EVE: A Domain-Specific LLM Framework for Earth Intelligence

Àlex R. Atrio[1,*,†], Antonio Lopez[1,*], Jino Rohit[1,*], Yassine El Ouahidi[2], Marcello Politi[1], Vijayasri Iyer[1], Umar Jamil[2], Sébastien Bratières[1,3], Nicolas Longépé[4]

[1]Pi School, [2]Mistral AI, [3]Translated, [4]ESA Φ-lab

Abstract

We introduce **Earth Virtual Expert (EVE)**, the first open-source, end-to-end initiative for developing and deploying domain-specialized LLMs for Earth Intelligence. At its core is `EVE-Instruct`, a domain-adapted 24B model built on Mistral Small 3.2 and optimized for reasoning and question answering. On newly constructed Earth Observation and Earth Sciences benchmarks, it outperforms comparable models while preserving general capabilities. We release curated training corpora and the first systematic domain-specific evaluation benchmarks, covering MCQA, open-ended QA, and factuality. EVE further integrates RAG and a hallucination-detection pipeline into a production system deployed via API and GUI, supporting 350 pilot users so far. All models, datasets, and code are ready to be released under open licenses as contributions to our field at huggingface.co/eve-esa and github.com/eve-esa.

1 Introduction

Earth Observation (EO) (the acquisition of information about Earth's physical, chemical, and biological systems using remote sensing technologies, primarily satellites) and Earth Sciences research generates vast amounts of high-value knowledge. Yet this knowledge remains fragmented across heterogeneous sources and formats, creating a significant barrier for potential users such as practitioners and decision makers. It also remains crucial for experts in Earth Observation and the wider Earth sciences to continually broaden their understanding across adjacent fields, given the increasingly interdisciplinary nature of modern environmental challenges. Accessing this information typically requires deep expertise, limiting comprehensive understanding. This fragmentation creates a trust gap: decision-makers require transparent and scientifically robust EO and Earth Sciences insights that traditional systems struggle to provide (Knutti, 2019). As the community moves toward Earth Action, the ability of decision systems to support environmental decisions and interventions, there is growing demand for systems that not only retrieve information, but interpret and reason across heterogeneous sources. Earth Intelligence (EI) aims to provide this integrative reasoning layer to support informed and reliable decision-making.

Recent advances in LLMs enable natural-language interaction with complex knowledge ecosystems, yet general-purpose models lack the domain specificity and rigorous evaluation required for reliable EI applications. Addressing this gap requires an end-to-end approach combining domain adaptation, retrieval grounding, reliability mechanisms, and deployment.

We introduce EVE, an open and modular framework for EO and Earth Sciences, developed within an initiative of the European Space Agency (ESA) Φ-lab and deployed in a six-month pilot serving 350 users. The system integrates heterogeneous knowledge sources, including encyclopedic, institutional, and scientific publisher content, enabling grounded reasoning across diverse EO and Earth Sciences domains. Our contributions are:

1. `EVE-Instruct`: a specialized 24B LLM for EI.
2. A curated EO and Earth Sciences corpus (2.8B tokens) and large-scale synthetic instruction dataset (10.7B total tokens).
3. The first manually created EO and Earth Sciences evaluation benchmarks (5693 samples) covering diverse forms of Question-Answering (QA) and factuality.
4. A deployed RAG- and hallucination-aware chat system accessible via GUI and API.
5. Open release of models, datasets, and code

*Equal contribution.
†Corresponding author: alex.atrio@picampus-school.com

to support reproducible domain-specific LLM development.

In our experiments, `EVE-Instruct` consistently outperforms comparable models in its size range on our specific benchmarks, while preserving general capabilities, demonstrating that carefully engineered domain-specific systems can achieve strong practical performance without substantially increasing model size.

2 Related Work

Recently, domain-specialized LLMs are increasingly achieving performance comparable to or exceeding that of general-purpose models, contingent upon several parts of the system design. Large-scale pretraining corpora reflect different tradeoffs in diversity, scale, filtering, and reproducibility. The Pile (Gao et al., 2021) integrates heterogeneous sources to maximize coverage, RedPajama-V2 (Weber et al., 2024) emphasizes scale and flexible quality control, Dolma (Soldaini et al., 2024) prioritizes reproducible preprocessing, and FineWeb (Penedo et al., 2024) focuses on large-scale filtering and deduplication, including an instructional subset. Together, these efforts advance general-domain data curation but do not directly address the challenges of domain-specific language modeling.

In scientific and EO or Earth Sciences domains, corpus design and domain-adaptive pretraining is central to performance. INDUS (Bhattacharjee et al., 2024) and K2 (Deng et al., 2024) show that curated scientific corpora and continuous pretraining strengthen domain fidelity and reasoning, while AstroLLaMA (Nguyen et al., 2023), AstroMLab (de Haan et al., 2024), and COSMOSAGE (de Haan, 2025) demonstrate similar gains by training on scientific publications and observational data, including at compact model scales.

Beyond corpus adaptation, recent work explores spatial reasoning and geospatial system integration. GeoLLM (Manvi et al., 2023) shows that LLMs encode geographic knowledge that can be enhanced through grounding with structured geodata. Frameworks like GeoGPT (Zhang et al., 2024a) and BB-GeoGPT (Zhang et al., 2024b) combine LLMs with GIS toolchains for spatial analysis, while agent-based approaches such as GeoAgent (Chen et al., 2024), UrbanGPT (Li et al., 2024), and ChatGeoAI (Mansourian and Oucheikh, 2024) enable autonomous and conversational geospatial reasoning.

Several frameworks address hallucination evaluation in LLMs. FEVER (Thorne et al., 2018), TruthfulQA (Lin et al., 2022), and HaluEval (Li et al., 2023a) provide fact-verification and truthfulness benchmarks, including different hallucination types. LLM-Oasis (Scirè et al., 2025) introduces a large-scale benchmark for end-to-end factuality evaluation. SelfCheckGPT (Manakul et al., 2023) proposes reference-free hallucination detection, while RARR (Gao et al., 2023) reduces factual errors through retrieval-based attribution and revision.

3 EVE System Overview

The deployed EVE system consists of modular components that jointly generate grounded responses (Figure 1):

- **`EVE-Instruct`**: core LLM for answer generation, query rewriting, and summarization (Section 6).
- **Knowledge Bases (KB)**: curated domain-specific sources (open-access, proprietary,[1] ESA documents, and private collections) totaling ∼365k documents, supporting hybrid semantic and metadata retrieval (Section 8).
- **Retrieval Pipeline**: selects relevant documents based on query content and filters (Section 8).
- **Chat System**: manages dialogue state, memory, and context allocation (Appendix D).

4 EO and Earth Sciences Text Corpus

We curate a large-scale EO and Earth Sciences corpus by manually selecting 172 sources spanning 22 trusted publishing institutions (see Table 6 in Appendix A) using a custom scraping framework. The final corpus is comprised of 5.3B tokens: 4.2B from open-access sources and 1.1B from Wiley proprietary content (see Section 3). We use filtered subsets of this corpus for synthetic data generation (Section 6.1) and RAG (Section 8). We publicly release 2.8B tokens of the open-access portion in due consideration of licensing conditions, as detailed in Appendix A.

Our data processing pipeline transforms raw documents into clean, structured training data. First, we extract machine-readable text from the original files with Trafilatura (Barbaresi, 2021) for

[1] Provided under a partnership agreement with Wiley.

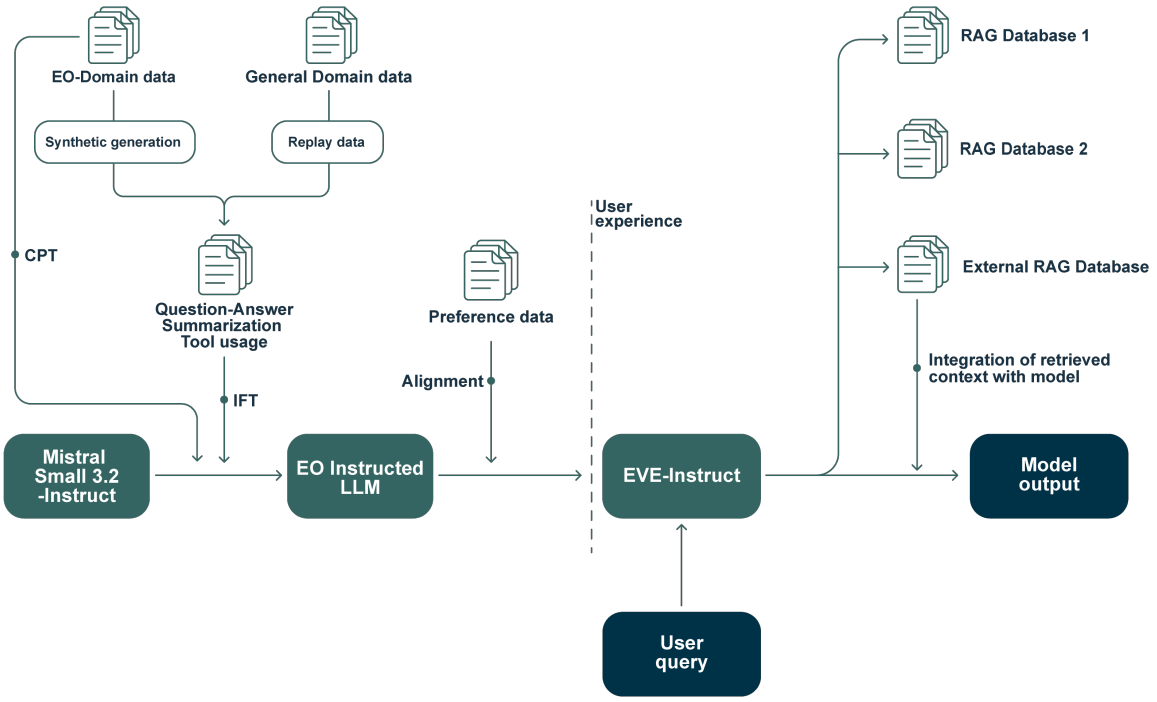

Figure 1: System architecture of EVE depicting component interactions.

HTML, and Nougat (Blecher et al., 2023) OCR for PDFs, selected after benchmarking for LaTeX-based formula and table extraction. We apply SHA-256 based hashing at file level to remove exact duplicates and MinHash LSH[2] to remove near-duplicate text segments within the file itself. We perform lightweight post-processing to correct OCR noise and malformed LaTeX using rule-based normalization and an LLM-based syntax repair module. We use Microsoft Presidio[3] with Flair `ner-english-large` model (Akbik et al., 2019) to anonymize author names as `[AUTHOR]` and emails as `[EMAIL]` to ensure removal of personally identifiable information. We extract structured metadata (e.g., DOI, URL, title, journal) for academic PDFs by identifying DOIs via regular expressions and querying CrossRef API.[4]

5 EO and Earth Sciences Benchmark

Due to the lack of standardized benchmarks for dialogue and NLP capabilities in EO and Earth Sciences, we construct manually curated evaluation sets targeting domain-relevant tasks (Table 1). To our knowledge, these constitute the first systematic benchmarks within the domain for language modeling. The datasets are built in two stages: candidate samples are generated by both humans and LLMs, and subsequently reviewed and refined by independent human annotators. We recruited 25 EO and Earth Sciences experts as human annotators and provided them with annotation guidelines. Table 2 provides representative examples from each category.

Task Type	Samples
MCQA (Multiple Answer)	431
MCQA (Single Answer)	1261
Hallucination Detection	2326
Open-Ended QA (No context)	1257
Open-Ended QA (with Context)	418

Table 1: EO and Earth Sciences benchmark suite. Multiple Choice Question Answering (MCQA) differ in the number of correct options. Hallucination detection is a balanced binary QA classification task. Open-ended QA evaluate self-contained and context-grounded questions.

6 Model Fine-tuning

Adapting an instruction-tuned LLM to our target domain requires incorporating domain-specific knowledge without degrading its instruction-following, conversational stability, or tool-use behavior. The 5.3B tokens of EO and Earth Sciences text (Section 4) are sufficient to consider

[2] github.com/ekzhu/datasketch
[3] github.com/microsoft/presidio
[4] crossref.org

Task Type	Example
MCQA (Multiple Answer)	**Q:** Which statements are true about surging glaciers? A) Surging glaciers are subject to cyclical flow instabilities B) Surging glaciers flow at a constant rate down a mountain C) [. . .] a glacier's velocity moves up to 100 times faster than normal D) Surging glaciers suddenly increase in mass → **Answers: A, C**
MCQA (Single Answer)	**Q:** What is the main obstacle to using optical EO in the aftermath of a severe weather event? A) Sensor calibration issues B) Atmospheric interference C) Low light conditions D) Cloud cover → **Answer: D**
Hallucination Detection	**Q:** What advantages do radar systems offer for environmental monitoring compared to optical sensors? **Hallucinated Answer:** Radars are active sensors that create their own electromagnetic signals. Radars cannot operate at night and are ineffective in hazy or smoky environments. SAR [. . .] uses the movement between the antenna and the target area [. . .] to create high-resolution images for remote sensing. **Correct Answer:** Radars are active sensors [. . .] Radars can operate day and night, and penetrate clouds, haze, and smoke. SAR [. . .] → **Label: Hallucinated** (underlined span)
Open-Ended QA	**Q:** What is the calving front of an ice stream? **A:** The point where icebergs are lost from a glacier.
Open-Ended QA w/ Context	**Q:** How does LiDAR map buildings? *[3 source documents provided as context]* **A:** LiDAR maps buildings by sending laser pulses that reflect off rooftops, walls, and other structural surfaces, recording precise 3D coordinates. These points are stored in point clouds [. . .]

Table 2: Representative examples from each evaluation category in the EO and Earth Sciences benchmark suite (see Table 1). In hallucination detection, underlined spans are the annotated hallucinated content.

full-parameter fine-tuning over LoRA (Hu et al., 2022), but insufficient for standalone continuous pretraining (CPT) (Gururangan et al., 2020; Zixuan et al., 2023). In preliminary experiments, the higher learning rates required for factual acquisition led to degradation of instruction-following behavior. To address this trade-off, we adopt a fine-tuning strategy that interleaves instruction fine-tuning (IFT) and long-form text, each mixing general-domain replay data with synthetic EO and Earth Sciences text. This enables domain adaptation while preserving general interactive capabilities. Both training and synthetic data generation are performed using an internal training framework.

6.1 Fine-tuning Data Synthesis

Our fine-tuning data consists of two components: long-form text and instruction-formatted text. Detailed distributions are provided in Table 3. Due to licensing conditions of source materials, we publicly release a curated subset of 10.7B tokens (20.9M input, 60.1M output, and 10.6B context tokens), of the full dataset used for training.

Long-form text. Our long-form fine-tuning data consists of long-form general-domain replay data alongside EO and Earth Sciences long-form text, which in turn consists of (i) a small portion of EO and Earth Sciences raw text from our corpus, and (ii) synthetically generated EO and Earth Sciences text. The former consists of either random samples from the corpus (Raw) or high-quality filtered samples (Best Chunks). The latter is generated with an Active Reading (Lin et al., 2025) pipeline, which reorganizes salient content to concentrate factual information and reinforce terminology, using either task-specific or predefined strategies. Strategy selection is performed by Mistral Medium 3.1, while Mistral Small 3.2 performs generation to maintain distributional alignment with the base model, as advised in Lin et al. (2025).

Instruction-formatted text. EO and Earth Sciences documents from our corpus (Section 4) are transformed into instruction–response pairs, including: (i) contextual and non-contextual Question Answering (QA) (ContextQA, SelfQA), (ii) long and multi-document QA (LongQA), (iii) multi-hop QA (Shen et al., 2025), (iv) self-referential alignment prompts (role, developer, and capability specification). We use various high-quality models for generation, including: Mistral Large 3, GPT-4o Mini, Mistral Medium 3.1, Qwen3-235B, DeepSeek-R1, DeepSeek v3.1, Qwen2.5-72B. This

is mixed with instruction-formatted replay data.

Quality control and filtering. We use LLM-based judges to assess domain relevance, factual quality and grounding. Long-form and Instruct text filtering uses either of the larger models used for generation as listed just above, with judge labels ("Best", "Good" or "Bad") determining retained samples. In total, we generate approximately 21B tokens of synthetic data, from which a filtered subset forms the synthetic pool used in the final training mixture (Table 3).

Long-form text: 30% (10B)		Instruction-formatted data: 70% (23.5B)	
Long-Form Replay	50%	Instruct-formatted Replay	60%
EO and Earth Sciences	**50%**	**EO and Earth Sciences**	**40%**
Raw	2%	ContextQA (Best)	12%
Best Chunks	14%	ContextQA (Good)	21%
Active Reading (Agnostic)	28%	SelfQA	2.6%
Active Reading (Specific)	6%	MultiHop QA	2.1%
		Long QA	2.6%

Table 3: Distribution of training data across long-form and instruction-formatted data.

6.2 Fine-tuning mixing long-form and instruction text

We fine-tune Mistral Small 3.2 (24B parameters, 128k context) interleaving instruction-formatted with long-form text within the same training runs. Increasing the proportion of EO and Earth Sciences data improves domain-specific benchmarks, but comes at the cost of reduced performance on general capabilities, particularly tool usage and structured reasoning. Replay data mitigates catastrophic forgetting and stabilizes interaction behavior. To address this trade-off, during fine-tuning, we vary (i) the ratio of long-form versus instruction-formatted text and (ii) the proportion of general domain replay versus domain-specific data within each type, so that the percentages presented in Table 3 are cumulative ratios taken over the entire fine-tuning process. Further, we use a learning rate schedule intermediate between typical IFT and CPT settings to balance factual integration and alignment stability. Finally, since runs trained with different data mixtures trade off domain and general performance differently, we merge checkpoints from ten runs using uniform parameter interpolation.

Our choice of replay-based training with checkpoint merging is motivated by stability and scalability in a production setting. We explored alternatives including LoRA (Hu et al., 2022) and regularization-based methods during development, but found that interleaved replay with checkpoint merging offered the best trade-off between domain acquisition and capability retention at our training scale. Recent work has proposed complementary strategies such as self-synthesized rehearsal (Huang et al., 2024) and selective parameter freezing (Hui et al., 2025), which may offer further improvements and constitute promising directions for future work.

6.3 Alignment

We apply Online Direct Preference Optimization (Qi et al., 2024) as a final alignment stage to refine formatting, stylistic consistency, and preference adherence. We follow the same alignment recipe and preference training data as in Liu et al. (2026). This final stage improves formatting consistency and adherence to preference signals, while preserving domain knowledge acquired during earlier training.

7 Evaluation

Setup. We evaluate `EVE-Instruct` on both domain-specific benchmarks (Section 5) and general-domain benchmarks to assess domain gains and preservation of general capabilities. We compare against the parent model and four additional LLMs of comparable scale (∼24B parameters).

For open-ended benchmarks, we adopt the LLM-as-a-judge framework (Wang et al., 2023) to evaluate answer correctness. Each candidate response is scored by an LLM judge conditioned on the question, reference answer, and, when applicable, retrieved context, using a 0–5 scale with predefined criteria (Appendix G). To improve robustness and mitigate individual model bias, we aggregate scores from a panel of judges[5] (Verga et al., 2024) and report the mean normalized score. Following Li et al. (2023b), we additionally conduct pairwise preference evaluation (Win Rate), where judges compare two candidate responses and select a winner or tie. The win rate of model A over model B is computed as the average preference across N evaluators: $\mathrm{WR}_A = \frac{1}{N}\sum_{i=1}^{N} \frac{\mathrm{wins}_{A_i} + 0.5 \cdot \mathrm{ties}_i}{\mathrm{wins}_{A_i} + \mathrm{ties}_i + \mathrm{losses}_{A_i}}$

Discussion. As shown in Table 4, `EVE-Instruct` achieves the highest performance across both MCQA benchmarks (single- and multiple-answer), indicating effective incorporation of domain-specific knowledge through fine-tuning. On the hallucination detection task, it

[5]Mistral Large 3, GPT-4.1 mini, DeepSeek-V3.2, and Qwen3-235B-A22B

Model	Size (B)	MCQA Mult. IoU	MCQA Mult. Acc.	MCQA Sing. Acc.	Hallucin. F1	Open-Ended Judge	Open-Ended EVE WR	Open-Ended w/ Context Judge	Open-Ended w/ Context EVE WR	Rank ↓
Llama4 Scout	109-A17	80.32	71.23	91.67	66.08	87.37	53.95	71.73	58.31	3.67
Qwen3	30-A3	78.40	66.36	93.02	81.30	94.92	50.70	**81.81**	52.12	2.67
Gemma3	27	73.60	57.54	87.31	75.07	94.41	50.92	78.31	51.58	3.83
Mistral Small 3.2	24	80.19	70.30	83.51	82.19	91.78	51.69	71.93	57.27	3.50
`EVE-Instruct`	24	**86.12**	**77.73**	**96.35**	**84.70**	**96.40**	—	78.28	—	**1.33**

Table 4: Model performance across EO and Earth Sciences benchmark tasks presented in Table 1 (0-shot). EVE WR (win rate) indicates percentage of pairwise comparisons where `EVE-Instruct` is preferred over the comparison model ($> 50\%$ means EVE is preferred). Rank ↓ (lower is better) reports the average per-metric rank across MCQA multiple (IoU and Accuracy), MCQA single (Accuracy), Hallucination (F1), Open-ended QA (Judge), and Open-Ended QA with Context (Judge).

attains the highest F1 score, reflecting improved discrimination between factual and non-factual responses. `EVE-Instruct` also leads competing models on open-ended QA without context under both the LLM-as-a-judge and Win Rate evaluations. When retrieval context is provided, Qwen3 achieves the highest LLM-as-a-judge score; however, `EVE-Instruct` remains competitive and obtains the strongest pairwise preference results, suggesting comparable overall response quality, despite smaller size.

To assess whether domain specialization impacts general capabilities, we report category-level averages across a suite of general-domain benchmarks in Table 5. Each category represents the mean score over multiple underlying benchmarks, whose full breakdown is provided in Appendix B. Tool Calling, Instruction Following, and Chat Quality correspond to internal evaluation suites from Mistral. Across all categories, `EVE-Instruct` maintains or improves performance with respect to its parent model, indicating that domain-specific adaptation does not degrade general-domain or chat capabilities. To address the potential overlap between model families used for synthetic data generation (Section 6.1) and the LLM-as-a-judge panel, we extend the evaluation with two independent judge families not involved in data generation: Claude Sonnet 4.6 and Gemini 2.5 Flash. As shown in Appendix Table 10, the resulting rankings are nearly identical (maximum shift: ± 0.25), confirming that the original panel does not exhibit systematic bias favoring `EVE-Instruct`.

8 Grounded Generation

We developed a RAG pipeline that grounds responses in relevant documents from our KBs (Section 3), reducing hallucinations and extending knowledge beyond the training data.

Category	Small 3.2	`EVE-Instruct`	Δ
Math & Reasoning	50.8	**54.9**	+4.1
Coding	55.6	**56.5**	+0.9
Knowledge	67.7	**69.0**	+1.3
Tool Calling	87.9	**90.9**	+3.0
Instruction Following	80.1	**81.2**	+1.1
Chat Quality	90.8	**91.7**	+0.9
Overall	72.2	**74.0**	+1.8

Table 5: General-domain performance after domain adaptation (category-level averages over several standard benchmarks, 0–100 scale).

Documents are chunked into ∼512-word segments via a two-pass strategy (first by document sections, then by paragraphs or sentences) that preserves LaTeX and Markdown formulas and tables. Uninformative chunks are filtered using statistical heuristics, then enriched with metadata and embedded using `Qwen3-Embedding-4B` (Zhang et al., 2025). Embeddings are stored through binary quantization in Qdrant.[6]

For chunk retrieval, we first apply a query rewriting step using `EVE-Instruct` by incorporating conversational context, disambiguating, and optimizing retrieval. For each KB, the top $2K$ chunks are retrieved via embedding similarity with optional metadata filtering. The candidates are then re-ranked using `Qwen3-Reranker-4B` (Zhang et al., 2025), and the top K documents are selected.

Hallucination Detection. To address the issue of factual hallucinations while keeping average answer latency low, we implement a pipeline which starts with hallucination detection and, based on the outcome, optionally proceeds to answer revision. In the first stage, `EVE-Instruct` performs fact-checking, acting as an evaluator, and produces

[6] qdrant.tech/documentation/guides/quantization/

a binary hallucination label as well as a justification. If flagged for hallucination, the query is reformulated using the justification to address identified issues using newly retrieved documents. With the retrieved documents, the model generates a revised response, encouraging more conservative and fact-grounded answers. Then, inspired by Ji et al. (2023), the model critiques the original response using both prior and newly retrieved evidence to produce a revised answer. Finally, the model ranks the original and revised outputs based on factuality and supporting evidence, selecting the most reliable response.

9 Deployment

EVE was deployed as a production system supporting 350 users during a six-month pilot from September 2025. The architecture consists of: (i) a single-node Qdrant vector database storing 4.2M dense embeddings with binary quantization; (ii) `EVE-Instruct`, hosted on RunPod serverless infrastructure with dynamic scaling (1–30 workers) across NVIDIA A100/H100 GPUs; (iii) an Amazon DocumentDB cluster for user management, chat history, and application metadata; (iv) an AWS EC2 backend;[7] and (v) an AWS CloudFront CDN-managed frontend. A detailed description of the end-to-end system is provided in Appendix D.

10 Conclusion and Future Work

In this paper, we introduced **Earth Virtual Expert (EVE)**, an open and modular end-to-end system for building, evaluating, and deploying domain-specialized LLMs for EO and Earth Sciences. EVE combines (i) large-scale curation and processing, (ii) domain-adapted `EVE-Instruct` built on Mistral Small 3.2 24B, (iii) domain-specific evaluation benchmarks, and (iv) retrieval-augmented and hallucination-aware grounded generation in a production deployment. Across our domain benchmarks, `EVE-Instruct` improves over other strong models in its parameter range on multiple-choice QA, hallucination detection, and open-ended instruction-following domain specific benchmarks, while remaining competitive on general capabilities. Beyond offline evaluation, the system has been deployed in a 6-month pilot serving 350 users via GUI and API, demonstrating that domain-specific, grounded scientific assistants can be delivered with practical latency and cost constraints. We release models, code, curated corpus, manually-created domain benchmarks, and a substantial portion of the synthetically-generated fine-tuning dataset used to create `EVE-Instruct`.

As EO and Earth Sciences advance toward Earth Action (ESA, 2024), there is increasing need to integrate textual knowledge with spaceborne observations, in-situ data, and Earth system models. Recent advances in Geospatial Foundation Models, Vision–Language Models, and spatial embeddings enable joint text–data representations that support reasoning and decision-making (Longépé et al., 2025). Building on this, we aim to extend EVE beyond text into a multimodal, agentic platform capable of reasoning over imagery and geospatial data, and supporting multi-step scientific workflows for large-scale EO and Earth Sciences analysis and data-driven inference.

Limitations

We highlight key limitations: (i) Licensing prevents redistribution of 1.1B Wiley tokens (∼21% of the corpus); we release the open-access subset, pipelines, and synthetic data, so exact reproduction requires independent access to licensed or non-redistributable content. (ii) Evaluation coverage remains limited in task diversity and scale, including human evaluation. (iii) Grounded generation depends on retrieval coverage and data freshness. (iv) The current system is text-only and does not reason directly over EO and Earth Sciences imagery or structured geospatial data.

CO_2 footprint

We estimate the carbon footprint of synthetic data generation, fine-tuning, and evaluation at about 38 tonnes of CO_2 equivalent, based on GPU energy consumption and regional carbon intensity factors.

Acknowledgments

This work was supported by ESA Φ-lab under the Foresight Element of the FutureEO programme. We thank Imperative Space for their domain expertise, as well as Translated and Sapien for data annotation. We also thank Hiroshi Araki, Matteo Cacciola, Kumar Tulsi, and Eva Gmelich Meijling for their technical support during the development of EVE. We thank Andreas Vlachos for his guidance on hallucination detection.

[7] Instance type t3.large (2 vCPUs, 8 GB RAM, 320 GB).

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Adrien Barbaresi. 2021. Trafilatura: A web scraping library and command-line tool for text discovery and extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 122–131, Online. Association for Computational Linguistics.

Bishwaranjan Bhattacharjee, Aashka Trivedi, Masayasu Muraoka, Muthukuma ran Ramasubramanian, Takuma Udagawa, Iksha Gurung, Nishan Pantha, Rong Zhang, Bharath Dandala, Rahul Ramachandran, and 1 others. 2024. Indus: Effective and efficient language models for scientific applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 98–112.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*.

Yuxing Chen, Weijie Wang, Sylvain Lobry, and Camille Kurtz. 2024. An llm agent for automatic geospatial data analysis. *Preprint*, arXiv:2410.18792.

Riccardo Corrente, Marcello Politi, Vijayasri Iyer, Sandesh Katakam, Sébastien Bratières, and Tomas Navarro. 2026. SatcomLLM: First domain adaptation of LLMs to satellite communications. `https://huggingface.co/esa-sceva`. Hugging Face organization page, ESA SCEVA project. Accessed: 2026-04-25.

Tijmen de Haan. 2025. cosmosage: A natural-language assistant for cosmology. *Astronomy and Computing*, 51:100934.

Tijmen de Haan, Yuan-Sen Ting, Tirthankar Ghosal, Tuan Dung Nguyen, Alberto Accomazzi, Azton Wells, Nesar Ramachandra, Rui Pan, and Zechang Sun. 2024. Astromlab 3: achieving gpt-4o level performance in astronomy with a specialized 8b-parameter large language model. *arXiv preprint arXiv:2411.09012*.

Cheng Deng, Tianhang Zhang, Zhongmou He, Yi Xu, Qiyuan Chen, Yuanyuan Shi, Luoyi Fu, Weinan Zhang, Xinbing Wang, Chenghu Zhou, Zhouhan Lin, and Junxian He. 2024. K2: A foundation language model for geoscience knowledge understanding and utilization. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 161–170. ACM.

ESA. 2024. Earth observation science strategy, earth science in action for tomorrow's world. 42pp.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2023. Rarr: Researching and revising what language models say, using language models. In *Proceedings of ACL 2023*, pages 16477–16493. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8342–8360.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Xuechun Hu. 2026. Responsible open-source AI: From principles to practice.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1416–1428.

Tingfeng Hui, Zhenyu Zhang, Shuohuan Wang, Weiran Xu, Yu Sun, and Hua Wu. 2025. Hft: Half fine-tuning for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12791–12819.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.

Reto Knutti. 2019. Closing the knowledge-action gap in climate change. *One Earth*, 1(1):21–23.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Junyi Li, Xiaoxuan Zhang, and 1 others. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of EMNLP 2023*. Association for Computational Linguistics.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024. Urbangpt: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2024)*, pages 5351–5362.

Jessy Lin, Vincent-Pierre Berges, Xilun Chen, Wen-Tau Yih, Gargi Ghosh, and Barlas Oğuz. 2025. Learning facts at scale with active reading. *arXiv preprint arXiv:2508.09494*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of ACL 2022*, pages 3214–3252. Association for Computational Linguistics.

Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, and 1 others. 2026. Ministral 3. *arXiv preprint arXiv:2601.08584*.

Nicolas Longépé, Hamed Alemohammad, Anca Anghelea, Thomas Brunschwiler, Gustau Camps-Valls, Gabriele Cavallaro, Jocelyn Chanussot, Jose Manuel Delgado, Begüm Demir, Nikolaos Dionelis, Paolo Fraccaro, Anna Jungbluth, Robert E. Kennedy, Valerio Marsocci, Muthukumaran Ramasubramanian, Raul Ramos-Pollan, Sujit Roy, Gencer Sümbül, Devis Tuia, and 2 others. 2025. Earth action in transition: Highlights from the 2025 esa–nasa international workshop on ai foundation models for eo [space-agencies]. *IEEE Geoscience and Remote Sensing Magazine*, 13(4):457–462.

Ali Maatouk, Fadhel Ayed, Nicola Piovesan, Antonio De Domenico, Merouane Debbah, and Zhi-Quan Luo. 2025. Teleqna: A benchmark dataset to assess large language models telecommunications knowledge. *IEEE Network*.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of EMNLP 2023*, pages 9004–9017. Association for Computational Linguistics.

Ali Mansourian and Rachid Oucheikh. 2024. Chatgeoai: Enabling geospatial analysis for public through natural language, with large language models. *ISPRS International Journal of Geo-Information*, 13(10):348.

Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David Lobell, and Stefano Ermon. 2023. Geollm: Extracting geospatial knowledge from large language models. *arXiv preprint arXiv:2310.06213*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037.

Tuan Dung Nguyen, Yuan-Sen Ting, Ioana Ciuca, Charles O'Neill, Ze-Chang Sun, Maja Jabłońska, Sandor Kruk, Ernest Perkowski, Jack Miller, Jason Jason Jingsh Li, and 1 others. 2023. Astrollama: Towards specialized foundation models in astronomy. In *Proceedings of the Second Workshop on Information Extraction from Scientific Publications*, pages 49–55.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2024)*, pages 30811–30849.

Biqing Qi, Pengfei Li, Fangyuan Li, Junqi Gao, Kaiyan Zhang, and Bowen Zhou. 2024. Online dpo: Online direct preference optimization with fast-slow chasing. *arXiv preprint arXiv:2406.05534*.

Alessandro Scirè, Andrei Stefan Bejgu, Simone Tedeschi, Karim Ghonim, Federico Martelli, and Roberto Navigli. 2025. Truth or mirage? towards end-to-end factuality evaluation with llm-oasis. *Computational Linguistics*, pages 1–41.

Zhiyu Shen, Jiyuan Liu, Yunhe Pang, and Yanghui Rao. 2025. Hopweaver: Synthesizing authentic multi-hop questions across text corpora. *arXiv preprint arXiv:2505.15087*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, and 1 others. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: A large-scale dataset for fact extraction and verification. In *Proceedings of NAACL-HLT 2018*, pages 809–819. Association for Computational Linguistics.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Cunxiang Wang, Sirui Cheng, Qipeng Guo, Yuanhao Yue, Bowen Ding, Zhikun Xu, Yidong Wang, Xiangkun Hu, Zheng Zhang, and Yue Zhang. 2023. Evaluating open-qa evaluation. *Advances in Neural Information Processing Systems*, 36:77013–77042.

Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, and 1 others. 2024. Redpajama: An open dataset for training large language models. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2024)*, pages 116462–116492.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Yifan Zhang, Xinyi Wang, Zekun Li, and 1 others. 2024a. Geogpt: Understanding and processing geospatial tasks through an llm-based framework. *arXiv preprint arXiv:2307.07930*.

Yifan Zhang, Zhiyun Wang, Zhengting He, Jingxuan Li, Gengchen Mai, Jianfeng Lin, Cheng Wei, and Wenhao Yu. 2024b. Bb-geogpt: A framework for learning a large language model for geographic information science. *Information Processing & Management*, 61(5):103808.

Ke Zixuan, Shao Yijia, Lin Haowei, Konishi Tatsuya, Kim Gyuhak, Liu Bing, and 1 others. 2023. Continual pre-training of language models. In *Proceedings of The Eleventh International Conference on Learning Representations (ICLR-2023)*.

A Corpus Creation

We provide additional technical details, statistics, and validation results for the data collection and processing pipeline to build the corpus presented in Section 4. We assemble the corpus to cover the breadth of EO and Earth Sciences knowledge, including subtopics such as satellite imagery analysis, climate modeling, geospatial data processing, and environmental monitoring. The majority of sources are peer-reviewed domain-specific publishers (e.g., MDPI, NCBI), as well as reputable sources (e.g., ESA, NASA), and mainstream sources (Wikipedia, arXiv). We present a full corpus distribution of the released open source data in Table 6.

Data Scraping. We use Selenium Webdriver[8] for automated browser navigation, paired with Brightdata Web Unlocker.[9] This allows us to manage request rates, rotate IP addresses, solve captchas and maintain compliance with each websites.

Data Cleaning. We implemented a multi-stage data cleaning pipeline to improve corpus quality and remove extraction artifacts:

[8] selenium.dev/documentation/webdriver/

[9] brightdata.com/

Source	Released Tokens	%	Licence
MDPI	1.3 B	46.6	CC-BY
Copernicus	723 M	25.9	CC-BY
NCBI	485 M	17.4	CC-BY
ISPRS	74.1 M	2.7	CC-BY
Wiley	71.6 M	2.6	CC-BY
Elsevier	43.7 M	1.6	CC-BY
Cambridge Press	40.3 M	1.4	CC-BY
Springer	25.4 M	0.9	CC-BY
Taylor & Francis	11.8 M	0.4	CC-BY
AMS	5.7 M	0.2	CC-BY
SAGE	3.8 M	0.1	CC-BY
NASA	2.5 M	0.09	CC-BY
arXiv	2.1 M	0.08	CC-BY
IEEE	992 k	0.04	CC-BY
EGUP	671 k	0.02	CC-BY
Oxford Academic	576 k	0.02	CC-BY
IOP Science	507 k	0.02	CC-BY
Frontiers	307 k	0.01	CC-BY
EOGE	245 k	0.01	CC-BY
EOS	96 k	0.003	CC-BY
MIT	83 k	0.003	CC-BY
UK Met Office	7 k	0.0002	CC-BY
Total	**2.8 B**	**100**	

Table 6: Token distribution across data sources in the released subset of the open-access EO and Earth Sciences corpus (2.8B tokens released out of 4.2B total open-access tokens).

1. Nougat Artifact Removal (Blecher et al., 2023): we remove residual tags and artifacts introduced during PDF parsing (e.g., `<WARNING>`, `<ERROR>`).
2. Merged Word Correction: we detect and correct tokenization errors where numeric prefixes are concatenated with words (e.g., `1Introduction` → `1 Introduction`).
3. OCR Duplication Removal: we apply MinHash-based near-duplicate detection to identify repeated text segments. We further detect and remove OCR-induced duplicates via adjacency patterns (i.e., repeated spans with minimal or no intervening characters).
4. Rule-Based Filtering: we remove low-information lines (e.g., sequences of repeated symbols) and normalize formatting by collapsing excessive whitespace (e.g., replacing three or more consecutive newlines with two).

Data Extraction. To select an OCR system for scientific document extraction, we first construct a benchmark of 1k PDFs and evaluate multiple OCR tools. Ground-truth annotations are generated using a high-fidelity pipeline combining image encoding and GPT-4–based transcription. We measure OCR quality using the Normalized Levenshtein Similarity (NLS) (Levenshtein, 1966). Let

$\hat{y}$ denote the predicted text and y the ground-truth text. Let $\mathrm{LD}(\hat{y}, y)$ denote their Levenshtein distance, and $\mathrm{len}(\cdot)$ the sequence length. The metric is defined as:

$$\mathrm{NLS}(\hat{y}, y) = 1 - \frac{\mathrm{LD}(\hat{y}, y)}{\max\big(\mathrm{len}(\hat{y}), \mathrm{len}(y)\big)}. \quad (1)$$

The NLS score ranges from 0 (no similarity) to 1 (exact match), quantifying agreement between OCR output and reference text. As shown in Table 7, several tools achieve high text-level similarity, but only Nougat consistently captures structured scientific content, including formulas and tables. This balance between textual fidelity and structural preservation, together with competitive latency (0.01s per page), motivates our selection for downstream processing.

Tool	Text	Formulas	Tables	Avg.
Markitdown	0.81	0.00	0.00	0.27
Docling	0.79	0.00	0.40	0.40
Pymupdf4llm	0.81	0.00	0.26	0.36
Pypdf2	**0.84**	0.00	0.00	0.28
Marker	0.80	0.31	**0.42**	0.51
Unstructured	0.79	0.00	0.00	0.26
Pdfminer	0.81	0.00	0.00	0.27
Nougat	0.74	**0.55**	0.41	**0.57**

Table 7: OCR benchmark results (Normalized Levenshtein Similarity) across tools, evaluating text, formula, and table extraction. The Avg. column reports the mean across the three categories.

B Evaluation

In addition to the category-level averages reported in Table 5 on general-domain benchmarks, we provide the full set of underlying benchmark results in Table 8. These detailed results show that `EVE-Instruct` retains broad general capabilities after domain adaptation.

Additionally, we report results on the EO benchmarks in Table 1, comparing `EVE-Instruct` with larger-scale models to complement the comparable-size comparisons in Table 4. As shown in Table 9, `EVE-Instruct` remains competitive even against substantially larger models, indicating strong efficiency in domain-specific performance.

Finally, we show that our domain adaptation and replay fine-tuning yield positive transfer to other technical domains, even without domain-specific training. We evaluate this in Table 11, in the telecommunications and satellite communications domain using both a multiple-choice QA benchmark, TelQNA (Maatouk et al., 2025), and an open-ended QA dataset, Satcom Open-Ended (Corrente et al., 2026).

Benchmark	Small 3.2	EVE	Δ
aime25@16	26.7	**35.2**	+8.5
aime24@16	37.1	**41.2**	+4.1
math	**88.6**	88.4	-0.2
Average (Math & Reasoning)	*50.8*	***54.9***	*+4.1*
livecodebench_p@1	36.4	**39.1**	+2.7
mbpp_p@1	**74.7**	73.9	-0.8
Average (Coding)	*55.6*	***56.5***	*+0.9*
super_gpqa_5shot	38.8	**40.5**	+1.7
mmlu_redux_5shot	82.1	**82.7**	+0.6
mmlu_5shot	80.6	**81.7**	+1.1
mmlu_astronomy_5shot	92.1	**97.4**	+5.3
naturalqs_5shot	**33.9**	33.5	-0.4
triviaqa_5shot	**78.8**	78.1	-0.7
Average (Knowledge)	*67.7*	***69.0***	*+1.3*
Average (Tool Calling)	*87.9*	***90.9***	*+3.0*
Average (Instruction Following)	*80.1*	***81.2***	*+1.1*
Average (Chat Quality)	*90.8*	***91.7***	*+0.9*
Overall	*72.2*	***74.0***	*+1.8*

Table 8: Evaluation results (0–100 scale) comparing Mistral Small 3.2 and `EVE-Instruct` across general-domain benchmarks. Category averages are shown for each task group. Tool Calling, Instruction Following, and Chat Quality correspond to private internal evaluation.

C RAG Evaluation

We detail the evaluation and design choices underlying the RAG pipeline introduced in Section 8. While the main text describes the system, we provide here a systematic analysis of chunking, embedding, and reranking to justify the final configuration.

We evaluate the impact of key design dimensions, including chunking strategy, embedding model, reranker, chunk size (512 vs. 1024), and quantization. In particular, we compare the hierarchical two-pass chunker (Section 8) against a standard fixed-length character chunker, and assess both quantized and non-quantized variants. For embedding and reranking, we focus on two representative models: Qwen3-Embedding-4B (Zhang et al., 2025), a top-performing and efficient model on the MTEB leaderboard (Muennighoff et al., 2023)[10], and INDUS (Bhattacharjee et al., 2024), an encoder-only model trained specifically for scientific domains.

[10] huggingface.co/spaces/mteb/leaderboard

		MCQA Mult.		MCQA Sing.	Hallucin.	Open-Ended		Open-Ended w/ Context		
Model	Size (B)	IoU	Acc.	Acc.	F1	Judge	EVE WR	Judge	EVE WR	Rank ↓
GPT-4.1	1800*	**87.56**	78.19	94.37	81.58	96.48	49.77	**86.65**	49.22	2.83
Qwen3	235-A22	87.40	**80.97**	95.16	84.40	**97.05**	48.24	86.10	50.09	**2.17**
MiniMax m2.5	230A10	84.82	77.72	94.95	83.77	91.00	51.10	81.57	51.20	5.17
GPT OSS	120A5	84.56	76.79	89.77	89.92	94.20	50.30	84.80	50.70	4.83
Mistral Medium 3.1	200*	85.44	76.33	95.00	76.89	96.45	50.20	86.44	50.99	4.17
GPT-5 nano	20*	84.40	76.10	91.99	**90.94**	92.20	50.20	84.40	48.60	5.33
EVE-Instruct	24	86.12	77.73	**96.35**	84.70	96.40	—	78.28	—	3.50

Table 9: Extension of Table 4 to larger-scale models under the same evaluation setup. Rank ↓ (lower is better) reports the average per-metric rank across MCQA multiple (IoU and Accuracy), MCQA single (Accuracy), Hallucination (F1), Open-ended QA (Judge), and Open-Ended QA with Context (Judge). *Model size is reported when publicly available; otherwise estimated internally.

		Open-Ended				Open-Ended w/ Context						
		Claude Judge		Gemini Judge		Claude Judge		Gemini Judge				
Model	Size (B)	Score	EVE WR	Score	EVE WR	Score	EVE WR	Score	EVE WR	$\text{Rank}_{\text{prev}}$ ↓	Rank ↓	ΔRank ↓
Llama Scout 4	109-A17	69.22	52.7	78.09	54.9	55.64	59.83	62.44	59.45	3.67	3.50	−0.17
Qwen3	30-A3	82.75	51.2	88.89	51.2	69.13	53.71	64.59	53.23	2.67	2.75	+0.08
Gemma 3	27	80.82	51.6	87.55	51.4	62.48	52.75	63.49	51.91	3.83	3.88	+0.05
Mistral 3.2 Small	24	73.76	51.0	84.04	52.5	57.21	57.19	60.28	56.69	3.50	3.75	+0.25
EVE-Instruct	24	83.19	—	90.19	—	61.77	—	72.44	—	**1.33**	**1.13**	−0.20

Table 10: Comparison of models on Open-Ended and Open-Ended with Context tasks using two judges (Claude and Gemini - anthropic/claude-sonnet-4.6 and google/gemini-2.5-flash). $\text{Rank}_{\text{prev}}$ refers to the single-judge baseline ranking from Table 4. Rank ↓ (lower is better) is the average per-metric rank across the four judge score columns. ΔRank denotes the difference between the two rankings.

		TelQNA	Satcom Open-Ended
Model	Size (B)	Acc.	Judge
Llama4 Scout	109-A17	74.57	75.00
Qwen3	30-A3	73.81	**83.00**
Gemma3	27	70.84	81.25
Mistral Small 3.2	24	72.63	76.50
EVE-Instruct	24	**75.72**	81.25

Table 11: Model performance on TelQNA and Satcom Open-Ended benchmarks (0-shot).

C.1 Token-level Evaluation Framework

Conventional information retrieval metrics emphasize document ranking order, yet large language models demonstrate relative insensitivity to where relevant content appears within their context window. Furthermore, when query-relevant information spans multiple chunks, inter-chunk ranking becomes inherently ambiguous. Drawing from Chroma's framework,[11] we implement a token-granularity evaluation protocol for our retrieval pipeline. We construct a semi-synthetic evaluation set by prompting an LLM to generate queries grounded in the corpus, along with corresponding relevant text excerpts. This approach avoids contamination of embedding models and enables domain-specific evaluation. Each query–excerpt pair is evaluated using the following metrics:

- **Intersection over Union (IoU)**: Measures overlap between retrieved and ground-truth tokens. Penalizes redundancy when the same relevant tokens appear across multiple chunks.
- **Precision**: Token-level signal-to-noise ratio of retrieved tokens that are relevant. Reflects how much irrelevant context is introduced.
- **Recall**: Measures retrieval completeness by calculating the fraction of ground-truth relevant tokens successfully retrieved. Indicates whether the system captures all necessary information to answer the query.
- **Document Recall**: Percentage of documents containing at least one relevant chunk.
- **Passage Recall**: Fraction of retrieved chunks that contain at least one relevant token.

Together, these metrics capture both retrieval quality and efficiency: token-level metrics (IoU, precision, recall) assess fine-grained relevance, while document and passage recall provide complementary coarse-grained coverage signals.

[11] research.trychroma.com/evaluating-chunking

C.2 Discussion

Embedder. Table 12 compares embedding models across chunking strategies, sizes, and quantization. Quantization has negligible impact on retrieval quality, while providing clear gains in memory and inference efficiency.[12] Qwen3-Embedding-4B consistently outperforms INDUS embedder across all configurations, particularly in recall. Increasing chunk size to 1024 improves document and passage recall but degrades IoU and precision due to additional irrelevant tokens. Similarly, the Recursive chunker achieves higher recall but at the cost of substantially lower IoU, indicating increased redundancy. We therefore select the Two-pass chunker with Qwen3-Embedding-4B. We fix a chunk size of 512, since qualitative evaluation by users in the chat interface consistently favored shorter, more focused chunks.

Reranker. Table 13 reports reranking results at top-10 with $K{=}20$ retrieved candidates. Qwen3-4B-reranker consistently improves over retrieval-only baselines across all embedding configurations. In contrast, INDUS reranker can degrade performance when paired with stronger embeddings, suggesting weaker calibration in high-quality retrieval settings. The best overall results are obtained with chunk size 1024 and the Qwen3-4B reranker. However, we retain chunk size 512 in the final system: qualitative evaluation favours shorter, more focused chunks, and the performance gap after reranking is relatively small.

D System Architecture

EVE is deployed as a full-stack application comprising a React frontend, FastAPI backend, and a conversation management layer.

D.1 Conversation Management

Memory management. To maintain conversational continuity, we use a rolling summarization strategy rather than retaining the full dialogue. At turn t the model receives the previous turn $t-1$ in full, along with a compressed summary $S_{t_0}^{t-2}$ of all earlier turns. The most recent turn is always preserved verbatim to support immediate follow-up questions. Given the substantial length of each turn comprising the query, the generated answer, and the retrieved context a new summary is produced at every step by prompting `EVE-Instruct` to condense the previous summary and the latest turn: $S_{t_0}^{t-1} = \textit{summarize}(S_{t_0}^{t-2},\, t-1)$.

Context Management. To balance the different components of the prompt, we enforce a fixed token budget:

- **User query:** capped at 30K tokens and truncated if exceeded.
- **Retrieved context:** limited to 7K tokens, with low-similarity chunks dropped until the limit is met.
- **Conversation summary:** constrained to 5K tokens, enforced during summary generation.
- **Model response:** allocated 15K tokens.
- **Previous turn:** allocated 57K tokens.

Figure 2 illustrates the end-to-end architecture of the deployed EVE system, including query processing, hybrid retrieval, re-ranking, grounded generation, and conversational state management.

D.2 Backend

The EVE backend is a FASTAPI[13] service that handles data access, conversation state and document management. It is paired with Amazon DocumentDB for storage.

The state of each user is persisted with credentials, individual conversations and messages with timestamps, and the documents and collections used during retrieval. Authentication uses JWT tokens for protected routes along with CORS restriction. Every action performed on the interface is logged in a MongoDB instance. We have a dedicated internal dashboard that monitors user usage trends, feedback, types of queries and documents used, document level click rate and performance metrics. We use uvicorn[14] web server with multiple worker processes to handle concurrent requests. We also make use of lifespan hooks for database connections. The service is containerized using Docker.

D.3 Frontend

The EVE frontend is a single-page React application built with TypeScript and Vite. The chat interface streams model responses as they are generated using Server-Sent Events,[15] so users see answers appear token by token. Long conversations use list

[12] We compute latency difference on a various kinds of retrievals on a subset, and observe between 66.6% and 99.2% reduction in latency in different setups when quantizing.

[13] github.com/fastapi

[14] github.com/Kludex/uvicorn

[15] developer.mozilla.org/en-US/docs/Web/API/Server-sent_events

Chunker	Chunk Size	Embedding	Doc recall	Passage recall	IoU	Precision	Recall
Two-pass	512	INDUS 512	91.60	57.70	2.77	2.78	77.00
Two-pass	512	INDUS 512*	91.60	50.80	2.64	2.65	73.20
Two-pass	512	INDUS 1024	83.24	38.60	2.34	2.35	59.90
Two-pass	512	INDUS 1024*	82.60	38.50	2.32	2.33	59.70
Two-pass	512	Qwen3-Embedding-4B	95.70	65.60	2.93	**2.94**	85.30
Two-pass	512	Qwen3-Embedding-4B*	95.60	63.60	**2.94**	**2.94**	85.00
Two-pass	1024	INDUS 512*	90.10	53.50	1.77	1.78	73.50
Two-pass	1024	INDUS 1024*	80.00	40.70	1.69	1.69	60.30
Two-pass	1024	Qwen3-Embedding-4B*	**96.40**	68.80	2.23	2.23	87.60
Recursive	1024	INDUS 512*	90.00	61.80	1.22	1.22	79.40
Recursive	1024	INDUS 1024*	78.50	44.60	1.04	1.05	63.70
Recursive	1024	Qwen3-Embedding-4B*	95.43	**70.70**	1.38	1.38	**88.10**

Table 12: Performance comparison of different embedding models across chunking strategies and chunk sizes. All metrics are expressed as percentages with two decimal precision. Precision and recall are per token, considering 10 as the number of passages retrieved. * indicates quantized embeddings. The Two-pass chunker refers to the approach presented in Section 8. The Recursive chunker is based on LangChain's `RecursiveCharacterTextSplitter`.

Collection	Chunk Size	Reranker	Ref Retrieved Ratio @10		MRR @10	
			Retrieval	Reranked	Retrieval	Reranked
INDUS 512*	512	INDUS	51.00	54.10	45.90	55.70
INDUS 512*	512	Qwen3-4B	51.00	54.70	45.90	61.70
Qwen3-Embedding-4B*	512	INDUS	63.30	62.80	62.60	60.10
Qwen3-Embedding-4B*	512	Qwen3-4B	63.30	65.10	62.60	69.30
Qwen3-Embedding-4B*	1024	Qwen3-4B	**68.60**	**71.60**	**65.40**	**73.40**
Qwen3-Embedding-4B*	1024	INDUS	**68.60**	65.90	**65.40**	61.00

Table 13: Performance comparison at top-10 with K=20 retrievals. **Ref Retrieved Ratio @10** measures the percentage of queries for which at least one relevant chunk appears in the top-10 reranked results. **MRR @10** (Mean Reciprocal Rank) is the average of the reciprocal rank of the first relevant chunk across queries, rewarding systems that place relevant chunks higher in the ranked list. All metrics are expressed as percentages with two decimal precision. * indicates quantized embeddings.

virtualization[16] to stay fast even with many messages. A side panel shows the retrieved document chunks (with basic metadata) and lets users pin or remove sources. A settings view exposes key RAG and generation controls such as model choice, temperature, retrieval depth, and safety/tuning options.

The frontend manages server data with React Query[17] and local UI state with React. Forms use React Hook Form[18] and Zod[19] for client-side validation. When a user sends a message, the client triggers retrieval and generation, then renders partial tokens as they arrive over SSE. Conversation metadata, settings presets, and recent sources are cached and refreshed on a schedule to balance freshness and responsiveness.

Errors from the backend are mapped to user-friendly messages for common issues such as rate limits, context size limits, or empty retrieval results. For transient failures, the UI supports retries with backoff[20] and allows users to cancel an in-progress stream.

UI performance is improved with list virtualization, memoized content blocks, and deferring work for off-screen panels. Vite provides code splitting and tree-shaking to keep the bundle small. The production build is deployed via a GitHub Actions CI/CD pipeline to Amazon S3 and served through CloudFront with compression and aggressive caching for static assets. Runtime configuration is supplied via Vite environment variables, with secrets managed outside of version control.

E Pilot Programme

The EVE platform underwent a structured pilot evaluation to assess its readiness as a domain-specific research assistant for the EO and Earth science community. This section describes the pilot setup and discusses the key findings that emerged from user engagement data and qualitative feedback.

The pilot programme enrolled 350 participants

[16] tanstack.com/virtual
[17] tanstack.com/query
[18] react-hook-form.com
[19] zod.dev
[20] docs.aws.amazon.com/general/latest/gr/api-retries.html

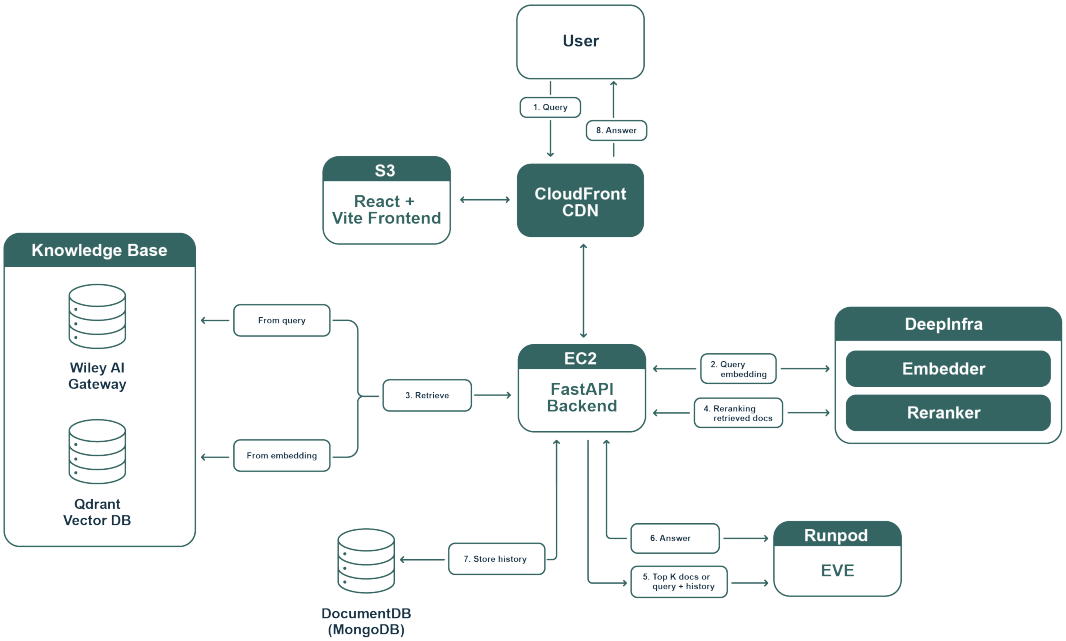

Figure 2: End-to-end architecture of the deployed EVE system.

drawn from ESA technical staff, EO researchers, and affiliated stakeholders. Data collection combined multiple complementary methods: super-user interviews, one-to-one meetings with targeted stakeholders, structured questionnaires distributed to all participants, and direct usage telemetry from the EVE platform. The evaluation period was designed as an exploratory phase, encouraging users to test EVE across a broad range of EO queries rather than integrate it into daily workflows.

Participants spanned academia and research (37.7%), industry and commercial (29.1%), space and EO agencies (11.2%), government (5.8%), and students or unaffiliated users (13.5%), with 78% coming from research, industry, or agency backgrounds. In terms of EO expertise, 61.2% identified as expert or very familiar, 21.3% as moderately familiar, and 17.0% as having low familiarity with the domain.

Over the pilot period (September 2025 - March 2026), users generated 2,622 messages across 1,424 conversations, totalling approximately 6.8M input and 2.46M output tokens, with 40 documents uploaded. The RAG pipeline was active in 86.7% of interactions. Monthly message volume peaked in September (943) and gradually declined through March (68), consistent with an exploratory evaluation. Knowledge base usage was distributed across the Wiley AI Gateway (42.7%), EVE open-access corpus (40.3%), ESA KB (7.3%), Wikipedia EO (5.3%), and user-uploaded documents (4.4%). Document retrieval followed a heavy-tailed distribution: 60.7% of documents were retrieved only once, while the most frequently accessed document was retrieved 380 times. Mean latency was 15.62s for answer generation and 12.75s time to first token, with retrieval at 1.44s, re-ranking at 2.02s, and query embedding at 0.94s. System reliability, measured over a final-phase subset of 245 messages, showed 188/245 successful LLM calls, 211/245 successful retrievals, and 220/245 successful re-ranking operations.

Overall, the results position EVE as a high-potential domain-specific research assistant with clear relevance for EO applications, but with limited operational maturity at this stage. While the concept of domain-adapted LLMs for ESA scientific and engineering workflows was positively received, the pilot highlights key limitations in knowledge coverage, factual reliability, and human–AI interaction design. Addressing these aspects will be critical for enabling sustained deployment in professional settings.

F Compliance

Beyond model architecture, EVE is designed as an open, European-aligned system, with efficiency and regulatory compliance treated as core design constraints. The compact architecture enables efficient deployment, and all components (model weights, data pipelines, and infrastructure) are released under open licenses where legally permissible. In parallel, we conducted a structured compliance and governance analysis covering data sourcing, copyright, licensing, and responsible deployment. A detailed account is provided in a dedicated whitepaper, which presents an applied framework for developing and open-sourcing AI systems under regulatory constraints, including data provenance, anonymization, and retrieval-augmented architectures. We refer the reader to the full document for further details[21] (Hu, 2026).

G Prompts

For reproducibility, we provide the prompt templates used across evaluation and data generation pipelines. These prompts cover (i) evaluation under the LLM-as-a-judge framework (Section 7) and (ii) synthetic data generation and filtering procedures (Section 6.1). Minor variations of these templates are used across settings (e.g., with or without retrieval context, or for pairwise evaluation).

G.1 LLM-as-a-Judge Evaluation Prompt

For open-ended evaluation with retrieval context, we adopt an LLM-as-a-judge framework in which a judge model scores candidate responses conditioned on the question, reference answer, and retrieved context. The prompt used for this setting is shown in Figure 3. Variants of this template are used for pairwise Win Rate evaluation and for no-context evaluation, where retrieval passages are omitted.

G.2 Active Reading Chunk Filter

Prior to the Active Reading synthesis pipeline (Section 6.1), corpus chunks are filtered by an LLM judge, in Figure 4. The judge assigns one of three ratings: *Best* (high quality and highly relevant to EO), *Good* (relevant but mediocre, or high quality but little related), or *Bad* (poor quality or off-topic). Only *Best*-rated chunks are passed to Active Reading; *Good* chunks may appear in the raw long-form mixture.

[21] zenodo.org/records/18415713

LLM-as-a-judge

```
You are evaluating answers about Earth Observation (EO)
and Remote Sensing (RS).

**Domain Terminology Check**

Key abbreviations in EO/RS context:
- "EO" = Earth Observation - NOT "Essential Oils"
- "MSI" = Multispectral Instrument (e.g., Sentinel-2) -
NOT computer monitors
- "SAR" = Synthetic Aperture Radar -
NOT "Specific Absorption Rate"

**IMPORTANT: Accuracy Over Length**

Do NOT reward length or verbosity for its own sake.
A concise, accurate answer that captures the key facts
from the reference should
score EQUAL to a longer answer covering the same content.

- A 3-sentence accurate answer = A 10-sentence answer
with the same core information
- Only award higher scores for additional detail if it
adds meaningful, accurate
information beyond the reference
- Extra bullet points or formatting without additional
substance should NOT
increase the score

**Scoring Rules:**

- **Score 0-1**: Answer misinterprets core domain terms
OR contains major factual errors
- **Score 2**: Answer is vague or off-topic but doesn't
misinterpret terms
- **Score 3**: Answer is partially correct,
understands domain context
- **Score 4**: Answer correctly captures key facts
from the reference with good domain understanding
- **Score 5**: Answer is accurate, demonstrates clear
EO/RS domain expertise,
AND adds meaningful context

**Key Principle**: If the answer interprets domain
terms incorrectly
(e.g., "EO" as "Essential Oils", "MSI" as computer
monitors), score 0-1 regardless
of other content quality.

Question: "{question}"
Answer: "{output}"
Reference: "{reference}"

{format_instructions}
```

Figure 3: LLM-as-a-judge evaluation prompt for Open-Ended with retrieval context.

G.3 Active Reading Strategy Generation

For task-specific Active Reading, the model is prompted to both generate questions from a source chunk and devise active learning strategies tailored to each question. The prompt used for this process is shown in Figure 5.

G.4 Active Reading Predefined Strategy Selection

For predefined Active Reading, the model selects from a fixed set of predefined strategies based on strict eligibility rules applied to the source chunk. At most 2 strategies are selected per chunk. The

Active Reading chunk filter prompt

```
You are an expert in Earth Observation and Remote Sensing.
Your task is to assess the quality and relevance of the
following text chunk for training a specialized LLM.

The model covers 25 EO sub-disciplines, including:
atmospheric science, earth monitoring, environmental
science, geospatial intelligence, LULC, LiDAR,
multispectral/hyperspectral imaging, SAR and InSAR,
hydrology, oceanography, cryosphere, agriculture,
forestry, disaster monitoring, urban planning, geology,
climate science, thermal sensing, photogrammetry,
GNSS and geodesy, and geoinformatics.

Assign one of three ratings:
- "Best": highly relevant to one or more sub-disciplines
  AND high quality (accurate, informative, well-written).
- "Good": relevant but mediocre quality, OR high quality
  but only tangentially related.
- "Bad": poor quality or unrelated to Earth Observation.

Text Chunk: {text}
```

Figure 4: Prompt used to filter corpus chunks before Active Reading synthesis. Only *Best*-rated chunks enter the Active Reading pipeline.

Active Reading strategy generation prompt

```
You are an expert in Earth Observation and Remote Sensing,
covering SAR, LiDAR, Multispectral Imaging, LULC,
Atmospheric Science, Hydrology, and more.

Your task has two parts:

1. Generate Questions: generate ~4 distinct, insightful
   questions answerable from the document, covering its
   key information and nuances.

2. Devise Strategies: for EACH question, devise 2 diverse
   active learning strategies that help deeply internalize
   the *type* of knowledge required, not the answer itself.

Example strategies:
- Conceptual Visualization: diagram a processing chain
  (e.g., DEM generation from an InSAR pair).
- Comparative Analysis: contrast C-band vs. L-band SAR
  for biomass estimation.
- Practical Scenario: plan which sensor to task for a
  flood response and justify the choice.
- Analogy: explain spectral signature using how the human
  eye distinguishes colors.
- Data Interpretation: outline a rule-based classifier
  for coastal land cover using NDVI and band ratios.
- Problem Formulation: write a research question with
  required data and expected outcome.

Be creative and go beyond these examples to maximize
deep, conceptual understanding.

Document: {text}
```

Figure 5: Prompt used for task-specific Active Reading. The model first generates questions from the source chunk, then devises two active learning strategies per question to guide synthesis.

selection prompt, including the rule-based criteria, is shown in Figure 6.

Active Reading predefined strategy selection prompt

```
You are an expert curriculum designer for an advanced
Earth Observation course. Your task is to be extremely
selective and choose only the most appropriate strategies
for the given document.

Document: {text}

STRICT SELECTION RULES — only select a strategy if the
document clearly and substantially meets its criteria:

1. paraphrastic_restatement: ONLY IF the document
   contains more than 100 words of dense technical
   information.
2. acronym_glossary: ONLY IF at least 6 technical
   acronyms are explicitly defined using the pattern
   Full Name (Acronym). Acronyms without this pattern
   or non-technical ones (e.g., USA, EU, AI, ML) do
   not count. Acronyms must relate to Earth Observation.
3. timeline_generation: ONLY IF the document contains
   at least 10 distinct dates, years, or time-related
   events.
4. workflow_description: ONLY IF the document explicitly
   describes a complex procedural sequence or steps.
5. technical_tutorial: ONLY IF the document's primary
   focus is to explain a specific, named technique
   in detail.

From the strategies that pass these rules, select at
most 2 of the most impactful ones.
```

Figure 6: Prompt used for predefined Active Reading. The model applies strict eligibility rules to select at most 2 strategies from a fixed predefined set.

G.5 Active Reading Data Generation

Once strategies are selected, each is applied to its source chunk to generate the final synthetic training document. The generation template is shown in Figure 7.

Active Reading data generation prompt

```
Here is a learning strategy:
{strategy}

Apply this strategy to the following document:
{text}

Generate only the resulting document based on the
strategy. Do not add any conversational text or
introductions.
```

Figure 7: Prompt used to generate synthetic training documents by applying a selected Active Reading strategy to a source chunk. The model is explicitly instructed to output only the resulting document.

G.6 SelfQA Generation

SelfQA samples are derived from existing context-grounded QA pairs by reformulating them into fully self-contained questions that do not require access to a source document. The corresponding prompt is shown in Figure 8.

SelfQA generation prompt

```
You are an expert in Earth Observation (EO). Your task
is to transform a context-grounded QA pair into a fully
self-contained sample suitable for instruction fine-tuning.

ORIGINAL PAIR:
- Question: {question}
- Answer: {answer}

Generate a new Question and Answer pair following these
rules strictly:

1. Self-Contained Question:
  - Must NOT require the source document to be understood.
  - Do NOT use phrases like "in the text" or "according
    to the document".
  - Integrate necessary context directly. For example,
    transform "What is its resolution?" into "What is
    the spatial resolution of the Sentinel-2 MSI sensor?"
  - Must be clear and unambiguous on its own.

2. High-Quality Answer:
  - Base the answer on the original answer and source
    document. Modify form, not content.
  - Must be correct, complete, and detailed.
  - Written in full, explanatory, pedagogical sentences.
```

Figure 8: Prompt used to generate SelfQA samples. Context-grounded QA pairs are reformulated into self-contained questions that can be answered from the model's parametric knowledge alone.

G.7 ContextQA Quality Filtering

Generated ContextQA samples are filtered by an LLM judge following a two-step assessment: hard filters that immediately assign a *Wrong* rating for critical failures, followed by quality evaluation. The five-point scale maps to the labels used in Table 3: *Best*, *Good*, *Mid*, *Bad*, and *Wrong*. The filtering prompt is shown in Figure 9.

ContextQA quality filter prompt

```
You are an expert quality analyst with deep knowledge in
Earth Observation (EO). Act as a strict quality gate for
a generated instruction-tuning sample.

INPUT:
- Question: {question}
- Answer: {answer}

ASSESSMENT FLOW:
1. Hard Filters: if the sample is not relevant to EO,
   has poor SFT style (not conversational or detailed),
   contradicts the source document, or is too trivial,
   assign Wrong immediately.
2. Quality Evaluation: if it passes, evaluate question
   quality and answer correctness and completeness.

RATING SCALE:
- Best (top ~1%): flawless. The question uncovers a deep
  or non-obvious aspect of the document. The answer is
  correct, complete, and exceptionally well-written with
  rich context, examples, or analogies.
- Good (top ~5%): strong but not Best. The question is
  non-trivial and well-posed. The answer is correct and
  complete but lacks the deeper insight of Best.
- Mid: usable, with minor flaws. The question may be
  slightly basic or the answer correct but too concise.
- Bad: significant issues. The answer is partially
  correct or incomplete, or the question is ambiguous.
- Wrong: fails a hard filter or is factually wrong.
```

Figure 9: Prompt used to filter ContextQA samples. A two-step assessment first applies hard filters, then evaluates quality on a five-point scale. *Best* and *Good* samples are retained for training as described in Table 3.