
AirTrafficGen: Configurable Air Traffic Scenario Generation with Large Language Models

Dewi Sid William Gould
The Alan Turing Institute
London, England, NW1 2DB
United Kingdom
dgould@turing.ac.uk

Benjamin Carvell
NATS
Whiteley, England, PO15 7FL
United Kingdom
benjamin.carvell@nats.co.uk

George De Ath
University of Exeter
Exeter, England, EX4 4QJ
United Kingdom
g.de.ath@exeter.ac.uk

Nick Pepper
The Alan Turing Institute
London, England, NW1 2DB
United Kingdom
npepper@turing.ac.uk

Abstract

The manual design of scenarios for Air Traffic Control (ATC) training is a demanding and time-consuming bottleneck that limits the diversity of simulations available to controllers. To address this, we introduce a novel, end-to-end approach, *AirTrafficGen*, that leverages large language models (LLMs) to automate and control the generation of complex ATC scenarios. Our method uses a purpose-built, graph-based representation to encode sector topology (including airspace geometry, routes, and fixes) into a format LLMs can process. Through rigorous benchmarking, we show that state-of-the-art models like Gemini 2.5 Pro, OpenAI o3, GPT-oss-120b and GPT-5 can generate high-traffic scenarios while maintaining operational realism. Our engineered prompting enables fine-grained control over interaction presence, type, and location. Initial findings suggest these models are also capable of iterative refinement, correcting flawed scenarios based on simple textual feedback. This approach provides a scalable alternative to manual scenario design, addressing the need for a greater volume and variety of ATC training and validation simulations. More broadly, this work showcases the potential of LLMs for complex planning in safety-critical domains.

1 Introduction

Air traffic control is a complex, safety-critical task that necessitates rigorous selection and training of new air traffic control officers [ATCOs, UK Civil Aviation Authority, 2024]. Trainee competency is assessed in simulations using handcrafted traffic scenarios designed to test specific skills: for example, recognising and resolving potential aircraft conflicts. The complexity of these scenarios is maintained at a level appropriate for training. Designing such scenarios is therefore demanding, time-consuming and expensive, requiring significant expert resource. This limitation restricts both the number and the diversity of training scenarios. These challenges apply equally to the construction of scenarios for validating proposed changes to operating procedures, forming a barrier to entry for effective airspace change [EUROCONTROL, 2014].

Controlled airspace is divided into geographical units known as sectors, each governed by sector-specific procedures. Hence, designing representative validation scenarios requires substantial domain

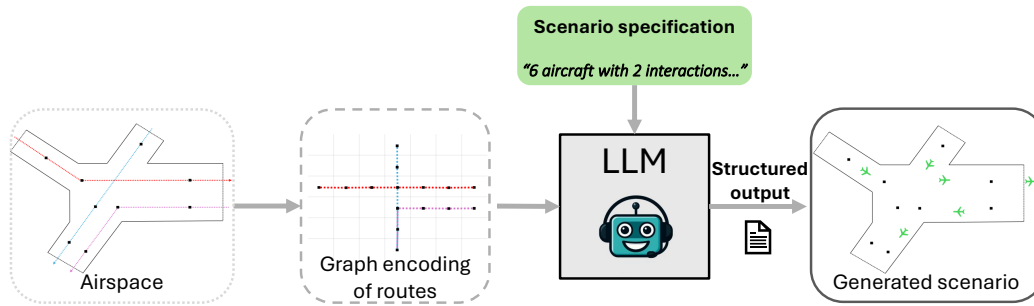


Figure 1: Overview of the AirTrafficGen framework for **fully configurable** scenario generation on **arbitrary** airspace geometries. Given a sector of airspace (a three-dimensional volume with prescribed routes on which aircraft can fly – denoted by coloured lines) and a **text-based scenario specification**, we engineer a prompt that encodes the relevant spatial information for an LLM to create a highly specific air traffic scenario. The framework converts the sector geometry into a grid-based graph, designed using key air traffic control safety length-scales. The output is in a structured JSON format, which we feed into a simulation environment.

expertise and sector-specific knowledge. Automating scenario generation in both contexts could markedly increase scenario quantity and diversity. Therefore, a method is needed to *controllably* generate new interacting scenarios while respecting existing route structures and sector procedures. This paper explores the novel application of Large Language Models [LLMs, Devlin et al., 2019, Radford et al., 2018] as a principled alternative to handcrafted scenario generation.

We present an end-to-end framework for fully configurable scenario generation: AirTrafficGen. A core novelty is our rigorous benchmarking of LLM capabilities on this complex task. First, we introduce a novel mapping from three-dimensional airspace sectors to discrete graphs, engineered to fit within an LLM’s context window. Within this setting, we systematically benchmark the reasoning capabilities of state-of-the-art LLMs across orthogonal subtasks required for scenario generation. Specifically, we evaluate competency along four reasoning axes: (1) **aircraft spatial density** – handling higher traffic loads; (2) **temporal reasoning** – varying scenario duration; (3) **sector complexity** – adjusting route interactivity; and (4) **interaction modelling** – engineering scenarios of differing complexity. Our benchmarks test LLM ability to generate scenarios with differing aircraft counts, durations, and sector complexities, providing granular insight into model strengths and limitations. Finally, we implement an end-to-end pipeline that demonstrates full controllability and practical utility in producing realistic, challenging air traffic scenarios. See Figure 1 for a concise overview.

The contributions of our paper are as follows:

- Novel benchmarking showcasing the strengths and limitations of LLMs in **spatial**, **temporal**, and **spatio-temporal** reasoning crucial for complex air traffic scenario generation.
- A purpose-built, **graph-based knowledge representation** that enables LLMs to ingest and reason over intricate spatio-temporal air traffic data.
- An **engineered-prompting framework** that provides fine-grained control over scenario characteristics including interaction presence, type and location.
- Empirical demonstration of the method’s applicability across diverse route configurations.

2 Background on Air Traffic Control

The role of an ATCO is to ensure the safe, orderly and expeditious transit of aircraft through their sector. Before each flight, the aircraft operator submits a flight plan specifying the route as a sequence of GPS waypoints, or *fixes*. This plan informs the sector ATCO of the aircraft’s intended lateral track and its requested *exit flight level* (the altitude at which it will leave the sector). Flight levels are

reported in hundreds of feet (e.g., FL 250 corresponds to 25,000ft). Figure 2a depicts a sector and its fixes. Because the sector contains only a finite set of fixes, the number of possible routes through it is also finite. The i^{th} aircraft within a scenario can be characterised by the following information: aircraft-type, spawn time, initial flight level, h_i , requested/exit flight level, e_i , and route.

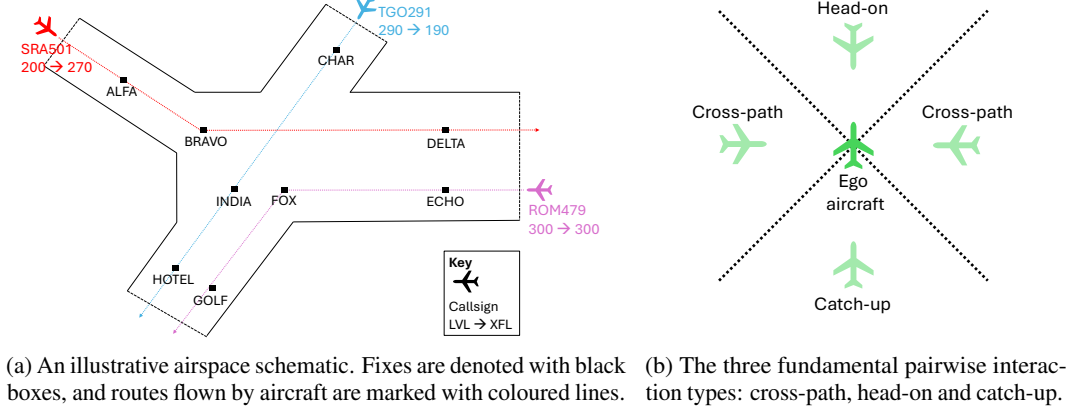


Figure 2: Overview of relevant airspace (left) and aircraft interaction types (right).

An ATCO’s overriding priority is to maintain separation minima between all aircraft in the sector. They must formulate deconfliction plans that remain robust even in the event of communication failure, greatly increasing task complexity. The task is further complicated by the presence of large epistemic uncertainties concerning aircraft performance (see, e.g., Pepper and Thomas, 2024). Aircraft pairs are deemed to be *relevant traffic* when the ATCO may need to issue instructions to assure safety, at ranges that are much greater than the separation standards for that airspace. No formal metric determines if two aircraft are relevant to one another; this depends on the ATCO’s judgement, the specific sector, and the wider operational context. For this reason, we define a *relevancy metric*, based on discussions with ATCOs. Under this metric, all aircraft in a scenario are non-interacting if, on their current trajectories, no pair with overlapping flight level ranges¹ come within 20 nautical miles (nmi) of one another. This threshold equates to a 2–3-minute look-ahead time depending on the ground speed of the aircraft, which is comparable to the timescales used in short-term conflict detection (see, e.g. Radanovic et al. 2018).

A *non-interacting* scenario is one where no aircraft are considered relevant to each other, allowing them to operate independently. Independent aircraft reduce complexity because instructions issued to one need not consider the safety of others. A constraint when designing air traffic scenarios is that an aircraft should not be relevant traffic for at least 2 minutes after entering the controlled sector. This reflects operations within a sectorised airspace, in which aircraft are passed between different sectors in such a way that they do not pose an immediate safety issue in the new sector [EUROCONTROL, 2023].

Interactive scenarios exhibit several distinguishing features. At a high level, specifying the number of interactions, and *types* of these interactions, is an effective measure of “interactivity” or scenario *complexity*. Interactions are classified into three types according to the configurations of the participating aircraft (see Figure 2b). At the level of classification, any multi-aircraft interaction can be decomposed into groups of pairwise interactions.

3 Related Work

Earlier studies generated air traffic scenarios by algorithmically re-working recorded data to insert interactions [Oaks and Paglione, 2002, Oaks et al., 2003] or by keeping a human in the loop [Signor et al., 2004]. More recently, Stefani et al. [2025] proposed an automated generator to validate machine learning conflict-resolution tools. Yet few studies create realistic air traffic scenarios from scratch in synthetic airspace.

¹The *flight level range* of an aircraft is defined as the range $(\min(h_i, e_i), \max(h_i, e_i))$.

In contrast, machine learning methods have been used to generate complex scenarios across domains as varied as economics [Flaig and Junike, 2022], healthcare [Arvanitis et al., 2022], energy [Dong et al., 2022], and transportation [Feng et al., 2023]. In transportation, scenario-based testing is used to assess autonomous driving systems Ding et al. [2023], Cai et al. [2022]. Several advanced, data-intensive methods such as diffusion [Xu et al., 2025] and generative adversarial networks [GANs, Demetriou et al., 2023] have been investigated to generate automotive scenarios. However, these data-intensive methods are poorly suited to our setting, where the synthetic airspace datasets for ATCO trainees are small.

Road-traffic scenarios resemble air traffic ones: both involve multiple vehicles interacting on procedurally constrained route structures. Road traffic is tightly constrained by lanes and rules, whereas aircraft have three-dimensional freedom as ATCOs can direct them off their filed routes. Safety definitions also differ: air traffic management employs high redundancy and conservative thresholds, making fine-grained controllability harder. Moreover, road-traffic scenarios typically revolve around a *single "ego-vehicle"*, whereas air traffic scenarios evaluate a controller’s handling of the *complete traffic flow*. Furthermore, road traffic scenarios are primarily benchmarked by collision rates involving an agent-controlled ego vehicle [Xu et al., 2022]. In contrast, air traffic scenarios lack such an agent, and safety is measured in a more subtle manner: the controlling technique of ATCOs is assessed across a diverse range of competencies based on different configurations of aircraft. This difference requires designing ATC scenarios with a unique degree of controllability over scenario characteristics.

A key requirement for testing automotive driving agents is the ability to generate automotive scenarios controllably, allowing the user to select scenarios from the same distribution as the training data or to adversarially generate scenarios. Hence, LLMs have recently been applied to the generation of road traffic scenarios [Zhang et al., 2024, Chang et al., 2024, Lu et al., 2024, Cai et al., 2025]. LLM-based generation therefore requires an interpretable encoding of the road network topology. In this paper, geometric information concerning sector routes is encoded in a graph-based representation, which is described in the next section.

4 Graph Representation of Air Traffic Control Scenarios

Continuous airspace and aircraft trajectories must be discretised for an LLM framework. The structure of our proposed discretisation is motivated by the length-scale used to determine whether aircraft are relevant traffic to one another.

To represent the spatial relationships between the fixes along routes efficiently for LLM consumption, we project the routes into a graph-based representation, where nodes of the graph are separated by 20 nmi. Figure 3 illustrates this process for two example routes, with Figure 3a displaying example routes prior to the encoding and Figure 3b shows the corresponding graph. The process can aggregate multiple fixes into a single graph node, which is a useful feature as sectors typically contain a high density of fixes in proximity.

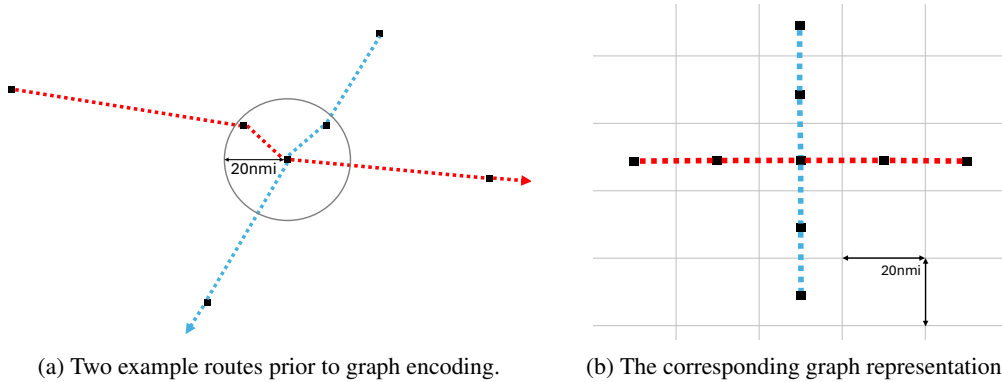


Figure 3: Converting sector routes to a graph representation. The three fixes encircled are all within 20 nmi, meaning that in the graph representation they will be projected onto the same node. The legs of the routes are interpolated with nodes every 20 nautical miles. Notice that we throw away unimportant kinks in the routes for maximal simplicity, retaining only the route lengths and intersections.

This method is motivated by the need to retain only essential information for configurable generation: route *length* (to simulate traversal time) and route *intersections* (to manage aircraft interactions). See [Google DeepMind and Kaggle, 2025] for some related work involving the encoding of spatial environments for LLMs within the context of games.

4.1 Aircraft Dynamics

Aircraft traverse the graph along their assigned routes. Each aircraft is classified as fast (one node at each time-step) or slow (one node every two time-steps) to reflect a wide range of performance characteristics present within controlled airspace [Hodgkin et al., 2025]. Slow speeds correspond to turboprops, and fast speeds to jet engines when converting our discretised scheme to any simulator. Due to the large discretisation length-scale and broad definition of relevant traffic, this simplification is designed to have a limited impact on the fidelity of our method. See Figure 4 for an example scenario, with Figure 8 showing the route structure of the synthetic sector in greater detail.

4.2 Interactions on the Graph

In the discrete model, an interaction occurs when (1) two aircraft occupy the same node simultaneously, or (2) two aircraft swap nodes in one time-step. By design, the graph construction ensures that no edges cross without meeting at a node; therefore, these two events capture *every* instance where aircraft pass within the 20 nmi threshold.

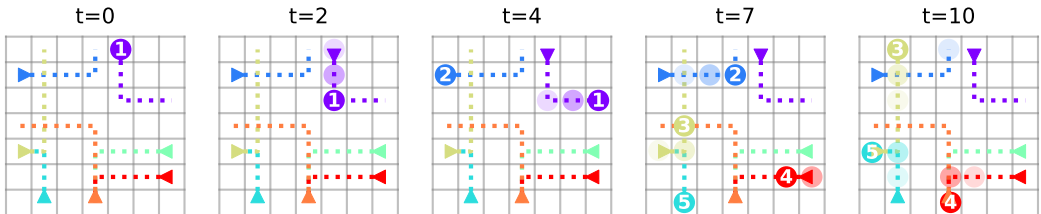


Figure 4: An example *non-interacting* scenario for the traffic volume benchmark. Aircraft spawn at the start of their routes, and move in discretised time units along the route, one grid cell at a time.

5 LLM Prompting Framework

This section details our multistep prompting framework. The full prompts used in this paper are detailed in the Supplementary Material. The LLM receives two inputs: 1) Specification: a text description of the required scenario, and 2) Sector geometry: permitted routes and their layout. Routes are converted to the graph formalism of Section 4.

The prompt first establishes the core task, defining the scenario rules and settings, and provides detailed instructions on how aircraft move and the definition of an interaction. This is followed by a “high-level strategy” section which acts as the core of the prompt. This explicitly directs the model to split the task into three phases: 1) Sector analysis (identify route intersections), 2) Aircraft placement strategy, and 3) Internal verification and iterative refinement.

The framework’s key feature is the third phase, where the model is prompted to internally verify its own scenario by rolling out trajectories to check for unintended interactions. We prompt the LLM to return a scenario in a fixed JSON format where each entry includes a spawn time (integer), a route (string identifier), and aircraft speed (1 or 2). The JSON files generated by the LLMs were formatted so that they could be parsed by BluebirdDT, a probabilistic digital twin of en route airspace [The Alan Turing Institute, 2024]. All scenario screenshots used in this work are generated using this twin.

6 Benchmarking Scenario Reasoning

In this section, we evaluate how well contemporary large language models perform each component of air traffic scenario generation. The task provides a natural testing ground for spatial reasoning and planning. To quantify these skills, we introduce four novel benchmarks.

To emulate the manual design process, where lateral conflicts are typically planned before vertical separation is assigned, our benchmarks are initially restricted to the 2D **lateral plane** ($h_i \equiv e_i \equiv H, \forall i$). This isolates the problem of creating or avoiding lateral conflicts before Section 7 introduces the vertical dimension. Handling large aircraft counts or dense interaction patterns demands strong spatial awareness, temporal reasoning and careful forward planning. Succeeding in this task requires simulating how aircraft move, tracking all possible interactions, and making tactical decisions about parameters to best avoid or create them.

6.1 Benchmark Suite

We begin with the fundamental spatial and temporal reasoning required to understand air traffic scenarios. Each benchmark asks the LLM to produce a *non-interacting* scenario that meets specific constraints. See Table 1 for the full list of models we benchmark in this work. To ensure statistical confidence in our results, we test every parameter set on **ten** synthetic sectors (Figure 4 shows one example). The code required to run these benchmarks, including these synthetic sectors, is included in the Supplementary Material. All benchmarks are **automatically verifiable**: after generating trajectories on the graph, we can directly count the number of interactions.

Traffic Volume Benchmark. For a fixed scenario length and sector complexity, we measure LLM performance at creating non-interacting scenarios with increasing numbers of aircraft. See Table 2 for the results of this benchmark.

$$N \in \{2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30\}, \quad T = 12, \quad N_{\text{routes}} = 7, \quad N_{\text{intersections}} = 7. \quad (1)$$

Here, N is the number of aircraft, T is the number of scenario time units, N_{routes} is the number of routes in the sector and $N_{\text{intersections}}$ is the number of graph nodes occupied by more than one route (a proxy for sector complexity). Figure 4 illustrates an example sector used in this benchmark.

Scenario Length Benchmark. For a fixed number of aircraft and sector complexity, we measure LLM performance at creating non-interacting scenarios of increasing length. See Table 3 for the results of this benchmark.

$$T \in \{12, 15, 18, 21, 24\}, \quad N = 8, \quad N_{\text{routes}} = 7, \quad N_{\text{intersections}} = 7. \quad (2)$$

Sector Complexity Benchmark. Next, we design a benchmark to focus on measuring the capability of models to handle sectors of increasing intrinsic complexity (measured by the number of intersection points of the routes in the sector). We do this for a fixed number of aircraft and scenario length. See Table 4 for the results of this benchmark.

$$N_{\text{intersections}} \in [4, 14], \quad N = 8, \quad T = 12, \quad N_{\text{routes}} = 7. \quad (3)$$

Success is measured by the **mean number of unique pairs of interacting aircraft** (MUIP) across the 10 synthetic sectors; the ideal value is zero. All models are compared with a random baseline score which is computed by averaging over 500 scenarios created by sampling spawn times, routes and speeds from their valid sets uniformly. In Figures 11 and 12, we give two examples of Gemini-2.5-Pro’s responses on the $N = 30$ traffic volume benchmark, on two different sectors.

Lastly, we design a **controllability** benchmark, which assesses an LLM’s ability to construct a target number of interactions. We use the parameters

$$T = 12, \quad N = 10, \quad N_{\text{routes}} = 7, \quad N_{\text{intersections}} = 7, \quad (4)$$

and prompt for $\{1, 2, 3, 4, 5\}$ *unique* interacting pairs. Success is gauged by the **mean absolute difference between the number of unique interacting pairs and the input number of interacting pairs** (MADIP) across the 10 synthetic sectors. See Table 5 and Figure 9 for the results of this benchmark.

All experiments used the OpenRouter inference platform OpenRouter [2025]. Specific prompts for each benchmark are given in the Supplementary Material. For all experiments, we used a temperature of 1.0, $\text{top_p} = 1.0$, $\text{top_k} = 0.0$, and a maximum token budget of 35,000. For each task, we use the first scenario generated by the LLM that satisfies the required format, rather than generating multiple candidates. If a model failed to produce a valid output due to token limits, the token budget was increased in steps of 10,000 until success. No model required more than 50,000 tokens to produce a valid response on all benchmarks. The combined cost of all the experiments carried out in this paper was under \$150 USD.

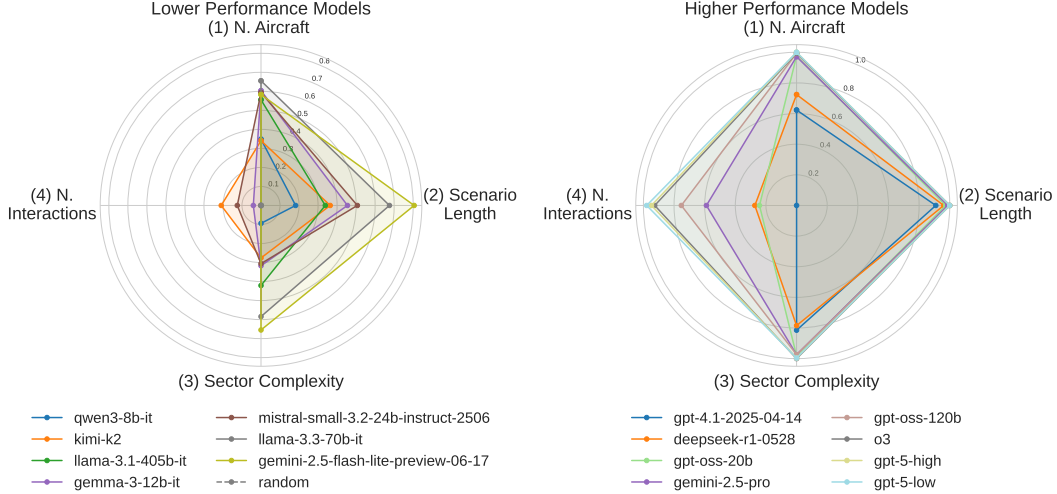


Figure 5: LLM capabilities across four benchmarked axes.

6.2 Benchmarking Results

The complete benchmark results appear in Appendix C. Figure 5 summarises model capabilities; here *ability* is measured as a normalised score $\in [0, 1]$ relative to the mean score of the random baseline (labelled with the *rand* subscript). The scores are calculated as follows:

$$\mu_{1,2,3} = \left[1 - \frac{\text{MUIP}}{\text{MUIP}_{\text{rand}}} \right]_+, \quad \mu_4 = \left[1 - \frac{\text{MADIP}}{\text{MADIP}_{\text{rand}}} \right]_+, \quad (5)$$

where $[x]_+ = \max(0, x)$, a value of 0 matches or underperforms the random baseline, and 1 corresponds to perfect performance.

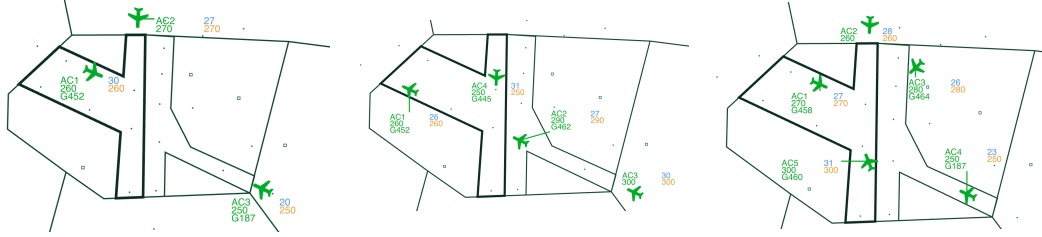
We observe considerable variation in model performance. Interestingly, performance across capability dimensions is not fully correlated. For example, mistral-small-24b-it handles scenario length better than llama-405b-it, but not sector complexity. The most advanced models, GPT-5, o3 and Gemini-2.5-Pro, are the most effective, with GPT-5 performing near-optimally across all benchmarks, whereas Gemini-2.5-Pro falters at the upper end of the interaction task. We also observe strong performance of the open-source GPT-oss models, in particular with the 120b variant outperforming most advanced proprietary models. This result is clearly demonstrated in Appendix C, Figure 10, where we compare model performance against cost - plotting the Pareto frontier.

We observe that most models struggle to perform better than random at the fourth benchmark. It should be noted that the latter elements of the controllability benchmark, involving generating 4 or 5 interactions in a scenario with 10 aircraft, are incredibly difficult and go beyond the scope of design control required to make realistic air traffic scenarios. Moreover, the full task includes the vertical dimension, giving designers an extra degree of freedom to alleviate congestion.

7 Configurable and Flexible Air Traffic Scenario Generation

We have observed that some advanced reasoning models are capable of designing high-traffic non-interacting scenarios and produce promising results when prompted to generate interactions in the lateral plane. We therefore turn to the central requirement of realistic simulation: **fine-grained controllability**. To meet the full complexity and fidelity demanded by operational air traffic control simulations, we add the vertical dimension: each aircraft now carries an initial flight level and a requested exit flight level in the structured JSON output. Human controllers find the task markedly easier once the vertical dimension is available, because potential lateral conflicts can be resolved by altitude separation.

This section demonstrates the breadth of controllable parameters that scenario designers can exploit when generating air traffic scenarios. For illustration, we employ Gemini-2.5-Pro to generate a wide



(a) Generate a scenario with four aircraft. There should be two aircraft which interact in a cross-path manner. The remaining two aircraft should not interact with anything. (b) Generate a scenario with four aircraft. Three aircraft should follow one another in trail, but not interacting. The fourth should be a climber which climbs through all the levels of the three in trail. (c) Generate a scenario with six aircraft. There should be an interaction in which three aircraft are involved. All other aircraft should be non-interacting.

Figure 6: Three examples of fine-grained scenario controllability using AirTrafficGen. Each sub-caption details the prompt used. Images are generated using BluebirdDT, where labels next to each aircraft show its ground speed (GXXX, in knots), initial flight level (in orange), and exit flight level (in blue). The scenarios show: (a) an interaction between AC1 and AC2, while a fourth aircraft is yet to spawn; (b) AC1, AC2, and AC3 in trail, while AC4 climbs through their levels and across their route as requested; and (c) a triple interaction created using AC1, AC2, and AC3.

spectrum of specific scenarios. Our testing includes fine-grained instructions with detailed requirements including number of aircraft, conflicts (including location, time and type), as well as general traffic patterns (including features like “in trail” or “climbing through levels”). A further strength is the ability to **adapt existing scenarios** to new requirements. This includes adding new aircraft or modifying the parameters of existing aircraft to precisely tailor a scenario to specific requirements. This adaptability highlights the flexibility and operational relevance of AirTrafficGen.

To illustrate the control offered by AirTrafficGen, we provide concrete examples:

1. **Pairwise Interactions:** Appendix E and Figure 13 show precisely engineered scenarios with all three fundamental pairwise interactions: cross-path, head-on, and catch-up.
2. **Sophisticated Controllability:** Figure 6 presents three complex scenarios that involve a higher number of aircraft, intricate and detailed instructions, and combinations of various control parameters.
3. **Scenario Modification:** Finally, in Figure 7, we provide an example of controllability in terms of *modifying an existing scenario*. This demonstrates how our method can take a pre-existing traffic scenario and adapt it to new requirements, such as increasing its complexity or introducing new elements.

The method is both powerful and generalisable across sectors and operational contexts, and it directly addresses the laborious and time-intensive nature of handcrafting scenarios. Although the full complexity of high-traffic scenarios is challenging to convey in static images, the provided screenshots with lower traffic densities effectively demonstrate the precise and controlled generation capabilities of our method. It is worth stressing that the end-to-end pipeline, including the graph-based knowledge representation, LLM-driven generation, and simulation of scenarios within BluebirdDT, delivers a complete and fully functional system for controllable air traffic scenario generation.

Furthermore, we observed that advanced reasoning models can respond to corrective feedback, refining a scenario based on its evaluated outcome; see Appendix D. In particular, when asked to generate *non-interacting* scenarios, we observe that, given the ability to respond to automatically generated feedback highlighting any interacting aircraft pairs that violate the requirement, several models achieve *perfect performance* on the varying number of aircraft benchmark. For example, in Figure 12 we show one of the $N = 30$ traffic volume benchmarks that Gemini-2.5-Pro fails to solve. Given the feedback on *how it failed*, the model was able to adapt its solution and pass the benchmark. For verifiable scenario specifications (e.g., the specific number of aircraft or interactions), this feedback mechanism is automatic and scalable.

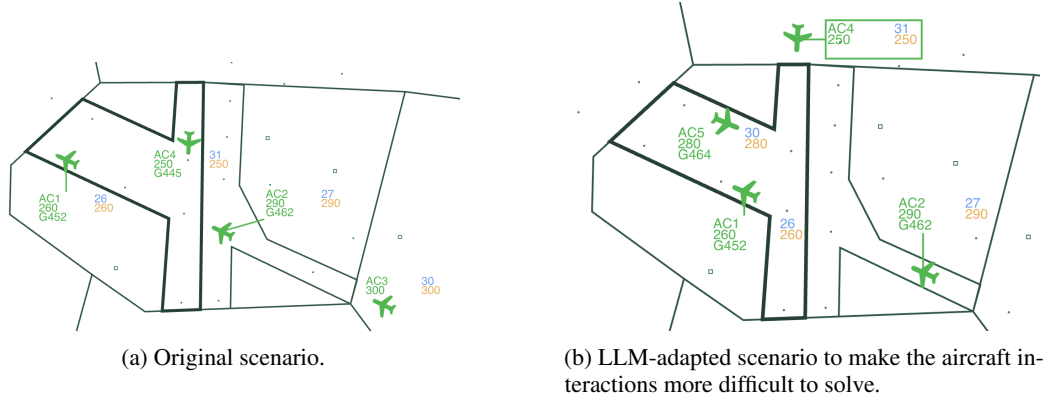


Figure 7: When prompted to make the interaction in (a) harder to solve by adding a new aircraft, the LLM has added AC5. AC5 has flight level overlap with AC2 and AC4, adding to the complexity of the scenario. An ATCO may solve (a) by climbing AC4 to an intermediate (but still safe) level of 280. However, the presence of AC5 at this level blocks this potential solution.

8 Discussion

This paper introduces *AirTrafficGen*, a novel, end-to-end framework for configurable generation of air traffic scenarios, leveraging advanced reasoning capabilities of large language models. It offers a systematic, scalable alternative to the labour-intensive process of handcrafting scenarios. By encoding the three-dimensional problem into a graph-based representation that LLMs can process, we achieve fine-grained control over the characteristics of scenarios.

Our benchmarking methodology provides granular insights into LLM performance across crucial spatial, temporal and interaction reasoning axes. Our graph representation ensures that every generated scenario can be automatically rolled out and verified by counting interactions. State-of-the-art models such as Gemini-2.5-Pro and OpenAI’s o3, GPT-oss and GPT-5 models reliably create high-traffic non-interacting scenarios and precisely engineer diverse interaction types, including pairwise conflicts and more complex multi-aircraft interactions. In the future, given the automatically verifiable nature of our benchmarks, it would be interesting to study the effect of recent methods in prompt optimisation Agrawal et al. [2025].

A key strength of our method is its ability to **adapt existing scenarios**, tailoring them to specific instructions by adding new aircraft or modifying existing scenario data. This flexibility, along with sector-agnostic deployment, supports rapid prototyping and the creation of varied, challenging scenarios. Section 7 demonstrated fine-grained controllability in a range of examples. Future work will include quantitative human-in-the-loop trials in which human ATCOs assess how well generated scenarios meet more nuanced specifications.

Looking ahead, future work will explore the integration of more complex operational elements, such as **holding patterns**, standard airport terminal departure and arrival procedures (SIDs and STARs), and the dynamic introduction of **unexpected events**. Investigating mechanisms for human feedback and editing within the generation loop could provide valuable refinements, combining an LLM’s generative power with expert domain knowledge.

In conclusion, this research demonstrates the remarkable potential of LLMs in complex planning tasks within air traffic control. The framework overcomes the limits of traditional scenario generation and creates an opportunity for more efficient, adaptable, and diverse simulations for ATCO training and operational validation.

Acknowledgments and Disclosure of Funding

This work was supported by the EPSRC grant EP/V056522/1 (aka Project Bluebird), an EPSRC Prosperity Partnership between NATS, The Alan Turing Institute and the University of Exeter.

References

- Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziemis, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. Gepa: Reflective prompt evolution can outperform reinforcement learning, 2025. URL <https://arxiv.org/abs/2507.19457>.
- Theodoros N Arvanitis, Sean White, Stuart Harrison, Rupert Chaplin, and George Despotou. A method for machine learning generation of realistic synthetic datasets for validating healthcare applications. *Health Informatics Journal*, 28(2):14604582221077000, 2022.
- Jinkang Cai, Weiwen Deng, Haoran Guang, Ying Wang, Jiangkun Li, and Juan Ding. A survey on data-driven scenario generation for automated vehicle testing. *Machines*, 10(11), 2022. ISSN 2075-1702. doi: 10.3390/machines10111101.
- Xuan Cai, Xuesong Bai, Zhiyong Cui, Danmu Xie, Daocheng Fu, Haiyang Yu, and Yilong Ren. Text2scenario: Text-driven scenario generation for autonomous driving test, 2025. URL <https://arxiv.org/abs/2503.02911>.
- Cheng Chang, Siqi Wang, Jiawei Zhang, Jingwei Ge, and Li Li. Llmscenario: Large language model driven scenario generation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(11):6581–6594, 2024. doi: 10.1109/TSMC.2024.3392930.
- Google DeepMind. Gemini 2.5 flash-lite. Released June 17, 2025; model card on Google Cloud, 2025a. Stable version publicly available.
- Google DeepMind. Gemini 2.5 pro. Released June 17, 2025; Technical report available on arXiv, 2025b. arXiv:2507.06261, model card on Google Cloud.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- Andreas Demetriou, Henrik Alfvåg, Sadegh Rahrovani, and Morteza Haghiri Chehrehghani. A deep learning framework for generation and analysis of driving scenario trajectories. *SN Computer Science*, 4(3):251, 2023.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.
- Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. A survey on safety-critical driving scenario generation—a methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):6971–6988, 2023. doi: 10.1109/TITS.2023.3259322.
- Wei Dong, Xianqing Chen, and Qiang Yang. Data-driven scenario generation of renewable energy production based on controllable generative adversarial networks with interpretability. *Applied Energy*, 308:118387, 2022.
- EUROCONTROL. Validation & Implementation Considerations. *European Airspace Concept Workshops for PBN Implementation*, 2014. URL https://www.icao.int/MID/Documents/2014/PBN%20Workshop-Tunis/14%20Validation%20and%20Implementation_vJUL13.pdf.
- EUROCONTROL. Guidelines for the coordination and transfer of control in atc, 2023. URL <https://www.eurocontrol.int/publication/guidelines-coordination-and-transfer-control-atc>.
- Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3567–3575. IEEE, 2023.

- Solveig Flaig and Gero Junike. Scenario generation for market risk models using generative neural networks. *Risks*, 10(11), 2022. ISSN 2227-9091. doi: 10.3390/risks10110199.
- Gemma Team. Gemma 3 technical report, 2025. URL <https://arxiv.org/abs/2503.19786>.
- Google DeepMind and Kaggle. Kaggle game arena: A benchmarking platform where ai models compete in strategic games. <https://blog.google/technology/ai/kaggle-game-arena/>, August 2025.
- Amy Hodgkin, Nick Pepper, and Marc Thomas. Probabilistic simulation of aircraft descent via a hybrid physics-data approach, 2025. URL <https://arxiv.org/abs/2504.02529>.
- Qiuqing Lu, Xuanhan Wang, Yiwei Jiang, Guangming Zhao, Mingyue Ma, and Shuo Feng. Multimodal large language model driven scenario testing for autonomous vehicles. *arXiv preprint arXiv:2409.06450*, 2024.
- Meta AI. Llama 3.3 model card. https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/, 2024a.
- Meta AI. Meta-Llama-3.1-405B-Instruct model card. <https://huggingface.co/meta-llama/Meta-Llama-3.1-405B-Instruct>, 2024b.
- Mistral AI. Mistral-small-24b-instruct-2501 model card. <https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501>, 2025.
- Moonshot AI. Kimi K2: Open agentic intelligence, 2025. URL <https://arxiv.org/abs/2507.20534>.
- R. Oaks, Shurong Liu Shurong Liu, D. Zhou, M. Paglione, and W.J. Hughes. Methodology for the generation of air traffic scenarios based on recorded traffic data. In *Digital Avionics Systems Conference, 2003. DASC '03. The 22nd*, volume 1, pages 5.C.3–5.1–7 vol.1, 2003. doi: 10.1109/DASC.2003.1245861.
- R.D. Oaks and M. Paglione. Generation of realistic air traffic scenarios using a genetic algorithm. In *Proceedings. The 21st Digital Avionics Systems Conference*, volume 1, pages 2A1–2A1, 2002. doi: 10.1109/DASC.2002.1067908.
- OpenAI. Gpt-4.1. Released April 14, 2025, 2025a. Available via OpenAI API: <https://openai.com/index/gpt-4-1/>.
- OpenAI. Openai o3 model, 2025b. <https://openai.com>.
- OpenAI. gpt-oss-120b and gpt-oss-20b Model Card. https://cdn.openai.com/pdf/419b6906-9da6-406c-a19d-1bb078ac7637/oai_gpt-oss_model_card.pdf, Aug 2025a.
- OpenAI. Introducing GPT-5. <https://openai.com/index/introducing-gpt-5/>, Aug 2025b.
- OpenRouter. OpenRouter: Unified API for LLMs. <https://openrouter.ai>, 2025.
- Nick Pepper and Marc Thomas. Learning generative models for climbing aircraft from radar data. *Journal of Aerospace Information Systems*, 21(6):474–481, 2024. doi: 10.2514/1.I011359.
- Qwen Team. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- Marko Radanovic, Miquel Angel Piera Eroles, Thimjo Koca, and Juan Jose Ramos Gonzalez. Surrounding traffic complexity analysis for efficient and stable conflict resolution. *Transportation Research Part C: Emerging Technologies*, 95:105–124, 2018. ISSN 0968-090X. doi: 10.1016/j.trc.2018.07.017.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.
- David Signor, Paul Davis, Sandra Lozito, Anthony Andre, Doug Sweet, and Erin Wallace. Efficient air traffic scenario generation. In *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, 2004. doi: 10.2514/6.2004-6399.

- T. Stefani, J. M. Christensen, A. A. Giriya, et al. Automated scenario generation from operational design domain model for testing ai-based systems in aviation. *CEAS Aeronautical Journal*, 16: 197–212, 2025. doi: 10.1007/s13272-024-00772-4.
- The Alan Turing Institute. Project Bluebird: An AI system for air traffic control, 2024. URL <https://www.turing.ac.uk/research/research-programmes/project-bluebird>.
- UK Civil Aviation Authority. Cap2331: Air traffic controllers – licensing and training, 2024. URL <https://www.caa.co.uk/publication/download/21022>.
- Chejian Xu, Wenhao Ding, Weijie Lyu, Zuxin Liu, Shuai Wang, Yihan He, Hanjiang Hu, Ding Zhao, and Bo Li. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. *Advances in Neural Information Processing Systems*, 35:25667–25682, 2022.
- Chejian Xu, Aleksandr Petiushko, Ding Zhao, and Bo Li. Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(8):8797–8805, Apr. 2025. doi: 10.1609/aaai.v39i8.32951.
- Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15459–15469, 2024.

A Example synthetic sector for benchmarking experiments

Figure 8 illustrates the route structures in one of the synthetic sectors used for benchmarking.

Set of Allowed Routes Through Sector (with directionality)

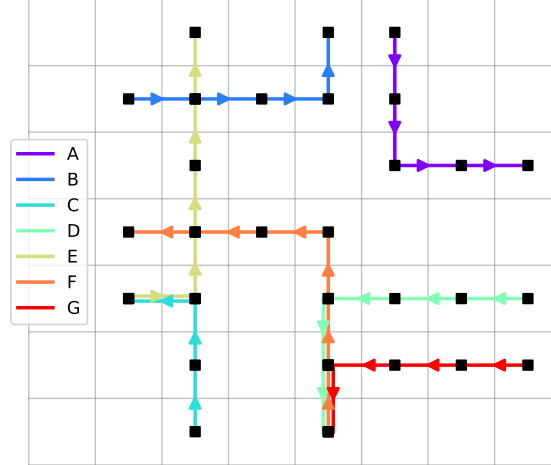


Figure 8: One of the synthetic sectors used in the benchmarking.

B Full List of Models

We provide a full list of models included in our benchmarking experiments in Table 1.

Table 1: The models used in the benchmarking experiments and whether they support extended reasoning or thinking modes. Where not stated, the model versions used were the latest available models on OpenRouter as of 12th August 2025. Pricing data reflects the *average* cost per million output tokens on the OpenRouter platform as of 12th August 2025.

Model	Reasoning	\$/M Output Tokens
qwen-3-8b-it Qwen Team [2025]	✓	0.138
mistral-small-3.2-24b-instruct-2506 Mistral AI [2025]	–	0.20
llama-3.3-70b-it Meta AI [2024a]	–	0.23
gpt-oss-20b OpenAI [2025a]	✓	0.28
gemma-3-12b-it Gemma Team [2025]	–	0.33
gemini-2.5-flash-lite-preview-06-17 DeepMind [2025a]	✓	0.40
gpt-oss-120b OpenAI [2025a]	✓	0.48
llama-3.1-405b-it Meta AI [2024b]	–	2.51
deepseek-r1-0528 DeepSeek-AI [2025]	✓	2.55
kimi-k2 Moonshot AI [2025]	–	2.58
gpt-4.1-2025-04-14 OpenAI [2025a]	–	8.0
o3 OpenAI [2025b]	✓	8.0
gpt-5 OpenAI [2025b]	✓	10.0
gemini-2.5-pro DeepMind [2025b]	✓	12.50

C Benchmark Results

In Tables 2, 3, 4 and 5 we present the benchmarking results for the varying scenario traffic volume, length, sector complexity, and interaction number benchmarks, respectively.

Table 2: Benchmarking Results for varying number of aircraft. Values quoted are the average number of unique interacting pairs of aircraft computed across 10 generated scenarios on 10 synthetic sectors. Optimal behaviour (marked with a checkmark) is zero interacting pairs.

Number of Aircraft	2	3	4	5	6	7	8	9	10	15	20	25	30
random	0.1	0.2	0.4	0.6	0.9	1.4	1.8	2.3	2.9	6.7	12.2	19.3	28.4
kimi-k2	✓	0.1	0.2	0.2	0.3	1.1	0.8	1.5	0.8	4.4	9.6	11.1	20.8
gpt-4.1-2025-04-14	✓	✓	✓	0.1	0.1	0.2	✓	0.3	0.4	2.7	3.6	5.7	16.0
gemini-2.5-flash-lite-preview-06-17	✓	0.1	✓	0.2	0.2	0.7	0.3	0.6	1.1	2.2	5.6	6.6	14.6
qwen3-8b-it	✓	0.1	0.1	0.6	0.6	0.9	2.0	2.0	1.2	6.2	8.4	17.3	12.2
llama-3.3-70b-it	✓	✓	✓	0.1	0.3	1.1	1.3	1.0	1.1	2.9	4.7	2.9	11.2
llama-3.1-405b-it	✓	0.1	0.1	0.2	0.4	1.2	0.9	1.5	1.5	3.5	8.0	6.6	11.0
deepseek-r1-0528	✓	✓	✓	✓	✓	0.1	0.2	✓	0.4	1.3	4.0	6.1	9.2
gemma-3-12b-it	✓	✓	0.1	0.1	0.3	0.4	0.8	0.9	1.5	5.4	3.3	8.8	9.0
mistral-small-3.2-24b-instruct-2506	✓	✓	0.2	✓	0.5	1.1	0.7	1.1	1.3	4.2	7.0	9.9	5.4
gemini-2.5-pro	✓	✓	✓	✓	✓	✓	✓	✓	0.1	0.6	0.3	0.3	0.9
gpt-oss-20b	✓	✓	✓	✓	✓	✓	✓	✓	0.1	0.1	0.7	1.0	0.2
o3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0.2	0.2	✓
gpt-oss-120b	✓	✓	✓	✓	✓	0.1	0.1	✓	0.2	✓	0.1	✓	✓
gpt-5-low	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
gpt-5-high	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3: Benchmarking results for varying scenario length. Values quoted are the average number of unique interacting pairs of aircraft computed across 10 generated scenarios on 10 synthetic sectors. Optimal behaviour (marked with a checkmark) is zero interacting pairs.

Scenario Length	12	15	18	21	24
random baseline	1.8	1.7	1.6	1.4	1.2
qwen3-8b-it	1.3	1.2	1.0	1.7	1.2
kimi-k2	1.0	1.6	1.0	0.4	0.9
llama-3.1-405b-it	1.3	0.9	1.3	0.7	0.9
gemma-3-12b-it	0.6	0.9	0.9	1.2	0.6
mistral-small-3.2-24b-instruct-2506	0.6	1.0	0.7	1.0	0.5
gemini-2.5-flash-lite-preview-06-17	0.4	0.4	0.3	0.2	0.2
gpt-4.1-2025-04-14	✓	0.4	0.1	0.1	0.1
llama-3.3-70b-it	0.8	0.4	0.9	0.3	0.1
gemini-2.5-pro	✓	✓	✓	✓	0.1
deepseek-r1-0528	0.2	✓	0.1	✓	✓
gpt-oss-20b	0.1	✓	✓	0.1	✓
o3	✓	✓	✓	✓	✓
gpt-oss-120b	✓	✓	✓	✓	✓
gpt-5-low	✓	✓	✓	✓	✓
gpt-5-high	✓	✓	✓	✓	✓

Figure 9 summarises the controllability benchmark: box plots of the number of interactions each model generates at different targets, revealing systematic under- and over-generation. While o3 stays close to the targets in all settings, Gemini-2.5-Pro struggles with higher targets. Most models increase conflicts as the target increases, but do not reliably match the requested counts, indicating limited fine-grained control.

Figure 10 compares the overall skill level across the four benchmarks (computed as the sum of the normalised skills) against inference cost. The dotted red line represents the Pareto front. It is particularly interesting to observe that the GPT-oss models achieve close to optimal scores at a fraction of the cost.

C.1 Case Study: High-Volume Scenario Generation ($N = 30$)

We present two Gemini-2.5-Pro outputs on the $N = 30$ traffic-volume benchmark (Figures 11 and 12). The model is successful in Figure 11 and fails in Figure 12, illustrating both the complexity of the task and, in the success case, the strategy employed by a reasoning-capable model.

Table 4: Benchmarking results for increasing sector complexity. Values quoted are the average number of unique interacting pairs of aircraft computed across 10 generated scenarios on 10 synthetic sectors. Optimal behaviour (marked with a checkmark) is zero interacting pairs.

Sector Route Intersections	4	5	6	7	8	9	10	11	12	13	14
random	1.9	1.9	1.9	2.3	2.3	2.7	2.5	2.7	2.9	2.7	3.1
qwen3-8b-it	1.3	1.6	1.2	2.3	1.4	2.4	2.3	2.6	3.8	3.5	3.1
kimi-k2	1.2	1.6	0.9	1.4	1.7	2.0	2.0	1.4	2.1	2.2	3.0
mistral-small-3.2-24b-instruct-2506	1.0	0.7	1.1	1.4	1.9	1.8	1.9	1.9	2.3	1.8	2.9
gemma-3-12b-it	1.2	1.2	0.8	1.7	0.9	1.5	1.1	2.1	2.3	2.8	2.8
llama-3.1-405b-it	0.5	1.2	0.6	1.0	0.9	1.9	1.8	1.8	1.6	1.9	2.4
deepseek-r1-0528	✓	✓	0.1	0.6	0.8	0.7	0.2	0.4	0.8	0.5	1.7
gemini-2.5-flash-lite-preview-06-17	0.6	0.9	0.4	0.2	0.4	0.9	0.9	1.2	1.0	1.3	1.5
llama-3.3-70b-it	1.0	0.5	0.4	1.0	1.2	0.8	1.2	0.7	1.7	1.5	1.2
gpt-4.1-2025-04-14	0.3	0.1	0.2	0.5	0.4	0.7	0.2	0.2	0.9	0.7	0.8
gemini-2.5-pro	✓	✓	✓	✓	✓	✓	✓	0.1	0.5	0.1	✓
gpt-oss-20b	✓	✓	✓	0.1	0.1	0.1	0.1	✓	0.1	✓	✓
o3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
gpt-oss-120b	✓	✓	✓	0.2	✓	0.1	✓	✓	0.2	0.3	✓
gpt-5-low	✓	✓	✓	✓	✓	✓	✓	✓	0.1	✓	✓
gpt-5-high	✓	✓	✓	✓	✓	✓	✓	✓	✓	0.1	✓

Table 5: Benchmarking results for increasing number of input conflicts. Values quoted are the mean absolute difference between the number of generated interacting pairs of aircraft and the input number. This average is computed across 10 generated scenarios on 10 synthetic sectors. Checkmarks denote perfect performance across all 10 synthetic sectors (a mean absolute difference of zero).

Number of Interactions	1.0	2.0	3.0	4.0	5.0
random baseline	1.95	1.55	1.49	1.92	2.58
llama-3.3-70b-it	1.30	1.30	1.70	2.20	4.20
deepseek-r1-0528	0.30	0.60	0.90	1.20	3.90
openai/gpt-oss-20b	✓	✓	1.10	3.00	3.80
gpt-4.1-2025-04-14	0.50	2.20	1.40	2.40	3.40
gemini-2.5-flash-lite-preview-06-17	1.90	2.00	3.30	1.20	3.00
gemma-3-12b-it	1.20	1.20	1.60	2.20	2.90
llama-3.1-405b-it	0.90	1.50	2.20	2.10	2.80
qwen3-8b-it	1.80	1.90	2.10	2.20	2.80
mistral-small-3.2-24b-instruct-2506	0.90	1.50	1.50	1.90	2.50
kimi-k2	0.90	1.20	1.30	2.20	1.90
gemini-2.5-pro	✓	0.10	0.60	1.50	1.70
openai/gpt-oss-120b	✓	0.50	0.50	1.60	0.80
o3	✓	✓	✓	0.40	0.30
gpt-5-high	✓	✓	0.10	0.20	0.20
gpt-5-low	✓	✓	0.10	✓	0.10

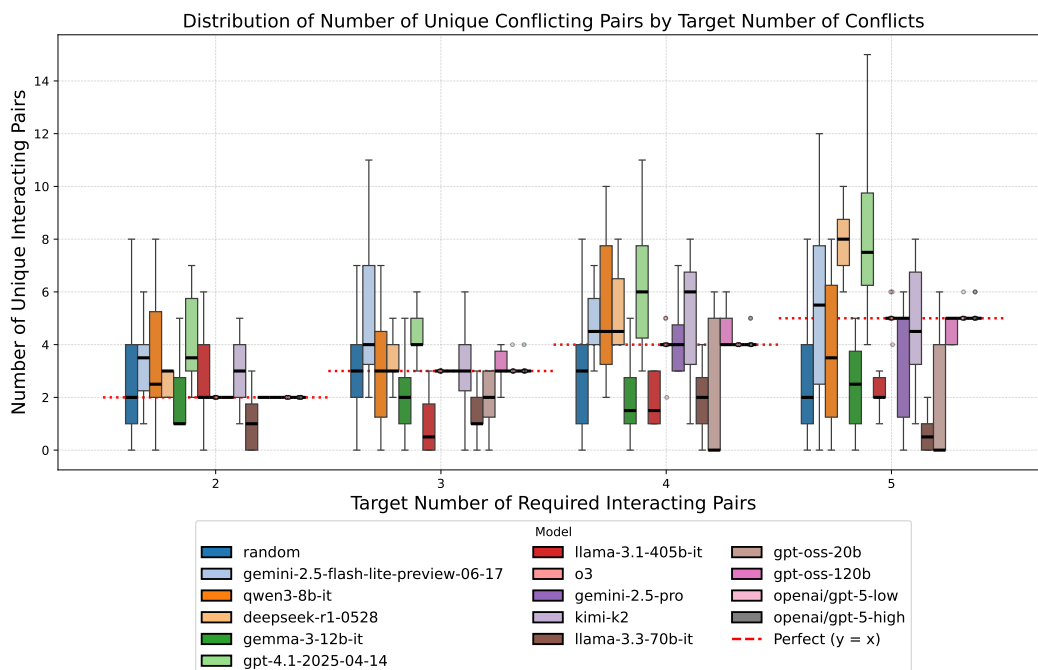


Figure 9: Benchmarking results for increasing number of input conflicts. o3 and GPT-5 saturate the metric, and therefore we include their individual data points to maintain readability.

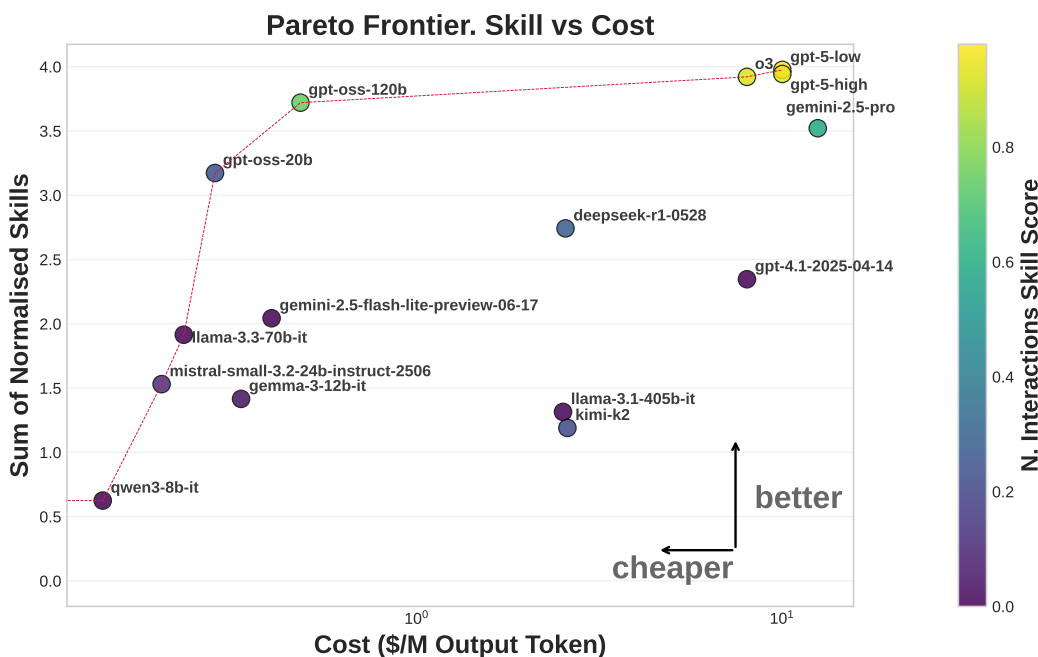


Figure 10: Comparing sum of normalised skills across all four benchmarks against inference cost for all models. The red dotted line represents the Pareto Frontier: lying on this line means that there is no model which is both cheaper and more skilful.

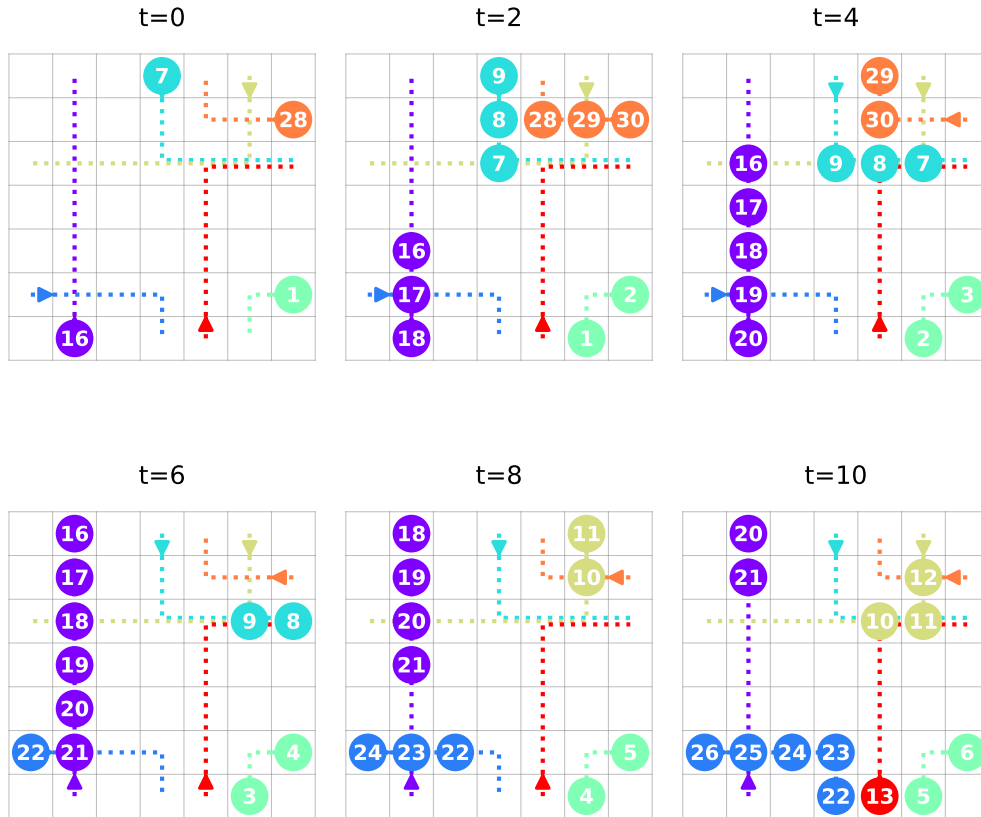


Figure 11: Gemini-2.5-Pro **solving** the $N = 30$ benchmark: there are no interactions in this scenario. It devises a strategy where aircraft are systematically spawned in sequences along the same route, thus avoiding conflicts. Sequences of aircraft on intersecting routes avoid one another by careful selection of spawn times. (Snapshots shown every two time units for convenience).

D Feedback Loop Experiments

In this section, we study the effect of giving models access to feedback on their attempts. We evaluate the four best-performing models from the varying number of aircraft benchmark (excluding GPT-5, which saturated the benchmark). In Table 6, we present their performance *after receiving feedback about their performance*. Using our automated verifier, we give each model five scored attempts to produce a non-interacting scenario. Specifically, we inform the model which aircraft are involved in interactions when it fails to create a non-interacting scenario. We observe that **in all cases, the models achieve perfect performance after a single round of feedback**.

Furthermore, it is interesting to note that the models achieve this by making *significant* changes to the scenario. In all cases, they modify the parameters of multiple aircraft, not just those involved in the highlighted interactions.

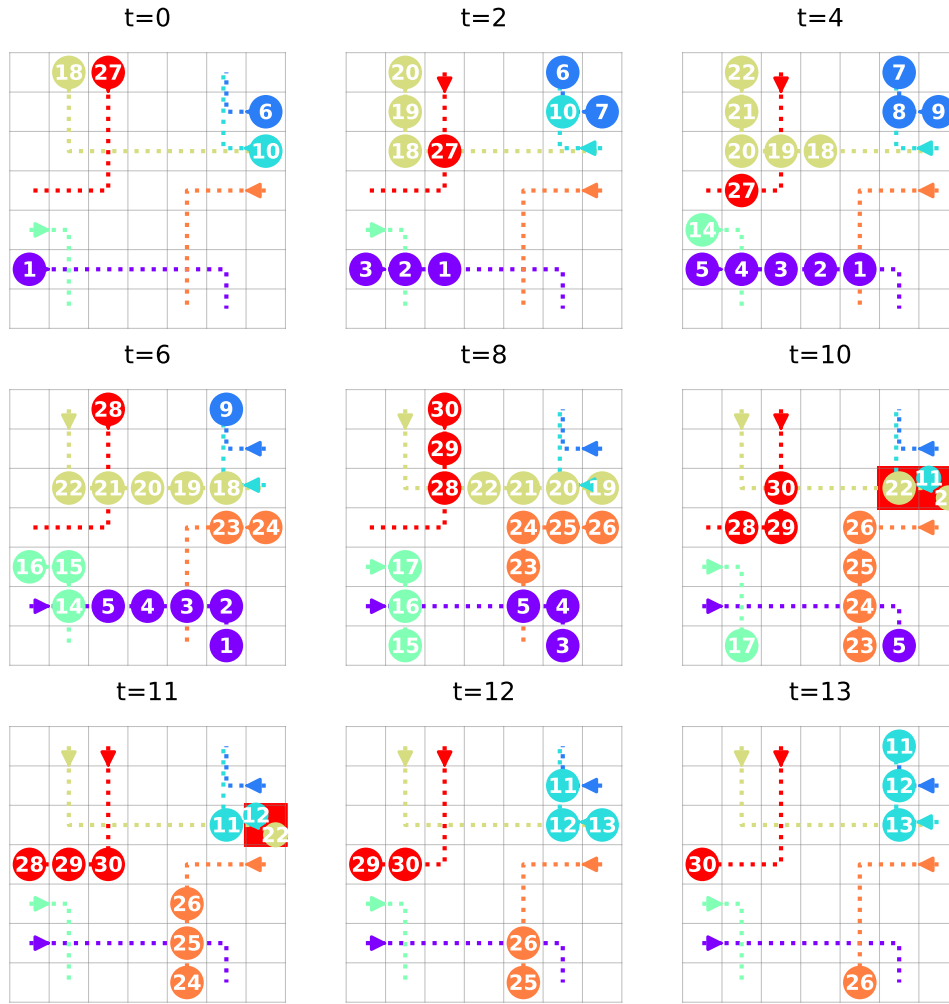


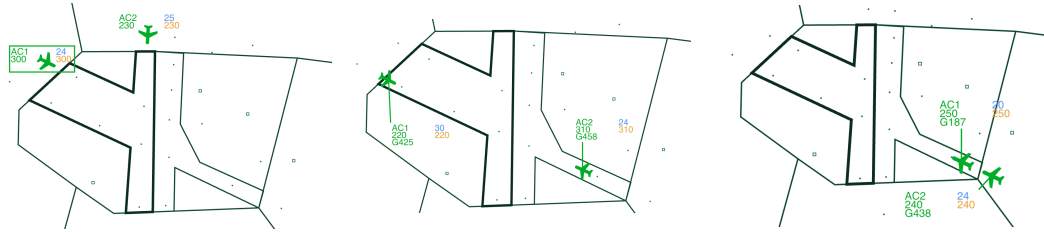
Figure 12: Gemini-2.5-Pro **failing to solve** an $N = 30$ benchmark: there are three interactions in this scenario. In this case, at time $t = 10$ and $t = 11$ aircraft 11 interacts with aircraft 21 and 22 respectively, in a head-on manner. Similarly, at $t = 11$, aircraft 12 interacts with aircraft 22. Notice that, similar to Figure 11, the model has attempted to thread sequences of aircraft along the same route one-by-one. In this case, it has miscalculated the time at which aircraft 21 would reach the end of its route. (Snapshots shown every two time units for convenience).

Table 6: Benchmarking results for varying numbers of aircraft. On a subset of the benchmark, we compare the performance of five models to their performance **after being given access to automated verification of their solutions and up to five opportunities to correct any errors**. Values quoted are the average number of unique interacting pairs of aircraft computed across 10 generated scenarios on 10 synthetic sectors. Optimal behaviour (marked with a checkmark) is zero interacting pairs. All models in all cases needed only **one attempt** to correct all their mistakes.

Number of aircraft	10		15		20		25		30	
	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
gemini-2.5-pro	0.1	✓	0.6	✓	3.2	✓	0.3	✓	0.9	✓
gpt-oss-20b	0.1	✓	0.1	✓	0.7	✓	1.0	✓	0.2	✓
o3	✓	✓	✓	✓	0.2	✓	0.2	✓	✓	✓
gpt-oss-120b	0.2	✓	✓	✓	0.1	✓	✓	✓	✓	✓

E Extra Controllability Experiments

In Figure 13 we show that the method is able to accurately generate all three types of pairwise interactions.



(a) Generate a scenario in which two aircraft interact in a crossing-paths manner.

(b) Generate a scenario in which two aircraft interact in a head-on manner.

(c) Generate a scenario in which two aircraft interact in a catch-up configuration.

Figure 13: Three examples of fine-grained scenario controllability using AirTrafficGen. Note that in (c) the two aircraft have two very different ground speeds (denoted GXXX), meaning that AC2 will catch up and overtake AC1. Note that differing ground speeds arise from the simulator when converting “slow” and “fast” movers in our discretised scheme to turboprops and jets respectively. The same aircraft type will fly at different speeds according to its flight level.