a policy update step using TRPO minimizing:

$$\hat{\mathbb{E}}_{\tau_i}[-\nabla_\theta log\pi_\theta(a|s)Q(s,a) - \lambda_H \nabla H(\pi_\theta)] \tag{2.25}$$

with respect to θ, where Q denotes the action-value function computed using $-logD_w(s,a)$ as the reward.

Algorithm 3: GAIL

Input: Expert trajectories $\tau_E \sim \pi_E$, initial policy π_{θ_0} and discriminator parameters w_0
Output: Policy π_θ
for *i=0,1,2,...* **do**
  Sample trajectories $\tau_i \sim \pi_{\theta_i}$
  Update the discriminator's parameters w by the ascending the gradient:
    $\hat{\mathbb{E}}_{\tau_i}[\nabla_w log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w log(1 - D_w(s,a))]$
  Estimate advantages $\hat{A}^{\pi_{\theta_{old}}}_{GAE}$, according to $\pi_{\theta_{old}}$
  Take a KL-constrained natural gradient descent step using TRPO with
    $\hat{\mathbb{E}}_{\tau_i}[-\nabla_\theta log\pi_\theta(a|s)Q(s,a) - \lambda_H \nabla H(\pi_\theta)]$,
    where Q denotes the action-value function computed using $-logD_w(s,a)$ as the reward function.

Algorithm 7 presents the GAIL algorithm.

GAIL is a sample efficient method able to reach expert performance using few demonstrations in complex environments. Techniques based on this adversarial game between the policy and the reward models are referred to as adversarial imitation.

2.1.4.3.1 Directed InfoGAIL

Many real-world problems can be modeled as hierarchical tasks where an agent performs different high-level subtasks in sequence. These subtasks are realized by primitive or low-level actions. Directed InfoGAIL [71], extends GAIL by introducing high level latent variables that control the low-level actions modeling intra-trajectory sub-task variations. In practice Directed InfoGAIL automatically discovers and disentangles such intra-trajectory sub-task variations underlying the expert demonstrations.

To do so, this method corresponds a latent variable c to sub-tasks variations within a demonstration. Given a latent code the agent's policy produces actions according to the distribution of a specific sub-task corresponding to this latent code. Latent codes can be considered in a hierarchical manner as high-level actions, initiating sub-tasks that are realized by executing low level policy actions. In a more technical level, Directed InfoGAIL, forces the policy to generate trajectories that maximize the directed or causal information flow from trajectories to the sequence of latent variables: Given a trajectory τ of state action pairs $(s_i, a_i), i = 0, ..., t$, generated up to current time point t, $\tau^{(1:t)} = (s_1, ..., a_{(t-1)}, s_t)$, the following variational lower bound $L_1(\pi, q_\phi)$ of the directed information $I(c;\tau)$ is derived by using an approximate posterior $q_\phi(c^t \mid c^{(1:t-1)}, \tau^{(1:t-1)})$ instead of the true posterior $p(c^t \mid c^{(1:t-1)}, \tau^{(1:|\tau|)})$:

$$L_1(\pi, q_\phi) = \sum_t \mathbb{E}_{c^{(1:t)} \sim p(c1:t), a^{(t-1)} \sim \pi(\cdot \mid s^{(t-1)}, c^{(1:t-1)})} log\ (q_\phi(c^t \mid c^{(1:t-1)}, \tau^{(1:t)})) + H(c) \leq I(c; \tau) \tag{2.26}$$

Thus, at each time point t, this method learns a posterior distribution over the latent code c at t, given the latent factors discovered up to $t-1$, and the trajectory up to t. This lower bound extends the generator's objective with the maximization of the directed information between the latent variables and the state action pairs.

To learn the approximate posterior $q_\phi(c^t \mid c^{(1:t-1)}, \tau^{(1:t-1)})$ Directed InfoGAIL uses a VAE. In this context the encoder approximates the true posterior $p(c^t|c^{1:t-1}, \tau^{1:|\tau|})$, predicting the latent codes while performing tasks. The decoder learns a policy π, generating actions, given the state and the predicted latent code, according to the demonstrated examples. To do so, VAE minimizes the following objective:

$$L_{VAE} = -\sum_t \mathop{\mathbb{E}}_{c^t \sim q_\phi} [log\ \pi(a^t|s^t, c^{1:t})] + \sum_t D_{KL}(q_\phi(c^t|c^{1:t-1}, \tau^{1:t})||p(c^t|c^{1:t-1})) \tag{2.27}$$

As will be discussed in section 4.6.3, motivated by the Hierarchical RL literature and Directed InfoGAIL this study considers a hierarchical structure of the ATCO's behavior and uses a VAE to model this structure.

2.2 Air Traffic Management

According to International Civil Aviation Organisation (ICAO), ATM comprises the following main services:

- Airspace Management (ASM)
- Air Traffic Flow and Capacity Management (ATFCM)
- Air Traffic Service (ATS)

These services have complementing purposes towards a shared goal: providing safe and efficient aircraft movement during all phases of operations.

The ASM service is responsible for managing the airspace, segregating it into airspace volumes of specific capacity called *sectors*. Sector *capacity* is defined as the number of aircraft that can pass through the sector in a given period of time and is a measure of the workload that the ATCOs assigned to the sector can handle.

The main objective of the ATFCM service is to protect the ATC service from overload by regulating traffic flows to balance the sectors' demand and capacity. This is mainly achieved by making flight plan changes. *Flight plans* are documents submitted to the ATS before departure and provide information about the flights' intended route or flight path.

The main purpose of the ATS is to prevent collisions between aircraft and is performed by the ATC service. To prevent collisions between aircraft, the ATC system requires that certain separation minima are imposed between aircraft. This is achieved by the ATCOs, performing the CD&R task, monitoring traffic and imposing real-time clearances and regulations on aircraft in case the defined separation minima are predicted to be violated.

2.2.1 Conflict Detection and Resolution

As already discussed, to maintain the risk of collision between aircraft in acceptable levels, the ATC system requires that the aircraft do not breach certain separation minima both at the horizontal and vertical axes. According to ICAO Document 4444 the minimum prescribed horizontal separation when using surveillance systems is 5 Nautical Miles (NM). This may be further reduced or increased by the ATS authority based on the surveillance systems' capabilities and the situation created between the aircraft. According to ICAO documents, the specified minimum vertical separation for Instrument Flight Rules (IFR) flights is 1000 ft (300 m) below Flight Level (FL) 290 and 2000 ft (600 m) from FL290 and above. When reduced vertical separation minima apply, this changes to 1000 ft (300 m) below FL410 and 2000 ft (600 m) from FL410 and above.[1] A *loss of separation* is defined as the violation of separation minima in controlled airspaces, whereas a *conflict* is defined as a *predicted* violation of the separation minima.

Nowadays conflicts are detected and resolved by the Planning Controller (PC) and the Executive Controller (EC). The PC and EC have different responsibilities but collaborate closely in order to ensure efficient and safe traffic flow through their sector.

The PC's responsibilities mainly involve coordination with the upstream and downstream sectors. Specifically, for each incoming flight the PC assesses the entry conditions to his/hers sector, proposed by the upstream sector also in relation to the flight plan and identifies potential problems. Such problems include conflicts with other flights or problems regarding the potential exit conditions. If conflicts are detected then the PC in collaboration with the EC will assess, if the conflicts can be resolved by the EC. Finally the PC must agree on the exit conditions with the PC of the downstream sector, making changes to the flight plan if needed. If the PC agrees with the EC and the PC of the upstream and downstream sectors the incoming flight is accepted to pass through the sector. In any other case, amendments must be made by negotiating the entry conditions with the PC of the upstream sector or making flight plan changes to agree on safe exit conditions with the PC of the downstream sector. As a last resort the PC can reject a flight if he/she cannot ensure safe passage through his/her sector.

Tactical conflict detection and resolution is executed by ECs detecting and resolving conflicts in their respective sectors, also coordinating actions with the ECs of the downstream sectors. In contrast to that, which happens today, in a flight-centric ATC one may ignore sectors: Conflicts are detected in a temporal and spatial granularity which is larger than that of flights in sectors and ATCOs are responsible for managing specific aircraft throughout their flight segment within a larger airspace. Flight-centric ATC will allow better traffic distribution avoiding underloaded sectors while also reducing fuel consumption and emissions, enhancing predictability, improving operational and cost efficiency while maintaining safety.

Thus the planning conflict detection and resolution process suggests changes in the flight plan. At the tactical phase it implies changes of the actual flight trajectory, given the trajectory up to the current time point, the current flight plan, and/or prediction(s) on the evolution of the trajectory from that time point and on. Prediction is crucial here, since the future position of the aircraft is uncertain and the uncertainty grows larger with longer time horizons, limiting the confidence on predictions. This, combined to the (uncertain) evolution of other trajectories, implies uncertainties in conflicts that ATCOs have to assess towards prescribing resolution actions.

CD&R involves human expertise and informed judgment. Thus, it is very difficult to hand-craft

[1] https://www.skybrary.aero/index.php/Separation_Standards

criteria which will drive a system to decide whether a conflict deserves a certain reaction at a particular time point, which conflict should be resolved among several co-occurring ones, and at which time point during the evolution of the involved trajectories one should react to a conflict, especially in long-term horizons (i.e. beyond 15-20 minutes).

This thesis presents ML methods for planning conflicts-free trajectories. This is close to the flight-centric ATC concept as it involves predicting trajectories and detecting and resolving conflicts on a per trajectory basis and for large spatio-temporal horizons (for the whole flight). Also the proposed approach considers preferences/best practices of ATCOs and human tolerance as these are revealed by historical data on executing flight trajectories and resolving conflicts.

Part II

Planning Conflicts-Free Trajectories

This part presents the main contributions of this thesis. It includes three chapters corresponding to different methodological steps addressing the objectives of this study. Specifically the overall methodology comprises the following steps:

1. Data-driven prediction of flight trajectories per OD pair.
2. Data-driven modeling of the ATCOs' behavior in resolving conflicts.
3. Conflicts-free trajectory planning.

Chapter 3

Data-driven prediction of flight trajectories per origin-destination pair

This chapter presents the first methodological step towards planning conflicts-free trajectories. This step formulates the trajectory prediction problem, without explicitly considering conflicts, as a data-driven IL problem. Aiming to imitate the experts "shaping/evolving" trajectories, this study devises AI/ML methods that learn policy models incorporating preferences, strategies, practices etc. in an aggregated way, as revealed by historical data. In this context, the trajectory prediction problem has been formulated as an IL problem and the GAIL IL method has been selected to learn the models. To evaluate the effectiveness and efficiency of the approach, experiments on trajectories among different OD pairs report on the following measures regarding the accuracy of the predictions: (a) Root Mean Squared Error (RMSE) in meters in each of the 3 dimensions, as well as in 3D, (b) Along-Track Error (ATE), (c) Cross-Track Error (CTE), and (d) Vertical Error (VE), between predicted and historical trajectories. Results show the effectiveness and efficiency of this approach, and show that GAIL can be effective (in terms of accuracy of predictions) even with a small number of historical trajectories, able to provide accurate long-term predictions, compared to state of the art trajectory prediction approaches.

This chapter is structured as follows: Section 3.1 presents related work regarding trajectory prediction. Section 3.2 specifies the problem of trajectory prediction as an IL problem. Section 3.3 formulates the trajectory prediction problem and presents methods for solving it. Finally, section 3.4 evaluates the methods proposed using real world data and section 3.5 concludes this chapter.

3.1 Related Work

Recent data-driven efforts in the field of aircraft trajectory prediction have explored the application of statistical analysis and machine learning techniques. A comprehensive review of trajectory prediction methods in different domains can be found in [2]. As far as aircraft trajectory prediction is concerned, most approaches make specific assumptions concerning the types of aircraft considered (e.g. [50], the operational phase considered (e.g. climbing, being in terminal airspace, etc.) (e.g. [33], [85]), the short look-ahead time (as in [33] and [15]), or they consider specific constraints that aim to constrain the possible predictions [2].

This study applies IL methods, specifically the GAIL algorithm, to predict trajectories in the aviation domain based on historical datasets. Compared to other approaches GAIL is able to learn trajectory models with no specific requirements on specifying trajectory constraints, and with minimal data pre-processing requirements.

State of the art approaches in the ATM domain that are closely related to this study are those in [4], [51], [2] and [73].

In more detail, authors in [4] introduce a stochastic approach, modeling trajectories in space and time by using a set of spatiotemporal 4D joint data cubes, enriching these with aircraft motion parameters and weather conditions. This approach computes the most likely sequence of states derived by a hidden markov model, which has been trained over trajectories enriched with weather variables. The algorithm computes the maximal probability of the optimal state sequence, which is best aligned with the observation sequence of the aircraft trajectory. When compared to the approach proposed in this thesis, that method uses state action descritization, whereas in this thesis GAIL is applied in a continuous state-action space.

In reference [51], authors propose a tree-based matching algorithm to construct image-like feature maps from high-fidelity meteorological datasets. They then model the trajectory points as conditional gaussian mixtures with parameters to be learned from the proposed deep generative model, which is an end-to-end convolutional recurrent NN that consists of a Long Short-Term Memory (LSTM) encoder network and a mixture density LSTM decoder network. It must also be noted that, that approach requires flight plans, as well as a number of actual trajectory points prior to prediction. The method proposed in this thesis seems to be more effective in terms of predicted trajectory deviations from the actual trajectories in all dimensions: Without requiring any information that will guide/constrain predictions.

The approach in [2] is a "constrained" approach, learning the deviations of trajectories from flight plans and reporting low deviations per waypoint. This is in contrast to the approach presented in this thesis, which does not exploit any information constraining the predicted trajectory.

Applearn [73] is an apprenticeship learning IL approach for the trajectory prediction problem, assuming that the reward function is a linear combination of basis functions on state variables. This method has been proposed as an alternative to the method used in this study, aiming to study the effectiveness of linear reward functions. In contrast to Applearn the algorithm used in this study does not impose such constraints on the reward function.

In [79], authors propose a deep learning model that predicts aircraft trajectories, while modeling and incorporating into the prediction process aircraft tactical intent. The proposed model exploits the encoder-decoder architecture to capture hidden patterns in the trajectories evolution and also uses a convolutional layer and gated recurrent units to capture temporal patterns between trajectory points perceived by the model. In the context of the proposed approach, tactical intent refers to the list of waypoints the aircraft is set to traverse. In contrast to the approach presented in this thesis, which predicts whole trajectories between an OD pair, that work considers only the en route phase of flights and has a prediction horizon of 1 to 10 minutes. Another difference between the work proposed in this study and the work in [79] is that this study exploits weather data and does not require flight plan information, whereas the method presented in [79] exploits flight plans and does not consider weather data.

Finally, authors in [67] explore different versions of hybrid-recurrent neural networks, combining feature extraction layers that exploit convolution or self-attention with recurrent layers that

model the sequential nature of the trajectory prediction problem. The proposed approach exploits flight plan information combined with weather data. Utilizing flight plans is in contrast to the work presented in this thesis which does not exploit any information constraining the predicted trajectory.

Concluding this section, this study explores the application of imitation learning, specifically GAIL, to predict trajectories in the aviation domain based on historical datasets. Compared to other approaches GAIL is able to learn trajectory models with no specific requirements on specifying trajectory constraints (i.e. flight plans [51], [2]), operating on continuous state-action spaces (compared to [4]). When compared to other IL methods (i.e. [73]) GAIL does not impose constraints on the form of the reward function.

3.2 Data-Driven Aircraft Trajectory Prediction

In the aviation domain, a trajectory is defined as the description of movement of an aircraft both in the air and on the ground.[1]. This description can be provided by a chronologically ordered sequence of aircraft states. Most relevant state variables are airspeeds, 3D position (determined by latitude, longitude and geodetic altitude), the bearing or heading and the instantaneous aircraft mass.

Following a data-driven approach, the aim is to exploit historical 4D aircraft trajectories whose states include 3D aircraft position with timestamps, in conjunction to contextual information providing useful features in the prediction process, such as weather conditions at each state, traffic, special events occurring etc. Adding variables in a trajectory state results in a trajectory with *enriched points* or *enriched states*, thus to an *enriched trajectory*:

An *enriched trajectory state* or *enriched trajectory point* of a trajectory of length $|T|$, is defined to be a triplet $s_{r,i}$=<p_i, t_i, v_i>, where p_i is a point in the 3D space, v_i is a vector consisting of categorical and/or numerical variables and t_i is a timestamp, with $i \in [0, |T| - 1]$. An *enriched trajectory* T is defined to be a sequence of enriched states $s_{r,i}$=<p_i, t_i, v_i>, $i \in [0, |T| - 1]$.

A *predicted trajectory* can be defined as the future evolution of the aircraft state as a function of (a) the current flight conditions (e.g. a given state), (b) a forecast of contextual features (e.g. of weather conditions) and (c) a "policy" specifying how the aircraft intends to transit among subsequent states.

Casting the trajectory prediction to a data-driven problem, and assuming a set $\mathbf{T}_E = \{T_{E,i} | i = 1, 2, 3, ...\}$ of historical, demonstrated enriched trajectories, the *trajectory prediction problem* can be defined as follows: Given $\mathbf{T}_E$ and a reward function R, the objective is to predict a trajectory T_π, such that it maximizes:

$$\mathbb{E}_\pi[R(\langle p, t, v \rangle, a)] \tag{3.1}$$

where $\mathbb{E}$ denotes the expected cumulative reward for all states $s = \langle p, t, v \rangle$ along the predicted trajectory by following a policy $\pi(a|s)$, prescribing the probability of applying an action a at state s. Actually, following equation 3.1, the ultimate objective is to find the policy π that determines the generation of a maximal-expected-cumulative-reward predicted trajectory T_π, formally:

[1]https://ext.eurocontrol.int/lexicon/index.php/Trajectory

$$\arg\max_{\pi} \mathbb{E}_{\pi}[R(\langle p, t, v\rangle, a)] \tag{3.2}$$

The reward function may take several forms depending on how the problem is approached: For instance, considering specific trajectories (e.g. flight plans, or cluster medoids) as constraints (e.g. as in [26]), and measuring the adherence of predictions to these constraints, the reward function may take the form of a distance function between these trajectories and predicted trajectories. Generally, in a data-driven trajectory prediction process, the reward function measures the adherence of predictions to given trajectories (e.g. those provided as constraints, or those demonstrated, i.e. those generated by an expert policy). This issue is further discussed later while specifying the trajectory prediction problem as an imitation process, in section 3.2.1.

Furthermore, formula 3.2 includes the trajectory enriched states and actions performed: The formulation indicates separately the 4D position information with timestamps and other variables enriching states. Indeed, additional features may be considered in the reward function, such as weather variables, traffic, airspace crossed, etc. Also, different prediction processes may have different prediction objectives: For instance, one may predict the aircraft position at specific time instances, or predict the time instance that a specific position will be reached, or the position together with the corresponding timestamp, or even predict some of the contextual features, such as airspace crossed at specific time instances. The aim in this work is to predict the 3D aircraft position at specific time instances, given forecasts for contextual features. Actions executed at each state determine how the aircraft intends to evolve its trajectory towards the next state. The actions set A may vary between different approaches.

3.2.1 Problem Specification

Let us assume a set $\mathbf{T}_E$= $\{T_{E,i}, i = 1, ...N\}$ of historical, enriched aircraft trajectories generated by an expert policy π_E. These trajectories have various numbers of states, and therefore, various lengths $|T_{E,i}|$. The objective is to find a policy that minimizes the difference between the expected cumulative reward of the predicted trajectories and of the trajectories in $\mathbf{T}_E$, given an approximation of the reward function that penalizes any state-action pair generated by any policy in $\Pi - \{\pi_E\}$. As shown in [37], this objective is equivalent to finding a policy π that brings the distribution of the state-action pairs generated by it, as close as possible to the distribution of the state-action pairs demonstrated by trajectories in $\mathbf{T}_E$.

In this study the aim is to predict the 3D aircraft position at specific time instants, given an initial state at time instant t_0: Specifically, this study aims at determining the evolution of the trajectory in space every Δt seconds, i.e. at time instances $t_i = t_0 + (\Delta t * i)$, $i = 1, 2, 3...$, given the position of the aircraft at time instance t_0.

Specifically, the *data-driven aircraft trajectory prediction problem is specified as an IL task* as follows: Given a set $\mathbf{T}_E$= $\{T_{E,i}, i = 1, ...N\}$ of historical aircraft trajectories, and a time step Δt, the goal is to determine a policy $\pi \in \Pi$ which, given the initial state of aircraft s_0, maximizes the expected cumulative reward at any time instant $t = t_0 + (\Delta t * i)$, $i = 1, 2, 3...$, according to a reward function that assigns high reward to trajectories in T_E and low reward to trajectories generated by any policy $\Pi - \{\pi_E\}$.

3.3 Predicting aircraft trajectories with IL methods

3.3.1 States and Actions

Following the formulation of the trajectory prediction problem as an IL task presented in section 3.2.1 the following states and actions are considered.

States comprise the following information:

- The aircraft's position in terms of longitude (l), latitude(f) and altitude (h).
- The following weather features: temperature, geopotential height, u-component of wind, v-component of wind.
- The timestamp.

A crucial decision concerns the set A of actions considered, which should adequately and unambiguously specify the evolution of the trajectories. In the approach proposed here, and very close to the General Adversarial Imitation from Observations approach described in [77], the focus is on states, rather than on actions that determine the movement of the aircraft from state to state. This approach is motivated by considering the following: (a) Expert trajectories do not specify in any way the actions applied in any state and thus, these have to be determined a posteriori under specific assumptions that may bring noise into the learning process; (b) there are several possibilities of instruction combinations for evolving the aircraft state, at different levels of detail, which result in a high-dimensional state-action space, and which require considering constraints between combinations of instructions; (c) what we aim to actually predict is the evolution of aircraft states in the 4D space (i.e. regarding position and time); and (d) the IL approach that we take aims to bring the distribution of state-action pairs of the imitator close to the corresponding distribution of the expert.

The set A, considered here, contains all the possible combinations of differences in all 3 spatial dimensions between subsequent trajectory states' position information, given the constraint that each difference must be feasible within the constant Δt period considered, w.r.t. aircraft capabilities (e.g. maximum speed). Specifically, the considered action set, depends on how the position information is represented: Given a position in terms of longitude, latitude and altitude (l, f, h), actions take the form of (Δl, Δf, Δh), and the position in the next state is determined by $(l + \Delta l, f + \Delta f, h + \Delta h)$.

Indeed, these actions can be determined by the demonstrated trajectories unambiguously and very efficiently, although in low-quality surveillance data space-time constraints concerning the evolution of aircraft states may be violated. This action set has three additional important effects: (a) The resolution of the predicted trajectory can be tuned by changing the Δt. (b) Given a specific Δt (e.g. 5 seconds), and the evolution of the trajectory until reaching the destination airport, the Estimated Time of Arrival (ETA) can be determined, which is simply ($\Delta t * |T_\pi|$), given the predicted trajectory T_π. (c) The transition between subsequent positions is deterministic given the first state and an action.

3.3.2 GAIL for predicting flight trajectories

As discussed in section 2.1.4.3, GAIL employs a generative trajectory model G that models π and a discriminative classifier D that distinguishes between state action pairs generated by π and those in the demonstrated data. Both π and D are represented by function approximators with weights θ and w, respectively. Following the implementation described in [37], GAIL alternates

between an Adam [45] gradient step on w to increase the objective function stated in equation (2.23) with respect to D, and a step on θ using the TRPO algorithm [68] to decrease the objective function (2.23). As presented in section 2.1.3.2.1 TRPO optimizes the following objective:

$$\begin{aligned} &\underset{\theta}{minimize}\mathbb{E}_{s\sim\pi_{\theta_{old}},a\sim\pi_{\theta_{old}}}[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}Q^{\pi_{\theta_{old}}}(a|s)] \\ &subject\ to\ \mathbb{E}_{s\sim\pi_{\theta old}}[D_{KL}(\pi_{\theta_{old}}(\cdot|s)\|\pi_\theta(\cdot|s))] \leq \delta \end{aligned} \tag{3.3}$$

where $\pi_{\theta_{old}}$ is the prior-to-update (old) policy $\pi_{\theta_{old}}$, π_θ is the updated policy with parameters θ, $Q_{\theta_{old}}$ is the state-action value function of the old policy and δ is a constant that constraints the KL divergence between $\pi_{\theta_{old}}$ and π_θ, preventing the policy from changing too much due to noise in the policy gradient. In this work the TRPO optimization problem is solved as described in [68] Appendix C, using the conjugate gradient method and a line search. In the setting considered here λ_H is set to 0 (zero), so $-\lambda_H H(\pi)$ is omitted from the equation (2.25), following the practice demonstrated in [37].

The implemented method, instead of approximating Q, utilizes a separate critic model to approximate the state advantage defined as $A^\pi_{DV_t} = A_{DV}(s_t, a_t|\pi) = Q^\pi(s_t, \pi(s_t)) - V^\pi(s_t)$, aiming to lower the gradient variance. Specifically, the GAE approach presented in section 2.1.3.2.2 is followed here, providing a balance between low variance and a small amount of bias introduced. Specifically, the advantage from the sampled state-action pairs is estimated as follows:

$$\hat{A}^\pi_{GAE_t} := (1-\lambda)(\hat{A}^{\pi,1}_{DV_t} + \lambda\hat{A}^{\pi,2}_{DV_t} + \lambda^2\hat{A}^{\pi,3}_{DV_t} + \dots) \tag{3.4}$$

where $\gamma \in [0, 1]$ is the discounting factor, λ a hyper-parameter and

$$\hat{A}^{\pi,k}_{DV_t} := -V^\pi(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V^\pi(s_{t+k}) \tag{3.5}$$

Algorithm 4 shows the aforementioned procedure in more detail. Specifically, G is pre-trained using BC . Then, at each GAIL iteration, the algorithm samples from the initial state distribution and generates roll-out trajectories. It uses the generated state-action samples and the samples of the historical trajectories to update the D parameters w. D is updated with cross entropy loss that pushes the output for the demonstrated state-action samples closer to 0 (zero) and π_θ state-action samples closer to 1 (one). Next, the imitation algorithm takes a policy step using the TRPO [68] update rule and $-logD(s, a)$ as the reward function approximation to update θ. It must be noted that the t parameter in the denotation of the approximation of the state advantage in Algorithm 4 is left implicit, for simplicity of the presentation.

Algorithm 4: GAIL for predicting flight trajectories

Input: Expert trajectories $\tau_E \sim \pi_E$, initial policy π_{θ_0} and discriminator parameters w_0;
Output: Policy π_θ Initialize policy using BC . **for** *i=0,1,2,...* **do**
  Sample trajectories $\tau_i \sim \pi_{\theta_i}$;
  Update D parameters w with the gradient;
  $\hat{\mathbb{E}}_{\tau_i}[\nabla_w log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w log(1 - D_w(s,a))]$;
  Estimate advantages $\hat{A}^{\pi_{\theta_{old}}}_{GAE}$, according to $\pi_{\theta_{old}}$;
  Take a policy step using the TRPO rule with reward function $-log(D_w(s,a))$:;
  Take a KL-constrained natural gradient step with;
  $\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}\hat{A}^{\pi_{\theta_{old}}}_{GAE}]$ subject to;
  $\mathbb{E}_{s\sim\rho_{\theta old}}[D_{KL}(\pi_{\theta_{old}}(\cdot|s)\|\pi_\theta(\cdot|s))] \leq \delta$

The GAIL generative model G and the discriminator D are implemented using two NNs consisting of two dense layers of 100 nodes, each layer with $tanh$ activation. The input for G corresponds to the position and temporal variables per state, and the other variables enriching a trajectory state. D takes as additional input the three action variables. Thus, the input of G and D depends on the way positions and actions are formulated. G has a dense output layer with size equal to the number of action variables, while the output layer of D has one node. G outputs for each action variable the mean of a Gaussian distribution with the logarithm of standard deviation equal to 0.6, resulting to a stochastic policy. BC minimizes the Mean Square Error (MSE) between demonstrated actions and the policy actions, over the training set, using Adam optimization. This has been trained with 100 epochs and 10 fold cross validation.

3.4 Experimental Evaluation

3.4.1 Experimental Setting

Datasets exploited in the presented experiments include (a) surveillance data consisting of radar tracks for flights between 3 OD pairs: Barcelona to Madrid (BCN-MAD) during July 2019 (308 trajectories), London Heathrow to Rome Fiumicino (LHR-FCO) during July 2019 (219 trajectories), and Helsinki to Lisbon (HEL-LIS) during July 2019 (44 trajectories); (b) weather data obtained from National Oceanic and Atmospheric Administration (NOAA); and (c) aircraft models' ids. Datasets used in this study are described in more detail in section A.2 of the appendix. Prediction of long flights regarding LHR-FCO and HEL-LIS is the more challenging problem, as it involves large time horizon prediction, with many uncertainties during trajectory evolution.

Trajectories in these datasets have been pre-prosessed, cleaned and enriched with five (5) numerical variables corresponding to 4 meteorological features at any trajectory state position and time, provided by NOAA, and the aircraft model of each trajectory. The NOAA features are *temperature, geopotential height, u-component of wind, v-component of wind.*

The pre-processing stage interpolates points in trajectories, so that two subsequent points have a temporal distance of $\Delta t = 5$ seconds in the case of the short BCN-MAD trajectories and $\Delta t = 10$ seconds in the case of the long LHR-FCO and HEL-LIS trajectories. This task assumes constant velocity between subsequent trajectory points and calculates the position of the aircraft every Δt seconds. It finally keeps only the points occurring every Δt seconds along the original trajectory. The cleaning task aims to detect incomplete trajectories starting or finishing away from any of the airports, as well as flights showing inconsistent behavior (e.g. covering a significant distance within an unreasonably small amount of time), due to imperfections in the raw data.

The prediction accuracy achieved is measured at the pre-tactical phase (starting from a position in the origin airport) and at the tactical phase (starting from any point en-route), introducing a parameter M in {0,0.2, 0.5, 0.7}. M determines the initial state of the prediction, i.e. the state in the actual trajectory after $(M \times FlightDuration)$ minutes, starting from t_0.

Trajectory prediction accuracy is reported using the following measures: **(a)**RMSE in meters in each of the 3 dimensions, as well as in 3D, **(b)** ATE, **(c)** CTE, and **(d)** VE. ATE and CTE are computed according to the methodology proposed in [29]. As shown in Figure 3.1, the along track error is measured parallel to the predicted trajectory, while the cross track error is measured perpendicular to the predicted course. VE measures the difference in altitude between the predicted and the corresponding test (actual) trajectory.

To compute the ATE, CTE and VE, the predicted trajectory is aligned with the corresponding test

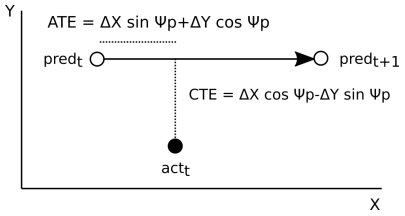


Figure 3.1: Along-Track Error (ATE) and Cross-Track Error (CTE) errors w.r.t. the predicted trajectory's points at times t and $t+1$, denoted by $pred_t$ and $pred_{t+1}$, and the actual trajectory's point act_t at time t. $\Delta X = X_p - X_a$ is the difference in the X dimension (longitude) of $pred_t$ (X_p) and act_t (X_a). $\Delta Y = Y_p - Y_a$ is the difference in the Y dimension (latitude) of $pred_t$ (Y_p) and act_t (Y_a). Ψp denotes the bearing of the predicted trajectory (i.e. the angle between the direction of the trajectory and the North).

(actual) trajectory in the time dimension, so as to calculate the errors between trajectory points with the same timestamp. As the predicted trajectory may have different length (different number of points) compared to the test trajectory, the points of the longer trajectory (predicted or actual), are compared to the last point of the shorter trajectory, as this is the last known position of the aircraft. Finally, provided results include **(c)** the ETA error, given the predicted ETA and the arrival time of test trajectories. All errors ATE, CTE, VE and ETA are signed errors, but their absolute values are used in order to report on average scores from multiple experiments, providing a clear indication of the errors.

The RMSE error is computed for each predicted trajectory point after computing its corresponding point in the test trajectory using the Dynamic Time Warping [7] [58] method. RMSE errors are computed in each of the 3 dimensions using the formula

$$RMSE(var) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(var_{pred} - var_{actual}\right)^2}$$

and the 3D RMSE error using the formula

$$RMSE_{3D} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{\sqrt{\sum_{d=1}^{Dim}(var_{d_{pred}} - var_{d_{actual}})^2}}{Dim}}.$$

where, N is the number of trajectory point pairs compared, Dim is the number of dimension variables considered per point, var is the variable corresponding to a certain dimension and $pred$ and $actual$ indicate the predicted and actual trajectories, respectively.

Specifically, the average of RMSE is reported for the longitude, latitude and all three dimensions (3D), as well as the average of ATE, CTE, VE and ETA are reported for 20 independent experiments per experimental case. The division of the historical trajectories for training and testing purposes is done randomly for each of the individual experiments using 90% of them as expert trajectories and 10% as test trajectories .

GAIL is trained for 1500 batches. At each round the policy generates a batch of 50000 state-action samples. The number of episodes needed for this number of samples is not constant. At each episode the method randomly selects a starting point regarding a trajectory in the training set and uses G to generate roll-outs. Roll-outs terminate either when a trajectory point lies within a 8km radius from the destination airport, for the cases of BCN-MAD and LHR-FCO and 14km for HEL-LIS or when the trajectory has 1000 points for the cases of BCN-MAD and LHR-FCO and 1900 for HEL-LIS, or when it evolves in positions out of the *prediction area* (defined below). These 50000 samples, along with all the expert samples, are used for training the Dis-

criminator D. Specifically, we use Adam optimization and 100 epochs to maximize equation (2.23) w.r.t. the D parameters w.

The *prediction area* is a 3D area in which generator's roll-outs are allowed to evolve. While the prediction area for short trajectories can be the 2D trajectory bounding box (along with a maximum allowed altitude of 40000 feet), long trajectories are included in significantly larger bounding boxes (sometimes including the whole continent). To address this issue, the prediction area is the area determined by the 1×1 degree cells that the demonstrated trajectories cross, expanded by additional cells of 1×1 degree in every direction of the initial area to provide a kind of "buffer" that gives room for the GAIL generator to explore. For example, red dots in Figure 3.2 indicate the corners of the bounding box of the LHR-FCO trajectories. The green color specifies the cells crossed by demonstrated trajectories and the blue color the buffer area.

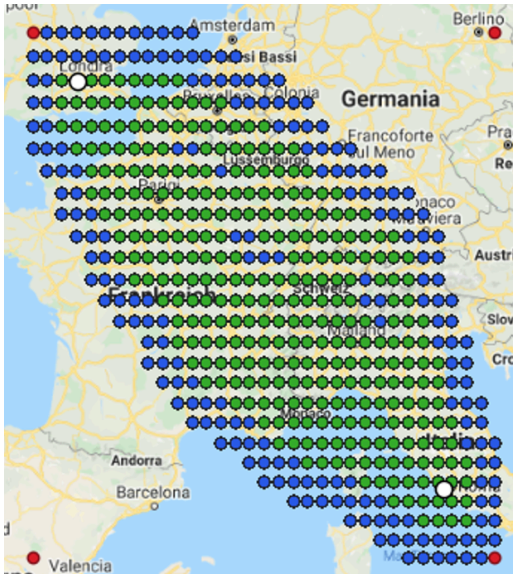


Figure 3.2: Specification of the bounding box and of the prediction area for LHR-FCO trajectories.

3.4.2 Experimental Results

Table 3.1 shows the average RMSE error of the predicted vs the actual (test) trajectory in meters for each of the three dimensions and in 3D, together with the average absolute ATE, CTE, and VE, in meters. It also reports the average error of the ETA in seconds for each case. The table is split to parts corresponding to the different OD pairs examined, starting from the short trajectories and going into the longer ones with fewer samples, and for each pair the results provided by the GAIL method are reported, for different values of M.

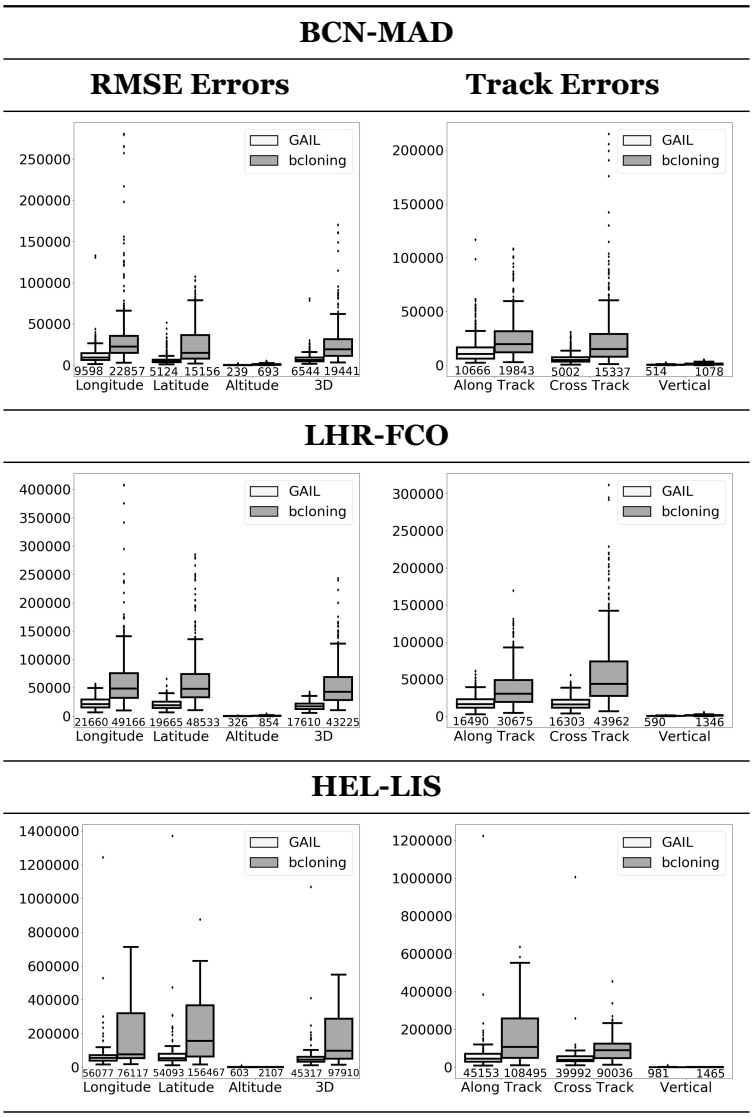
Figures in Table 3.2 show box plots for all the measures. The x axis specifies the error measured. Horizontal lines of each box plot represent the 25^{th}, the 50^{th}, the 75^{th} and the 100^{th} percentile. Diamonds indicate outliers and the numbers indicate the medians. The left column provides RMSE and the right the along and cross track errors. These box plots correspond to the cases where $M = 0$.

Not surprisingly, the GAIL method provides consistently better results compared to the BC baseline: Indeed, table 3.2 box-plots show that GAIL reports smaller errors with narrower deviations, and very small number of outliers compared to BC. In addition to that, low deviations of predicted from the actual trajectories, compared to state of the art methods provide firm

Table 3.1: Prediction Errors (in meters) and ETA (in seconds)

					BCN-MAD				
	M	**Long**	**Lat**	**Alt**	**3D**	**ATE**	**CTE**	**VE**	**ETA**
GAIL	0	11994.99	6214.72	282.63	8005.49	13923.3	6392.61	574.13	263.06
	0.2	10021.46	5549.65	317.87	6774.39	11541.14	6159.43	516.15	259.81
	0.5	8680.78	5103.39	391.5	5977.16	11360.9	6330.23	510.78	239.35
	0.7	6327.37	4078.96	273.41	4519.16	9751.45	5284.18	375.73	158.9
					LHR-FCO				
	M	**Long**	**Lat**	**Alt**	**3D**	**ATE**	**CTE**	**VE**	**ETA**
GAIL	0	23371.12	20888.65	372.33	18351.99	18689.42	17874.3	636.2	457.1
	0.2	24427.65	20568.25	359.35	18733.01	19273.79	19362.4	621.78	615.12
	0.5	20274.53	18497.25	370.98	16209.15	15758.31	16399.46	629.06	791.83
	0.7	14313.57	14444.13	539.25	12126.93	13205.18	11294.95	659.35	910.28
					HEL-LIS				
	M	**Long**	**Lat**	**Alt**	**3D**	**ATE**	**CTE**	**VE**	**ETA**
GAIL	0	88448.14	95173.02	1096.41	75950.75	77341.27	59731.61	1074.71	801.44
	0.2	91184.7	100957.56	1062.17	79921.51	81309.09	52941.16	1052.04	978.19
	0.5	90334.3	92006.24	1090.32	76575.08	81468.87	49669.47	1252.01	1080.75
	0.7	77587.38	76998.23	1966.88	64771.15	81990.64	46206.48	1691.44	1113.12

Table 3.2: Prediction Errors box plots: Numbers below the boxes indicate the medians.



evidence of the IL approach efficacy, even in very long trajectories spanning the European continent and with few training examples. It must be noted that the lack of benchmarks hinders the systematic comparison of different trajectory prediction methods. However, here we attempt a comparison of methods and findings, aiming to show the importance of our contributions.

In more detail, authors in [4] report that the mean value for the cross-track error is 12.601km when the sign is omitted or -3.444km when signed. Given that this method does not exploit other information regarding operational aspects, but flown trajectories enriched with weather variables in an OD pair (Atlanta, Miami) with distance similar to BCN-MAD, we can conjecture that our proposed method provides a much lower error in RMSE, as well as in along and cross track errors. This is achieved without limiting the resolution of trajectories' representation, while learning/predicting in continuous action-state space.

Compared to reference [51], our proposed method seems to be more effective in terms of predicted trajectory deviations from the actual trajectories in all dimensions: Without requiring any information that will guide/constrain predictions, we report on vertical errors larger than 2800 ft but with a much lower 3D RMSE (increased by 1.21) for an increase of the trajectory length by 0.76 regarding the HEL-LIS case.

Regarding the approach in [2], the 3D RMSE reported for that approach is 1.78 greater than the 3D RMSE of GAIL for the BCN-MAD pair, given the same set of demonstrated trajectories.

Finally, regarding Applearn [73], although it is quite competent to other state of the art approaches, it fails to achieve the accuracy of GAIL: The 3D RMSE reported is 2 times greater than the 3D RMSE of GAIL for the BCN-MAD pair, and approximately 7.2 and 21.3 times, for the LHR-FCO and HEL-LIS pairs, respectively, given the same data sets of demonstrated trajectories. Given these results, the linearity assumption on the cost function seems to reduce prediction accuracy, although Applearn scores better predictions when starting from a state close to the destination airport (i.e. when $M > 0.5$), which is something to be explored in the future.

Table 3.1 shows that the proposed method is quite effective to predict the whole trajectory at the pre-tactical stage ($M = 0$), while all measures are reduced in all cases, except from some of the cases while increasing M, i.e. while we select a starting point far from the origin airport, simulating the tactical stage: This happens for instance in the prediction of very long trajectories regarding HEL-LIS. The average along and cross track errors may increase while increasing M in these cases, due to the complexities of the trajectories while approaching the destination airport (i.e. due to holding patterns, maneuvers, etc.). Thus, it seems that a more refined approach must be used to address the landing part of the trajectory more accurately. This is also the case for the ETA error: If the holding patterns are eliminated while measuring errors for the LHR-FCO pair, unsigned ETA errors are of 67.82, 61.47, 45.94, 35.22 (signed -14.77, -24.22, -17.51, -8.74) seconds, for $M = 0, 0.2, 0.5, 0.7$, respectively. Similar patterns are recorded for the other error measures, providing evidence to the conjecture about the destination airports with complex holding patterns and multiple modes of approach.

Concluding this section, figure 3.3 shows an example of a predicted (black) vs the corresponding historical (red) trajectory between HEL-LIS.

Figure 3.3: Example of a predicted (black) vs the corresponding historical (red) trajectory between HEL-LIS

3.5 Conclusions

In this work the data-driven trajectory prediction problem is specified as an IL task. Towards solving this problem a prediction method using the GAIL state of the art method was presented utilizing a critic model for estimating the state advantage.

Evaluation results show the effectiveness of the method to make accurate predictions for the whole trajectory (i.e. with a prediction horizon until reaching the destination airport) both at the pre-tactical (i.e. starting at the departure airport at a specific time instant) and at the tactical (i.e. from any state while flying) stages, compared to state of the art approaches. Findings are discussed with respect to results reported by state of the art trajectory prediction methods, although a direct and systematic comparison required methods to be trained using the same sets of demonstrated, flown trajectories.

Future plans include (a) verifying the effectiveness of the method for different OD airports, (b) exploiting flight plans to constrain the prediction pipeline, (c) extending the method to deal

inherently with different modes of trajectory evolution, and (d) generalizing beyond specific OD pairs.

Chapter 4

Data-driven modeling of the ATCOs' behavior in resolving conflicts

This chapter presents the second methodological step towards planning conflicts-free trajectories. This step models the ATCOs' behavior in resolving conflicts using data-driven AI/ML techniques. In general, according to the problem specifications made in this study, this implies learning "when" the ATCO will react to resolve a detected conflict, and "how" he/she will react.

More specifically, towards this goal, this study proposes a two-stage data-driven methodology towards meeting the following two objectives:

1. Formulate the ATCO reaction prediction problem, towards building a model of ATCO reactions for resolving conflicts. The aim is to answer "whether" and "when" the ATCO decides to apply an action to resolve a conflict. Towards predicting the ATCO timely reactions to resolve conflicts, this study trains a VAE imitating the demonstrated ATCO behavior in a supervised way. The proposed method has been evaluated in two different operational settings (sector-related and sector-ignorant), reporting on the precision, recall and f1-score of predictions. A weighted version of these measures is introduced, to deal with the inherent uncertainties regarding (a) the evolution of trajectories, (b) the detection of conflicts (which are not specified in the dataset), and (c) the ATCO reaction.

2. Formulate the ATCOs' policy modeling problem, towards building a model of ATCO behavior for resolving conflicts. The aim is to answer "how" the ATCO reacts (i.e. what resolution actions he/she applies) in the presence of conflicts. Towards predicting the ATCO policy, thus, predicting the resolution action the ATCO prescribes in case that he/she reacts in a potential detected conflict, this study evaluates comparatively classification methods using NNs, RF, GTB, while also exploring techniques that are robust to label noise such as SEAL and active-passive loss functions. Also an IL method based on the GAIL framework is reported in the appendix. To evaluate the different methods, this study reports the precision, recall, f1-score and the Matthews Correlation Coefficient (MCC) between the predictions and the resolution actions of the dataset.

This chapter is structured as follows: Section 4.1 presents related work. Section 4.2 specifies the problems of predicting the ATCOs' rections and modeling the ATCOs' policy. Sections 4.3, 4.4, 4.5 present the methodology stages, the data sources used and the trajectory states respec-

tively. Section 4.6 presents the methods used for solving the ATCOs' reactions prediction problem. Section 4.7 presents the methods used for modeling the ATCOs' policy. Finally section 4.8 evaluates method for predicting the ATCOs' reactions and modeling the ATCOs' policy and section 4.9 concludes this chapter.

4.1 Related Work

In recent years multiple works consider the problem of assisting the ATCO with the CD&R task.

[63] provides a survey of CD&R research both on manned and unmanned aviation. [38] proposes a methodology to address strategic planning involving continent scale traffic. This method finds an optimal de-conflicted route and a departure time for each flight, relying on a hybrid-metaheuristic optimization algorithm that combines the advantages of simulated annealing and of hill-climbing local search methods. Compared to this study, that method operates at the strategic phase of operations providing de-conflicted 4D trajectories before the aircraft departs. Also trajectory de-confliction is achieved using optimization algorithms without considering the preferences, constraints and tolerance of the ATCOs.

In [19] the authors propose a light propagation algorithm inspired by nature in order to avoid congestion areas at the pre-tactical phase and generate conflicts-free 4-D trajectories at the tactical phase. A drawback of this method is that it explores the search space in real time and does not generalize beyond the scenario it solves. This can result in delayed decisions due to large computation times i.e. 17 hours for one day of traffic over the French airspace, making the method not a viable option for the tactical phase. Also contrary to this study where actual surveillance trajectories are used, in that work trajectories used are sampled from flight plans. However, in reality there are deviations from the flight plans due to different reasons i.e. delays. Finally, as with the previous methods human preferences tolerance and constraints are not considered.

In [21] the authors propose a genetic algorithm based approach to the en-route conflict resolution problem at the tactical phase of operations. As with the previous approach trajectories are sampled from flight plans, the method does not generalize beyond the specific scenario it is trained on and also ATCO behavior is not considered.

In [74] authors use a lattice-based search space exploration AI planner to perform conflict resolution. The lattice-based search space exploration method proposed in this work provides explanations for why (or why not) a resolution actions was (or not chosen) chosen. Additionally actions are prioritized based on predefined rules aiming to be in par with the ATCOs' logic. Although this is close to our interest there are the following major differences between this work and ours. Firstly, authors in [74] focus on explainability, whereas the goal in this study is to model the ATCO behavior. Secondly, they try to mimic the ATCO logic by applying a predefined prioritization on actions which is independent of the specific situation, i.e., state of flights. On the contrary the aim in this thesis is to model the ATCO preferences and constraints as these are revealed in situations arising in the historical datasets.

In [5] authors exploit a hidden markov model to predict at the pre-tactical phase the evolution of trajectories based on historical trajectories and weather observations. The proposed method uses these predictions to detect conflicts and assign conflict-related probabilities to states. Resolution actions are decided by a variant of the Viterbi algorithm. In contrast to this study, work in [5] operates at the pre-tactical level of operations and does not consider the preferences of the ATCOs.

The application of RL methods on the CD&R task has also received a lot of attention. Authors in [57] propose ResoLver, a system based on an enhanced graph convolutional RL method. ResoLver operates in a multi-agent setting where each agent represents a flight performing the CD&R task jointly with other agents. The proposed approach aims to provide high-quality solutions w.r.t. stakeholders interests while addressing operational transparency issues.

Authors in [60] and [59] explore a DRL approach, based on deep deterministic policy gradient to resolve conflicts between two aircraft in the presence of uncertainty. In [17] authors formulate the problem as a multi-agent RL problem and propose a message passing actor critic model inspired by graph convolutional RL [41], while also exploiting message passing NNs [28].

Authors in [39] use multi-agent deep deterministic policy gradient to resolve conflicts, also considering time, fuel consumption and airspace complexity. Works in [60], [59], [17] use synthetic datasets, while in [39] the authors use flight plan data. This thesis exploits surveillance data, since during the tactical phase flights deviate from their flight plans for different reasons. More importantly, in contrast to this study none of the previous works incorporates into the decision making process human expertise and tolerance.

Closer to the proposed approach are methods that somehow consider the ATCO preferences, either in a data-driven way as in [12], [78] and [64], or by using rules and procedures derived from human experts as in [22].

The work reported in [12] proposes a conflict resolution method operating at the strategic phase of operations. This method projects the aircraft's position into the future using the latest updated flight plans. The proposed methodology utilizes a data-driven model that a) classifies the conflict resolution maneuvers according to the relationship between the aircraft involved in the conflict and b) clusters the conflict resolution actions, considering the centroid of each cluster as a possible solution. Next, the method utilizes an ϵ-constrained multi-objective optimization method to find the Pareto-optimal solutions w.r.t. the minimization of fuel consumption and the maximization of the likelihood of the resolution being implemented by an ATCO. Opposed to this study, work presented in [12] considers the strategic phase of operations. Additionally this study is based on recorded controllers' actions, whereas in that work authors compute conflicts using flight plans and assume that deviations between the planned trajectories and the actually flown ones that resolve such conflicts correspond to ATCO actions.

In [78] the authors propose a conflict resolution advisory system, able to incorporate human preferences. The system uses an interactive conflict solver for acquiring and characterizing human resolutions in conjunction to a RL agent that learns to resolve conflicts incorporating the characteristics of human resolution acquired by the interactive conflict solver. That work focuses on heading changes, deciding the trajectory change point. The trajectory change point is the point at which an aircraft after changing its heading to resolve a conflict will turn again towards its initial track. In contrast to this study where real wold data are used, in this work data of ATCO resolution actions are gathered by recording the trajectory change point decided by the ATCO on synthetic scenarios. Also, the aim of that work is to predict the trajectory change point, not considering the point at which the ATCO reacted, nor the type of the resolution action, as only heading change was considered.

The method proposed in [64] aims to provide personalized advisories to controllers. Authors train a convolutional NN on individual controller's data recorded from a human-in-the-loop simulation to predict conflict resolution actions. The exploited dataset is in the form of solution space diagrams, integrating various critical parameters of the CD&R problem. As in [78], in this work [64] synthetic datasets were used. Data resolution actions are gathered by recording

the resolution actions decided by university staff and students, with varying experience levels in performing ATC control tasks, on synthetic scenarios presented to them.

Finally, [22] describes an algorithm that provides 4D conflict resolution trajectories, based on a set of rules and procedures derived from human experts and from operational insights and analytical studies that reveal the characteristics of efficient conflict resolution techniques. In that work ATCO preferences are not learned from historical real world data but are prioritized based on fixed rules. Also the problem of predicting the ATCO reactions is not considered.

This thesis proposes supervised deep learning techniques to learn models of ATCO behavior in resolving conflicts. Modeling the ATCO behavior implies learning when the ATCO will react towards resolving a conflict, and which resolution action the ATCO will decide. Regarding $when$ the ATCO will react towards resolving a detected conflict, this study is the first to formulate and address this problem to our knowledge. Regarding how the ATCO will react, this thesis advances the state of the art in CD&R automation by a) exploiting recorded ATCO resolution actions on historical surveillance trajectories and b) formulating and addressing the ATCO reaction problem, considering both abstract and low-level ATCO reactions, to imitate the ATCO.

4.2 Problem Specification

This section starts with some definitions regarding domain terminology, and then proceeds to specify the ATCO reaction prediction and the ATCO policy modeling problems.

4.2.1 Definitions

Definitions regarding aircraft trajectories, enriched aircraft trajectories and predicted trajectories are introduced in section 3.2, but are repeated here for reasons of chapter conciseness.

An aircraft trajectory is a chronologically ordered sequence of aircraft states, without an explicit consideration on actions shaping the trajectory: $T = (s_0, s_1, ...s_{|T|-1})$.

Aircraft trajectory states include 3D aircraft position with timestamps, in conjunction to contextual features. Adding contextual features in a trajectory state results in a trajectory with *enriched points* or *enriched states*, thus to an *enriched trajectory*.

An *enriched trajectory state* or *enriched trajectory point* of a trajectory of length $|T|$, is defined to be a triplet $s_{r,i} = \langle p_i, t_i, v_i \rangle$, where p_i is a point in the 3D space, v_i is a vector consisting of categorical and/or numerical variables, and t_i is a timestamp, with $i \in [0, |T| - 1]$. An *enriched trajectory* T is defined to be a sequence of enriched states $s_{r,i}$=$\langle\ p_i, t_i, v_i \rangle$, $i \in [0, |T| - 1]$.

A predicted trajectory T_p, is defined to be a specification of the future evolution of the aircraft state as a function of (a) the current flight conditions (e.g. an initial aircraft state, contextual features that affect a flight, actual weather conditions etc.), (b) a forecast of contextual features (e.g. forecast of weather conditions at specific points/regions, or predicted states of other aircraft) and (c) a "policy" on how the trajectory evolves, i.e. a specification of how the aircraft is to transit among subsequent enriched states starting from the initial state.

Given the evolution of a trajectory T^t, up to t, a predicted trajectory from that time point will be denoted as T_p^t. A set of such predictions, comprising potential trajectory evolutions of T^t, is denoted by $\mathbf{T}_p^t$.

The Closest Point of Approach (CPA)[1] of an aircraft i w.r.t. another aircraft j, is the position of i at which the distance between the two aircraft is assessed to be minimum, w.r.t. their estimated trajectory evolutions. The CPA can be computed in the horizontal plane, in the vertical plane or in all 3 dimensions. The time at the CPA is the time at which the smallest distance between the two aircraft is estimated to occur.

The Crossing Point (CP) [2] of a pair of aircraft $\langle i, j \rangle$ is the point at which the tracks of the aircraft intersect. The track[3] of an aircraft is defined as the projection of the aircraft trajectory on the earth's surface.

Given a spatiotemporal area SA, *neighboring trajectories in SA* are those trajectories that co-occur in SA, i.e., trajectories having 3D points in SA and equal timestamps, satisfying also a set of constraints regarding their tracks, CPA and CP. More formally:

$Neigh(SA, t) = \{(T_i, T_j) |$ *There is at least one point* (s_i, t) *in* T_i *and onepoint* (s_j, t) *in* T_j, *s.t. it holds that* $in(s_i, SA)$ *and* $in(s_j, SA)$ *at time point* t *and also the aircraft flying* T_i^t *and* T_j^t *satisfy a set of constraints CR.*

T_i^t and T_j^t denote the trajectories of aircraft i and j up to time point t.

The predicate $in(s, SA)$ is true when the 3D spatial point corresponding to s is in the spatial region SA.

The set CR includes the following constraints :

a. Considering the actual aircraft states:

- Aircraft altitude difference at the current time point is less than $2 * d_{v_{th}}$ feet.

b. Considering the predicted evolution T_{pi}^t, T_{pi}^t of trajectories:

- Aircraft have not crossed the crossing point;

- The tracks of the aircraft cross in less than ct_{th} minutes;

- The horizontal distance at the CPA is less than $cpa_{d_{h_{th}}}$ NM;

- The time to the CPA is less than $cpa_{t_{th}}$ minutes.

- Aircraft altitude difference at the CPA is less than $d_{v_{th}}$ feet.

Aircraft i, j flying trajectories T_i^t and T_j^t in $Neigh(SA, t)$ are considered to be in conflict.

The parameter ct_{th} is the crossing time threshold, $cpa_{d_{h_{th}}}$ is the horizontal distance threshold at the CPA, $cpa_{t_{th}}$ is the time to CPA threshold, and $d_{v_{th}}$ is the vertical distance threshold. Parameters ct_{th} and $cpa_{t_{th}}$ set the time horizon in which conflicts are detected. Specifically, ct_{th} is set to 20 minutes, as this is the time horizon used by the planning controllers to detect conflicts. As aircraft can reach the CPA after having reached the crossing point, the value of $cpa_{t_{th}}$ is set to 30 minutes. Finally, $d_{v_{th}}$ is set to the vertical separation minimum (1000 ft under FL410 and 2000 ft from FL410 and over) and $cpa_{d_{h_{th}}}$ is set to 15 NM.

[1] https://www.skybrary.aero/index.php/Closest_Point_of_Approach_(CPA)
[2] https://www.skybrary.aero/index.php/Vectoring_Geometry
[3] https://www.skybrary.aero/index.php/Heading,_Track_and_Radial

It must be noted that a large horizontal distance threshold $cpa_{d_{h_{th}}}$ of 15NM at the CPA is used, in order to include margins of error when estimating ATCOs' observations in triggering their reactions. In so doing, further uncertainties in detecting conflicts are incorporated in the process. Indeed, ATCOs take margins of error perceiving flights,[4] even when the predicted horizontal distance between them at the CPA is greater than the horizontal separation minimum. Such margins of error are important to ensure safety.

As already pointed out in section 1.2.2, historical data should ideally indicate the observations perceived by ATCOs before applying a conflict resolution action. Such observations should provide the features that drove the application of a specific action instead of others in ATCOs' repertoire of actions, and concern situations involving specific aircraft trajectories, the prediction of the evolution of the trajectories before the "intervention" of ATCOs, and the assessment of conflicts. However, the historical datasets that this work exploits, provided by the Spanish Automated NORVASE Takes (ATON) platform, indicate only the type of the resolution actions instructed by ATCOs (e.g., change speed), and not the actions in full detail (e.g., how speed has been changed and for how long), and the effects of ATCOs' resolution actions (i.e., the loss-free trajectory), but not the rationale behind them. This lack of information presents challenges to the training of AI/ML systems, since it necessitates recovering the important observations that the ATCOs perceived or assessed, driving their decision. This entails exploiting expert knowledge to assess traffic as the ATCOs would, reveal the potential conflicts the ATCOs might have observed, and associate these potential conflicts with the prescribed resolution actions. Revealing such conflicts from historical datasets is not a trivial task in the ATC domain, as (a) the evolution of the trajectories is uncertain and ATCOs' assessments and practices may vary due to various reasons; and (b) to ensure safety, even when the predicted horizontal distance between flights at the CPA exceeds the horizontal separation minimum, ATCOs allow margins of error when perceiving flights. These margins may not be so large as assumed here, but it must be emphasized that this study does so in order to reveal what the ATCOs perceive prior to reaction (i.e., features associating potential conflicting situations with resolution actions). In addition to the above challenges, associating potential conflicting situations with ATCOs' resolution actions may introduce noise in the training of AI/ML methods, given the uncertainty in determining which features of assessed conflicts are those that provided the rationale for the ATCOs' actions. This issue is further discussed subsequently as the "labels noise" problem.

Table 4.1: Problem-specific parameters.

Parameter	Description	Value
ct_{th}	The crossing time threshold.	20 min
$cpa_{t_{th}}$	Time to closest point of approach (CPA) threshold	20 min
$cpa_{d_{h_{th}}}$	The horizontal distance threshold between aircraft at the CPA.	15 NM
$d_{v_{th}}$	The vertical distance threshold.	1000 ft below flight level (FL) 420, 2000 ft else

Now, given a specific (focal/own) trajectory T_f and a spatiotemporal area SA, the set of *neighboring* and thus *conflicting trajectories to T_f in SA at a specific time point t* (or the set of trajectories interacting with T_f in SA at time point t), denoted $Neigh(T_f, SA, t)$, are defined

[4]These margins may not be so large as it is assumed here, but it must be emphasized that this is done in order to reveal the situation that ATCOs perceive prior to their reaction, not recorded in historical data sources.

to be those that a) have at least one point spatially close to the focal trajectory point at the time instance t, according to a horizontal distance measure $horizontal\_distance$ and a distance threshold D_{th}, and b) satisfy the constraints CR. Formally:

$Neigh(T_f, SA, t) = \{T|$*There is a point* (s_i, t) *in* T *and a point* (s_j, t) *in* T_f, *s.t. it holds that* $in(s_i, SA)$ *and* $in(s_j, SA)$ *at time point* t, *their horizontal distance is within specific limits, i.e.,* $horizontal\_distance(s_i, s_j) \leq D_{th}$ *and the aircraft flying* T_f^t *and* T^t *satisfy the set of constraints CR*$\}$.

It must be noted that although the detection of conflicts is carried out using the aircraft CPA, the additional rules for the identification of neighbor trajectories allows (a) detecting conflicts that correspond to the recorded ATCO resolution actions, which are not explicitly provided in historical datasets, and (b) filtering out aircraft that might be in conflict, but are not considered by the ATCOs at a specific time point (e.g., because the time to CPA is large). Rules can be refined when datasets include further information on conflicts.

4.2.2 ATCOs' Reaction Prediction Problem Specification

Given a set $\mathbf{RA}_E$ of historical ATCOs' conflict resolution actions associated to historical trajectories in $\mathbf{T}_E$, the goal is to learn a model that predicts *when* and *how* the ATCO will react in assigning a conflict resolution action, when conflicts are detected.

So, in order to be able to imitate the behavior of the ATCOs, and consequently the evolution of the trajectories due to conflicts, given $\mathbf{T}_E$ and $\mathbf{RA}_E$, the objective is to learn models that solve the following problems:

1. Predict at any trajectory point the ATCO's mode of behavior, deciding whether the ATCO would issue a resolution action, and
2. Model the ATCO policy, predicting the resolution action the ATCO would decide, if any.

Modes of the ATCOs' behavior, in the more abstract form, include: "Not Assigning resolution action" and "Assigning resolution action". Thus, modes represent *when*, i.e., at which points of the trajectory, the ATCO issues a resolution action.

Given the above, the *ATCOs' Reaction Prediction Problem* is about predicting at a time point t $whether$, $when$ and how the ATCO will react regarding a particular flight that executes a focal trajectory T_f^t, given $Neigh(T_f, SA, t)$ in a spatial area of responsibility SA.

This problem comprises (a) detecting conflicts by identifying neighboring trajectories in the spatio-temporal region SA, (b) determining the exact time point t_A, s.t. $t \leq t_A \leq t_c$ for issuing a resolution action, if any; where t_c is the time point at which the conflict is detected, and (c) deciding the resolution action to be applied.

It must be noted that, given multiple aircraft executing neighboring trajectories, this study does not consider the problem of deciding which of the involved aircraft must maneuver to resolve any such conflict.

4.2.3 Modeling the ATCO policy Problem Specification

As stated in section 4.2.2, the problem of ATCOs' reaction prediction concerns predicting $whether$, $when$, and how ATCOs will react to conflicts involving a particular aircraft executing the (focal) trajectory T_f^t, and aircraft flight trajectories in $Neigh(T_f, SA, t)$, given a spatial area of responsibility SA.

The ATCO reaction prediction problem involves (a) the detection of potential conflicts with identified trajectories in $Neigh(T_f, SA, t)$, (b) deciding the time point t_c for issuing a resolution action, given the time points at which conflicts occur, and (c) deciding the resolution action to be applied at that time point, thus shaping the future evolution of the trajectory for resolving conflicts.

The problem of *modeling the ATCO policy* refers to (c) and is about predicting the resolution action the ATCO would decide. Specifically, the problem of *modeling the ATCO policy* is about deciding at any time point t_c *how* the ATCOs will react, i.e., what conflict resolution action will apply in a focal trajectory $T_f^{1:t_c}$, given conflicts involving that trajectory and trajectories in $Neigh(T_f, SA, t_c)$.

As already discussed in section 4.2.1, ATCO events indicate the $type$ of the conflict resolution action instructed, e.g., speed change, and do not indicate further details about the resolution action. The actual resolution action cannot be revealed from the available data sources, due to a lack of information regarding the evolution of the trajectories if no resolution action applies, in conjunction with the uncertainty of how the trajectory evolves. Therefore, the specific problem addressed here is about predicting, at any time point t_c in the en route phase of flights, *the type of conflict resolution action* that ATCOs apply in $T_f^{1:t_c}$ (focal trajectory), given conflicts involving the trajectory $T_f^{1:t_c}$ and trajectories in $Neigh(T_f, SA, t_c)$.

Casting the imitation problem as a classification problem, conflicts are classified according to the type of ATCO resolution actions in a supervised way, according to demonstrated ATCO events. Here, the task does not take into account the evolution of the conflicts, but only the characteristics of the conflicts when they occur. This entails a major difference of the classification task from the "classical" ATCO imitation task: the classification task prescribes a resolution action as learned by demonstration samples, but without considering the effects of this action. This may provide limitations to the generalization abilities of the classification task, as it does not exploit the fact that in similar conflicting situations the most valuable actions are those that shape trajectories in ways similar to those demonstrated.

4.3 Methodology stages

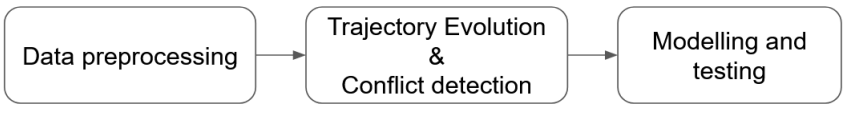


Figure 4.1: Methodology stages for predicting the ATCOs' reactions and the ATCOs' policy

This section describes the methodology proposed for predicting the ATCOs' reactions and the modeling of the ATCOs' policy. Figure 4.1 depicts the stages of this methodology, and subsequent sections describe each stage in detail, starting with the available data sources and their association. Briefly:

The *data pre-processing* stage associates the available data sources of historical, flown, and thus

conflicts-free trajectories and ATCO events. The pre-processed datasets are used for training and testing the AI/ML models.

The *trajectory evolution and conflict detection* stage estimates for any trajectory T, at any time point t, its potential evolution $\mathbf{T}_p^{t:t'}$ up to a time point $t' > t$, identifies potential neighbor trajectories within the area of responsibility, and computes features regarding the conflicts assessed to be associated with the historical ATCO events.

Finally, having detected potential conflicts associated with ATCO events, models for predicting the ATCOs' reactions and modeling the ATCOs' policy are trained and tested at the *modeling and testing* stage.

Subsequent sections present the data sources used, as well as each of the stages.

4.4 Data sources

Data sources comprise (a) surveillance data (IFS radar tracks) of operational quality, regarding flights' trajectories (provided from the Spanish ATC Platform SACTA, Automated System of Air Traffic Control (SACTA), (b) ATCOs' events that provide information regarding resolution actions assigned to flights (provided from ATON) and (c) sector configuration data (provided from the Spanish ATC Platform SACTA).

Sector configuration data provide the schedule of deployed sector configurations, as well as the catalog of possible sector configurations and are used in the sector related experimental setting, presented in section 4.8.1.1.

Surveillance data include radar track points per flight with temporal distance between consecutive points of approximately 5 seconds. The pre-processing phase, ensures a constant temporal distance between consecutive trajectory points, by interpolating trajectory points at time points with a value multiple of 5 seconds.

The surveillance dataset provides the aircraft position (longitude, latitude, altitude) and the timestamp at any point. Also, information that identifies a trajectory, such as the callsign, the origin and destination airports, is provided. Given the surveillance data set, an area SA and a particular trajectory, one can determine at any time point t the corresponding trajectory T^t within SA, i.e. the trajectory up to that time point, as well as $Neigh(T^t, SA, t)$.

The ATCOs' events dataset provides information regarding conflict resolution actions issued by ATCOs. It provides the callsign of the trajectory, the origin and destination airports, the timestamp of the resolution action and the type of the resolution action: $\langle$ *callsign, origin airport, destination airport, resolution action type*$\rangle$[5]. This information enables the association of ATCOs' conflict resolution actions with trajectory points.

Specifically, an ATCO's event for a resolution action RA is associated to a trajectory T, when the following conditions are satisfied:

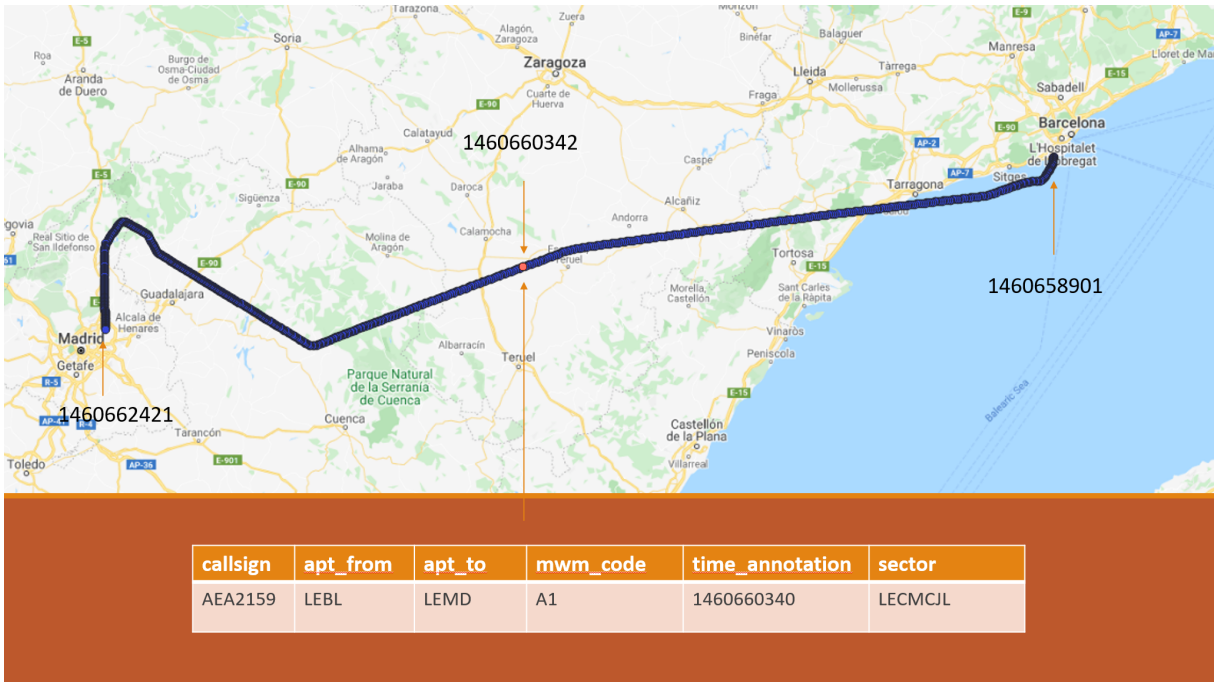
1. *RA.callsign = T.callsign*
2. *RA.departure_airport = T.departure_airport*

[5]This study considers only the type of the resolution action, as the computation of the ATCOs' policy for the exact resolution action is out of scope.

3. *RA.destination_airport = T.destination_airport*

4. Timestamp of the first T point $\leq$ timestamp of $RA \leq$ timestamp of the last T point.

Given that the above conditions hold for T and RA, the trajectory point that is temporally closer to the RA, is associated with the RA.



callsign	apt_from	apt_to	mwm_code	time_annotation	sector
AEA2159	LEBL	LEMD	A1	1460660340	LECMCJL

Figure 4.2: Trajectory points (blue points) associated with a corresponding ATCOs' event. The figure indicates the callsign, the departure (apt_from) and the destination airports (apt_to), the resolution action type (mwm_code), the time (time_annotation) and the sector in which the resolution was issued (sector). The (red) point in the middle of the trajectory depicts the point (with the timestamp 1460660342) associated to the ATCOs' event.

Figure 4.2 depicts an example of associating an ATCOs' event to the corresponding trajectory. The table reports the attributes of the ATCOs' event, the line is the trajectory and the point in the middle of the trajectory depicts the trajectory point which is closer in time to the ATCOs' resolution action, together with its timestamp (1460660342). Figure 4.2, indicates the timestamps of the ATCOs' event, of the first, and of the last point of the trajectory.

4.5 Trajectory states

Addressing the prediction of ATCOs' reactions and the ATCOs' policy modeling problems in a data-driven way, the training process necessitates having data associating ATCOs' observations regarding conflicts with specific reactions. In this stage, the aim is to reveal the conflicts that would occur if the ATCOs would not react at a time point t. It involves revealing the cases where a trajectory $T_f^{1:t}$ is in conflict with another trajectory $T_x^{1:t}$ and that are assessed to violate the separation minima according to estimated trajectory predictions, $T_{pf}^{t:t'}$ and $T_{px}^{t:t'}$.

To detect potential conflicts between trajectories, the CPA between pairs of flights is computed using the speed and course information included in the radar tracks. Then, the violation of separation minima is checked. The CPA is computed at the horizontal axes following the method-

ology presented in [59]. Having computed the CPA at the horizontal axes and the time to CPA, the vertical distance between the aircraft at the CPA is computed as follows. First the altitude of each aircraft at the CPA is calculated by multiplying the vertical speed of the aircraft by the time to CPA and adding their product to the aircraft's current altitude. Next the vertical distance between the aircraft at the CPA is calculated by subtracting their altitudes at the CPA and taking the absolute value of the difference.

Motivated by the bibliography on CD&R ([59, 60, 27, 17]), trajectory states include features (shown in Figure 4.3) comprising the relative bearing b_f with regard to a fixpoint (defined subsequently), the distance d_f from that fixpoint, the magnitudes of the aircraft horizontal (s_h) and vertical (s_v) speed, and the vector $v_i = \langle e_{i1}, \ldots, e_{ik} \rangle$, where each e_{ij} includes features of conflicts with neighbor trajectories T_j:

$$e_{ij} = \langle dh_{cpa_j}, dv_{cpa_j}, t_{cpa_j}, d_{cp_j}, t_{cp_j}, \sin(\mathrm{a}_j), \cos(\mathrm{a}_j), \sin(b_j), \cos(b_j) \rangle$$

As Figure 4.3 depicts, dh_{cpa_j} and dv_{cpa_j} are the horizontal and vertical distances of the ownship from an aircraft j at the CPA, and t_{cpa_j} is the time of the ownship to CPA. d_{cp_j} is the distance between the ownship and the aircraft j when the first of these is at the crossing point, and t_{cp_j} is the time until the first of the aircraft is at the crossing point. The intersection angle between the two trajectories is a_j, and b_j is the relative bearing of the ownship with regard to the aircraft j at the CPA. Considering the constraints CR presented in section 4.2.1, the following conditions hold w.r.t. the features in e_{ij}:

- $dh_{cpa_j} < cpa_{d_{th}}$;
- $dv_{cpa_j} < d_{v_{th}}$;
- $t_{cpa_j} < cpa_{t_{th}}$;
- $t_{cp_j} < ct_{th}$;

The $fixpoint$ is the 2D point at which the boundary of the considered spatiotemporal area SA crosses the line connecting the origin and the destination airports. The fixpoint provides a reference point and allows features to be independent from the airspace and OD pair considered. In doing so, the models trained are generic, supporting generalization beyond specific areas of responsibility and specific OD pairs.

Therefore, the state at a time point t of a flight trajectory T_i (ownship) is of the following form:

$$s_{r,t} = (< b_f, d_f, s_h, s_v >, t, < e_{i1}, \ldots, e_{ik} >)$$

where the observation vectors e_{ij} are defined for every $T_j \in Neigh(T_i, SA, t)$.

Neighbors are sorted in ascending order with regard to dh_{cpa_j} and the first k neighbors are considered. In this work, k is set to 3.

4.6 Solving the ATCO's reaction problem

4.6.1 Simulating uncertainty in trajectory evolution

Addressing the ATCOs' reaction prediction problem in a data-driven way (i.e. based on historical data), necessitates associating ATCOs' observations with specific reactions. As already pointed out, these observations are not recorded in a historical data set and they concern ATCOs'

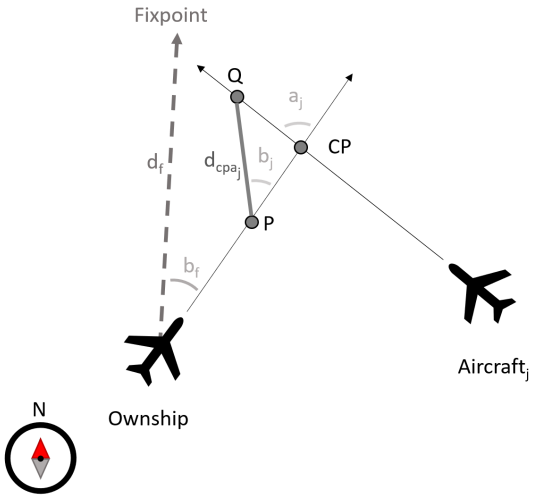


Figure 4.3: Features enriching points of the ownship's trajectory w.r.t. the aircraft flying a conflicting trajectory T_j.

estimations on the future evolution of trajectories, and conflicts detected between neighboring trajectories.

Determining these observations, involves detecting the conflicts that would occur if the ATCO would not react at a time point t. It involves estimating the cases where two trajectories T_1^t and T_2^t are in conflict and will violate the separation minima according to assessed trajectories' evolution T_{p1}^t and T_{p1}^t.

This is not trivial as the state of the aircraft after t is uncertain, and this uncertainty grows as the temporal horizon of estimation increases.

To estimate the potential evolution of a trajectory, the potential course and the potential horizontal speed of the aircraft are estimated. To do so, simple statistics on $\Delta course$ and Δs_h are computed from the historical trajectories. $\Delta course$ is the difference $course_{t+1} - course_t$, where $course_t$ is the aircraft's course at time point t, and Δs_h is the difference $s_{h_{t+1}} - s_{h_t}$, where s_{h_t} is the magnitude of the horizontal speed at time point t.

More specifically, given trajectories T_E in the surveillance data set, Δc and Δs_h are computed at each time point. Values are divided in n equi-height bins, where n is a hyper-parameter, set to 20. Using the median of each bin, this results to n values for the potential course deviation, and n values for the potential horizontal speed deviation, at any point.

Overall, given a trajectory T up to the current time point t, i.e., T^t, the course ($course^t$) of the aircraft and the magnitude of its speed ($speed^t$) at point s_t are computed by exploiting information provided at s_t and s_{t-1}. Adding zero or any of the n values for the potential deviations of $course^t$ and $speed^t$, results to $(n+1) * (n+1)$ potential trajectory evolutions in $\mathbf{T}_p^t$.

These potential trajectory evolutions are used in the CPA computation as follows. Given two aircraft i and j, and multiple potential trajectories evolutions at any time point t, $\mathbf{T}_{pi}^t$ and $\mathbf{T}_{pj}^t$, only the $T_{pi}^t \in \mathbf{T}_{pi}^t$ and the $T_{pj}^t \in \mathbf{T}_{pj}^t$ that result to the minimum horizontal distance of the flights at the CPA are considered: All trajectory state features e_{ij} regarding the neighboring trajectories T_i and T_j are computed based on those specific potential evolutions. The trajectories evolutions T_{pi}^t and the T_{pj}^t that result to the minimum horizontal distance of the flights at the CPA, are

determined by computing the corresponding CPA for each potential evolution, following the methodology presented in [59], and comparing the horizontal distances at the CPAs.

4.6.2 ATCO modes and resolution actions

As far as the ATCOs' reactions are concerned, as already pointed out, two levels of abstractions are considered: Modes of behavior and types of conflict resolution actions.

The set of modes comprises three high-level reactions:

- C_0: No conflicts detected, and no resolution action is applied.

- C_1: At least one conflict is detected, and a resolution action is applied.

- C_2: At least one conflict is detected but no resolution action is applied.

The last mode indicates ATCOs' tolerance to some conflicts, and allows delaying reactions after detecting conflicts and assessing the safety-criticality of a situation.

Types of conflict resolution actions are the following:

- A_0: "No resolution action"

- A_1: "Speed change"

- A_2: "Direct to waypoint"

Although the main focus here is to predict the mode of the ATCOs' behavior at specific times, the model is trained to predict categorical types of conflict resolution actions $A_i, i = 0, 1, 2$, as well as continuous actions regarding the trajectory evolution. The set of continuous actions comprise the change in course $\Delta course$, the change in the horizontal and vertical speeds Δs_h and Δs_v, and the time to the next point Δt. Subsequently, any action (either categorical or continuous) is denoted by a.

ATCOs' actions are shown to be important towards training models of timely ATCOs' reactions, and specify the policy of the ATCOs in conjunction to the policy of trajectory evolution at a fine level of detail.

4.6.3 Learning timely reactions

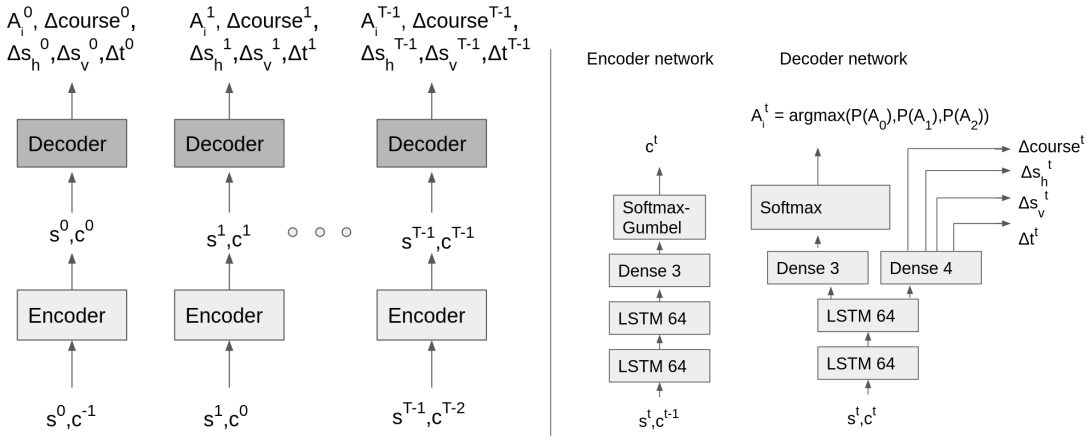
Motivated by the Hierarchical Reinforcement Learning literature [18], [76], [49] this study considers a hierarchical structure of ATCO reactions where abstract high-level reactions, corresponding to modes of the ATCO behavior (indicating whether to issue a resolution action in the presence of a conflict) are refined by means of fine low-level reactions that imply the evolution of aircraft state in a specific manner.

This hierarchical structure can be modeled by VAEs straightforwardly, as demonstrated in Directed InfoGAIL [71]. In the context of Directed InfoGAIL the encoder approximates the true posterior $p(c^t|c^{1:t-1}, T^{1:|T|})$, predicting the latent codes while performing tasks. The decoder learns a policy, generating actions, given the state and the predicted latent code, according to the demonstrated examples. Thus, the VAE provides a hierarchical structure, where the encoder predicts the mode of behaviour c (high-level actions), and the decoder predicts the policy (low-level) actions $\pi(a|s, c)$, given the state s and the predicted c.

In this study the VAE model is trained to imitate the demonstrated ATCOs' policy in a supervised way. Modes of behavior are decided by the encoder. These are exploited by the policy, which is represented by the Decoder network, prescribing low-level conflict resolution actions.

The encoder and decoder networks are trained by exploiting enriched trajectory points, the associated ATCOs' reaction modes and resolution actions.

Regarding the overall architecture of the method, as shown in Figure 4.4a, the encoder network, given the mode c predicted at time point $t-1$ and the state s at time point t, predicts the mode at the current time point t. The decoder network takes as input the predicted mode and the state at the current time point t and predicts the probabilities of low-level, categorical and continuous actions at time point t. Figure 4.4 shows the modes c^t, categorical actions a^t and continuous actions $\Delta course$, Δs_v, Δs_h, Δt predicted at each time point t by the encoder and decoder networks given the state s^t and mode c^{t-1}.



(a) (b)

Figure 4.4: Modes c^t, categorical actions a^t and continuous actions $\Delta course, \Delta s_v, \Delta s_h, \Delta t$ predicted at each time point t by the encoder and decoder networks given the state s^t and mode c^{t-1}.Figure (a) shows the overall architecture of the method, while Figure (b) shows the architectures of the encoder and decoder in detail.

The errors regarding the predicted resolution actions propagate backwards from the decoder. The encoder aims to minimize the categorical cross entropy loss between the distribution of modes in the dataset and the distribution predicted by the encoder.

Formally, the loss function of VAE L_{VAE} is as follows:

$$\begin{aligned} L_{VAE}(\pi, q) = &- \mathbb{E}_{(c^t \sim q, (a^t, s^t) \sim p_{data})}[log\ \pi_\theta(a^t|s^t, c^t)] \\ &- \mathbb{E}_{((c^t, c^{t-1}, s^t) \sim p_{data})}[log\ q_\phi(c^t|c^{(t-1)}, s^t)] \end{aligned} \tag{4.1}$$

where π_θ is the decoder's policy, q_ϕ is the encoder network, a, s and c are the actions, states and modes, respectively, p_{data} denotes the data distribution and t the timestep.

To train the VAE, for the continuous low-level actions the MSE is minimized, and for the categorical actions the categorical cross entropy between the distribution of actions in the data set and the distribution of the decoder predictions is minimized.

As modes are categorical variables the Gumbel-softmax trick [40] is used to obtain samples from a categorical distribution.

Algorithm 5 specifies the overall VAE training algorithm.

Algorithm 5: VAE training algorithm

Randomly initialize encoder's weights ϕ, decoder's weights θ $c^{-1} = C_0$;
for *epoch in total epochs* **do**
  for *mini-batch in mini-batches* **do**
    for s^t *in mini-batch* **do**
      $c^t = q_\phi(\cdot|c^{(t-1)}, s^t)$ (Encoder prediction);
      $(A_i^t, \Delta course^t, \Delta s_h^t, \Delta s_v^t, \Delta t^t) = \pi_\theta(\cdot|s^t, c^t)$ (Decoder prediction);
    Compute for predictions made:;
    - $mse(\Delta)$: mean square error for continuous actions $\Delta course^t, \Delta s_h^t, \Delta s_v^t, \Delta t^t$;
    - $cross\_entropy(modes)$: cross entropy for modes c_t;
    - $cross\_entropy(Actions)$: cross entropy for categorical actions $A_i^t, i \in \{0, 1, 2\}$;
    Minimize $mse(\Delta) + cross\_ent(modes) + cross\_ent(Actions)$ w.r.t. θ, ϕ;

The encoder and decoder networks consist of two layers of 64 LSTM nodes each, with $tanh$ activation. Additionally, the dncoder has a dense output layer with linear activation and number of nodes equal to the number of high level actions. Similarly, the decoder comprises two dense output layers: One with linear activation and a number of nodes equal to the number of categorical actions, and another layer for the continuous actions, with linear activation and number of nodes equal to the number of types of continuous actions. To minimize the loss function for both the encoder and VAE, the Adam optimizer has been used. The network architecture was based on the architecture reported in [18].

Compared to the VAE specified in [71], presented in section 2.1.2.1, here the following amendments have been made: (a) $D_{KL}(q_\phi(c|x)||p(c))$ between the distribution $q_\phi(c|x)$ and the prior distribution $p(c)$ of modes is not minimized. (b) The uniform distribution of modes in [71] pushes the model to predict equal probabilities for the different modes: This does not work well in modelling ATCOs' reactions. Also (c) the log-likelihood of the state-mode pairs observed in the dataset w.r.t. the model's parameters is maximized. In practice this is done by minimizing the categorical cross entropy loss between the distribution of modes in the dataset and the distribution predicted by the encoder.

4.7 Solving the ATCOs' policy learning problem

4.7.1 ATCO resolution actions

The set of resolution action types considered are the following:

-A_1: "Speed change resolution action"

-A_2: "Direct to waypoint resolution action"

-A_3: "Radar vectoring resolution action"

These types, according to the ATCO events dataset, are the most frequent ones, providing most of the examples, constituting 96% of the total set of resolution actions in the en route phase of operations.

4.7.2 Modeling the ATCOs policy

According to the formulation of the ATCO policy learning problem as a classification task, the goal is to predict the type of the resolution action a prescribed by the ATCOs at any point t_c, where a state s_r with conflicts occurs. Therefore, the aim is to learn a model that maps states s_r to conflict resolution action types a. Formally, the inputs of the models are states, as specified in Section 4.5, i.e., $s_r = (\langle b_f, d_f, s_h, s_v \rangle, \langle e_{i1}, ..., e_{ik} \rangle)$, and the outputs are resolution action types. The samples for training the AI/ML models are states labeled with resolution action types.

This section reports the details of the classification methods used to model the ATCOs policy.

Neural Networks As discussed in section 2.1.1.1 NNs [8] are function approximators able to model complex non-linear functions, and have been applied with great success in many regression and classification problems, as well as for imitating experts' behavior using behavior cloning [16, 61]. Although closely related to the objective considered here, behavior cloning solves a sequential decision problem. In this work, given the historical samples, the classification models predict only the type of the resolution action at a specific conflicting state, without considering subsequent aircraft states. A NN is trained using gradient descent [3, 65, 10], tuning its learnable parameters towards optimizing a loss function based on the training examples provided. For the task of modeling the ATCOs' policy the cross-entropy loss is applied. Formally, the following objective is minimized:

$$L_{CE} = -\sum_i^N p(a|s_r) log p_\theta(a|s_r),$$

where a denotes the resolution action type, p the probability distribution of resolution action types given trajectory states, as revealed by the dataset, and p_θ the corresponding distribution as predicted by the model.

This "simple" NN classifier is also augmented with an attention module. The attention module is a convolution layer based on a multi-head dot product attention kernel [80] that models interactions between the ownship and the aircraft executing neighbor trajectories. Specifically, this module introduces vectors of learnable parameters (weights), denoted by W_Q, W_K, and W_V, for the projection of features into "queries", "keys" and "values", respectively. Dot product attention kernels model interactions by performing dot product multiplication between the query and key values. In the single head attention case, queries, keys, and values are represented with single vectors. In multi-head attention kernels, these vectors are split into a number of vectors, equal to the number of heads. Considering M attention heads, the interaction between the ownship i and one of its neighbors $j \in Neigh(i, SA, t_c)$ is modeled by the attention head att_{ij}^m, where m indexes one of the M heads, as follows:

$$att_{ij}^m = \frac{exp(a^{att} W_Q^m h_i^{att} (W_K^m h_j^{att})^T)}{\sum_{k \in Neigh(i,SA,t_c) \cup \{i\}} exp(a^{att} W_Q^m h_i^{att} (W_K^m h_k^{att})^T))} \tag{4.2}$$

where a^{att} is a scaling factor and W_Q is multiplied with a hidden representation h_i^{att} of the ownship's features $\langle b_f, d_f, s_h, s_v \rangle$ and W_K is multiplied with a hidden representation h_j^{att} of the ownship neighbours' features, e_{ij}. The hidden state h_i^{att} results from passing the input features $\langle b_f, d_f, s_h, s_v \rangle$ from an encoding layer, while h_j^{att} results from passing the input features e_{ij} from an encoding layer (a dense layer is used in this case). The M attention heads are combined to the output of the attention module as follows:

$$h_i' = \sigma(concatenate[\sum_{j \in Neigh(i,SA,t_c) \cup \{i\}} att_{ij}^m W_V^m h_j^{att}, \forall m \in M]) \tag{4.3}$$

where the σ function is a NN layer. In this study case query, key and value projections and the σ function are implemented using dense layers of 128 nodes each.

The architecture of the NN, with and without the attention module, and the hyperparameters, are specified in Figure 4.5.

Specifically, the NN classifier without the attention module comprises two dense hidden layers with 64 nodes, each with tanh activation and L2 (weight decay) regularization [9]. The NN with the attention module passes the output of the attention module to the NN classifier. Figure 4.5 specifies how the hyperparameters of the networks are set. In order to avoid overfitting, an early stopping mechanism has been used: This is a regularization technique that determines the best amount of epochs to train. Based on the early stopping algorithm described in [30], the model's training is stopped when the validation error does not improve, and the model is retrained on the training and validation sets for the best number of epochs.

As already pointed out in section 4.2.1, it is likely that labels (i.e., historical ATCOs resolution action types) of revealed conflicts are noisy, containing trajectory states that are wrongly associated with a resolution action type. Noisy labels can be introduced by (i) the data, as is often the case with real-world data, and by (ii) the conflict detection methodology, which may reveal conflicting situations that do not correspond to the actual ATCO events indicated in the dataset.

According to the survey presented in [72], noise is categorized as instance-independent label noise or instance-dependent label noise. Instance-independent label noise could depend on the label, called label-dependent or asymmetric noise, or could be uniform among all classes, called symmetric noise. Different methods deal with different types of noise in various ways. Some methods change the network architecture to model label noise [34], while others apply forms of regularization to increase models' robustness, as in [83]. Others propose robust loss functions, as in [52]; make adjustments to the loss function, as in [14]; or select training samples that are noise-free with high probability, as in [54].

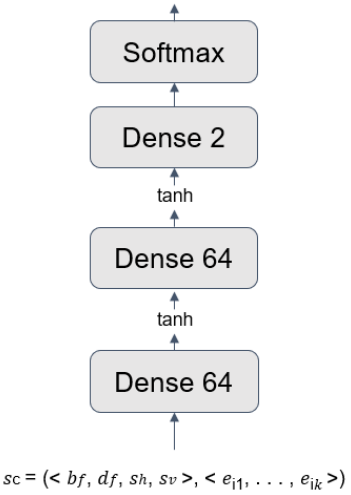
To address label noise, the Self-Evolution Average Label (SEAL) method was chosen. SEAL is a label refurbishment method presented in [14]. SEAL trains an NN multiple times from scratch. At each SEAL iteration, the model's output on all samples for each epoch is recorded. Then, for each sample, SEAL computes the average output value over all epochs. This value is considered to be an approximation of the true (not noisy) label and is used as the sample's label for the next SEAL iteration. In this study, five SEAL iterations are used.

SEAL has the following advantages: It can deal with high rates of noise, does not need a noise-free validation set (in contrast to other methods), uses all training samples (in contrast to sample selection methods), and is robust to instance-dependent noise, which is the most complex form of noise. SEAL is a good fit for the case considered in this study, as a noise-free subset of the dataset is not available and label noise is likely to be instance-dependent.

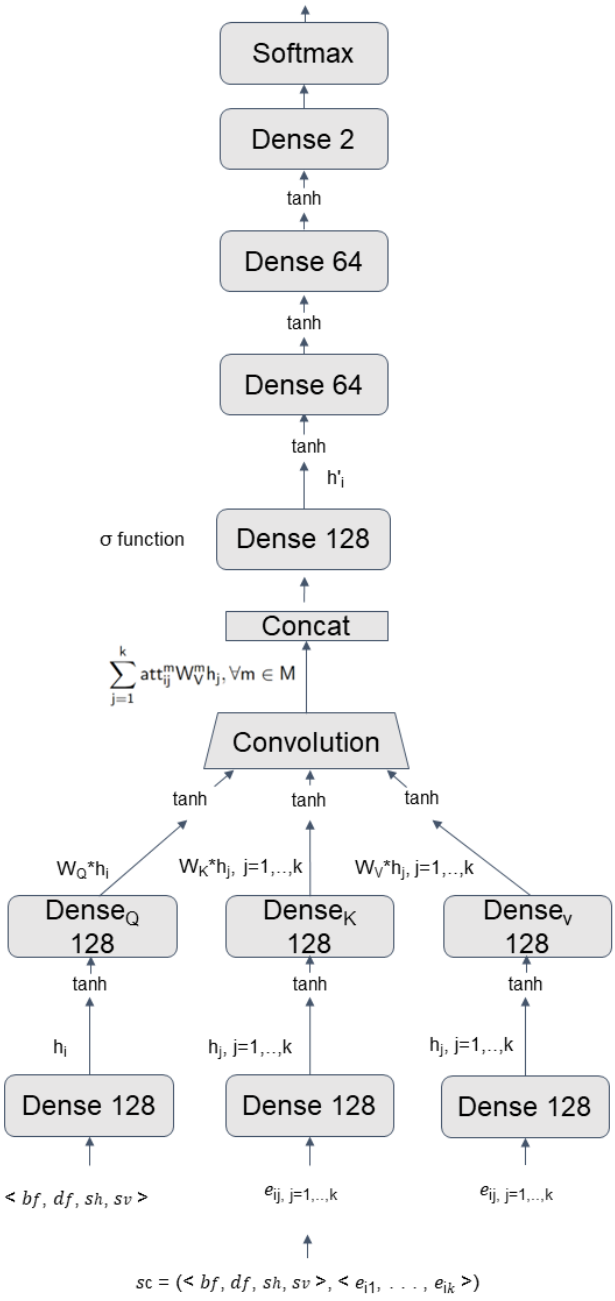
Active-passive loss functions presented in [52] provide an alternative way to address label noise, and were also tested in this work. An active-passive loss function is the weighted sum of an active and a passive loss function. Formally, it has the following form, $\Psi_{AP} = \alpha^{AP} * \Psi_{Active} + \beta^{AP} * \Psi_{Passive}$, where $\alpha^{AP}, \beta^{AP} > 0$ are coefficients that balance the two loss functions and ψ denotes a loss function. A loss function is considered active if it only optimizes the model's learnable parameters with regard to the correct labels of the sample. It aims to increase the probability the model assigns to the sample's label. Passive loss functions aim to decrease the probability the model assigns to at least one incorrect label. For the active-passive loss function to be robust to label noise, both the active and the passive loss functions should be robust. Functions

Hyper-parameter	Description	Value
Learning rate	Scales the step of the optimization	0.0003
Loss function	The minimization objective	Cross entropy

(a) NN hyperparameters



(b) The NN classifier architecture



(c) The NN+att classifier architecture

Figure 4.5: The Neural Network (NN) and Neural Network with attention (NN+att) hyperparameters (**a**), the NN classifier without attention (**b**), and the NN classifier with attention (**c**). $Dense_Q$, $Dense_K$, and $Dense_V$ denote the query, key, and value projections, respectively.

that are not robust to label noise can be made robust using the following normalization form, $\Psi_{norm} = \Psi(f(s_r), a)/sum_{j=1}^{K}\Psi(f(s_r), j)$, where f denotes the model, s_r is the model's input, a represents the corresponding labels and K is the number of different labels. In this study the following active-passive loss function combinations from [52] were used: Normalized Focal Loss (NFL) + Mean Absolute Error (MAE), NFL + Reverse Cross Entropy (RCE), Normalized Cross Entropy (NCE) + MAE, NCE+RCE. Table 4.2 reports the active-passive loss functions used for conciseness.

Table 4.2: Active and passive Loss functions. p_θ denotes the probability output of the model, p the distribution over different labels as revealed by the dataset and K is the number of labels.

Name	Formula	type
NFL	$\frac{-\sum_{k=1}^{K} p(k\|s_r)(1-p_\theta(k\|s_r))^\gamma log p_\theta(k\|s_r)}{-\sum_{j=1}^{K}\sum_{k=1}^{K} p(y=j\|s_r)(1-p_\theta(k\|s_r))^\gamma log p_\theta(k\|s_r)}$	Active
MAE	$\sum_{k=1}^{K} \|p_\theta(k\|s_r) - p(k\|s_r)\|$	Passive
RCE	$-\sum_{k=1}^{K} p_\theta(k\|s_r) log p(k\|s_r)$	Passive
NCE	$= \frac{-\sum_{k=1}^{K} p(k\|s_r) log p_\theta(k\|s_r)}{-\sum_{j=1}^{K}\sum_{k=1}^{K} p(y=j\|s_r) log p_\theta(k\|s_r)}$	Active

Random Forest and Gradient Tree Boosting As discussed in section 2.1.1.2, DTs [46] are tree-like models used for classification and regression. This work employs decision tree ensembles such as RF and GTB for classification.

Table 4.3 reports the hyperparameters used in creating DTs for the RF and GTB ensembles used in this work.

The RF [11] implementation used in this study uses bootstrapping and random input selection in training. The ensembled prediction is computed as the average predicted probability for each class of the decision trees.

As in many real-world datasets, the available ATCO events dataset is imbalanced, as samples corresponding to a resolution action type, specifically resolution action type A_3, constitute a small proportion of the data. Many classification algorithms cannot accurately predict the minority class of imbalanced datasets as they minimize the overall error and tend to ignore rare samples. In [13], the authors proposed balanced RF and weighted RF to deal with the class imbalance problem. Balanced RF balances the bootstrap samples by randomly selecting the same sample number for all classes. On the other hand, weighted RF assigns weights to each class, giving higher weights to minority classes in order to penalize errors made on samples of the minority classes more heavily.

In this work, the RF and weighted RF implementation of scikit-learn and the balanced RF implementation of imbalanced-learn are used. The hyperparameters have been set as shown in Tables 4.3 and 4.4.

GTB, which is used in this work, is a specific case of GB where decision trees [46] are used as the base learners.

In this work, the GTB implementation of scikit-learn is used. Tables 4.3 and 4.5 report on hyperparameters.

4.8 Experimental Evaluation

This section presents the experimental evaluation of the methods applied on the problems of predicting the ATCOs' reactions and modeling the ATCOs' policy. Methods for predicting ATCOs' reactions are evaluated in section 4.8.1, while methods for modeling the ATCOs' policy are evaluated in section 4.8.2.

Table 4.3: Hyperparameters of the decision trees used for the Random Forest (RF) and Gradient Tree Boosting (GTB) algorithms. Descriptions are from scikit-learn.

Hyperparameter	Description	RF Value	GTB Value
criterion	The function to measure the quality of a split.	gini	friedman mse
max_depth	The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure (containing one class only) or until all leaves contain less than min_samples_split samples.	None	3
min_samples_split	The minimum number of samples required to split an internal node.	2	2
min_samples_leaf	The minimum number of samples required to be at a leaf node.	1	2
min_weight_fraction_leaf	The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node.	0	0
max_features	The number of features to consider when looking for the best split. If “sqrt”, then max_features is equal to the square root of the total number of features. If None, then max_features is equal to the number of total features.	sqrt	None
max_leaf_nodes	Trees grown with max_leaf_nodes in best-first fashion. Best nodes are defined as a relative reduction in impurity. If None then unlimited number of leaf nodes.	None	None
min_impurity_decrease	A node will be split if this split induces a decrease in the impurity greater than or equal to this value.	0	0
ccp_alpha	Complexity parameter used for minimal cost-complexity pruning. The subtree with the largest cost complexity that is smaller than ccp_alpha will be chosen. When 0, no pruning is performed.	0	0

4.8.1 Predicting ATCOs’ reactions

This section presents the experimental evaluation of the methods used for predicting the ATCOs’ reactions. This section is structured as follows: first subsection 4.8.1.1 presents the experimental setting, then subsection 4.8.1.2 presents the datasets and the pre-processing, subsection 4.8.1.3 presents the evaluation methodology and subsection 4.8.1.4 reports the experimental results.

Table 4.4: Hyperparameters used for the Random Forest (RF) algorithm. Descriptions are from scikit-learn.

Hyperparameter	Description	Value
n_estimators	The number of trees in the ensemble.	100
bootstrap	Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.	True
oob_score	Whether to use out-of-bag samples (samples that have not been used when bootstrapping) to estimate the generalization score.	False
class_weight	Weights associated with classes. If None, all classes are supposed to have weight one.	None
max_samples	If bootstrap is True, draw a number of samples from the training set to train each base estimator. If None, then draw a number of samples equal to the size of the training set.	None
random_state	Controls both the randomness of the bootstrapping of the samples used when building trees (if bootstrap=True) and the sampling of the features to consider when looking for the best split at each node (if max_features < total number of features).	None

4.8.1.1 Experimental Setting

The proposed method is evaluated in two different types of settings w.r.t. the Area of Responsibility (AoR) chosen: a) The sector-related setting, and b) the sector-ignorant setting simulating the flight-centric concept.

In the sector-related case the AoR SA corresponds to a sector crossed by the trajectory of the ownship. Given that neighboring flights are all flights in SA following the constraints in CR (according to the definition of neighboring flights), the horizontal distance threshold D_{th} used to define neighboring flights is set to infinity.

In the sector-ignorant case, a rectangular area covering the Iberian Peninsula, which is the region to which the dataset refers to, is segregated in cells of size 0.5 degrees longitude and latitude in order to create an index of the positions of flights in each cell at each time point. This allows fast access to the flights of each cell at each time point, making the identification of neighboring flights more efficient in terms of computational time. For each trajectory point in this area the neighboring flights w.r.t. a focal trajectory are limited to those with a distance threshold D_{th} of 5 cells in the longitude (approx. 231 km) and latitude (approx. 308km) dimensions. In this

Table 4.5: Hyperparameters of the gradient tree boosting algorithm. Descriptions are from scikit-learn.

Hyperparameter	Description	Value
n_estimators	The number of trees in the ensemble.	100
loss	The loss function to be optimized.	log_loss
learning rate	Shrinks the contribution of each tree.	0.1
subsample	The fraction of samples to be used for fitting the individual base learners.	1
init	An estimator that is used to compute the initial predictions. If None, the initial estimator predicts the classes' priors.	None
validation_fraction	The proportion of training data to set aside as the validation set for early stopping. Only used if n_iter_no_change is set to an integer.	Not used
n_iter_no_change	Used to decide if early stopping will be used to terminate training when the validation score does not improve. If None, early stopping is disabled.	None
random_state	Controls the random seed given to each Tree estimator at each boosting iteration. In addition, it controls the random permutation of the features at each split.	None
tol	Tolerance for early stopping. When the loss is not improving by at least tol for n_iter_no_change iterations (if set to a number), the training stops.	1e-4

case, the area defined by the area covering the Iberian Peninsula and D_{th} specifies SA and follows the movement of the ownship. The value of D_{th} was chosen empirically in order for SA to include a sector and adjacent sectors, so as to detect as many conflicts as possible in the airspace of interest.

Figure 4.6 shows the SA area considered in the sector-ignorant case (area covered by the grid), the focal trajectory of the ownship (red trajectory or dark gray in grayscale) and neighboring trajectories in $Neigh(T_f, SA, t)$. The ownship's position at time t is shown in white (middle point). The area defined by D_{th} w.r.t. the ownship's position is depicted by the inner (red) rectangle. The (yellow) dot in the upper part of the grid is a fixpoint. Specifically, the fixpoint is the point at which the edge of the SA box towards the destination airport crosses the line connecting the origin and the destination airports.

Figure 4.6: The SA area and the area defined by D_{th} (red rectangular area) w.r.t. the ownship's position (white dot) in the sector-ignorant case.

4.8.1.2 Data sets and Pre-processing

To evaluate methods for solving the ATCOs' reaction prediction problem this study exploits data from the Spanish airspace, considering flights over Spain, without sacrificing the generality of the methods introduced. The data sources comprise:

- Surveillance data: operational quality data with actual flights (raw) trajectories (Spanish ATC Platform SACTA).
- Sector configuration data: the schedule of deployed sector configurations, as well as the catalog of possible sector configurations (Spanish ATC Platform SACTA), used for the sector-related setting.
- ATCO events: provides actions taken by the ATCOs in order to ensure safety of flights (provided by ATON).

As discussed in section 4.6.2, the trajectory points of all trajectories in T_E are annotated using the modes C_0 ("No conflicts detected, and no resolution action has been applied"), C_1 ("At least one conflict is detected, and a resolution action has been applied"), and C_2 ("At least one conflict is detected but no resolution action has been applied").

In so doing, two problems occur:

1. The dataset is imbalanced regarding the modes. The typical case is to have one resolution action and one point with C_1 mode for a trajectory with 700 points (699 points corresponding to modes C_0 and C_2).
2. Following a data-driven approach and exploiting data with flown (thus conflicts-free) trajectories, according to the methodology presented in Section 4.6.1, there are cases where there is an ATCOs' resolution action for a trajectory but no conflicts are detected.

Table 4.6: Prior distribution of modes (C_0, C_1, C_2) for different subsampling $step$ values computed on the dataset for the sector-ignorant experimental setting.

$\Delta step$	C_0	C_1	C_2
1	0.63933797	0.05469366	0.30596837
2	0.61360892	0.10181661	0.28457447
3	0.59235067	0.14231008	0.26533926
4	0.57311712	0.17803213	0.24885076
5	0.55696083	0.20905254	0.23398663
6	0.54101562	0.23763021	0.22135417

The first problem is tackled by data augmentation and subsampling: The trajectory points in a time window of 250 seconds before the ATCOs' resolution action are annotated with C_1, excluding the points at which no conflicts are detected. This, somehow addresses the uncertainty of ATCOs about the time to issue a resolution action. The point at which the resolution action was issued is called the *actual* Resolution Action Trajectory Point (RATP) and the annotated points due to data augmentation, before the *actual* RATP, are called *annotated* Resolution Action Trajectory Points (RATPs). Subsampling is applied to trajectory points with modes C_0 and C_2, keeping one trajectory point with any of these modes every $\Delta step$ trajectory points. Table 4.6 reports the prior distribution of modes C_0, C_1 and C_2 for different $\Delta step$ sizes considering the dataset of the sector-ignorant experimental setting. Given these distributions, after experimentation, $\Delta step$ is set equal to 6 (i.e. corresponding to 30 seconds).

Regarding the second problem, trajectories with an associated ATCOs' resolution action but with no detected conflicts in a time window of $window\_duration$ seconds before the actual RATP, are filtered out. The trajectory point in the specified time window at which at least one conflict is detected and is temporally closest to the point with the ATCOs' resolution action, is indicated to be the *actual* RATP. In this study $window\_duration$ is set to 70s. The evaluation methodology follows consistently this choice, as described in Section 4.8.1.3.

The dataset contains trajectories between 5 different OD pairs, all from 2017: Malaga (LEMG) - Gatwick (EGKK), Malaga (LEMG) - Amsterdam (EHAM), Lisbon (LPPT) - Paris (LFPO), Zurich (LSZH) - Lisbon (LPPT) and Geneva (LSGG) - Lisbon (LPPT). Only ATCOs' resolution actions issued at the en-route phase of operations are considered here, and the climb and descent parts of the trajectories are filtered out. In addition, only trajectories that have at least one ATCOs' resolution action and an associated actual RATP are considered. This results to 255 enriched trajectories corresponding to 344 resolution actions for the sector relevant case and 668 trajectories corresponding to 791 resolution actions for the sector-ignorant case. It must be noted here that the available ATCOs' events dataset covers the Spanish airspace and thus only the points of the trajectories that are in this airspace are considered. However, the proposed method is generic and can be applied in any airspace, independently of the configuration of sectors, as experiments in the sector-ignorant setting show.

4.8.1.3 Evaluation methodology

To evaluate the accuracy of predictions made by the proposed models, weighted versions of precision and recall are defined. Weighted versions of precision and recall penalize predicted RATPs based on their temporal distance to the actual or annotated RATPs. Doing so, provides the flexibility needed for comparing the predicted RATPs against the actual and annotated

RATPs. This flexibility is necessary, as ATCOs' timely reactions may differ in the same situation, if this situation occurs at different times and/or for different ATCOs. Weighted measures use a score function that takes as input the temporal distance between RATPs using a Gaussian distribution with std =5, as justified below.

Formally, the score function is as follows:

$$score(x) = \frac{\frac{1}{5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x}{5n})^2}}{\frac{1}{5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{0}{5n})^2}} = e^{-\frac{1}{2}(\frac{x}{5n})^2} \tag{4.4}$$

This function is depicted in Figure 4.7, taking values between 0 and 1. The parameter n is a factor translating the temporal distance x to a number of n-sec-intervals corresponding to points in the x-axis. The parameter n is set equal to 5, given that decisions are made using the temporal granularity of 5 seconds. By using different values for the standard deviation and n one can tune this score function, which may be considered to estimate the probability that an ATCO's reaction happens with a temporal difference x compared to the ATCOs' reaction recorded in the data.

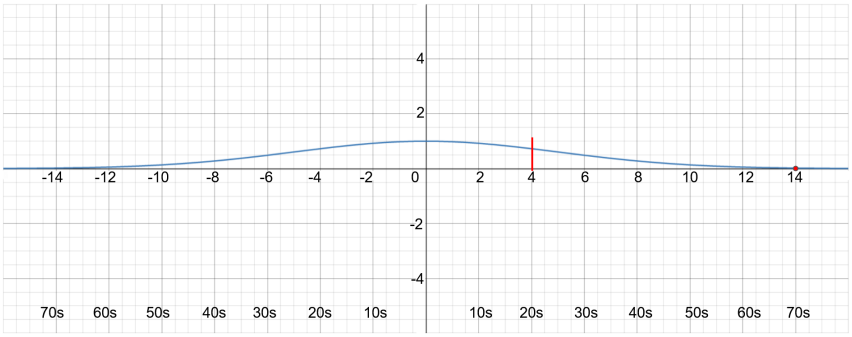


Figure 4.7: Score function: Axis x values correspond to x/n, when $n = 5$, and the temporal distance x in seconds is shown at the bottom. Axis y shows the score.

Specifically, setting n to 5 the temporal distance is synced to the change of slope when x=20 seconds, at $x/5$=4, indicated by a small (red) line in Figure 4.7 at x-axis point 4: Thus, the score is reduced more drastically when the temporal distance between the predicted and the actual reactions is greater than 20 seconds. The parameter n also controls the time window out of which the score is approximately 0. For n=5 this is set to approx. 70s, shown in Figure 4.7 with the (red) dot at x-axis point 14. This is also the value used for the $window\_duration$ used when setting the actual RATP during preprocessing, as mentioned in Section 4.8.1.2.

Next paragraphs discuss how the score function is used to calculate precision and recall in experiments, weighting true/false positives/negatives per type of reaction. For example, let us consider a case where the model reacts to a conflict few seconds later or earlier than the reaction recorded in the dataset. This, due to the small temporal distance between the predicted and the actual RATPs, will be penalized lightly, assuming that both decisions were driven by the same contextual features. On the other hand, a large temporal distance may be due to different contexts in which reactions occur, or to the lack of model's capability to react at the right time. In either case, the predicted reaction differs considerably from the actual one.

To explain the evaluation scheme, let us consider the (high and low level) reactions in their most general form. The following two generic classes of ATCOs' reactions are distinguished:

- G_0: Not Assigning Resolution Action

- G_1: Assigning Resolution Action

Class G_0 includes modes C_0 and C_2 and resolution action A_0, whereas class G_1 includes mode C_1 and resolution actions A_1,A_2. Thus, both G_0 and G_1 cases are specialized by subclasses of modes and low-level types of ATCOs' resolution actions.

Subsequently, to present the evaluation methodology in its general form, $G_{0_j}, j = 1, .., k$ and $G_{1_j}, j = 1, .., l$ denote the subclasses of each class G_0 and G_1, respectively, without committing to specific subclasses of ATCOs' reactions.

Tables 4.7 and 4.8 describe in detail how false and true positives (FP, TP), as well as false and true negatives (FN, TN) for G_0 and G_1 are weighted when computing precision and recall. The weighting scheme applies to either modes or ATCOs' resolution actions, as these tables present.

Table 4.7: False Positives (FP) and True Positives (TP) weights based on the score function, given the time point of the prediction t_p and the time point t_a of the closest actual RATP or the time point t_{aa} of the closest actual or annotated RATP. Elements on the diagonal of the table are true positives (TP) and dashes indicate that the weighted measures do not apply between modes and resolution actions.

actual \ predicted		G_0			G_1		
		C_0	C_2	A_0	C_1	A_1	$A2$
G_0	C_0	$w_{TP} = 1$ $w_{FP} = 0$	$w_{FP} = 1$ $w_{TP} = 0$	-	$w_{FP} = 1 - score(\lvert t_p - t_{aa}\rvert)$ $w_{TP} = 1 - w_{FP}$	-	-
	C_2	$w_{FP} = 1$ $w_{TP} = 0$	$w_{TP} = 1$ $w_{FP} = 0$	-	$w_{FP} = 1 - score(\lvert t_p - t_{aa}\rvert)$ $w_{TP} = 1 - w_{FP}$	-	-
	A_0	-	-	$w_{TP} = 1$ $w_{FP} = 0$	-	$w_{FP} = 1 - score(\lvert t_p - t_{aa}\rvert)$ $w_{TP} = 1 - w_{FP}$	$w_{FP} = 1 - score(\lvert t_p - t_{aa}\rvert)$ $w_{TP} = 1 - w_{FP}$
G_1	C_1	$w_{FP} = score(\lvert t_p - t_a\rvert)$ $w_{TP} = 1 - w_{FP}$	$w_{FP} = score(\lvert t_p - t_a\rvert)$ $w_{TP} = 1 - w_{FP}$	-	$w_{TP} = 1$ $w_{FP} = 0$	-	-
	A_1	-	-	$w_{FP} = score(\lvert t_p - t_a\rvert)$ $w_{TP} = 1 - w_{FP}$	-	$w_{TP} = 1$ $w_{FP} = 0$	$w_{FP} = 1$ $w_{TP} = 0$
	A_2	-	-	$w_{FP} = score(\lvert t_p - t_a\rvert)$ $w_{TP} = 1 - w_{FP}$	-	$w_{FP} = 1$ $w_{TP} = 0$	$w_{TP} = 1$ $w_{FP} = 0$

Table 4.8: False Negatives (FN) and True Negatives (TN) weights based on the score function, given the time point of the prediction t_p and the time point t_a of the closest actual RATP or the time point t_{aa} of the closest actual or annotated RATP. Elements on the diagonal of the table are true positives (TP) and dashes indicate that the weighted measures do not apply between modes and resolution actions.

actual \ predicted		G_0			G_1		
		C_0	C_2	A_0	C_1	A_1	$A2$
G_0	C_0	$w_{TP} = 1$ $w_{FP} = 0$	$w_{FN} = 1$ $w_{TN} = 0$	-	$w_{FN} = 1 - score(\lvert t_p - t_{aa}\rvert)$ $w_{TN} = 1 - w_{FN}$	-	-
	C_2	$w_{FN} = 1$ $w_{TN} = 0$	$w_{TP} = 1$ $w_{FP} = 0$	-	$w_{FN} = 1 - score(\lvert t_p - t_{aa}\rvert)$ $w_{TN} = 1 - w_{FN}$	-	-
	A_0	-	-	$w_{TP} = 1$ $w_{FP} = 0$	-	$w_{FN} = 1 - score(\lvert t_p - t_{aa}\rvert)$ $w_{TN} = 1 - w_{FN}$	$w_{FN} = 1 - score(\lvert t_p - t_{aa}\rvert)$ $w_{TN} = 1 - w_{FN}$
G_1	C_1	$w_{FN} = score(\lvert t_p - t_a\rvert)$ $w_{TN} = 1 - w_{FN}$	$w_{FN} = score(\lvert t_p - t_a\rvert)$ $w_{TN} = 1 - w_{FN}$	-	$w_{TP} = 1$ $w_{FP} = 0$	-	-
	A_1	-	-	$w_{FN} = score(\lvert t_p - t_a\rvert)$ $w_{TN} = 1 - w_{FN}$	-	$w_{TP} = 1$ $w_{FP} = 0$	$w_{FN} = 1$ $w_{TN} = 0$
	A_2	-	-	$w_{FN} = score(\lvert t_p - t_a\rvert)$ $w_{TN} = 1 - w_{FN}$	-	$w_{FN} = 1$ $w_{TN} = 0$	$w_{TP} = 1$ $w_{FP} = 0$

Specifically, the tables distinguish the following cases:

1. **False positives/negatives** of a subclass of G_0.

1(a) *False Positives (FP)*: In this case the model falsely predicts a subclass G_{0_j} of G_0, while the dataset indicates either (i) G_1 "Assigning Resolution Action", or (ii) a subclass G_{0_k} of G_0 with $j \neq k$.

1(b) *False Negatives (FN)*: In this case the model falsely predicts either (i) G_1 "Assigning Resolution Action", or (ii) a subclass of G_0, G_{0_k}, while the dataset indicates the subclass G_{0_j}, $k \neq j$.

2. **False positives/negatives** of a subclass of G_1.

2(a) *False Positives (FP)*: In this case the model falsely predicts either (i) a subclass G_{1_j} of G_1 "Assigning Resolution Action" while the dataset indicates either G_0 "Not Assigning Resolution Action", or (ii) a subclass G_{1_k} of G_1 with $j \neq k$.

2(b) *False Negatives (FN)*: In this case the dataset indicates G_{1_j} but the model falsely predicts either (i) G_0 "Not Assigning Resolution Action", or (ii) a subclass of G_1, G_{1_k} where $k \neq j$.

3. **True Positives** (TP) of a subclass of either G_0 or G_1. TP are those cases where the model correctly predicts a a subclass G_{i_j}, of class G_0 or G_1. At these cases a score w_{TP} equal to 1 is assigned. Additionally, given that FP are assessed with weight w_{FP}, then for the corresponding cases TP are assigned weight $(1 - w_{FP})$. Thus, TP are calculated using the following formula: $\sum_{i=1}^{\#TP} 1 + \sum_{i=1}^{\#FP}(1 - w_{FP_i})$.

4. **True Negatives** (TN) of a subclass of G_0 or G_1.

True negatives are those cases where the model correctly does not predict a subclass G_{i_j} of either G_0 or G_1. Then, it is scored with weight $w_{TN} = 1$. In addition, given that FN predicted are assessed with w_{FN}, the TN for the corresponding cases are assigned with weights $(1 - w_{FN})$. Thus, TN are calculated using the following formula: $\sum_{i=1}^{\#TN} 1 + \sum_{i=1}^{\#FN}(1 - w_{FN_i})$.

Considering the different cases presented in Tables 4.7 and 4.8 the weighted versions of precision, recall and f1-score, namely WP, WR and Wf1 respectively, are defined as follows:

$$WP = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{\#TP} 1 + \sum_{i=1}^{\#FP}(1 - w_{FP_i})}{[\sum_{i=1}^{\#TP} 1 + \sum_{i=1}^{\#FP}(1 - w_{FP_i})] + \sum_{i=1}^{\#FP} w_{FP_i}} = \frac{\sum_{i=1}^{\#TP} 1 + \sum_{i=1}^{\#FP}(1 - w_{FP_i})}{\sum_{i=1}^{\#TP} 1 + \sum_{i=1}^{\#FP} 1} \tag{4.5}$$

$$WR = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{\#TP} 1 + \sum_{i=1}^{\#FP}(1 - w_{FP_i})}{[\sum_{i=1}^{\#TP} 1 + \sum_{i=1}^{\#FP}(1 - w_{FP_i})] + \sum_{i=1}^{\#FN} w_{FN_i}} \tag{4.6}$$

$$Wf1 = 2 * \frac{WP * WR}{WP + WR} \tag{4.7}$$

In these formulae, #TP is the number of true positives, #FP is the number of false positives and #FN is the number of false negatives. It must be noted that when w_{FP_i} and w_{FN_i} are equal to 1, WP, WR and Wf1 revert to the standard precision, recall and f1 measures.

4.8.1.4 Experimental Results

Subsequent paragraphs report on the results achieved by the proposed VAE model in predicting ATCOs' reactions, and the results achieved by training the encoder network (baseline). This shows the difference in performance between the two methods, caused by the decoder's error backwards propagation, supporting the conjecture that to model the ATCOs' reactions one needs to jointly consider the resolution actions applied by the ATCOs. The VAE and the baseline methods are evaluated by running 10 experiments with two times repeated 5-fold cross validation, training the models for 1000 epochs per experiment, with mini-batches of size 32, using the real-world pre-processed data. Results report on the 95% confidence interval of the non-weighted and weighted precision, recall and f1-scores. In chapter 5 the VAE model will also be compared to a RF model when considering the problem of combining different models towards a conflicts-free trajectories planning method. As will be discussed for that case a constant time step between trajectory points is necessary and VAE is compared against other methods to evaluate its performance.

Table 4.9: Experimental results of the sector-ignorant case achieved by the VAE and the encoder (Enc). Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes and the resolution actions of ATCOs, for the non-weighted and weighted measures.

model	**modes non-weighted**	**modes weighted**	**actions non-weighted**	**actions weighted**
VAE	precision	precision	precision	precision
	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.976 \pm 0.006$ $C_2 : 0.934 \pm 0.012$	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.982 \pm 0.005$ $C_2 : 0.990 \pm 0.000$	$A_0 : 0.975 \pm 0.004$ $A_1 : 0.635 \pm 0.028$ $A_2 : 0.646 \pm 0.035$	$A_0 : 0.998 \pm 0.003$ $A_1 : 0.640 \pm 0.026$ $A_2 : 0.653 \pm 0.035$
	recall	recall	recall	recall
	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.936 \pm 0.014$ $C_2 : 0.976 \pm 0.006$	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.989 \pm 0.002$ $C_2 : 0.983 \pm 0.005$	$A_0 : 0.990 \pm 0.000$ $A_1 : 0.549 \pm 0.026$ $A_2 : 0.670 \pm 0.022$	$A_0 : 0.993 \pm 0.003$ $A_1 : 0.589 \pm 0.028$ $A_2 : 0.709 \pm 0.021$
	f1-score	f1-score	f1-score	f1-score
	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.956 \pm 0.008$ $C_2 : 0.954 \pm 0.008$	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.985 \pm 0.004$ $C_2 : 0.986 \pm 0.004$	$A_0 : 0.985 \pm 0.004$ $A_1 : 0.588 \pm 0.017$ $A_2 : 0.656 \pm 0.021$	$A_0 : 0.993 \pm 0.003$ $A_1 : 0.610 \pm 0.017$ $A_2 : 0.679 \pm 0.023$
Enc	precision	precision	-	-
	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.950 \pm 0.010$ $C_2 : 0.870 \pm 0.038$	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.959 \pm 0.009$ $C_2 : 0.975 \pm 0.009$		
	recall	recall		
	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.863 \pm 0.053$ $C_2 : 0.951 \pm 0.011$	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.969 \pm 0.017$ $C_2 : 0.961 \pm 0.011$		
	f1-score	f1-score		
	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.904 \pm 0.032$ $C_2 : 0.909 \pm 0.020$	$C_0 : 1.000 \pm 0.000$ $C_1 : 0.964 \pm 0.009$ $C_2 : 0.967 \pm 0.006$		

Table 4.9 reports the 95% confidence interval of the precision, recall and f1-score, achieved by the VAE and the encoder (Enc) for the ATCOs' modes of behavior and the resolution actions, for the sector-ignorant case. Columns "modes non-weighted"/"actions non-weighted" and "modes weighted" / "actions weighted" report respectively on the non-weighted and the weighted ver-

sions of the measures for modes and resolution actions. As the encoder does not predict resolution actions, the corresponding columns are empty.

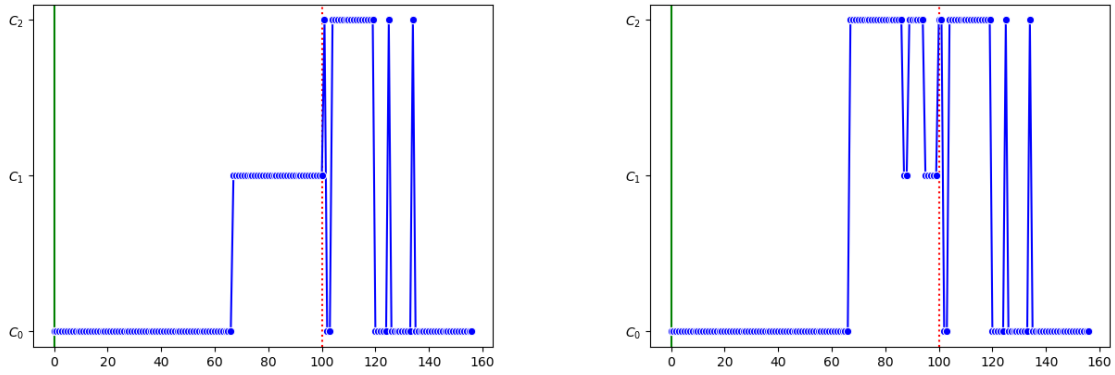
Regarding the modes of ATCOs' reaction, results show that both the VAE and the encoder achieve over 0.9 f1-score on all modes, for the non-weighted and the weighted measures, with VAE achieving the best results with a weighted f1-score greater or equal to 0.985 ± 0.004 on all modes. Also the VAE outperforms the encoder on all measures, weighted or not, although the encoder is really competitive. The largest differences between the models are observed w.r.t. the precision of mode C_2 and the recall of mode C_1 for the non-weighted measures. For mode C_2 VAE achieves a precision of 0.934 ± 0.012 whereas the encoder achieves a precision of 0.870 ± 0.038. For mode C_1 the VAE and the encoder achieve a recall of 0.936 ± 0.014 and 0.863 ± 0.053, respectively. This shows that there are cases where the encoder should assign a resolution action but it fails to do so, as it predicts mode C_2. For the VAE, such cases are rather rare.

An interesting observation is that the non-weighted precision of mode C_1 is higher of that of mode C_2, while the non-weighted recall of mode C_1 is lower than that of mode C_2. Regarding the weighted measures, the situation is quite the opposite, as the weighted precision of mode C_1 is lower of that of mode C_2, while the weighted recall of mode C_1 is higher than that of mode C_2. Regarding the differences between the weighted and non-weighted measures these can be explained as follows (this is further discussed in subsequent paragraphs): According to the non-weighted measures that penalize all errors equally, there are cases where the model should react by predicting mode C_1 at a specific trajectory point, but it does not, as it instead predicts mode C_2. On the other hand, when considering the weighted measures, the model predicts mode C_1 *near* the RATPs, in points that according to the dataset are annotated as C_2. These are mostly points that succeed the actual RATPs and precede the start of the maneuver that implements the resolution action. Note that maneuvers implementing the resolution actions do not begin instantly after the ATCOs' reaction, as pilots need some time to react to the ATCOs' instruction. Points succeeding the actual RATP and preceding the start of the maneuver have features that are close to the features of the corresponding actual RATPs, so they are penalized lightly.

As shown in Figure 4.9, results for the prediction of ATCOs' resolution actions are not so impressive as those achieved on the prediction of modes. The non-weighted f1-score for the A_1 and A_2 resolution actions is 0.588 ± 0.017 and 0.656 ± 0.021, respectively, and the weighted f1-score is 0.610 ± 0.017 and 0.679 ± 0.023, respectively. The prediction of ATCOs' resolution actions are further explored in this thesis in the sections regarding the ATCOs' policy modeling problem.

Observing the weighted and non-weighted measures w.r.t. the modes of the ATCOs' reactions, it is evident that the weighted measures are higher than the non-weighted. To better understand the difference between the non-weighted and the weighted measures, Table 4.8 shows one of the trajectories with the highest difference between the weighted and non-weighted f1-score for the sector-ignorant case. "Predicted" shows the modes predicted by the VAE model, whereas "Expert" shows the modes reported in the dataset. X-axis: the sequence number of the trajectory states. Y-axis: the modes. (Blue) Dots denote the mode at each point. (Green) Solid vertical lines at x=0 show the start of the trajectory, while (red) dashed vertical lines at x=100 indicate the point with actual RATP. Figures show that the model predicts mode C_1 near the actual RATPs. Also, the model predicts mode C_2 instead of mode C_1 at points far from the actual RATPs (in most cases), and such errors are penalized lightly by the weighted measures.

Regarding the sector-related setting, Table 4.10, similarly to Table 4.9, reports the 95% confidence interval of the non-weighted and weighted versions of precision, recall and f1-score, achieved by the VAE and the encoder (Enc) for the ATCOs' modes and the resolution actions.



(a) Expert

(b) Predicted

Figure 4.8: One of the trajectories with the highest difference between the weighted and non-weighted f1-score. "Predicted" shows the modes predicted by the VAE model, whereas "Expert" shows the modes reported in the dataset. X-axis: the sequence number of the trajectory states. Y-axis: the modes. (Blue) Dots denote the mode at each point. (Green) Solid vertical lines at x=0 show the start of the trajectory, while (red) dashed vertical lines at x=100 indicate the point with actual RATP.

As far as the modes of the ATCOs are concerned, the VAE network achieves f1-score of at least 0.835 ± 0.025 (C_2 mode) on all modes, for both the non-weighted and the weighted measures, while the encoder achieves f1-score of at least 0.774 ± 0.024 (C_2 mode). VAE achieves the best results with a weighted f1-score of at least 0.945 ± 0.015 in all modes. The VAE model outperforms the encoder model on all measures, weighted or not.

Similarly, for the sector-related case, weighted measures are higher than the non-weighted. This shows that in many cases, as explained in the sector-ignorant case, the model makes false predictions that are penalized lightly by the weighted measures. As it is also shown in Table 4.10, the non-weighted precision of mode C_1 is higher of that of mode C_2, while the non-weighted recall of mode C_1 is lower than that of mode C_2. Regarding the weighted measures, the situation is quite the opposite, as the weighted precision of mode C_1 is lower of that of mode C_2, while the weighted recall of mode C_1 is higher than that of mode C_2. As explained in the sector-ignorant case, this implies that according to the non-weighted measures, there are cases where the model should assign a resolution action to a specific trajectory point by predicting mode C_1, but it does not, as in that particular point it instead predicts mode C_2. On the other hand, according to the weighted measures, the model predicts mode C_1 near the actual RATPs even for points that are annotated as C_2.

Regarding the predictions of resolution actions, in the sector-related case results are not good: The f1-score of the A_2 resolution action is 0.384 ± 0.075 for the non-weighted and 0.419 ± 0.076 for the weighted measure. As already pointed out, this is further explored when considering the ATCOs' policy modeling problem.

Table 4.11 provides evidence of statistical significance when comparing the performance of the VAE and the encoder w.r.t. the prediction of modes. Specifically this table reports the p-values computed by applying the Wilcoxon signed rank test on the average of the f1-scores achieved by the VAE and the encoder. The samples of the populations tested are the averages over the modes of the f1-scores (weighted and non-weighted) of all experiments performed using the

Table 4.10: Experimental Results of the sector-related case achieved by the VAE and the encoder (Enc). Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes and the resolution actions of ATCOs, for the non-weighted and weighted measures.

model	modes non-weighted	modes weighted	actions non-weighted	actions weighted
VAE	precision $C_0 : 1.000 \pm 0.000$ $C_1 : 0.919 \pm 0.029$ $C_2 : 0.791 \pm 0.048$ recall $C_0 : 1.000 \pm 0.000$ $C_1 : 0.835 \pm 0.045$ $C_2 : 0.893 \pm 0.036$ f1-score $C_0 : 1.000 \pm 0.000$ $C_1 : 0.873 \pm 0.025$ $C_2 : 0.835 \pm 0.025$	precision $C_0 : 1.000 \pm 0.000$ $C_1 : 0.929 \pm 0.028$ $C_2 : 0.960 \pm 0.011$ recall $C_0 : 1.000 \pm 0.000$ $C_1 : 0.965 \pm 0.012$ $C_2 : 0.919 \pm 0.033$ f1-score $C_0 : 1.000 \pm 0.000$ $C_1 : 0.945 \pm 0.015$ $C_2 : 0.940 \pm 0.014$	precision $A_0 : 0.952 \pm 0.012$ $A_1 : 0.604 \pm 0.087$ $A_2 : 0.439 \pm 0.099$ recall $A_0 : 0.981 \pm 0.007$ $A_1 : 0.566 \pm 0.070$ $A_2 : 0.362 \pm 0.076$ f1-score $A_0 : 0.966 \pm 0.005$ $A_1 : 0.569 \pm 0.039$ $A_2 : 0.384 \pm 0.075$	precision $A_0 : 0.993 \pm 0.003$ $A_1 : 0.611 \pm 0.088$ $A_2 : 0.446 \pm 0.100$ recall $A_0 : 0.983 \pm 0.006$ $A_1 : 0.661 \pm 0.068$ $A_2 : 0.428 \pm 0.093$ f1-score $A_0 : 0.986 \pm 0.004$ $A_1 : 0.620 \pm 0.036$ $A_2 : 0.419 \pm 0.076$
Enc	precision $C_0 : 1.000 \pm 0.000$ $C_1 : 0.863 \pm 0.020$ $C_2 : 0.740 \pm 0.037$ recall $C_0 : 1.000 \pm 0.000$ $C_1 : 0.805 \pm 0.031$ $C_2 : 0.809 \pm 0.024$ f1-score $C_0 : 1.000 \pm 0.000$ $C_1 : 0.833 \pm 0.022$ $C_2 : 0.774 \pm 0.024$	precision $C_0 : 1.000 \pm 0.000$ $C_1 : 0.874 \pm 0.021$ $C_2 : 0.950 \pm 0.009$ recall $C_0 : 1.000 \pm 0.000$ $C_1 : 0.955 \pm 0.011$ $C_2 : 0.857 \pm 0.022$ f1-score $C_0 : 1.000 \pm 0.000$ $C_1 : 0.913 \pm 0.013$ $C_2 : 0.902 \pm 0.011$	-	-

VAE and the encoder. In all settings the p-value is significantly lower than 0.05 showing that the difference in the performance between the VAE and the encoder is statistically significant.

Table 4.11: p-values computed by applying the Wilcoxon signed rank test on the unweighted average of the f1-scores (weighted and non-weighted) achieved by the VAE and the encoder when predicting the modes of the test set. The samples of the populations tested are the unweighted averages over the modes of the f1-scores (weighted and non-weighted) of the 10 experiments achieved by the VAE and the encoder.

setting	modes non-weighted	modes weighted
sector-ignorant	0.0050	0.0049
sector-related	0.0117	0.0051

Comparing the performance of the models between the different settings, it is observed that models perform better for the sector-ignorant setting, rather than for the sector-related setting. This could be due to the following: First, due to the difference of the size of the dataset between the two settings. Indeed, the dataset for the sector-related case is approximately $1/3$ the size of the dataset for the sector-ignorant case. Second, flights from adjacent sectors may contribute

to conflicts in the current sector (AoR) or in the downstream sector of a flight. However, these flights are not considered in the sector-related case, therefore the corresponding conflicts are not detected.

As CD&R in the ATM domain is safety critical, Table 4.12 reports the cases where the models do not react in critical situations, i.e., in cases where a conflict is detected and the ATCO reacts with a resolution action. The "setting" column reports the experimental setting, and columns "VAE #cases" and "Encoder #cases" report the number of cases within 95% confidence interval for the VAE and the encoder, respectively.

For the sector-ignorant setting the average number of cases for VAE is 6, which is 3.79% of the resolution actions in the test set. The encoder on the other hand reports less cases where it did not react at all in a critical situation, with an average number of 3.2 cases, which corresponds to 2.02% of the resolution actions in the test set. For the sector-related setting the number of such cases for VAE is 5.2 corresponding to 7.6% of the resolution actions in the test set. For the encoder the number of such cases is smaller, with an average value of 2.6 cases, corresponding to 3.78% of the resolution actions in the test set.

Considering the results reported in Tables 4.9 and 4.10, in conjunction to the number of cases where the models did not react at all to critical situations reported in Table 4.12, it is true that the predictions of VAE, when compared with those of the encoder, fit better the modes of the test set. Specifically, the window of points in which the VAE model predicts mode C_1 is more close to that of the expert, compared to the encoder predictions. The encoder on the other hand has less cases where it did not react at all to critical situations, compared to VAE. This, as it is discussed subsequently, can be explained by the probabilities that each model assigns to modes in each case.

Table 4.12: Number of cases within the 95% confidence interval where the models do not predict a resolution action to any of the annotated or actual RATPs or any point in the time window of 70s near the actual or annotated RATPs.

Setting	**VAE #cases**	**Encoder #cases**
sector-ignorant	6.000 ± 2.146	3.200 ± 1.746
sector-related	5.200 ± 1.311	2.600 ± 1.022

Table 4.13 depicts the probability assigned to each mode by each model (VAE or encoder (Enc)), at every point in 10 trajectories. Column "Setting" denotes the sector-ignorant or sector-related experimental setting. The x-axis shows the sequence number of trajectory states and the y-axis the probability of each mode. Solid (green) vertical lines denote the start of each of the 10 trajectories, while (red) dashed lines denote the actual RATPs. Numbers over the solid (green) lines denote the sequence number of each trajectory. As it is shown, for the sector-ignorant case, VAE provides more "confident" predictions compared to the encoder, assigning higher/lower probabilities to modes. For the sector-related case both models provide quite confident predictions, and at most points the probabilities assigned to each mode are either high or low.

The differences observed in the sector-ignorant case regarding the magnitude of the probabilities assigned to the different modes by each model, could explain a) why the predictions of the VAE are more consistent with the actual data, compared to the predictions made by the encoder, and b) why the encoder has less cases where it did not react at all to critical situations, compared to VAE. VAE assigns high/low probabilities to modes, making more confident predictions, and predicting more consistently in a window of points, without fluctuating between

predicted modes from point to point. The encoder, on the other hand, assigns more mid-range probabilities to modes and could alternate more easily between modes predicted on consecutive points. This implies that VAE, when predicting a resolution action, will be more "committed" to the prediction of the resolution action for a window of points, thus fitting the dataset around the RATPs better than the encoder. On the other hand, the encoder, being less "confident" and more "fluid" in its predictions, can predict mode C_1, even in points in windows where it mostly predicts mode C_2. This can result to less cases where it does not react at all to critical situations.

4.8.2 Modeling ATCO's policy

This section presents the experimental evaluation of the methods used for predicting the ATCOs' reactions. This section is structured as follows: first subsection 4.8.2.1 presents the experimental setting, then subsection 4.8.2.2 presents the datasets and the pre-processing, and subsection 4.8.2.3 reports the experimental results.

4.8.2.1 Experimental Setting

To evaluate the proposed methods, this study simulates the flight-centric concept using the sector ignorant setting presented in section 4.8.1.1. The sector related setting is not considered here as it resulted to worst model performance compared to the sector ignorant case. This as discussed in section 4.8.1.4 could be a) due to the smaller dataset size in the sector related case or b) because flights from adjacent sectors are not considered although they may contribute to conflicts in the current sector (AoR) or in the downstream sector of a flight.

According to the sector ignorant setting, the airspace considered is restricted to a rectangular area covering the Iberian Peninsula, as Figure 4.6 shows. Cells of size 0.5×0.5 degrees (longitude and latitude) segregate this area, creating an index of the positions of trajectories in each cell at each time point. This allows fast access to the trajectories of each cell at each time point, making the identification of neighbor trajectories computationally more efficient. For each point of a focal trajectory, only the neighbor trajectories within a distance threshold D_{th} of five cells in longitude (approx. 231 km) and latitude (approx. 308km) are considered. This has as the effect that the area specified by D_{th} in SA "follows" the focal trajectory points, i.e., the movement of the ownship.

Figure 4.6 shows the SA area considered. The focal trajectory of the ownship is indicated by red (dark gray in grayscale) and other trajectories are the neighbor trajectories in $Neigh(T_f, SA, t)$. The ownship position (i.e., the focal trajectory point) at time t is shown in white (middle point). The area defined by D_{th} with respect to the ownship's position is depicted by the red rectangle in SA. The yellow dot in the upper part of the grid is the fixpoint.

To evaluate the effectiveness of the methods in predicting the type of the ATCOs' resolution actions, five-fold cross-validation is performed, splitting the trajectories into 20% test trajectories and 80% training trajectories. Measures used comprise the mean precision, mean recall, and mean f1-score for the resolution action types, A_1, A_2, and A_3, and also the mean MCC across all folds.

Considering the true and predicted classes as two random variables, the MCC is the correlation coefficient between these random variables. For binary classification problems, the MCC is calculated as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.8}$$

where, TP, FP, FN, and TN, denote the number of true positives, false positives, false negatives, and true negatives, respectively. Therefore, MCC equal to 1 indicates a perfect prediction and 0 (zero) a random prediction.

In the multiclass case, MCC is calculated as follows:

$$MCC = \frac{c^{mcc} \times s^{mcc} - \sum_k^K p_k \times t_k}{\sqrt{((s^{mcc})^2 - \sum_k^K p_k^2) \times ((s^{mcc})^2 - \sum_k^K t_k^2)}} \tag{4.9}$$

with C^{mcc} being the confusion matrix for K classes, $t_k = \sum_i^k C_{ik}^{mcc}$ the number of times class k truly occurred, $p_k = \sum_i^K C_{ki}^{mcc}$ the number of times class k was predicted, $c^{mcc} = \sum_k^K C_{kk}^{mcc}$ the total number of samples correctly predicted, and $s^{mcc} = \sum_i^K \sum_j^K C_{ij}^{mcc}$ the total number of samples.

4.8.2.2 Data sets and preprocessing

To evaluate methods for modeling the ATCOs' policy this study exploits data from the Spanish airspace, considering flights over Spain, without sacrificing the generality of the methods introduced. The data sources comprise:

- Surveillance data: operational quality data with actual flights (raw) trajectories (Spanish ATC Platform SACTA).
- ATCO events: provides actions taken by the ATCOs in order to ensure safety of flights (provided by ATON).

Following a data-driven approach and exploiting data with flown trajectories, ATCO events must be associated with potential conflicts that have been assessed to exist, either at the point of the ATCO event or in a time window of $window\_duration$ seconds prior to the ATCO event. However, there are cases where there is an ATCO resolution action for a trajectory but no potential conflicts can be revealed. These cases are filtered out. The trajectory point (if any) in the specified time window at which a potential conflict is revealed and is temporally closest to the point of the ATCO event is considered to be the actual resolution action point (mentioned as RATP).

The data include aircraft trajectories flown in 2017 between five different origin–destination pairs: Malaga (LEMG)–Gatwick (EGKK), Malaga (LEMG)–Amsterdam (EHAM), Lisbon (LPPT)–Paris (LFPO), Zurich (LSZH)–Lisbon (LPPT), and Geneva (LSGG)—Lisbon (LPPT). In this study, the en route phase of flights is considered, and thus resolution actions and trajectory points corresponding to the climb and descent phases of the flights are filtered out. In addition, only trajectories that have at least one ATCO resolution action of the considered types and an associated RATP are considered. This results in a total of 793 resolution actions associated with 634 trajectories, consisting of 326 "speed change", 374 "direct to", and 79 "radar vectoring" actions.

It must be noted that although the available ATCO events dataset covers the Spanish airspace, the proposed method is generic, and can be applied in any airspace.

4.8.2.3 Experimental results

This section presents the experimental results achieved by the AI/ML methods considered, in a comparative way.

Table 4.14 reports the experimental results achieved by the NN classifier with an attention mechanism (NN+att) and without attention (NN), in addition to the RF and the GTB algorithms. The columns report the 95% confidence interval of precision, recall, f1-score, and MCC with regard to the resolution action types of ATCOs.

As shown in Table 4.14, the RF method achieves the best results in terms of the mean MCC and f1-score achieved on the test set, with a mean MCC equal to 0.51 and a mean f1-score equal to 0.73 for resolution action type A_1, 0.76 for resolution type A_2, and 0.38 for resolution action A_3. The second-best results are provided by the GTB algorithm, achieving a mean MCC value of 0.48 and a mean f1-score 0.69 for resolution action type A_1, 0.74 for resolution type A_2, and 0.48 for resolution action type A_3.

The NN algorithms reported a reduced mean MCC and f1-score compared to RF and GTB. Among the variations tested, the NN+att achieves a mean MCC value of 0.44 and mean f1-score of 0.66, 0.72, and 0.41 for resolution action types A_1, A_2, and A_3, respectively.

The effect of the attention module on the accuracy of the predictions, compared against the variant without attention (NN) is positive: the mean values of the MCC and f1-score increase and the confidence interval becomes narrower. This implies better and more stable performance (reduced standard deviation for independent experiments) among the different folds. This improvement implies that the modeling of interactions between the ownship and its neighbors using a convolution layer results in more useful representations of states.

Considering the capacity of the models, the RF and GTB methods achieve a strong positive correlation between true and predicted A_1, A_2, and A_3 resolution action types on the training set, with MCC values ranging from 1 to 0.96, respectively. The f1-scores are also high for RF and GTB, with values in the interval [0.94, 1]. The MCC and f1-scores of NN and NN+att computed for the training set are not so high, since the training stops when the early stopping mechanism detects overfitting.

Table 4.13: Scatterplots depicting the probability assigned to each mode by each model (VAE or encoder (Enc)), at every point in 10 trajectories. Column "Setting" denotes the sector-ignorant or sector-related experimental setting. The x-axis shows the sequence number of trajectory states and the y-axis the probability of each mode. Solid (green) vertical lines denote the start of each of the 10 trajectories, while (red) dashed lines denote the actual RATPs. Numbers over the solid (green) lines denote the sequence number of each trajectory.

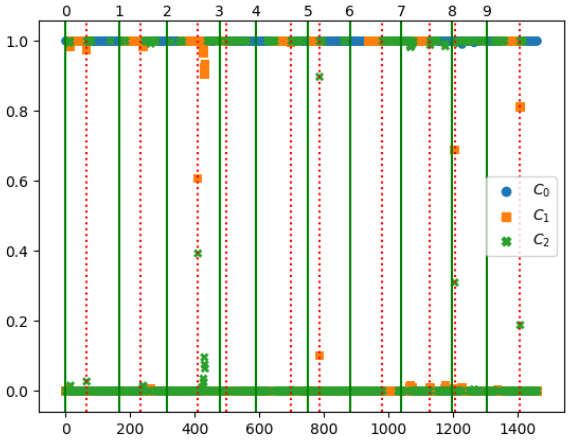
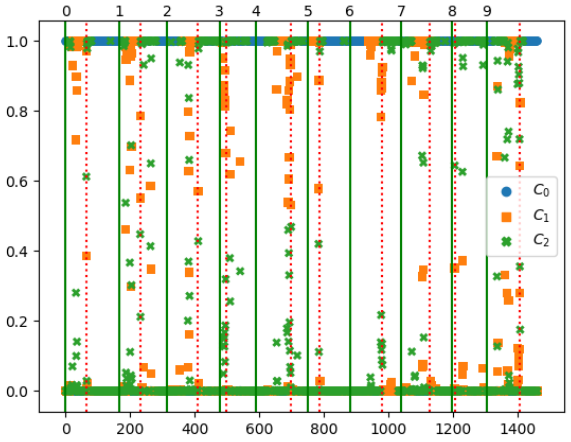
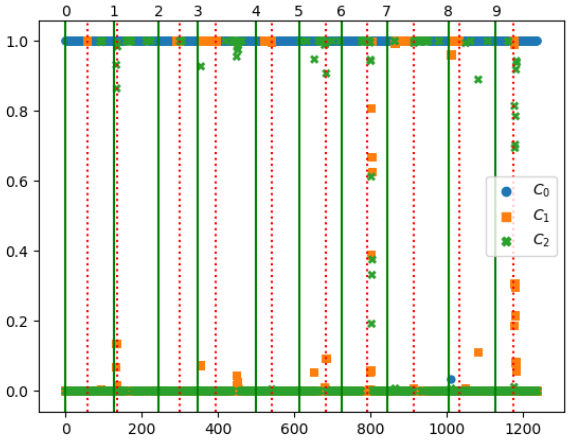
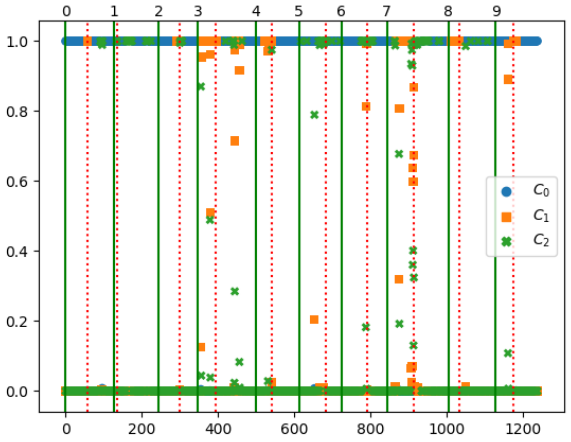
Setting	**VAE**	**Enc**
sector-ignorant		
sector-related		

Table 4.14: Experimental results achieved by the NN classifier with an attention mechanism (NN+att) and without attention (NN), in addition to the Random Forest (RF) and the GTB algorithms. Columns report the 95% confidence interval of precision, recall, f1-score, and MCC with regard to the resolution action types of ATCOs.

Method	**Dataset**	**Precision**			**Recall**			**f1-Score**			**MCC**
		A_1	A_2	A_3	A_1	A_2	A_3	A_1	A_2	A_3	
NN	train	0.81 ± 0.05	0.86 ± 0.04	0.83 ± 0.04	0.86 ± 0.04	0.87 ± 0.03	0.58 ± 0.11	0.83 ± 0.04	0.87 ± 0.04	0.68 ± 0.08	0.72 ± 0.07
	test	0.59 ± 0.06	0.70 ± 0.09	0.57 ± 0.11	0.71 ± 0.12	0.63 ± 0.10	0.32 ± 0.11	0.64 ± 0.06	0.66 ± 0.07	0.40 ± 0.09	0.37 ± 0.09
NN+att	train	0.75 ± 0.02	0.78 ± 0.10	0.83 ± 0.08	0.74 ± 0.18	0.85 ± 0.04	0.44 ± 0.03	0.74 ± 0.11	0.81 ± 0.05	0.58 ± 0.04	0.59 ± 0.10
	test	0.61 ± 0.05	0.77 ± 0.06	0.53 ± 0.12	0.73 ± 0.11	0.70 ± 0.13	0.33 ± 0.09	0.66 ± 0.03	0.72 ± 0.06	0.41 ± 0.10	0.44 ± 0.05
RF	train	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	test	0.68 ± 0.04	0.76 ± 0.08	0.92 ± 0.13	0.78 ± 0.07	0.77 ± 0.00	0.26 ± 0.12	0.73 ± 0.05	0.76 ± 0.04	0.38 ± 0.15	0.51 ± 0.07
GTB	train	0.96 ± 0.01	0.98 ± 0.01	1.00 ± 0.00	0.99 ± 0.01	0.98 ± 0.01	0.89 ± 0.02	0.97 ± 0.01	0.98 ± 0.01	0.94 ± 0.01	0.96 ± 0.01
	test	0.67 ± 0.06	0.74 ± 0.06	0.68 ± 0.14	0.72 ± 0.05	0.75 ± 0.02	0.38 ± 0.06	0.69 ± 0.04	0.74 ± 0.03	0.48 ± 0.06	0.48 ± 0.04

Given the presented results, the following observations are made:

- All methods have difficulty in predicting the A_3 resolution action type accurately.
- The number of samples might be small, especially for training NN models.
- As discussed in Section 4.7, this study deals with historical data that do not demonstrate the conflicts occurred; thus, it is likely that the labels (in this case, historical ATCO resolution action types) of revealed conflicts are noisy.

Regarding the first issue, considering that resolution action type A_3 is the minority class, the application of different techniques that deal with class imbalance to improve the results of the best method were studied. Specifically, sample weights, i.e., RF with class weights, and resampling, i.e., balanced RF [6], were used. Such approaches achieved better results for the minority class but reduced the accuracy of the predictions for the other classes, resulting in a reduced overall accuracy. Specifically, balanced RF achieved a mean MCC score of 0.44 and f1-scores of 0.65, 0.72, and 0.42 for resolution action types A_1, A_2, and A_3, respectively.

Considering the second issue, the training data were augmented by using the trajectory points at which a conflict is detected, in a time window of 250 s before the RATP, as potential conflicting states. These points were labeled with the action type of the corresponding RATP. This is on par with the uncertainty of ATCOs regarding the time to issue a resolution action. This approach did not result in better results, either for the NN, or for the RF classifiers. This suggests that this type of data augmentation is not beneficial. This could be due either to mislabeled samples inserted into the training set, increasing the label noise, or because this type of augmentation does not effectively cover the feature space.

Noisy labels, as already discussed in Section 4.7, could be introduced by (i) the data itself, and by (ii) the conflict detection methodology. To these aspects (iii) the data augmentation process should be added.

As discussed in Section 4.7, to address label noise this study opted for SEAL. Furthermore, using SEAL on the augmented data improved the accuracy of the predictions. As shown in Table 4.15, NN with the attention module, SEAL, and data augmentation (NN+att+SEAL+augm) achieved an MCC score of 0.46, which is a +0.02 improvement over NN with the attention module (NN+att). The f1-score for the A_1, A_2, and A_3 resolution action types was 0.68, 0.73, and 0.47, respectively. However, this improvement was small and did not manage to outperform RF nor GTB. An important reason for the results achieved by the methods robust to label noise in this case, is that such methods are usually validated in noise-free test sets. In this case, the test set contained samples that were as noisy as the training dataset, and, thus, although the method could be robust to label noise, this effect is not evident in a noisy test set.

Finally, NN+att with the active-passive loss functions presented in [52] and with data augmentation (NN+att+AP loss+augm) were also tested. This method did not prove to be robust in instance-dependent noise and did not achieve better results than NN+att. As commented above, noisy samples in the test set could "hide" the effectiveness of the AP loss. As reported in Table 4.15, the best MCC score achieved was 0.40 and the f1-score for the A_1, A_2, and A_3 resolution action types was 0.65, 0.71, and 0.35, respectively. These results were achieved using the normalized cross entropy loss combined with the reverse cross entropy loss and setting the a and b hyperparameters to $a = 10$ and $b = 0.1$.

[6] https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedRandomForestClassifier.html

Table 4.15: Experimental results achieved by balanced RF, NN with attention SEAL and data augmentation (NN+att+SEAL+augm), and NN with attention active-passive loss and data augmentation (NN+att+AP loss+augm). Columns report the 95% confidence interval of precision, recall, f1-score, and MCC with regard to the resolution action types of ATCOs.

Method	**Dataset**	**Precision**			**Recall**			**f1-Score**			**MCC**
		A_1	A_2	A_3	A_1	A_2	A_3	A_1	A_2	A_3	
Balanced RF	train	0.83 ± 0.01	0.91 ± 0.02	0.57 ± 0.02	0.81 ± 0.02	0.79 ± 0.02	1.00 ± 0.00	0.82 ± 0.02	0.84 ± 0.02	0.72 ± 0.02	0.70 ± 0.03
	test	0.68 ± 0.07	0.75 ± 0.07	0.34 ± 0.05	0.63 ± 0.05	0.70 ± 0.05	0.56 ± 0.11	0.65 ± 0.06	0.72 ± 0.06	0.42 ± 0.06	0.44 ± 0.08
NN+att+AP loss+augm	train	0.61 ± 0.01	0.72 ± 0.01	0.65 ± 0.23	0.73 ± 0.04	0.70 ± 0.04	0.20 ± 0.17	0.67 ± 0.01	0.71 ± 0.02	0.28 ± 0.19	0.41 ± 0.02
	test	0.62 ± 0.03	0.70 ± 0.01	0.67 ± 0.26	0.68 ± 0.04	0.72 ± 0.05	0.25 ± 0.06	0.65 ± 0.03	0.71 ± 0.02	0.35 ± 0.03	0.40 ± 0.02
NN+att+SEAL+augm	train	0.99 ± 0.01	0.99 ± 0.00	0.99 ± 0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.01	0.99 ± 0.01
	test	0.66 ± 0.03	0.72 ± 0.04	0.71 ± 0.24	0.71 ± 0.07	0.74 ± 0.02	0.36 ± 0.04	0.68 ± 0.04	0.73 ± 0.03	0.47 ± 0.06	0.46 ± 0.06

Experiments were ran on an AMD Ryzen 9 3900X 12-Core Processor and NN-based models also utilized a GeForce RTX 2080 Ti Graphics Processing Unit (GPU). Regarding the computational efficiency of the methods, predictions of samples were instant (provided in milliseconds) for all methods. Considering the training time, methods based on RF and GTB completed training in less than one minute. NN-based methods needed more training time, but were still time-efficient, as one experiment completed training in less than an hour. In the case of SEAL, this must be multiplied by the number of SEAL iterations. When comparing NN+att to NN, NN+att is more computationally expensive than NN, as it has more learnable parameters than NN.

4.9 Conclusions

This section concludes this chapter providing remarks regarding data-driven AI/ML methods for the detection and resolution of conflicts. Specifically, this study decomposes the problem of modeling the ATCOs' behavior in CD&R in a data-driven way into the following subproblems: a) predicting the ATCOs' reactions and b) modeling the ATCOs' policy. Subsection 4.9.1 provides concluding remarks for the problem of predicting the ATCOs' reactions, while subsection 4.9.2 provides concluding remarks for the problem of modeling the ATCOs' policy.

4.9.1 ATCOs' reactions

This study formulates the problem of CD&R as a data-driven problem, aiming to learn ATCOs' reactions as a hierarchical task involving high-level reactions, representing the mode of the ATCOs' behavior, and low-level reactions representing ATCOs' conflict resolution actions. The proposed approach uses a VAE in the context of a deep learning methodology to imitate ATCOs' behaviour in a hierarchical manner. The proposed method is evaluated using real world data in two different experimental settings: the sector-ignorant and the sector-related. To train the proposed model, this study proposes a data-driven method for simulating the evolution of trajectories, incorporating uncertainty and revealing the conflicts that ATCOs may have assessed before reacting. Also as discussed in section 4.8.1.2 this study uses data augmentation and subsampling to tackle the dataset's imbalance w.r.t. the ATCOs' modes. Subsampling modes C_0 and C_2, keeping one trajectory point with any of these modes every $\Delta step$ trajectory points results to a time step of $\Delta step * 5$s for these points. On the other hand points of mode C_1 have a time step of 5 s. This difference in the time step between trajectory points will be further discussed in chapter 5, when considering how models predicting trajectories, ATCO reactions and the ATCOs' policy can be combined into a unified method towards planning conflicts-free trajectories, as to develop such method a constant time step between trajectory points is needed. To evaluate the proposed method, as well as any other data-driven methods that aim to solve the ATCOs' reaction prediction problem, weighted measures of precision, recall and f1 have been proposed. These measures have been used to compare the VAE model against a basic model comprising only an encoder. This comparison delves into the difference that the backwards propagation of the VAE's decoder errors make to the performance of VAE.

According to the experimental evaluation, both models (VAE and encoder) accurately predict the mode of the ATCOs' behavior either in the sector-ignorant or in the sector-related setting. The VAE achieves consistently better results than the encoder w.r.t. the weighted and non-weighted measures in both settings. The encoder on the other hand seems to perform better w.r.t. the number of cases where the models do not react at all to critical situations. This said, it must be pointed out that the number of such cases for both models is very small, however such cases will be further explored in future work.

Regarding the predictions of resolution actions, results achieved by VAE are not so impressive as those achieved on the prediction of modes, and this was further explored in this study in sections considering the modeling of the ATCOs' policy.

Finally, regarding the two experimental settings, models perform better at the sector-ignorant case. This could be due to the difference of the size of the training datasets in the two settings, as well as to the fact that flights from adjacent sectors may contribute to conflicts in the current or downstream sector: These cases of conflicts are not detected in the sector-related case.

Summarizing, the findings of this research are as follows:

– The proposed methodology accurately predicts the modes of the ATCOs' behavior, predicting whether and when the ATCO reacts in the presence of conflicts.

– According to the experimental evaluation, the VAE model assigns high/low probabilities to modes, predicting a mode without major/frequent fluctuations in predictions. The encoder on the other hand assigns more mid-range probabilities to modes, and fluctuates frequently between modes in consecutive trajectory points. As a result, the predictions of the VAE are more consistent with the expert reactions compared to the predictions made by the encoder, although the encoder has less cases where it did not react at all to critical situations, compared to VAE. The accuracy of predictions implies that errors propagating backwards from the decoder to the encoder play indeed an important role to the quality of the VAE model learned.

– Predictions of the low-level ATCOs' conflict resolution actions performed by the VAE model are not as accurate as the predictions of the ATCOs' modes of reaction.

– Regarding the different experimental settings explored, models perform better at the sector-ignorant case compared to the sector-related. This is something to be further explored in future work.

4.9.2 ATCOs' policy

Regarding the modeling of ATCOs policy, this study addresses challenges that result from inherent imperfections of historical data sets recording trajectories and associated ATCOs events, and makes the following contributions:

1. It specifies a formulation of learning the ATCOs policy problem as a classification task;
2. It studies enhanced AI/ML methods to learn models of ATCOs policy from real-world historical data sets;
3. It evaluates the proposed AI/ML methods using real-world data.

The methodology followed towards addressing data limitations, and the training of AI/ML models, entails exploiting ATCOs' expert knowledge regarding (a) the assessment of traffic and potential conflicts the ATCOs might observed, and associating these conflicts with the recorded resolution actions; and (b) how conflicts are resolved by ATCOs, as this is revealed by ATCOs events.

Results show that classification methods, such as RF, GTB and NNs achieve good accuracy on predicting ATCOs actions given specific conflicts, but they have limitations which are mostly due to the imperfections of historical data sets exploited.

Indeed, resolution actions predicted by models learned using a data driven approach, as done

in this study, will be in the best case as good as the actions included in the historical data sets. If the data set includes ATCOs actions that perform poorly by not effectively solving the detected conflicts, the models will repeat such actions under nearly the same circumstances: Thus it is important having data sets containing effective ATCOs resolution actions, according to specific objectives. Also, as discussed, the ATCOs observations that triggered the ATCOs resolution actions are essential for the learning process. Historical data sets used in this study do not include these ATCOs observations. To address this issue the aircraft's position is projected into the future in order to reveal the potential conflicts and the corresponding ATCO observations that triggered the ATCO resolution action. This is challenging and introduced noise in the learning process. Deviating from the actual ATCO observations can have a negative effect on the models' performance.

Other methods such as RL algorithms have the ability to explore the state space in a trial and error fashion and apply optimization in terms of specific factors, such as conflicts resolved, nautical miles added to the trajectory due to resolution actions, fuel consumption etc. Such techniques in many cases provide more effective and efficient actions w.r.t. the optimization objective when compared to human decisions. However, to increase effectiveness and trustworthiness of automated decision making agents [81], especially in safety critical domains as the ATC, actions proposed should be similar to actions taken by human experts.

Further research involves investigating the combination of supervised learning methods with RL techniques, in order to provide resolution actions considering the ATCOs preferences, while also optimizing specific objectives with respect to the efficiency of the resolution actions.

Finally, future research involves addressing the ATCOs policy learning problem as a multi-stage IL task, considering the evolution of conflicts: This is rather challenging, and data sets with conflicting situations associated with ATCOs events are necessary.

Chapter 5

Towards Planning Conflicts-Free Trajectories

This chapter aims to answer if and to what extent methods for trajectory prediction and CD&R suffice for creating a method for planning conflicts-free trajectories. To do so, this study presents a straightforward way of combining the models for trajectory prediction and CD&R presented in the previous chapters into a unified approach for planning conflicts-free trajectories. It proceeds to evaluate the resulting method using real world data.

This study reveals interesting pitfalls and challenges arising when combining the models. These are described in the sections that follow, together with proposals on how they can be mitigated. Specifically this study develops and tests a purely data driven approach, that exploits the trained independent models for trajectory prediction and for modeling the ATCOs' behavior and combines them in a sequential manner, presenting challenges and problems to be addressed in the future regarding needed data and ways to combine ATCO models towards conflicts-free trajectories.

This chapter is structured as follows: Section 5.1 specifies the problem of conflicts-free trajectory planning. Section 5.2 presents an overview of the proposed framework for conflicts-free trajectory planning. Section 5.3 presents the features perceived by each model and also the methods used. Section 5.4 evaluates the overall method using real world data providing insights and future steps towards conflicts-free trajectory planning. Finally section 5.5 concludes this chapter.

5.1 Problem specification

Given a set $\mathbf{T}_E$ of historical trajectories and a set $\mathbf{RA}_E$ of historical ATCO conflict resolution actions (ATCO events) associated to trajectories in $\mathbf{T}_E$, the problem of *conflicts-free trajectory planning* is about predicting trajectories adhering to the preferences of the airspace users as these are revealed by $\mathbf{T}_E$, whose conflicts are being resolved according to the ATCOs' preferences demonstrated in $\mathbf{RA}_E$ in conjunction to $\mathbf{T}_E$. Preferences involved include the airspace user's intent, i.e., preferred route, and also ATCOs' preferences concerning "whether", "when", and "how" the ATCOs will react to resolve detected conflicts.

Towards this goal this study exploits (a) models for predicting flight trajectories, given a set of historic loss-free trajectories, without considering conflicts, (b) models of the ATCO behavior

consisting of (b.i) models for predicting ATCO reactions focusing on "whether" and "when" the ATCO decides a resolution action and (b.ii) models of the ATCOs' policy predicting the type of the resolution action the ATCO would prescribe.

Models for predicting flight trajectories without considering conflicts aim at predicting how trajectories would evolve according to historical trajectories in $\mathbf{T}_E$. Trajectories in $\mathbf{T}_E$ are shaped by different stakeholders and thus learning from historical trajectories without considering further information about the stakeholders' actions results to models incorporating preferences, strategies, practices etc. of the different stakeholders in an aggregated way.

This study combines models for predicting flight trajectories without considering conflicts, with models of the ATCO behavior, in order to predict flight trajectories from the same distribution as the ones in $\mathbf{T}_E$, while resolving conflicts as the ATCOs' would, according to the ATCOs' preferences demonstrated in $\mathbf{RA}_E$ in association with $\mathbf{T}_E$. To achieve this, having models predicting flight trajectories without considering conflicts and models of the ATCO behavior, it is also necessary to model, how flight trajectories evolve, when a resolution action is decided and the aircraft performs a specific maneuver in order to resolve the conflict.

Having predicted the time point t_A at which the ATCO will issue a resolution action, this study considers the problem of *predicting the evolution of the conflict resolution maneuver*, as predicting the trajectory evolution starting from the time point t_A, that a resolution action RA is issued, and until the time point t_{EoM} at which the maneuver ends.

Solving the problem of predicting the evolution of the conflict resolution maneuver requires having data about the resolution action RA and also about the time point t_{EoM} at which the maneuver ends. As already discussed the ATCO events dataset includes only the type of the resolution action and not the resolution action in its full detail. Also, there is no indication of the end of the maneuver in the available data and precisely determining the t_{EoM} at which a maneuver ends using historical trajectories and ATCO resolution actions is a complex task. This study fills this gap by a) learning models that predict the maneuver evolution based on the type of the resolution action RA, b) considering t_{EoM} to be the point at which the next resolution action is issued, if any, or else the time point at which the trajectory ends.

5.2 A sequential framework for planning conflicts-free trajectories

This section presents a straightforward way of combining models trained for predicting ATCO reactions, models of ATCO policy, and trajectory models learned from imitation learning techniques without considering conflicts, into an integrated method for predicting conflicts-free trajectories. The presented method trains the different models independently and combines them in a sequential Trajectory Prediction Pipeline (TPP). The aim here is to highlight possible limitations and challenges arising when combining the pre-trained models in a sequential way and ways to overcome them. This paves the way towards a method for conflicts-free trajectory planning that provides accurate prediction of trajectories while effectively solving conflicts.

This framework exploits the following models:

- Trajectory Prediction Model without Considering Conflicts (TPMwoCC): This model solves the trajectory prediction problem without considering conflicts,as it is introduced in section 3.2.1. Given features describing the position of the aircraft, this model predicts the evolution of the trajectory in space.

- ATCOs' behavior models:
  - ATCO Reaction Predictor (ARP): This is an ATCO reaction prediction model, as those already introduced in section 4.6, predicting *whether* and *when* the ATCO decides a resolution action.
  - Resolution Action Type Predictor (RATPr): This is an ATCO policy prediction model, as those already introduced in section 4.7, predicting *how* the ATCO will react, deciding the type of the resolution action the ATCO will prescribe.
- Models predicting the evolution of the conflict resolution maneuver (CRMP): Such models predict how the trajectory will evolve, when executing a conflict resolution maneuver, given features describing the position of the aircraft and its neighboring aircraft.

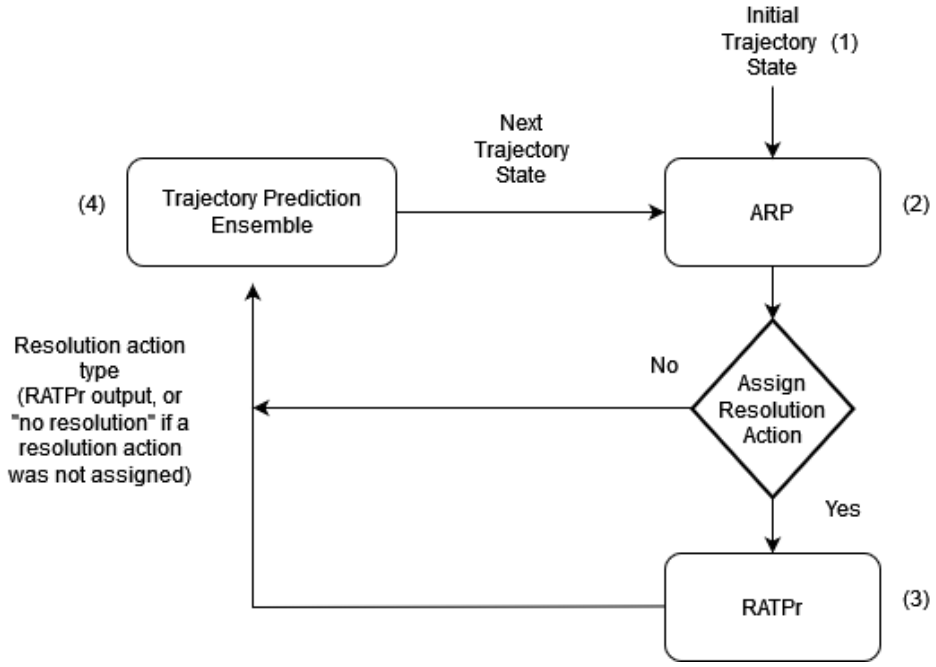


Figure 5.1: Outline of the models' combination towards providing conflicts-free trajectories. Given an initial historical state (1), the ARP predicts whether a resolution action will be applied (2). If the ARP predicts the assignment of a resolution action, then RATPr decides the type of the resolution action (3). Finally, the trajectory prediction ensemble is executed (4) controlling the ownship's movement

Given these models, the conflicts-free trajectory planning problem can be specified as a sequential decision making problem as follows: Given an initial historical trajectory point, at each time step t, first it must be decided whether a resolution action will be applied. If a resolution action is issued, the type of the resolution action to be applied must be decided and the corresponding maneuver should be applied in order to transition to the next trajectory point. In case a resolution action is not issued, this implies that there is no need to interfere to the trajectory's evolution and thus the previous "mode" of trajectory's evolution should be continued, i.e. if a maneuver is currently executed, then it should continue its execution, else if no maneuver is executed, there is no need to apply one.

Figure 5.1 shows how the different models are combined into a unified framework. Starting from an initial historical state (1) the ARP predicts whether a resolution action will be applied at the current time point (2). If the ARP does not predict the assignment of a resolution action

the trajectory prediction ensemble is executed (4) with input “no resolution action”. If the ARP predicts the assignment of a resolution action then the RATPr decides the type of the resolution action (3) and the trajectory prediction ensemble is executed (4), taking as input the type of the resolution action decided by the RATPr. Finally, the trajectory prediction ensemble controls the ownship’s movement, predicting the next trajectory state. The trajectory prediction ensemble, shown in detail in figure 5.2, decides the change in the bearing and horizontal and vertical speeds of the aircraft based on the current state and the type of the resolution action prescribed.

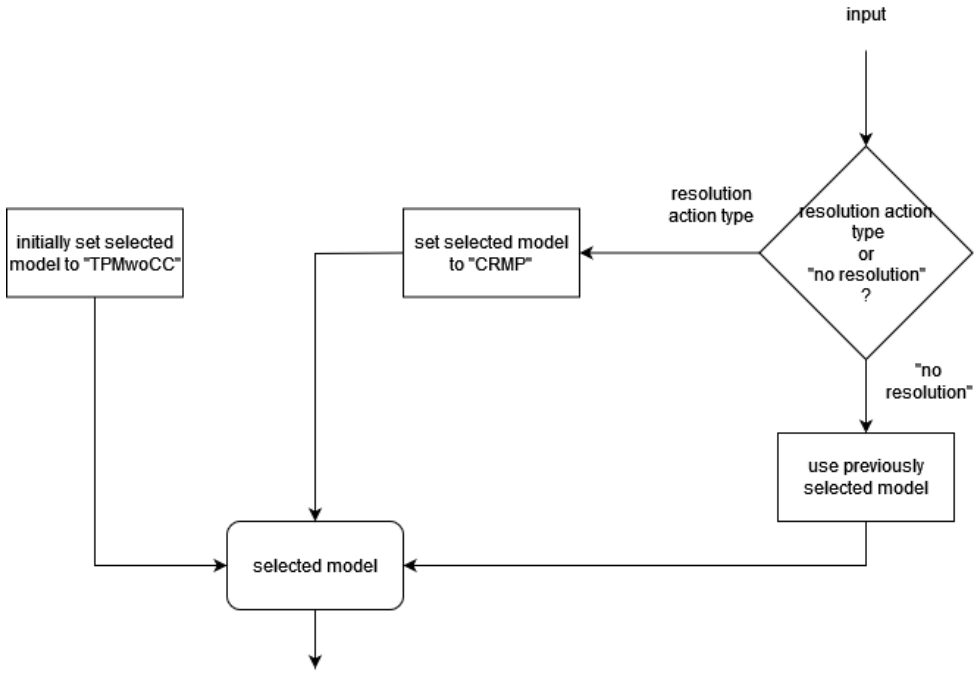


Figure 5.2: The trajectory prediction ensemble, exploiting different trajectory prediction models to predict how the ownship’s trajectory will evolve. Initially the model set for use is the TPMwoCC. If a resolution action is assigned the selected model changes to CRMP. If no resolution action should be applied the previously selected model is used.

Figure 5.2 shows how the trajectory prediction ensemble selects which trajectory prediction model to use, according to the prescribed resolution action type. Initially the model set for use is the TPMwoCC. If a resolution action is assigned the selected model changes to CRMP. If no resolution action should be applied the model used at the previous iteration of the method is used.

To implement the presented framework the following requirements must be satisfied:

- R1: Models used in this method should be agnostic of the OD airport pair, so as the overall method can be applied to any airspace and OD pair.
- R2: The ARP must make predictions using a constant time step in order to monitor consistently the trajectory predicted by the Trajectory Prediction Ensemble.

  With regard to the second point, it must be recalled that in Chapter 4 the prior distribution of ATCO reactions’ modes was highly imbalanced. To balance the distribution of ATCO’s modes, trajectory points of modes C_0 and C_2 were sub-sampled, keeping one trajectory point every $\Delta_{step} = 6$ points, resulting to a time step of 30 seconds. On the other hand, points of mode C_1 were not sub-sampled since they belong to the minority class. This

means that the time step between trajectory points of mode C_1 was kept at 5 seconds. Thus using a constant time step for all trajectory points, regardless their mode, corresponds to changing the prior of the ATCO's modes. This prior shift is critical on the performance of ML models, when it implies testing the model on a different distribution than the one it was trained on.

5.3 Models for conflicts-free trajectory planning

This section presents the features perceived and predicted by the models and also the methods used along the pipeline.

5.3.1 Features considered by the models

Next sections present the features that the different models perceive and predict, while considering the limitations proposed by the datasets, regarding data availability and also inherent data limitations. Models exploited comprise: a) a trajectory prediction model without considering conflicts, b) models of the ATCO behavior predicting the ATCO reactions and modeling the ATCO policy and c) models predicting the evolution of the conflict resolution maneuver decided.

5.3.1.1 Trajectory prediction model without considering conflicts

For this model, the formulation of states and actions presented in section 3.3 is extended. Changes made, address the following: a) the state-action formulation considered here is able to generalize beyond OD pairs, taking a step towards satisfying requirement R1, presented in section 5.2, b) weather data are not exploited as the focus here is on CD&R.

Following the formulation of trajectory states presented in section 4.5, a $fixpoint$ is used as a reference point, in order to allow features to be independent from the airspace and origin–destination pair considered. This results to a formulation which does not exploit the "absolute" spatial position of the aircraft, supporting generalization beyond specific areas of responsibility and specific OD pairs and thus, satisfying requirement R1. As a $fixpoint$, the 2D point at which the boundary of the considered spatiotemporal area SA crosses the line connecting the origin and the destination airports is used.

Given the $fixpoint$, the following states and actions are considered:

State features include a) the relative position of the aircraft w.r.t. a fixpoint, specified as the difference $\langle fixpoint_x - aircraft_x, fixpoint_y - aircraft_y \rangle$, where $\langle fixpoint_x, fixpoint_y \rangle$ denote transformed longitude and latitude coordinates to 2D Cartesian coordinates of the fixpoint and $\langle aircraft_x, aircraft_y \rangle$ denote transformed longitude and latitude coordinates of the aircraft's position, b) the aircraft's altitude (h), c) the relative bearing of the aircraft w.r.t. a fixpoint denoted by b_f, as well as the magnitudes of the aircraft's horizontal (s_h) and vertical (s_v) speeds. Longitude and latitude are transformed to 2D Cartesian coordinates using the EPSG:2062[1] projection.

Actions affect the aircraft's positions by applying changes to the aircraft's bearing, and horizontal and vertical speeds. Specifically actions are of the form $(\Delta b_f, \Delta s_h, \Delta s_v)$. Then the position of the aircraft at time point t is determined by: a) the constant period Δt between two consecutive time points t and $t-1$ and b) the bearing and the horizontal and vertical speeds $(b_{f,t}, s_{h,t}, s_{v,t})$,

[1] https://epsg.io/2062

which are given by ($b_{f,t-1} + \Delta b_f$, $s_{h,t-1} + \Delta s_h$, $s_{v,t-1} + \Delta s_v$). Specifically the position of the aircraft from state $t-1$ to state t changes according to the following functions:

$$aircraft_{x,t} = aircraft_{x,t-1} + sin(b_{f,t}) * s_h * \Delta t \tag{5.1}$$

$$aircraft_{y,t} = aircraft_{y,t-1} + cos(b_{f,t}) * s_h * \Delta t \tag{5.2}$$

$$h_t = h_{t-1} + s_v * \Delta t \tag{5.3}$$

5.3.1.2 Models for predicting the ATCO reactions and modeling the ATCO policy

Models for predicting the ATCO reactions and modeling the ATCO policy perceive the following feature vector s_r presented in section 4.5, repeated here for conciseness.

$$s_r = (b_f, d_f, s_h, s_v, v_i)$$

where b_f is the relative bearing with regard to a fixpoint (defined in section 5.3.1.1), d_f is the distance from that fixpoint, s_h and s_v are the magnitudes of the aircraft's horizontal and vertical speeds respectively, and v_i is the vector $\langle e_{i1}, \ldots, e_{ik} \rangle$, where each e_{ij} includes features of conflicts with neighbor trajectories T_j as these are described in section 4.2.1:

$$e_{ij} = \langle dh_{cpa_j}, dv_{cpa_j}, t_{cpa_j}, d_{cp_j}, t_{cp_j}, \sin(a_j), \cos(a_j), \sin(b_j), \cos(b_j) \rangle$$

dh_{cpa_j} and dv_{cpa_j} are the horizontal and vertical distances of the ownship from an aircraft j at the CPA, and t_{cpa_j} is the time of the ownship to CPA. d_{cp_j} is the distance between the ownship and the aircraft j when the first of these is at the crossing point, and t_{cp_j} is the time until the first of the aircraft is at the crossing point. The intersection angle between the two trajectories is a_j, and b_j is the relative bearing of the ownship with regard to the aircraft j at the CPA.

The set of ATCO reaction modes comprises three high-level reactions, introduced in section 4.6.2:

- C_0: No conflicts detected, and no resolution action is applied.

- C_1: At least one conflict is detected, and a resolution action is applied.

- C_2: At least one conflict is detected but no resolution action is applied.

Types of conflict resolution actions considered here are the following:

- A_1: "Speed change"

- A_2: "Direct to waypoint"

Resolution action A_3: "Radar vectoring" is not considered here as the number of available samples of this action is small for training the maneuver prediction model.

5.3.1.3 Models predicting the trajectory's evolution during the execution of a maneuver

Models for predicting the trajectory evolution during the execution of a maneuver, in addition to features regarding the ownship's position and direction of movement, also consider neighboring aircraft. This is done so as potential conflicts caused by neighboring flights affect how a maneuver is executed, i.e., towards which waypoint the ownship will be directed in case of a

direct-to maneuver, or how and when its speed will change in case of a speed change maneuver. Following the trajectory states formulation presented in section 4.5, the state considered by the TPMwoCC, presented in section 5.3.1.1, is enriched with information about neighboring aircraft. Specifically, the vector $v_i = \langle e_{i1}, \ldots, e_{ik} \rangle$, also presented in the previous section, is considered, where each e_{ij} includes features of conflicts with neighbor trajectories T_j.

To model the maneuver evolution, the model must consider the different conflict resolution action types. This can be achieved either by a) using an indication in the state and training a single model for all resolution action maneuvers or by b) training a different model per resolution action type. This work follows the simpler second approach and trains a maneuver prediction model per resolution action type. As already stated, data indicate only the resolution action type (e.g. speed change) and not the resolution in its full detail (e.g. how speed has been changed in case of a speed change resolution action). Thus resolution action maneuvers can only be distinguished according to the corresponding action type. This results to training one model per resolution action type. This creates challenges, as given a resolution action type the trajectory evolution can vary to a large extent (e.g. both acceleration and deceleration maneuvers correspond to the speed change type).

Based on the two resolution action types considered in the ATCO policy modeling formulation the following two CRMP models are considered:

- Trajectory Predictor, Direct-to Maneuver (TPDT): This model predicts how the trajectory will evolve when executing a “direct-to” maneuver.
- Trajectory Predictor, Speed Change Maneuver (TPSC): This model predicts how the trajectory will evolve when executing a “speed change” maneuver.

5.3.2 Methods used

This study presents the devised framework, combining models of ATCO behavior for resolving conflicts with models of predicting trajectories, with the goal to provide conflicts-free trajectories. It also presents and evaluates an instantiation of this framework using the following algorithms for each model. To train the trajectory prediction models this study uses the GAIL algorithm presented in section 3.3.2. To train models for predicting the ATCO reactions, predicting “whether” and “when” the ATCO will react, this study considers in addition to the VAE method presented in section 4.6.3 the RF method presented in section 2.1.1.3. For modeling the ATCOs’ policy training model that predicts “how” the ATCO will react, this study uses the RF algorithm presented in section 4.7.2.

5.4 Experimental evaluation

To evaluate the proposed methods, this study simulates the flight-centric concept using the sector ignorant setting presented in section 4.8.1.1. The sector related setting is not considered here ,because, as discussed in section 4.8.1.4, it resulted to worst model performance compared to the sector ignorant case. Presented methods are evaluated using 5-fold cross validation.

5.4.1 Data sets and preprocessing

To evaluate the conflicts-free trajectory planning pipeline this study exploits data from the Spanish airspace, considering flights over Spain, without sacrificing the generality of the methods introduced. The data sources comprise:

- Surveillance data: operational quality data with actual flights (raw) trajectories (Spanish ATC Platform SACTA).
- ATCO events: provides actions taken by the ATCOs in order to ensure safety of flights (provided by ATON).

Data sources exploited in this study are described in more detail in the appendix, section A.2.

The datasets used to train the trajectory prediction models (TPMwoCC, TPDT and TPSC) contain trajectories between Lisbon (LPPT)- Paris (LFPO). For training models of ATCO behavior (ARP, RATPr) ATCO resolution actions corresponding to the following 5 OD pairs are used: Malaga (LEMG)–Gatwick (EGKK), Malaga (LEMG)–Amsterdam (EHAM), Lisbon (LPPT)–Paris (LFPO), Zurich (LSZH)–Lisbon (LPPT), and Geneva (LSGG)–Lisbon (LPPT). This is done so, in order to collect a sufficiently large number of samples for training. For testing the TPP the test sets of all participating methods have been synced, i.e. the corresponding trajectories are excluded from training.

Regarding the models of ATCO behavior this study reports results on a) test sets containing samples from all OD pairs and b) test sets of the TPP containing trajectories between LPPT-LFPO. This study considers only ATCO resolution actions issued at the en-route phase of operations and filters out the climb and descent parts of the trajectories. In addition, only trajectories that have at least one ATCO resolution action and an associated RATP are considered.

ATCO events are associated to potential conflicts (recall that the datasets do not provide these conflicts) that have been detected, either at the point of the ATCO event or in a time window of $window\_duration$ seconds prior to the ATCO event. However, there are cases where there is an ATCO resolution action for a trajectory but no conflicts are detected. These cases are filtered out. The trajectory point (if any) in the specified time window at which a conflict is detected and is temporally closest to the point where the ATCO resolution action is indicated, is considered to be the actual resolution action point (mentioned as RATP).

The final dataset for all 5 OD pairs includes 793 resolution action associated with 634 trajectories consisting of 326 “speed change” and 374 “direct to” resolution actions.

Learning multimodal behavior is a challenging task that is not addressed by GAIL, which is the IL algorithm used in this work. Thus the focal trajectories between LPPT and LFPO are clustered w.r.t. the aircraft’s position and the models of the pipeline are trained and tested on the cluster with the most trajectories.

This results to 136 trajectories corresponding to 174 resolution actions consisting of 97 “speed change” and 77 “direct to” resolution actions. Finally, it must be noted that the available ATCO events dataset covers the Spanish airspace and thus only the points of the trajectories that are in this airspace are considered. However, the proposed method is generic, and can be applied in any airspace.

Next subsections present the datasets used and the pre-processing applied to provide these datasets to each of the models used in the pipeline.

5.4.1.1 TPMwoCC

For this model surveillance data were used. Weather data are not exploited here. The preprocessing stage interpolates points in trajectories, so that two subsequent points have a temporal distance of $\Delta t = 5$ and this is also the time step used in for this model.

5.4.1.2 TPDT and TPSC

To train the TPDT and TPSC models, predicting how the trajectory will evolve after a specific resolution action is issued, this study exploits surveillance data and ATCO events. In this case ATCO events were not used in the features perceived by the model but only to identify the starting point of a maneuver. Specifically training samples are maneuvers corresponding to subtrajectories comprising trajectory points from the point of the resolution action and on, until either the end of the trajectory or the next resolution action, if any. Then the resulting subtrajectories are grouped according to the type of the resolution action, direct to or speed change, training one model for each type. As in the TPMwoCC case, trajectories are interpolated so that two subsequent points have a temporal distance of $\Delta t = 5$ and this is also the time step used for this model.

5.4.1.3 ARP and RATPr

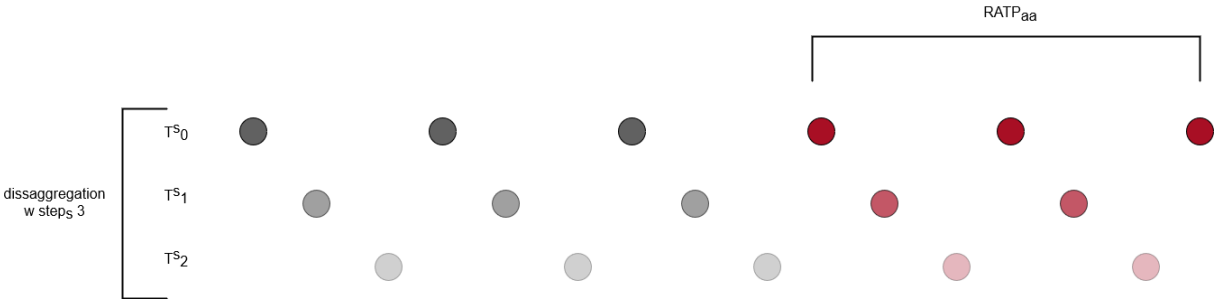
For the ARP and the RATPr models the data sources used comprise surveillance data and ATCO events. As described in section 4.8.1.2, when training the ARP model this study applies data-augmentation to tackle the datataset imbalance w.r.t. the different modes of the ATCO reactions: The trajectory points in a time window of 250 seconds before the ATCOs' resolution action are annotated with C_1, except the points at which no conflicts are detected. For the combined method considered in this section subsampling can be applied only to a limited extent, in order to satisfy requirement R2 (section 5.2): a constant time step between trajectory points must be used when testing the framework. As discussed when defining requirement R2, using a constant time step for all trajectory points, changes the prior of the modes of the ATCO's reactions.

This study tries to mitigate this prior shift by considering a subsampling step, denoted as $step_s$, and disaggregating each trajectory to $step_s$ subtrajectories. Denoting the initial trajectory T and its size $|T|$ and considering the enriched trajectory points $s_{r,t}$ at each time point $t \in 0, ..., |T|$, T is disaggregated into $step_s$ subtrajectories T^s as follows: To create each subtrajectory, start from the trajectory point $s_{r,i}$ for each of $i \in \{0, ..., step_s - 1\}$ and consider each point every $step_s$ trajectory points. Thus each subtrajectory T^s_i with $i \in \{0, ..., step_s - 1\}$ will contain the trajectory points $s_{r,i+k*step_s}$ for all k for which $k * step_s \leq |T|$. The set of trajectory points containing an actual $RATP$ of T and its corresponding annotated $RATPs$ is denoted by $RATP_{aa}$. Recall that the annotated RATPs are trajectory points that precede the actual RATP in a time window of 250 seconds and at which a conflict is detected. Such points are annotated with C_1 in order to balance the prior distribution of the ATCO modes of behavior. This also, somehow addresses the uncertainty of ATCOs about the time to issue a resolution action. Figure 5.3 shows the pre-processing steps for $step_s = 3$. Figure 5.3a shows the original annotated trajectory T. Red points denote $RATP_{aa}$ points and the difference in the opacity denotes which points correspond to each subtrajectory T^s_i. These are shown in the disaggregated form in figure 5.3b.

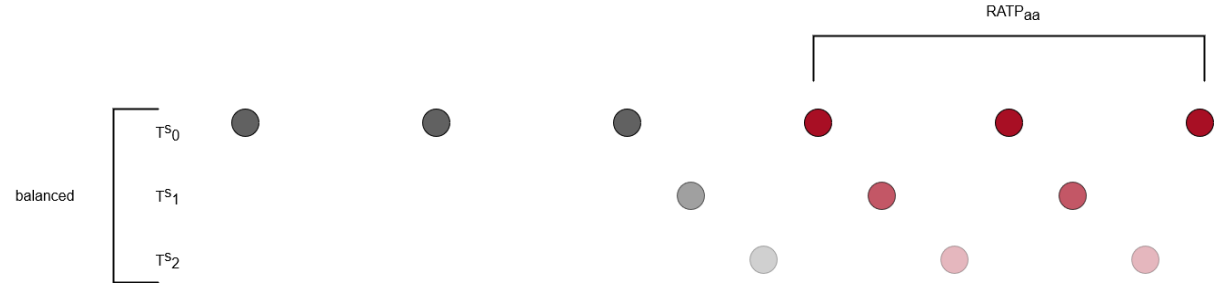
To balance the ATCO modes of behavior the subtrajectories T^s_i for $i \geq 1$ are processed as follows: a) split T^s_i at each $RATP_{aa}$, b) filter out trajectory points not in some $RATP_{aa}$, but keep the point that precedes each $RATP_{aa}$. Figure 5.3c shows the disaggregated trajectories corresponding to T after this prior balancing step. Recall, that the VAE model, introduced in section 4.6.3 considers in its input features the previously predicted mode of ATCO behavior. The point preceding each $RATP_{aa}$ is kept here, so that such models, have the chance to perceive the correct mode at $RATP_{aa}$ and not a randomly initialized one. This study sets $step_s = 6$ corresponding to a time step of 30 seconds (recall points have a temporal distance of $\Delta t = 5$).

To train ARP models this study uses subtrajectories produced by the aforementioned procedure. The idea behind this preprocessing scheme is to exploit all data points used to train the VAE

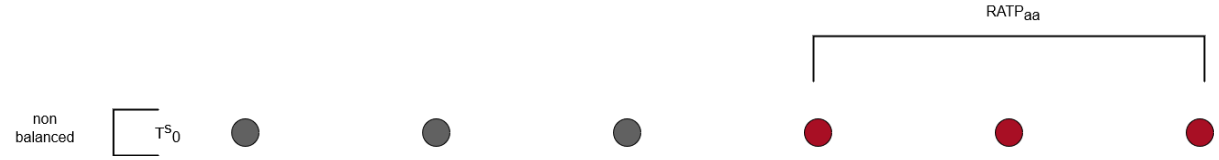
(a) The original annotated trajectory



(b) Disaggregated subtrajectories of T



(c) Trajectories corresponding to the balanced case.



(d) Trajectories corresponding to the non balanced case.

Figure 5.3: The preprocessing steps for $step_s = 3$. Figure 5.3a shows the original annotated trajectory T. Red points denote $RATP_{aa}$ points and difference in the opacity denotes which points correspond to each subtrajectory T_i^s. These are shown in the disaggregated form in figure 5.3b. Figure 5.3c shows the disaggregated trajectories corresponding to T for the balanced case and figure 5.3d shows which subtrajectory corresponds to T for the non-balanced case.

model in chapter 4, where good performance was achieved by balancing the prior distribution of the ATCO modes, while maintaining a constant time step. In the experiments below the *balanced resampling strategy*, denotes the test trajectories that have been preprocessed using the above procedure. These are shown in Figure 5.3c. The *non-balanced resampling strategy* denotes that testing is applied on subtrajectories T_0^s per trajectory T. These are shown in figure 5.3d.

5.4.2 Experimental results

This section presents results achieved by all AI/ML methods incorporated in the overall TPP method. The presented framework is a first approach towards conflicts-free trajectory planning and aims to answer if, and to what extent methods for trajectory prediction and CD&R suffice for creating a method for planning conflicts-free trajectories. In doing so this study presents challenges, emerging when combining the models and ways that these challenges can be over-

come.

This section is structured as follows:

- First the ARP model is tested on historical trajectories, on the balanced and non-balanced cases. The purpose of this evaluation is two-fold:
  - On the one hand, when using historical trajectories, the corresponding historical ATCO reactions are available, providing the ground truth required to evaluate the model. This is not the case for trajectories generated by the TPP, as when testing the TPP, the ARP model predicts the ATCO reactions on trajectories generated by the TPMwoCC. Thus, having tested the ARP model on historical trajectories, is critical to understand results with regard to trajectories generated by TPP.
  - On the other hand, results on the balanced and non-balanced cases, show how the prior shift introduced by the constant time step affects the model. At this point the VAE model is also compared against an RF model and the model that is most resilient to the prior shift is selected to be used as the ARP.
- Next, results of the TPP are reported. This is done, in order to evaluate if, and to what extent methods for trajectory prediction and CD&R suffice, for creating a method for planning conflicts-free trajectories. When testing the TPP the ARP model predicts the ATCO reactions on trajectories generated by the TPMwoCC. On these trajectories there is no ground truth regarding the ARP's predictions, as historical ATCO reactions are not available. Thus to better understand the TPP's behavior this study examines whether/how the TPP reacted to losses that occurred.
- Finally, the trajectory prediction methods are evaluated. Previous analysis shows that evaluating their performance is important. This contributes towards understanding whether/how the TPP reacted to losses that occurred. Trajectory prediction methods evaluated here include the following: TPMwoCC, the individual models for the maneuver prediction, TPDT and TPSC and the whole pipeline (TPP).

Tables 5.1 and 5.2 report experimental results for the ATCO's modes of behavior, achieved by the VAE and the RF using 5-fold cross validation and considering samples from all OD pairs for the non-balanced and balanced cases, respectively. Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes of ATCOs' behavior, for the non-weighted and weighted measures. The non-balanced case is critical to understand how models perform when tested on the same setting as the one they will operate when used in the TPP. On the other hand the balanced case complements the comparison between the models, showing how they perform when there is no distribution shift between the training and testing sets. Results show that when considering the non balanced case the RF model outperforms the VAE. On the other hand the VAE provides more accurate predictions than the RF in the balanced case. This implies that the RF model is more robust to the distribution shift while the VAE model could potentially perform better if the distribution shift is minimized. As the setting at which the ARP model will operate when used in the TPP corresponds to the non-balanced case, the RF model is selected to be used in the TPP pipeline.

Finally Table 5.3 completes the evaluation of the ARP model on historical data by presenting results achieved by the RF model, on the TPP's test set containing flight trajectories of the LPPT-LFPO OD pair. Results report the precision, recall and f1 score. Evaluation shows that the model achieves high f1-scores for the C_0 and C_2 modes but low f1-score for the C_1 class. The weighted

Table 5.1: Experimental results, achieved by the VAE and the RF using 5-fold cross validation for the non-balanced case considering samples from all OD pairs. Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes of ATCOs' behavior, for the non-weighted and weighted measures.

model	resampling	test set ODs	modes non-weighted	modes weighted
VAE	Non-Balanced	All	precision C0: 1.00 ± 0.00 C1: 0.30 ± 0.04 C2: 0.88 ± 0.01 recall C0: 1.00 ± 0.00 C1: 0.32 ± 0.03 C2: 0.86 ± 0.02 f1-score C0: 1.00 ± 0.00 C1: 0.31 ± 0.03 C2: 0.87 ± 0.01	precision C0: 1.00 ± 0.00 C1: 0.32 ± 0.04 C2: 0.97 ± 0.00 recall C0: 1.00 ± 0.00 C1: 0.69 ± 0.03 C2: 0.88 ± 0.01 f1-score C0: 1.00 ± 0.00 C1: 0.44 ± 0.04 C2: 0.92 ± 0.01
RF	Non-Balanced	All	precision C0: 1.00 ± 0.00 C1: 0.46 ± 0.04 C2: 0.91 ± 0.01 recall C0: 1.00 ± 0.00 C1: 0.51 ± 0.07 C2: 0.91 ± 0.02 f1-score C0: 1.00 ± 0.00 C1: 0.48 ± 0.04 C2: 0.90 ± 0.01	precision C0: 1.00 ± 0.00 C1: 0.50 ± 0.04 C2: 0.98 ± 0.00 recall C0: 1.00 ± 0.00 C1: 0.83 ± 0.03 C2: 0.91 ± 0.02 f1-score C0: 1.00 ± 0.00 C1: 0.62 ± 0.03 C2: 0.94 ± 0.01

measures show that the model achieves low precision w.r.t. the mode C_1. This is because the number of positive samples for this mode is small, thus a small number of false positive predictions greatly reduces precision. On the other hand the model achieves high weighted recall showing that the model reacts at critical cases where a resolution action should be issued.

Next the whole pipeline is evaluated. Table 5.4 reports for different horizontal separation thresholds the average number of Losses of Separation (LsOS) per trajectory (Avg LsOS / trajectory) for the historical data, TPMwoCC and TPP. To compute this the total LsOS accross all folds are divided by the total number of trajectories. Results show that TPP is far from achieving the performance seen in the historical data, but achieves a small constant improvement over TPMwoCC in terms of the average number of LsOS per trajectory. As models for modeling the ATCO behavior (ARP and RATPr) operate on trajectories produced by TPMwoCC, there are no historical ATCO reactions for these trajectories and thus there is no ground truth for evaluating their performance. So in order to better understand the factors resulting to the observed performance of the TPP, this study examines why these LsOS occurred and whether/how the TPP reacted to these.

Table 5.2: Experimental results achieved by the VAE and the RF for the balanced case using 5-fold cross validation considering samples from all OD pairs. Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes of ATCOs' behavior, for the non-weighted and weighted measures.

model	resampling	test set ODs	modes non-weighted	modes weighted
VAE	Balanced	All	precision C0: 1.00 ± 0.00 C1: 0.86 ± 0.01 C2: 0.82 ± 0.02 recall C0: 1.00 ± 0.00 C1: 0.81 ± 0.02 C2: 0.87 ± 0.02 f1-score C0: 1.00 ± 0.00 C1: 0.83 ± 0.01 C2: 0.84 ± 0.01	precision C0: 1.00 ± 0.00 C1: 0.87 ± 0.01 C2: 0.95 ± 0.00 recall C0: 1.00 ± 0.00 C1: 0.94 ± 0.01 C2: 0.89 ± 0.01 f1-score C0: 1.00 ± 0.00 C1: 0.90 ± 0.01 C2: 0.92 ± 0.01
RF	Balanced	All	precision C0: 1.00 ± 0.00 C1: 0.80 ± 0.03 C2: 0.64 ± 0.03 recall C0: 1.00 ± 0.00 C1: 0.52 ± 0.06 C2: 0.87 ± 0.02 f1-score C0: 1.00 ± 0.00 C1: 0.62 ± 0.05 C2: 0.74 ± 0.02	precision C0: 1.00 ± 0.00 C1: 0.83 ± 0.02 C2: 0.92 ± 0.01 recall C0: 1.00 ± 0.00 C1: 0.82 ± 0.03 C2: 0.92 ± 0.02 f1-score C0: 1.00 ± 0.00 C1: 0.83 ± 0.02 C2: 0.92 ± 0.01

Table 5.5 reports the average number of LsOS per trajectory that were predicted (Avg Predicted LsOS/ trajectory) for the TPP and the average number of LsOS per trajectory where the ARP issued a resolution action (Avg Predicted, Reacted LsOS / trajectory). A Loss of Separation (LOS) between the ownship and an aircraft is considered to be predicted if a conflict was detected between the ownship and that aircraft in the prediction horizon that the model uses. The model is considered to have reacted to a LOS between the ownship and the aircraft, if a conflict was detected between the ownship and that aircraft in the prediction horizon that the model uses, and the model issued a resolution at the time point that the conflict was detected.

Results show, over 50% of the LsOS were predicted and TPP did not react to most of them. However recall that a LOS is considered predicted if a conflict was detected between the ownship and the aircraft participating in the LOS in the prediction horizon that the model uses. These also include LsOS for which the corresponding conflict was detected for a brief moment e.g. at only one time point prior to the LOS and potentially with a large time difference between the detection of the conflict and the realization of the LOS. In such cases the detected conflict could concern a different LOS than the one realized. This means that although a conflict was detected between the flights participating at the LOS, the LOS is practically "sudden".

Table 5.3: Experimental results of the sector-ignorant case achieved by the RF on the LPPT-LFPO OD pair. Columns report the 95% confidence interval of precision, recall and f1-score w.r.t. the modes of the ATCOs behavior, for the non-weighted and weighted measures.

model	resampling	test set ODs	modes non-weighted	modes weighted
RF	Non-Balanced	LPPT-LFPO	precision	precision
			C0: 1 ± 0.00	C0: 1 ± 0.00
			C1: 0.41 ± 0.05	C1: 0.45 ± 0.04
			C2: 0.87 ± 0.02	C2: 0.97 ± 0.00
			recall	recall
			C0: 1 ± 0.00	C0: 1 ± 0.00
			C1: 0.52 ± 0.07	C1: 0.84 ± 0.04
			C2: 0.81 ± 0.02	C2: 0.83 ± 0.01
			f1-score	f1-score
			C0: 1 ± 0.00	C0: 1 ± 0.00
			C1: 0.46 ± 0.05	C1: 0.58 ± 0.05
			C2: 0.84 ± 0.01	C2: 0.90 ± 0.01

Table 5.4: The average number of LsOS per trajectory for the historical data, TPMwoCC and TPP.

Separation threshold	Historical	TPMwoCC	TPP
$5NM$	0.23	1.91	1.35
$7NM$	0.46	2.77	2.12
$10NM$	1.20	4.42	3.46
$12NM$	1.82	5.71	4.50
$15NM$	2.90	7.31	5.90

Table 5.5: The average number of LsOS per trajectory that were predicted (Avg Predicted LsOS/ trajectory) for the TPP case, and the average number of LsOS per trajectory that were predicted and the TPP reacted to (Avg Predicted, Reacted LsOS / trajectory).

Separation threshold	Avg Predicted LsOS / trajectory	Avg Predicted and Reacted LsOS / trajectory
$5NM$	1.	0.41
$7NM$	1.52	0.58
$10NM$	2.3	0.86
$12NM$	2.83	1.01
$15NM$	3.36	1.17

Motivated by this observation, to better understand the ARP's behavior, predicted LsOS are divided into consistently and inconsistently predicted LsOS. As consistently predicted, are defined those where a conflict was detected between the ownship and the aircraft participating in the LOS at least D_{times} times in the prediction horizon, and the last time before the LOS that were predicted, was at most D_t seconds before the LOS. D_{times} was set to 5 which is the median value of the times a LOS was predicted and D_t was set to 120s as it was empirically considered a valid value given that the ARP's step is 30 s. As inconsistently predicted are considered the predicted LsOS that are not consistently predicted.

Table 5.6 shows the average consistently predicted LsOS per trajectory and the average reactions to consistently predicted LsOS per trajectory for the different separation thresholds.

Table 5.6: The average LsOS per trajectory that were consistently predicted (Avg consistently predicted LsOS/trajectory) and the average reactions of the ARP to consistently predicted LsOS per trajectory (Avg reactions to consistently predicted LsOS /trajectory)

Separation threshold	**Avg consistently predicted LsOS/ trajectory**	**Avg reactions to consistently predicted LsOS /trajectory**
$5NM$	0.41	0.21
$7NM$	0.62	0.31
$10NM$	0.83	0.41
$12NM$	0.93	0.5
$15NM$	1.01	0.56

Results show that the ARP model reacts to approximately 50% of these cases.

Summarizing, reported results presented up to this point show that the TPP model does not react to conflicts potentially leading to LsOS. For some of these cases conflicts are inconsistently predicted and the LOS is practically "sudden". To better understand the TPP's behavior, cases where conflict detection is consistent have been detected and used for further evaluation. The TPP issues a resolution action at 50% of these cases.

We conjecture that inconsistently predicted LsOS are caused by small errors made by the trajectory prediction models. For example, historical trajectories most of the time maintain a certain altitude at the en route phase. On the other hand, trajectory prediction models tend to have small fluctuations at the vertical plane. Such changes could result to sudden LsOS as the conflict detection mechanism considers a vertical speed that is changed abruptly by the trajectory prediction model.

To provide evidence for this conjecture, Figure 5.4 shows the number of average trajectory points per trajectory, where the altitude changes over 5 ft for different altitude intervals, for historical trajectories and for trajectories predicted by TPMwoCC. As it is highly unlikely that the model will ever predict zero vertical speed change, really small changes i.e. less than 5ft between consecutive points are filtered out. Figure 5.4 shows that for the third case, which corresponds to the en route phase of the flights, trajectories predicted by the TPMwoCC have a much greater number of altitude changes.

Regarding consistently predicted LsOS, the TPP does not react to 50% of the cases although it achieves high recall on historical trajectories. This means that either a) the model should have reacted to consistently predicted LsOS but does not react because such cases are not sufficiently represented during training, or b) the model correctly does not react because in the real world case the conflict would solve itself, implying that trajectory prediction is not that accurate.

Regarding a) there is a number of potential reasons:

1. Trajectory prediction methods deviate significantly from the real world distribution represented in the dataset w.r.t. to the neighboring flights.

2. The ATCO would issue a resolution action that is not considered in this study. This study filters out ATCO resolution actions of "flight level change" and "radar vectoring", as the number of samples is small.

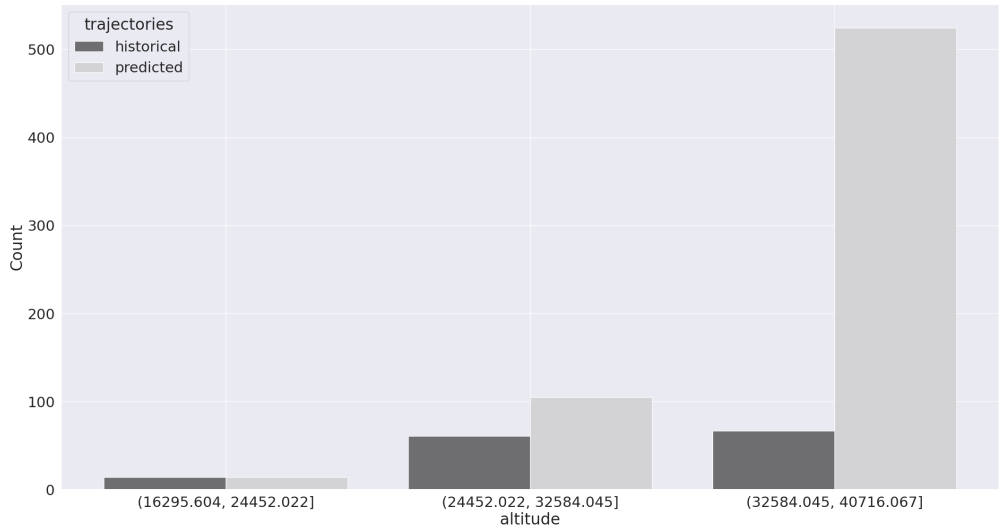


Figure 5.4: Number of average trajectory points per trajectory, where the altitude changes over 5 ft for different altitude intervals shown in the X-axis, for historical trajectories and for trajectories predicted by TPMwoCC.

3. More data are needed to sufficiently cover the state space. Recall that the dataset including all 5 OD pairs contains 634 trajectories, while the number of trajectories between LPPT-LFPO is 136.

4. There is a significant gap between what the ATCO sees and what the model observes according to the conflict detection methodology and the dataset. As the ATCO observations are not included in the dataset it is not possible to validate how close the conflict detection methodology that is applied here is to the conflict detection methodology used by the ATCOs'.

As the performance of the trajectory predictors seems to be of importance in order to understand the presented results, next the trajectory prediction models are evaluated and the trajectory prediction formulation presented in this chapter is compared to the formulation presented in Chapter 3.

Table 5.7 shows the average RMSE error of the predicted vs the actual (test) trajectory in meters for each of the three dimensions and in 3D, together with the average absolute ATE, CTE, and VE, in meters. The table reports the results provided by the TPMwoCC, the trajectory prediction formulation of chapter 3, TPDT, TPSC and the pipeline (TPP). Notice that the trajectory prediction formulation of chapter 3 is applied here without features regarding the weather conditions as the focus here is on CD&R. Results show that trajectory prediction methods used in this study have good performance with a worst case average 3D error of 15359 meters performed by TPDT. Comparing the formulation of TPMwoCC presented in this chapter with the TPMwoCC formulation of chapter 3, the TPMwoCC formulation used in this chapter performs slightly worst but has the benefit of being independent of the OD pair that is applied to. Finally the TPP performs worst in terms of trajectory prediction compared to the TPMwoCC and it does not manage to considerably reduce LsOS as reported in table 5.4.

Table 5.7: Prediction Errors (in meters)

Method	Long	Lat	Alt	3D	ATE	CTE	VE
TPMwoCC	15962.04	17671.4	566.96	13920.7	27095.97	7990.72	516.80
TP chap.3	15249.82	14826.13	331.09	12503.52	21901.75	9391.16	316.23
TPDT	17876.15	19067.87	299.47	15389.52	14792.71	13766.58	273.82
TPSC	12779.64	12630.11	301.2	10582.87	12358.96	12108.28	243.58
TPP	17937.71	18862.06	524.8	15229.16	25073.05	12121.72	510.21

5.5 Conclusions

The aim of this chapter is to investigate if and to what extent methods for trajectory prediction and CD&R suffice for creating a method for planning conflicts-free trajectories. Towards this, this study presented a straightforward way of combining the models for trajectory prediction and CD&R into a unified approach for planning conflicts-free trajectories. By evaluating the presented method using real world data, this study aims to detect challenges, emerging when combining the models and ways that these can be overcome. By doing so, this study provides insights and issues to be addressed towards a method for conflicts-free trajectory planning that provides accurate prediction of trajectories while effectively solving conflicts. Specifically, regarding the combination of ML models towards a method for predicting conflicts-free trajectories this study:

1. Specifies a formulation of the conflicts-free trajectory prediction as a data driven problem;
2. Specifies the prediction of the conflict resolution maneuver problem.
3. It studies a framework combining different models towards providing conflicts-free trajectories based on real-world historical data sets;
4. It evaluates the proposed framework using real-world data.

Motivated by the prior shift introduced by the sub-sampling used to balance the modes of the ATCO's reactions, this study firstly evaluated the ARP model on historical trajectories. Results show that, as expected, the prior shift has a negative effect hindering the models' performance. Regarding the prior shift this study evaluates and compares two models. The VAE model of Chapter 4 and an RF model. Results show that the RF model is more resilient to the prior change, while the VAE model provides better results when the prior change is minimized. An important observation that will be revisited when analyzing the TPP's performance is, that despite the prior shift the RF method achieves high weighted recall, 0.84, for mode C_1 on the non-balanced case. This shows that the model reacts when it should in most cases when tested on historical trajectories.

Next, this study proceeds to evaluate the TPP. According to the experimental evaluation, although there is a small improvement on the LsOS that occurred when the models for conflict detection and resolution are employed, the TPP performs poorly compared to the historical LOS free trajectories and there are cases where LsOS occur. As there is no available ground truth when testing the TPP, this study focuses on whether/how the TPP reacted to LsOS. According to the experimental evaluation there are cases where LsOS were not predicted consistently and were practically "sudden". This could be due to changes made to the predicted trajectory

by the trajectory predictors, that were not anticipated by the conflict detection methodology. An example of such behavior constitutes the vertical fluctuations that the trajectory prediction models tend to have. On the other hand a considerable percentage of LsOS, 40%, are consistently predicted losses and at 50% of the cases the ARP did not react. Based on the fact that the ARP model achieves high recall for mode C_1 on historical trajectories but does not react to 50% of the consistently predicted LsOS this study conjectures a number of potential reasons based on which the following future research directions towards conflicts-free trajectory planning are outlined:

- First and foremost to solve the CD&R problem the ATCO solves, models have to observe the situation the ATCO observed. As already discussed during this study, available datasets do not include these observations. Having such observations could have an immense impact on model accuracy, and could also help align the conflict detection methodology with how the ATCOs' detect conflicts. A potential future direction towards acquiring ATCO observations is to build a framework that records what the ATCO observes in real time. In addition to this, ATCO resolution actions should be recorded in more detail as currently only the type of the resolution action is available. This creates challenges when combining the models learned into a unified pipeline, as given a resolution action type the trajectory evolution can vary to a large extent due to action type realizations (i.e. both acceleration and deceleration maneuvers correspond to the speed change type).

- Another important observation regards trajectory prediction. Although predicted trajectories are pretty accurate regarding the spatial evolution of flight trajectories, there are other factors that if considered could lead to more accurate trajectory prediction in the context of conflicts-free trajectories. Examples of such cases are the following:

  - There are specific characteristics of the flight trajectories that predicted trajectories should adhere to. Vertical fluctuations experienced in this study do not adhere to how flight trajectories evolve at the en route phase. This reveals the need for further metrics in order to measure adherence to specific features of real world flight trajectories.

  - Predicted trajectories are of the real world distribution when considering spatial features, but there are potentially other factors that are not considered, resulting to predictions being aggregations over these hidden factors hindering the model's accuracy. Such factors could be for example flight plan information. This could have as an effect unrealistic encounters with neighboring flights.

- Finally, gathering more data and using more resolution action types could improve the coverage of the state space resulting to improved performance.

Other future directions involve combining the models of the ATCO behavior developed in this study with methods directly optimizing the CD&R task in terms of effectiveness and efficiency (e.g. miles traveled, number of resolution actions taken, distance between flights etc.). Aligning models that optimize a specific task with human behavior has received a lot of attention recently. This is critical in many domains where safety is important and human practices, preferences and constraints imposed must be respected. In [6],[75] authors aim to increase the trustworthiness of the system, help the human decision maker to understand when to trust the system and ultimately lead the human to finding an optimal policy for the specific task based on the suggestions made by the machine. In [56] authors consider the natural language processing domain and align the models with human behavior in order to avoid toxic comments and generally harmful behavior from their model. They use human labelers to evaluate and la-

bel machine outputs and then apply reinforcement learning to optimize the model based on the recorder human feedback. In this context models of human behavior developed in this study can be exploited to provide a reward term that aligns models that directly optimize the CD&R task with human practices.

Part III

Conclusions

Chapter 6

Conclusions and future study

The main objective of this study is to explore and present state of the art AI/ML algorithms towards planning conflicts-free trajectories. By doing so, it aims to protect ATCOs from overload, reducing their workload and enabling them to deal with complex traffic situations. In the context of this study the conflicts-free trajectory planning task incorporates trajectory prediction and CD&R. Specifically the objective of the conflicts-free trajectory planning task is to predict the evolution of trajectories, while regulating flights to avoid loss of separation. This study follows a data driven approach emphasizing on imitating flights' trajectories and ATCOs' behavior according to demonstrations provided by historical data.

To achieve this main objective, this study advances the state of the art in two major and challenging topics:

1. Data-driven prediction of flight trajectories per OD pair.
2. Data-driven modeling of the ATCOs' behavior in resolving conflicts.
3. And provides a thorough study towards conflicts-free trajectory planning.

Regarding (1) this study formulates the data-driven trajectory prediction problem as an IL task and employs GAIL, a state of the art generative adversarial imitation method to solve it. Evaluation results on real world data show the effectiveness of the method to make accurate predictions for large time horizons (whole trajectory). Reported results were compared to other state of the art methods although a direct and systematic comparison required methods to be trained using the same sets of demonstrated, flown trajectories.

Future plans for the prediction of trajectories include (a) exploiting flight plans to constrain the prediction, (b) extending the method to deal inherently with different modes of trajectory evolution, and (c) generalizing effectively beyond specific OD pairs.

Regarding (2) this study proposes a two-stage data-driven methodology towards meeting the following two objectives:

- Formulate the ATCO reaction prediction problem, towards building a model of ATCO reactions for resolving conflicts. The aim is to answer "whether" and "when" the ATCO de- cides to apply an action to resolve a conflict. Towards predicting the ATCO timely reac- tions to resolve conflicts, this study trains a VAE imitating the demonstrated ATCO behavior in a hierarchical manner. To train the proposed model, this study proposes

a data-driven method for simulating the evolution of trajectories, incorporating uncertainty and revealing the conflicts that ATCOs may have assessed before reacting. The proposed method has been evaluated in two different operational settings (sector-related and sector-ignorant), reporting on the precision, recall and f1-score of predictions. A weighted version of these measures is introduced, to deal with the inherent uncertainties regarding (a) the evolution of trajectories, (b) the detection of conflicts (which are not specified in the dataset), and (c) the ATCO's reaction. According to the experimental evaluation, proposed models accurately predict the mode of the ATCOs' behavior either in the sector-ignorant or in the sector-related setting. Regarding the predictions of resolution actions, results achieved by VAE are not so impressive as those achieved on the prediction of modes, and this was further explored in this study in sections considering the modeling of the ATCOs' policy. Regarding the different experimental settings explored, models perform better at the sector-ignorant case compared to the sector-related. This is something to be further explored in future work.

- Formulate the ATCOs' policy modeling problem, towards building a model of ATCO behavior for resolving conflicts. The aim is to answer "how" the ATCO reacts (i.e. what resolution actions he/she applies) in the presence of conflicts. Towards this, this study formulates the ATCO policy modeling problem as a classification task and studies enhanced AI/ML methods to learn models of ATCOs' policy from real-world historical data sets. Finally it evaluates the proposed AI/ML methods using real-world data, reporting the precision, recall, f1-score and the Matthews Correlation Coefficient between the predictions and the resolution actions of the dataset. Results show that classification methods, such as RF, GTB and NNs achieve good accuracy on predicting ATCOs actions given specific conflicts, but they have limitations which are mostly due to the imperfections of historical data sets exploited.

Inherent data limitations encountered in this study constitute a challenging issue for the development of data-driven methods for conflict detection and resolution. These include the following:

- Historical expert samples (i.e. flown trajectories annotated with ATCO resolution actions) do not indicate, together with the resolution actions, the observations perceived by ATCOs before the resolution action, driving the specific action. To address this issue the aircraft's position is projected into the future in order to reveal the potential conflicts and the corresponding ATCO observations that triggered the ATCO resolution action. This is challenging and introduced noise in the learning process as not all conflicts detected by ATCOs are retrieved with precision. Deviating from the actual ATCO observations can have a negative effect on the models' performance.

- Historical datasets that this work exploits, provided by the Spanish ATON (Automated NORVASE Takes) platform, indicate only the type of the resolution actions instructed by ATCOs (e.g., change speed), and not the actions in full detail (e.g., how speed has been changed and for how long). Thus models of the ATCO policy predict the type of the resolution action that the ATCO would prescribe. This creates challenges when combining the models learned into a unified pipeline as it makes the prediction of the evolution of the trajectory when the aircraft implements a resolution action even harder.

- Different modes of ATCO behavior and resolution action types are imbalanced. To address this issue this study applies resampling techniques. But the change in the prior distribution when the models are tested on constant step trajectories has a negative effect on model performance.

Considering (3), the study provided towards conflicts-free trajectory planning, this study formulates the conflicts-free trajectory prediction as a data driven problem. It also specifies the prediction of the conflict resolution maneuver and addresses it in a data driven way. Accurately predicting how the maneuver will evolve is not completely addressed in this study as resolution actions are not reported in their full detail in the available datasets. Thus given a resolution action type the trajectory evolution can vary to a large extent (i.e. both acceleration and deceleration maneuvers correspond to the speed change type). Considering which maneuver of the prescribed type resolves the conflict more efficiently, could help address this problem more effectively and is a topic of future research, as the emphasis of this study is on data-driven approaches. In the context of this study a unified framework combining different models towards providing conflicts-free trajectories based on real-world historical data sets is developed and evaluated.

Results show that although there is a small improvement on the realizations of LsOS when the models for CD&R are employed compared to predicting trajectories without considering conflicts, multiple issues must be resolved to achieve effective and efficient conflict resolution. The most crucial issue to be considered is that to solve the CD&R problem the ATCO solves, models must observe the situation perceived by the ATCO, driving the specific action. As already discussed, available datasets do not include these observations. Having such observations could improve model performance, and also assist in aligning the conflict detection methodology with how the ATCOs' detect conflicts. To acquire the ATCO observations, a potential approach is to build a framework that records what the ATCO observes in real time. Also, currently only the type of the resolution action is available. Given a resolution action type the trajectory evolution can vary to a large extent due to action type realizations (i.e. both acceleration and deceleration maneuvers correspond to the speed change type). Thus, having only the resolution action type and not the resolution action in full detail creates challenges when combining the models learned into a unified pipeline. Therefore the ATCO resolution actions need to be recorded to their full detail.

Another important point is that following a purely data driven approach, the ATCO objective, which is to maintain the separation minima, remains implicit when optimizing the models, i.e., there is no direct signal that penalizes the model behavior when a LOS occurs. This can be addressed by including a reward term in the context of reinforcement learning that penalizes LOS. Following such an approach, models of ATCO behavior developed in this study can be combined with methods directly optimizing the CD&R task in terms of effectiveness and efficiency, considering additional factors to the separation minima maintenance, e.g., miles traveled, number of resolution actions taken, distance between flights etc.

Aligning models that optimize a specific task with human behavior has received a lot of attention recently. This is critical in many domains where safety is important and human tolerance must be respected. In [6],[75] authors aim to increase the trustworthiness of the system and help the human decision maker to understand when to trust the system, ultimately leading the human to finding an optimal policy for the specific task based on the suggestions made by the machine. In [56] authors consider the Natural Language Processing (NLP) domain and align the models with human behavior in order to avoid toxic comments and generally harmful behavior from their model. They use human labelers to evaluate and label machine outputs and then apply reinforcement learning to optimize the model based on the recorder human feedback. In this context models of human behavior developed in this study can be exploited to provide a reward term that aligns models that directly optimize the CD&R task with human intent.

Summarizing, based on the findings of this study future research directions include the follow-

ing: Enhancing the TPMwoCC model towards exploiting flight plan information, dealing inherently with different modes of trajectory evolution and generalizing beyond specific OD pairs. Gathering more detailed data containing observations perceived by ATCOs before the resolution action and recording the resolution actions in full detail. Also containing the probability the ATCO issues a resolution at a specific state could lead to more balanced datasets w.r.t. the ATCO modes of behavior. Finally and most interestingly, future research involves combining the models of ATCO behavior with models directly optimizing the CD&R task.

Bibliography

[1] Pieter Abbeel and Andrew Y Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 1.

[2] H. Georgiou et al. *Moving Objects Analytics: Survey on Future Location & Trajectory Prediction Methods*. 2018. arXiv: 1807.04639 [cs.LG].

[3] Shun-ichi Amari. "Backpropagation and stochastic gradient descent method". In: *Neurocomputing* 5.4-5 (1993), pp. 185–196.

[4] S. Ayhan and H. Samet. "Aircraft Trajectory Prediction Made Easy with Predictive Analytics". In: *Proc. of the 22nd ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining*. 2016, pp. 21–30. DOI: 10.1145/2939672.2939694.

[5] Samet Ayhan, Pablo Costas, and Hanan Samet. "Prescriptive analytics system for long-range aircraft conflict detection and resolution". In: *26th ACM SIGSPATIAL*. 2018, pp. 239–248.

[6] Nina L Corvelo Benz and Manuel Gomez Rodriguez. "Human-Aligned Calibration for AI-Assisted Decision Making". In: *arXiv preprint arXiv:2306.00074* (2023).

[7] Donald J Berndt and James Clifford. "Using dynamic time warping to find patterns in time series." In: *KDD workshop*. Vol. 10. 16. Seattle, WA, USA: 1994, pp. 359–370.

[8] Chris M Bishop. "Neural networks and their applications". In: *Review of scientific instruments* 65.6 (1994), pp. 1803–1832.

[9] Christopher M Bishop. "Regularization and complexity control in feed-forward networks". In: (1995).

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Vol. 4. 4. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738. DOI: 10.1117/1.2819119.

[11] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[12] Esther Calvo-Fernández et al. "Conflict-free trajectory planning based on a data-driven conflict-resolution model". In: *JGCD* 40.3 (2017), pp. 615–627.

[13] Chao Chen, Andy Liaw, Leo Breiman, et al. "Using random forest to learn imbalanced data". In: *University of California, Berkeley* 110.1-12 (2004), p. 24.

[14] Pengfei Chen et al. "Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 13. 2021, pp. 11442–11450.

[15] T. Cheng, D. Cui, and . Cheng. "Data mining for air traffic flow forecasting: a hybrid model of neural network and statistical analysis". In: *Proc. of the 2003 IEEE Intl. Conf. on Intelligent Transportation Systems* 1 (2003), pp. 211–215.

[16] Felipe Codevilla et al. "Exploring the limitations of behavior cloning for autonomous driving". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9329–9338.

[17] Ramon Dalmau and Eric Allard. “Air Traffic Control using message passing neural networks and multi-agent reinforcement learning”. In: *Proceedings of the 10th SESAR Innovation Days, Virtual Event* (2020), pp. 7–10.
[18] Thomas G Dietterich. “Hierarchical reinforcement learning with the MAXQ value function decomposition”. In: *Journal of artificial intelligence research* 13 (2000), pp. 227–303.
[19] Nourelhouda Dougui et al. “A light-propagation model for aircraft trajectory planning”. In: *Journal of Global Optimization* 56.3 (2013), pp. 873–895.
[20] Yan Duan et al. “Benchmarking deep reinforcement learning for continuous control”. In: *International conference on machine learning*. PMLR. 2016, pp. 1329–1338.
[21] Nicolas Durand and Jean-Baptiste Gotteland. “Genetic algorithms applied to air traffic management”. In: *Metaheuristics for hard optimization*. Springer, 2006, pp. 277–306.
[22] Heinz Erzberger. *Automated conflict resolution for air traffic control*. [National Aeronautics and Space Administration], Ames Research Center, 2005.
[23] C. Finn, S. Levine, and P. Abbeel. “Guided cost learning: Deep inverse optimal control via policy optimization”. In: *ICML*. 2016, pp. 49–58.
[24] Jerome H. Friedman. “Stochastic gradient boosting”. In: *Computational Statistics & Data Analysis* 38.4 (2002), pp. 367–378. ISSN: 0167-9473. DOI: https://doi.org/10.1016/S0167-9473(01)00065-2.
[25] Andrew Fuchs, Andrea Passarella, and Marco Conti. “Modeling, Replicating, and Predicting Human Behavior: A Survey”. In: *ACM Transactions on Autonomous and Adaptive Systems* (2023).
[26] H. V. Georgiou et al. “Semantic-aware aircraft trajectory prediction using flight plans”. In: *Intl Journal of Data Science and Analytics* (2019), pp. 1–14.
[27] Supriyo Ghosh et al. “A deep ensemble multi-agent reinforcement learning approach for air traffic control”. In: *arXiv:2004.01387* (2020).
[28] Justin Gilmer et al. “Message passing neural networks”. In: *Machine Learning Meets Quantum Physics*. Springer, 2020, pp. 199–214.
[29] Chester Gong and Dave McNally. “A methodology for automated trajectory prediction analysis”. In: American Institute of Aeronautics and Astronautics, 2004.
[30] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.
[31] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).
[32] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. “Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning.” In: *Journal of Machine Learning Research* 5.9 (2004).
[33] M. G. Hamed et al. “Statistical prediction of aircraft trajectory : regression methods vs point-mass model”. In: 2013.
[34] Bo Han et al. “Masking: A new perspective of noisy supervision”. In: *Advances in neural information processing systems* 31 (2018).
[35] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*. Vol. 49. 1. NBS Washington, DC, 1952.
[36] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.
[37] J. Ho and S. Ermon. “Generative adversarial imitation learning”. In: *NIPS*. 2016, pp. 4565–4573.
[38] Arianit Islami, Supatcha Chaimatanan, and Daniel Delahaye. “Large-scale 4D trajectory planning”. In: *Air Traffic Management and Systems II*. Springer, 2017, pp. 27–47.

[39] Ralvi Isufaj, David Aranega Sebastia, and Miquel Angel Piera. “Towards Conflict Resolution with Deep Multi-Agent Reinforcement Learning”. In: *ATM Seminar 2021* ().
[40] Eric Jang, Shixiang Gu, and Ben Poole. “Categorical Reparametrization with Gumble-Softmax”. In: *International Conference on Learning Representations (ICLR 2017)*. 2017.
[41] Jiechuan Jiang et al. “Graph Convolutional Reinforcement Learning”. In: *International Conference on Learning Representations*. 2019.
[42] Sham Kakade and John Langford. “Approximately optimal approximate reinforcement learning”. In: *Proceedings of the Nineteenth International Conference on Machine Learning*. 2002, pp. 267–274.
[43] Sham M Kakade. “A natural policy gradient”. In: *Advances in neural information processing systems*. 2002, pp. 1531–1538.
[44] Diederik P Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *stat* 1050 (2014), p. 1.
[45] Diederik P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2014. URL: http://arxiv.org/abs/1412.6980.
[46] Sotiris B Kotsiantis. “Decision trees: a recent overview”. In: *Artificial Intelligence Review* 39 (2013), pp. 261–283.
[47] Mark A Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AIChE journal* 37.2 (1991), pp. 233–243.
[48] Alex Krizhevsky and Geoffrey E Hinton. “Using very deep autoencoders for content-based image retrieval.” In: *ESANN*. Vol. 1. Citeseer. 2011, p. 2.
[49] Tejas D Kulkarni et al. “Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation”. In: *NIPS* 29 (2016), pp. 3675–3683.
[50] A. de Leege, M. van Paassen, and M. Mulder. “A Machine Learning Approach to Trajectory Prediction”. In: *AIAA Guidance, Navigation, and Control (GNC)*. DOI: 10.2514/6.2013-4782. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2013-4782. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2013-4782.
[51] Y. Liu and M. Hansen. *Predicting Aircraft Trajectories: A Deep Generative Convolutional Recurrent Neural Networks Approach*. 2018. arXiv: 1812.11670 [cs.LG].
[52] Xingjun Ma et al. “Normalized loss functions for deep learning with noisy labels”. In: *International conference on machine learning*. PMLR. 2020, pp. 6543–6553.
[53] *NextGen*. https://www.faa.gov/nextgen. Accessed: 2023-06-02.
[54] Duc Tam Nguyen et al. “Self: Learning to filter noisy labels with self-ensembling”. In: *arXiv preprint arXiv:1910.01842* (2019).
[55] International Civil Aviation Organization. *Annex 11 - Air Traffic Services*. 2001.
[56] Long Ouyang et al. “Training language models to follow instructions with human feedback”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744.
[57] George Papadopoulos et al. “Deep reinforcement learning in service of air traffic controllers to resolve tactical conflicts”. In: *Expert Systems with Applications* 236 (2024), p. 121234.
[58] Michael J. Pazzani and Eamonn J. Keogh. “Scaling up dynamic time warping for data mining applications”. In: (2000), pp. 285–289.
[59] Duc-Thinh Pham et al. “A machine learning approach for conflict resolution in dense traffic scenarios with uncertainties”. In: *ATM Seminar 2019*. 2019.
[60] Duc-Thinh Pham et al. “Reinforcement learning for two-aircraft conflict resolution in the presence of uncertainty”. In: *2019 IEEE-RIVF*. IEEE. 2019, pp. 1–6.
[61] Dean A Pomerleau. “Alvinn: An autonomous land vehicle in a neural network”. In: *Advances in neural information processing systems* 1 (1988).

[62] Yunchen Pu et al. “Variational autoencoder for deep learning of images, labels and captions”. In: *Advances in neural information processing systems* 29 (2016).
[63] Marta Ribeiro, Joost Ellerbroek, and Jacco Hoekstra. “Review of conflict resolution methods for manned and unmanned aviation”. In: *Aerospace* 7.6 (2020), p. 79.
[64] SJ van Rooijen et al. “Toward individual-sensitive automation for air traffic control using convolutional neural networks”. In: *Journal of Air Transportation* 28.3 (2020), pp. 105–113.
[65] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).
[66] Mayu Sakurada and Takehisa Yairi. “Anomaly detection using autoencoders with nonlinear dimensionality reduction”. In: *MLSDA 2014 Workshop on Machine Learning for Sensory Data Analysis*. 2014, pp. 4–11.
[67] Nora Schimpf et al. “A Generalized Approach to Aircraft Trajectory Prediction via Supervised Deep Learning”. In: *IEEE Access* (2023).
[68] J. Schulman et al. “Trust region policy optimization”. In: *ICML*. 2015, pp. 1889–1897.
[69] John Schulman et al. “High-Dimensional Continuous Control Using Generalized Advantage Estimation”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2016.
[70] *SESAR Joint Undertaking*. https://www.sesarju.eu/. Accessed: 2023-06-02.
[71] Mohit Sharma et al. “Directed-Info GAIL: Learning Hierarchical Policies from Unsegmented Demonstrations using Directed Information”. In: *International Conference on Learning Representations*. 2018.
[72] Hwanjun Song et al. “Learning from noisy labels with deep neural networks: A survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).
[73] Christos Spatharis, Konstantinos Blekas, and George A. Vouros. “Apprenticeship Learning of Flight Trajectories Prediction with Inverse Reinforcement Learning”. In: *11th Hellenic Conference on Artificial Intelligence*. SETN 2020. Athens, Greece: Association for Computing Machinery, 2020, pp. 241–249. ISBN: 9781450388788. DOI: 10.1145/3411408.3411427. URL: https://doi.org/10.1145/3411408.3411427.
[74] Mudhakar Srivatsa et al. “Towards AI-based Air Traffic Control”. In: *ATM Seminar 2021* ().
[75] Eleni Straitouri et al. “Improving expert predictions with conformal prediction”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 32633–32653.
[76] Richard S Sutton, Doina Precup, and Satinder Singh. “Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning”. In: *Artificial intelligence* 112.1-2 (1999), pp. 181–211.
[77] F. Torabi, G. Warnell, and P. Stone. *Generative Adversarial Imitation from Observation*. 2018. arXiv: 1807.06158 [cs.LG].
[78] Ngoc Phu Tran et al. “An intelligent interactive conflict solver incorporating air traffic controllers’ preferences using reinforcement learning”. In: *ICNS 2019*. IEEE. 2019, pp. 1–8.
[79] Phu N Tran et al. “Aircraft trajectory prediction with enriched intent using encoder-decoder architecture”. In: *IEEE Access* 10 (2022), pp. 17881–17896.
[80] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).
[81] Carl Westin et al. “Personalized and Transparent AI Support for ATC Conflict Detection and Resolution: an Empirical Study”. In: *12th SESAR Innovation Days* (2022).
[82] Ronald J Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In: *Machine learning* 8 (1992), pp. 229–256.

[83] Xiaobo Xia et al. “Robust early-learning: Hindering the memorization of noisy labels”. In: *International conference on learning representations*. 2021.
[84] Junyuan Xie, Linli Xu, and Enhong Chen. “Image denoising and inpainting with deep neural networks”. In: *NIPS* 25 (2012), pp. 341–349.
[85] Y. Yang, J. Zhang, and K.q. Cai. “Terminal-Area Aircraft Intent Inference Approach Based on Online Trajectory Clustering”. In: *TheScientificWorldJournal*. 2015.

Appendix A

A.1 Supplementary material on alternative problem formulation and AI/ML methods for learning the ATCOs' policy.

A.1.1 Learning the ATCOs policy problem as an IL problem

A.1.1.1 Problem specification

Casting the ATCOs policy problem as an IL problem, the aim is to learn a policy that mimics the ATCOs behaviour from demonstrations. In doing so, the problem has to be formulated as a sequential decision problem using a MDP . The goal is to learn a policy $\pi(a|s)$, which defines the conditional distribution over actions $a \in A$ for any state s of interest.

As any imitation problem, the data-driven CD&R task can be defined as follows: Given a set T_E of historical, demonstrated trajectories with conflict resolution actions (i.e., demonstrating ATCOs actions in resolving conflicts in the course of trajectory evolution), the objective is to determine a policy π and a reward function R_E that determines the generation of maximal-expected-cumulative-reward in any trajectory T_π.

The policy prescribes the probability of applying an action a at state s, so as the trajectory to evolve from that state on, maximizing the expected reward. In the context of the CD&R problem, the aim is not to imitate the evolution of the trajectory per se, but to "shape" the trajectory in the presence of conflicts, so as to imitate the resolution of conflicts, close to the demonstrations provided.

Following a data-driven approach, the reward function shows the adherence of predictions to behavioral patterns and policies, as these are demonstrated in historical cases. To avoid critical assumptions on reward functions, many IL methods do not require learning or crafting manually a reward function, and this is the approach is followed here.

At this point the idiosyncrasies of the episodic ATCOs policy imitation task must be clarified, in comparison to any other "classical" trajectory imitation tasks. While IL in its original form aims to imitate a sequential task (i.e. learn the series of actions to be applied in states occurring at any time point $t \in \{0, 1, 2, 3, ...|T| - 1\}$ across a trajectory T), the ATCOs policy imitation is episodic and aims to model ATCOs' decisions on actions applied at specific states where conflicts have been detected, so as the conflict to be resolved in subsequent states. While the goal of the former is the imitation of trajectories' evolution, the goal of the later is the imitation of the resolution of conflicts.

In other words, the following hold: (a) The original imitation task aims to imitate the expert

at any trajectory point, while the ATCOs imitation task aims to imitate the ATCOs at specific time points where conflicts are detected; (b) The original imitation task aims at imitating the evolution of trajectories per se, while the ATCOs imitation task aims to imitate the resolution of the conflicts (or in other words: the trajectories of conflicts' evolution).

Therefore, this study formulates the ATCOs imitation task as an one-stage episodic task triggered in the presence of conflicts: The state space for this task does not include trajectory points, but observations regarding conflicts detected at trajectory points. As the aim is to imitate the resolution of conflicts by ATCOs, the task imitates decisions that cause maneuvers, which in their turn cause the evolution of the detected conflicts, from the time of their detection up to a time horizon, close to the resolution of conflicts demonstrated by historical conflicts-free trajectories.

Therefore, contrary to the classification task where only the trajectory points at which a resolution action was issued are considered, the IL task considers the evolution of the conflicts from the point where ATCOs instruct a conflict resolution action and on.

A.1.1.2 Solving the ATCOs policy problem by imitation

This section describes the GAIL IL method used to learn the ATCOs policy problem.

According to the ATCOs policy IL problem, given a set of historical trajectories annotated with ATCOs events, $\mathbf{T}_{RA_E}$, the aim is to imitate decisions on resolution actions that cause the evolution of the conflicts as demonstrated in historical data. Therefore, starting from a point t_c where conflicts are detected, and where a historical resolution action $ra_t \in RA_E$ was issued, the aim is to imitate the evolution of the conflicts' observations detected at t_c, similarly to what is demonstrated by $\mathbf{T}_{RA_E}$, for a finite horizon H, aiming to bring the distribution of the states generated by the resolution action decided by the imitator as close as possible to that of those demonstrated.

Given that GAIL explores different alternatives on resolution action types before it concludes to learn the demonstrated policy, there are two issues that we need to address: (a) First, although the action type prescribed by the ATCOs policy is demonstrated, we need to simulate the potential effects of all actions; and (b) the effects of any ATCOs action are stochastic, as aircraft may delay to conform, or the actual maneuver implemented may deviate from what actually prescribed.

Both problems are addressed by considering specific realizations of any resolution action type that can be applied. Specifically we consider the following realizations for the resolution action types:

1. $A_{1,k}$: "change the speed k knots", for $k \in \{\pm 10\}$

   – The course of the aircraft is decided according to its current flight plan, heading towards the next waypoint.

2. $A_{2,k}$: "direct to the k_{th} waypoint from the current position according to the flight plan", for $k = 1, ..., 4$

3. $A_{3,k}$: "change the course k degrees", for $k \in \{\pm 10\}$ and after 5 minutes head towards the next waypoint.

Then, the following rule applies according to the resolution action chosen at any time instance:

Table A.1: Parameters of the GAIL algorithm.

parameter	value
max_kl	0.9
discount factor	1
discriminator's learning rate	0.0001
discriminator's epochs	200

If the predicted resolution action type A_i is correct according to the dataset, then the trajectory will evolve exactly as in the historical data.
Else, the trajectory will evolve according to a specific realization $A_{i,k}\ of\ A_i$ selected randomly among options.

Having applied any resolution action, all subsequent states generated are provided to the GAIL discriminator D whose task is to assess whether the conflict evolves according to the demonstrations, i.e. whether the distribution of the generated states is according to the distribution of the demonstrated states.

As an enhancement of GAIL, the policy model is augmented with the attention module presented in Section 4.7.2.

For the policy model this study uses a NN consisting of two dense layers with 100 nodes each with tanh activation. When the attention module is used, the output of the module is passed as input to the network. For the discriminator a NN is used, consisting of two dense layers with 100 nodes each with tanh activation. To update the policy this study uses the TRPO policy gradient algorithm. To update the discriminator the Adam optimization is used with learning rate 0.0001.

Table A.1 reports the parameters of the GAIL algorithm.

A.1.2 Experimental Results

Table A.2 presents results achieved by the IL algorithm GAIL, exploiting an attention mechanism (GAIL+att) and without attention (GAIL). Columns report the 95% confidence interval of precision, recall, f1-score and the MCC w.r.t. resolution action types of ATCOs. GAIL variants are trained for 1500 mini-batches.

Comparing GAIL+att to GAIL we observe that the attention module has a positive effect on the accuracy of the predictions: The mean value of the MCC and of the f1-score increase.

When compared with the classification methods presented in table 4.14 we observe that the GAIL+att method ranks 4th after the RF, GTB and NN+att methods. GAIL+att reports a MCC value of 0.43 for the testing dataset, mean f1-score 0.65 for resolution action type A_1, 0.72 for resolution action type A_2 and 0.43 for resolution action A_3.

Table A.2: Experimental results achieved by the IL algorithm GAIL, exploiting an attention mechanism (GAIL+att) and without attention (GAIL). Columns report the 95% confidence interval of precision, recall, f1-score and the MCC w.r.t. resolution action types of ATCOs.

Method	**Dataset**	**Precision**			**Recall**			**f1-Score**			**MCC**
		A_1	A_2	A_3	A_1	A_2	A_3	A_1	A_2	A_3	
GAIL+att	train	0.76 ± 0.02	0.79 ± 0.03	0.85 ± 0.05	0.78 ± 0.04	0.87 ± 0.02	0.38 ± 0.04	0.77 ± 0.02	0.82 ± 0.01	0.53 ± 0.04	0.61 ± 0.02
	test	0.66 ± 0.10	0.69 ± 0.05	0.62 ± 0.22	0.65 ± 0.08	0.76 ± 0.06	0.33 ± 0.09	0.65 ± 0.08	0.72 ± 0.04	0.43 ± 0.12	0.43 ± 0.07
GAIL	train	0.75 ± 0.02	0.75 ± 0.03	0.80 ± 0.07	0.70 ± 0.05	0.85 ± 0.02	0.46 ± 0.03	0.73 ± 0.03	0.79 ± 0.01	0.58 ± 0.03	0.56 ± 0.03
	test	0.62 ± 0.08	0.65 ± 0.05	0.62 ± 0.12	0.57 ± 0.11	0.76 ± 0.08	0.31 ± 0.11	0.59 ± 0.07	0.70 ± 0.02	0.41 ± 0.11	0.36 ± 0.06

Finally, although classification methods outperform the GAIL method implementing the single-stage IL problem formulation, it must be noted that, GAIL learns to predict the resolution action applied by the ATCOs, considering the effects of different resolution actions i.e. how the aircraft state evolves, learning also to discriminate between expert and non-expert states. This has the following benefits: a) GAIL can incorporate further trajectory optimization objectives in a straight forward way by augmenting its reward function with other terms i.e. added nautical miles, fuel consumed, CO_2 emissions, etc, b) models learned by GAIL can be exploited by reinforcement learning methods that aim to solve conflicts and evolve the trajectories according to demonstrated maneuvers. These are issues to be investigated in the future, also in comparison to the models learned by other methods, by incorporating these models in conflicts-free trajectory planning methods.

A.2 Description of data this study relies on

The methodology proposed in this study exploits data from the Spanish airspace, considering flights over Spain, without sacrificing the generality of the methods introduced. The data sources comprise:

- Surveillance data: operational quality data with actual flights (raw) trajectories (Spanish ATC Platform SACTA)
- Flight plan data: all flight plan updates for any given flight, since flight plan creation, allowing continuous snapshots (Spanish ATC Platform SACTA)
- Sector configuration data: the schedule of deployed sector configurations, as well as the catalog of possible sector configurations (Spanish ATC Platform SACTA)
- Weather data: weather forecast information regarding the area corresponding to the trajectories considered (provided by the NOAA platform).
- Aircraft identification data: provides specific information on the aircraft flying a particular trajectory. (World Aircraft Database and ICAO Doc8643)
- ATCO events: provides actions taken by the Air Traffic Controllers in order to ensure safety of flights (provided by ATON).

In the following sections we describe the datasets in more detail and also their spatial and temporal coverage.

A.2.1 Surveillance data

This data set provides radar tracks of the Spanish airspace controlled by the Spanish ATC provider ENAIRE. A radar track is reported in tabular form, with a timestamp key and geospatial information. Tracks are updated with an interval of 5 seconds. The spatial area coverage of the data is the whole Spanish airspace. The temporal coverage of the data includes the years 2016, 2017, 2018. For this study we have used radar tracks over the Iberian Peninsula for the year 2017. The AI/ML methods have been trained using trajectories between 5 different OD pairs Malaga (LEMG) - Gatwick (EGKK), Malaga (LEMG) - Amsterdam (EHAM), Lisbon (LPPT)-Paris (LFPO), Zurich (LSZH) - Lisbon (LPPT) and Geneva (LSGG) - Lisbon (LPPT). In addition, we consider only trajectories that have at least one ATCO resolution action corresponding to a detected conflict. This results to 668 trajectories from 2017.

In addition to these datasets, for the purposes of evaluating trajectory imitation methods we exploited (a) radar tracks between 3 OD pairs: Barcelona to Madrid (BCN-MAD) during July 2019 (308 trajectories), London Heathrow to Rome Fiumicino (LHR-FCO) during July 2019 (219 trajectories), and Helsinki to Lisbon (HEL-LIS) during July 2019 (44 trajectories).

A.2.2 Flight Plan data

The Flight Plan data set is essential for the aviation domain, as it contains information that triggers a lot of operational decisions, both in the planning and execution phases. The data source that provides the data is a subsystem of the Spanish ATC platform (GIPV, Flight Plan Information Management System). The GIPV is a Flight Plan Report Manager Subsystem that contains information about flight plans that are being flown or going to be flown soon (to 15 hours), in the part of the airspace that is being operated under the responsibility of the Flight Plan Central Treatment. The spatial area coverage of the data is the whole Spanish airspace. The temporal coverage of the data includes the years 2016, 2017, 2018. For this study we have used radar tracks over the Iberian Peninsula for the year 2017.

A.2.3 Sector Configuration Data

The Airspace data set describes the existing airspace organization, with no gaps or overlaps, and all the possible ways of combining volumes to generate different operational sector configurations. This data set describes the schedule of sector configurations that have been effectively put in place in Spanish airspace. The temporal coverage of the data includes the years 2016, 2017, 2018. For this study we have used radar tracks over the Iberian Peninsula for 2017.

A.2.4 Weather Data

This data set provides the forecast of the weather conditions, at the position of an aircraft at any given time during its flight. Specifically, for each 4D position (latitude, longitude, altitude and time) it reports the values of the weather variables describing the weather conditions at that position. The most frequently used variables in the aviation domain, are the Temperature, the Pressure, and the two horizontal components of the Wind Speed, u and v. The available data cover the Iberian Peninsula and Canary Islands for the whole 2016 and July 2019.

For the purposes of evaluating trajectory imitation methods we have used weather data obtained from NOAA for 2019.

Aircraft Identification and Models For the identification of aircraft reported in surveillance data set, the World Aircraft Database is exploited . This data set provides specific information on the aircraft flying a particular trajectory (thus enriching the information available in the surveillance and flight plans data sets).

A.2.5 ATCO Events Dataset

As ATCO events we consider regulations assigned by the air traffic controllers to flights, in order to ensure that the minimum separation minima are not violated, and thus, aircraft fly safely. An ATCO event contains information about the callsign of the regulated flight, the origin airport, the destination airport, the timestamp of the event, the type of the event and the sector in which the event took place. This dataset is in .csv format and contains regulations assigned by the Air Traffic Controllers to flights that pass over the Spanish Flight Information Region (FIR). It contains several types of events made by the controller from which we consider as relevant to the conflict resolution problem the following:

- Flight level clearance due to traffic
- Speed adjustment due to traffic
- Direct to waypoint clearance due to traffic

The spatial area coverage of the data is the whole Spanish airspace. The temporal coverage of the data includes the years 2017, 2018.

For this study we exploit ATCO events over the Iberian Peninsula for the year 2017.

A.3 Conjugate gradient method

Following [68] Appendinx C, in order to compute the step direction of TRPO $H^{-1} * g$ ([43] equation (4)) this study uses the truncated conjugate gradient method which executes 10 iterations of the conjugate gradient algorithm proposed in [35]. Algorithm 6 presents the conjugate gradient algorithm used.

Algorithm 6: Conjugate Gradient

Result: $x == A^{-1} * b$

Initialize x_0 arbitrarily.

$r_0 := b - Ax_0$

$p_0 := r_0$

while *j= 0,1,..., until convergence* **do**

  $\alpha_j := (r_j, r_j)/(Ap_j, p_j)$

  $x_{j+1} := x_j + \alpha_j p_j$

  $r_{j+1} := r_j - \alpha_j A p_j$

  $\beta_j := (r_{j+1}, r_{j+1})/(r_j, r_j)$

  $p_{j+1} := r_{j+1} + \beta_j p_j$