Conditional Variational Autoencoders for aircraft type-specific trajectory generation

Arnault Motte*, Xavier Olive*, Jérôme Morio*

*ONERA – DTIS
Université de Toulouse
Toulouse, France

***Abstract*—Trajectory generation has traditionally relied on methods based on physical aircraft performance models. Recently, data-driven machine learning approaches have demonstrated interesting statistical properties, proving useful in applications such as collision risk modelling and risk estimation, albeit often at the expense of physical realism. Physics-informed machine learning is a promising long-term solution to this limitation, but such models are typically difficult to train in practice. In this paper, we investigate an alternative approach by adapting a VampPrior-based Variational Autoencoder (VAE) architecture from a previous contribution into a Conditional Variational Autoencoder (CVAE). This enables the training of a single model that can handle multiple aircraft types. The proposed CVAE achieves performance comparable to that of dedicated VAE models trained separately for each aircraft type, while additionally addressing two challenges: (i) providing a generative model for sparsely represented aircraft types, for which no specific model can be trained, and (ii) improving overall regularisation of the generated trajectories.**

***Keywords*—air traffic management, trajectory generation, conditional variational autoencoders**

I. Introduction

Trajectory generation refers to the task of predicting or simulating the future evolution of an aircraft's trajectory, given an initial state and context. Generated flight paths should capture both the physical behaviour of the aircraft (e.g., airplane, helicopter, UAV) and the statistical properties of the spatio-temporal environment in which they evolve. When the process is carried out online, while the aircraft is in flight, it is referred to as trajectory prediction. Trajectory generation plays a central role in applications such as capacity prediction, airspace and procedure design, and collision risk modelling.

Traditionally, trajectory generation has relied on physical models of aircraft performance, which incorporate aerodynamic equations, aircraft performance models [1], [2], and operational procedures [3], [4]. While these approaches offer physical interpretability and fidelity, they are not always best suited to certain studies. In contexts such as collision risk modelling or probabilistic safety assessment, capturing the statistical variability of real operations is more important than maintaining strict physical realism, making data-driven trajectory generation particularly attractive [5].

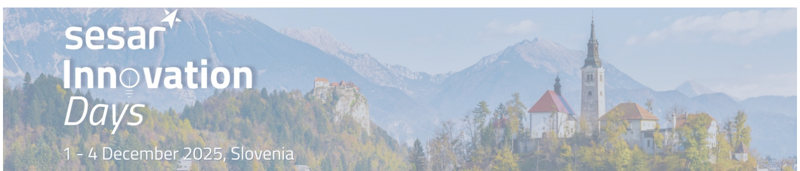
In previous work, we explored various techniques such as copula estimation [6] and variational autoencoders with a VampPrior [7] as generative frameworks for data-driven trajectory modelling. These models demonstrated promising results in capturing the diversity and variability of air traffic while maintaining sufficient control over the generative process. Their probabilistic nature made them particularly well suited to risk-related applications [8], where accounting for uncertainty is as important as modelling central tendencies.

However, a key limitation of these models is that they do not explicitly account for aircraft-specific characteristics, performance, or physical constraints. As a result, the generated outputs may include unflyable trajectories or distributions that fail to capture the peculiarities of specific aircraft types. This reduces realism and limits the usefulness of the models in contexts where the precise timing of aircraft reaching specific milestones is critical.

A straightforward solution would be to train separate models for each aircraft type. In practice, this proves difficult, as illustrated in the case study of Paris Orly airport presented in [8]. Orly serves as a major hub for domestic and short-range flights, with medium-sized aircraft, such as the Airbus A320 family or the Boeing 737, well represented, while routes to overseas territories, traditionally operated from Orly, are covered by wide-body aircraft such as the Airbus A350 or the Boeing 777, which appear only sparsely in the data. Heavier aircraft exhibit distinct climb profiles, yet training dedicated models for them is hindered by the limited number of historical samples. Extending the dataset further back in time does not resolve this issue, as procedures have evolved over the years, making older trajectories less representative of current operations.

To address this issue, we propose the use of conditional variational autoencoders (CVAE), with the aircraft type as the conditioning variable. This approach enables the training of a single generative model capable of handling multiple aircraft types while still accounting for their specific characteristics. The model shares the same structure across types, allowing data from all aircraft to contribute to its refinement. For well-represented aircraft types, the CVAE achieves performance comparable to that of dedicated models, while for sparsely represented types with insufficient data to train a stand-alone model, it still produces trajectories with good physical realism, particularly in the climb phase.

The remainder of this paper is structured as follows. Section II reviews related work on trajectory generation from both physical and machine learning perspectives. Section III describes our modelling approach based on conditional variational autoencoders. Section IV presents the datasets used in our experiments. Section V reports the experimental results and analysis. Finally, Section VI concludes the paper and discusses perspectives for future work.



II. Literature Review

Historically, trajectory generation in air traffic management has relied heavily on deterministic performance models grounded in aerodynamics and operational procedures [1], [2]. While these models ensure physical plausibility, they often fail to capture the variability observed in real-world operations, especially in the context of collision risk assessment and probabilistic safety evaluation.

More recently, the aviation community has increasingly relied on advanced machine learning models for trajectory synthesis, particularly when uncertainty representation is required. There is a growing consensus that hybridization could be key: purely data-driven methods may capture statistical variability but can generate implausible outcomes, whereas purely physical models ensure consistency but lack flexibility. Hodgkin et al. [9] proposed a hybrid physics–data approach for probabilistic simulation of aircraft descents, illustrating how the combination of deterministic physical models with probabilistic learning can yield trajectories that are both operationally realistic and statistically diverse. Complementing this perspective, [10] developed a trajectory generation method rooted in data mining, extracting characteristic patterns from historical trajectories and recombining them to produce realistic flight paths. Their work emphasizes the potential of data-driven methods to exploit latent structures in large trajectory datasets while still preserving operational plausibility.

Beyond hybrid strategies that balance physics and data, another methodological direction with strong potential for trajectory generation comes from continuous-time neural architectures. Neural Ordinary Differential Equations (Neural ODEs), introduced by [11], replace discrete layers in neural networks with parametrized differential equations that are solved numerically. This paradigm is naturally aligned with the continuous dynamics of aircraft motion, offering a way to embed dynamical laws directly into learned representations. Neural ODEs open promising perspectives for generating trajectories that are not only statistically realistic but also dynamically consistent with physical motion. Their interpretability and ability to incorporate domain-specific priors make them an attractive complement to existing generative frameworks.

Nevertheless, physics-informed machine learning remains challenging in practice: embedding complex aerodynamic constraints or operational procedures directly into neural architectures often leads to models that are difficult to train, are computationally expensive, and often unstable when applied to heterogeneous real-world data. These limitations have motivated the search for alternative strategies that balance realism and tractability. In contrast, data-driven generative models are able to learn directly from historical observations, reproducing the statistical diversity of traffic flows while remaining computationally efficient. In this context, Variational Autoencoders (VAEs) and CVAEs offer an appealing compromise between traditional physics-based approaches and purely statistical methods.

VAEs have emerged as a cornerstone in deep generative modelling, enabling the synthesis of realistic, high-dimensional data while maintaining a compact latent representation of the underlying distribution [12]. By jointly learning an encoder that maps data into a probabilistic latent space and a decoder that reconstructs samples, they provide a principled framework for data generation under uncertainty—an essential property for safety-critical domains, where representing variability is as important as modelling central tendencies. Early work in trajectory modelling explored copula-based density estimation [6] or VAE architectures with advanced priors such as VampPrior [7], showing promising results in generating diverse synthetic trajectories. However, these models struggled to capture aircraft-specific performance characteristics, leading to unrealistic outcomes for underrepresented aircraft categories.

III. Methodology

When working on collision risk estimation, relying solely on observed trajectories to conduct Monte Carlo simulations is not sufficient. Loss of separation events are rare, meaning that an extremely large number of simulations is needed to observe only a limited number of cases. In addition, observed datasets are often too limited in both size and diversity to capture the full range of traffic variability. Using only empirical data would therefore lead to a poor approximation of the distribution tails, which are precisely the regions of interest for safety analysis.

To address these challenges, we generate synthetic trajectories using deep generative models, e.g. the Variational Autoencoder (VAE). The majority of deep generative models have in common that they project input data into a lower-dimensional latent space, in which the important characteristics of a trajectory are more salient and where it is easier to define a generative process for sampling. The VAE achieves this by mapping input trajectories through the encoder into latent representations which are lower-dimensional distributions (often Gaussian), called posterior distributions. In the constructed latent space, the posterior distributions are learned so that the resulting mixture is close to an a priori defined distribution (often a standard Gaussian), called the prior distribution. New trajectories are then generated by sampling from this prior distribution and decoding the samples back into trajectory space.

In our earlier work [7], we adapted the VAE architecture to the problem of trajectory generation. Temporal convolutional networks (TCN) [13] were employed in the encoder and decoder to account for the sequential nature of trajectory data. Furthermore, we adopted the Variational Mixture of Posteriors prior (VampPrior) [14], which improves the match between the latent posterior and prior distributions. In this configuration, the prior distribution is learned trough the training process: the prior distribution is defined as a mixture of posterior distributions, whose parameters μ_{VP} and σ_{VP} are generated thanks to learned pseudo-inputs. This allows for sampling within specific components of the prior, thereby focusing on specific regions of the latent space, describing desirable characteristics of trajectories such as procedures of interest.

The Conditional Variational Autoencoder (CVAE) is an extension of the standard VAE designed for situations where generation is assumed to depend on additional information that is observable. Instead of simply modelling the distribution $p(x)$ of the data, the CVAE models the conditional distribution

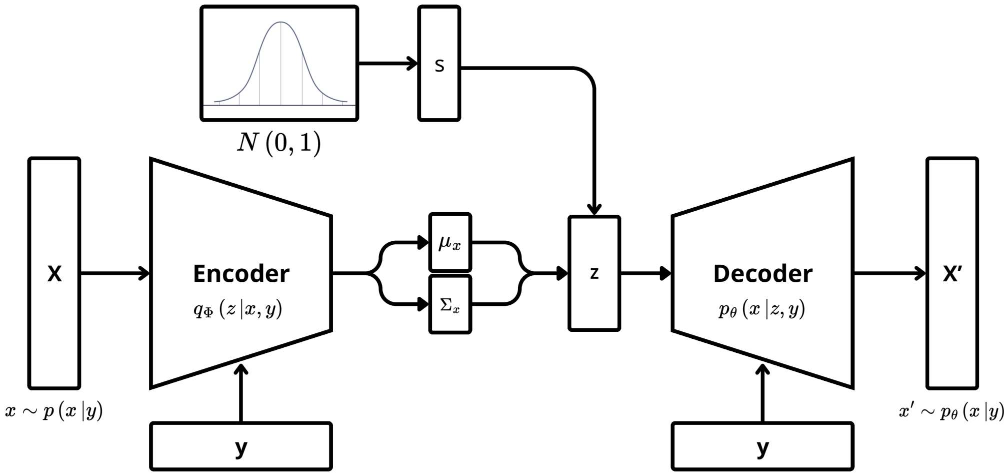


Figure 1. Architecture of a Conditional Variational Autoencoder (CVAE)

$p(x|y)$, where y represents some observed context such as a label, or an aircraft type in our context. This allows the model to better tailor the estimated generative distribution given the condition. Formally, the conditional distribution is expressed as

$$p_\theta(x|y) = \int_{\mathcal{Z}} p_\theta(x, z|y)\, dz = \frac{p(z|y)p_\theta(x|z,y)}{p_\theta(z|x,y)},$$

with z denoting the latent variables and θ the parameters of the generative model. Since exact computation of this integral is intractable, the CVAE, like the VAE, relies on variational inference [15] to approximate the posterior.

In practice, the CVAE introduces the conditioning variable y into both the encoder and the decoder. The encoder learns an approximate posterior $q_\phi(z|x,y)$, which captures how the latent representation z depends jointly on the data x and the condition y. The decoder then uses both z and y to generate samples through $p_\theta(x|z,y)$. This design ensures that the conditioning variable directly influences the generation process (Figure 1). In practice, the conditioning variable corresponds to the aircraft type. During generation, the user specifies the aircraft type and samples from the corresponding conditional latent distribution, optionally providing an initial state if the output is to be replayed in a simulator.

While one could in principle also define a conditional prior $p(z|y)$, many implementations assume a simpler standard Gaussian prior $p(z)$. Although this simplification makes the model easier to train, it may reduce the effectiveness of conditioning. For this reason, we stick to models with a properly conditioned prior because of the strong dependence we assume on y.

To adapt the CVAE for trajectory generation, we build upon the baseline VAE architecture enhanced with TCNs and a VampPrior (Figure 2). The encoder and decoder are composed of multiple residual TCN blocks, which allow the model to capture temporal dependencies across the trajectory sequence while maintaining efficient training. Categorical labels, such as the aircraft type, are one-hot encoded and, if available, can be concatenated with additional continuous features. These condition vectors are projected via fully connected layers to match the sequence length of the trajectories, enabling seamless integration into the TCN input.

We rely on one-hot encoding because the number of aircraft types considered is small and fixed, and the resulting conditional components in the latent space remain directly interpretable. For larger categorical spaces, such as airline, route, or runway, learned embeddings would likely be preferable.

Within the encoder, the trajectory x and processed condition y are mapped to a conditional posterior $q_\phi(z|x,y) = \mathcal{N}(\mu_x, \Sigma_x)$, while the decoder reconstructs the trajectory from a latent vector z concatenated with a transformed y, producing $p_\theta(x|z,y)$.

The ELBO is optimised as

$$\log p_\theta(x|y) \geq \mathbb{E}_{z\sim q_\phi}[\log p_\theta(x|z,y)] - D_{KL}[q_\varphi(z|x,y) \,\|\, p(z|y)]$$

where $p(z|y)$ is a label-conditioned prior, ensuring that latent representations respect the information contained in y.

Similarly to our previous approach with VAE, the VampPrior extends the expressiveness of the latent prior by defining it as a mixture of posteriors. In the conditional setting, the prior becomes

$$p(z|y) = \frac{1}{N}\sum_{i=1}^{N} \omega_i(y)\, q_\varphi\left(z|\chi_i\left(y\right), y\right)$$

where $\chi_i(y)$ are pseudo input conditioned by the label and generated through a dense neural network and $\omega_i(y)$ are label-dependent mixture weights. This allows the model to represent label-specific latent distributions and reduces the risk of posterior collapse.

Practical considerations such as memory limitations influence the number of TCN blocks and pseudo-inputs used, but also

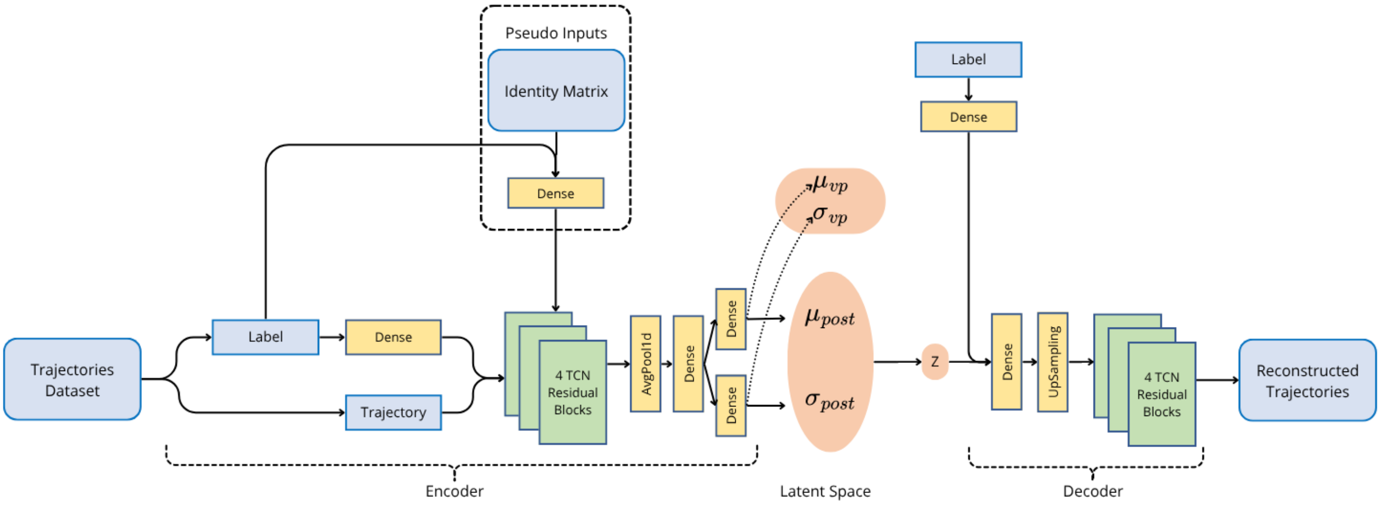


Figure 2. Architecture of the CVAE using TCN and VamPrior, the contribution of this paper

the number of categorial labels that can be considered. Residual connections within TCN blocks stabilise training, while fully connected layers project between latent and trajectory spaces. Overall, this CVAE architecture, combining TCN-based sequence modelling and a conditional VampPrior, generates trajectories that are temporally coherent and conditioned on the aircraft types.

IV. Datasets

In this study, we build upon the dataset used in our earlier work [8] by extending the observation window in order to have enough samples for many aircraft types. Specifically, we collected ADS-B surveillance data from the OpenSky Network covering traffic within a 50-nautical-mile radius around Paris–Orly Airport (`LFPO`) over a period of four years from January 2021 to December 2024.

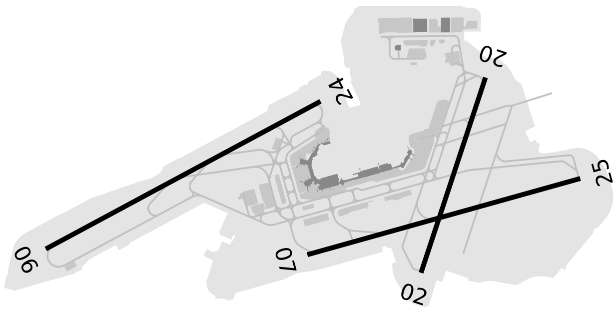


Figure 3. Runway layout of Paris–Orly Airport (`LFPO`).

As in our previous case study, the dataset was carefully curated to ensure that only relevant operations were retained. Non-standard movements such as calibration flights, test flights, and repositioning operations to or from other airports in the Paris terminal area were excluded. For this study, we only focused on departing trajectories from runway 07, starting 1.5 nautical mile beyond the departure runway end, ensuring consistency in the geometric representation of flight paths.

The choice to focus on departures from runway 07 is motivated by the specific characteristics of the corresponding SID procedures, illustrated in Figure 4. In this configuration, departing aircraft are expected to overfly the STAR flows converging through the `MOLBA` fix, which is relatively unusual from an ATC perspective, as explained in [8]. Ensuring that aircraft remain below a prescribed altitude is generally simpler than monitoring their climb performance, which can be challenging for heavy long-haul aircraft, especially in summer conditions.

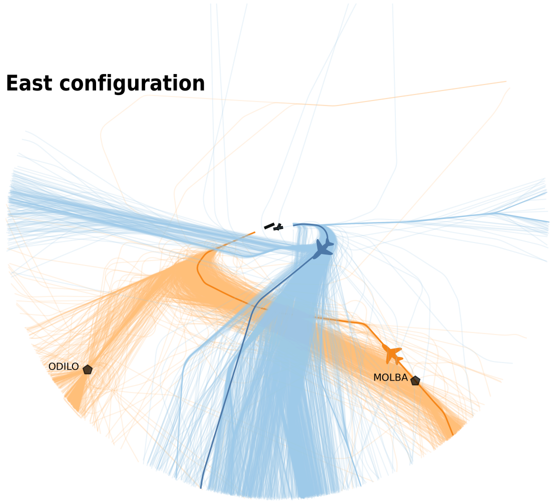


Figure 4. Sample of collected trajectories at Paris–Orly Airport in the *west configuration* (top) and *east configuration* (bottom).

A further advantage of analysing this procedure is that, because the departing flow systematically passes above the arriving one, aircraft are not required to level off around 10,000,ft for operational constraints related to the local environment. This feature avoids introducing biases in the dataset that could otherwise affect the training or inference phases of the model.

Other distinctive features of this dataset stem from the operational context at Paris–Orly Airport. Southbound traffic is particularly overrepresented, and the majority of flights consist of short- and medium-haul operations operated with medium-sized aircraft, primarily from the A320 and B737 families. Nonetheless, as highlighted in [8], potential losses of separations can mostly arise from long-haul aircraft (e.g., A359, B77W), whose climb performance may be insufficient under certain conditions.

Over the four-year period considered, and after filtering and curation, we retained more than 110,000 trajectories. The distribution of the most frequently represented aircraft types is reported in Table I.

TABLE I. DISTRIBUTION OF AIRCRAFT TYPES IN THE DATASET.

Aircraft Type	Count	Aircraft Type	Count
B738	31,358	A333	1,186
A320	21,732	A332	1,113
A321	9,573	B77W	649
A319	5,538	A339	493
A20N	3,690	A35K	484
A318	3,100	B737	454
A21N	2,181	B772	419
A359	1,523	AT45	324
E145	1,467	B788	266

The VAE and CVAE models were trained on trajectories resampled to 200 points within a 50 nautical mile radius. The input features were the track angle, ground speed, altitude, and a δ_t parameter, which scales the trajectories to account for the uniform resampling of flights with different durations. Latitude and longitude were retained in the dataset but not used directly in training; instead, they were reconstructed after generation by integrating the position derivatives.

Because of memory limitations, the CVAE and VAE models were trained on only five different aircraft types: B738, A321, A320, A318, A21N. Furthermore, to efficiently evaluate the performance of the CVAE and VAE on under-represented aircraft, **we deliberately limited the training to 108 trajectories of the A320 aircraft type**.

V. RESULTS

In this section, we evaluate the proposed CVAE framework against baseline VAE models to determine its ability to generate aircraft trajectories that are both statistically representative and physically plausible. Since trajectory generation must balance fidelity to observed operational data with compliance to aerodynamic and procedural constraints, we evaluate the generation along four different criteria. First, we assess physical realism by replaying generated trajectories in a high-fidelity air traffic simulator. We then examine statistical consistency by comparing lateral dispersion, vertical rate distributions, and climb profiles across aircraft types. Together, these analyses provide a comprehensive picture of how the CVAE captures aircraft-specific performance characteristics while benefiting from joint training across categories.

A. *Physical evaluation based on a simulator replay*

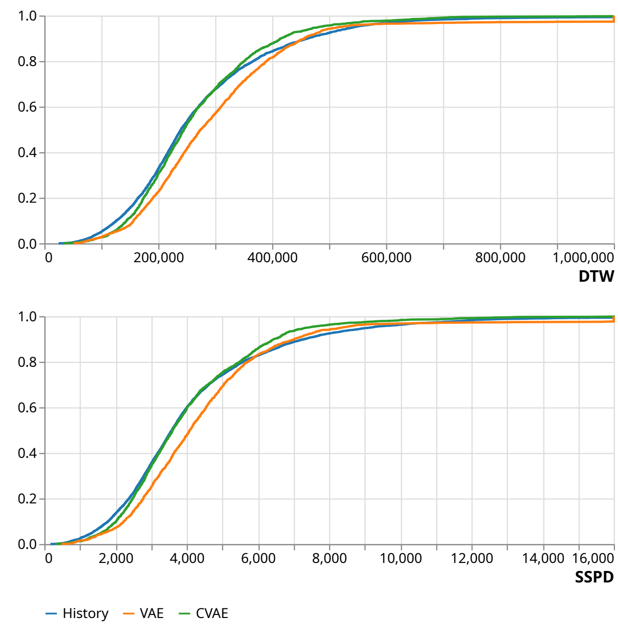


Figure 5. Cumulative distribution function of the distance between A320 aircraft trajectories (the less represented class) and their replay through the BlueSky simulator

In [5], we recommended assessing the physical realism of simulated trajectories by replaying them in a simulator, here the open-source BlueSky simulator [16]. Alignments with official navigational points in the area are detected by comparing the track angle of the trajectory with the bearing to the navigational point (within a tolerance of one degree), and the simulator is then piloted using `DIRECT TO` instructions.

We replayed three datasets: the reference set of real trajectories, trajectories generated with the VAE–VampPrior method, and trajectories generated with the CVAE–VampPrior method. Not all trajectories replay with the same level of accuracy, even for historical data, since not all influencing parameters can be reinjected into the simulator. To quantify the discrepancies, we computed trajectory distances between the original and replayed paths using the `traj-dist`[1] Python library. This library implements nine different distance metrics; here we report only two representative ones in Figure 5: Dynamic Time Warping (DTW) and the Symmetric Segment-Path Distance (SSPD).

Ideally, the cumulative distribution functions (CDFs) of these distances should converge to one at very small values. This is not observed for historical data, owing to the inherent difficulty of replaying real traffic, but generated trajectories should at least reach convergence as quickly as historical ones. In practice, the CVAE distributions outperform those of the VAE on both metrics, and in some cases even surpass historical trajectories on the less well-reconstructed cases. This suggests that the CVAE better captures the structural properties of the SID procedures than the VAE which was trained on very little data for the VAE model.

B. *Lateral distribution of generated trajectories*

Figure 6 shows 500 trajectories of A320 and A21N aircraft from the historical dataset, compared with trajectories generated using the baseline VAE with VampPrior (trained separately for each aircraft type) and the proposed CVAE with VampPrior (trained jointly across all aircraft types). For clarity, we only consider lateral profiles which are expected to match the historical statistics across all SID procedures.

The historical dataset exhibits well-defined routes in the four cardinal directions, with four distinct exit points to the south. The baseline VAE captures the overall distribution but also produces trajectories in regions where traffic is not typically observed, and shows higher variance around the centroids of the main routes.

By contrast, the proposed CVAE-based generation yields distributions more closely aligned with documented and historically flown routes. Cluster separation is more consistent with the historical data, and off-route samples are reduced. This suggests that the CVAE acts as a regulariser, with generated routes more concentrated around realistic patterns. Importantly, the CVAE benefits from training across multiple aircraft categories, leveraging the larger and more diverse dataset to better capture the underlying trajectory distribution.

C. *Distribution of vertical rates*

One natural metric for distinguishing aircraft types in their climb profiles is the climb rate (or vertical rate). A straightforward approach with historical data is to plot the distribution of the average vertical rate over the climb phase, which typically reveals that heavier aircraft exhibit lower climb rates than lighter ones.

[1]https://github.com/bguillouet/traj-dist

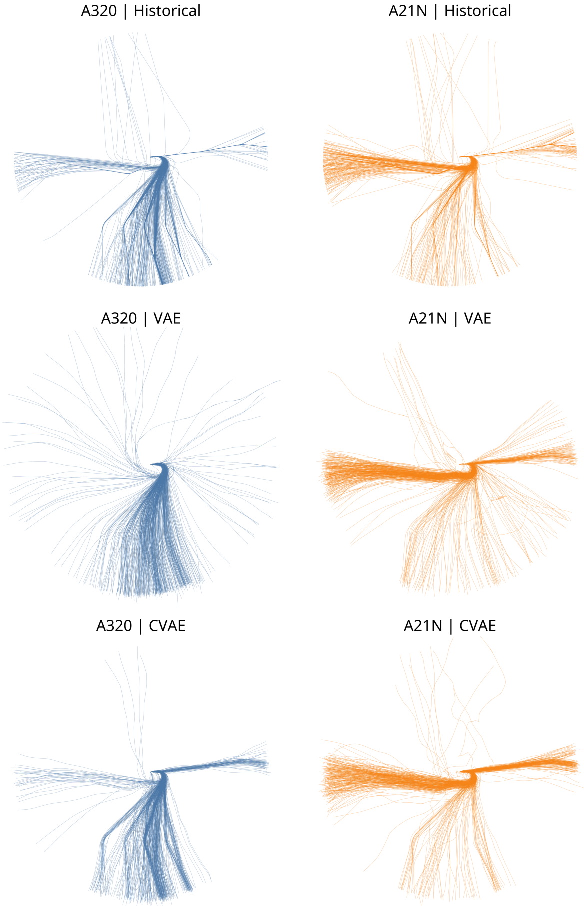


Figure 6. Lateral dispersion of historical and generated trajectories for two different aircraft types

The vertical rate is not included among the features used to train the model or to generate trajectories. Consequently, for generated samples the vertical rate is recomputed from the altitude profile.

Figure 7 shows these distributions for the most represented aircraft types (see Table I), and compares the performance of the VAE and CVAE models against this aspect. For the B738, the VAE trained on 30,000 trajectories (split into training and test sets) yields a distribution that matches the historical data almost perfectly. For the A21N, however, the historical dataset is more limited, with only 2,000 trajectories. When trained on 108 trajectories, the A320 results in a very inaccurate distribution than can only be improved with the CVAE.

In this case, the CVAE, which leverages data across all aircraft types, produces a distribution that more closely follows the historical one. This highlights the advantage of the conditional approach when data availability is uneven across categories.

D. *Performance in climb profiles*

Another common way to verify physical consistency is to plot the calibrated airspeed (CAS) against altitude. During training, wind effects were deliberately ignored, and only ground speed was used (with track angle as a component of the position derivative). While indicated airspeed provides a good

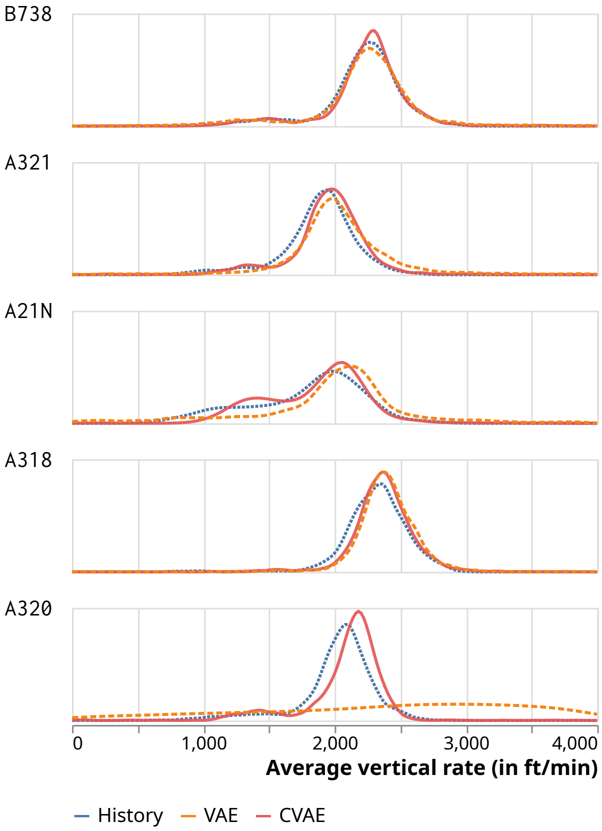


Figure 7. Distribution of average vertical rates across the 5 represented aircraft types. One VAE model has been trained for each aircraft type, but the same CVAE is used across all of them.

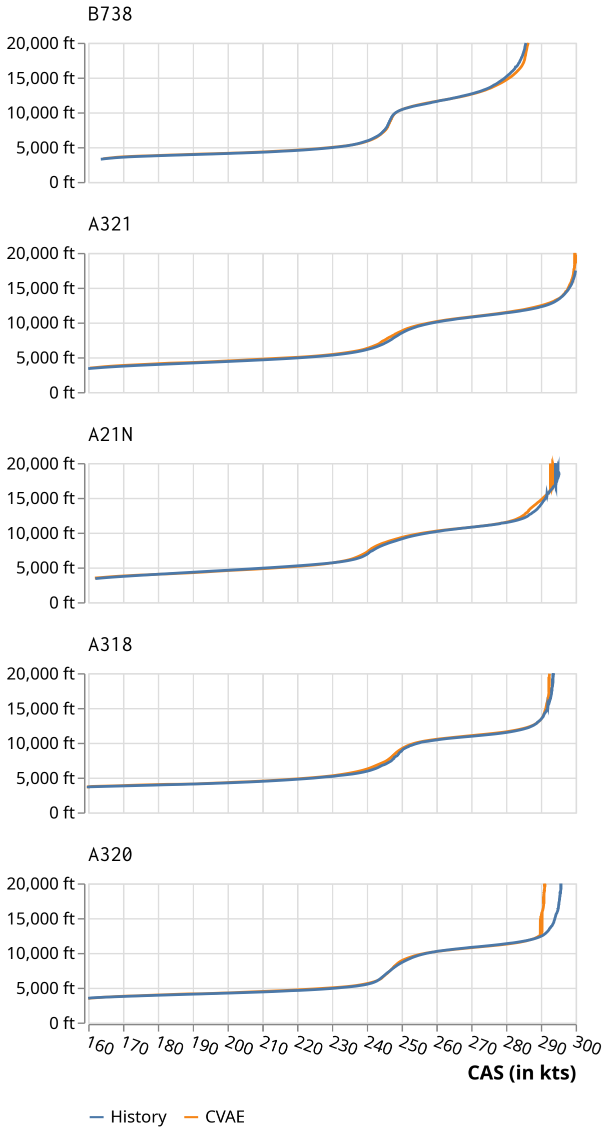


Figure 8. Historical and generated mean climb profiles across the 5 represented aircraft types. No wind model taken into account; the 250 kts limit below 10,000 ft seems respected.

approximation of CAS and can, in some areas, be available through Extended Mode S, it was not used as a training feature. Instead, CAS is recomputed from ground speed, assumed equal to true airspeed, using the standard atmosphere model (ISA).

Figure 8 compares the mean climb profile of the historical trajectory closest to the centroid of the training dataset with the mean climb profile from the generated dataset. Although individual profiles differ slightly, the model correctly reproduces the regulatory constraint that CAS must remain below 250 kts below 10,000 ft. Minor deviations may be attributed to the approximation used in computing CAS. Residual noise is also still visible, likely due to imperfect preprocessing of the original data.

VI. Conclusion and Future Work

In this work, we proposed a simple yet effective extension of the VAE VamPrior framework presented in [7] and exemplified in [8] to a conditional setting, showing that even modest architectural modifications can significantly improve the quality and realism of generated trajectories. The CVAE model manages to capture the specificities in climb profiles for several aircraft types, even less represented, while remaining less complex and easier to train than more elaborate architectures. In particular, it demonstrates an ability to use data from other categories and to regularise generation, yielding trajectories that adhere more closely to documented procedures and physical constraints, without requiring additional model complexity or expensive training strategies.

Future work will explore several directions. First, improving the model further by integrating additional contextual information and refining the conditioning mechanism. Second, incorporating physics-informed architectures, such as neural ordinary differential equations [11], to better embed dynamical constraints directly into the generative process. Finally, we aim to investigate transfer learning strategies, leveraging trajectories of well-represented aircraft types from other airports and adapting them to procedures of the case study airport, in order to address data scarcity for less frequent types.

References

[1] J. Sun, J. M. Hoekstra, and J. Ellerbroek, "OpenAP: An Open-Source Aircraft Performance Model for Air Transportation Studies and Simulations," *Aerospace*, vol. 7, p. 104, July 2020.

[2] A. Nuic, D. Poles, and V. Mouillet, "BADA: An advanced aircraft performance model for present and future ATM systems," *International Journal of Adaptive Control and Signal Processing*, vol. 24, pp. 850–866, Aug. 2010.

[3] M. Waller, J. Rigopoulos, D. Blackman, and T. Berreen, "Considerations in the application of dynamic programming to optimal aircraft trajectory generation," in *IEEE Conference on Aerospace and Electronics*, (Dayton, OH, USA), pp. 574–579, IEEE, 1990.

[4] D. Delahaye, P. Tsiotras, and E. Feron, "Mathematical Models for Aircraft Trajectory Design: A Survey," in *Proceedings of the 3rd ENRI International Workshop on ATM/CNS*, (Tokyo, Japan), 2013.

[5] X. Olive, J. Sun, M. C. R. Murça, and T. Krauth, "A Framework to Evaluate Aircraft Trajectory Generation Methods," in *Proceedings of the 14th USA/Europe Air Traffic Management Research and Development Seminar*, 2021.

[6] T. Krauth, J. Morio, X. Olive, B. Figuet, and R. Monstein, "Synthetic Aircraft Trajectories Generated with Multivariate Density Models," in *Proceedings of the 9th OpenSky Symposium*, (Brussels, Belgium), Nov. 2021.

[7] T. Krauth, A. Lafage, J. Morio, X. Olive, and M. Waltert, "Deep generative modelling of aircraft trajectories in terminal maneuvering areas," *Machine Learning with Applications*, vol. 11, p. 100446, Dec. 2022.

[8] T. Krauth, B. Figuet, X. Olive, and J. Morio, "Collision Risk Assessment in Terminal Manoeuvring Areas based on Trajectory Generation Methods," in *Proceedings of the 15th USA/Europe Air Traffic Management Research and Development Seminar*, (Savannah, GA), June 2023.

[9] A. Hodgkin, N. Pepper, and M. Thomas, "Probabilistic Simulation of Aircraft Descent via a Hybrid Physics-Data Approach," Apr. 2025.

[10] X. Gui, J. Zhang, X. Tang, D. Delahaye, and J. Bao, "A Novel Aircraft Trajectory Generation Method Embedded with Data Mining," *Aerospace*, vol. 11, p. 648, Aug. 2024.

[11] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural Ordinary Differential Equations," in *Proceedings of the 32nd Conference on Neural Information Processing Systems*, (Montréal, Canada), 2018.

[12] D. P. Kingma, M. Welling, *et al.*, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[13] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[14] J. Tomczak and M. Welling, "Vae with a vampprior," in *International conference on artificial intelligence and statistics*, pp. 1214–1223, PMLR, 2018.

[15] C. W. Fox and S. J. Roberts, "A tutorial on variational bayesian inference," *Artificial intelligence review*, vol. 38, pp. 85–95, 2012.

[16] J. Hoekstra and J. Ellerbroek, "Bluesky atc simulator project: an open data and open source approach," in *Proceedings of the 6th International Conference for Research in Air Transportation*, (Philadelphia, PA), 2016.