Anemoi: An Agentic Framework for Weather Science

Anonymous Author(s)
Affiliation
Address
email

Abstract

Foundation models for weather science are pre-trained on vast amounts of structured numerical data and outperform traditional weather forecasting systems. However, these models lack language-based reasoning capabilities, limiting their utility in interactive scientific workflows. Large language models (LLMs) excel at understanding and generating text but cannot reason about high-dimensional meteorological datasets. We bridge this gap by building a novel agentic framework for weather science. Our framework includes a Python code-based environment for agents (`AnemoiWorld`) to interact with weather data, featuring tools like an interface to WeatherBench 2 dataset, geoquerying for geographical masks from natural language, weather forecasting, and climate simulation capabilities. We design `Anemoi`, a multi-turn LLM-based weather agent that iteratively analyzes weather datasets, observes results, and refines its approach through conversational feedback loops. We accompany the agent with a new benchmark, `AnemoiBench`, with a scalable data generation pipeline that constructs diverse question-answer pairs across weather-related tasks, from basic lookups to advanced forecasting, extreme event detection, and counterfactual reasoning. Experiments on this benchmark demonstrate strong promise for LLM agents to help weather scientists reason about meteorological data more effectively.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across diverse scientific domains [6], revolutionizing fields from drug discovery [66, 59] and materials science [29, 23] to network biology [53]. These models excel at processing textual content such as scientific literature, source code [24], and structured data tables [64]. However, their application to domains requiring reasoning over high-dimensional numerical data remains limited [55].

Meteorology offers a compelling yet challenging case study, as combining natural language reasoning with complex atmospheric data has the potential to greatly advance weather research. Weather prediction is a critical scientific challenge, with profound implications spanning agriculture, disaster preparedness, transportation, and energy management [2]. The field has witnessed remarkable progress through machine learning approaches, with foundation models [45, 27, 28, 5, 46] now achieving state-of-the-art performance in medium-range forecasting, often surpassing traditional physics-based numerical simulations [42, 4]. However, current weather models operate exclusively on structured numerical datasets such as reanalysis data, cannot incorporate valuable alternative modalities like textual weather bulletins or field station reports, and crucially, lack interactive natural language interfaces for querying or reasoning.

These highly technical tool interfaces (typically FORTRAN namelists) and esoteric formats (like GRIB and HDF) create substantial barriers for non-experts that severely limits the accessibility of valuable weather and climate data. Traditional meteorological workflows therefore require expert

Submitted to the Scaling Environments for Agents workshop (NeurIPS 2025). Do not distribute.

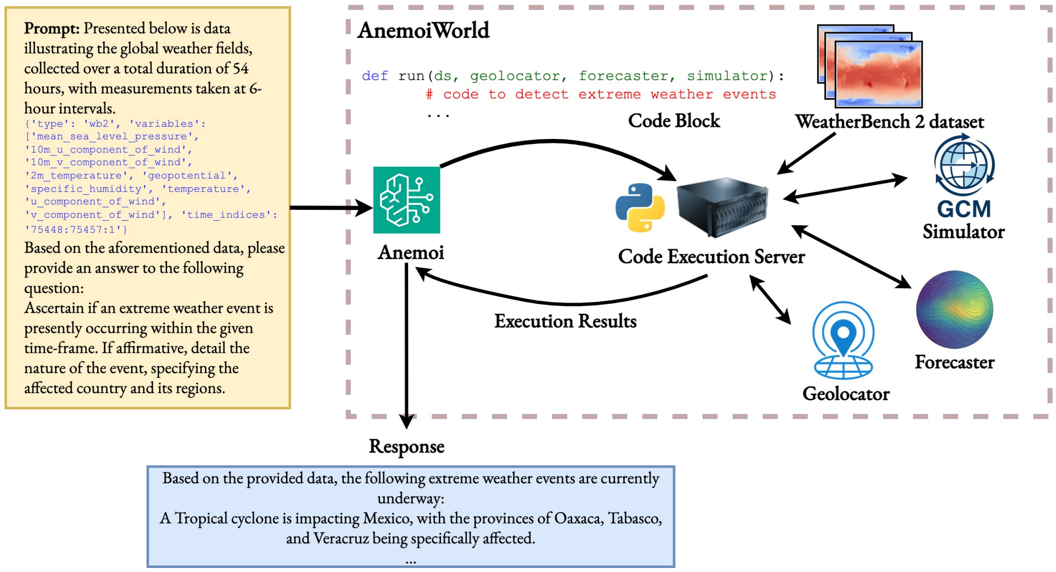

Figure 1: **Overview**: We develop `Anemoi`, an agentic framework for weather science. Given a query, the LLM-based agent `Anemoi` writes a code block which is sent to the code execution server. The server orchestrates several tools to execute the code block and returns the execution results to the agent. The agent either decides to execute more code to refine its output or respond back to the user.

interpretation to translate computational outputs into actionable insights, increasing costs and limiting their utility in human-in-the-loop decision-support systems.

Multimodal LLMs can handle data from diverse modalities and offer a potential pathway to address these challenges. Models capable of jointly processing text with images [56, 1, 32, 37, 31, 35, 36], video [65, 40, 63, 12, 34, 62], and audio [14, 13, 15, 57, 58, 17, 20] have shown impressive cross-modal reasoning abilities. Yet atmospheric data poses unique challenges: its spatiotemporal, multi-channel structure is fundamentally different from conventional modalities, requiring specialized approaches for effective integration with language models. Initial attempts to bridge this gap have shown promise but remain limited in scope. Early vision-language approaches to meteorology [10, 30, 39] have focused on narrow applications like extreme weather prediction using restricted variable subsets, falling short of general-purpose meteorological reasoning. More recent multimodal weather-language models [54] demonstrate the potential of this direction but still fail to match established baselines across many important meteorological tasks. This persistent gap highlights a fundamental challenge: despite significant progress in both weather foundation models and LLMs, no existing system successfully unifies meteorological data with natural language reasoning for broad, interactive scientific applications.

We address this challenge by first introducing an agentic environment that enables LLMs to interact programmatically with meteorological data and models. We setup `AnemoiWorld`, a comprehensive execution environment that exposes weather-focused capabilities through easy-to-use Python APIs. The system includes interfaces to the WeatherBench 2 dataset [49], geoquery functionality for translating between coordinates and named locations, state-of-the-art forecasting models [46], and physics-based simulators. A FastAPI backend parallelizes code execution from LLM-generated queries.

We then develop two code-generating systems of increasing sophistication within this agentic framework. `Anemoi-Direct` generates Python code in a single step to solve weather problems directly [19]. `Anemoi-Reflective` employs an iterative execution–refinement workflow: it executes code to manipulate weather data, analyzes the results, and refines both code and output before providing a final answer. Both approaches can automatically detect and correct errors produced during code execution. Figure 1 gives an overview of our entire agentic pipeline.

To systematically evaluate these approaches, we construct `AnemoiBench`, a comprehensive benchmark built on ERA5 reanalysis data [21] from WeatherBench 2 [49]. The benchmark combines human-authored and semi-synthetic tasks spanning 2062 question–answer pairs across 46 distinct tasks. Tasks range from basic data lookups and forecasting to challenging research problems involving extreme event detection, forecast report generation, and prediction and counterfactual analysis. We also implement robust evaluation schemes to assess the scientific accuracy of all generated answers across diverse meteorological reasoning tasks. We summarize our key contributions below.

- We develop `AnemoiWorld`, an agentic environment providing unified Python APIs for meteorological data, forecasting models, and climate simulation tools.
- We introduce two code-generating systems that leverage `AnemoiWorld`: `Anemoi-Direct` for single-step code generation and `Anemoi-Reflective` for iterative execution-refinement workflows to solve open-ended meteorological problems.
- We curate `AnemoiBench`, a challenging weather reasoning benchmark with 2062 question-answer pairs across 46 meteorological task types.
- Our evaluation shows that LLM agents achieve encouraging results on the benchmark, suggesting that they can be effective assistants to weather scientists.

2 Related Work

Weather Foundation Models. Neural network-based weather forecasting systems [28, 48, 5, 47, 45, 7, 46] have revolutionized meteorological prediction by demonstrating superior performance compared to conventional physics-based approaches [42] while being significantly more computationally efficient. Nevertheless, these architectures are predominantly trained for forecasting. In particular, they do not support conversational interfaces or cross-domain reasoning capabilities.

Agentic frameworks for scientific discovery Agentic frameworks implement the perceive–reason–plan–act loop by pairing LLMs with tools, memory, and feedback to pursue long-horizon goals. Core patterns include interleaving reasoning with tool calls (ReAct [61]), self-critique with episodic memory (Reflexion [51]), and self-supervised learning of API use (Toolformer [50]). General-purpose libraries such as AutoGen provide a standard interface for multi-agent conversation and tool invocation, making these patterns reusable across tasks [60].

In many scientific applications, these frameworks appear as domain agents and self-driving labs. In chemistry, ChemCrow couples an LLM controller with a curated set of expert tools for synthesis and analysis [9], while Coscientist integrates retrieval, code execution, and laboratory APIs to plan and run experiments end-to-end [8]. Biomedical agents extend the approach across literature, databases, and analysis workflows (e.g., Biomni [22]). Despite these advances across multiple scientific domains, weather science remains largely unexplored territory for agentic approaches.

General-Purpose Vision-Language Models. Multi-modal vision language models [33, 1, 32, 31, 37, 38, 35, 36] demonstrate strong visual reasoning capabilities on general-purpose evaluation benchmarks. However, adapting these models for applications in weather science presents considerable difficulties. Standard VLM architectures assume RGB visual inputs and exhibit weaknesses in quantitative analytical tasks. Meteorological data presents fundamentally different challenges through high-dimensional, structured atmospheric measurements requiring specialized integration approaches for language model compatibility. While weather-language hybrid models [54] seem promising, they underperform relative to domain-specific baselines across critical meteorological applications.

Multimodal Weather Datasets. Recent research has developed several multimodal frameworks that combine weather observations with textual information. These include the Terra collection [11], which integrates geographical imagery with descriptive text for general earth observation, and ClimateIQA [10], which focuses on extreme weather detection through wind measurement analysis. Similarly, WeatherQA [39] specializes in severe weather interpretation using remote sensing data and expert commentary, while CLLMate [30] connects media reports with ERA5 observations for weather event classification. Despite these valuable contributions, existing frameworks are narrow in scope. They concentrate on narrow applications or utilize only small subsets of atmospheric variables. This approach overlooks a fundamental characteristic of atmospheric dynamics: weather systems involve complex multi-scale interactions across numerous meteorological parameters. To address these limitations, our benchmark incorporates diverse weather reasoning tasks, both human-implemented and semi-synthetically generated, that span across most WeatherBench2 data channels.

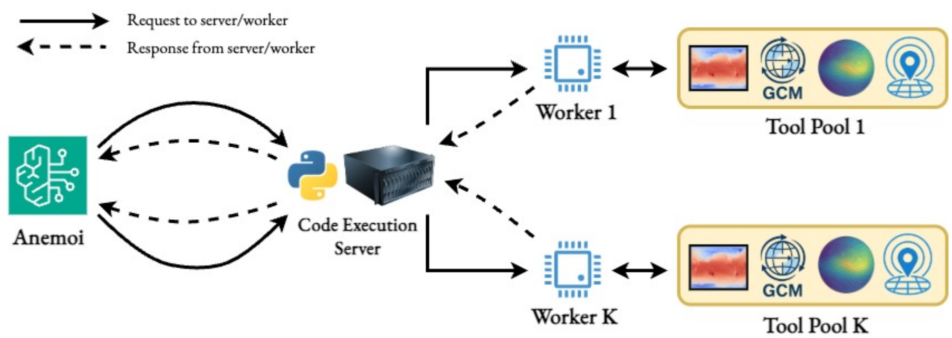

Figure 2: **Code Execution Server.** `Anemoi` sends parallel requests to the server, which distributes them to available workers. Each worker acquires resources from tool pools, loads datasets, injects tools into the execution environment, executes code, and returns results or errors to the agent.

3 `Anemoi`: An Agentic Framework for Weather Science

3.1 `AnemoiWorld`: An Agentic Environment for Weather Science

The fragmented nature of weather science tools makes it challenging for LLMs to effectively leverage them for scientific tasks. To address this, we introduce `AnemoiWorld`, a comprehensive agentic environment that unifies weather science capabilities from diverse tools through a clean Pythonic interface. Given a question, we leverage LLMs' ability [19, 25] to generate Python code and execute it in a sandboxed environment. The output is then fed back to the model, along with any execution errors. We design high-level APIs for the tools for ease of use, and include documentation extracted from the docstrings in the models context at inference time.

The environment encompasses several essential weather science tools:

1. **WeatherBench 2 Data Indexer.** The environment provides the model access to the data through the `xarray` dataset interface.
2. **Geolocator.** This tool provides comprehensive geospatial functionality for weather data analysis. It handles forward geocoding (place names to coordinates) and reverse geocoding (coordinates to location names) using the Natural Earth dataset [44]. Key operations include finding geographic features at specific coordinates, retrieving boolean masks and area-weighted maps for regions, listing sublocations, and calculating geodistances. Built using `geopandas` and `shapely`, it maintains precomputed spatial caches for fast lookups.
3. **Forecaster.** We incorporate the Stormer model [46], a transformer-based neural weather prediction system trained on WeatherBench 2. We chose it for its strong performance at short to medium range forecasts while being orders of magnitude more efficient than traditional numerical models. Our implementation abstracts checkpoint loading and preprocessing, providing a simple interface to run forecasts from arbitrary atmospheric initial conditions and return outputs as `xarray` datasets.
4. **Simulator.** Our JAX-GCM simulator is an intermediate complexity atmospheric model built on NeuralGCM's dynamical core [26]. It incorporates physical parameterizations from the SPEEDY Fortran model [42], including radiation, moist physics (clouds and convection), and vertical and horizontal diffusion. We use the default T32 configuration (approximately 3.5° resolution) with 8 vertical layers. Built on JAX, we can run 5-day simulations in only $\approx 25s$ on an A100 GPU.

Code Execution Server. `AnemoiWorld` requires a system capable of handling multiple weather analysis tasks simultaneously without resource conflicts. We implement a FastAPI-based server-client architecture where clients send code execution requests to a dedicated execution server that processes them in parallel. The system maintains resource pools for each tool component to prevent contention and enable true parallelism. Each pool contains one or more instances of the above tools. A resource manager implements acquire/release semantics to ensure each execution thread has exclusive access to a complete set of tools while preventing deadlocks.

Each execution follows a strict protocol: acquire resources from pools, load requested datasets, inject tool instances into the execution environment, and execute user code with timeout protection. The system captures all outputs and error information, which are sent back to the client for further processing by the agent. Figure 2 provides an overview of the server.

3.2 The Anemoi Family of Weather Agents

We design agentic systems that leverage AnemoiWorld to solve complex meteorological tasks. Our approach constructs prompts containing comprehensive documentation of AnemoiWorld tools, variable descriptions, units, and coordinate systems. The models generate Python functions using these tools to solve the given questions, which execute on AnemoiWorld's code execution server. Any execution errors or timeouts are returned to the models, which regenerate code until the error is resolved. We implement two distinct systems that differ in their execution strategy and refinement approach. Both systems intentionally maintain simple designs to isolate and measure the agentic capabilities of LLMs for solving weather science problems.

Anemoi-Direct generates a complete Python solution in one attempt and reports the execution output as the final answer. This model runs the error-correction loop for a maximum of 5 times.

Anemoi-Reflective implements a multi-turn workflow that alternates between code generation and execution phases. The agent executes individual code blocks and receives the output as observations. The execution results are fed back to the LLM, which analyzes the observations and decides on the next step. This iterative process enables the model to assess the scientific plausibility of outputs, identify anomalies or mistakes in results, and refine subsequent code blocks to address logical errors. We run the interaction loop for a maximum of 20 times per question.

4 AnemoiBench: A Comprehensive Weather Benchmark

Weather science problems require complex analysis of multi-scale atmospheric patterns, statistical modeling of trends, and integration of diverse datasets from numerical models and expert reports. We introduce AnemoiBench, a comprehensive benchmark that evaluates how effectively LLMs can assist in real-world meteorological workflows. The benchmark comprises 46 distinct meteorological tasks with answers derived from curated weather reports and human-generated or verified code.

4.1 Dataset Curation

We base our tasks around the ERA5 reanalysis dataset [21], specifically from WeatherBench 2 [49]. The dataset provides global atmospheric data from 1979 to 2022. We use 1.5° spatial resolution with 6-hourly temporal resolution.

The capabilities measured by our curated tasks range from basic data lookups and computations to more advanced problems involving forecasting, challenging research problems including extreme event detection, forecast report generation, prediction analysis, and counterfactual reasoning. We design tasks with increasing difficulty levels based on the complexity of tool usage required to answer them, from simple single-step data queries to multi-step analytical workflows. Table 5 provide an overview of the task types we implement as part of our benchmark.

For each task-type, we define natural language templates with placeholders such as location, variable, and time window. To create task-specific examples, these placeholders are filled by randomly sampling inputs, and the corresponding ground truth is computed deterministically using human-written or human-verified synthetic code applied to the raw ERA5 data. To add more diversity to the training dataset, we use an LLM (specifically, GPT-4o) to reword questions generated by our data curation pipeline. Figure 4 shows an example template, and a sample generated from it.

Using our framework, we construct a benchmark dataset comprising 2062 test samples spread across 46 tasks. For a detailed breakdown of dataset statistics, please refer to Appendix A.1. We provide more details about how the tasks are implemented in the subsequent sections.

4.1.1 Human-generated tasks

The human-generated tasks span all difficulty levels and represent realistic meteorological queries curated in conjunction with a domain expert. For each task, a graduate student created a question template and wrote Python code to answer the query. Easy tasks focus on basic data retrieval operations like finding extrema, querying specific values, and identifying locations with particular weather conditions. Medium-difficulty tasks introduce forecasting elements, asking for future weather predictions at specific locations and times. Hard tasks incorporate more complex analytical concepts such as anomaly detection relative to baselines and counterfactual scenario analysis. The most

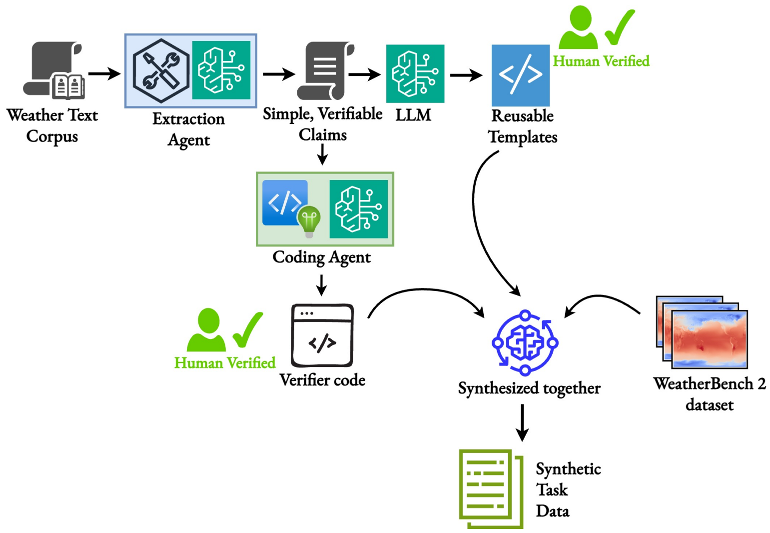

Figure 3: **Semi-synthetic task generation pipeline**: Semi-synthetic pipeline for generating weather benchmark tasks. Weather-related texts are processed by a claim extraction agent to identify scientifically meaningful observational claims, which are then verified against ERA5 meteorological data through automated code generation. Verified claims are transformed into reusable templates and manually reviewed. We can combine the verifier code with the templates and Weatherbench data to produce samples.

challenging tasks demand comprehensive meteorological expertise and mirror real-world operational workflows. These include extreme weather event detection, comprehensive weather assessments, and generation of detailed forecast discussions that span regional to global scales. For instance, ENSO outlook reports require synthesizing complex interactions between multiple atmospheric and oceanic variables to produce coherent, scientifically grounded forecasts. We source the expert-generated weather discussion reports from several online sources, such as the NOAA website[1] and IRI Seasonal Climate Forecasts/Outlooks. For extreme weather event tasks, we use records from the EM-DAT international disaster database [16], matching event entries by date and location to the ERA5 data.

4.1.2 Semi-synthetic task generation

To increase task diversity, we implement a semi-synthetic pipeline that transforms unstructured weather-related text into verifiable benchmark tasks. Figure 3 provides an overview of the procedure. The process begins with a claim extraction agent that analyzes weather texts from various sources, using an LLM to identify scientifically meaningful observational claims about weather phenomena. The agent focuses on quantifiable changes, trends, extremes, and relationships between variables.

These claims undergo verification through an automated agent that generates executable Python code to validate each claim against the ERA5 data. This verification step ensures that extracted claims are not only linguistically coherent but also scientifically accurate when tested against actual meteorological observations. The verified claims are then transformed into reusable templates that support both quantitative measurements and qualitative comparisons, allowing generation of diverse benchmark examples through parameter substitution.

We generate multiple candidate templates through this approach. Finally, we manually review them for scientific interest and code correctness. In this way, we generate 32 distinct synthetic task types.

4.2 Evaluation Metrics

Since all our tasks are designed around weather tasks with objectively correct answers, we design an evaluation pipeline that can assess the scientific correctness of the answers produced by the models. The model answers fall into five primary categories: **numeric**, **temporal**, **spatial (location-based)** and **descriptive**. Given that model outputs are in natural language, we evaluate them through a multi-stage process:

[1]https://www.wpc.ncep.noaa.gov/discussions/hpcdiscussions.php?disc=pmdepd

1. **Verification:** Determine whether the models response contains a relevant and valid answer. At this stage, we merely assess whether or not the response has an appropriate answer to the given question, and not its correctness. We use `gpt-4.1-mini` for this purpose.
2. **Extraction:** Extract the specific answer from the model response using another LLM prompt.
3. **Scoring:** Apply scoring methods specific to the type of question, which are detailed below.

Numerical Answers. For numerical responses, we report the Standarized Median Absolute Error between the predicted and reference values. In addition, we also report the $25\%, 75\%$ and 99% quantiles of the standarized absolute error to provide a more complete picture of the error distribution. To compare across variables with different scales and units, we divide the absolute error by the standard deviation of the corresponding variable in the dataset.

Time-based Answers. We evaluate tasks with time values as responses using Median Absolute Error. We omit the standarization step, since all the answers are in the same units (that is, hours). Like the numerical answers case, we also report the $25\%, 75\%$ and 99% quantiles.

Location-based Answers. For questions whose answers are geographic locations, we first match the extracted location name to one of the expected entries from the NaturalEarth dataset (e.g., mapping "USA" to "United States of America"). For countries, we use the `country_converter` library [52]. For other geographic entities such as continents and water bodies, we apply fuzzy string matching [3], accepting matches above a predefined similarity threshold.

To quantitatively assess the geographic deviation between predicted and reference locations, we employ the Earth Mover's Distance (EMD) [43] as a primary evaluation metric. We begin by generating surface area-weighted masks over a latitude–longitude grid for both the predicted and reference locations. These masks are normalized to form probability distributions. To account for the curvature of the Earth, we compute pairwise distances between grid points using geodesic distance. The EMD is then calculated using the `POT` library [18]. As a complementary metric, we also report Location Accuracy, which simply measures whether the predicted and reference location strings are an exact match.

Descriptive Answers. To evaluate descrptive answers, we employ a decomposition-and-aggregation approach where the model's response is first parsed into individual discussion points, each of which is then probabilistically scored against the claims in the reference answer to determine whether it supports or refutes the ground truth. Using logit probabilities from language model inference, the system calculates how strongly each extracted point aligns with the reference material by comparing the likelihood of `SUPPORTS` versus `REFUTES` tokens, converting these into numerical scores that capture the degree of alignment [41]. The final evaluation aggregates these individual point scores into an overall discussion quality metric, enabling fine-grained assessment that accounts for both the factual accuracy and argumentative coherence of complex, multi-faceted responses rather than treating the entire discussion as a monolithic unit.

Extreme Weather Tasks. In order to evaluate the extreme-weather tasks, we report two metrics: (1) F1 score, which only assesses whether the model correctly predicts the *occurrence* of an extreme event anywhere in the world, without considering event type or exact location. (2) Earthmover's Distance, which measures the agreement between the reference and predicted list of countries.

Model	LLM	SAE (Q25) (↓)	SAE (Q50) (↓)	SAE (Q75) (↓)	SAE (Q99) (↓)
`Anemoi-Reflective`	`gpt-5-mini`	2.68e-08	0.029	0.513	**17.753**
`Anemoi-Direct`	`gpt-5-mini`	**0.0**	**0.018**	**0.288**	55.859
Text Only LLM	`gpt-5-mini`	0.290	0.935	2.172	27.285
`Anemoi-Reflective`	`gpt-5-nano`	0.0001	0.053	0.955	**12.002**
`Anemoi-Direct`	`gpt-5-nano`	**0.0**	**0.049**	**0.751**	443.282
Text Only LLM	`gpt-5-nano`	0.265	1.074	2.799	3116.1
`Anemoi-Reflective`	`gemini-2.5-flash`	**0.0**	0.028	0.212	297.8
`Anemoi-Direct`	`gemini-2.5-flash`	**0.0**	**0.012**	**0.142**	**58.29**
Text Only LLM	`gemini-2.5-flash`	0.530	1.309	3.798	25228.7
`Anemoi-Reflective`	`Qwen3-Coder-30B`	0.015	0.245	**1.187**	21528.3
`Anemoi-Direct`	`Qwen3-Coder-30B`	**0.004**	**0.175**	1.256	**387.21**
Text Only LLM	`Qwen3-Coder-30B`	0.354	1.345	6.521	13381.4

Table 1: Output validity and error metric quantiles for numerical tasks. SAE stands for standardized absolute error, the absolute error divided by the standard deviation of the relevant variable in the data.

Model	LLM	AE (Q25) (↓)	AE (Q50) (↓)	AE (Q75) (↓)	AE (Q99) (↓)
`Anemoi-Reflective`	`gpt-5-mini`	**0.0**	**0.0**	**12.0**	**146.1**
`Anemoi-Direct`	`gpt-5-mini`	**0.0**	**0.0**	**12.0**	156.0
Text Only LLM	`gpt-5-mini`	12	30	72	26841.6
`Anemoi-Reflective`	`gpt-5-nano`	**0**	**0**	**6**	39521.3
`Anemoi-Direct`	`gpt-5-nano`	**0**	**0**	18	5.57e18
Text Only LLM	`gpt-5-nano`	12	36	93	**190.68**
`Anemoi-Reflective`	`gemini-2.5-flash`	**0**	**0**	**6**	**122.04**
`Anemoi-Direct`	`gemini-2.5-flash`	**0**	**0**	48	8.85e18
Text Only LLM	`gemini-2.5-flash`	12	36	72	157.92
`Anemoi-Reflective`	`Qwen3-Coder-30B`	**0**	18	37.5	504377
`Anemoi-Direct`	`Qwen3-Coder-30B`	**0**	**6**	**24**	8.67e18
Text Only LLM	`Qwen3-Coder-30B`	6	24	54	**144**

Table 2: Absolute error quantiles for time tasks, in units of hours.

Model	LLM	Location Accuracy (%)(↑)	EMD (km) (↓)	Extreme Weather F1 (↑)
`Anemoi-Reflective`	`gpt-5-mini`	**89.05**	**2084.39**	0.432
`Anemoi-Direct`	`gpt-5-mini`	77.11	2317.97	**0.466**
Text Only LLM	`gpt-5-mini`	16.92	5916.13	0.421
`Anemoi-Reflective`	`gpt-5-nano`	65.17	**2354.28**	**0.212**
`Anemoi-Direct`	`gpt-5-nano`	**72.14**	2549.28	0.184
Text Only LLM	`gpt-5-nano`	15.42	5132.35	0
`Anemoi-Reflective`	`gemini-2.5-flash`	66.67	**2237.11**	0.382
`Anemoi-Direct`	`gemini-2.5-flash`	**75.62**	2400.46	**0.425**
Text Only LLM	`gemini-2.5-flash`	9.45	3069.93	0.247
`Anemoi-Reflective`	`Qwen3-Coder-30B`	**27.86**	3115.14	0.292
`Anemoi-Direct`	`Qwen3-Coder-30B`	15.92	**2224.72**	0.260
Text Only LLM	`Qwen3-Coder-30B`	14.43	6130.85	**0.586**

Table 3: Location metrics for location answer-based questions. EMD stands for Earth mover's Distance.

Model	LLM	% Valid Outputs (↑)	Discussion Score (↑)	Boolean F1 (↑)
`Anemoi-Reflective`	`gpt-5-mini`	95.00	**0.264**	0.538
`Anemoi-Direct`	`gpt-5-mini`	94.52	0.255	**0.585**
Text Only LLM	`gpt-5-mini`	**95.49**	0.238	0.369
`Anemoi-Reflective`	`gpt-5-nano`	90.30	**0.351**	0.452
`Anemoi-Direct`	`gpt-5-nano`	**97.48**	0.267	**0.496**
Text Only LLM	`gpt-5-nano`	91.78	0.344	0.397
`Anemoi-Reflective`	`gemini-2.5-flash`	89.23	0.275	**0.658**
`Anemoi-Direct`	`gemini-2.5-flash`	**96.75**	0.235	0.594
Text Only LLM	`gemini-2.5-flash`	73.96	**0.383**	0.222
`Anemoi-Reflective`	`Qwen3-Coder-30B`	85.55	0.292	0.484
`Anemoi-Direct`	`Qwen3-Coder-30B`	**87.49**	0.231	**0.490**
Text Only LLM	`Qwen3-Coder-30B`	86.03	**0.344**	0.397

Table 4: Overall percentage of valid outputs, numerical score (0-1) for discussion questions, and F1 score for boolean questions.

5 Experimental Results

We evaluate model performance across all task types from Section 4. As a zero-shot baseline, we test a pre-trained frontier language model on weather reasoning questions using only natural language metadata, that is, no structured weather data or numerical inputs. We use OpenAI `gpt-5-mini`, OpenAI `gpt-5-nano`, Google `gemini-2.5-flash`, and `Qwen3-Coder-30B-A3B-Instruct` as backend models for our `Anemoi` agents. Tables 1 to 4 report results on `AnemoiBench` for all models and the text-only baseline.

The `Anemoi` agents significantly outperform the text-only baseline across all tasks, demonstrating the agentic framework's ability to effectively leverage the numerical data from WeatherBench. The agents excel at numerical and temporal tasks, achieving very low absolute errors at the 25th and 50th percentiles. For location prediction, `Anemoi-Reflective` with `gpt-5-mini` achieves a strong performance, with 89.05% accuracy and an EMD score of 2084.39. The agents show promise in extreme weather detection (F1 scores > 0.4) and weather claim validation (best F1: 0.658). However, all models struggle with report generation, with the best achieving only 0.351 on discussion scores. The relatively strong performance of Text Only LLMs in report generation shows the importance of LLM priors in this task and the room for improvement in effectively using code as part of the natural language report generation process. We find that, relative to the other LLMs tested, `Qwen3-Coder-30B-A3B-Instruct` shows comparatively weak performance across most tasks, perhaps due to the model's smaller size and the long-context code generation necessary to solve the tasks.

While Direct variants typically perform better on numerical tasks, Reflective variants show greater resilience against extreme errors (99th percentile). This suggests self-reflection helps detect anomalies like wrong magnitudes or unit mismatches. Reflective variants also outperform Direct variants in report generation, likely because Direct models produce rigid responses since they directly output the program outputs to text. For a more detailed breakdown of the results grouped by difficulty level, refer to Appendix A.3.

6 Conclusion

We tackled the challenging problem of enabling LLMs to reason over high-dimensional weather data by developing, to our knowledge, the first agentic model for meteorology. Our contributions include: (1) `AnemoiWorld`, an agentic environment with comprehensive meteorological tools, (2) the `Anemoi` family of agents that leverage these tools, and (3) a scalable data pipeline producing a large, diverse benchmark dataset (`AnemoiBench`). Our empirical evaluation shows that the agentic framework enables effective reasoning about meteorological data, significantly outperforming text-only baselines. The agents excel at most tasks but struggle with complex challenges like forecast report generation. Beyond advancing weather science, our work provides a sandbox for developing more effective agentic workflows. Future work could explore using larger datasets to train agents that produce more scientifically accurate responses.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[2] Richard B Alley, Kerry A Emanuel, and Fuqing Zhang. Advances in weather prediction. *Science*, 363(6425):342–344, 2019.

[3] Max Bachmann, layday, Georgia Kokkinou, Jeppe Fihl-Pearson, dj, Henry Schreiner, Moshe Sherman, Michał Górny, pekkarr, Delfini, Dan Hess, Guy Rosin, Hugo Le Moine, Kwuang Tang, Nicolas Renkamp, Trenton H, glynn, and odidev. rapidfuzz/rapidfuzz: Release 3.6.1, December 2023.

[4] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

[5] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

[6] Abeba Birhane, Atoosa Kasirzadeh, David Leslie, and Sandra Wachter. Science in the age of large language models. *Nature Reviews Physics*, 5(5):277–280, 2023.

[7] Cristian Bodnar, Wessel P Bruinsma, Ana Lucic, Megan Stanley, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan Weyn, Haiyu Dong, Anna Vaughan, et al. Aurora: A foundation model of the atmosphere. *arXiv preprint arXiv:2405.13063*, 2024.

[8] Daniil A Boiko, Robert MacKnight, Ben Kline, Gabe Gomes, et al. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023.

[9] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, Philippe Schwaller, et al. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.

[10] Jian Chen, Peilin Zhou, Yining Hua, Dading Chong, Meng Cao, Yaowei Li, Zixuan Yuan, Bing Zhu, and Junwei Liang. Vision-language models meet meteorology: Developing models for extreme weather events detection with heatmaps. *arXiv preprint arXiv:2406.09838*, 2024.

[11] Wei Chen, Xixuan Hao, Yuankai Wu, and Yuxuan Liang. Terra: A multimodal spatio-temporal dataset spanning the earth. *Advances in Neural Information Processing Systems*, 37:66329–66356, 2024.

[12] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

[13] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

[14] Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*, 2023.

[15] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

[16] D. Delforge, V. Wathelet, R. Below, C. Lanfredi Sofia, M. Tonnelier, J. A. F. van Loenhout, and N. Speybroeck. EM-DAT: The Emergency Events Database. *International Journal of Disaster Risk Reduction*, page 105509, 2025.

[17] Seungheon Doh, Keunwoo Choi, and Juhan Nam. Talkplay: Multimodal music recommendation with large language models. *arXiv preprint arXiv:2502.13713*, 2025.

[18] RÃ©mi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, AurÃ©lie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, LÃ©o Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[19] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR, 2023.

[20] Sreyan Ghosh, Zhifeng Kong, Sonal Kumar, S Sakshi, Jaehyeon Kim, Wei Ping, Rafael Valle, Dinesh Manocha, and Bryan Catanzaro. Audio flamingo 2: An audio-language model with long-audio understanding and expert reasoning abilities. *arXiv preprint arXiv:2503.03983*, 2025.

[21] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, 146(730):1999–2049, 2020.

[22] Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, et al. Biomni: A general-purpose biomedical ai agent. *bioRxiv*, 2025.

[23] Kevin Maik Jablonka, Qianxiang Ai, Alexander Al-Feghali, Shruti Badhwar, Joshua D Bocarsly, Andres M Bran, Stefan Bringuier, L Catherine Brinson, Kamal Choudhary, Defne Circi, et al. 14 examples of how llms can transform materials science and chemistry: a reflection on a large language model hackathon. *Digital discovery*, 2(5):1233–1250, 2023.

[24] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*, 2024.

[25] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*.

[26] Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, et al. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024.

[27] Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar. Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In *Proceedings of the platform for advanced scientific computing conference*, pages 1–11, 2023.

[28] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

[29] Ge Lei, Ronan Docherty, and Samuel J Cooper. Materials science in the era of large language models: a perspective. *Digital Discovery*, 3(7):1257–1272, 2024.

[30] Haobo Li, Zhaowei Wang, Jiachen Wang, Alexis Kai Hon Lau, and Huamin Qu. Cllmate: A multimodal llm for weather and climate events forecasting. *arXiv preprint arXiv:2409.19058*, 2024.

[31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[32] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[33] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

[34] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5971–5984, 2024.

[35] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[36] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[39] Chengqian Ma, Zhanxiang Hua, Alexandra Anderson-Frey, Vikram Iyer, Xin Liu, and Lianhui Qin. Weatherqa: Can multimodal language models reason about severe weather? *arXiv preprint arXiv:2406.11217*, 2024.

[40] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[41] Veeramakali Vignesh Manivannan, Yasaman Jafari, Srikar Eranky, Spencer Ho, Rose Yu, Duncan Watson-Parris, Yian Ma, Leon Bergen, and Taylor Berg-Kirkpatrick. Climaqa: An automated evaluation framework for climate question answering models. In *The Thirteenth International Conference on Learning Representations*.

[42] Franco Molteni, Roberto Buizza, Tim N Palmer, and Thomas Petroliagis. The ecmwf ensemble prediction system: Methodology and validation. *Quarterly journal of the royal meteorological society*, 122(529):73–119, 1996.

[43] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704, 1781.

[44] Natural Earth. Natural earth data. `https://www.naturalearthdata.com/`, 2024. Accessed: 2024-11-15.

[45] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.

[46] Tung Nguyen, Rohan Shah, Hritik Bansal, Troy Arcomano, Romit Maulik, Rao Kotamarthi, Ian Foster, Sandeep Madireddy, and Aditya Grover. Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. *Advances in Neural Information Processing Systems*, 37:68740–68771, 2024.

[47] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[48] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, 2025.

[49] Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024.

[50] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

[51] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. 36:8634–8652, 2023.

[52] Konstantin Stadler. The country converter coco - a python package for converting country names between different classification schemes. *Journal of Open Source Software*, 2(16):332, 2017.

[53] Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

[54] Sumanth Varambally, Veeramakali Vignesh Manivannan, Yasaman Jafari, Luyu Han, Zachary Novack, Zhirui Xia, Salva Rühling Cachay, Srikar Eranky, Ruijia Niu, Taylor Berg-Kirkpatrick, Duncan Watson-Parris, Yian Ma, and Rose Yu. Aquilon: Towards building multimodal weather LLMs. In *ICML 2025 Workshop on Assessing World Models*, 2025.

[55] Lei Wang, Shan Dong, Yuhui Xu, Hanze Dong, Yalu Wang, Amrita Saha, Ee-Peng Lim, Caiming Xiong, and Doyen Sahoo. Mathhay: An automated benchmark for long-context mathematical reasoning in llms. *arXiv preprint arXiv:2410.04698*, 2024.

[56] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*, 2022.

[57] Junda Wu, Zachary Novack, Amit Namburi, Jiaheng Dai, Hao-Wen Dong, Zhouhang Xie, Carol Chen, and Julian McAuley. Futga: Towards fine-grained music understanding through temporally-enhanced generative augmentation. *arXiv preprint arXiv:2407.20445*, 2024.

[58] Junda Wu, Zachary Novack, Amit Namburi, Jiaheng Dai, Hao-Wen Dong, Zhouhang Xie, Carol Chen, and Julian McAuley. Futga-mir: Enhancing fine-grained and temporally-aware music understanding with music information retrieval. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2025.

[59] Kehan Wu, Yingce Xia, Pan Deng, Renhe Liu, Yuan Zhang, Han Guo, Yumeng Cui, Qizhi Pei, Lijun Wu, Shufang Xie, et al. Tamgen: drug design with target-aware molecule generation through a chemical language model. *Nature Communications*, 15(1):9360, 2024.

[60] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

[61] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[62] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[63] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, 2023.

[64] Xiaokang Zhang, Jing Zhang, Zeyao Ma, Yang Li, Bohan Zhang, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, et al. Tablellm: Enabling tabular data manipulation by llms in real office usage scenarios. *arXiv preprint arXiv:2403.19318*, 2024.

[65] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *arXiv preprint arXiv:2212.04501*, 2022.

[66] Yizhen Zheng, Huan Yee Koh, Maddie Yang, Li Li, Lauren T May, Geoffrey I Webb, Shirui Pan, and George Church. Large language models in drug discovery and development: From disease mechanisms to clinical trials. *arXiv preprint arXiv:2409.04481*, 2024.

A Appendix

A.1 Dataset Details

Table 5 details all the tasks in `AnemoiBench`, and table A.1 reports the number of samples generated grouped by difficulty and type.

ID	Natural Language Description	Answer Type	Difficulty	Type
1	Which location experienced the highest/lowest average variable	Location	Easy	Human
2	What is the min/max/mean variable in location	Numerical	Easy	Human
3	Which sublocation has the highest/lowest recorded variable	Location	Easy	Human
4	How many hours from start did location experience extremum	Temporal	Easy	Human
5	What is the variable value at location at specific time	Numerical	Easy	Human
6	What will the variable be in location after time interval (forecast)	Numerical	Medium	Human
7	When will location experience its extremum in future period (forecast)	Temporal	Medium	Human
8	Difference between max and min within region (forecast)	Numerical	Medium	Synthetic
9	Maximum difference between two regions (forecast)	Numerical	Medium	Synthetic
10	Maximum value in region (forecast)	Numerical	Medium	Synthetic
11	By how much minimum will fall below threshold in first N days (forecast)	Numerical	Medium	Synthetic
12	By how much minimum will be below threshold across region (forecast)	Numerical	Medium	Synthetic
13	Maximum day-to-day decrease between consecutive days (forecast)	Numerical	Medium	Synthetic
14	Maximum value observed anywhere in region (forecast)	Numerical	Medium	Synthetic
15	How much mean will differ between two regions (forecast)	Numerical	Medium	Synthetic
16	Difference in mean between two regions (forecast)	Numerical	Medium	Synthetic
17	Accumulated total in region (forecast)	Numerical	Medium	Synthetic
18	Time-averaged value of variable in region (forecast)	Numerical	Medium	Synthetic
19	How much area-averaged value will increase from current (forecast)	Numerical	Medium	Synthetic
20	Maximum value expected in region (forecast)	Numerical	Medium	Synthetic
21	Minimum value averaged over region (forecast)	Numerical	Medium	Synthetic
22	Fraction p of grid points will exceed threshold (forecast)	Yes/No	Medium	Synthetic
23	Temporal trend will exceed threshold (forecast)	Yes/No	Medium	Synthetic
24	Spatial difference between regions will exceed threshold (forecast)	Yes/No	Medium	Synthetic
25	Count of grid points will exceed threshold (forecast)	Yes/No	Medium	Synthetic
26	Minimum will exceed threshold in $> N\%$ of grid points (forecast)	Yes/No	Medium	Synthetic
27	Variable will exceed threshold at $> N\%$ of grid points (forecast)	Yes/No	Medium	Synthetic
28	Which locations experienced unusual anomaly vs baseline	List of locations	Hard	Human
29	Cumulative sum of positive anomalies above threshold (forecast)	Numerical	Hard	Synthetic
30	Maximum spatial extent exceeding threshold simultaneously (forecast)	Numerical	Hard	Synthetic
31	At least N consecutive days will exceed threshold (forecast)	Yes/No	Hard	Synthetic
32	Maximum will exceed threshold on at least N distinct days (forecast)	Yes/No	Hard	Synthetic
33	Maximum will exceed threshold on each of the final N days (forecast)	Yes/No	Hard	Synthetic
34	Maximum will exceed threshold for N consecutive days from day X (forecast)	Yes/No	Hard	Synthetic
35	Regional max and mean will simultaneously meet conditions (forecast)	Yes/No	Hard	Synthetic
36	Simultaneous conditions will occur in two regions (forecast)	Yes/No	Hard	Synthetic
37	Crossover between conditions will occur in timeframe (forecast)	Yes/No	Hard	Synthetic
38	Regional fraction exceeding threshold will meet criteria (forecast)	Yes/No	Hard	Synthetic
39	Variable will be within range for $> N$ contiguous grid points (forecast)	Yes/No	Hard	Synthetic
40	Zonal gradient will exceed threshold per degree longitude (forecast)	Yes/No	Hard	Synthetic
41	How will variable change in lead time if variable is modified (counterfactual)	Numerical	Hard	Human
42	Identify if extreme weather event will occur in next N hours (forecast)	Descriptions	Very Hard	Human
43	Check if extreme weather event is happening now	Descriptions	Very Hard	Human
44	Generate global 3-month climate forecast report (forecast)	Descriptions	Very Hard	Human
45	Provide detailed US meteorological analysis and forecast (forecast)	Descriptions	Very Hard	Human
46	Generate ENSO climate update and outlook (forecast)	Descriptions	Very Hard	Human

Table 5: Complete set of Weather Tasks, grouped by difficulty.

Difficulty	Human Tasks	Human Samples	Synthetic Tasks	Synthetic Samples	Total Samples
Easy	5	800	0	0	800
Medium	2	156	20	256	412
Hard	2	329	12	153	482
Very Hard	5	393	0	0	393
Total	**14**	**1,678**	**32**	**384**	**2,062**

Table 6: **Dataset Statistics**: Number of samples grouped by difficulty and type

A.2 Example from the dataset

The following data shows a snapshot of the global weather fields.

{data}

Based on the above data, answer the following question:

Which {geofeature} experienced the {extremum_direction} average {variable}?","Based on the provided data, {answer} experienced the {extremum_direction} average {variable} over the specified time-period, with an average {variable} of {answer_numeric}."

Example Template

The following data shows a snapshot of the global weather fields.


```
{'type': 'wb2', 'variables': ['mean_sea_level_pressure',
'10m_u_component_of_wind', '10m_v_component_of_wind',
'2m_temperature', 'geopotential', 'specific_humidity',
'temperature', 'u_component_of_wind', 'v_component_of_wind'],
'time_indices': '54746:54747:1'}
```


Based on the above data, answer the following question: Which continent experienced the highest average Surface temperature?

Based on the provided data, Africa experienced the highest average Surface temperature over the specified time-period, with an average Surface temperature of 303.5 K.

Generated Sample

Figure 4: (left) Example Template from which samples are generated (right) A sample generated using the template.

A.3 Performance by Difficulty Level

Below, we include a detailed breakdown of performance metrics by question difficulty level, as defined in Table 5, for models `gpt-5-mini`, `gemini-2.5-flash`, and `Qwen3-Coder-30B`. Model performance decreases as question difficulty increases, demonstrating the utility of including multiple difficulty levels in `AnemoiBench`.

LLM	Model Variant	SAE (Q25) (↓)	SAE (Q50) (↓)	SAE (Q75) (↓)	SAE (Q99) (↓)
`gpt-5-mini`	`Anemoi-Reflective`	**0.000**	**0.000**	**0.000**	0.881
`gpt-5-mini`	`Anemoi-Direct`	**0.000**	**0.000**	0.006	**0.400**
`gpt-5-mini`	`Text Only LLM`	0.219	0.648	1.265	17.933
`gemini-2.5-flash`	`Anemoi-Reflective`	**0.000**	**0.000**	0.053	2.204
`gemini-2.5-flash`	`Anemoi-Direct`	**0.000**	**0.000**	**0.022**	**0.457**
`gemini-2.5-flash`	`Text Only LLM`	0.395	1.015	2.583	13176.500
`Qwen3-Coder-30B`	`Anemoi-Reflective`	**0.000**	0.049	0.294	35.237
`Qwen3-Coder-30B`	`Anemoi-Direct`	**0.000**	**0.014**	**0.211**	**19.039**
`Qwen3-Coder-30B`	`Text Only LLM`	0.304	0.902	14.039	17287.300

LLM	Model Variant	AE (Q25) (↓)	AE (Q50) (↓)	AE (Q75) (↓)	AE (Q99) (↓)
`gpt-5-mini`	`Anemoi-Reflective`	**0.000**	**0.000**	**0.000**	138.000
`gpt-5-mini`	`Anemoi-Direct`	**0.000**	**0.000**	**0.000**	**132.180**
`gpt-5-mini`	`Text Only LLM`	12.000	24.000	48.000	144.000
`gemini-2.5-flash`	`Anemoi-Reflective`	**0.000**	**0.000**	**0.000**	**120.000**
`gemini-2.5-flash`	`Anemoi-Direct`	**0.000**	**0.000**	**0.000**	8.770e+18
`gemini-2.5-flash`	`Text Only LLM`	12.000	24.000	54.000	150.600
`Qwen3-Coder-30B`	`Anemoi-Reflective`	**0.000**	12.000	42.000	514405.000
`Qwen3-Coder-30B`	`Anemoi-Direct`	**0.000**	**0.000**	**18.000**	8.867e+18
`Qwen3-Coder-30B`	`Text Only LLM`	12.000	30.000	60.000	**144.180**

LLM	Model Variant	Location Acc. (%) (↑)	EMD (km) (↓)	% Valid Outputs (↑)
`gpt-5-mini`	`Anemoi-Reflective`	**89.050**	**431.215**	**98.880**
`gpt-5-mini`	`Anemoi-Direct`	77.110	882.658	97.380
`gpt-5-mini`	`Text Only LLM`	16.920	5474.230	96.620
`gemini-2.5-flash`	`Anemoi-Reflective`	66.670	1336.040	94.750
`gemini-2.5-flash`	`Anemoi-Direct`	**75.620**	**1310.790**	**98.880**
`gemini-2.5-flash`	`Text Only LLM`	9.450	2336.640	52.250
`Qwen3-Coder-30B`	`Anemoi-Reflective`	**27.860**	2421.150	**81.750**
`Qwen3-Coder-30B`	`Anemoi-Direct`	15.920	**1389.560**	81.620
`Qwen3-Coder-30B`	`Text Only LLM`	14.430	4909.190	76.880

Table 7: Performance metrics for **Easy** difficulty questions across all models. The data is split into three tables for readability. Lower is better for metrics with (↓), higher is better for (↑). Best results for each LLM family are in bold.

LLM	Model Variant	SAE (Q25) (↓)	SAE (Q50) (↓)	SAE (Q75) (↓)	SAE (Q99) (↓)
gpt-5-mini	Anemoi-Reflective	0.004	0.077	0.259	13.851
gpt-5-mini	Anemoi-Direct	**0.002**	**0.042**	**0.175**	**10.980**
gpt-5-mini	Text Only LLM	0.135	0.544	1.388	59.262
gemini-2.5-flash	Anemoi-Reflective	**0.010**	0.064	0.228	9.635
gemini-2.5-flash	Anemoi-Direct	**0.010**	**0.054**	**0.212**	**9.603**
gemini-2.5-flash	Text Only LLM	0.090	0.703	2.783	99417.800
Qwen3-Coder-30B	Anemoi-Reflective	**0.018**	0.158	**0.692**	2281.640
Qwen3-Coder-30B	Anemoi-Direct	0.026	**0.154**	1.024	**82.795**
Qwen3-Coder-30B	Text Only LLM	0.037	0.394	1.750	31195.900

LLM	Model Variant	AE (Q25) (↓)	AE (Q50) (↓)	AE (Q75) (↓)	AE (Q99) (↓)
gpt-5-mini	Anemoi-Reflective	**0.000**	**18.000**	**42.000**	670.680
gpt-5-mini	Anemoi-Direct	4.500	**18.000**	79.500	**167.940**
gpt-5-mini	Text Only LLM	54.000	87.000	132.000	143831.000
gemini-2.5-flash	Anemoi-Reflective	**0.000**	**6.000**	**30.000**	**131.040**
gemini-2.5-flash	Anemoi-Direct	18.000	54.000	120.000	8.271e+18
gemini-2.5-flash	Text Only LLM	30.000	72.000	126.000	176.760
Qwen3-Coder-30B	Anemoi-Reflective	**6.000**	18.000	36.000	**100.560**
Qwen3-Coder-30B	Anemoi-Direct	12.000	24.000	36.000	146383.000
Qwen3-Coder-30B	Text Only LLM	**6.000**	**12.000**	**30.000**	87.840

LLM	Model Variant	% Valid Outputs (↑)	Boolean F1 (↑)
gpt-5-mini	Anemoi-Reflective	84.750	**0.632**
gpt-5-mini	Anemoi-Direct	87.340	0.545
gpt-5-mini	Text Only LLM	**99.220**	0.296
gemini-2.5-flash	Anemoi-Reflective	96.380	**0.714**
gemini-2.5-flash	Anemoi-Direct	**97.930**	0.711
gemini-2.5-flash	Text Only LLM	81.140	0.087
Qwen3-Coder-30B	Anemoi-Reflective	87.600	0.514
Qwen3-Coder-30B	Anemoi-Direct	**89.150**	**0.600**
Qwen3-Coder-30B	Text Only LLM	84.240	0.240

Table 8: Performance metrics for **Medium** difficulty questions across all models. The data is split into three tables for readability. Lower is better for metrics with (↓), higher is better for (↑). Best results for each LLM family are in bold.

LLM	Model Variant	SAE (Q25) (↓)	SAE (Q50) (↓)	SAE (Q75) (↓)	SAE (Q99) (↓)
gpt-5-mini	Anemoi-Reflective	0.471	1.056	1.538	247.345
gpt-5-mini	Anemoi-Direct	**0.230**	**0.833**	**1.365**	214.499
gpt-5-mini	Text Only LLM	1.214	2.493	5.837	**11.871**
gemini-2.5-flash	Anemoi-Reflective	0.042	0.464	1.149	9912.290
gemini-2.5-flash	Anemoi-Direct	**0.000**	**0.086**	**0.815**	429.637
gemini-2.5-flash	Text Only LLM	1.152	1.552	5.003	**65.061**
Qwen3-Coder-30B	Anemoi-Reflective	0.980	1.350	4.267	196339.000
Qwen3-Coder-30B	Anemoi-Direct	**0.954**	**1.349**	**3.465**	133072.000
Qwen3-Coder-30B	Text Only LLM	1.311	3.440	6.986	**247.793**

LLM	Model Variant	EMD (km) (↓)	% Valid Outputs (↑)	Boolean F1 (↑)
gpt-5-mini	Anemoi-Reflective	**3856.950**	**93.780**	0.505
gpt-5-mini	Anemoi-Direct	3922.410	91.080	**0.600**
gpt-5-mini	Text Only LLM	8523.200	85.680	0.388
gemini-2.5-flash	Anemoi-Reflective	**3052.840**	74.900	**0.636**
gemini-2.5-flash	Anemoi-Direct	3363.630	**89.630**	0.545
gemini-2.5-flash	Text Only LLM	9802.010	82.990	0.259
Qwen3-Coder-30B	Anemoi-Reflective	4472.600	82.570	**0.473**
Qwen3-Coder-30B	Anemoi-Direct	**4352.850**	85.680	0.447
Qwen3-Coder-30B	Text Only LLM	9738.010	**91.290**	0.436

Table 9: Performance metrics for **Hard** difficulty questions across all models. The data is split into two tables for readability. Lower is better for metrics with (↓), higher is better for (↑). Best results for each LLM family are in bold.

LLM	Model Variant	EMD (km) (↓)	Extreme Weather F1 (↑)
gpt-5-mini	Anemoi-Reflective	8729.210	0.432
gpt-5-mini	Anemoi-Direct	8130.290	**0.466**
gpt-5-mini	Text Only LLM	**7661.750**	0.421
gemini-2.5-flash	Anemoi-Reflective	7598.740	0.382
gemini-2.5-flash	Anemoi-Direct	**6855.860**	**0.425**
gemini-2.5-flash	Text Only LLM	8226.770	0.247
Qwen3-Coder-30B	Anemoi-Reflective	8519.790	0.292
Qwen3-Coder-30B	Anemoi-Direct	9009.340	0.260
Qwen3-Coder-30B	Text Only LLM	**7621.380**	**0.586**

LLM	Model Variant	% Valid Outputs (↑)	Discussion Score (↑)
gpt-5-mini	Anemoi-Reflective	98.220	**0.264**
gpt-5-mini	Anemoi-Direct	**100.000**	0.255
gpt-5-mini	Text Only LLM	**100.000**	0.238
gemini-2.5-flash	Anemoi-Reflective	88.550	0.275
gemini-2.5-flash	Anemoi-Direct	**100.000**	0.235
gemini-2.5-flash	Text Only LLM	**100.000**	**0.383**
Qwen3-Coder-30B	Anemoi-Reflective	94.400	0.293
Qwen3-Coder-30B	Anemoi-Direct	**100.000**	0.231
Qwen3-Coder-30B	Text Only LLM	**100.000**	**0.344**

Table 10: Performance metrics for **Very Hard** difficulty questions across all models. The data is split into two tables for readability. Lower is better for metrics with (↓), higher is better for (↑). Best results for each LLM family are in bold.