GATS: A Time-Series Dataset for Addressing General Aviation Flight Safety

Aidan LaBella [1 *] **Aditya Iyer** [1 *] **Charles Duong** [1 *] **Nathan DePiero** [1 *] **Pak Iong Long** [1 *] **Elise Carman** [1 *]
Randall Balestriero [1] **Travis Desell** [2]

Abstract

Assessing general aviation operations has become critical for improving the safety of airspace systems worldwide. Yet, machine learning research in the domain is nearly nonexistent due to the extremely limited amount of publicly available flight data. To encourage research in airspace safety and general aviation as a whole, we release GATS, a dataset comprising more than 7,000 flights anonymously sampled with permission from the privately held US National General Aviation Flight Information Database (NGAFID), corresponding to 10,641 total hours of data recordings. This dataset sets itself apart from previous works with its inclusion of 2 new aircraft types and 76 different flight data sensor parameters, including navigational information and aircraft orientation. We benchmark this dataset on 2 aviation-domain tasks. The first is aircraft classification, a proof-of-concept problem to establish that advanced machine learning methods can be applied effectively on time-series flight data. The second is missing data reconstruction, a more rigorous safety-critical task necessary in real-world environments where sensors can fail and information must be restored for flight analysis purposes. We achieved near-perfect accuracy on the aircraft classification task, but failed to generate meaningful reconstructions on the missing data task. The poor performance on the second task with the chosen models indicates the opportunity for future research into better techniques for understanding and improving flight safety using this dataset. [1]

1. Introduction

General aviation safety remains a critical concern as the National Airspace System (NAS) continues to grow in size and complexity. Between 2002 and 2022, the United States recorded 1,205 general aviation accidents, of which 214 were fatal (National Transportation Safety Board, 2024). These accidents result from a complex interaction of factors such as pilot error, mechanical failures, adverse weather, and aging aircraft. Improving safety in this domain requires more than basic sensor flags and post-flight debriefings. It calls for data-driven methods that can learn from extensive flight data and anticipate risks before they materialize.

Beyond accident prevention, modern machine learning has the potential to transform aviation more broadly. Detailed flight representations can enable predictive modeling for enhanced warnings during flights, detailed and tailored analytics for post-flight debriefing, custom and automated planning for air traffic control, and general improvement in the efficiency and safety of aviation operations.

However, there is a substantial lack of publicly available data sources for general aviation. Most flight data is collected by manufacturers, governments, or flight schools and is typically unavailable because it is proprietary or subject to pilot privacy concerns. For educational training flights, federal regulations such as the Family Educational Rights and Privacy Act (FERPA, 20 U.S.C. 1232g) requires that the identification information (e.g., operator name, location and aircraft tail number) is anonymized before any student-related data can be shared. This legal constraint, while essential for protecting privacy, further limits the availability of open datasets. The data that is available tends to be task or aircraft-specific, limiting generalizability. This scarcity has inhibited the development of machine learning for the domain.

To address this data shortage problem, we introduce **GATS**, an anonymous, FERPA-compliant **G**eneral **A**viation **T**ime-**S**eries dataset of 10,641 flight hours from 7,679 student pilot training flights across 97 propeller-driven aircraft on 3 different airframe types. Each flight in GATS includes various features that describe key aspects such as engine performance, aircraft orientation, environmental conditions, navigation information, and flight instrument readings. Together, these categories provide a holistic view of each flight and enable more precise analysis of aircraft behavior and conditions that lead to incidents.

In our study, we provide 2 aviation-relevant benchmark tasks that help establish the validity and complexity of this dataset:

[1] **Huggingface Dataset (Anonymous Author Version Link):** https://huggingface.co/datasets/NGAFID2025ICML/NGAFID-LOCI-GATS-Anonymous

aircraft classification and missing data reconstruction. Each task serves a different purpose. Aircraft classification is to help the audience understand how machine learning is applied on aviation data. Missing data reconstruction is a critical task that addresses sensor failures, which is a real-world condition that can negatively affect an aircraft's flight and accident analysis.

These experiments evaluated both self-supervised learning (SSL) and supervised learning techniques. The SSL models employed masked autoencoders and contrastive learning, both of which are well-suited to learning robust feature representations of a flight that can be used in flight analysis and other downstream tasks that do not pertain to specific labels. The supervised model utilized a deep neural network architecture (Yang et al., 2022; Xiao et al., 2020), which is effective for task-specific applications when labeled data are available. Together, these approaches offer a comprehensive assessment of how the dataset can be applied across different machine learning paradigms.

Our contributions can be summarized accordingly:

- We introduce GATS, a novel general aviation dataset consisting of thousands of hours of flight data across 3 aircraft platforms.
- We provide an anonymized, FERPA-compliant dataset in both raw and preprocessed versions, with documentation and implementation details to ensure interpretability and reproducibility.
- We build 2 SSL models based on contrastive learning and masked autoencoder techniques.
- We evaluate the dataset on 2 domain-relevant tasks using both the constructed SSL models and pre-existing supervised learning frameworks.

This dataset addresses the shortage of public data in the aviation domain, encouraging advancement in the aviation field and enabling a new domain to evaluate cutting-edge machine learning techniques on time-series data.

2. Related Works

2.1. National General Aviation Flight Information Database (NGAFID)

The NGAFID (LaBella et al., 2022) is a FAA-funded project where any aviator can upload and analyze their own flight data.

Previous works with NGAFID data mainly focus on identifying anomalous flight and aircraft performance conditions, as well as tasks like phase of flight and approach type classification. (LaBella et al., 2022) implemented a novel method for identifying dangerous flight conditions, such as Loss of Control In-Flight (LOC-I) (Balogh, 2017) and aerodynamic stalls (Administration, 2009). Using subject matter expert validation and a controlled test flight, this work showed that it was possible to calculate probabilistic metrics indicative of the risk that a flight will experience one of these dangerous conditions at any given timestep.

(Yang et al., 2022) leveraged mining aircraft logbook records (Akhbardeh et al., 2021) to create training data for aircraft predictive maintenance, namely the Cessna 172S airframe. This work utilized a convolutional multi-headed self-attention model (ConvMHSA) (Xiao et al., 2020) to output the probability that a flight occurred before or after maintenance. This work also produced an associated dataset (Yang & Desell, 2022) with 28 sensor parameters, unlike GATS, which contains 76 parameters. Moreover, this dataset only contains flight data from one aircraft type (the Cessna 172S), unlike GATS, which contains 3 different aircraft types, including the PA-28-181 and PA-44-180. Additionally, this dataset contains only aircraft engine parameters, which do not allow for the analysis of flight safety with regard to pilot inputs.

There have also been works that utilize NGAFID data for phase of flight and approach type identification. (Karboviak et al., 2018) developed a method to classify the type and quality of the approach of a flight in the NGAFID. (Lyu et al., 2024) utilized minimally-supervised learning and self-organized maps (MS-SOMs)(Kohonen, 1990) to predict the phase of flight with limited labeled data. Work by (Clachar, 2015) looked into using supervised and unsupervised approaches for finding atypical flight patterns in NGAFID data, i.e. anomaly detection.

2.2. Time-Series Data Problems

Classification Time-series classification (Ismail Fawaz et al., 2019) is niche in the machine learning community, especially when compared to the vast amount of literature on vision and language tasks. Current deep learning approaches include LSTM-RNNs (Karim et al., 2017; Kong et al., 2025), CNNs (Wang et al., 2017; Ismail Fawaz et al., 2020), and convolutional transformers (Yang et al., 2022). Many time-series classification tasks come from the medical community (Huang et al., 2024), for tasks such as ECG data classification (Gupta et al., 2024; Sakib et al., 2023). Another common example of time-series data found in the literature is related to stock market data, such as stock price classification (Kumari et al., 2024), risk analysis (Petchpol & Boongasame, 2025), and trading behavior analysis (Kong et al., 2020).

Forecasting Time-series forecasting (Lim & Zohren, 2021) has been a popular task for many applications, such

as coal power plant parameter prediction (Lyu et al., 2021), air quality forecasting (Anggraini et al., 2024), and weather dynamics (Zhang et al., 2024). Forecasting has the potential to be applied to NGAFID for tasks such as engine performance indicators, weather metrics (such as wind speeds), and pilot input readings. Such tasks have been accomplished before with evolutionary neural networks using NGAFID Cessna 172S engine data (Desell et al., 2014; 2020; ElSaid et al., 2021), including methods to increase explainability (Murphy et al., 2024).

Regression Regression tasks, such as missing column reconstruction, can be a very useful task for many downstream applications where missing and/or noisy sensor readings occur frequently. Past works have utilized electrical signals to reconstruct data for arc-fault detection (Jiang & Zheng, 2023), remote sensing (Zhou et al., 2023), and molecular biology (Laporte et al., 2024). Parameter regression has the potential to be used in NGAFID data because many parameters have missing readings. Other potential applications could include regressing parameters that are not recorded by the aircraft's flight data recorder, such as flap settings or rudder inputs.

Anomaly Detection Previous works have looked at transformer models with data from the physical sciences (Xu et al., 2021), graph-attention networks (Zhao et al., 2020), as well as autoencoders and RNN-based models (Delibasoglu & Heintz, 2024). Anomaly detection could be a useful tool for NGAFID data as it can identify potentially unsafe flying patterns that are not already found by classification or forecasting models.

While there are many existing tasks that utilize time-series data for classification, forecasting, regression, and anomaly detection, the NGAFID data stands out because it is subject to the external physical environment, as well as human inputs and mechanical processes. There exist very few other datasets that are dependent on these factors, making the NGAFID data unique.

3. A Novel General Aviation Dataset

GATS was extracted directly from the NGAFID, particularly for two US flight schools over a two-month period and consists of 10,641 flight hours from 7,679 flights spanning 3 airframe types. Only NGAFID developers and administrators are able to access all of its data, as individual users can only see their own data. In cooperation with the NGAFID and its stakeholders, we obtained permission to organize a subset of the NGAFID's data, with operator-identifying attributes removed for privacy.

The NGAFID's main feature, event detection, provides a set of 20 dangerous event definitions. The occurrences of these events are logged on a per flight basis.

3.1. Data Collection Methodology

The data collected for this work mostly consists of student pilot training and evaluation flights. Due to FERPA, any identifying information such as operator name, location and/or tail number is anonymized. Additionally, this data was pulled directly from the NGAFID resulting in extra columns, such as `Stall Index` and `Density Ratio`, which are calculated based on raw sensor readings. These extra columns are used by the LOC-I and Stall Index event calculations (LaBella et al., 2022; Balogh, 2017) to derive probabilistic metrics that indicate the likelihood that the event will occur.

3.2. Dataset Statistics and Analysis

Aircraft Types As shown in Appendix B.2, this dataset is comprised of three aircraft types: Cessna 172S, PA-28-181, and PA-44-180, depicted in Appendix B.1. While the Cessna 172S is high wing, both PA-28-181 and PA-44-180 are low wing. Additionally, Cessna 172S and PA-28-181 have a single engine while PA-44-180 has two. Eleven of the columns in the raw column set correspond to the PA-44-180's second engine.

Potentially Dangerous In-Flight Events The LOC-I events in this dataset correspond to special incidents within a flight. Our dataset comprises 8,802 events occurrences of the 20 event types listed in Appendix G.1. Additionally, Appendix G.4 shows the distribution of the number of events per flight. Of the 7,679 flights, only 2,616 experience one or more events and only 1,525 experience two or more.

Appendix G.6 shows the most prevalent event types across the dataset. The foremost being Low Airspeed on Climbout, which occurs in just over 1,250 of the 2,616 eventful flights. This event describes aircraft takeoffs with unexpectedly low airspeed. Another frequent event is High Altitude Stalls, which describe the occurrence of an aircraft stall above a threshold altitude. These occur in around 1,100 flights. Overall, these events are more likely to occur in GATS due to the dataset comprising of only training flights. Specifically, in training, student pilots practice many techniques including High Altitude Stalls, thus leading to higher frequency counts. The remaining event types are half as common, occurring in 500 flights or less.

Appendix G.5 examines the number of occurrences of high altitude stall events in flights. Unlike most other events which occur at most 6 times throughout a flight, high altitude stalls can occur at most 47 times.

Flight lengths The flights in the dataset have a variety of durations, ranging from 3 minutes to 9 hours. The distribution of these flight lengths is pictured, in Figure 3.

NaN Values The columns in the dataset represent different sensors on the aircraft and their readings. These may contain NaN values because of sensor failures at certain timesteps, or from missing data fields for specific aircraft types. Overall, the dataset contains 2.49% percent NaN values. These NaN values are concentrated in a subset of the columns, as seen in Appendix H.

Flight Count per Column Because different aircraft are equipped with different sensor configurations, the set of recorded columns varies across flights and is not standardized. The occurrences of columns by number of flights is pictured in Appendix F.2.

3.3. Preprocessing

Because of the previously described aspects of the dataset, we provide both a preprocessing script and preprocessed version of the dataset. This allows for a set of flights with uniform columns and length without any NaN values. A more detailed preprocessing methodology can be found in Appendix C.

3.4. Models and Benchmarks

To demonstrate the practical value and versatility of this dataset, we establish two critical benchmark tasks in aviation: airframe classification and masked column regression. The first benchmark assesses whether the dataset supports simple yet meaningful classification problems. Airframe classification is divided into two subtasks—airframe type classification, which identifies the specific aircraft platform among three possible types, and airframe class classification, which distinguishes between single-engine and multi-engine aircraft. These tasks reflect core capabilities needed for aircraft identification in real-world applications, such as maintenance scheduling, training simulations, and automated air traffic control systems. The second benchmark, masked column regression, focuses on reconstructing missing sensor data—a frequent and high-stakes problem in aviation where sensor failures can impair both post-flight analysis and in-flight decision-making. Accurate reconstruction of missing values supports flight understanding and enhances the reliability of downstream safety systems. Details about setup, models, and results can be found in Appendix D.

4. Limitations

This dataset provides a valuable interface for machine learning researchers to study general aviation flight data and have a direct impact on flight safety analysis. However, the dataset itself includes limitations that future users must be aware of. The most prominent are the derived columns. As stated in Section D.2, several feature columns in this dataset are calculated from other feature columns. For instance, simple angle of attack (`AOASimple`) is directly calculated from columns such as pitch (`pitch`). Thus, machine learning models could shortcut learning the underlying representation of the data and can instead learn the simple correspondences. A full table of the correspondences can be viewed in Appendix E. A second limitation of this data is the large number of NaN values in the feature columns. Because this is real-world flight data, sensors may not always be operational, and so we invite future researchers to use their own preprocessing methods to fill these values in more advanced or appropriate ways in their own works. Another limitation of the dataset was that certain sensor fields are inherently different between aircraft. Lastly, a limitation is this was general aviation flight training data and not commercial aviation data. This means that this dataset applies to researchers studying general aviation problems, with the caveat that this data applies more directly to student pilots, rather than trained aviators.

5. Conclusions and Future Work

This paper addresses the lack of publicly available aviation data by introducing GATS, a novel and expansive time-series dataset of FERPA-compliant real-world flight recordings. We evaluate the data on two benchmark tasks, showcasing that machine learning is a valid method for analyzing this data, and providing an example task that would have a direct impact on improving flight safety.

We identify the aircraft classification tasks to be a simple problem, and the performances for both supervised and self-supervised models tested were very high for aircraft type classification, with accuracy divergence only happening in aircraft class classification. These results indicate the compatibility of the GATS dataset in machine learning frameworks. On the other hand, we observe that the missing data reconstruction task is challenging for both the self-supervised learning models tested, regardless of the training parameters (varying mask lengths and ratios). While the masked autoencoder method performed better than the contrastive technique, they both ultimately failed to accurately restore the data. Thus, it remains unclear what the optimal model solution is for this regression task, and future research efforts should be made to solve this problem with alternative architectures. A few potential methods that researchers could experiment with include attention (Vaswani et al., 2023), joint-embedding based architectures (Assran et al., 2023; Bardes et al., 2024), and non-contrastive SSL techniques (Oquab et al., 2024).

We encourage researchers to explore the GATS dataset as a benchmark for developing and evaluating machine learning methods on real-world, safety-critical time-series data.

References

Administration, U. S. F. A. *Pilot's Handbook of Aeronautical Knowledge*. Skyhorse Publishing Inc., 2009.

Ahunt. 1957 cessna 172 skyhawk photograph. https://commons.wikimedia.org/wiki/File:Cessna172Skyhawk1957model01.jpg, 2005. Public domain.

Akhbardeh, F., Alm, C. O., Zampieri, M., and Desell, T. Handling extreme class imbalance in technical logbook datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4034–4045, 2021.

Anggraini, L., Noersasongko, E., Purwanto, and Marjuni, A. Evaluation of time series forecasting techniques for air quality prediction: Case study of no2 levels. *2024 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp. 522–527, 2024. URL https://api.semanticscholar.org/CorpusID:274372458.

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., and Ballas, N. Self-supervised learning from images with a joint-embedding predictive architecture, 2023. URL https://arxiv.org/abs/2301.08243.

Balogh, S. Use of foqa data to estimate the probability of vehicle upset or loss of control in-flight. Master's thesis, University of North Dakota, Grand Forks, ND, USA, 2017.

Bardes, A., Garrido, Q., Ponce, J., Chen, X., Rabbat, M., LeCun, Y., Assran, M., and Ballas, N. Revisiting feature prediction for learning visual representations from video, 2024. URL https://arxiv.org/abs/2404.08471.

Bordes, F., Balestriero, R., Garrido, Q., Bardes, A., and Vincent, P. Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning. *arXiv preprint arXiv:2206.13378*, 2022.

Burdett, M. Piper pa-28-180 cherokee c over northrepps. https://commons.wikimedia.org/wiki/File:G-AVRZ_Piper_PA-28-180_Cherokee_C_at_Northrepps.jpg, 2021. Creative Commons Attribution–ShareAlike 2.0 Generic (CC-BY-SA 2.0).

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Clachar, S. A. Identifying and analyzing atypical flights by using supervised and unsupervised approaches. *Transportation research record*, 2471(1):10–18, 2015.

Delibasoglu, I. and Heintz, F. Time series anomaly detection leveraging mse feedback with autoencoder and rnn. In *Time*, 2024. URL https://api.semanticscholar.org/CorpusID:273506497.

Desell, T., Clachar, S., Higgins, J., and Wild, B. Evolving neural network weights for time-series prediction of general aviation flight data. In *Parallel Problem Solving from Nature–PPSN XIII: 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings 13*, pp. 771–781. Springer, 2014.

Desell, T., ElSaid, A., and Ororbia, A. G. An empirical exploration of deep recurrent connections using neuroevolution. In *Applications of Evolutionary Computation: 23rd European Conference, EvoApplications 2020, Held as Part of EvoStar 2020, Seville, Spain, April 15–17, 2020, Proceedings 23*, pp. 546–561. Springer, 2020.

ElSaid, A., Karns, J., Lyu, Z., Ororbia, A. G., and Desell, T. Continuous ant-based neural topology search. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pp. 291–306. Springer, 2021.

Feichtenhofer, C., Fan, H., Li, Y., and He, K. Masked autoencoders as spatiotemporal learners, 2022. URL https://arxiv.org/abs/2205.09113.

Gupta, P., Murugan, A. A., Chordia, D., Yannam, P. K. R., and Gupta, M. A comparative study of few-shot learning methods for 1-d ecg time-series classification. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, pp. 604–611. IEEE, 2024.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Huang, N., Wang, H., He, Z., Zitnik, M., and Zhang, X. Repurposing Foundation Model for Generalizable Medical Time Series Classification, October 2024. URL http://arxiv.org/abs/2410.03794. arXiv:2410.03794 [cs].

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, March 2019. ISSN 1573-756X. doi: 10.1007/s10618-019-00619-1. URL http://dx.doi.org/10.1007/s10618-019-00619-1.

Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

Jiang, R. and Zheng, Y. Series arc fault detection using regular signals and time-series reconstruction. *IEEE Transactions on Industrial Electronics*, 70:2026–2036, 2023. URL https://api.semanticscholar.org/CorpusID:248142366.

Karboviak, K., Clachar, S., Desell, T., Dusenbury, M., Hedrick, W., Higgins, J., Walberg, J., and Wild, B. Classifying aircraft approach type in the national general aviation flight information database. In *Computational Science–ICCS 2018: 18th International Conference, Wuxi, China, June 11–13, 2018, Proceedings, Part I 18*, pp. 456–469. Springer, 2018.

Karim, F., Majumdar, S., Darabi, H., and Chen, S. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669, 2017.

KGG1951. Piper PA–44 seminole (lz-fto) over bulgaria. https://commons.wikimedia.org/wiki/File:Piper-pa-44.jpg, 2010. Image licensed under CC BY-SA 3.0.

Kohonen, T. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

Kong, A., Azencott, R., and Zhu, H. Pattern recognition in trading behaviors before stock price jumps: new method based on multivariate time series classification. *arXiv: Statistical Finance*, 2020. URL https://api.semanticscholar.org/CorpusID:226290158.

Kong, Y., Wang, Z., Nie, Y., Zhou, T., Zohren, S., Liang, Y., Sun, P., and Wen, Q. Unlocking the power of lstm for long term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11968–11976, 2025.

Kumari, B. A., Manju, K., Amrutha, M., Amrutha, M., and Neeraja, C. Time series data classification for precise stock market price prediction using ml. *2024 International Conference on Integrated Circuits and Communication Systems (ICICACS)*, pp. 1–6, 2024. URL https://api.semanticscholar.org/CorpusID:269241138.

LaBella, A. P., Karns, J. A., Akhbardeh, F., Desell, T., Walton, A. J., Morgan, Z., Wild, B., and Dusenbury, M. Optimized flight safety event detection in the national general aviation flight information database. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 1570–1579, 2022.

Laporte, M. H., Gambarotto, D., Bertiaux, E., Bournonville, L., Louvel, V., Nunes, J. M., Borgers, S., Hamel, V., and Guichard, P. Time-series reconstruction of the molecular architecture of human centriole assembly. *Cell*, 187:2158 – 2174.e19, 2024. URL https://api.semanticscholar.org/CorpusID:269034316.

Lim, B. and Zohren, S. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Lyu, Z., Patwardhan, S., Stadem, D., Langfeld, J., Benson, S., Thoelke, S., and Desell, T. Neuroevolution of recurrent neural networks for time series forecasting of coal-fired power plant operating parameters. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1735–1743, 2021.

Lyu, Z., Thapa, P., and Desell, T. Minimally supervised topological projections of self-organizing maps for phase of flight identification. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2024.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-SNE. 9(86):2579–2605. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Murphy, J., Kar, D., Karns, J., and Desell, T. Exa-gp: Unifying graph-based genetic programming and neuroevolution for explainable time series forecasting. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 523–526, 2024.

National Transportation Safety Board. Civil aviation dashboard, 2024. URL https://www.ntsb.gov/safety/StatisticalReviews/Pages/CivilAviationDashboard.aspx. Accessed: 2024-07-02.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma,

V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Petchpol, K. and Boongasame, L. Enhancing predictive capabilities for identifying at-risk stocks using multivariate time-series classification: A case study of the thai stock market. *Applied Computational Intelligence and Soft Computing*, 2025. URL https://api.semanticscholar.org/CorpusID:278320898.

Robertson, S. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

Sakib, M., Mustajab, S., and Siddiqui, T. Deep learning-based heartbeat classification of 12-lead ecg time series signal. In *2023 4th International Conference on Data Analytics for Business and Industry (ICDABI)*, pp. 273–278. IEEE, 2023.

Silva, T. S. Exploring simclr: A simple framework for contrastive learning of visual representations. *https://sthalles.github.io*, 2020. URL https://sthalles.github.io/simple-self-supervised-learning/.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Wang, Z., Yan, W., and Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585. IEEE, 2017.

Xiao, X., Zhang, D., Hu, G., Jiang, Y., and Xia, S. Cnn–mhsa: A convolutional neural network and multi-head self-attention combined approach for detecting phishing websites. *Neural Networks*, 125:303–312, 2020.

Xu, J., Wu, H., Wang, J., and Long, M. Anomaly transformer: Time series anomaly detection with association discrepancy. *ArXiv*, abs/2110.02642, 2021. URL https://api.semanticscholar.org/CorpusID:238408395.

Yang, H. and Desell, T. A large-scale annotated multivariate time series aviation maintenance dataset from the ngafid. *arXiv preprint arXiv:2210.07317*, 2022.

Yang, H., LaBella, A., and Desell, T. Predictive maintenance for general aviation using convolutional transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 12636–12642, 2022.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., and Eickhoff, C. A transformer-based framework for multivariate time series representation learning, 2020. URL https://arxiv.org/abs/2010.02803.

Zhang, W., Huang, J., Wang, R., Wei, C., Huang, W., and Qiao, Y. Integration of mamba and transformer - mat for long-short range time series forecasting with application to weather dynamics. *2024 International Conference on Electrical, Communication and Computer Engineering (ICECCE)*, pp. 1–6, 2024. URL https://api.semanticscholar.org/CorpusID:272653578.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., and Zhang, Q. Multivariate time-series anomaly detection via graph attention network. *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 841–850, 2020. URL https://api.semanticscholar.org/CorpusID:221507538.

Zhou, J., Menenti, M., Jia, L., Gao, B., Zhao, F., Cui, Y., Xiong, X., Liu, X., and Li, D. A scalable software package for time series reconstruction of remote sensing datasets on the google earth engine platform. *International Journal of Digital Earth*, 16:988 – 1007, 2023. URL https://api.semanticscholar.org/CorpusID:257712743.

A. Licensing

- The GATS dataset is licensed under CC-BY 4.0.
- The images of the aircraft types in Appendix B.1 are licensed as follows:
  - Cessna 172S — public domain
  - Piper PA-28 — CC-BY-SA 2.0
  - Piper PA-44 — CC-BY-SA 3.0
- This paper is licensed under CC-BY-SA 4.0.
- The benchmark code is licensed under the MIT License.

B. Aircraft Types

B.1. Aircraft Type Photos


Figure 1. **Airframe Types in Dataset (left to right)**: Cessna 172 Skyhawk, Piper PA-28 Cherokee, Piper PA-44 Seminole*. Cessna 172 and Piper PA-28 are both single engine aircrafts, while the Piper PA-44 is a multi-engine aircraft. *1957 Cessna 172 Skyhawk. Photo © Ahunt — Public domain(Ahunt, 2005). No changes made. Piper PA–28–180 Cherokee C (reg. G-AVRZ) at Northrepps. Photo © Mike Burdett, licensed CC-BY-SA 2.0 (Burdett, 2021). *No changes made.* Piper PA–44 Seminole (LZ-FTO) over Bulgaria. Photo © KGG1951 — licensed CC-BY-SA 3.0 (KGG1951, 2010). *No changes made.*

B.2. Aircraft Types Table

Table 1. Aircraft Types

Aircraft Type	Count
Cessna 172S	4481
PA-28-181	2214
PA-44-180	984

C. Preprocessing Methodology

To address the difference in flight lengths described in Section 3.2, we chose a fixed length of 9,995 rows (seconds). We dropped flights that were less than 1,000 rows and greater than 9,995 rows. We then padded the remaining flights to 9,995 rows by filling in the padding to be the last occurring value for each column. Functionality to perform this for arbitrary lengths and filling techniques are contained in the preprocessing utilities. We dropped the short flights ($< 1,000$) because the padding technique would make the flight mostly noise, or the flight itself never got off the ground (parking or taxiing). We dropped the long flights ($> 9,995$) because truncating the flight would remove the landing of the flight, effectively disrupting the representation of the flight as a whole.

To address the presence of NaN values described in Section 3.2, we replaced these NaN values with the most recent occurring value in the column. We also provide functionality for this and replacing with 0's in the preprocessing utilities.

To address the inconsistent feature sets described in Section 3.2, we picked a subset of columns that covered most of the flights as indicated in Appendix F.2. This resulted in the column set as seen in Appendix F.1. We dropped all of the flights that did not have this column set, and for the flights that did, we only selected those columns. This means that for the multi-engine PA-44-180, any parameter to do with the second engine, such as engine 2 RPM (`e2rpm`), was dropped. This effectively made the multi-engine aircraft resemble a single-engine aircraft, but maintaining certain multi-engine performance capabilities according to the remaining data columns from the flight. For instance, fuel quantity level (`fqtyl`), a column that persisted after dropping, in the multi-engine aircraft is much higher (> 45 gallons at peak) than compared to a single-engine aircraft (≈ 27 gallons at peak). This limitation is one that can be improved upon with alternative preprocessing schemes by future users of the dataset.

D. Models and Benchmarks

To evaluate performance on these benchmarks, we train three model architectures: a self-supervised contrastive learning model, a self-supervised masked autoencoder, and a supervised ConvMHSA model. All models are trained on the same data splits of 70% training, 15% validation, and 15% testing, following the preprocessing steps described in Section 3.3. For the aircraft classification task, we use accuracy as the performance metric. In the missing data reconstruction task, we use Mean Squared Error (MSE) and Mean Absolute Error (MAE). While these models serve as strong baselines, we invite the research community to explore and benchmark alternative approaches using this dataset.

D.1. Model Descriptions

SSL Contrastive Architecture Contrastive representation learning is a self-supervised learning technique that aims to learn representations by contrasting positive and negative pairs as data samples. The central idea in contrastive learning is to maximize agreement between positive pairs which represent similar examples while minimizing agreement with negative pairs which are distinct examples.

To adapt the contrastive learning to time-series data, we use an event-occurrence based scoring metric described in Appendix K to generate positive and negative pairs. We assign positive pairs as the examples with the most similar scores, and the negative pairs as all others.

Our contrastive learning architecture, built on top of SimCLR (Silva, 2020), uses ResNet-18 (He et al., 2015) as the backbone. We append a two-layer projection head to this backbone. This is composed of a linear layer, a ReLU activation, and another linear layer, which maps the high-dimensional latent embeddings to a lower-dimensional space for contrastive learning.

Our training process builds on the SimCLR framework introduced by Chen et al. (Chen et al., 2020), employing the NT-Xent variant of the Noise Contrastive Estimation (NCE) loss to maximize similarity between positive pairs and minimize it between negative pairs. We maintain a diverse pool of negative samples by leveraging large batch sizes. This ensures that our learned representations are able to capture temporal and contextual patterns that are essential to general aviation data.

After training, we used guillotine regularization (Bordes et al., 2022) to ensure the model's ability to generalize. Next, we ran time-series flights through our trained model to extract embeddings. We normalized the embeddings using standard scaling to ensure uniform feature contributions and then use dimensionality reduction using t-Distributed Stochastic Neighbor Embedding (t-SNE) (Maaten & Hinton) to evaluate the quality of the representations and investigate their structure.

For further evaluation, we labeled the extracted embeddings with additional metadata, such as flight safety scores and aircraft types, shown in Appendix L. This enabled clustering analysis to determine whether similar flights or types of aircraft were grouped effectively in the representation space. We gathered visualizations using scatter plots, where our embeddings were labeled using metadata attributes.

SSL Masked Autoencoder For the missing data reconstruction task, we first use a self-supervised masked autoencoder as the baseline method. Masked autoencoders (Feichtenhofer et al., 2022) provide a strong baseline because they are trained to minimize reconstruction error, enabling them to learn compact representations of temporal patterns and effectively capture dependencies in time-series data like flight trajectories.

The masked autoencoder architecture consists of two primary components: an encoder and a decoder. The encoder is a bidirectional LSTM (Hochreiter & Schmidhuber, 1997) that transforms the input features into a hidden representation. Due to its bidirectional nature which processes the sequence in both forward and backward directions, the encoded output has twice the dimensionality of the hidden layer. The decoder, also an LSTM layer, takes this encoded representation and maps

it back to the original input dimension, aiming to reconstruct the unmasked, original input sequence. We considered using other architectures such as transformers (Vaswani et al., 2023), but we believe that with the limited amount of data and high computation cost of attention (9,995 time steps for each flight), our results would not be ideal.

During training, flight data is first normalized by computing the mean and standard deviation for each column to ensure consistent scaling across samples. We modify the noise mask originally introduced by Zerveas et al. (Zerveas et al., 2020), which randomly masks parts of the input according to a specified mask ratio. The masked input is fed into the model, while the unmasked original sequence serves as the reconstruction target. The model is trained using MSE loss to measure the difference between the reconstructed and original data, and optimized using the Adam algorithm over multiple epochs.

Supervised ConvMHSA The design choices for this supervised model are discussed in Section 2.1. This model serves as a primary comparison tool to the self-supervised contrastive framework. We conduct two main tasks: airframe model classification and airframe type classification. For airframe model classification, we train on the three airframe types included in this dataset: Cessna 172, PA-44-180 and PA-28-181. For airframe class classification, we split these three aircraft into two classes based on the number of engines they have. The Cessna 172 and PA-28 fall into the single-engine aircraft class, while the sole occupant of the multi-engine class is the PA-44-180.

D.2. Benchmark Tasks

Aircraft Classification For the aircraft classification task, we first use the supervised ConvMHSA. We use the dataset's default test/train/val split, and train the model with the default preprocessed data where each flight is one training instance.

Table 2. **Aircraft Classification Test Set Accuracy:** These accuracy scores indicate that the pure supervised trained model is generally better at classification tasks versus a contrastive SSL + supervised head. Further testing with alternative and larger models can be used to understand why SSL is not performing well, but given the high accuracy of supervised models, it is clear that this task is easily solvable.

Task	Accuracy
ConvMHSA Airframe Model Classification	99%
ConvMHSA Airframe Class Classification	100%
SimCLR Airframe Model Classification	82%
SimCLR Airframe Class Classification	30%

The supervised model was then compared against the contrastive learning self-supervised model. For the aircraft model classification subproblem, a 1 layer 3-classification head was appended to the end of the frozen self-supervised trained ResNet. The extended model was then trained and validated with the training and validation data in a supervised manner. For the airframe class classifier, the same training, validation, and testing methodology as the airframe type classifier was taken, with the inclusion of a binary linear classification layer rather than a linear 3-class head. After training both models, they were each tested on the test set and achieved high accuracy in the airframe type classifier, with the supervised model being near-perfect. They differed in the airframe class classification as the self-supervised model had very poor accuracy compared to the supervised model, as seen in Table 2.

The most likely explanation for aircraft model classification achieving very high accuracy in both models is that the task is too simple, and there are interdependencies among the feature columns and unique qualities in the dataset that allow large neural networks to learn the distribution quickly. For instance, in the multi-engine aircraft, true airspeeds (`TAS`) can exceed over 170 knots ($\approx$195 mph), whereas the single-engine aircraft can only get up to about 120 knots ($\approx$140 mph), due to the performance capabilities of aircraft with more engines. Thus, if the model learns that speeds over a certain limit belong to multi-engine aircraft, this becomes a shortcut for future unseen data.

A possible reason for why the aircraft class classification accuracy is much poorer for the self-supervised model is due to the removal of any engine 2 parameters for the preprocessed dataset, as described in Section 3.3. After performing a t-SNE visualization (Maaten & Hinton) for clustering the representations viewed from the ResNet backbone in Appendix L.2, it is clear that the multi-engine PA-44-180 and the single-engine PA-28 are being clustered together. Thus, even if a supervised learning head is appended to the self-supervised backbone, the model might not be able to separate between the PA-44-180 and PA-28-181, leading to inaccurate classification. Additionally, the pure supervised method might not suffer from this vulnerability due to it having a different backbone architecture and having the labels from the very beginning of the training rather than just a single layer at the end, such as the self-supervised method.

Missing Data Reconstruction For the missing data reconstruction task, the masked autoencoder model was compared against the contrastive learning self-supervised model. To orient the contrastive learning model to this task, we froze the ResNet weights and appended a single layer regression head to the end of the model. This meant that the regression head was the same output dimension as the input flight. During training, masked flights would be passed through the model and the output would be compared with the unmasked flight using MSE to calculate loss. We also followed the same training process for both self-supervised frameworks. We conducted experiments using six different combinations of masking ratio and mean mask length to train each model, as seen in Appendix I.

During testing, the same performance metrics as the masked autoencoder would be determined on the contrastive model with the test set: MAE and MSE. Both models were measured on normalized values to collect the evaluation metrics. Additionally, we observed the inference results of a random mask with a masking ratio of 0.5 and a mean mask length of 60 was applied to all test data. The results, shown in Table 3, indicate that the autoencoder model had some success with reconstructing the values, whereas the self-supervised model performed significantly worse.

As an example of the results, in Appendix M, we plot the original and reconstructed values for aircraft pitch across the three different aircraft types for the masked autoencoder model. Pitch was chosen as it is a critical parameter for an aircraft that directly impacts the safety of flight. As seen in these plots, the masked autoencoder is able to reconstruct some of the missing values, but still does not completely restore the original data.

It is important to note that the data used to train and test the model contain columns that are derived from each other, as seen in Appendix E. This is an issue because we could be masking a column that is dependent on another. Therefore, the model could find a shortcut and learn the dependency, and thus reconstruct unmasked values trivially by applying such a dependency. Nevertheless, the poor performance on this task indicates the potential for other models to be applied.

Table 3. **Reconstruction Model Comparison:** This table shows the MAE and MSE for the Masked Autoencoder and Contrastive + Regression Head models. Clearly, the contrastive framework performs much worse than the autoencoder, indicating that our contrastive setup may not provide a meaningful representation for downstream regression tasks.

Model	MAE	MSE
Masked Autoencoder	0.46	0.62
Contrastive + Regression Head	4.44	25.50

E. Aircraft Sensor Information

Column	Definition	Derived?	Depends On
afcson	Autopilot On/Off	no	N/A
altagl	Altitude above ground level	no	N/A
altb	Barometric altitude	no	N/A
altgps	Altitude from GPS	no	N/A
altind	Indicated altitude	no	N/A
altmsl	Altitude above mean sea level	no	N/A
altmsllagdiff	Change in altmsl between consecutive time points	yes	altmsl
amp1	Ammeter on the main battery (+ charging, - discharging)	no	N/A
amp2	Ammeter on the standby battery (+ charging, - discharging)	no	N/A
aoasimple	Simplified angle of attack	yes	Pitch, VSpd, IAS, OAT, BaroA
baroa	Barometric altimeter	no	N/A
cas	Calibrated airspeed	no	N/A
com1	Radio 1 frequency	no	N/A
com2	Radio 2 frequency	no	N/A
coordinationindex	Coordination index: measures yaw misalignment from heading	yes	Roll, IAS, VSpd, OAT, BaroA, HDG

crs	Course	no	N/A
densityratio	Ratio of actual air density to standard air density	yes	BaroA, OAT
e1cht1	Engine 1, cylinder 1 head temperature	no	N/A
e1cht2	Engine 1, cylinder 2 head temperature	no	N/A
e1cht3	Engine 1, cylinder 3 head temperature	no	N/A
e1cht4	Engine 1, cylinder 4 head temperature	no	N/A
e1chtdivergence	Engine 1 cylinder head temperature divergence	no	N/A
e1egt1	Engine 1, cylinder 1 exhaust gas temperature	no	N/A
e1egt2	Engine 1, cylinder 2 exhaust gas temperature	no	N/A
e1egt3	Engine 1, cylinder 3 exhaust gas temperature	no	N/A
e1egt4	Engine 1, cylinder 4 exhaust gas temperature	no	N/A
e1egtdivergence	Engine 1 exhaust gas temperature divergence	no	N/A
e1fflow	Engine 1 fuel flow rate	no	N/A
e1map	Engine 1 manifold air pressure	no	N/A
e1oilp	Engine 1 oil pressure	no	N/A
e1oilt	Engine 1 oil temperature	no	N/A
e1rpm	Engine 1 revolutions per minute	no	N/A
e2cht1	Engine 2, cylinder 1 head temperature	no	N/A
e2egt1	Engine 2, cylinder 1 exhaust gas temperature	no	N/A
e2egt2	Engine 2, cylinder 2 exhaust gas temperature	no	N/A
e2egt3	Engine 2, cylinder 3 exhaust gas temperature	no	N/A
e2egt4	Engine 2, cylinder 4 exhaust gas temperature	no	N/A
e2egtdivergence	Engine 2 exhaust gas temperature divergence	no	N/A
e2fflow	Engine 2 fuel flow	no	N/A
e2map	Engine 2 manifold absolute pressure	no	N/A
e2oilp	Engine 2 oil pressure	no	N/A
e2oilt	Engine 2 oil temperature	no	N/A
e2rpm	Engine 2 revolutions per minute	no	N/A
fqtyl	Fuel quantity (left tank)	no	N/A
fqtyr	Fuel quantity (right tank)	no	N/A
gndspd	Ground speed	no	N/A
hal	Horizontal alert limit	no	N/A
hcdi	Horizontal course deviation indicator	no	N/A
hdg	Heading	no	N/A
hplfd	Horizontal protection level	no	N/A
hplwas	Previous horizontal protection level	no	N/A
ias	Indicated airspeed	no	N/A
latac	Lateral acceleration	no	N/A
loc-iindex	Loss of control index	yes	Pitch, VSpd, IAS, OAT, BaroA, HDG, Roll

magvar	Magnetic variation	no	N/A
normac	Normal acceleration	no	N/A
oat	Outside air temperature	no	N/A
pitch	Pitch	no	N/A
pichc	Pitch Command	no	N/A
roll	Roll	no	N/A
rollc	Roll Command	no	N/A
stallindex	Stall index	yes	Pitch, VSpd, IAS, OAT, BaroA
tas	True airspeed	no	N/A
totalfuel	Total fuel	no	N/A
trk	Track	no	N/A
trueairspeed(ft/min)	True airspeed (in feet per minute)	yes	IAS, BaroA, OAT
val	Voltage validity	no	N/A
vcdi	Vertical course deviation indicator	no	N/A
volt1	Main bus voltage (alternators and main battery)	no	N/A
volt2	Essential bus voltage (standby battery)	no	N/A
vplwas	Previous vertical protection level	no	N/A
vspd	Vertical speed	no	N/A
vspdcalculated	Vertical speed derived from barometric altitude	yes	AltB
vspdg	Vertical speed guide	no	N/A
wnddr	Wind direction	no	N/A
wndspd	Wind speed	no	N/A

F. Preprocessed Column Set

F.1. List of columns from preprocessing

vspdg
e1egtdivergence
crs
vspdcalculated
trk
normac
altmsl
vspd
oat
vplwas
hplwas
baroa
wnddr
e1oilp
ias
latac
e1egt1
densityratio
e1oilt
altmsllagdiff
pitch
pichc
rollc
tas
fqtyr
totalfuel
trueairspeed(ft/min)
hplfd
magvar
wndspd
e1egt2
altgps
amp1
fqtyl
volt1
e1fflow
altagl
altb
roll
stallindex
e1egt3
e1rpm
e1egt4
hdg
aoasimple
gndspd

F.2. Flight Count per Column

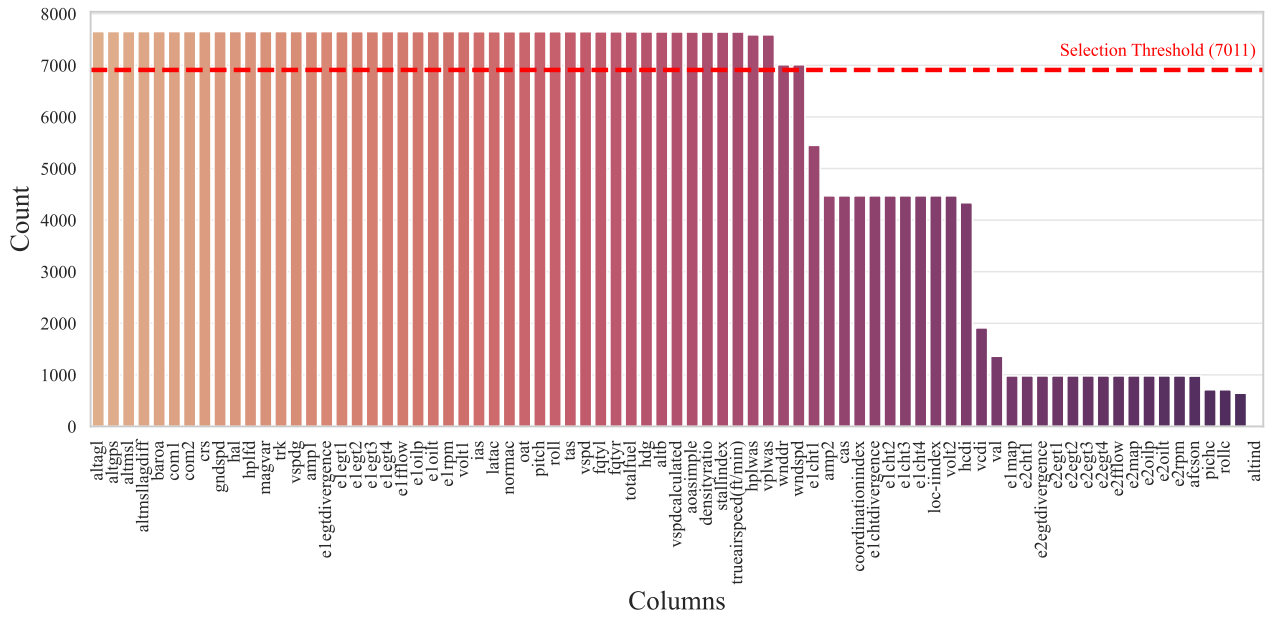

Figure 2. **Flight Count per Column:** The selection threshold depicted above guided our column set selection. Columns utilized by fewer than 7000 flights were dropped from consideration. Many of the columns with the smallest flight counts correspond to second engine sensors. Flights operated with 2 engine aircrafts only constitute 9% of our dataset.

G. Dangerous Events

G.1. Event Type Set

No.	Event Type
1	High Lateral Acceleration
2	Low Fuel
3	Cylinder Head Temperature
4	Low Altitude Stall
5	Low Pitch
6	Low Airspeed on Climbout
7	VSI on Final
8	Low Airspeed on Approach
9	Roll
10	Engine Shutdown Below 3000 Ft
11	Low Oil Pressure
12	Airspeed
13	High Altitude Stall
14	Low Altitude Spin
15	High Altitude Spin
16	Proximity
17	High Pitch
18	Altitude
19	High Vertical Acceleration
20	Low Ending Fuel

Table 5. List of Different Event Types

G.2. Event Definitions

Table 6: Event Definitions from NGAFID

Event Name	Aircraft Type	Event Definition
Airspeed	Cessna 172S	An Airspeed event occurs when (IAS $>$ 154) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Airspeed	PA-28-181	An Airspeed event occurs when (IAS $>$ 163) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Airspeed	PA-44-180	An Airspeed event occurs when (IAS $>$ 202) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Altitude	Cessna 172S	An Altitude event occurs when (AltMSL $>$ 12800) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Altitude	PA-28-181	An Altitude event occurs when (AltMSL $>$ 12800) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Altitude	PA-44-180	An Altitude event occurs when (AltMSL $>$ 12800) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
CHT Sensor Divergence	Cessna 172S	A CHT Sensor Divergence event occurs when (E1 CHT Divergence $>$ 100) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.

Event Name	Aircraft Type	Event Definition
CHT Sensor Divergence	PA-28-181	A CHT Sensor Divergence event occurs when (E1 CHT Divergence $>$ 100) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Cylinder Head Temperature	PA-28-181	A Cylinder Head Temperature event occurs when (E1 CHT1 $>$ 500 OR E1 CHT2 $>$ 500 OR E1 CHT3 $>$ 500 OR E1 CHT4 $>$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Cylinder Head Temperature	PA-44-180	A Cylinder Head Temperature event occurs when (E1 CHT1 $>$ 500 OR E2 CHT1 $>$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
EGT Sensor Divergence	Cessna 172S	An EGT Sensor Divergence event occurs when (E1 EGT Divergence $>$ 400) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
EGT Sensor Divergence	PA-28-181	An EGT Sensor Divergence event occurs when (E1 EGT Divergence $>$ 400) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
EGT Sensor Divergence	PA-44-180	An EGT Sensor Divergence event occurs when (E1 EGT Divergence $>$ 400 OR E2 EGT Divergence $>$ 400) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Engine Shutdown Below 3000 Ft	Cessna 172S	An Engine Shutdown Below 3000 Ft event occurs when (E1 RPM $<$ 100 AND AltAGL $>$ 500 AND AltAGL $<$ 3000) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Engine Shutdown Below 3000 Ft	PA-28-181	An Engine Shutdown Below 3000 Ft event occurs when (E1 RPM $<$ 100 AND AltAGL $>$ 500 AND AltAGL $<$ 3000) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Engine Shutdown Below 3000 Ft	PA-44-180	An Engine Shutdown Below 3000 Ft event occurs when (AltAGL $>$ 500 AND AltAGL $<$ 3000 AND (E1 RPM $<$ 100 OR E2 RPM $<$ 100)) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
High Altitude LOC-I	Cessna 172S	A High Altitude LOC-I event occurs when (LOC-I Index $\geq$ 1 AND AltAGL $\geq$ 1500) is triggered at least 2 times within 1 seconds, and ends when no trigger occurs for 1 seconds.
High Altitude Stall	Cessna 172S	A High Altitude Stall event occurs when (Stall Index $\geq$ 1 AND AltAGL $\geq$ 1500) is triggered at least 2 times within 1 seconds, and ends when no trigger occurs for 1 seconds.

Event Name	Aircraft Type	Event Definition
Low Airspeed on Approach	Cessna 172S	A Low Airspeed on Approach event occurs when (IAS $<$ 56 AND AltMSL Lag Diff $<$ 0 AND AltAGL $>$ 100 AND AltAGL $<$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Airspeed on Approach	PA-28-181	A Low Airspeed on Approach event occurs when (IAS $<$ 57 AND AltMSL Lag Diff $<$ 0 AND AltAGL $>$ 100 AND AltAGL $<$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Airspeed on Approach	PA-44-180	A Low Airspeed on Approach event occurs when (IAS $<$ 66 AND AltMSL Lag Diff $<$ 0 AND AltAGL $>$ 100 AND AltAGL $<$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Airspeed on Climbout	Cessna 172S	A Low Airspeed on Climbout event occurs when (IAS $\geq$ 20 AND IAS $<$ 59 AND AltMSL Lag Diff $>$ 0 AND AltAGL $>$ 100 AND AltAGL $<$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Airspeed on Climbout	PA-28-181	A Low Airspeed on Climbout event occurs when (IAS $\geq$ 20 AND IAS $<$ 52 AND AltMSL Lag Diff $>$ 0 AND AltAGL $>$ 100 AND AltAGL $<$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Airspeed on Climbout	PA-44-180	A Low Airspeed on Climbout event occurs when (IAS $\geq$ 20 AND IAS $<$ 70 AND AltMSL Lag Diff $>$ 0 AND AltAGL $>$ 100 AND AltAGL $<$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Altitude LOC-I	Cessna 172S	A Low Altitude LOC-I event occurs when (LOC-I Index $\geq$ 1 AND AltAGL $\leq$ 1500 AND AltAGL $\geq$ 100) is triggered at least 2 times within 1 seconds, and ends when no trigger occurs for 1 seconds.
Low Altitude Stall	Cessna 172S	A Low Altitude Stall event occurs when (Stall Index $\geq$ 1 AND AltAGL $\geq$ 1500 AND AltAGL >= 100) is triggered at least 2 times within 1 seconds, and ends when no trigger occurs for 1 seconds.
Low Ending Fuel	Cessna 172S	An Average fuel the past 15 seconds was less than 8.25
Low Ending Fuel	PA-28-181	An Average fuel the past 15 seconds was less than 8.00
Low Ending Fuel	PA-44-180	An Average fuel the past 15 seconds was less than 17.56
Low Fuel	Cessna 172S	A Low Fuel event occurs when (Total Fuel $<$ 8.25 AND Pitch $<$ 5) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Fuel	PA-28-181	A Low Fuel event occurs when (Total Fuel $<$ 8.0 AND Pitch $<$ 5) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.

Event Name	Aircraft Type	Event Definition
Low Fuel	PA-44-180	A Low Fuel event occurs when (Total Fuel $<$ 17.56 AND Pitch $<$ 5) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Oil Pressure	Cessna 172S	A Low Oil Pressure event occurs when (E1 OilP $<$ 25 AND E1 RPM $>$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Oil Pressure	PA-28-181	A Low Oil Pressure event occurs when (E1 OilP $<$ 20 AND E1 RPM $>$ 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Low Oil Pressure	PA-44-180	A Low Oil Pressure event occurs when ((E1 OilP $<$ 25 AND E1 RPM $>$ 500) OR (E2 OilP $<$ 25 AND E2 RPM $>$ 500)) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
VSI on Final	Cessna 172S	A VSI on Final event occurs when (VSpd <= -1500 AND AltAGL <= 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
VSI on Final	PA-28-181	A VSI on Final event occurs when (VSpd <= -1500 AND AltAGL <= 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
VSI on Final	PA-44-180	A VSI on Final event occurs when (VSpd <= -1500 AND AltAGL <= 500) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Roll	Cessna 172S	A Roll event occurs when (Roll < -60 OR Roll > 60) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Roll	PA-28-181	A Roll event occurs when (Roll < -60 OR Roll > 60) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Roll	PA-44-180	A Roll event occurs when (Roll < -60 OR Roll > 60) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.
Proximity	Cessna 172S	An Aircraft within 500 ft of another aircraft and above above 50ft AGL
Proximity	PA-28-181	An Aircraft within 500 ft of another aircraft and above above 50ft AGL
Proximity	PA-44-180	An Aircraft within 500 ft of another aircraft and above above 50ft AGL

G.3. Distribution of Flight Lengths

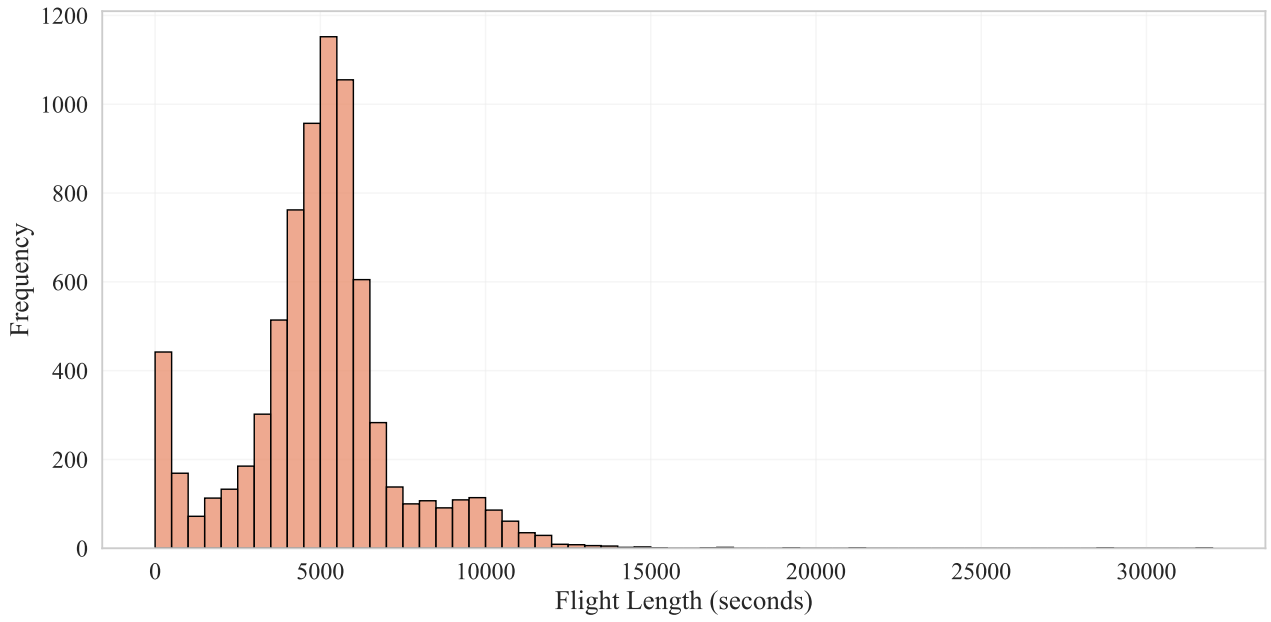

Figure 3. **Distribution of Flight Lengths:** The number of flights with specific flight lengths. Our preprocessed dataset includes flights longer than 1,000 and shorter than 9,995 seconds. As shown, flights longer than 9,995 seconds have fewer than 200 occurrences, indicating that training flights typically do not last longer than 3 hours. Additionally, flights shorter than 1,000 seconds could be associated with pilots powering on the aircraft to park or taxi, rather than a full flight.

G.4. Distribution of Number of Events

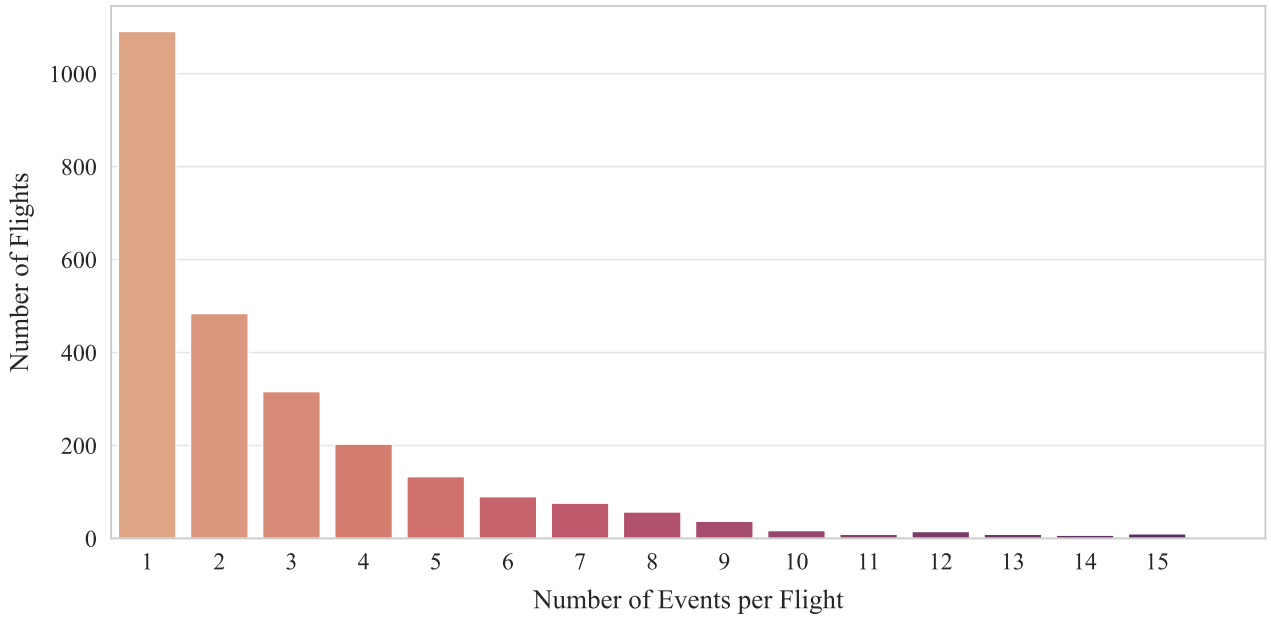

Figure 4. Distribution of Number of Events Across Flights

G.5. Distribution of High Altitude Stalls

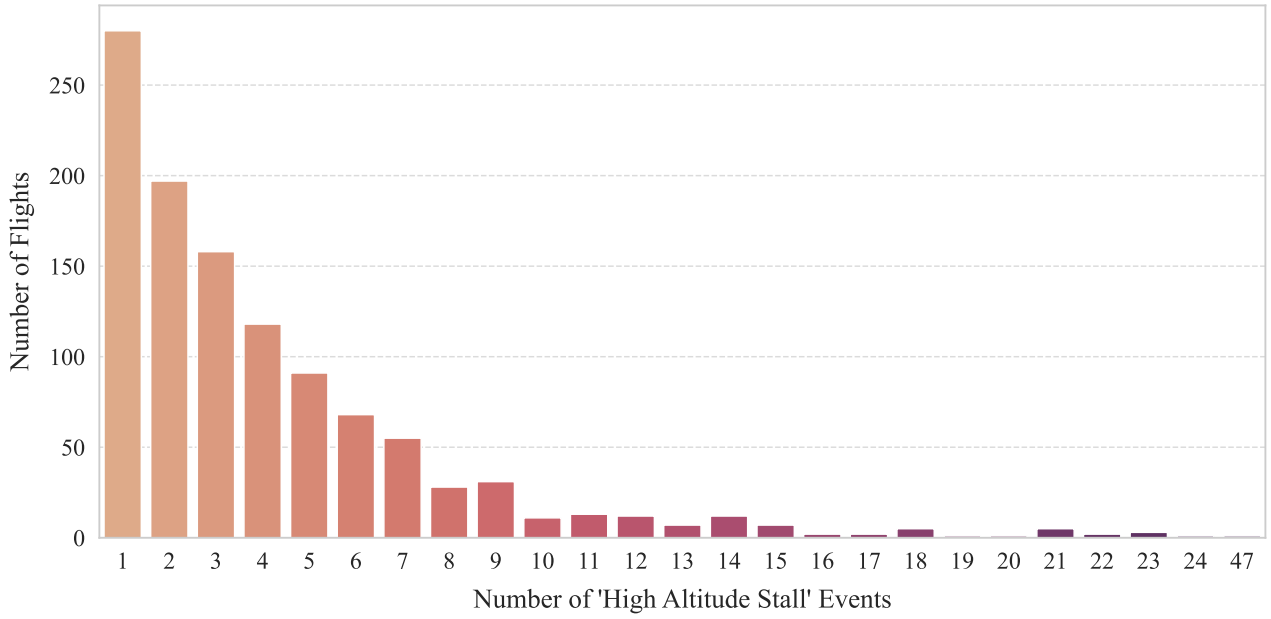

Figure 5. Distribution of High Altitude Stalls Across Flights

G.6. Number of Flights per Event

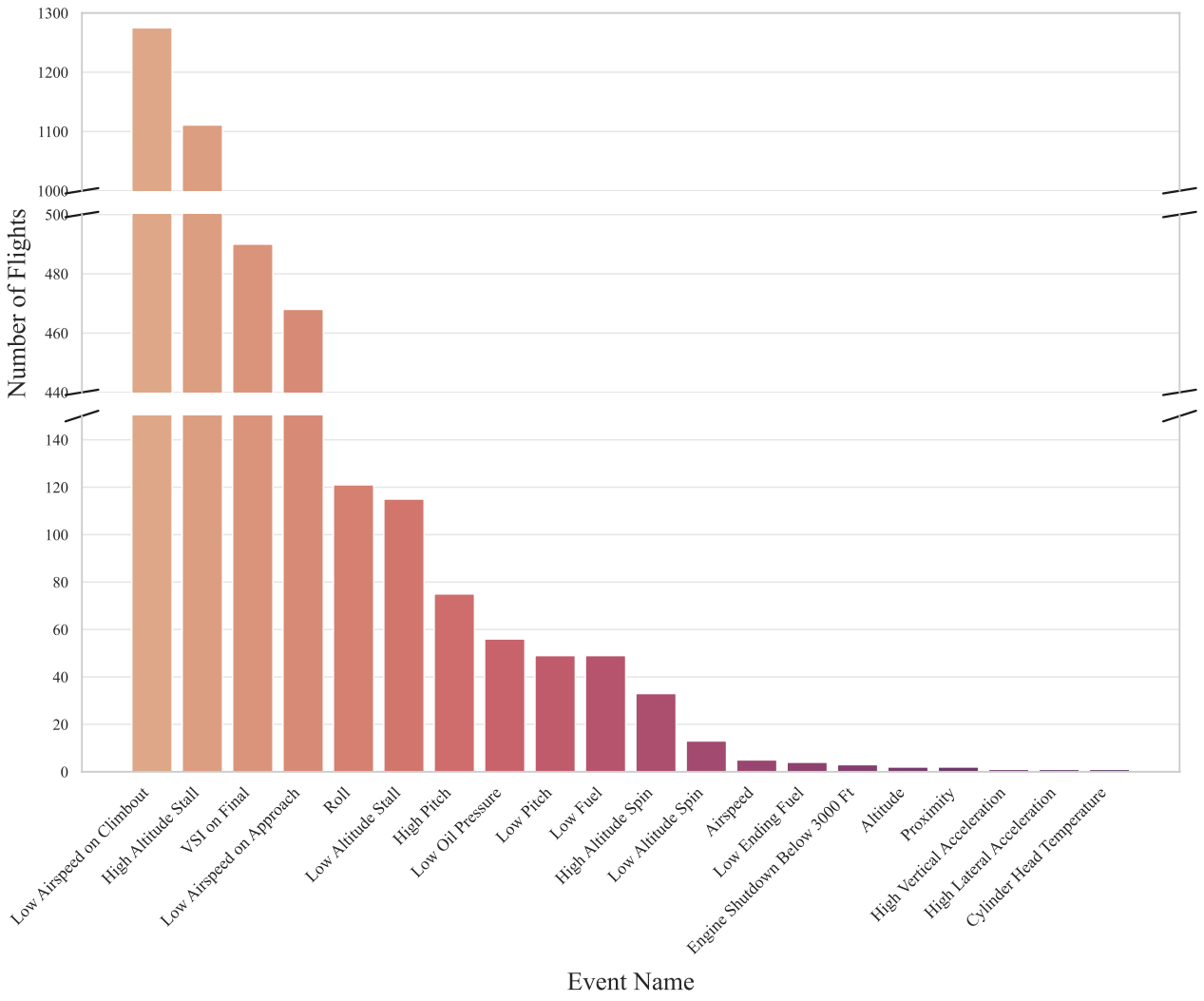

Figure 6. Number of Flights per Event

H. NaN Values Distribution

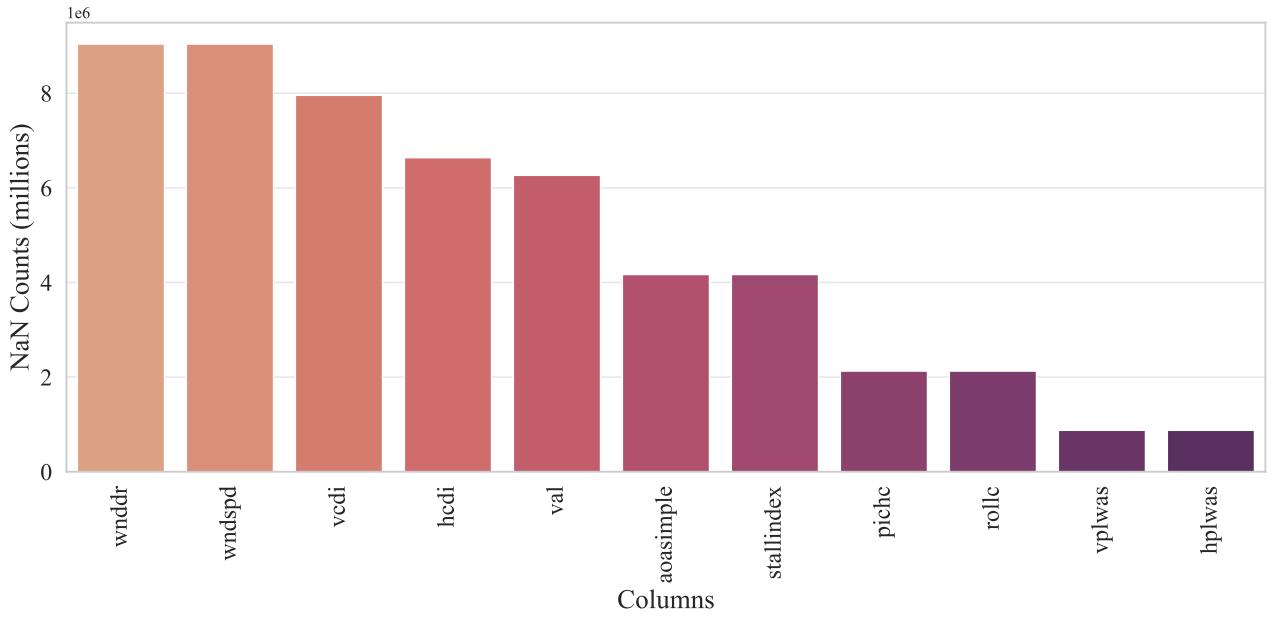

Figure 7. NaN values by column (filtered for columns with >100k NaN values)

I. Masking Parameters and Results

Autoencoder (Masking Strategy)	MAE	MSE
0.2 Masking Ratio / 5 Mean Mask Length	0.338998	0.393149
0.2 Masking Ratio / 60 Mean Mask Length	0.398348	0.509312
0.5 Masking Ratio / 5 Mean Mask Length	0.376519	0.443657
0.5 Masking Ratio / 60 Mean Mask Length	0.462753	0.621275
0.8 Masking Ratio / 5 Mean Mask Length	0.384678	0.459701
0.8 Masking Ratio / 60 Mean Mask Length	0.520065	0.705929

Table 7. **Evaluation metrics for Masked Autoencoder(MAE, MSE):** Masked Autoencoder uses six masking strategies on masked column regression. These results indicate that the best masking strategy is the lowest masking possible. This makes sense, but would cause the model to learn less meaningful representations, as it would be able to shortcut solutions easier with less masked input.

J. Case Study

To demonstrate the practical validity and real-world applicability of the GATS dataset, we conducted two case studies aimed at verifying that sensor data aligns with established aerodynamic and mechanical principles. These analyses serve to confirm that the dataset accurately captures expected relationships between different flight parameters, providing users with increased confidence in the dataset's realism and reliability for machine learning applications.

Case 1 – Aircraft Orientation: In the first case, we tested the correlation of aircraft orientation parameters. The most prominent parameters are altitude above sea level (`altmsl`), vertical speed of the aircraft (`vspd`), and pitch of the aircraft (`pitch`). Figure 8 visualizes the 3 parameters for a randomly chosen Cessna 172 flight. To smooth out the data, we use a EMA (Exponential Moving Average) filter with $\alpha = 0.1$. As can be seen from the figure, as the *pitch* increases, which means as the aircraft in performing a climb maneuver, the `vspd` and `altmsl` accordingly goes up, with a slight lag. Additionally, as the aircraft pitches down to stop the climb, the `altmsl` begins to stabilize at an altitude, and `vspd` drops to around 0 accordingly.

Case 2 – Engine Information: In the second case, we tested the correlation of engine sensor parameters. The essential features for this test included aircraft engine 1 RPM *e1rpm*, aircraft engine 1 - 1st exhaust gas temperature *e1egt1*, and aircraft engine 1 oil temperature *e1oilt*. Figure 9 visualizes these sensor readings with a EMA (Exponential Moving Average) filter with $\alpha = 0.1$. As can be seen from the figure, as the aircraft accelerates or adds power, the RPM increases, which is followed by increases in exhaust gas temperature and oil temperature. This adheres to the logic that increases in engine power increase the mechanical motion in the internal engine and adds more fuel and gas for more combustion, growing oil and exhaust temperatures accordingly.

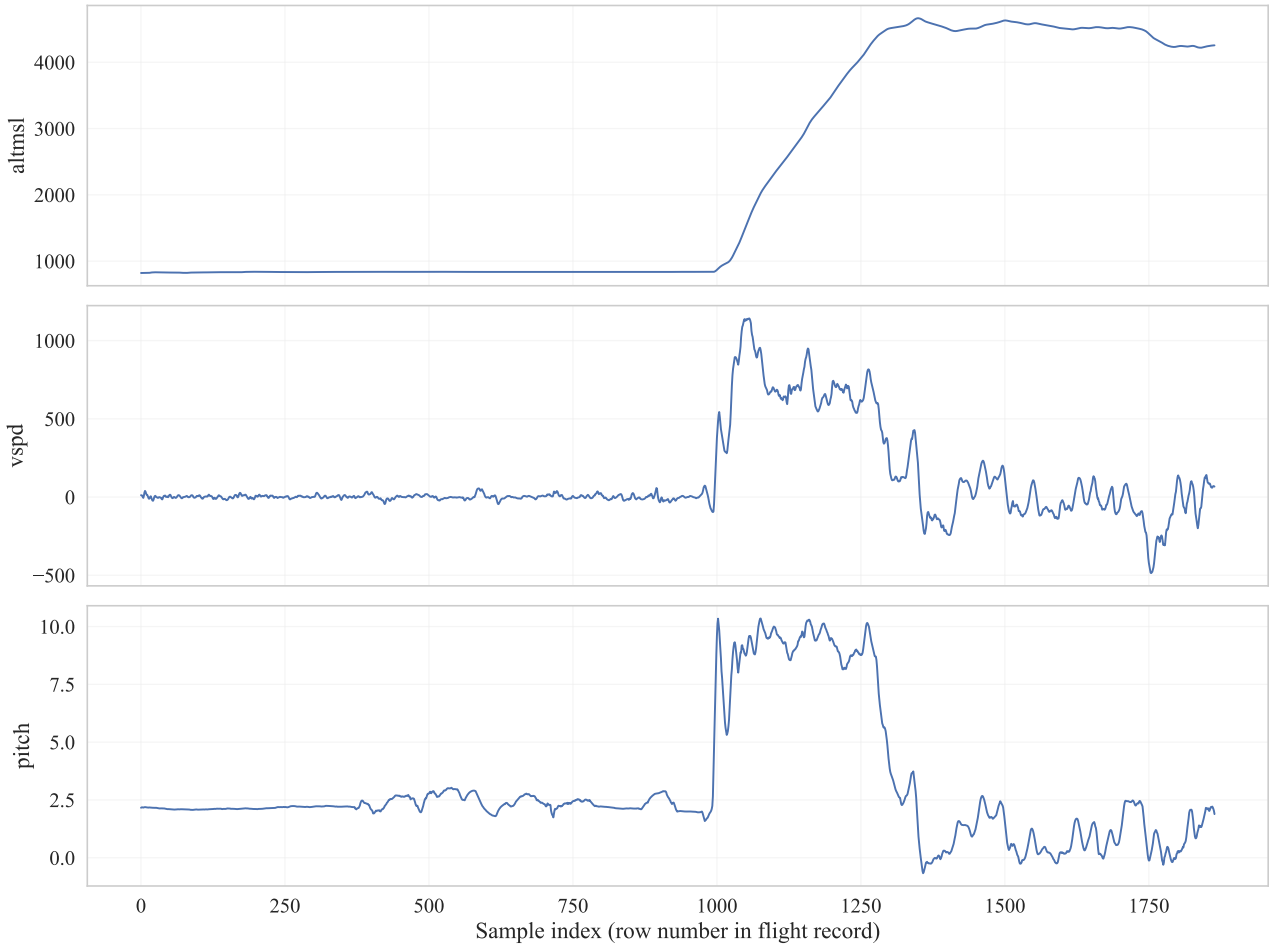

Figure 8. **Cessna 172 Aircraft Orientation Correlation:** Mean Sea Level Altitude (top), Vertical Speed (middle), Aircraft Pitch (bottom).

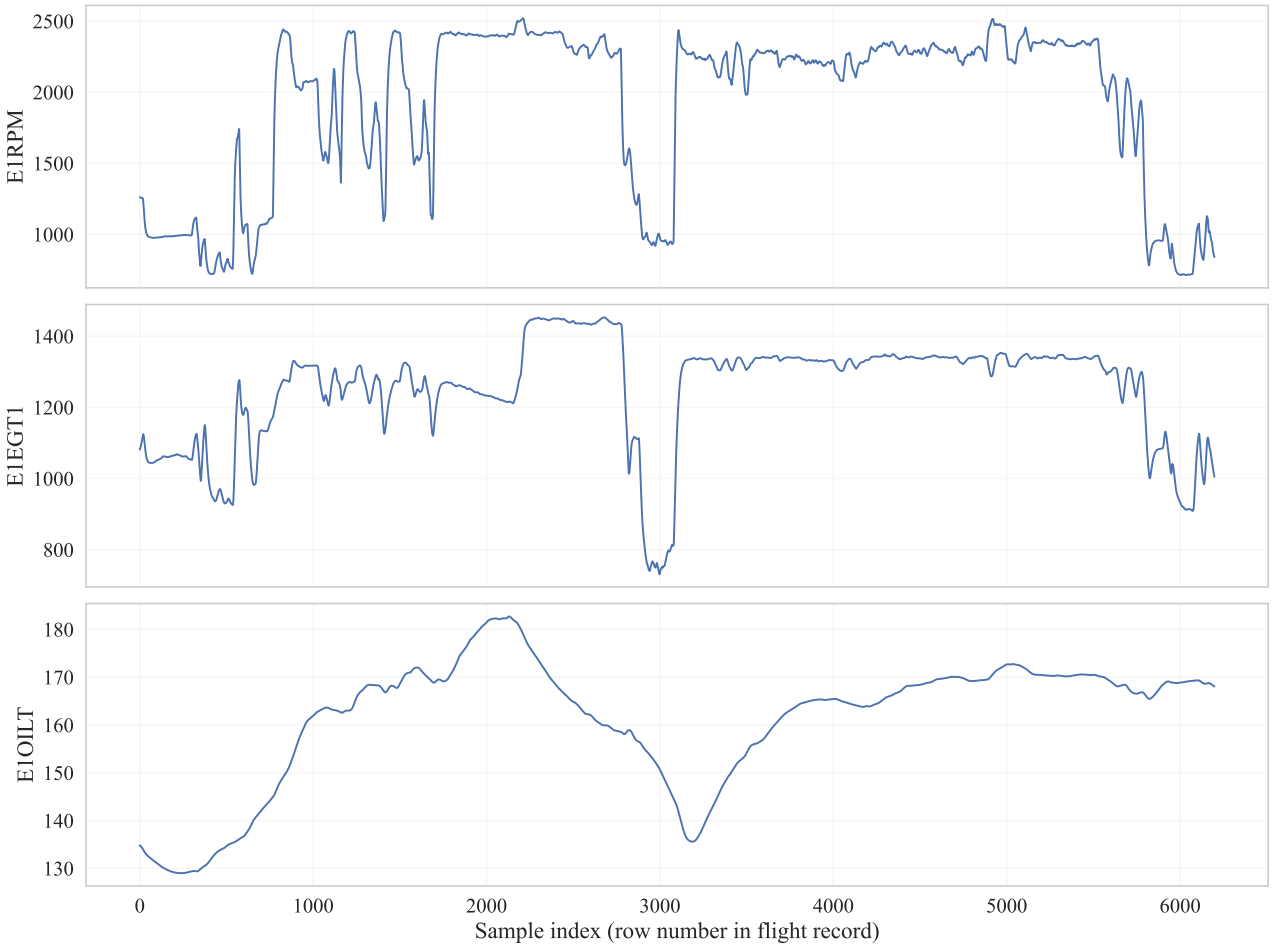

Figure 9. **Cessna 172 Engine Information Correlation:** Engine 1 RPM (top), Engine 1- 1st Exhaust Gas Temperature (middle), Engine 1 Oil Temperature (bottom).

K. TF-IDF Event Based Scoring

Some of the flights in this dataset are additionally associated with events. As described in Section 3.2, these events span 20 different types ranging from High Vertical Acceleration to Low Ending Fuel.

In the context of general aviation, we hypothesize that an event's local and global frequency corresponds to the perceived safety level of the flight it occurs in. To dynamically score a flight's safety based on the contribution of each individual in-flight event, we adapted the TF-IDF algorithm (Robertson, 2004) commonly used in NLP tasks. This allows us to assign importance scores to events, balancing their frequency within a flight and their rarity across the dataset. The sum of all events within a flight are aggregated to represent the safety score of the entire flight. Higher scores correspond to unsafe flights while lower scores correspond to safer flights.

To tailor the algorithm to our needs, we modified the representation of the variables to better reflect the unique characteristics of our data, defining

$$W_{x,y} \;=\; lf_{x,y}\,\log\!\Big(\frac{N}{gf_x}\Big) \tag{1}$$

where $W_{x,y}$ represents the score of event x within flight y, $lf_{x,y}$ is its frequency in that flight, N is the total number of flights, and gf_x is the number of flights containing event x.

For our experiments, we labeled each flight with its calculated TF-IDF safety score once determined. The closest flight scores within a batch are positive pairs, whereas the rest of the flights are negative pairs with respect to the positive pair.

Looking ahead, we aim to explore alternative methods for cluster labeling, including expert labeling. By consulting field experts, we can assign weights to different events based on their relative danger. We acknowledge that within the set of dangerous events, some are significantly more hazardous than others. Incorporating expert-determined weights to our current labeling method would enable us to create more accurate representation labels, ultimately leading to a more robust learned representation of flight safety.

L. TF-IDF Based Clustering

L.1. Contrastive Clustering TFIDF

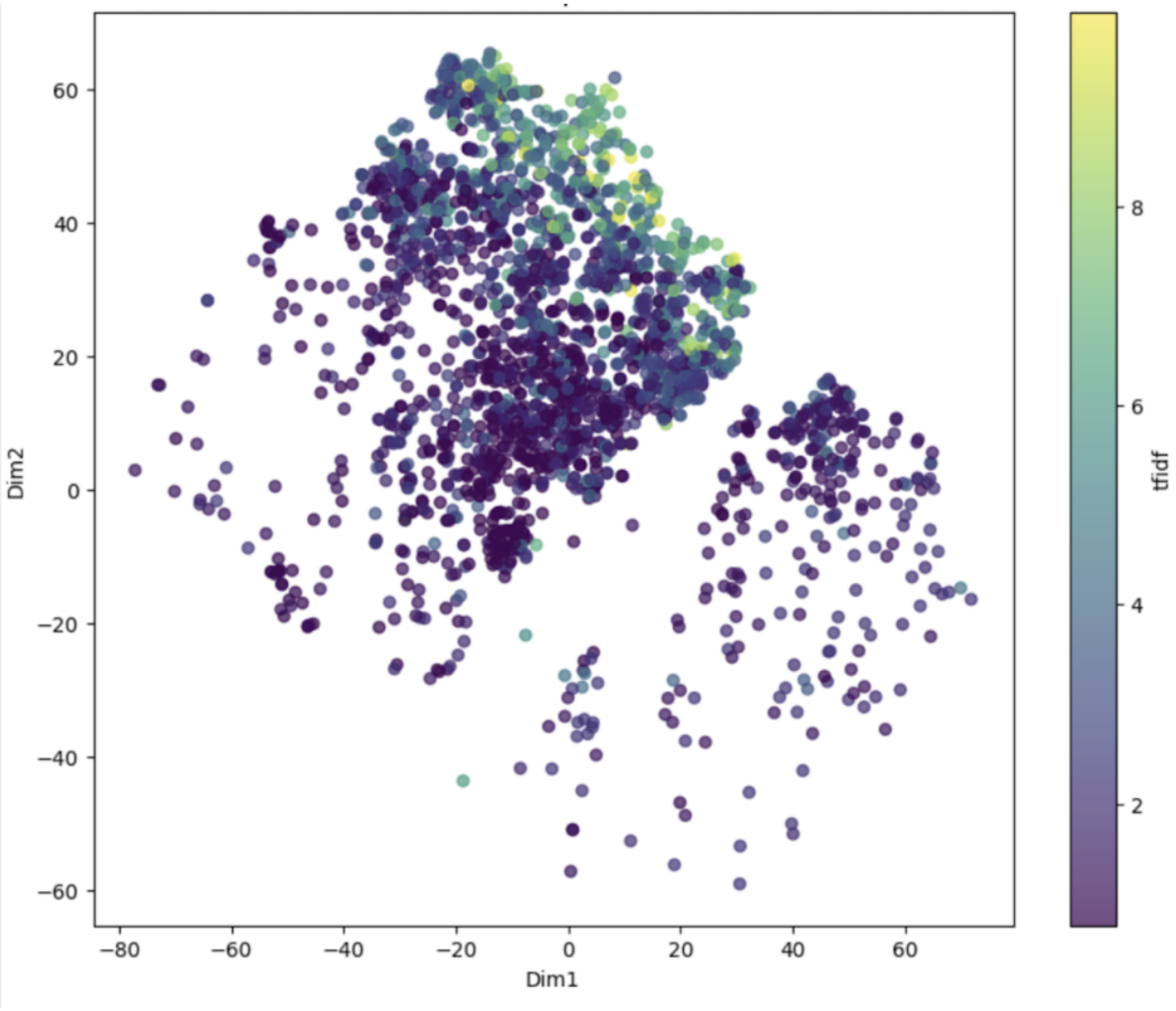

Figure 10. **Contrastive Representation:** Aircraft TF-IDF Score Clustering based on t-SNE

L.2. Contrastive Clustering Aircraft Type

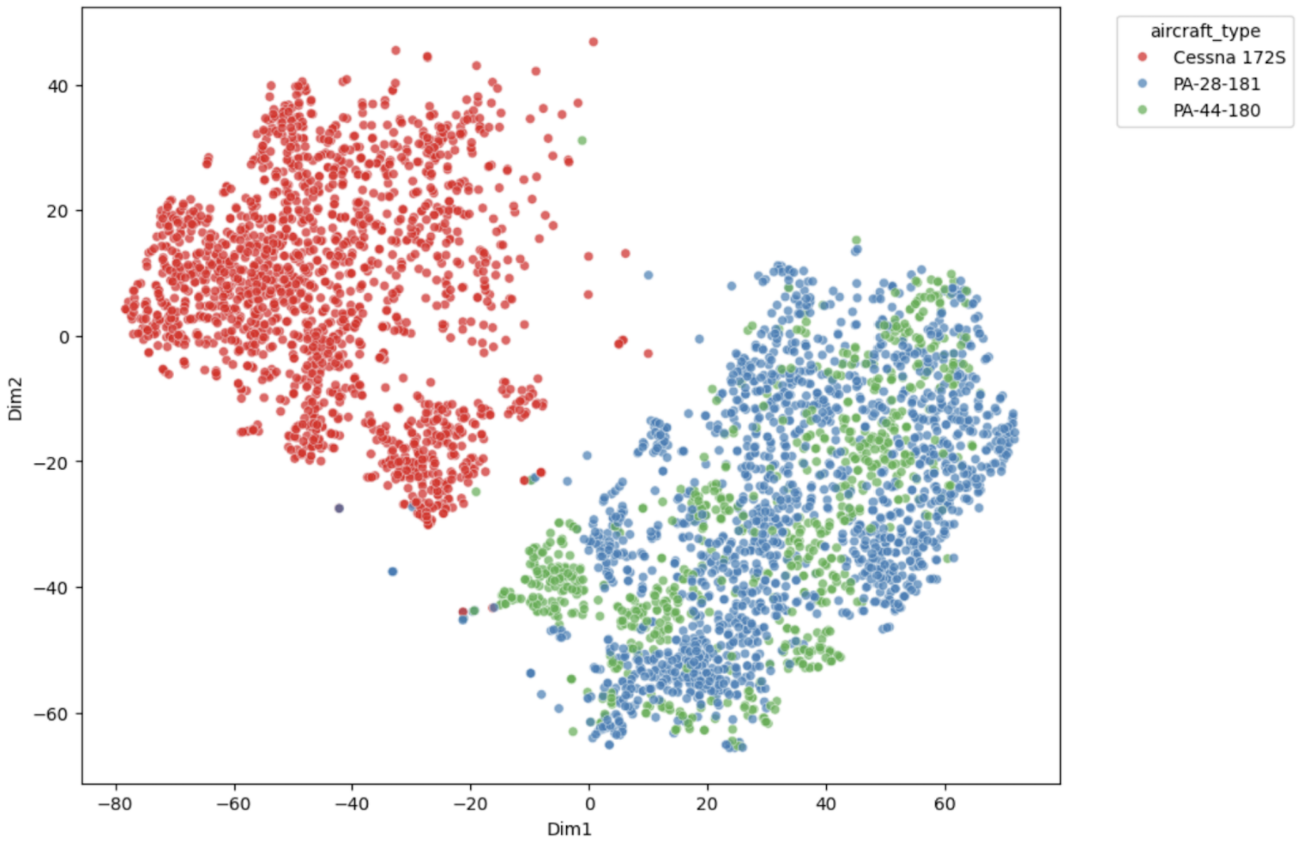

Figure 11. **Contrastive Representation:** Aircraft Type Clustering based on t-SNE

M. Masked Autoencoder Reconstruction Visualizations

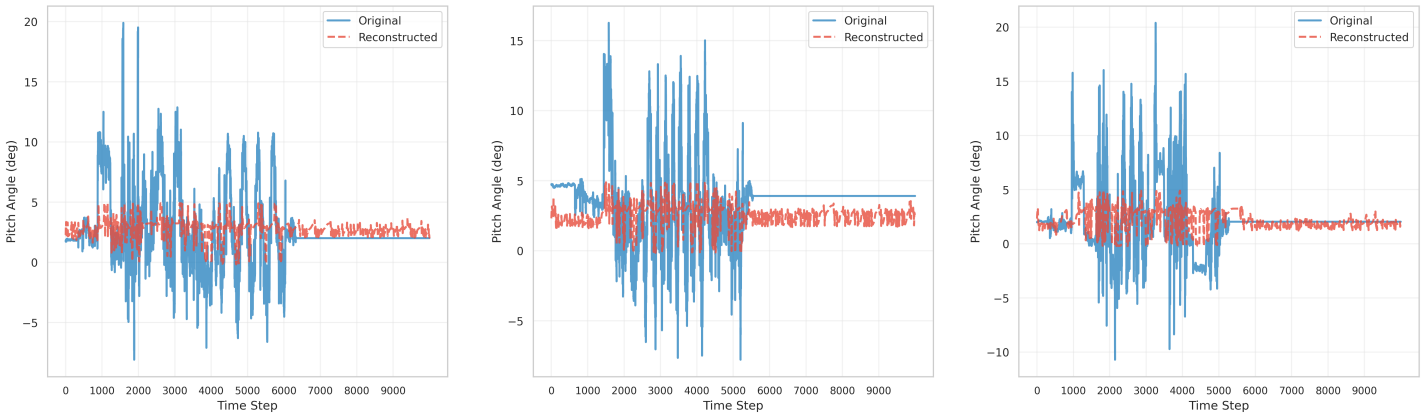

Figure 12. **Pitch Reconstruction Visualization:** Each aircraft type (left to right: Cessna 172S, PA-28-181, PA-44-180) with 0.2 masking ratio and 5 mean mask length.

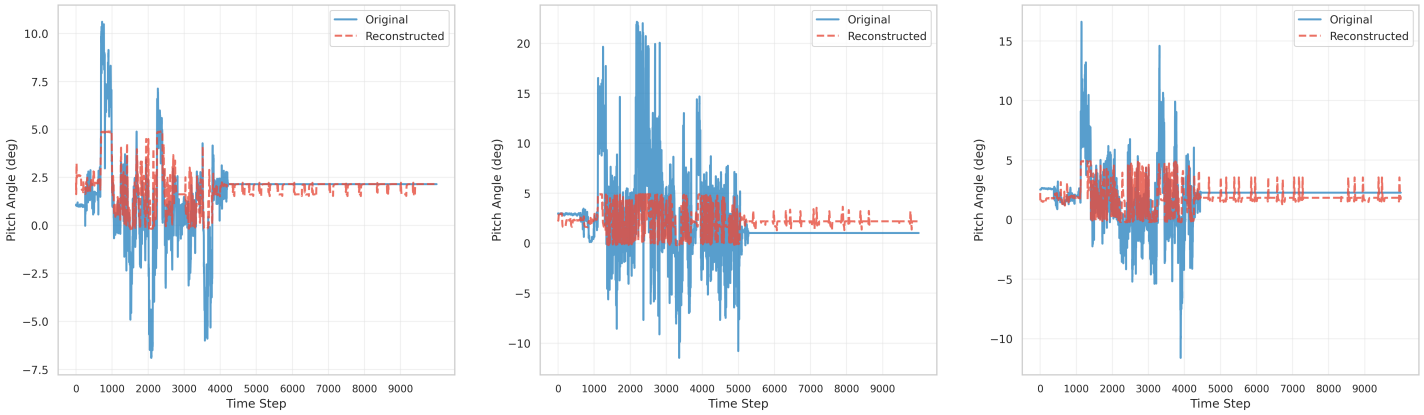

Figure 13. **Pitch Reconstruction Visualization:** Each aircraft type (left to right: Cessna 172S, PA-28-181, PA-44-180) with 0.2 masking ratio and 60 mean mask length.

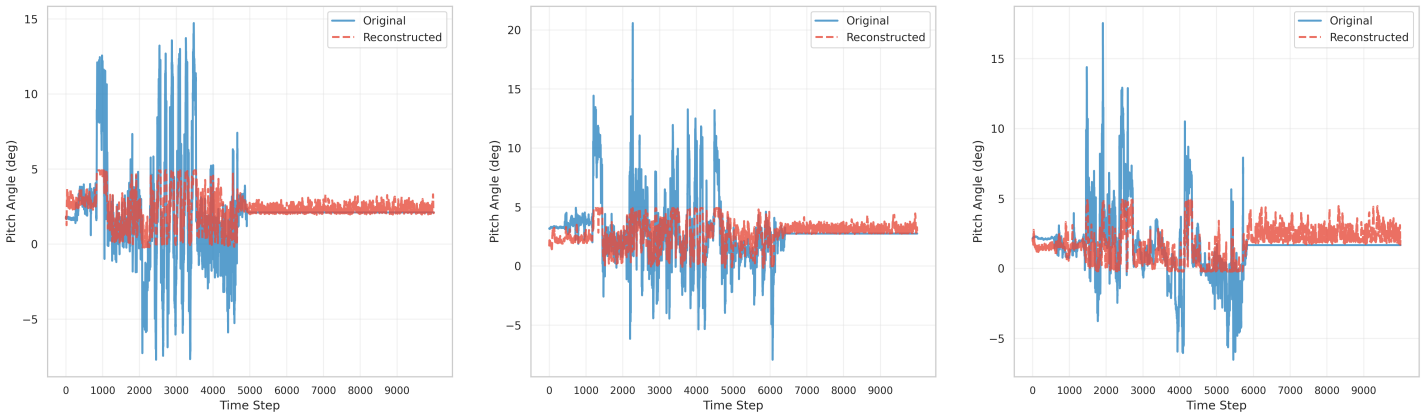

Figure 14. **Pitch Reconstruction Visualization:** Each aircraft type (left to right: Cessna 172S, PA-28-191, PA-44-180) with 0.5 masking ratio and 5 mean mask length.

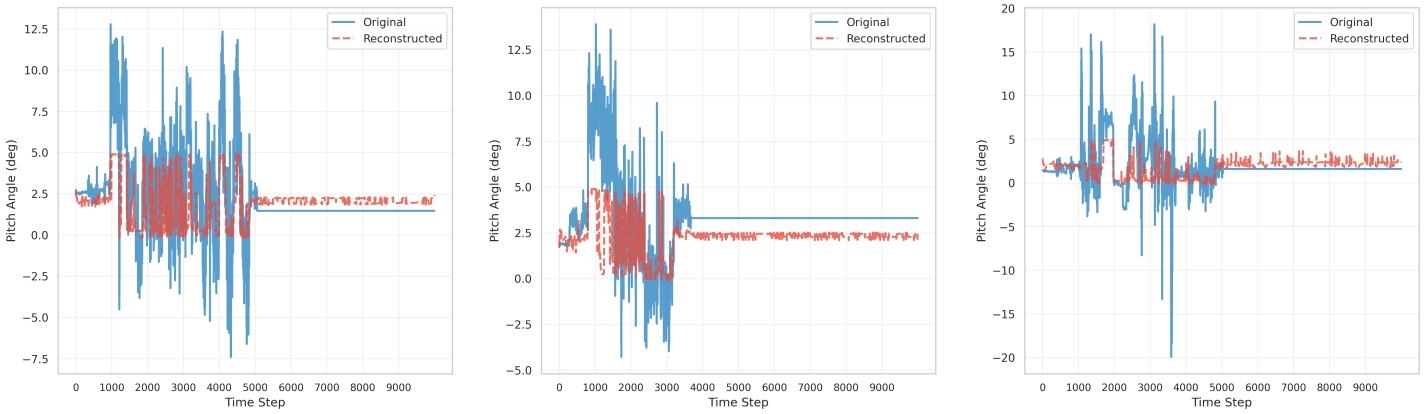

Figure 15. **Pitch Reconstruction Visualization:** Each aircraft type (left to right: Cessna 172S, PA-28-181, PA-44-180) with 0.5 masking ratio and 60 mean mask length.

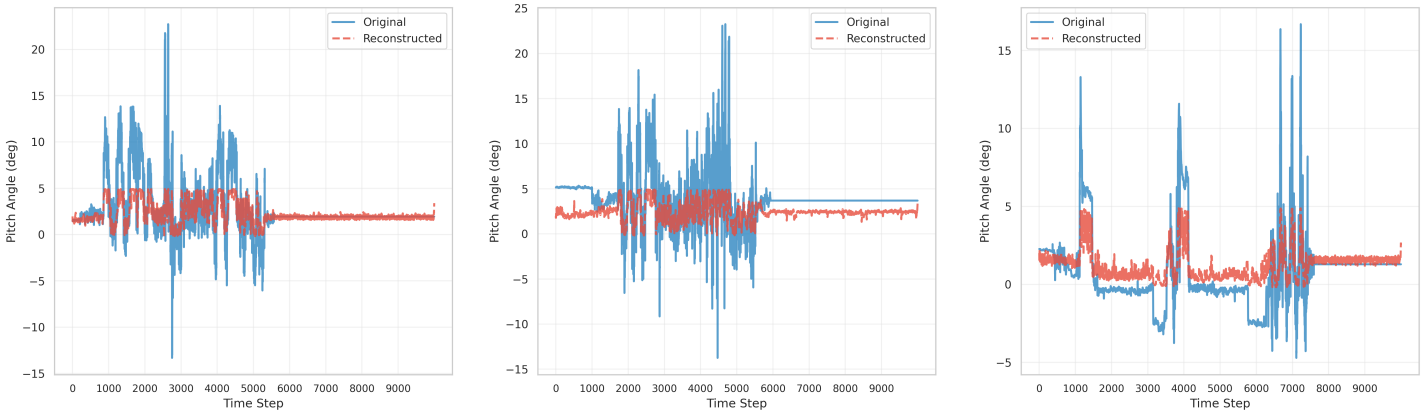

Figure 16. **Pitch Reconstruction Visualization:** Each aircraft type (left to right: Cessna 172S, PA-28-181, PA-44-180) with 0.8 masking ratio and 5 mean mask length.

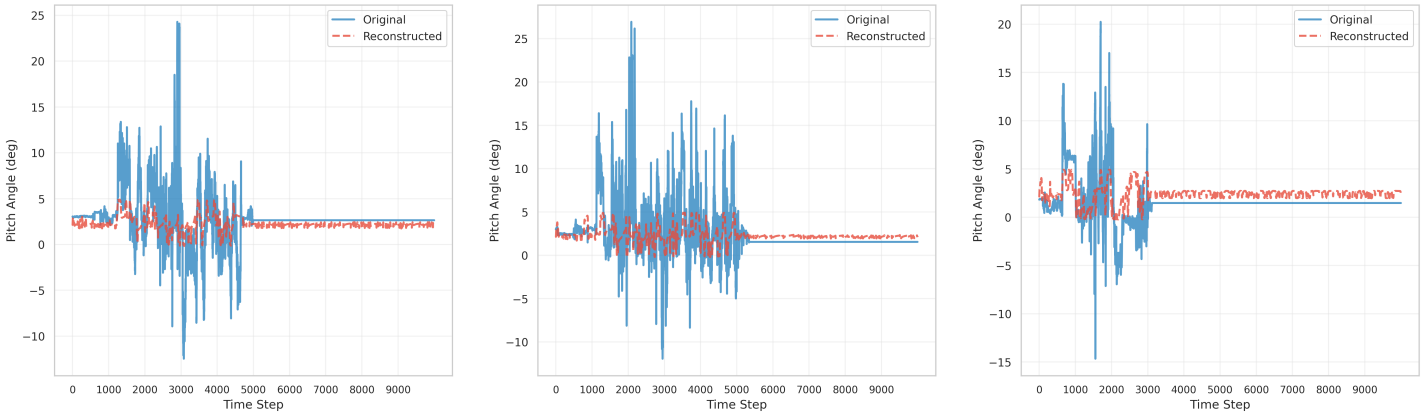

Figure 17. **Pitch Reconstruction Visualization:** Each aircraft type (left to right: Cessna 172S, PA-28-181, PA-44-180) with 0.8 masking ratio and 60 mean mask length.

N. SSL Training Hardware System

Both SSL models were trained on a single NVIDIA Quadro RTX 6000 GPU (24GB VRAM, Turing architecture, 4608 CUDA cores), with access to 32 CPU cores and approximately 375GB of usable system memory. The experiments were conducted on the Brown University Oscar cluster using Slurm-managed GPU nodes, and each job was allocated 1 GPU and 1 CPU core per node. One model took approximately 5 hours to complete, and 6 such models were trained under similar resource configurations (one model for each masking configuration). Disk usage was minimal (8.6GB used of 187GB).