

# MATHEMATICAL REASONING FOR UNMANNED AERIAL VEHICLES: A RAG-BASED APPROACH FOR COMPLEX ARITHMETIC REASONING

**Mehdi Azarafza**

Department of Computer Science  
Hamm-Lippstadt University of Applied Sciences  
mehdi.azarafza@hshl.de

**Mojtaba Nayyeri**

Institute for Artificial Intelligence  
University of Stuttgart  
mojtaba.nayyeri@ipvs.uni-stuttgart.de

**Faezeh Pasandideh**

Department of Computer Science  
Hamm-Lippstadt University of Applied Sciences  
faezeh.pasandideh@hshl.de

**Steffen Staab**

Institute for Artificial Intelligence  
University of Stuttgart  
Steffen.Staab@ipvs.uni-stuttgart.de

**Achim Rettberg**

Department of Computer Science  
Hamm-Lippstadt University of Applied Sciences  
achim.rettberg@hshl.de

## ABSTRACT

Autonomous UAV operation necessitates reliable mathematical reasoning for tasks such as trajectory planning and power management. While traditional flight control relies on hardcoded equations, recent Large Language Models (LLMs) offer potential for more flexible problem-solving but struggle with reliably selecting and applying correct mathematical formulations and executing precise multi-step arithmetic. We propose RAG-UAV, a retrieval-augmented generation framework designed to improve the mathematical reasoning of several LLMs (including GPT o1/Turbo, Llama-3.2/3.3, Mistral, and DeepSeek R1) in UAV-specific contexts by providing access to relevant domain literature. To conduct an initial assessment, we introduce the UAV-Math-Bench (a 20-question pilot problem set) of UAV-centric mathematical problems across four difficulty levels. Our experiments suggest that incorporating retrieval substantially increases exact answer accuracy (achieving up to 75% with o1), reduces instances of incorrect formulation selection (from 25% without RAG to 5% with RAG in a limited setting), and decreases numerical errors, reducing Mean Squared Error (MSE) by orders of magnitude for the best-performing models. This pilot study indicates that RAG can enable general-purpose LLMs to function as more reliable tools for engineering analysis, although direct real-time flight control requires further investigation and validation on a larger scale. All data are available<sup>1</sup>.

---

<sup>1</sup><https://github.com/Mehdiazarafza/UAV-RAG>

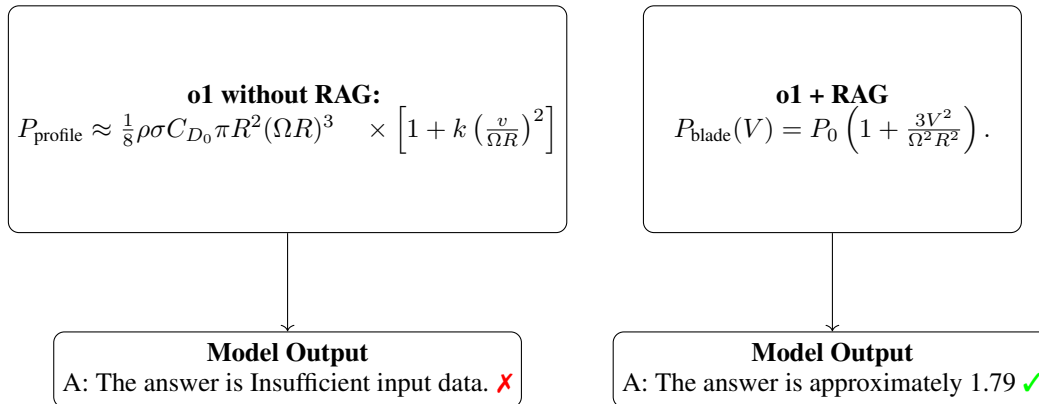


Figure 1: Comparison of Standard o1 and RAG with Prompting for Arithmetic Reasoning in UAV Applications

## 1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are aircraft that operate without an onboard pilot, enabling a wide range of applications from surveillance and mapping to delivery and emergency response. Autonomous UAV missions routinely demand multi-step mathematical reasoning; for instance, balancing energy budgets while re-planning swarms in variable wind. Existing flight-control code handles fixed equations, but engineers still spend time locating the right aerodynamic, communication channel, or power consumption formulas and checking the arithmetic. This process demands a solid understanding of physics, systems engineering, domain-specific formulas, and mathematical reasoning.

Large Language Models (LLMs), pretrained on vast corpora including physics texts and UAV design literature, offer a powerful new tool for assisting engineers. By leveraging their embedded knowledge, LLMs can help identify relevant equations, suggest performance evaluation strategies, validate reasoning steps, and even automate portions of UAV planning and analysis. This positions LLMs as promising co-pilots in the engineering design loop, augmenting human expertise and accelerating development cycles. While LLMs show promise for interpreting engineering queries and suggesting relevant design principles, they also exhibit important limitations. Their responses are constrained by the static nature of their pretraining data and may lack access to up-to-date, domain-specific documentation, standards, or experimental datasets. Moreover, LLMs may “hallucinate” (generating plausible but incorrect equations or assumptions), especially when faced with technical edge cases or incomplete prompts. Retrieval-Augmented Generation (RAG) offers a practical solution by grounding LLM outputs in authoritative external sources. In this hybrid approach, relevant documents are retrieved at query time and used to condition the model’s response. This improves factual accuracy, enhances trustworthiness, and enables more context-aware assistance for engineers working on UAV design and operations.

## 2 MOTIVATION EXAMPLE

Consider the task of computing the blade power of a UAV in forward flight, given specific parameters. As illustrated in Figure 1, a prominent LLM without access to external knowledge fails to identify the correct physical formulation, deeming the provided data “insufficient” (Appendix A.3). In contrast, when augmented with relevant domain literature via Retrieval-Augmented Generation (RAG), the same model successfully retrieves the appropriate equation, follows detailed reasoning steps, and computes an accurate result (Appendix A.5). This example highlights a key challenge: LLMs may possess implicit knowledge but struggle with reliable access and application of domain-specific mathematical models, motivating our investigation into RAG as a solution. Figure 1 illustrates the results generated by the o1 model for the query:

“Consider the following information:

- The constant value  $P_0$ : 1.5
- The forward speed  $v$ : 110 km/h
- The blade angular velocity  $\Omega$ : 300 radians/second
- The rotor radius  $R$ : 0.4 m

Based on this information, compute the Blade power of UAV in forward flight.”

### 3 CONTRIBUTION

In this work, we investigate how RAG can enhance the mathematical reasoning capabilities of LLMs for domain-specific UAV applications. We equip several LLMs with a vector-based retrieval mechanism over a corpus of UAV literature. Our contributions include:

- RAG-UAV Framework: We propose and implement a RAG framework tailored for improving mathematical and physical reasoning in UAV scenarios.
- UAV-Math-Bench: We construct a categorized evaluation set of 20 UAV-centric mathematical reasoning problems across four difficulty levels to facilitate initial evaluation.
- Empirical Analysis: We conduct preliminary experiments comparing RAG-augmented and non-RAG LLM performance (ChatGPT-Turbo), quantifying improvements in answer accuracy, formulation selection, and numerical error across different models and problem complexities.
- Analysis of RAG Benefits: We observe that RAG leads to substantial performance gains, particularly on more complex, multi-step problems, validating its potential for enhancing LLM-based engineering analysis in this domain.

### 4 RELATED WORK

We review the relevant literature in two key areas related to this work: (a) retrieval-augmented generation for reasoning, and (b) LLM in UAVs.

#### 4.1 RETRIEVAL-AUGMENTED GENERATION FOR REASONING

Retrieval-augmented generation (RAG) is a method that combines language model generation with conventional information retrieval to extend the knowledge and reasoning limitations of LLMs. RAG was shown to significantly boost performance on knowledge-intensive NLP tasks by allowing a seq2seq model to consult a non-parametric memory (a Wikipedia index) during generation Lewis et al. (2020). By conditioning on retrieved text, the model produced more accurate and specific outputs than a purely parametric baseline and even surpassed specialized retrieve-and-read pipelines on open-domain question answering. RAG’s general recipe is to embed the query, use vector similarity search to find semantically relevant documents, and then inject those documents (or snippets) into the model context.

While early applications of RAG focused on factual QA and knowledge retrieval, recent studies have adapted the concept to improve logical and mathematical reasoning. Other works have proposed related ideas, such as retrieving analogous solved problems to guide the model (a form of case-based reasoning) or using knowledge graphs to provide constraints during reasoning. Our approach leverages this insight by providing each arithmetic query with related reference content drawn from a prepared corpus (papers), effectively bridging pure linguistic reasoning with algorithmic assistance from examples.

#### 4.2 LLM IN UAVS

The intersection of LLMs and UAVs has opened new avenues for high-level autonomous decision-making in aerial robotics. Researchers have begun to integrate large language models into UAV control loops for tasks that require understanding complex instructions and making context-aware decisions. For example, TypeFly demonstrated an end-to-end system where an LLM translates natural language commands into drone flight actions, effectively piloting a quadcopter via language

Wang et al. (2025). Another work focuses on simulation and use of UAV and IoT sensor data in SUMO simulations. Similarly, the REAL framework incorporated an LLM as part of a UAV’s planning and control system, using the model to reason about goals and constraints during navigation. Tagliabue et al. (2024). Another work introduces an AI-driven framework that integrates Large Language Models (LLMs) with UAV and IoT sensor data in SUMO simulations Moraga et al. (2025). Our work aligns with this goal by tackling one particular weakness - mathematical reasoning – which is vital for UAV decision-making (e.g., calculating distances, timing sequences, or resource allocations). By improving mathematical accuracy through RAG, we aim to reduce one source of error in LLM-driven UAV systems, thereby contributing to safer and more trustworthy autonomous operations.

## 5 METHODOLOGY

### 5.1 RAG-UAV FRAMEWORK

We propose a RAG framework designed to augment LLMs with domain-specific mathematical knowledge. As illustrated in Figure 2, the framework operates in several stages: (1) A user query is input, typically a mathematical problem related to UAVs. (2) The query is embedded into a vector representation using a chosen embedding model. (3) This query embedding is used to perform a similarity search against a vector database containing embedded text segments from a corpus of UAV literature. (4) The most relevant text segments are retrieved. (5) The retrieved chunks are combined with the original user query into a structured prompt. (6) This augmented prompt is fed to the LLM, which generates a response that includes the solution to the mathematical problem.

### 5.2 CONSTRUCTING UAVS’ VECTOR DATABASE

Our knowledge base was constructed from a corpus of five research papers relevant to UAV mathematical modeling Wang et al. (2022); Ryu & Kim (2024); Liu et al. (2022); Yu et al. (2023); Abubakar et al. (2023). Initially, our investigation involved processing PDFs directly fed to the models. However, we identified inaccuracies in the models’ output that were subsequently attributed to errors during PDF parsing rather than inherent limitations in their core reasoning abilities. To accurately capture mathematical formulations, we processed the PDF documents using Mathpix<sup>2</sup> to convert them into LaTeX format. This process aimed to preserve equations and symbols more accurately than direct PDF parsing. The resulting LaTeX output was segmented into page-level chunks, yielding a total of 162 segments for the five papers. These segments were then embedded using two different models: all-mpnet-base-v2 (768 dimensions) Face (2025) for open-source LLMs and OpenAI’s text-embedding-ada-002 (1536 dimensions) Zilliz (2025) for OpenAI models. The resulting vector representations were stored in a Chroma vector database. For running open-source models, we used Ollama Ollama (2025), and for OpenAI models, we used the API. The corpus is limited to five papers and is intended as a controlled experimental setting rather than a comprehensive or diverse knowledge base. Some evaluation questions are derived from these sources.

### 5.3 LLM PROMPTING STRATEGY

The prompt template used to query the LLMs incorporates both the original problem statement and the retrieved text segments. The template instructs the model to utilize the provided documents for identifying relevant formulations but allows the model to use its internal capabilities for performing the numerical calculations. The full prompt template is provided in Appendix A.1.

### 5.4 UAV-MATH-BENCH: EVALUATION PROBLEM SET

To evaluate the complexity of computational tasks and analytical reasoning required in solving the provided problems, the questions are categorized into four distinct levels: Basic, Intermediate, Advanced, and Expert. The evaluation set serves to demonstrate the potential of the RAG-UAV framework and highlight performance differences between models and task complexity. This

<sup>2</sup><https://mathpix.com/>

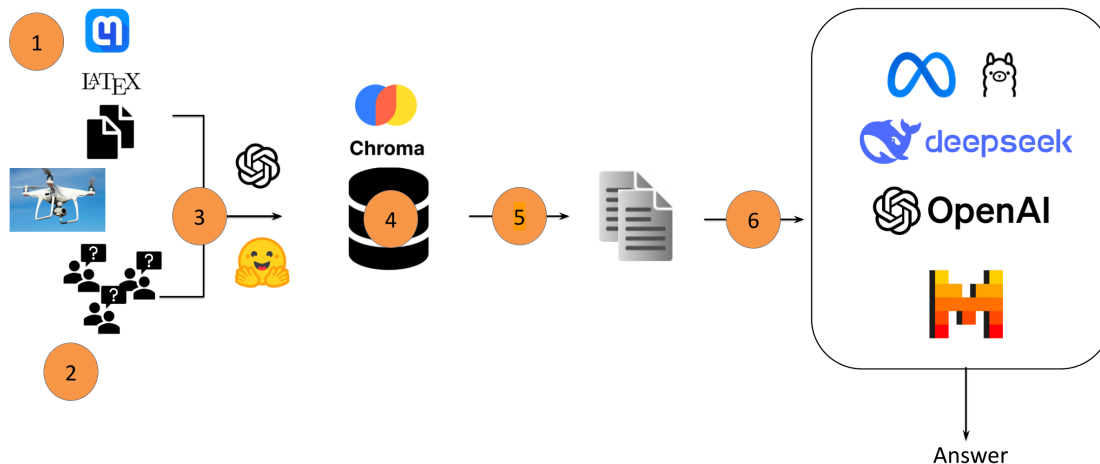


Figure 2: Workflow for RAG-UAV

classification considers the number of parameters involved, the mathematical complexity, and the depth of domain-specific knowledge required to obtain a solution. The development of UAVs is intrinsically linked to this increasing complexity. This evaluation set consists of 20 questions and is designed as a pilot study to explore model behavior rather than a statistically robust benchmark. As UAV technology advances, the challenges associated with their deployment grow more intricate, requiring sophisticated problem-solving techniques, advanced mathematical models, and a deep understanding of engineering principles. The categorization of difficulty levels thus reflects the rising complexity of real-world UAV-related problems encountered by professionals today.

#### 5.4.1 BASIC LEVEL QUESTIONS

Basic level questions require direct computation based on a given set of parameters, typically involving straightforward mathematical formulas. These problems involve fundamental arithmetic or algebraic manipulations without requiring extensive domain knowledge.

- **Q1-1:** Computation of elevation angle speed based on coordinate positions.
- **Q1-2:** Calculation of blade power using given constants and basic aerodynamics equations.
- **Q1-3:** Computational latency determination based on CPU allocation and task parameters.
- **Q1-4:** Atmospheric attenuation factor computation using given wavelength and scattering distribution.
- **Q1-5:** Received power computation using path loss and channel gain formulas.

At this level, UAVs are typically used for simple tasks such as altitude measurement and power consumption estimation, often relying on basic physics and aerodynamics.

#### 5.4.2 INTERMEDIATE LEVEL QUESTIONS

Intermediate questions extend beyond direct computations and require additional steps, such as probability computations or logarithmic transformations. These problems involve a moderate level of engineering knowledge and mathematical formulation.

- **Q2-1:** Probability computation of a line-of-sight (LoS) connection based on given probability constants.

- **Q2-2:** Induced power computation in forward UAV flight using additional aerodynamics principles.
- **Q2-3:** Data transmission rate calculation using Shannon’s theorem.
- **Q2-4:** Achievable data rate computation requiring logarithmic calculations.
- **Q2-5:** Power consumption due to communication, incorporating multiple power-related parameters.

At this intermediate level, UAVs become more complex in their operations, handling tasks such as communication protocols, energy management, and data transmission in various environmental conditions.

#### 5.4.3 ADVANCED LEVEL QUESTIONS

Advanced questions involve multi-step problem-solving approaches that integrate multiple mathematical principles. These computations require a solid grasp of signal processing, aerodynamics, and wireless communication concepts.

- **Q3-1:** Path loss calculation for a LoS connection incorporating frequency and speed of light.
- **Q3-2:** Parasite power computation based on aerodynamic drag and air density factors.
- **Q3-3:** Downlink transmission rate computation using transmission bandwidth and noise power.
- **Q3-4:** Bandwidth allocation to meet a specific data rate requirement.
- **Q3-5:** Power consumption due to hovering, integrating multiple aerodynamic and energy parameters.

In this phase, UAVs are employed for complex tasks such as networked communication between UAVs, long-duration flights, and advanced environmental monitoring, where power efficiency and reliable data transmission are critical.

#### 5.4.4 EXPERT LEVEL QUESTIONS

Expert-level questions demand an extensive understanding of multiple concepts and their integration into a single computational framework. These problems often require optimization, iterative solutions, or advanced probabilistic and statistical analyses.

- **Q4-1:** Average path loss computation involving LoS and NLoS path loss models.
- **Q4-2:** Total UAV power consumption in forward flight, integrating multiple power components.
- **Q4-3:** Transmission delay computation based on bandwidth, power, and noise parameters.
- **Q4-4:** Utility ratio determination requiring multi-variable calculations and resource allocation.
- **Q4-5:** Comprehensive UAV power consumption computation incorporating aerodynamic, communication, and electrical parameters.

At the expert level, UAVs are operating in highly sophisticated environments, with tasks requiring optimal energy management, high-level coordination, and complex decision-making algorithms, often for missions such as autonomous fleet management or large-scale environmental surveying. Table 1 categorizes the questions by difficulty, with additional details on the number of variables and nested equations.

Table 1: Categorization of questions by difficulty level, number of variables, and nested equations.

Difficulty	Question ID	# Variables	# Nested Eq.
Basic	Q1-1	2	1
	Q1-2	3	1
	Q1-3	2	1
	Q1-4	2	1
	Q1-5	3	1
Intermediate	Q2-1	2	2
	Q2-2	4	2
	Q2-3	3	1
	Q2-4	3	1
	Q2-5	4	2
Advanced	Q3-1	4	2
	Q3-2	5	2
	Q3-3	4	2
	Q3-4	3	2
	Q3-5	5	3
Expert	Q4-1	6	3
	Q4-2	6	3
	Q4-3	5	3
	Q4-4	6	4
	Q4-5	7	4

Each question corresponds to one UAV reasoning benchmark case.

## 6 EVALUATION AND RESULTS

This section presents the experimental setup and results demonstrating the impact of RAG on the mathematical reasoning performance of various LLMs on the UAV-Math-Bench. We compare RAG-augmented models against ChatGPT non-RAG baselines and other models with RAG using three primary metrics: Exact Answer Accuracy, Formulation Selection Error Rate, and Mean Squared Error (MSE). Due to the limited size of the evaluation set (20 questions), each example corresponds to a 5% change in accuracy. Therefore, results should be interpreted as indicative trends rather than statistically significant findings.

### 6.1 EVALUATION METRICS

**Exact Answer Accuracy:** This metric measures the percentage of problems for which the model produces a final numerical answer that precisely matches the ground truth answer provided by human experts. Answers must match within a predefined tolerance for floating-point comparisons ( $10^{-4}$ ), where applicable.

**Formulation Selection Error Rate:** This metric quantifies the percentage of problems where the model either selects an incorrect mathematical formula or fails to identify any relevant formula from the provided context. This is assessed through the expert manual inspection of the model’s step-by-step reasoning process.

**Mean Squared Error (MSE):** We calculate the MSE between the model’s final numerical answer and the ground truth answer across the benchmark. Due to potentially large differences between model outputs and ground truths, we report results using  $\log_{10}(\text{MSE})$  for clearer visualization. A lower MSE indicates closer agreement with the reference answer.

Table 2: Performance comparison of evaluated LLMs.

Metric	Model							
	Chat	o1	o1-p	Turbo	M7	L3	R1	L70
COR	30	75	70	60	5	10	50	35
CFO	75	95	85	75	70	60	90	65
MSC	45	15	10	20	55	50	35	30
MisC	60	21.1	11.8	26.7	78.6	83.3	38.9	46.2

All values are percentages. COR: Correct; CFO: Correct formulation; MSC: Miscalculation; MisC: Miscalculation among correctly formulated answers. Chat = ChatGPT; M7 = Mistral 7B; L3 = LLaMA 3.2 3B; L70 = LLaMA 3.3 70B; R1 = DeepSeek R1.

## 6.2 COMPARISON OF MODEL PERFORMANCE

To ensure a comprehensive comparison, we employed the correct formulations and verified the answers in all LLMs. The complete set of answers, the corresponding questions and the details of the model are available on the GitHub repository of the paper.

Table 2 compares the performance of several LLMs based on key metrics. We evaluated both correct and incorrect answers; if a model did not answer or asked for more information, we counted that as zero. ChatGPT is the only model without RAG. In this work, a full ablation is not included for all models (with and without RAG). Among the models, o1 performed the best, with 75% correct answers and the lowest error in calculations. OpenAI’s o1 model consistently identified the correct formulation and computed values close to the reference answers, resulting in a considerably lower MSE compared to others. This model also provided optimized and well-structured answers, demonstrating high reasoning capabilities for complex questions.

Notably, OpenAI’s o1 model demonstrated a degree of semantic robustness, capable of interpreting physical constants such as the speed of light despite variations in user input, including misspellings and non-standard variable assignments. For instance, when presented with the statement defining ‘G’ as the “speed light”, the model inferred the intended concept, correlating ‘c’ with the symbol for the speed of light.

Models like Mistral 7B and LLaMA 3.2 3B gave mostly wrong answers and made many mistakes. Mistral 7B demonstrated lower performance overall. DeepSeek 70B Guo et al. (2025) and LLaMA 3.3 70B gave more accurate answers than smaller models, and DeepSeek made fewer mistakes overall. DeepSeek provides very detailed and precise answers (Appendix A.6) and is considered the best open-source LLM model in our evaluation. ChatGPT-4-Turbo without RAG and Turbo with RAG were in the middle - they could answer some questions but often miscalculated. This shows that larger models and better training are important in answering technical questions such as those about RAG UAVs. The o1-preview model demonstrated the most robust performance, exhibiting a low miscalculation rate of 11.76% when correct formulations were achieved. In contrast, Mistral 7B and Llama 3.2 3B showed markedly higher miscalculation rates (78.57% and 83.33%, respectively), suggesting a significant challenge in maintaining computational accuracy for these models.

Figure 4 compares the performance of several LLMs based on MSE. Since MSE values are extremely large due to the significant differences between human responses and model predictions, we used a  $\log_{10}$  scale to make the evaluation clearer. The results show that o1 outperforms all other models dramatically with an MSE of just 214.74. In particular, significant deviations can occur depending on the model’s ability to select and apply the correct formulation. For example, in one benchmark question, the expert-provided ground truth is 64,207,051.66, while ChatGPT returned a significantly lower value of 70.2. Similar differences are observed in several models. However, the o1 model consistently identified the correct formulation and computed values close to the reference answers, resulting in a considerably lower MSE compared to others.

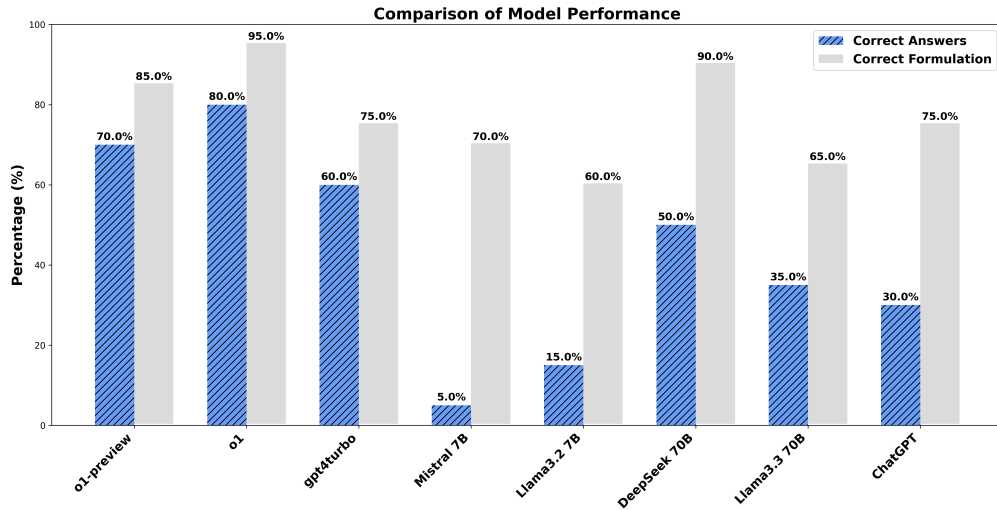
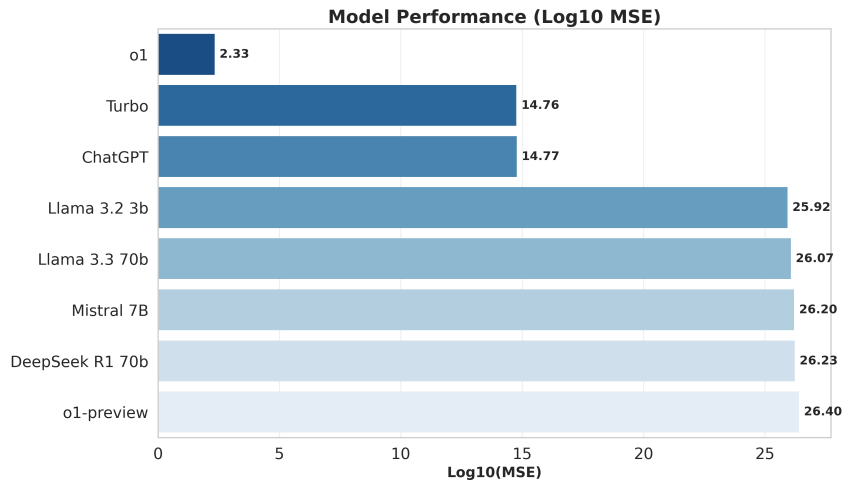


Figure 3: Compare state-of-the-art LLMs

Figure 4: MSE Comparison ( $\log_{10}(\text{MSE})$ )

### 6.3 PERFORMANCE BY DIFFICULTY LEVEL

In this section we provide a comparative evaluation of eight large language models across four difficulty tiers, summarized in Table 3.

The o1 model outperforms all others, achieving 80% accuracy even on Expert-level questions. While most models handle Basic questions well, performance drops notably for harder levels. Models like Mistral 7B and LLaMA 3.2 3B perform poorly overall, especially on Advanced and Expert tasks. ChatGPT and DeepSeek R1 show moderate accuracy, but only o1 maintains consistently high performance across all levels. Interestingly, the o1 model achieved 80% accuracy on Expert-level problems, a performance not mirrored at the Intermediate level, where two errors occurred. One specific instance involved a slight numerical deviation in the model’s output (0.5535 vs. the correct 0.5), classified as incorrect under rigorous human expert assessment, even though the model itself indicated the result was ‘approximately 0.50.’ For another mistake from the model, there was a unit conversion error: we assumed  $\theta$  is in radians and applied a conversion to degrees, but o1 already considered the angle in degrees. This led to an incorrect input to the exponential term and significantly altered the result. While Expert problems involve more complex chains of reasoning and integration of multiple concepts, the model might excel at retrieving and structuring these high-level

Table 3: Accuracy comparison across difficulty levels for evaluated models.

Level	Model							
	Chat	o1	o1-p	Turbo	M7B	L3B	R1	L70B
Basic	60	80	100	100	20	0	60	60
Intermediate	0	60	80	60	0	20	40	60
Advanced	40	80	60	60	0	0	80	20
Expert	20	80	40	20	0	20	20	0

All values are percentages. Chat = ChatGPT; o1-p = o1-preview; M7B = Mistral 7B; L3B = LLaMA 3.2 3B; R1 = DeepSeek R1; L70B = LLaMA 3.3 70B.

solutions. However, it may be less robust to handling refined details like precise numerical tolerances or implicit unit conventions that were critical for correctness in some Intermediate problems.

#### 6.4 EXAMINATION OF ATTENTION PATTERNS

Understanding the internal processes of LLMs, particularly in complex tasks like mathematical reasoning, remains an active research area Gao et al. (2023). To gain preliminary insights into how the DeepSeek R1 model processes input and potentially leverages information, we visualized the attention patterns of its first layer for a selected question and its generated answer. The figure 5 shows attention patterns for head 15 of the first layer, revealing similar diagonal and localized attention structures within the generated sequence.

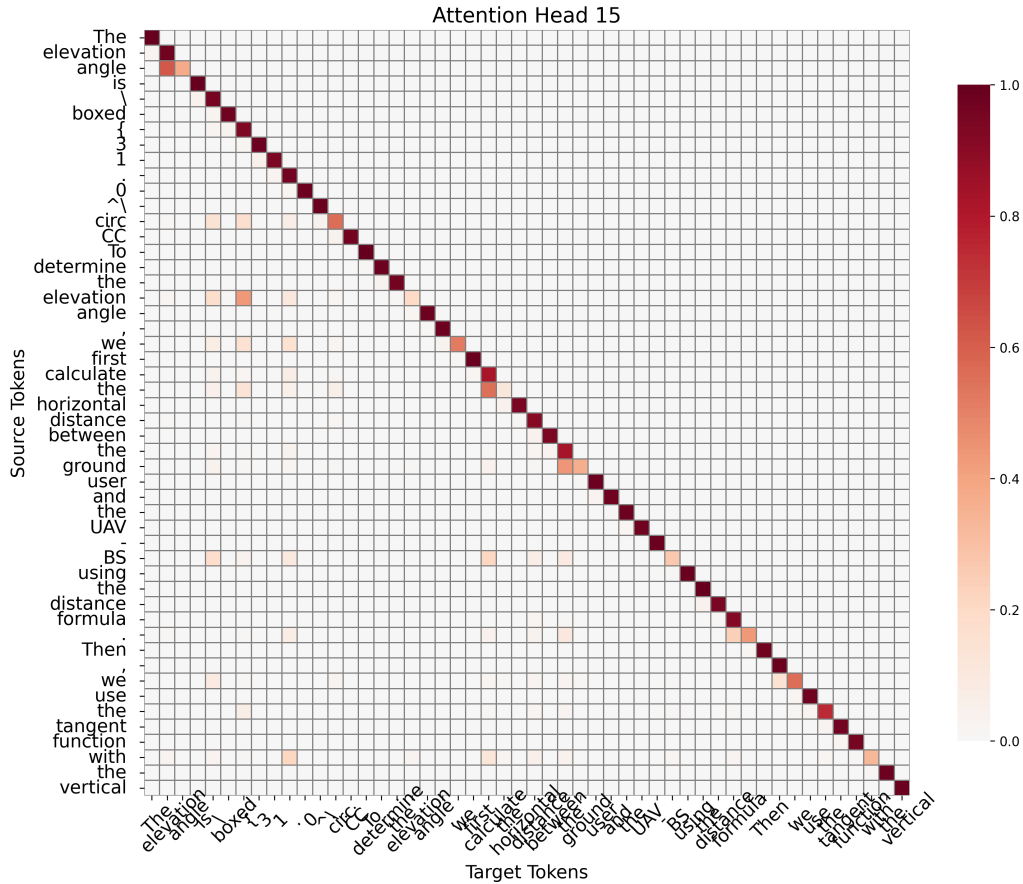


Figure 5: Head 15 (Layer 1) Attention Matrix of DeepSeek R1

## 7 CONCLUSION

In this work, we demonstrated that RAG can substantially improve the mathematical reasoning capabilities of LLMs, particularly within the context of domain-specific UAV applications requiring access to physical models and accurate multi-step computations. By providing LLMs with relevant external knowledge via RAG, we observed a significant increase in their ability to correctly select formulations and compute accurate numerical results on a set of UAV-centric mathematical problems. Our experiments showed consistent gains in accuracy and substantial reductions in numerical error across several LLMs when augmented with RAG. These improvements were most pronounced on problems involving multiple steps and nested operations, suggesting that RAG effectively mitigates common failure modes of standalone LLMs on complex mathematical tasks. While this study utilizes a limited problem set, the findings indicate that RAG holds significant potential for enabling LLMs to function as more reliable tools for engineering analysis in technical domains like UAV development, by providing them with the necessary domain knowledge to perform complex calculations more accurately.

## ACKNOWLEDGMENTS

This paper is the result of preliminary work by Hamm-Lippstadt University of Applied Sciences, Germany, on the EdgeAI-Trust project “Decentralized Edge Intelligence: Advancing Trust, Safety, and Sustainability in Europe.” This work was supported by the EdgeAI-Trust project under grant agreement No 101139892, funded by the Chips Joint Undertaking (Chips JU). The Chips JU receives support from the European Union’s Horizon Europe research and innovation program and from National Authorities, particularly for this project, from Germany, Austria, France, Italy, Spain, Denmark, Belgium, Turkey, Greece, Latvia, Cyprus, and Slovenia.

## REFERENCES

- Attai Ibrahim Abubakar, Michael S. Mollel, Oluwakayode Onireti, Metin Ozturk, Iftikhar Ahmad, Syed Muhammad Asad, Yusuf Sambo, Ahmed Zoha, Sajjad Hussain, and Muhammad Ali Imran. Coverage and throughput analysis of an energy efficient uav base station positioning scheme. *Computer Networks*, 232:109854, 2023.
- Hugging Face. all-mpnet-base-v2: Pretrained sentence transformer model. <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>, 2025. Accessed: March 25, 2025.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, et al. Deepseek-r1: Incentivizing reasoning capability in large language models via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-Tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- Qian Liu, Hairong Liang, Rui Luo, and Qilie Liu. Energy-efficiency computation offloading strategy in uav-aided v2x network with integrated sensing and communication. *IEEE Open Journal of the Communications Society*, 3:1337–1346, 2022.
- Alejandro Moraga, Javier de Curto, Ignacio de Zarza, and Carlos T. Calafate. Ai-driven uav and iot traffic optimization: Large language models for congestion and emission reduction in smart cities. *Drones*, 9:248, 2025.

Ollama. Ollama open-source model platform. <https://ollama.com/>, 2025. Accessed: March 10, 2025.

Kyungho Ryu and Wooseong Kim. Energy efficient deployment of aerial base stations for mobile users in multi-hop uav networks. *Ad Hoc Networks*, 157:103463, 2024.

Andrea Tagliabue, Kota Kondo, Tong Zhao, Mason Peterson, Claudius T. Tewari, and Jonathan P. How. Real: Resilience and adaptation using large language models on autonomous aerial robots. In *Proceedings of the IEEE 63rd Conference on Decision and Control (CDC)*, pp. 1539–1546, 2024.

Leiyu Wang, Haixia Zhang, Shuaishuai Guo, and Dongfeng Yuan. Deployment and association of multiple uavs in uav-assisted cellular networks with the knowledge of statistical user position. *IEEE Transactions on Wireless Communications*, 21(8):6553–6567, 2022.

Wenhao Wang, Yanyan Li, Long Jiao, and Jiawei Yuan. Gsce: A prompt framework with enhanced reasoning for reliable llm-driven drone control. *arXiv preprint arXiv:2502.12531*, 2025.

Liangkun Yu, Xiang Sun, Sihua Shao, Yougan Chen, and Rana Albelaihi. Backhaul-aware drone base station placement and resource management for fso-based drone-assisted mobile networks. *IEEE Transactions on Network Science and Engineering*, 10(3):1659–1668, 2023.

Zilliz. text-embedding-ada-002: Ai model for text embeddings. <https://zilliz.com/ai-models/text-embedding-ada-002>, 2025. Accessed: March 25, 2025.

## A APPENDIX

### A.1 PROMPT

#### LLM Prompt

To answer the questions, use only the documents provided. Formulations are provided for all the questions in the provided documents; however, to calculate the result, you may use your calculation skills:

context

Question: question

### A.2 QUESTION 1-2

#### Question 1-2

Consider this information:

- The constant value  $p_0$ : 1.5
- The forward speed  $v$ : 110 km h<sup>-1</sup>
- The blade angular velocity  $\Omega$ : 300 rad s<sup>-1</sup>
- The rotor radius  $R$ : 0.4 m

Based on this information, compute the Blade power of UAV in forward flight?

## A.3 O1 RESPONSE WITHOUT RAG

**Part of the o1 Response without RAG for Question 1-2**

In helicopter-type rotor analyses, one typically needs at least:

- The vehicle’s weight (or total thrust the rotor must produce).
- The air density ( $\rho$ ).
- Basic blade-geometry parameters (number of blades, chord, airfoil drag coefficient, etc.).
- A suitable rotor-theory formula (e.g., blade-element or momentum theory).

Simply giving

- $p_0 = 1.5$  (an unspecified “constant”),
- forward speed  $v = 110$  km/h ( $\approx 30.6$  m/s),
- angular velocity  $\Omega = 300$  rad/s,
- rotor radius  $R = 0.4$  m

does not pin down the aerodynamic power.

**WHY MORE INFORMATION IS NEEDED**

1. In forward flight, total rotor power typically has at least two major components:
  - Induced (or “lift-induced”) power, often estimated from momentum theory.
  - Profile (or “parasite”) power due to drag on the spinning blades themselves, often estimated from blade-element theory.
2. Both components depend on air density ( $\rho$ ) and on how much thrust the rotor must generate (which in turn depends on the UAV’s weight and flight condition).
3. Even if one assumes the thrust requirement is known, there are empirical or semi-empirical “constants” in the formulas (such as blade drag coefficients, tip-loss factors, etc.) that must be specified or lumped into a known expression before a numeric answer can come out.

**TYPICAL EXAMPLE FORMULAS**

By way of illustration (not a complete solution), in many helicopter texts one finds a “profile-power” formula of the form

$$P_{\text{profile}} \approx \frac{1}{8} \rho \sigma C_{D_0} \pi R^2 (\Omega R)^3 \times \left[ 1 + k \left( \frac{v}{\Omega R} \right)^2 \right],$$

where

- $\sigma$  is the rotor solidity (depends on blade chord and number of blades),
- $C_{D_0}$  is the airfoil’s drag coefficient,
- $k$  is an empirical factor ( $\approx 4$  to  $6$ ),
- $v$  is forward speed, and
- $\Omega R$  is the blade tip speed.

Even here, you can see one must know  $\rho$ ,  $\sigma$ ,  $C_{D_0}$ , and so on.

**CONCLUSION**

Because the problem as posed does not provide the rotor thrust (or vehicle weight), the air density, or enough blade-geometry/aerodynamic details, there is no way to turn the given numbers into a single “watts” (or kW) value for rotor power. One can only proceed once those other parameters and the specific power formula are specified.

## A.4 ANSWER FROM O1 FOR QUESTION 1-2 WITH RAG

Question 1-2:

**GPT o1 Answer with RAG**

Consider this information:

- The constant value  $p_0$ : 1.5
- The forward speed  $v$ : 110 km h<sup>-1</sup>
- The blade angular velocity  $\Omega$ : 300 rad s<sup>-1</sup>
- The rotor radius  $R$ : 0.4 m

Based on this information, compute the Blade power of UAV in forward flight?

## A.5 O1 RESPONSE WITH RAG

**o1 Response with RAG for Question 1-2**

Below is a concise way to obtain the “blade power” term for forward flight from the documents. From (66) in the provided text, the blade-profile (or “blade”) power in forward flight is the first bracketed term, namely

$$P_{\text{blade}}(V) = P_0 \left( 1 + \frac{3V^2}{\Omega^2 R^2} \right).$$

Given Data

- $P_0 = 1.5$  (constant), Forward speed  $V = 110$  km h<sup>-1</sup>
- Blade angular velocity  $\Omega = 300$  rad s<sup>-1</sup>, Rotor radius  $R = 0.4$  m

Calculation Steps Step 1: Convert V into meters per second

$$V(\text{m s}^{-1}) = 110 \times \frac{1000}{3600} \approx 30.56 \text{ m s}^{-1}.$$

Step 2: Compute the ratio  $V^2/(\Omega^2 R^2)$ 

- $\Omega^2 R^2 = (300 \text{ rad s}^{-1})^2 \times (0.4 \text{ m})^2 = 90\,000 \times 0.16 = 14\,400 \text{ (m s}^{-1})^2$ .
- $V^2 \approx (30.56 \text{ m s}^{-1})^2 \approx 933.9136 \text{ (m/s)}^2$ .
- Hence,  $\frac{V^2}{\Omega^2 R^2} \approx \frac{933.9136}{14400} \approx 0.064855$ .

(Using the input’s rounded value for  $V^2/(\Omega^2 R^2) \approx 0.0649$  for the next step as per the provided text.)

Step 3: Substitute into the blade-power expression: Using the approximation  $V^2/(\Omega^2 R^2) \approx 0.0649$ :

$$1 + 3 \times 0.0649 \approx 1 + 0.1947 = 1.1947.$$

$$P_{\text{blade}}(V) = 1.5 \times 1.1947 \approx 1.79205.$$

(Rounding to 1.79 as per the provided text.)

Result: Thus, the blade power at  $V = 110$  km h<sup>-1</sup> is approximately 1.79 (in the same power units as  $P_0$ ).

## A.6 DEEPSEEK R1 ANSWER FOR QUESTION 1-1

Question 1-1:

**DeepSeek R1 Answer with RAG for Question 1-1**

Consider this information:

1. Ground user's position: (47,23)m
2. UAV-BS's position: (65,73,15)m

Based on this information, what speed is elevation angle?

**Part of the DeepSeek R1 Response with RAG**

So, first step: calculate the horizontal distance between the two points. Since both are given in x and y coordinates, I can use the Euclidean distance formula for 2D:

$$\text{Horizontal distance} = \sqrt{(65 - 47)^2 + (73 - 23)^2} \quad (1)$$

Let me compute that:

$$65 - 47 = 18$$

$$73 - 23 = 50$$

So, horizontal distance =

$$\text{Horizontal distance} = \sqrt{18^2 + 50^2} = \sqrt{324 + 2500} = \sqrt{2824} \quad (2)$$

Calculating  $\sqrt{2824}$ . Let's see:  $53^2$  is 2809, so  $\sqrt{2824}$  is a bit more than 53. Maybe approximately 53.11 meters.

Now, the vertical distance is given as 15 meters.

So now I have a right triangle with opposite side (vertical) = 15m and adjacent side (horizontal)  $\approx$  53.11m.

The elevation angle  $\theta$  can be found using the tangent function:

$$\tan(\theta) = \frac{\text{opposite}}{\text{adjacent}} = \frac{15}{53.11} \quad (3)$$

Let me compute that: 15 divided by approximately 53.11 is roughly 0.2823.

Now, to find  $\theta$ , I take the arctangent of 0.2823. Using a calculator,  $\arctan(0.2823)$  is about 15.7 degrees.

Wait, let me double-check my calculations because sometimes it's easy to mix up the sides. The elevation angle is indeed calculated using  $\tan(\theta) = \text{opposite}/\text{adjacent}$ , which in this case is vertical over horizontal distance. So yes, that should be correct.

Alternatively, I could use the sine function since  $\sin(\theta) = \frac{\text{opposite}}{\text{hypotenuse}}$ . Let me compute the hypotenuse first:

$$\text{Hypotenuse} = \sqrt{\text{horizontal}^2 + \text{vertical}^2} = \quad (4)$$

$$\sqrt{53.11^2 + 15^2} \quad (5)$$

Calculating 53.11 squared: approx 2824, and 15 squared is 225, so total under the root is  $2824 + 225 = 3049$ .

$$\text{Hypotenuse} \approx \sqrt{3049} \approx 55.22 \text{ meters} \quad (6)$$

Then

$$\sin(\theta) = \frac{15}{55.22} \approx 0.2717 \quad (7)$$

Taking arcsin of that gives  $\theta \approx 15.75$  degrees, which aligns with my previous calculation using tangent. So both methods give me around 15.7 to 15.8 degrees, so I can be confident it's approximately 15.7 degrees.

**Continue answer for DeepSeek R1: Answer with RAG**

I think that's the elevation angle between the ground user and the UAV-BS based on their positions.

The elevation angle between the ground user at (47,23)m and the UAV-BS at (65,73,15)m is calculated as follows:

**1. Horizontal Distance Calculation:**

- $\Delta x = 65 - 47 = 18\text{m}$
- $\Delta y = 73 - 23 = 50\text{m}$
- Horizontal distance =  $\sqrt{18^2 + 50^2} \approx 53.11\text{m}$

**2. Vertical Distance:**

- Vertical distance = 15m (z-coordinate of UAV-BS)

**3. Elevation Angle Calculation using Tangent:**

$$\tan(\theta) = \frac{\text{opposite}}{\text{adjacent}} = \frac{15}{53.11} \approx 0.2823 \quad (8)$$

$$\theta \approx \arctan(0.2823) \approx 15.7^\circ \quad (9)$$

**4. Verification using Sine:**

$$\text{Hypotenuse} = \sqrt{53.11^2 + 15^2} \approx 55.22\text{m} \quad (10)$$

$$\sin(\theta) = \frac{15}{55.22} \approx 0.2717 \quad (11)$$

$$\theta \approx \arcsin(0.2717) \approx 15.75^\circ \quad (12)$$