



An Efficient Dual-Agent Framework for Generating and Evaluating Synthetic Aviation Safety Reports using Large Language Models

Xiao Jing^{*}, Jirat Bhanpato[†], Mayank V. Bendarkar[‡], Dimitri N. Mavris[§]
Georgia Institute of Technology, Atlanta, Georgia, 30332

This study proposes a dual-agent framework leveraging Large Language Models (LLMs) to generate and evaluate synthetic aviation safety reports, addressing challenges like class imbalance and domain-specific requirements. By assigning separate generation and evaluation roles to specialized agents operating concurrently on distinct GPUs, the framework optimizes resource utilization and enhances content quality. Results demonstrate improved processing efficiency compared to single-GPU systems and enhanced content quality through iterative refinement. This research highlights the effectiveness of dual-agent LLM systems in producing high-quality synthetic data, reducing reliance on human evaluations and advancing NLP applications in safety-critical domains.

I. Introduction and Background

AVIATION safety reports represent one of the most valuable sources of information for identifying emerging safety threats and developing mitigation strategies. These detailed narratives, collected through systems such as the Aviation Safety Reporting System (ASRS) and the National Transportation Safety Board (NTSB), document incidents involving weather, technical malfunctions, human factors, and procedural violations [1–3]. The systematic analysis of these narratives through Natural Language Processing (NLP) techniques has shown significant potential for extracting actionable safety insights, enabling aviation organizations to proactively address risks before they escalate into serious accidents [4, 5]. Recent advances in deep learning and machine learning methodologies across diverse domains [6–8] have further enhanced these analytical capabilities, enabling more sophisticated approaches to aviation safety data processing. This analytical foundation has been strengthened through complementary approaches including specialized text classification models [9], named entity recognition for aviation safety reports [10], and structured question-answering frameworks for systematic knowledge extraction from incident narratives [11].

However, a critical challenge emerges when analyzing aviation safety data: severe and potentially catastrophic events are inherently rare, creating extreme class imbalance in safety databases [3, 9]. While routine incidents and minor procedural deviations are well-represented in reporting systems, safety-critical events that carry the highest risk potential remain severely underrepresented. This imbalance is particularly problematic because these rare events often provide the most crucial insights for preventing future accidents. Traditional NLP models, when trained on such imbalanced datasets, struggle to accurately identify and classify these critical safety events, potentially missing early warning signals that could prevent catastrophic outcomes [12, 13].

Data augmentation through synthetic report generation offers a promising approach to address this safety-critical imbalance, enabling the creation of additional training examples for underrepresented but high-risk event categories [14]. In our parallel study [15], we demonstrate how knowledge graph-enhanced methods can generate synthetic aviation safety reports that target specific incident types. However, the deployment of Large Language Models (LLMs) for synthetic data generation in safety-critical domains like aviation faces a fundamental challenge: ensuring quality and regulatory compliance requires extensive human oversight. The generated content must adhere to strict domain-specific standards, including regulatory requirements and standard operating procedures (SOPs) [16]. Furthermore, the potential for hallucinations or inaccurate technical details in generated safety narratives poses significant risks when dealing with sensitive safety information [17].

^{*}Graduate Research Assistant, Aerospace Systems Design Laboratory, and AIAA Student Member

[†]Research Engineer I, Aerospace Systems Design Laboratory, and AIAA Young Professional Member

[‡]Research Engineer II, Aerospace Systems Design Laboratory, and AIAA Member

[§]Georgia Tech Distinguished Regents Professor and Director of ASDL, AIAA Fellow

The bottleneck in scaling synthetic aviation safety data generation lies primarily in the evaluation process. Current evaluation methods either rely heavily on costly human expert annotations or employ single-agent automated approaches that lack the domain-specific expertise needed to assess aviation safety content quality [18, 19]. Manual evaluation by aviation safety experts, while thorough, is prohibitively expensive and time-consuming for the large-scale data generation needed to address class imbalance effectively. Existing automated evaluation techniques often fail to capture the nuanced requirements of aviation safety reporting, including technical accuracy, regulatory compliance, and incident classification precision [20].

To address these aviation safety challenges, we propose a novel dual-agent framework that minimizes human intervention in quality assurance while maintaining the rigorous standards required for safety-critical applications. Our approach assigns separate roles for generation and evaluation to two distinct LLM agents [21] operating concurrently, with the evaluator agent specifically designed to assess aviation safety narratives against domain-specific criteria including adherence to SOPs [16], technical accuracy, and regulatory compliance. This architecture enables scalable production of high-quality synthetic safety reports without the prohibitive costs of extensive human oversight, directly addressing the evaluation bottleneck that has limited synthetic data adoption in aviation safety applications.

Our primary research focus centers on developing a quality-assured pipeline for aviation safety data augmentation that can reliably produce synthetic reports meeting industry standards. We hypothesize that a specialized dual-agent system with aviation domain-specific evaluation criteria can achieve the quality assurance needed for safety-critical applications while dramatically reducing human oversight requirements. Additionally, we investigate whether multi-dimensional quality assessment incorporating technical accuracy, structural compliance, and safety relevance can provide reliable automated evaluation that matches human expert judgment for aviation safety content.

Our key contributions directly address aviation safety data challenges. First, we develop a dual-agent architecture that enables quality-assured generation of aviation safety narratives, addressing the evaluation bottleneck that has limited synthetic data adoption in safety-critical domains. Second, we design a comprehensive evaluation framework incorporating aviation-specific criteria and regulatory requirements, ensuring generated content meets industry standards without extensive human review. Third, we implement efficient communication protocols that enable real-time quality feedback and iterative improvement, optimizing both safety content quality and production efficiency. Finally, we provide empirical validation demonstrating that our approach can produce aviation safety reports that meet domain expert standards while significantly reducing human evaluation costs.

This research advances the practical deployment of AI agent systems [22] in aviation safety by solving the fundamental challenge of quality assurance in synthetic data generation. By reducing reliance on costly human evaluations while maintaining rigorous safety standards, our dual-agent framework enables the large-scale data augmentation needed to address critical class imbalance issues in aviation safety databases, ultimately contributing to improved safety outcomes through enhanced incident detection and risk assessment capabilities.

II. Methodology

A. Overview

Building upon established multi-agent LLM architectures [23, 24], our framework implements specialized Generator and Evaluator agents that operate concurrently to ensure both efficiency and quality in synthetic aviation safety report production. The architecture addresses the evaluation bottleneck in safety-critical data generation by automating quality assessment while maintaining aviation-specific compliance standards.

The architecture employs two specialized large language model agents operating in parallel, with a sophisticated communication mechanism that enables iterative quality improvement through structured feedback loops [25]. As illustrated in Figure 1, each agent is assigned dedicated GPU resources to optimize processing efficiency, while maintaining continuous communication through a JSON-based protocol [26]. Multi-agent LLM frameworks require robust, structured, and interpretable inter-agent communication for effective task coordination [27, 28]. JSON, as a lightweight and widely adopted data-interchange format, is especially suitable for LLM agent communication due to its efficiency, human readability, and strong support in both research and production environments [27, 28]. This design ensures both computational efficiency and effective quality control in the report generation process.

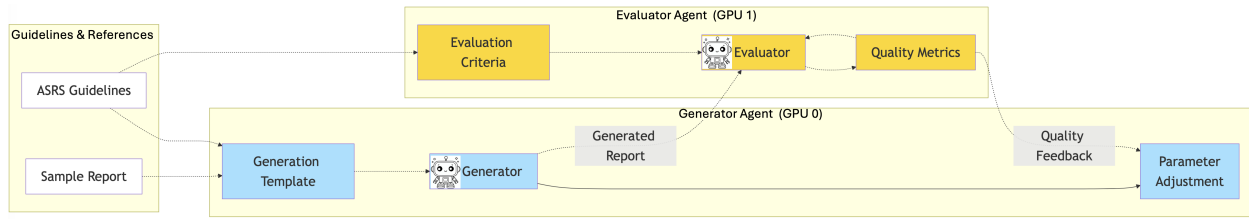


Fig. 1 Proposed Dual-Agent Framework Architecture. The framework consists of two specialized agents: a Generator Agent (bottom, GPU 0) and an Evaluator Agent (top, GPU 1). The agents operate in parallel, communicating through structured JSON messages while following ASRS guidelines. Dashed arrows indicate information flow and feedback loops between components.

B. ASRS Background and Guidelines

The Aviation Safety Reporting System (ASRS) *, established by NASA in 1976, serves as the world’s largest repository of voluntary aviation safety reports [29]. This system plays a critical role in enhancing aviation safety by collecting, processing, and analyzing over one million incident reports. These reports provide detailed narratives of various safety incidents, including technical malfunctions, human factors, and environmental influences. The ASRS’s standardized reporting framework ensures consistency and reliability in the data, which is crucial for identifying patterns and trends that may indicate emerging safety risks [2].

Our study focuses on this extensive repository, using ASRS reports and their corresponding anomaly labels as the primary dataset. Each report in the ASRS database is meticulously categorized into multiple anomaly events, enabling multilabel classification [9]. This categorization provides a robust foundation for developing and validating our dual-agent framework. By adhering strictly to ASRS guidelines, we ensure that the generated synthetic reports align with industry standards, maintaining both structural and semantic integrity necessary for real-world applications [12].

Our framework strictly adheres to ASRS guidelines and taxonomies [30], incorporating:

- Standardized Report Elements:
 - Initial Situation Description
 - Event Sequence Documentation
 - Contributing Factors Analysis
 - Human Factors Consideration [31]
 - Resolution and Recommendations
- Technical Components:
 - Aviation Terminology Standards
 - Flight Phase Classifications
 - Weather Condition Categories
 - Aircraft Systems Taxonomy
 - ATC Communication Protocols [32]
- Safety Classification Framework:
 - Primary Problem Categories
 - Contributing Factors Classification
 - Risk Assessment Metrics
 - Anomaly Event Types [33]

These elements form the backbone of our study, guiding both the Generator and Evaluator Agents. The Generator Agent uses these structured components to produce synthetic reports that emulate real ASRS narratives. Meanwhile, the Evaluator Agent assesses these reports based on their adherence to these standardized elements and technical components, ensuring high-quality outputs that align with aviation safety standards. By leveraging the ASRS’s comprehensive taxonomy and reporting framework, our dual-agent system aims to address the challenges of data scarcity, class imbalance, and domain-specific evaluation in aviation safety.

*<https://asrs.arc.nasa.gov/>

C. System Architecture

Our framework implements a sophisticated dual-agent architecture designed to maximize efficiency and effectiveness in generating and evaluating aviation safety reports. The architecture leverages two specialized agents, each tailored to its specific role within the system: the Generator Agent (GA) and the Evaluator Agent (EA):

1. Generator Agent (GA)

The Generator Agent is built upon the Mistral-7B-Instruct-v0.2 model [34], a large language model known for its robust instruction-following capabilities and high performance in structured text generation tasks. Mistral-7B has been optimized for low-latency inference, making it particularly suitable for high-throughput applications such as ours [35].

In this study, the GA is responsible for generating synthetic aviation safety reports that adhere to ASRS guidelines. To achieve this, we implement prompt engineering strategies that ensure domain compliance and technical accuracy. For example, our prompts incorporate mandatory ASRS structural elements such as "Initial Situation," "Event Sequence," and "Contributing Factors," while enforcing aviation-specific terminology standards including precise flight levels and standard ATC phraseology. The GA incorporates:

- Specialized Prompt Engineering:
 - Context-rich instruction format derived from ASRS guidelines
 - Structured output requirements aligned with ASRS taxonomies
 - Dynamic example integration based on target incident types
 - Adaptive parameter adjustment mechanism: The generation parameters (e.g., temperature, top-p) are dynamically adjusted based on feedback from the Evaluator Agent, enabling the system to iteratively refine report quality [36].

The generation quality and characteristics of large language models are primarily controlled through several key hyperparameters that influence the randomness, diversity, and coherence of generated text. In our framework, these parameters are dynamically adjusted based on evaluator feedback to optimize report quality for aviation safety content.

- Generation Control Parameters:
 - Temperature (*temp*): 0.3-0.7 (dynamically adjusted)
 - Top-p (*top_p*): 0.7-0.9 (context-dependent)
 - Repetition Penalty (*rep_pen*): 1.1-1.3 (adaptive)
 - Maximum Token Length: 512 tokens (optimized for ASRS format to accommodate typical report length while maintaining conciseness)

The adaptive generation of domain-specific narratives in the model aims to align synthetic data closely with the narratives of real-world aviation incidents, facilitating its use in downstream NLP tasks such as classification and risk assessment [3].

2. Evaluator Agent (EA)

The Evaluator Agent is built upon the Mistral-7B-Instruct-v0.2 model [34] as well and implements a multi-dimensional assessment framework inspired by Zhang et al.'s work on LLM evaluation frameworks [23]. This framework is specifically adapted to the aviation domain, where the accuracy and compliance of safety reports are paramount.

Zhang et al. [23] proposed a structured evaluation approach that combines technical accuracy, structural compliance, and contextual relevance. Building on this, our EA evaluates generated reports against three primary criteria with quantitative scoring as detailed in Section II.D:

- Technical Accuracy Assessment [37]:
 - Aviation terminology validation
 - ATC Procedure accuracy verification
 - Flight level and communication protocol checking
 - Weather condition consistency evaluation
- Structural Compliance Verification [30]:
 - ASRS format adherence
 - Required field completeness
 - Information sequence validation
 - Cross-reference consistency
- Safety Relevance Evaluation [38]:

- Incident type classification accuracy
- Contributing factor identification
- Risk assessment alignment
- Recommendation appropriateness

The EA uses a JSON-based communication protocol to provide comprehensive feedback to the GA. Quantitative scores include individual criterion scores for technical accuracy, structural compliance, and safety relevance, plus the weighted overall quality score computed according to Eq. 2. Qualitative improvement suggestions consist of structured textual feedback identifying specific deficiencies and providing actionable recommendations for content improvement. This dual-mode feedback enables both automated parameter optimization based on numerical scores and targeted content improvement through specific guidance, ensuring continuous refinement toward aviation industry standards [25].

In the present study, our primary objective is to validate the feasibility and practical effectiveness of the proposed dual-agent framework, establishing a foundation for future enhancements involving domain-adapted models and large-scale evaluation protocols.

D. Communication Protocol

To facilitate efficient collaboration between the GA (Generator Agent) and EA (Evaluator Agent), our framework employs an asynchronous JSON-based communication protocol as shown in Eq. 1:

$$M = \{evaluation, feedback, parameters\} \quad (1)$$

where:

- *evaluation*: Contains quantitative scores for technical accuracy, ASRS compliance, and safety relevance.
- *feedback*: Provides structured improvement suggestions to guide the Generator Agent.
- *parameters*: Includes dynamically adjusted generation parameters to enhance future iterations.

Drawing inspiration from distributed parameter server architectures [39, 40], the Generator continuously produces new reports, while the Evaluator operates independently and asynchronously queues evaluation feedback. The protocol enables the seamless exchange of evaluation metrics, feedback, and parameter updates, supporting real-time optimization of the generation process. The Generator updates its generation parameters only when sufficiently high-quality feedback is available, ensuring that both GPUs remain fully utilized throughout the experiment and significantly increasing overall pipeline throughput. This asynchronous, loosely coupled design minimizes synchronization bottlenecks that often hinder large-scale multi-agent systems, and maximizes resource efficiency by allowing generation and evaluation to proceed in parallel without unnecessary idle time. The approach implements two key principles from multi-agent learning: (1) quality-selective feedback aggregation, where the Generator only updates parameters when receiving feedback above a quality threshold, and (2) asynchronous coordination, where agents operate independently without waiting for each other to complete tasks [24, 25].

The protocol structure is inspired by recent work in multi-agent collaboration networks [25], and the feedback loop structures from asynchronous distributed learning frameworks [24, 39]. Specifically, our parameter update protocol follows a quality-thresholded, best-feedback policy, where the Generator Agent only adjusts its parameters upon receipt of feedback with a quality score surpassing a dynamic baseline. This design follows reinforcement learning principles [41], where the Generator Agent receives quality scores as numerical rewards for its generated reports and selectively adjusts its generation parameters to maximize future rewards, similar to how distributed machine learning systems update model parameters based on performance feedback [40, 42]:

- **Quality Assessment Metrics:**
Our quality assessment framework implements a conservative and rigorous scoring mechanism, designed to enable meaningful iterative improvements and effective inter-agent communication:

$$Q_{score} = \alpha T_{acc} + \beta S_{comp} + \gamma L_{rel} \quad (2)$$

where:

- T_{acc} : Technical accuracy score (0.3-0.7)

Building upon general principles outlined in aviation safety assessment practices [37], we establish a bounded scoring range for technical accuracy tailored to our specific application in aviation safety reporting.

The lower bound of 0.3 reflects baseline requirements, ensuring that all generated reports adhere to ASRS structural guidelines and leverage Mistral-7B’s fundamental language capabilities [34]. The upper bound of 0.7 accounts for the inherent limitations of non-fine-tuned models in highly specialized domains, consistent with best practices for AI model evaluation in safety-critical applications [43]. This scoring framework emphasizes the importance of domain-specific adaptation for new applications in aviation.

- S_{comp} : ASRS compliance score (0.3-0.7)

This bounded range for structural compliance aligns with established practices in aviation safety documentation evaluation [29]. The conservative upper limit allows room for future improvements through domain-specific fine-tuning, while the non-zero lower bound recognizes the base model’s inherent ability to maintain basic report structure.

- L_{rel} : Label relevance score (0.3-0.7)

The bounded scoring range for label relevance follows research by Zhang et al. [23] on LLM evaluation frameworks, where appropriate score boundaries help reduce noise and enable more meaningful quality differentiation. This range proves particularly effective for tracking incremental improvements in classification accuracy.

- Coefficients: $\alpha = 0.4, \beta = 0.3, \gamma = 0.3$

These empirically determined weights reflect the relative importance of each component, with slightly higher emphasis on technical accuracy given its critical role in aviation safety reporting.

This conservative scoring approach serves multiple purposes. First, by maintaining strict upper bounds, we create headroom for quality improvements through agent interaction and parameter optimization, which aligns with findings from Arif et al. [24] on effective multi-agent learning dynamics. Second, the bounded range helps clearly demonstrate quality improvements during the iterative refinement process, avoiding the saturation effects often seen with wider scoring ranges [23]. Third, the scoring boundaries acknowledge that even high-quality machine-generated aviation safety reports require human expert validation, reserving scores above 0.7 for future fine-tuned models incorporating domain expertise [20]. Finally, the granular scoring within this range provides clear feedback signals for parameter adjustment, enabling effective inter-agent learning as demonstrated in recent multi-agent LLM research [25].

- **Parameter Update Protocol:**

Algorithm 1 summarizes our asynchronous pipeline for generator-evaluator collaboration. Unlike traditional step-wise synchronous updates, the Generator produces reports without blocking, while the Evaluator processes each report as soon as it becomes available. Parameter updates are triggered only when feedback of sufficiently high quality is available in the feedback queue, thereby maximizing GPU utilization and ensuring robust, sample-efficient adaptation to generate aviation reports.

Algorithm 1 Asynchronous Dynamic Parameter Adjustment

```

1: Initialize  $\theta_0 = \{temp_0, top\_p_0, rep\_pen_0\}$  ▷ See Section II.C.1 for parameter definitions
2: while experiment is running do
3:    $R_j \leftarrow \text{GeneratorAgent}(\theta_j, \text{ASRS\_guidelines})$ 
4:   Place  $R_j$  into evaluation queue
5:    $R_k \leftarrow$  Retrieve from evaluation queue (asynchronously)
6:    $E_k, Q_k, F_k \leftarrow \text{EvaluatorAgent}(R_k)$ 
7:   if  $Q_k >$  dynamic baseline then
8:     Place  $F_k$  into feedback queue
9:   end if
10:  if new  $F^*$  in feedback queue and  $Q^* >$  threshold then
11:     $\theta_{j+1} \leftarrow \text{Update}(\theta_j, F^*)$ 
12:  end if
13: end while

```

E. Illustrative Pipeline Execution

Table 1 demonstrates the asynchronous nature of our pipeline. At each time step t_k , the Generator Agent creates report R_k independently of evaluation status, while the Evaluator Agent processes the most recently available report R_{k-1} in parallel. Parameter updates to the Generator are only performed when high-quality feedback F^* (as defined in Algorithm 1) is received and meets the dynamic quality threshold.

Table 1 Illustration of Asynchronous Pipeline Status

Time Step	Generator Action	Evaluator Action
t_1	Generate R_1	Idle
t_2	Generate R_2	Evaluate R_1
t_3	Generate R_3	Evaluate R_2
t_4	– (wait feedback)	Evaluate R_3
t_5	Update θ (if F^*)	–
...
t_N	Generate R_N	Evaluate R_{N-1}

F. Hardware Optimization

Our framework implements a dual-GPU, asynchronous parallel processing architecture, building on the efficiency gains of prior LLM pipeline studies [35, 36] and asynchronous distributed learning frameworks [39, 40]. The implementation utilizes two NVIDIA GeForce RTX 3090 GPUs, each with 24GB GDDR6X memory, as detailed in Table 2.

Table 2 Hardware Specifications

Component	Specification
GPU Model	2× NVIDIA GeForce RTX 3090
GPU Memory	24576 MiB GDDR6X per GPU
CUDA Version	12.6
Driver Version	560.35.03
Power Capacity	350W per GPU

1. Resource Allocation Strategy

Our dual-GPU architecture implements dedicated resource allocation to maximize computational efficiency and minimize resource contention. GPU 0 is exclusively dedicated to the Generator Agent, while GPU 1 handles all Evaluator Agent operations. This separation ensures independent 24GB memory spaces for each agent, eliminating memory contention that could degrade performance in shared GPU scenarios. The configuration also enables separate model caching with CUDA 12.6, allowing each agent to maintain optimized memory usage patterns specific to their computational requirements.

2. Task Allocation Efficiency Analysis

The Task Allocation Efficiency metric is critical for understanding the practical benefits of our dual-agent architecture in aviation safety applications. Unlike traditional sequential approaches where the evaluator must wait for generation to complete before assessing quality, our asynchronous design enables continuous operation. This efficiency gain is particularly important in aviation safety contexts where large-scale synthetic data generation is needed to address class imbalance issues—the faster we can generate quality-assured reports, the more effectively we can augment underrepresented safety categories.

Our efficiency is quantified by Equation 3:

$$E_{\text{async}} = \frac{T_{\text{sequential}}}{T_{\text{async}}} = \frac{T_{\text{gen}} + T_{\text{eval}} + T_{\text{comm}}}{\max(T_{\text{gen}}, T_{\text{eval}}) + T_{\text{comm}}} \quad (3)$$

The parameters in this equation reflect the core time components of our pipeline: T_{gen} represents the average generation time per report, T_{eval} captures the evaluation time, and T_{comm} accounts for communication overhead. This metric adapts pipeline efficiency formulas from distributed computing research [36, 39, 40], with explicit incorporation of communication overhead characteristic of asynchronous multi-agent pipelines. The communication time T_{comm} varies significantly between implementations: for sequential (single-GPU) processing, it refers to in-memory data transfer and function call overhead, while for parallel (dual-GPU) processing, it includes inter-GPU data transfer via PCIe, queue operations, and synchronization delays.

This efficiency metric directly impacts the scalability of synthetic aviation safety data generation. Higher efficiency values indicate better resource utilization and faster production of quality-assured reports, which is essential for addressing the class imbalance challenges in aviation safety databases. The asynchronous design typically achieves efficiency gains of 1.5-2.0x compared to sequential processing, enabling practical large-scale synthetic data generation that would otherwise be computationally prohibitive [44].

3. Memory Efficiency Optimization

Table 3 Memory Allocation and Efficiency Metrics

Memory Component	Specification
Model Loading per GPU	~14GB (Mistral-7B)
GPU Memory Utilization	<60% per agent
Memory Contention	Eliminated (dedicated VRAM)
Cache Management	CUDA-aware optimization

Our memory optimization strategy, summarized in Table 3, focuses on efficient resource utilization while maintaining system stability. Independent model loading ensures that each GPU maintains its own copy of the Mistral-7B model (~14GB), eliminating memory contention that could cause performance degradation. The dedicated VRAM allocation prevents the memory thrashing common in shared GPU environments, while CUDA-aware memory management optimizes cache usage patterns. This configuration maintains GPU memory utilization below 60% per agent, providing sufficient headroom for batch processing and avoiding out-of-memory errors during extended operation periods.

III. Results and Discussions

A. Experimental Setup

The evaluation in this study focuses on validating framework efficiency and verifying the core asynchronous mechanism using the base Mistral-7B model. To assess robustness across multiple safety categories, we selected three representative sets of verified ASRS anomaly labels: ("Human Factors", "Communication Breakdown"), ("Weather", "Turbulence Encounter"), and ("ATC Issue All Types", "Conflict Airborne Conflict"). For each label group, a curated ASRS report serves as the generation reference, guiding the model to produce corresponding synthetic safety narratives.

Each experiment consisted of 50 report generation and evaluation iterations, totaling 150 cycles across all label conditions. All experiments were performed on a workstation equipped with dual NVIDIA RTX 3090 GPUs (24GB each).

While our asynchronous dual-agent pipeline maximizes GPU utilization, running large LLMs like Mistral-7B in parallel imposes significant computational demands. Specifically, both GPU VRAM and system RAM face considerable pressure as intermediate reports, evaluations, and queue buffers must be maintained in memory throughout the process. Additionally, sustained concurrent workloads occasionally led to elevated system temperatures and resource contention, which constrained both the batch size and total iterations per session. These factors represent practical bottlenecks for multi-agent LLM inference on standard workstation hardware [45].

Given these hardware constraints, we restricted each experimental condition to 50 iterations to ensure reliable execution and avoid memory exhaustion. This limitation, while necessary for the current study, motivates our planned future deployment on high-performance computing infrastructure. Specifically, we intend to leverage Georgia Tech’s PACE cluster [46] to enable larger-scale experiments and enhanced parallelization. Additional optimizations, including on-disk buffering and distributed queue management will be essential for scaling beyond current limitations.

With these practical considerations in mind, our experiments were designed to validate three core capabilities of the dual-agent framework: (1) improved computational efficiency through dual-GPU parallelization compared to single-GPU setups, (2) the effectiveness of adaptive parameter adjustment for enhancing synthetic report quality, and (3) the evaluator agent’s capability to accurately discriminate between high- and low-quality reports. The following sections detail our experimental methodology and results.

B. Performance Analysis

To quantitatively assess the efficiency gains enabled by our dual-GPU asynchronous architecture, we conducted a controlled comparison with a single-GPU baseline. Both frameworks maintain identical functionality and communication patterns, but the single-GPU implementation executes Generator and Evaluator agents sequentially on a single device, while the dual-GPU setup executes each agent on a dedicated GPU with shared system RAM buffers.

Table 4 Performance Comparison: Single-GPU vs. Dual-GPU Implementation (150 Iterations total)

Metric	Single GPU	Dual GPU
Total Processing Time (s)	4531.50	2807.32
Avg Generation Time (s)	21.12	18.53
Avg Evaluation Time (s)	8.92	7.04
Avg Communication Time (s)	0.17	0.12
Average Per-iteration Time (s)	33.21	18.65
Throughput (reports/hour)	119	193
Efficiency (E_{async})	–	1.62

Table 4 summarizes the measured results. On the single-GPU setup, the average generation and evaluation times were 21.12s and 8.92s per report, respectively, with each communication step requiring 0.17s. The dual-GPU configuration achieved a generation time of 18.53s, evaluation time of 7.04s, and reduced communication latency of 0.12s per iteration, resulting in a total processing time of 2807.32s for 150 reports—substantially faster than the single-GPU’s 4531.50s.

Communication Latency Note: The observation that average communication time is lower in the dual-GPU setup (0.12s) compared to single-GPU (0.17s) is consistent with repeated multi-round testing. In our implementation, the dual-GPU setup benefits from reduced context switching overhead, as each GPU maintains its dedicated model state without interruption. In contrast, the single-GPU setup incurs additional overhead from task scheduling, context switching between generator and evaluator roles, and potential memory management operations. This effect is particularly pronounced in large-model inference workloads, where maintaining model states and managing memory allocation are critical factors [36, 47].

Task Allocation Efficiency: By applying our Task Allocation Efficiency in Asynchronous Parallel Processing formula (see Section II.F.2 Eq. 3), the dual-GPU architecture achieves an efficiency value of 1.62, representing a 62% increase in per-iteration throughput over the sequential baseline. This value falls within the theoretically expected range for loosely coupled agent-parallelism with minimal communication overhead [35, 36].

Overall, these results are consistent with established findings in distributed deep learning [35, 36, 47, 48], where optimized parallel model deployment and reduced resource contention significantly improve inference efficiency for large models. Our dual-GPU architecture achieves efficient scaling by eliminating context switching overhead and maintaining dedicated model states on each device. The measured speedup of 62% demonstrates that commodity workstation hardware can achieve substantial performance gains for multi-agent LLM workloads when computational resources are properly isolated and memory contention is minimized.

C. Quality Progression Analysis

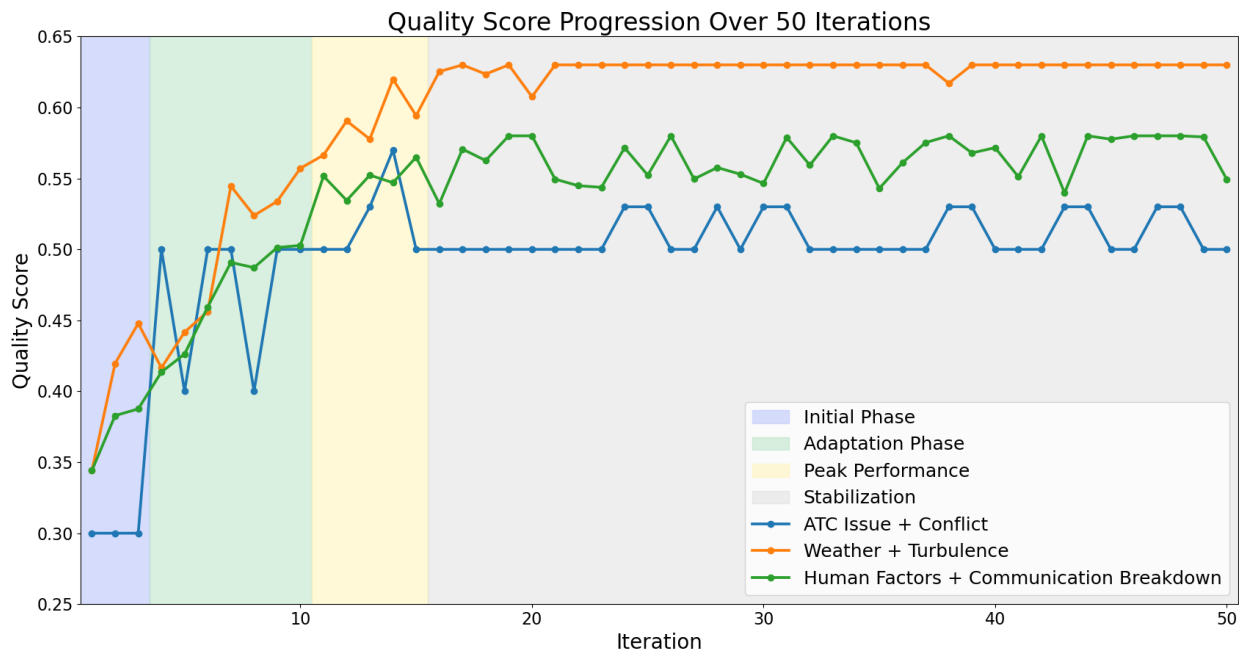


Fig. 2 Quality Score Progression Over 50 Iterations Across Three Representative Task Pairs. Distinct phases are visually segmented for interpretability.

Figure 2 illustrates the evolution of quality scores across 50 iterations for three representative task domains: *ATC Issue + Conflict*, *Weather + Turbulence*, and *Human Factors + Communication Breakdown*. The learning progression for each agent-evaluator pairing consistently exhibits four distinct phases, which are color-coded in the plot for clarity:

1) Initial Phase (Iterations 1–3):

- All models begin at a consistent baseline score (0.30–0.36), reflecting general large language model (LLM) capabilities prior to adaptation.
- Output narratives show only superficial compliance with ASRS report structure, lacking substantive domain detail.

2) Adaptation Phase (Iterations 4–10):

- Rapid improvement in quality scores (0.40–0.55), often accompanied by moderate fluctuations.
- Agents leverage evaluator feedback to adjust parameters, correct formatting, and improve technical accuracy.

3) Peak Performance Phase (Iterations 11–15):

- Quality scores reach local maxima, with group-specific ceilings (e.g., 0.63 for *Weather + Turbulence*, 0.57 for *Human Factors + Communication Breakdown*).
- Narratives during this window demonstrate the highest level of domain precision and structural fidelity, consistent with effective agent adaptation.

4) Stabilization Phase (Iterations 16–50):

- Scores plateau with low-amplitude oscillations, indicating convergence to stable agent policies.
- Distinct performance gaps between task pairs become apparent, reflecting varying difficulty and model generalization limits.

This four-stage progression highlights both the adaptability and the limits of our agent-based generation-evaluation loop: early learning is rapid, but domain/task complexity ultimately constrains attainable quality. The clear separation among the three task pairs further demonstrates the system’s sensitivity to domain-specific challenges.

Scoring Framework Rationale Our scoring framework spans a calibrated range of 0.30–0.70 for the overall quality score Q_{score} (as defined in Section II.D and displayed in Figure 2), motivated by domain requirements

and model characteristics. This range represents the strict mathematical bounds of our evaluation system, where individual component scores (T_{acc} , S_{comp} , L_{rel}) are each bounded within 0.3-0.7, and the weighted combination $Q_{score} = \alpha T_{acc} + \beta S_{comp} + \gamma L_{rel}$ is mathematically constrained to fall within this range given our coefficient values ($\alpha = 0.4, \beta = 0.3, \gamma = 0.3$). Our conservative evaluation approach and the inherent limitations of non-fine-tuned models in specialized domains result in observed scores consistently utilizing this full range. Future work will employ evaluator models fine-tuned on extensive high-quality aviation safety narratives, which may enable access to the full scoring range and more nuanced quality differentiation. The framework in this study is designed to adhere to the following:

- **Lower Bound (0.30):** Reflects the baseline structural compliance achievable by pre-trained LLMs without domain adaptation. This threshold is consistent with prior studies on automatic report generation in aviation safety [34, 37].
- **Upper Bound (0.70):** Reserved for expert-level accuracy and completeness, generally only achieved after substantial fine-tuning or human post-editing. It ensures a conservative ceiling, allowing for future advances.
- **Score Distribution Guidelines (adapted from Zhang et al. [23] for aviation domain):**
 - 0.30–0.40: Basic structural and minimal domain compliance.
 - 0.41–0.50: Improved accuracy, correct format, some technical detail.
 - 0.51–0.60: Strong domain understanding and contextual accuracy.
 - 0.61–0.70: Rare, expert-level completeness and technical precision.

These boundaries provide meaningful differentiation across experiments and are robust to both evaluator and task variability.

D. Qualitative Analysis of Generated Reports

To validate the effectiveness of our evaluator agent, we conduct a comparative qualitative analysis for the *ATC Issue All Types + Conflict Airborne Conflict* task. We select representative examples at the lower and upper ends of the score distribution produced by our evaluation agent, and perform a manual assessment to analyze whether the assigned scores faithfully reflect report quality.

Each report's overall score (Eq. 2, see Section II.D) is computed as a weighted sum of technical accuracy (T_{acc}), ASRS compliance (S_{comp}), and label relevance (L_{rel}) by the evaluator agent, using empirically determined coefficients ($\alpha = 0.4, \beta = 0.3, \gamma = 0.3$).

Low-Scoring Example (~ 0.30): "On [Date] at approximately [Time], I was acting as the Radar Controller for the Myrtle Beach (MYS) High Altitude sector covering altitudes above Flight Level Three Hundred Fifty (FL350). The weather conditions during this period were characterized by strong headwinds. I was monitoring several aircraft in the quartering teal sector when coordination became challenging due to workload factors. The pilot communications were unclear and the situation developed rapidly. . . "

Analysis: This narrative received a low score due to:

- **Technical accuracy (T_{acc}):** Inconsistent use of aviation terminology and awkward phrasing (e.g., "quartering teal sector").
- **ASRS compliance (S_{comp}):** Overly verbose initial description, lack of concise causal structure, and occasional digressions from standard ASRS narrative format.
- **Label relevance (L_{rel}):** The narrative provides only an ambiguous connection to the "Conflict Airborne" criteria, lacking a clearly articulated event sequence or explicit linkage between ATC actions and conflict resolution.

Due to these collective issues, the report yielded a score near the lower bound of our calibrated range.

High-Scoring Example (~ 0.57): "The incident occurred during VFR flight conditions with clear skies and no reported weather phenomena affecting visibility... At approximately [Exact Time], while serving as the radar controller for the MySACC Hi sector, I detected MLT X at FL390 near converging traffic and handled their descent request as per standard operating procedures. I issued "MLT X, descend and maintain FL370, traffic twelve o'clock." The pilot acknowledged and I coordinated with adjacent sectors to maintain separation... MLT X was operating under Visual Flight Rules (VFR) at FL390 when they requested a descent to FL. . . "

Analysis: This narrative achieved a higher score because:

- **Technical accuracy (T_{acc}):** Consistent use of standard ATC phraseology (e.g., "descend and maintain FL370, traffic twelve o'clock"), precise terminology, and clear causal statements. Note that while "MySACC Hi sector" is a generated sector name, the evaluator in this study focuses on proper phraseology and procedural accuracy

rather than facility name validation. Such hallucination issues with facility names and specific identifiers is addressed in our parallel work through knowledge graph-enhanced methods that incorporate verified aviation domain knowledge [15].

- **ASRS compliance** (S_{comp}): Concise structure aligned with ASRS reporting guidelines, logical sequence, and minimal extraneous detail.
- **Label relevance** (L_{rel}): Clear correspondence between actions described and both target labels: “ATC Issue” criteria are satisfied through controller workload, procedural decisions, and inter-sector coordination, while “Conflict Airborne” criteria are met through explicit mention of converging traffic, separation concerns, and conflict resolution actions.

The combination of technical rigor, format adherence, and label clarity places this example at the upper end of the observed score distribution.

These results demonstrate that our Evaluation Agent, leveraging the composite metric in Section II.D, assigns scores that closely align with human judgment of report quality. Manual inspection confirms that high-scoring narratives exhibit greater accuracy, structure, and label relevance than low-scoring examples, supporting the validity and practical interpretability of the automatic evaluation framework.

IV. Conclusions and Future Work

This study presents a dual-agent framework that leverages large language models to minimize human intervention in generating and evaluating synthetic aviation safety reports. Through the deployment of specialized Generator and Evaluator agents operating concurrently on dedicated GPUs, we have demonstrated an efficient approach that enables scaling up synthetic report production while maintaining quality control. By automating the evaluation process traditionally requiring costly human review, our framework addresses both the computational and human resource bottlenecks that have limited large-scale synthetic data generation in safety-critical domains. This approach directly tackles the class imbalance issue prevalent in aviation safety databases by making it feasible to generate substantial volumes of high-quality synthetic reports for underrepresented incident categories.

The empirical results support our architectural design decisions. By decoupling generation and evaluation into parallel processes, we observed a 62% improvement in throughput compared to sequential implementations. This efficiency gain provides a practical pathway toward generating sufficient volumes of synthetic data to address class imbalances in aviation safety datasets. The asynchronous communication protocol, utilizing JSON-based message passing and quality-thresholded parameter updates, enabled effective inter-agent collaboration while maintaining computational efficiency.

Our experiments reveal that LLMs can produce domain-compliant content when guided by systematic evaluation and iterative refinement. The observed four-phase learning progression across experimental conditions demonstrates the adaptability of language models when provided with structured, domain-specific feedback. This finding suggests potential applications for LLMs in other specialized domains where data scarcity and technical precision present similar challenges.

The multi-dimensional evaluation framework developed in this study decomposes aviation safety report quality into technical accuracy, structural compliance, and label relevance components. While our current scoring boundaries of 0.3 to 0.7 provide a functional baseline for assessment, we acknowledge these metrics represent an initial approximation. Future work will develop more granular evaluation criteria that better capture the nuances of aviation safety documentation, potentially incorporating subcategories for specific technical terminology, incident severity classifications, and regulatory compliance levels.

Our investigation also revealed practical constraints when deploying multi-agent LLM systems on commodity hardware. Memory bottlenecks and thermal limitations restricted our experimental scope to 50 iterations per condition, highlighting the gap between theoretical designs and implementation realities. These limitations provide valuable insights for researchers working on scalable LLM deployment and motivate our planned migration to high-performance computing infrastructure.

Looking forward, the integration with our parallel research on knowledge graph-enhanced generation presents promising opportunities for improvement. By combining the efficiency of our dual-agent architecture with structured aviation knowledge, we anticipate several enhancements. These include improved technical accuracy through aviation-specific knowledge integration, more precise evaluation capabilities guided by domain expertise, and refined parameter adjustment mechanisms that incorporate both statistical feedback and domain-specific constraints. The knowledge graph

approach could particularly strengthen the evaluator’s ability to verify technical details against established aviation standards and procedures.

Future development will also focus on expanding the evaluation framework beyond our current conservative metrics. We envision developing more sophisticated scoring mechanisms that capture finer-grained quality distinctions, potentially incorporating hierarchical evaluation criteria for different aspects of report quality. Additionally, creating specialized evaluation modules for different incident types could enable more targeted quality improvements in underrepresented safety categories. Advanced techniques such as reinforcement learning from automated feedback could further refine the evaluator’s assessment capabilities, allowing the system to learn optimal scoring patterns from large-scale generation-evaluation cycles without human intervention.

The principles established in this work, including asynchronous multi-agent collaboration, structured evaluation feedback, and adaptive parameter optimization, may find applications beyond aviation safety. Similar frameworks could potentially address data augmentation challenges in other safety-critical domains such as healthcare incident reporting or industrial accident documentation, where comparable requirements for accuracy and regulatory compliance exist.

In conclusion, this study advances synthetic aviation safety report generation through a practical dual-agent framework while establishing a foundation for deploying collaborative LLM agents in specialized domains. As the field continues to evolve, we hope our approach contributes to the broader effort of making LLMs more effective tools for addressing data challenges in safety-critical applications. The combination of our dual-agent architecture with knowledge-enhanced generation represents a step toward more sophisticated systems capable of producing high-quality domain-specific content with reduced human oversight.

References

- [1] “Risk Factors Associated with Weather-Related General Aviation Accidents,” Tech. Rep. NTSB/SS-05/01, National Transportation Safety Board, 2005. URL <https://www.ntsb.gov/safety/safety-studies/Pages/SS0501.aspx>.
- [2] DiFiore, A., and Cardosi, K., “Human Factors in Airport Surface Incidents: An Analysis of Pilot Reports Submitted to the Aviation Safety Reporting System (ASRS),” Tech. Rep. DOT/FAA/AR-06/5, Federal Aviation Administration, Office of Runway Safety & Operational Services, Washington, D.C., December 2006. URL https://rosap.ntl.bts.gov/view/dot/5878/dot_5878_DS1.pdf.
- [3] Jing, X., Sawant, K., Bendarkar, M. V., Elias, L. R., and Mavris, D. N., “Expanding Aviation Knowledge Graph Using Deep Learning for Safety Analysis,” *AIAA Aviation Forum and ASCEND 2024*, Las Vegas, NV, USA, 2024. doi:10.2514/6.2024-4603.
- [4] Chandra, C., Jing, X., Bendarkar, M., Sawant, K., Elias, L., Kirby, M., and Mavris, D., “Aviation-BERT: A Preliminary Aviation-Specific Natural Language Model,” *AIAA AVIATION 2023 Forum*, 2023. doi:10.2514/6.2023-3436.
- [5] Miyamoto, A., Bendarkar, M. V., and Mavris, D. N., “Natural Language Processing of Aviation Safety Reports to Identify Inefficient Operational Patterns,” *Aerospace*, Vol. 9, No. 8, 2022, p. 450. doi:10.3390/aerospace9080450.
- [6] Shi, L.-F., Liu, Z.-Y., Zhou, K.-J., Shi, Y., and Jing, X., “Novel Deep Learning Network for Gait Recognition Using Multimodal Inertial Sensors,” *Sensors*, Vol. 23, No. 2, 2023, p. 849.
- [7] Shi, Y., Jing, X., and Liu, L., “Micro-Bristle Robot Design via Different Surrogate Model Optimization Methods,” *2023 2nd International Conference on Innovations and Development of Information Technologies and Robotics (IDITR)*, Chengdu, China, 2023, pp. 12–17.
- [8] Jing, X., and Guo, M., “Data-Driven AI for Topology Optimization: A Comparative Study of CNN and Self-Attention Enhanced Architectures in Data-Constrained Scenarios,” *2024 3rd International Conference on Artificial Intelligence, Human-Computer Interaction and Robotics (AIHCIR)*, IEEE, Hong Kong, China, 2024. doi:10.1109/AIHCIR65563.2024.00048.
- [9] Jing, X., Chennakesavan, A., Chandra, C., Bendarkar, M. V., Kirby, M., and Mavris, D. N., “BERT for Aviation Text Classification,” *AIAA AVIATION 2023 Forum*, 2023, p. 3438. doi:10.2514/6.2023-3438.
- [10] Chandra, C., Ojima, Y., Bendarkar, M. V., and Mavris, D. N., “Aviation-BERT-NER: Named Entity Recognition for Aviation Safety Reports,” *Aerospace*, Vol. 11, No. 11, 2024, p. 890. doi:10.3390/aerospace11110890.
- [11] Oderinde, T., Chandra, C., Albertoli, L., Bhanpato, J., Bendarkar, M. V., and Mavris, D. N., “Aviation Safety Question Answering Dataset for Extracting Knowledge from Incident Reports,” *AIAA AVIATION Forum and ASCEND 2025*, 2025.
- [12] Yang, H., Gao, J., Yuan, Y., and Li, X., “Imbalanced Aircraft Data Anomaly Detection,” *arXiv preprint arXiv:2305.10082*, 2023. URL <https://arxiv.org/abs/2305.10082>.

- [13] Li, X., Gan, Z., Xu, Y., Liu, B., and Zheng, T., "Extracting Domain-Specific Chinese Named Entities for Aviation Safety Reports: A Case Study," *Applied Sciences*, Vol. 13, No. 19, 2023, p. 11003.
- [14] Zhao, H., Chen, H., Ruggles, T. A., Feng, Y., Singh, D., and Yoon, H.-J., "Improving Text Classification with Large Language Model-Based Data Augmentation," *Electronics*, Vol. 13, No. 13, 2024. doi:10.3390/electronics13132535, URL <https://www.mdpi.com/2079-9292/13/13/2535>.
- [15] Jing, X., Bhanpato, J., Bendarkar, M. V., and Mavris, D. N., "KG-Enhanced Synthetic Report Generation for Addressing Class Imbalance in Aviation Safety Data," *AIAA AVIATION Forum and ASCEND 2025*, 2025.
- [16] Federal Aviation Administration, "AC 120-71B: Standard Operating Procedures and Pilot Monitoring Duties for Flight Deck Crewmembers," , 2017. URL https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_120-71B.pdf, accessed: 2024-11-18.
- [17] Air Line Pilots Association, International, "Aviation Safety Action Program Reporting of Security Information," , 2017. URL <https://www.alpa.org/~media/ALPA/Files/pdfs/news-events/white-papers/white-paper-asap-reporting.pdf?1a=en>, accessed: 2024-11-18.
- [18] Szymanski, A., Gebreegziabher, S. A., Anuyah, O., Metoyer, R. A., and Li, T. J.-J., "Comparing Criteria Development Across Domain Experts, Lay Users, and Models in Large Language Model Evaluation," *arXiv preprint arXiv:2410.02054*, 2024. URL <https://arxiv.org/abs/2410.02054>.
- [19] Zhang, K., Yuan, S., and Zhao, H., "TALEC: Teach Your LLM to Evaluate in Specific Domain with In-house Criteria by Criteria Division and Zero-shot Plus Few-shot," *arXiv preprint arXiv:2407.10999*, 2024.
- [20] Chern, A., et al., "Can Large Language Models be Trusted for Evaluation? Scalable Meta-Evaluation of LLMs as Evaluators via Agent Debate," *arXiv preprint arXiv:2401.16788*, 2024.
- [21] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z.-Y., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., and Wen, J.-R., "A Survey on Large Language Model based Autonomous Agents," *Frontiers of Computer Science*, Vol. 0, No. 0, 2025, pp. 1–42. doi:10.1007/s11704-024-40231-1, arXiv:2308.11432.
- [22] Bansod, P. B., "Distinguishing Autonomous AI Agents from Collaborative Agentic Systems: A Comprehensive Framework for Understanding Modern Intelligent Architectures," *arXiv preprint arXiv:2506.01438*, 2025. URL <https://doi.org/10.48550/arXiv.2506.01438>.
- [23] Zhang, X., Yu, B., Yu, H., Lv, Y., Liu, T., Huang, F., Xu, H., and Li, Y., "Wider and Deeper LLM Networks are Fairer LLM Evaluators," *arXiv preprint arXiv:2308.01862*, 2023. URL <https://arxiv.org/abs/2308.01862>.
- [24] Arif, S., Farid, S., Azeemi, A. H., Athar, A., and Raza, A. A., "The Fellowship of the LLMs: Multi-Agent Workflows for Synthetic Preference Optimization Dataset Generation," *arXiv preprint arXiv:2408.08688*, 2024. URL <https://arxiv.org/abs/2408.08688>.
- [25] Qian, L., and et al., "Scaling Large-Language-Model-based Multi-Agent Collaboration," *arXiv preprint arXiv:2406.07155*, 2024.
- [26] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format," RFC 8259, IETF, December 2017. doi:10.17487/RFC8259, URL <https://www.rfc-editor.org/info/rfc8259>, accessed: 2025-06-07.
- [27] Xu, B., Peng, Z., Lei, B., Mukherjee, S., and Xu, D., "Decoupling Reasoning from Observations for Efficient Augmented Language Models," *International Conference on Learning Representations (ICLR)*, 2024. URL <https://openreview.net/forum?id=Cpgo06j6W1>.
- [28] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., and Zhang, X., "Large Language Model based Multi-Agents: A Survey of Progress and Challenges," *arXiv preprint arXiv:2402.01680*, 2024. doi:10.48550/arXiv.2402.01680, URL <https://arxiv.org/abs/2402.01680>, v2 [cs.CL] 19 Apr 2024.
- [29] NASA, "Aviation Safety Reporting System Program Briefing," <https://asrs.arc.nasa.gov/>, 2024.
- [30] "ASRS Coding Taxonomy," 2023. URL <https://asrs.arc.nasa.gov/search/dbol/databasecoding.html>, accessed: 2024-11-14.
- [31] Madeira, T., Melício, R., Valério, D., and Santos, L., "Machine Learning and Natural Language Processing for Prediction of Human Factors in Aviation Incident Reports," *Aerospace*, Vol. 8, No. 2, 2021, p. 47. doi:10.3390/aerospace8020047, URL <https://www.mdpi.com/2226-4310/8/2/47>.

- [32] Prinzo, O. V., “An Analysis of Approach Control/Pilot Voice Communications,” Tech. Rep. DOT/FAA/AM-96/26, Federal Aviation Administration, Office of Aviation Medicine, Washington, D.C., October 1996. URL https://www.faa.gov/sites/faa.gov/files/data_research/research/med_humanfacs/oamtechreports/AM96-26.pdf.
- [33] “ASRS Coding Taxonomy,” 2024. URL https://asrs.arc.nasa.gov/docs/dbol/ASRS_CodingTaxonomy.pdf, accessed: 2024-11-14.
- [34] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E., “Mistral 7B,” *arXiv preprint arXiv:2310.06825*, 2023. URL <https://arxiv.org/abs/2310.06825>.
- [35] He, J., and Zhai, J., “FastDecode: High-Throughput GPU-Efficient LLM Serving using Heterogeneous Pipelines,” *arXiv preprint arXiv:2403.11421*, 2024. URL <https://arxiv.org/abs/2403.11421>.
- [36] Patel, P., Choukse, E., Zhang, C., Shah, A., Goiri, I., Maleki, S., and Bianchini, R., “Splitwise: Efficient Generative LLM Inference Using Phase Splitting,” *Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA)*, ACM/IEEE, 2024, pp. 118–132. doi:10.1109/ISCA59077.2024.00019.
- [37] “Safety Management System,” Tech. Rep. FAA Order 8000.369C, Federal Aviation Administration, 2020. URL https://www.faa.gov/regulations_policies/orders_notices/index.cfm/go/document.current/documentNumber/8000.369.
- [38] Yang, C., and Huang, C., “Natural Language Processing (NLP) in Aviation Safety: Systematic Review of Research and Outlook into the Future,” *Aerospace*, Vol. 10, No. 7, 2023, p. 600. doi:10.3390/aerospace10070600.
- [39] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., and Ng, A. Y., “Large Scale Distributed Deep Networks,” *Advances in Neural Information Processing Systems 25 (NeurIPS 2012)*, 2012. URL <https://proceedings.neurips.cc/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html>, part of *Advances in Neural Information Processing Systems 25 (NeurIPS 2012)*.
- [40] Li, M., Andersen, D. G., Park, J. W., Smola, A. J., Ahmed, A., Josifovski, V., Long, J., Shekita, E. J., and Su, B.-Y., “Scaling Distributed Machine Learning with the Parameter Server,” *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2014)*, USENIX Association, Broomfield, CO, USA, 2014, pp. 583–598. URL https://www.usenix.org/conference/osdi14/technical-sessions/presentation/li_mu.
- [41] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, second, in progress ed., The MIT Press, Cambridge, Massachusetts, 2015. URL <https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf>.
- [42] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K., “Asynchronous Methods for Deep Reinforcement Learning,” *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, 2016, pp. 1928–1937. URL <https://proceedings.mlr.press/v48/mniha16.html>.
- [43] Xia, B., Lu, Q., Zhu, L., and Xing, Z., “An AI System Evaluation Framework for Advancing AI Safety: Terminology, Taxonomy, Lifecycle Mapping,” *arXiv preprint arXiv:2404.05388*, 2024. URL <https://arxiv.org/abs/2404.05388>, v3 [cs.SE] 15 May 2024.
- [44] Shah, J., Bikshandi, G., Zhang, Y., Thakkar, V., Ramani, P., and Dao, T., “FlashAttention-3: Fast and Accurate Attention with Asynchrony and Low-precision,” https://www.together.ai/blog/flashattention-3?utm_source=chatgpt.com, 2024. Colfax Research, Meta, NVIDIA, Princeton University, Together AI. Published July 11, 2024. Accessed 2025.
- [45] Alizadeh, K., Mirzadeh, I., Belenko, D., Khatamifard, S. K., Cho, M., Mundo, C. C. D., Rastegari, M., and Farajtabar, M., “LLM in a Flash: Efficient Large Language Model Inference with Limited Memory,” *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, Volume 1: Long Papers, Association for Computational Linguistics, 2024, pp. 12562–12584. URL <https://aclanthology.org/2024.acl-long.678>, aCL Anthology ID: 2024.acl-long.678.
- [46] Partnership for an Advanced Computing Environment (PACE), “PACE, Partnership for an Advanced Computing Environment,” <http://www.pace.gatech.edu>, 2017. Accessed: 2025-05-25.
- [47] Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B., “Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism,” *arXiv preprint arXiv:1909.08053*, 2020. doi:10.48550/arXiv.1909.08053, URL <https://arxiv.org/abs/1909.08053>, v4 [cs.CL] 13 Mar 2020.
- [48] NVIDIA Corporation, “NVIDIA Deep Learning Performance Documentation,” 2023. URL <https://docs.nvidia.com/deeplearning/performance/index.html>, last updated: July 27, 2023. Accessed: 2025-06-08.