ZEPHYRUS-R1: AGENTIC REINFORCEMENT LEARNING FOR WEATHER REPORTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has emerged as a key method for training AI models in scientific applications, but RL remains difficult when outputs cannot easily be verified against a known ground-truth answer. Natural-language weather forecast reports provide one such valuable domain example: these reports translate atmospheric data into assessments of weather evolution, hazards, and uncertainty, and they are valuable both to the general public and key decision makers. Unlike numerical weather-analysis questions, they admit many valid outputs, and deterministic verification is inherently limited due to the open-ended, natural-language setting. To solve these challenges, we develop **Zephyrus-R1**, a practical framework for RL on scientific weather forecast reporting. We focus on three key components of the agentic RL pipeline: reward design, curriculum learning, and an analysis of human-expert alignment. To encourage both report legibility and factual accuracy, we design a reward function that combines a meteorological rubric with bidirectional claim entailment, both evaluated against National Weather Service reference reports. Our framework optimizes this reward in a simulated tool-calling environment, enabling vision-language agents to gather and analyze forecast evidence across regions and lead times before producing a report. We use annotations from three meteorologists to audit the claim labels produced by our LLM judges. Finally, we introduce a curriculum learning process to help small models first develop tool use and scientific reasoning through RL on verifiable weather-analysis questions, then optimize the open-ended report reward directly. After training Qwen3.5-4B with **Zephyrus-R1**, we improve the mean across six judge–metric scores by 26% over the base model and achieve results competitive with frontier open-source models. Meanwhile, after training Qwen3.8-27B, we improve the same mean by 13% over the base model, by 6.7% over Kimi-K3, and by 2.7% over GPT-5.6-Luna.

1 INTRODUCTION

Reinforcement learning (RL) has driven remarkable progress in Large Language Models (LLMs) for mathematics, programming, and tool-assisted reasoning (Shao et al., 2024; Guo et al., 2025; Jin et al., 2025; Feng et al., 2025; Zheng et al., 2025). A key advantage of these domains is that success is often easy to verify: a proof is correct, a program passes its tests, or a numerical response is near its desired value. Many important scientific tasks, however, do not admit such straightforward evaluation. Even when model outputs can be grounded in scientific evidence, there may be no unique correct way to obtain a useful natural-language conclusion from it. This raises a fundamental challenge: how do we use RL to optimize a scientific reasoning task where the objective itself is ambiguous?

Operational forecast discussions, or weather reports, provide an example of one such challenging domain. These reports convert numerical weather forecasts into actionable, legible reports designed for human use. They are relevant to a broad set of readers, including public decision makers, business owners, and the general public. Producing a good report goes beyond making numerical predictions and demands synoptic (large-scale) weather pattern analysis, prioritization of reported phenomena, hazard identification, and communication of uncertainty. Different reports may produce different but equally useful summaries, while a fluent report can remain superficially plausible despite missing the most important event or mischaracterizing its cause. Due to the subjectivity involved, these tasks have traditionally been the domain of expert meteorologists. Thus, a core difficulty is not merely generating true meteorological text, but learning what constitutes a good operational report.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

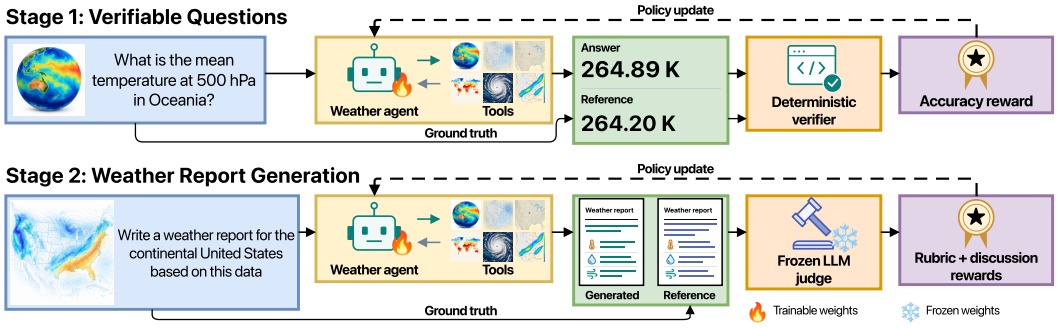

Figure 1: Zephyrus-R1 framework. Agents call weather tools, inspect returned data or errors, and produce answers or reports. Stage 1 rewards verified answers; Stage 2 scores reports with a frozen rubric and claim-entailment judge. Curriculum RL transfers the Stage-1 policy to report generation.

Recent work has explored the use of LLMs for meteorological reporting. Lawson et al. (2025) study multimodal interpretation of weather imagery, Zheng et al. (2026) instruction-tune a multimodal weather-report model, and Higgins et al. (2026) introduce a benchmark and domain-specific evaluation for synoptic reporting. Most closely related, Singh et al. (2026) use RL to generate forecast discussions from structured WeatherNext 2 inputs (Alet et al., 2026) with verifiable rewards for numerical accuracy, synoptic consistency, and formatting. To our knowledge, our work is the first to tackle weather reporting as an open-ended, agentic reinforcement learning problem.

We develop **Zephyrus-R1**, an agentic framework for weather-report generation in which a vision-language model acquires evidence through tools, synthesizes information across regions and lead times, and optimizes an open-ended report-level objective. The agent can inspect atmospheric data, forecasts, and plots before producing its report, which is scored using an LLM-evaluated six-criterion meteorological rubric and bidirectional claim entailment against National Weather Service reference reports; we audit the entailment judges' claim-label decisions against annotations from expert meteorologists. Because report generation simultaneously demands tool use, scientific reasoning, and open-ended synthesis, we train small models with a two-stage curriculum: RL first develops tool use and reasoning on weather-analysis questions with verifiable answers, then optimizes the report reward. This curriculum substantially improves Qwen3.5-4B's tool-use capabilities (Figure 3) and improves F1 scores over the base model by 67% under the training LLM judge and by 24% and 60% under two independent judges. On Qwen3.8-27B, curriculum RL achieves competitive state-of-the-art performance, outperforming Kimi-K3 on all metric–judge combinations and GPT-5.6-Luna on four of six combinations and in average overall score. Together, these results demonstrate that RL can improve open-ended scientific reporting when reward design, evidence acquisition, and training curriculum are designed jointly. Our contributions are the following:

- **Framework.** We introduce Zephyrus-R1, an RL framework for multi-turn, tool-using weather agents that combines weather-tool interaction, server-verifiable evidence acquisition, and judge-scored optimization of open-ended reports (Section 3).
- **Curriculum learning.** We train tool use and analytical reasoning on verifiable weather-analysis questions, and study the effect of this curriculum-based approach compared to directly training on report generation (Section 3.4).
- **Reward design.** We introduce a report-level reward that combines a meteorological rubric with bidirectional claim entailment to capture both report quality and factual coverage against a natural-language reference report (Section 3.2).
- **Domain expert evaluation.** Using annotations from three professional meteorologists, we compare expert and LLM judgments. We find that weather-claim judgment is non-trivial and that LLM–expert agreement is comparable to inter-expert agreement (Section 3.3).

2 Related Work

Agents for science and weather. LLM agents that interleave reasoning with tool calls (Yao et al., 2023; Shinn et al., 2023; Schick et al., 2023) have been applied across the sciences (Bran et al.,

2024; Boiko et al., 2023; Huang et al., 2025). In weather science, ZEPHYRUS provides a Python environment over WeatherBench 2 with a neural forecaster as a tool and a 49-task benchmark (Varambally et al., 2026), while SYNOPTICBENCH pairs forecast images with forecast discussions for VLM evaluation (Higgins et al., 2026). Most directly, AFDBENCH pairs structured WeatherNext 2 inputs with operational discussions and applies GRPO with domain-specific rewards for temperature accuracy, synoptic correctness, and format compliance (Singh et al., 2026). ZEPHYRUSBENCH presents weather-science tasks as natural-language question–answer items, spanning data retrieval and computation, multi-step analysis, and forecast-report generation (Varambally et al., 2026). Stage 1 uses its questions with machine-checkable answers; we refer to these as verifiable weather-analysis questions. Building on the ZEPHYRUS environment, our policy acquires analysis and forecast fields through multi-turn tool use and transfers from these questions to report generation.

RL for tool use and rewards for open-ended generation. Group-relative policy optimization and related methods are widely used with verifiable rewards on tasks with checkable answers (Shao et al., 2024; Yu et al., 2025; Guo et al., 2025; MiniMax, 2025), and extensions to multi-turn tool use train the policy on its own execution traces with outcome rewards (Jin et al., 2025; Feng et al., 2025; Zheng et al., 2025; Li et al., 2025). Reward hacking and collapse of tool use under weak supervision have been reported (Thaman, 2026; Hao et al., 2026). For open-ended text, learned reward models and LLM judges (Ouyang et al., 2022; Zheng et al., 2023) are made more targeted by rubrics (Gunjal et al., 2025; Viswanathan et al., 2025), reference-derived chains (Jiang et al., 2026), mined rubrics (Wei et al., 2026), and hybrid verifier-plus-reward-model rewards (Tao et al., 2025). Our report stage uses a frozen judge with fixed rubric and claim-level metrics, while the environment separately enforces the minimum tool trajectory needed for a weather report.

Curriculum learning. Ordering training from easy to hard (Bengio et al., 2009; Narvekar et al., 2020) improves RL fine-tuning of LLM reasoning when difficulty is staged or adapted, with evidence on mathematics, planning, inductive reasoning, and other reasoning domains (Shi et al., 2025; Chen et al., 2025; Parashar et al., 2025). These curricula span verifiable problem categories and domains. We study transfer from deterministically scored analysis to judge-scored reports, comparing the complete recipe with direct report RL (Section 4.2).

3 THE ZEPHYRUS-R1 FRAMEWORK

Our goal is to train a vision-language model-based agent to write accurate multi-day forecast discussions, given the state of the atmospheric weather conditions. To this end, we design an LLM-as-a-judge reward, assess its agreement with human expert judgments, and optimize the agent using a two-stage curriculum learning framework.

3.1 SETUP

Problem setting. We consider the task of generating multi-day operational forecast discussions from atmospheric weather data. Given access to the current atmospheric state and its evolution over the forecast horizon, the goal is to produce a discussion that summarizes the major weather patterns, their timing and location, and associated hazards. In this work, we focus on forecasts for the continental United States (CONUS).

To evaluate agents on this task and improve their performance, we follow the setup of Varambally et al. (2026). We pair historical ERA5 reanalysis fields from WeatherBench 2 (Rasp et al., 2024; Hersbach et al., 2020) with operational forecast discussions[1] issued by the National Weather Service (NWS) on the corresponding dates. These human-written discussions serve as reference reports for training and evaluation.

Agentic Environment. We formulate the task as a tool-using vision-language agent operating over atmospheric data. For each report-generation task, the agent receives the relevant date (excluding the exact year) and a collection of tools exposed through a Python API. It iteratively executes Python code to gather and analyze evidence from the numerical data to produce the forecast discussion. After

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

each execution, the environment returns the resulting outputs and any execution errors, allowing the agent to refine its analysis or repair unsuccessful tool calls. When the agent creates plots, these are additionally returned as image observations, allowing it to reason directly over spatial weather patterns. The interaction continues until the agent produces its final forecast discussion.

Toolset. Our environment builds on the toolset introduced in ZEPHYRUS (Varambally et al., 2026). The agent can query WeatherBench 2 atmospheric fields, obtain geographic masks for regions of interest using the geolocator tool, and access historical climatology for computing anomalies. It also has access to a forecaster that returns atmospheric fields at requested forecast lead times. In this work, we use an "oracle" forecaster that returns the corresponding future ERA5 reanalysis field rather than the output of a numerical weather prediction model. This isolates the agent's ability to analyze atmospheric evolution and generate a forecast discussion from errors introduced by the underlying numerical forecast.

In this work, we augment the environment by adding vision capabilities to the agent. It can now leverage a plotting tool to plot arbitrary images during its code interactions. The execution environment then passes these figures back as image observations, so that the model can inspect a spatial pattern rather than infer it solely from printed arrays or statistics. We also add access to the TempestExtremes tool (Ullrich & Zarzycki, 2017), which allows the agent to detect and track large-scale weather systems such as tropical and extratropical cyclones.

3.2 REWARD DESIGN FOR THE REPORT STAGE

To evaluate the quality of the generated reports and train agents to improve on this task, we need a reward function $Q(y, y^*)$ that measures the quality of a generated report y measured against a reference report y^*. Prior work on weather-report generation has found that standard text metrics such as BLEU and ROUGE are poor measures of forecast content quality, motivating more factual and domain-specific evaluation methods (Higgins et al., 2026; Varambally et al., 2026). We therefore seek to define a Q that rewards factual accuracy while penalizing hallucinations. To achieve this, we use two different LLM-as-a-judge metrics: a rubric score that compares y against y^* holistically, and a Discussion F1 score that measures claim-level factual precision and recall between y and y^*.

Rubric Score. We first construct a meteorological rubric that captures the qualities of a strong forecast discussion. To do so, we generate faithful paraphrases and systematically corrupted variants of human-written reference reports, then use an LLM to iteratively refine criteria that separate the two groups. After review, we freeze a six-criterion rubric covering reference fidelity, spatiotemporal specificity, hazard prioritization, meteorological reasoning, thermal and precipitation-type plausibility, and calibrated use of detail. We then use this fixed rubric to score generated reports. The judge receives the generated report, the corresponding reference report, and the rubric, and assigns a score to each criterion. We average the six scores and normalize the result to $[0, 1]$. The same rubric is used throughout training and evaluation. Appendix C.5 gives the full rubric and scoring anchors, while Appendix F.2 gives the judge prompt.

F1 Score. We also measure factual agreement at the level of individual claims. The judge LLM extracts atomic claims from both the generated and reference reports and labels each as *supported*, *refuted*, or *neutral* with respect to the other report. We define $P(y) = \dfrac{S_y}{S_y + F_y}$, $R(y) = \dfrac{S_{y^*}}{N_{y^*}}$, where S_y and F_y are the numbers of generated claims that are supported and refuted by the reference report, respectively, S_{y^*} is the number of reference claims supported by the generated report, and N_{y^*} is the total number of reference claims. To obtain the Discussion F1 score, we compute a harmonic mean of the precision and recall. This bidirectional claim-level metric follows the ZEPHYRUS (Varambally et al., 2026) "discussion score." Unlike ZEPHYRUS, which uses label probabilities derived from model logits, we use discrete LLM label decisions.

Report Grounding Requirement. To ensure that the model actually examines the atmospheric data rather than hallucinating a plausible report, we require a minimum amount of successful tool use before assigning a judge-based reward. We maintain a binary grounding flag $g(\tau)$, which is set to one only if the agent has read a valid field from the current atmospheric state, obtained valid fields at a forecast lead, and received the resulting evidence in its context. Reports submitted before these conditions are satisfied are rejected, and the interaction continues. If $g(\tau) = 0$ at the final turn,

the rollout receives a fixed penalty, and the judge scores do not contribute to the training reward. Appendix C.2 provides the implementation details.

Judge-scored quality and terminal reward. We combine the rubric and F1 scores as

$$Q_t(y) = w_t \,\mathrm{rubric}(y) + (1 - w_t)\, F1(y), \tag{1}$$

where w_t controls the relative contribution of the two metrics. We update w_t during training based on the variation of each score within the rollout group, so that neither metric dominates the reward. Appendix C.1 gives the weighting rule. The final reward for training is

$$R(\tau) = \begin{cases} -c, & \text{no usable report or } g(\tau) = 0, \\ Q_t(y) - p_{\text{fmt}}, & \text{otherwise}, \end{cases} \tag{2}$$

where c is a fixed penalty and p_{fmt} penalizes a report recovered from an untagged forced-final reply.

3.3 Human-expert audit of LLM judgment

In order to analyze the accuracy of our LLM-as-judge metrics against the judgments of human domain experts, we asked three meteorologists to annotate 100 claims each, using the same Supported/Neutral/Refuted definitions as given to our LLM judge. To create the claim set, we sample 5 report-writing tasks at random, uniformly sample from base-model LLMs as report generators (`Qwen3.5-4B`, `Qwen3.8-27B`, `GPT-5.6-Luna`), and gather 2 groups for each task: generated claims scored against a reference report, and reference claims scored against a generated report. The experts were not informed which reports were reference vs. generated, nor which generating model produced each claim.

Table 1: Claim-label agreement on 100 claim–report pairs. Accuracy (Acc.) is exact S/R/N agreement; within-1 excludes only Supported–Refuted reversals. For majority-vote comparisons, S/R/N ties are assigned NEUTRAL.

Comparison	Acc.	κ	Within-1
GPT-OSS–Expert$_1$	0.370	0.097	0.920
GPT-OSS–Expert$_2$	0.630	0.346	**0.960**
GPT-OSS–Expert$_3$	0.570	0.226	**0.960**
Expert$_1$–Expert$_2$	0.480	0.224	0.900
Expert$_1$–Expert$_3$	0.460	0.214	0.860
Expert$_2$–Expert$_3$	0.570	0.306	0.890
GPT-OSS–Expert maj.	0.630	0.327	**0.960**
Gemma–Expert maj.	0.570	0.289	0.920
Mistral–Expert maj.	0.660	**0.427**	0.930
LLM maj.–Expert maj.	**0.670**	0.423	0.940

Across the 100 claims, we generate Supported/Refuted/Neutral labels from our three LLM judges (GPT-OSS-120B, Gemma-3-27B-IT and Mistral-Small-3.2-24B). We then compare the LLM labels against the human-expert judgments, and the human experts between each other. Table 1 shows these results. We define Accuracy and Cohen's κ by using the S/R/N judgments as categorical labels and checking for exact matches, and we calculate 'Within-1' accuracy based on whether two judgments are not opposite for the claim (i.e. one Supported, one Refuted). For majority-vote rows, we use the majority of three voters and assign S/R/N ties to Neutral.

Our key finding is that this report claim annotation is a difficult task even for domain experts, with significant inter-expert disagreement. Observe that `GPT-OSS` has comparable agreement with experts as the experts do among themselves. In all comparisons, the LLM judges have a 'within-1' accuracy exceeding 0.9, showing that almost all label disagreements are not full semantic flips from Supported to Refuted or vice-versa relative to expert judgment. Finally, we observe that accuracy is maximized by comparing three-judge LLM majority vote against human expert majority vote, suggesting that aggregating judgments helps reduce annotator-specific disagreement.

We use `GPT-OSS` as the RL training judge. On this same claim set, it is also the harshest judge, with a mean score of 0.05, relative to Mistral's mean score of 0.29 and Gemma's of 0.57. These scores are calculated by counting Supported as $+1$, Neutral as 0, and Refuted as -1. Its conservative labels provide a stringent training signal, while the independent judges test whether improvements transfer beyond the training judge.

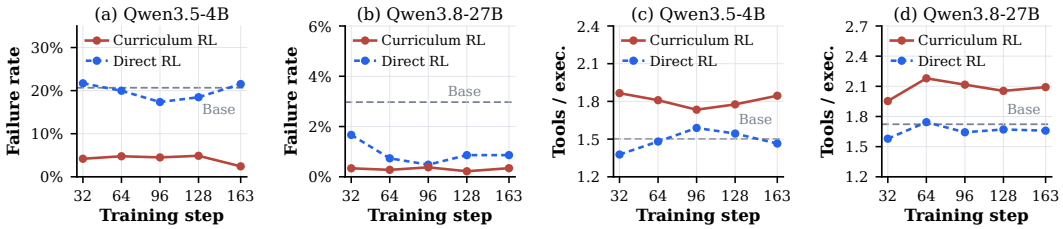

Figure 3: Tool-use diagnostics on the report-stage test set during report RL. Failure rate is shown for 4B (a) and 27B (b), and Tools/exec. for 4B (c) and 27B (d).

3.4 Curriculum Reinforcement Learning

Direct report training must jointly learn tool use, weather analysis, and report composition from one scalar reward. We therefore use a two-stage curriculum: Stage 1 trains tool use on verifiable weather-analysis questions; Stage 2 optimizes report generation with the reward from Section 3.2.

3.4.1 Stage 1: Fine-tuning on verifiable weather-analysis questions

Accuracy reward. Stage 1 uses eight easy, machine-verifiable question templates from Zephyrus-Bench (Varambally et al., 2026), which are listed in Appendix B.2. The questions ask the agent to compute regional statistics, identify spatial extrema and their timing, and detect climatological anomalies. Gold answers come from the benchmark metadata. Each question also specifies a required tool, which is the dataset for regional statistics and extrema and the climatology tool for anomaly questions. The verifier gives graded credit for numerical, location, time, and anomaly answers, as detailed in Appendix B.2. The reward is

$$r = s(\text{answer}, \text{gold}) \cdot \mathbf{1}[\text{required tool accessed}] - \rho(\text{turns}), \tag{3}$$

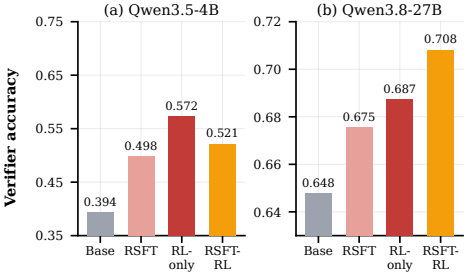

Figure 2: Verifier accuracy on held-out validation weather-analysis questions for the base policy and the RSFT, RL-only, and RSFT-RL fine-tuned policies: (a) Qwen3.5-4B; (b) Qwen3.8-27B.

where $s \in [0, 1]$ is the graded verifier and ρ applies a fixed deduction per turn beyond a turn allowance, up to a cap (Appendix B.2). An access counter checks the required tool.

Fine-tuning methods. Rejection-sampling fine-tuning (RSFT) retains full-credit, required-tool-access trajectories that pass the infrastructure and output-format checks specified in Appendix B.1. That ranking retains at most two trajectories per question. Stage-1 RL optimizes Equation (3), starting from the RSFT checkpoint (RSFT-RL) or the base policy (RL-only). All three methods raise verifier accuracy over the base policy at both scales (Figure 2); the best checkpoints, RL-only for Qwen3.5-4B (0.572) and RSFT-RL for Qwen3.8-27B (0.708), initialize Stage 2. Because credit requires accessing the question's required tool, these gains suggest improved tool use and reasoning over the returned data. The questions also exercise component skills of report generation: quantifying field values, locating features in space, timing their evolution, and identifying anomalies relative to climatology. Stage 1 can therefore provide a stronger starting point and tool-use behavior for the subsequent report RL. Appendix B gives the schedule, selected checkpoints, and results by type.

3.4.2 Stage 2: Reinforcement learning on the report generation task

Stage 2 trains on the report task with the reward of Section 3.2 (Equation 2). Beyond the Stage-1 skills, it requires forecast-tool use, hazard prioritization, and report composition. Curriculum RL initializes the policy from the selected Stage-1 checkpoint and resets the optimizer. On the other hand, Direct RL starts from the base policy and uses the same report data, prompt, reward rule, judge definitions, sampler, and report-stage length. The two approaches therefore differ only in their initialization and the additional Stage-1 training used by Curriculum RL.

Table 2: Weather-report results on the test events for the base policies and two Zephyrus-R1 training routes. Every score averages three complete judge passes; higher is better for both metrics, and bold marks the within-scale best. Green percentages give the relative change from the same-scale Base, computed from unrounded scores.

		GPT-OSS 120B		Gemma-3 27B-IT		Mistral-Small 3.2-24B		
Model	**Route**	**Rub.**	**F1**	**Rub.**	**F1**	**Rub.**	**F1**	**Avg**
Qwen3.5-4B	Base	0.082	0.138	0.507	0.621	0.391	0.342	0.347
	DIRECT RL	0.123 +51%	0.217 +57%	**0.575** +13%	0.693 +12%	**0.453** +16%	0.477 +40%	0.423 +22%
	CURRICULUM RL	**0.129** +57%	**0.231** +67%	0.514 +1%	**0.768** +24%	0.441 +13%	**0.548** +60%	**0.438** +26%
Qwen3.8-27B	Base	0.136	0.224	0.735	0.715	0.690	0.537	0.506
	DIRECT RL	0.167 +23%	**0.318** +42%	0.742 +1%	**0.770** +8%	0.719 +4%	**0.631** +18%	0.558 +10%
	CURRICULUM RL	**0.260** +91%	0.312 +39%	**0.778** +6%	0.736 +3%	**0.728** +6%	0.615 +15%	**0.571** +13%

Stage 1 reduces execution failures and broadens tool use. Every Stage-2 report must first meet the report grounding requirement, which requires working analysis and forecast calls. This overlaps with Stage 1 in requiring successful data access, while forecast acquisition is specific to the report stage. We therefore compare the two approaches on two execution-level diagnostics: the failure rate, the fraction of executions in which attempted tool use returns no valid result, and Tools/exec., the number of distinct tool families returning valid results per execution. At every evaluated report-RL checkpoint and at both scales, CURRICULUM RL has the lower failure rate and the higher Tools/exec. (Figure 3). At 4B, the terminal failure rate falls from 21.5% to 2.4%; at 27B, both initialization methods are reliable (0.9% versus 0.3%), and Tools/exec. rises from 1.66 to 2.09. Curriculum RL thus exhibits more reliable and broader evidence acquisition throughout report RL.

3.5 OPTIMIZATION

Both stages use group-relative policy optimization (Shao et al., 2024) with the clipped importance-sampling policy optimization (CISPO) objective (MiniMax, 2025). We only update the language-block layers using LoRA (Hu et al., 2022) adapters while the vision tower remains frozen. Each rollout is a single sequence containing the full interaction history, with its advantage computed relative to other rollouts in the group. Appendix C.6 specifies the loss and failure masks.

4 EXPERIMENTAL RESULTS

4.1 SETUP

Data. Our reference reports are forecast discussions for the continental United States issued by the NOAA National Weather Service, the same source used by the report task in ZEPHYRUS-BENCH (Varambally et al., 2026). Because these discussions are written by operational forecasters, they provide expert-quality targets without any manual annotation. For report RL, we train on 2,608 events from 1995-2018 and evaluate on a test set of 197 temporally disjoint events from 2019-2020. Appendix A gives field coverage, report preprocessing, and split-integrity checks. Stage-1 uses eight easy, automatically verifiable question templates from ZEPHYRUSBENCH. They cover regional statistics, geographic and temporal extrema, and one-sided, standardized, and percentile-envelope climatological anomaly detection; Appendix B.2 lists the individual templates. Stage-1 RL consumes 704 question draws from a 710-question training pool and uses a disjoint held-out validation set of 125 questions for checkpoint selection.

Metrics and judges. We report the rubric and bidirectional claim-entailment F1, as defined in Section 3.2. GPT-OSS-120B is the training judge, while Gemma-3-27B-IT and Mistral-Small-3.2-24B are independent judges, extracting and labeling their own claims using the fixed prompts and definitions in Appendix F.3 (Agarwal et al., 2025; Gemma Team et al., 2025; Mistral AI, 2025). Every report-quality score in Tables 2–3 averages three LLM judgments over the same frozen outputs.

For each execution request, the server trace records `tools_ok`, the distinct tool families that produced valid returns. Tools/exec. counts these families, while failure rate is the fraction of requests

Table 3: Leaderboard on the test events, ordered by Avg (higher is better). Every score averages three complete judge passes. Shaded rows are Zephyrus-R1 models trained with Curriculum RL. Avg is the unweighted mean of the six unrounded scores.

		GPT-OSS 120B		Gemma-3 27B-IT		Mistral-Small 3.2-24B		
#	Model	Rub.	F1	Rub.	F1	Rub.	F1	Avg
1	Qwen3.8-27B + Zephyrus-R1	**0.260**	**0.312**	**0.778**	0.736	**0.728**	0.615	**0.571**
2	GPT-5.6-Luna	0.170	0.297	0.713	**0.801**	0.698	**0.660**	0.556
3	Kimi-K3	0.168	0.286	0.736	0.732	0.721	0.570	0.536
4	DeepSeek-V4-Flash-Vision-Exp	0.153	0.262	0.706	0.729	0.689	0.570	0.518
5	GLM-5.3-Flash	0.156	0.224	0.728	0.726	0.699	0.567	0.517
6	MiniMax-M3	0.139	0.241	0.709	0.716	0.682	0.572	0.510
7	Qwen3.6-35B-A3B	0.126	0.197	0.619	0.708	0.530	0.481	0.444
8	Qwen3.5-122B-A10B	0.107	0.199	0.584	0.736	0.525	0.500	0.442
9	Qwen3.5-4B + Zephyrus-R1	0.129	0.231	0.514	0.768	0.441	0.548	0.438
10	Gemma-4-31B-IT	0.136	0.190	0.558	0.702	0.490	0.465	0.424
11	Qwen3.5-9B	0.099	0.155	0.525	0.670	0.426	0.424	0.383

with an empty `tools_ok`. We first average each measure within an episode and then over all 197 episode means, so episodes with more requests do not receive extra weight; an episode with no execution request receives a failure rate of one. We use both as process diagnostics in Section 3.4.2.

Systems. For both Qwen3.5-4B and Qwen3.8-27B, we compare the base model with DIRECT RL and CURRICULUM RL under the same report data, prompt, reward, sampler, and report-stage length. The leaderboard includes GPT-5.6-Luna and the eight open-weight agents listed in Table 3; each LLM produces one report per evaluation task (Yang et al., 2025; Qwen Team, 2026a;c; OpenAI, 2026; Kimi Team et al., 2026; GLM-5-Team et al., 2026; DeepSeek-AI, 2026; Lai et al., 2026; Gemma Team et al., 2026; Qwen Team, 2026b).

Training setup. The RL phase in each stage uses CISPO with group-centered advantages, 16 prompts and four rollouts per step, temperature 0.6, zero KL regularization, and rank-64 language-layer LoRA with the vision tower frozen (Hu et al., 2022). Stage-1 RL uses a 44-step, 13-turn schedule; report RL uses a single epoch, i.e. a 163-step, 13-turn schedule and the online-balanced reward. Report-stage learning rates are 10^{-5} for 4B and 3×10^{-5} for 27B. The curriculum routes start report RL from 4B RL-only step 44 and 27B RSFT-RL step 24. Appendix B details curriculum training, and Appendix C gives the report-stage settings, hardware, and checkpoint protocol.

4.2 MAIN RESULTS

Zephyrus-R1 improves report quality under all three judges. At 4B, CURRICULUM RL improves every score over the base policy (Table 2). F1 rises by 67%, 24%, and 60% under GPT-OSS, Gemma, and Mistral, while rubric score rises by 57%, 1%, and 13%. DIRECT RL also improves all six scores, confirming that report RL is effective without the curriculum. CURRICULUM RL yields higher F1 than DIRECT RL under every judge, showing that Stage 1 improves claim-level agreement at this scale. At 27B, CURRICULUM RL improves F1 by 39%, 3%, and 15% and rubric score by 91%, 6%, and 6%. Both routes again outperform the base on all scores. Relative to DIRECT RL, CURRICULUM RL achieves higher rubric scores under every judge but slightly lower F1. Thus, Stage 1 provides a consistent F1 advantage at 4B and a consistent rubric-quality advantage at 27B.

The trained agent is competitive with larger models. The 27B Zephyrus-R1 agent ranks first by the unweighted average of the six judge scores, at 0.571 versus 0.556 for GPT-5.6-Luna and 0.536 for Kimi-K3 (Table 3). It leads four of the six component columns, while GPT-5.6-Luna retains the highest Gemma and Mistral F1 scores. The 4B agent ranks ninth by the aggregate, but its 0.438 average is close to Qwen3.5-122B-A10B (0.442) and Qwen3.6-35B-A3B (0.444), and its F1 score exceeds both larger Qwen models under every judge. These comparisons establish competitiveness on this task rather than general model superiority because architectures and decoding settings differ.

Reference report (excerpt)

[. . .] **[S3]** Moisture from the Gulf of Mexico will cause widespread rain and thunderstorms along and ahead of the front. Severe weather and heavy rain causing flash flooding are both possible. [. . .]
[R1] The severe weather threat is forecast farther south across the Southeast and Southern Appalachians. [. . .]
[S1] A cold front coming into the Northern Plains late Friday will progress into the Upper Midwest on Saturday, causing showers along it. **[S2]** Temperatures are expected to cool below average in the Northern Rockies and Plains with this front, which could lead to light snow in those areas on Saturday. [. . .]

Curriculum RL report (excerpt)

[. . .] **[S1]** Over the next 24–48 hours the upper trough and its surface low will track east across the northern Plains and into the Upper Midwest and Great Lakes, bringing a broad band of showers and storms to the northern Plains, the Dakotas, and the Lakes through Friday into Saturday. **[S2]** Cooler air behind the front will cap daytime highs and allow snow showers along the northern Rockies and higher terrain where 850 mb temperatures remain below freezing. **[S3]** Meanwhile the southern Plains low and front will persist into the weekend, maintaining a risk of scattered to frequent thunderstorms with locally heavy rain, flash-flood potential, and the threat of severe weather given the warm, moist airmass and strong mid-level flow.
[R1] The Southeast and much of the South remain under the eastern flank of a 500 mb ridge, where conditions stay warm, stable, and largely dry, with 2 m temperatures in the low to mid 30s °C and only isolated afternoon activity. [. . .]

Figure 4: Terminal Qwen3.8-27B CURRICULUM RL report with author-mapped GPT-OSS supported (**S**) and refuted (**R**) spans.

4.3 QUALITATIVE REPORT COMPARISON

Figure 4 shows an excerpt from the terminal step-163 27B curriculum-trained checkpoint. The report correctly tracks the northern trough and surface low into the Upper Midwest and Great Lakes, links the trailing cold air to high-terrain snow, and identifies the southern-Plains severe-weather and flash-flood threat. Appendix C.3 traces the tool-based evidence behind this account. The report also makes a consequential regional error by describing the Southeast and much of the South as stable and largely dry despite the reference forecast of eastern-U.S. rainfall and a Southeast severe-weather threat. The refuted span therefore reflects a hazard-interpretation failure rather than weak prose. Overall, the policy produces a coherent broad-scale account but can still introduce a locally important contradiction. Appendix E shows how all three judges extract and label claims from the same report.

5 DISCUSSION

In this work, we tackled the problem of training LLM-based agents for weather report generation. We designed two LLM-as-a-judge based reward functions, and examined the alignment of their judgment against human expert judgment. Through a two-stage curriculum RL approach, our method transfers skills learned on simple, verifiable weather-analysis questions to judge-scored weather-report generation. On Qwen3.5-4B, CURRICULUM RL improves F1 over the base policy and DIRECT RL under all three judges at the final checkpoint. At the terminal 27B checkpoint, CURRICULUM RL improves every rubric and F1 score over the base policy. More broadly, our results suggest that non-verifiable scientific tasks can still benefit from RL when careful reward design is combined with curricula that bootstrap from verifiable intermediate skills.

Limitations. Our experiments focus on the continental United States because of data availability, although the framework itself is not location-specific. Our evaluation also relies on automated LLM judges, whose absolute scores and model rankings vary, and whose agreement with human experts is imperfect. At the same time, claim-level evaluation is itself challenging for domain experts, as reflected by the substantial inter-rater disagreement (Appendix D.3). Finally, we do not evaluate expert preference over complete generated reports. Incorporating such end-to-end human evaluation is an important direction for future work.

Single runs do not quantify optimization variance, and curriculum gains may reflect added compute as well as initialization. External model differences, oracle inputs, and one CONUS task limit claims about operational skill and transfer.

AI USE STATEMENT

Generative AI tools were used to polish the language and grammar of the manuscript; assist with code implementation, experiment-log inspection, and debugging; generate paraphrased and corrupted examples for rubric construction and iteratively refine the rubric; and summarize logged trajectories for presentation in the appendix. LLMs were also used as experimental judge models for report and claim evaluation, as described in the paper. All AI-assisted outputs, experimental claims, numbers, and citations were reviewed and verified by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

The weather data are public reanalysis products. The reference reports are public operational forecast discussions. Three collaborating meteorologists, who are also co-authors of this paper, provided claim-label annotations for evaluator auditing (Section 3.3; Appendix D.3). The trained models require expert review before operational use; evaluation assesses reports with oracle inputs.

REPRODUCIBILITY STATEMENT

Training and evaluation parameters are summarized in Table B.1, the Stage-1 verifiers in Appendix B.2, and the rubric definitions in Appendix C.5; Appendix C.5 reproduces the complete scoring anchors. Appendix F documents the report-generation and evaluation prompt templates; full prompt files will be released with the code. The optimization configuration is given in Section 4.1 and Appendix C.6. Code, adapters, reports, and judge outputs will be released at camera-ready.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. arXiv preprint arXiv:2508.10925, 2025.

Ferran Alet, Tom R. Andersson, Ilan Price, Stratis Markou, Andrew El-Kadi, Dominic Masters, Amy Li, Samier Merchant, Natalie Williams, Gregory Thornton, Ken MacKay, Olivia Graham, Akib Uddin, Ben Gaiarin, Devaja Shah, Elinor Kruse, Wallace Hogsett, David Zelinsky, John Cangialosi, Jonathan Martinez, James Franklin, Mark DeMaria, Kate Musgrave, Caroline L. Bain, Helen Titley, Jacklynn Stott, Remi Lam, Aaron Bell, Paul Komarek, Matthew Willson, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Operational tropical cyclone forecasting with ai. *Nature*, 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10953-2. URL https://doi.org/10.1038/s41586-026-10953-2.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

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

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for LLM reasoning. arXiv preprint arXiv:2505.14970, 2025.

DeepSeek-AI. DeepSeek-V4-Flash-Vision-Exp release: Multimodal API now live. https://api-docs.deepseek.com/news/news260821/, August 2026.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. ReTool: Reinforcement learning for strategic tool use in LLMs. arXiv preprint arXiv:2504.11536, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, et al. Gemma 3 technical report. arXiv preprint arXiv:2503.19786, 2025.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, et al. Gemma 4 technical report. arXiv preprint arXiv:2607.02770, 2026.

GLM-5-Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. GLM-5: from vibe coding to agentic engineering. arXiv preprint arXiv:2602.15763, 2026.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. arXiv preprint arXiv:2507.17746, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Yupu Hao, Zhuoran Jin, Huanxuan Liao, Kang Liu, and Jun Zhao. Why multi-step tool-use reinforcement learning collapses and how supervisory signals fix it. arXiv preprint arXiv:2606.26027, 2026.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

Timothy B. Higgins, Antonios Mamalakis, and Chirag Agarwal. SynopticBench: Evaluating vision-language models on generating weather forecast discussions of the future. arXiv preprint arXiv:2604.16451, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan Li, Lin Qiu, Gavin Li, Junze Zhang, Di Yin, Shruti Marwaha, Jennefer N. Carter, Xin Zhou, Matthew Wheeler, Jonathan A. Bernstein, Mengdi Wang, Peng He, Jingtian Zhou, Michael Snyder, Le Cong, Aviv Regev, and Jure Leskovec. Biomni: A general-purpose biomedical AI agent. bioRxiv preprint, 2025.

Yuxin Jiang, Yufei Wang, Qiyuan Zhang, Xingshan Zeng, Liangyou Li, Jierun Chen, Chaofan Tao, Haoli Bai, and Lifeng Shang. From verifiable dot to reward chain: Harnessing verifiable reference-based rewards for reinforcement learning of open-ended generation. *arXiv preprint arXiv:2601.18533*, 2026.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. arXiv preprint arXiv:2503.09516, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C., Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, et al. Kimi K3: Open frontier intelligence. arXiv preprint arXiv:2607.24653, 2026.

Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, Jinkai Hu, Jiayao Li, Rui Gao, Zekun Li, Songquan Zhu, Jingkai Zhou, and Pengyu Zhao. MiniMax sparse attention. arXiv preprint arXiv:2606.13392, 2026.

John R Lawson, Joseph E Trujillo-Falcón, David M Schultz, Montgomery L Flora, Kevin H Goebbert, Seth N Lyman, Corey K Potvin, and Adam J Stepanek. Pixels and predictions: Potential of gpt-4v in meteorological imagery analysis and forecast communication. *Artificial Intelligence for the Earth Systems*, 4(1), 2025.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. WebSailor: Navigating super-human reasoning for web agent. arXiv preprint arXiv:2507.02592, 2025.

MiniMax. MiniMax-M1: Scaling test-time compute efficiently with lightning attention. arXiv preprint arXiv:2506.13585, 2025.

Mistral AI. Mistral Small 3.2 (24B Instruct 2506). https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506, June 2025.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

OpenAI. GPT-5.6 system card. https://deploymentsafety.openai.com/gpt-5-6, 2026.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum reinforcement learning from easy to hard tasks improves LLM reasoning. arXiv preprint arXiv:2506.06632, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026a.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all. https://qwen.ai/blog?id=qwen3.6-35b-a3b, April 2026b.

Qwen Team. Qwen3.8-27B model card. https://huggingface.co/Qwen/Qwen3.8-27B, August 2026c.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. 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.

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. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. arXiv preprint arXiv:2504.05520, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Manmeet Singh, Somnath Luitel, Prabhjot Singh, Manraaj Banga, Naveen Sudharsan, and Josh Durkee. AFDBench: A reasoning-first AI scientist for National Weather Service forecast discussions. arXiv preprint arXiv:2608.24954, 2026.

Leitian Tao, Ilia Kulikov, Swarnadeep Saha, Tianlu Wang, Jing Xu, Sharon Li, Jason E. Weston, and Ping Yu. Hybrid reinforcement: When reward is sparse, it's better to be dense. arXiv preprint arXiv:2510.07242, 2025.

Kunvar Thaman. Reward hacking benchmark: Measuring exploits in LLM agents with tool use. arXiv preprint arXiv:2605.02964, 2026.

Paul A. Ullrich and Colin M. Zarzycki. TempestExtremes: A framework for scale-insensitive pointwise feature tracking on unstructured grids. *Geoscientific Model Development*, 10(3):1069–1090, 2017.

Sumanth Varambally, Marshall Fisher, Jas Thakker, Yiwei Chen, Zhirui Xia, Yasaman Jafari, Ruijia Niu, Manas Jain, Veeramakali Vignesh Manivannan, Zachary Novack, Luyu Han, Srikar Eranky, Salva Rühling Cachay, Taylor Berg-Kirkpatrick, Duncan Watson-Parris, Yian Ma, and Rose Yu. Zephyrus: An agentic framework for weather science. In *The Fourteenth International Conference on Learning Representations*, 2026.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Xiyu Wei, Qingwei Zong, Xiaoguang Li, Eugene Yu, and Sujian Li. QuRL: Rubrics as judge for open-ended question answering. In *The Fourteenth International Conference on Learning Representations*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. arXiv preprint arXiv:2503.14476, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, 2023.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. arXiv preprint arXiv:2504.03160, 2025.

Zinan Zheng, Yang Liu, Nuo Chen, Juepeng Zheng, Hong Cheng, and Jia Li. Weathersyn: An instruction tuning mllm for weather forecasting report generation. *arXiv preprint arXiv:2605.07522*, 2026.

A DATA

Input fields. The WeatherBench 2 fields span 1959 to 2023 at six-hour intervals on a one-degree equiangular grid. Each event contains surface pressure and temperature fields, winds, precipitation, land and ocean fields, and pressure-level geopotential, humidity, temperature, winds, and vertical velocity. The prompt exposes relative time indices and names the calendar date and weekday of the next five days because operational forecast discussions refer to days by name.

Report preprocessing and split checks. A fixed prompt removes author notes, map and model references, and disclaimers from each reference report. The mean reference lengths are approximately 1,800 characters in training and 2,000 in the test set. No test reference matches a training reference or shares its first 200 characters, and the test set contains no duplicate references. The model receives the month, day, and weekday but not the year. Live tool use is required, although this cannot exclude memorization of language from public reports.

B CURRICULUM DETAILS

Table B.1: Parameters used across the two-stage curriculum. Shared RL settings apply to both RL stages, and report-only settings apply to Stage 2.

Setting	Value	Role
Rejection sampling		
Base-policy rollouts per question	8	candidate trajectories for RSFT
Sampling	temperature 0.8	diversify successful trajectories
Retention cap	2 per question	limit repeated successes
Shared RL optimization		
Sampling	temperature 0.6, top-p 1.0	rollout sampling
Rollouts per prompt	4	group for the centered advantage
Prompts per step	16	64 rollouts per optimizer step
Maximum turns	13	turn 13 is the forced final turn
KL coefficient	0	no reference-policy term
LoRA rank / alpha	64 / 64	language linear layers with frozen vision tower
Loss	CISPO	clipped importance ratio, sequence mean
CISPO offsets $\epsilon_{\text{low}}/\epsilon_{\text{high}}$	1.0 / 4.0	clipped ratio interval $[0, 5]$
Report-stage reward		
Quality weighting	online, every step	inverse within-group standard deviation
Variance smoothing β	0.9	moving average across steps
Weight bounds	$[0.2,\ 0.8]$	keep both scores in the reward
Minimum scored groups	8	fewer groups keep the previous weight
Contract-fail or no-report reward	-0.05	terminal failure outcome
Coerced-format penalty	0.02	final reply without a solution tag
Minimum usable report	40 characters	training skips unusable reports
Report-stage episode budgets		
Observation cap	5,000 characters	printed output shown per turn
Evidence-summary cap	2,600 characters	server summary appended per turn
Prompt token budget	38,000	forces the final turn when exceeded
Episode token budget	46,000	forces the final turn when exceeded
Generation cap	10,000 tokens	one policy action
Images per turn / episode	2 / 8	long side resized to 768 pixels
Report test evaluation		
Rollouts per event	1	one report per system
Evaluation cadence	every 32 steps	each checkpoint scored by the three judges
Sampling	temperature 0.6, top-p 1.0	same as training for local models

Table B.2: The eight ZEPHYRUSBENCH question templates used in Stage 1 and the tool required by each template.

Question template	Answer	Required tool
Geographic feature with the highest or lowest time-mean field value	location	`ds`
Minimum, maximum, mean, or median field value over a named geographic feature	numerical	`ds`
Sublocation within a named parent feature with the highest or lowest recorded field value	location	`ds`
Hours from the initial time at which a named feature reaches its maximum or minimum field value	time	`ds`
Minimum, maximum, mean, or median field value over a named feature at a specified time	numerical	`ds`
Geographic features above or below a specified climatological percentile	anomaly list	climatology
Geographic features whose field differs from the climatological mean by a specified number of standard deviations	anomaly list	climatology
Geographic features outside a specified lower–upper climatological percentile envelope	anomaly list	climatology

The curriculum comprises rejection-sampling fine-tuning, Stage-1 RL on verifiable questions, and report RL. Table B.1 consolidates their sampling, rollout, optimization, reward, and episode-budget parameters. The following subsections give the stage-specific data, selection rules, and evaluation protocols.

B.1 REJECTION-SAMPLING FINE-TUNING

A rollout is eligible for RSFT only if it meets all five conditions below.

1. the answer receives full verifier credit.
2. the access counter for the question's required tool is positive.
3. no infrastructure or driver failure is recorded.
4. the terminal response contains a matched `<solution>` block whose payload parses under `json.loads` and contains no non-finite constants.
5. no turn has a protocol error from omitting both a recognized `<execute>` block and a recognized `<solution>` block.

The completed runs use required-tool access as the gate rather than the stricter semantic-success diagnostic. A trajectory remains eligible if the model repairs an error in its own code. Both corpora favor fewer execution errors, fewer turns, and shorter responses, in that order. For 27B, trajectories that avoid a forced final turn receive an additional preference. The sample index breaks any remaining tie. Vision is disabled for these numerical analysis tasks. RSFT trains a rank-64 LoRA adapter on the language layers for three epochs with learning rate 10^{-5} and batch size four. The 4B model produces 1,801 full-credit, required-tool-access rollouts (31.7%), compared with 3,125 (55.0%) for 27B. The larger base policy therefore solves the verifiable tasks more reliably before fine-tuning. The two-per-question cap keeps repeated successes from dominating either RSFT corpus.

B.2 STAGE-1 VERIFIER

Stage 1 uses the eight ZEPHYRUSBENCH question templates in Table B.2. The training and held-out validation questions are disjoint but follow the same templates. Answers may be numerical values, locations, times, or anomaly lists. Gold values come from the benchmark metadata. For location questions, we parse the gold place name from the reference response when building the data. The model returns JSON inside a solution tag. Table B.3 gives the required format and score ladder. Numerical questions use relative error $|\text{pred} - \text{gold}|/\max(|\text{gold}|, 10^{-9})$, with absolute-error bands when $|\text{gold}| < 1$.

The ladder gives partial credit to nearby numerical or temporal values, correct values paired with the wrong location, and partially correct anomaly sets. A separate tool check prevents an unsupported guess from qualifying as a successful trajectory.

Table B.3: Stage-1 answer formats and verifier scores. Numerical questions use absolute-error bands when the gold magnitude is below one. Anomaly lists receive graded set F1.

Type	Required content	Score 1.0	Score 0.5	Score 0
Numerical	one JSON number	relative error $\leq 1\%$ or absolute error ≤ 0.05	relative error $\leq 5\%$ or absolute error ≤ 0.15	otherwise
Time	one JSON number in hours	exact	within 6 h	otherwise
Location	JSON object with place name and value	normalized name equals or contains the gold name	wrong name, value inside the numerical 1.0 band	otherwise
Anomaly	JSON list of names (empty if none)	set F1 over normalized names, graded in $[0, 1]$	(graded)	no overlap

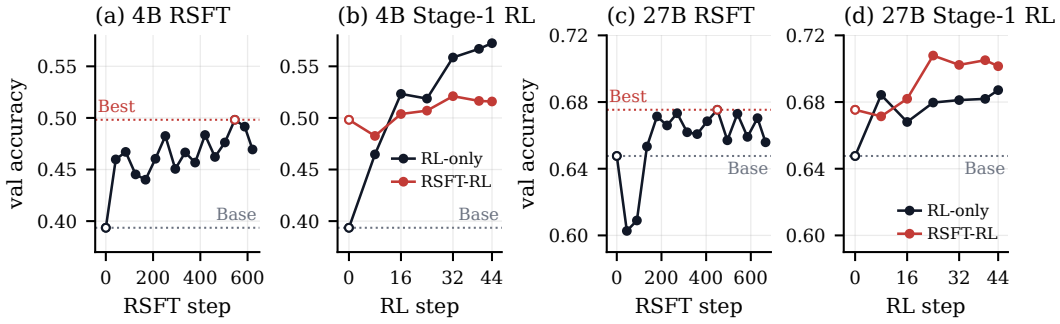

Figure B.1: Stage-1 accuracy on the 125-question held-out validation set for Qwen3.5-4B and Qwen3.8-27B. Filled circles show scored checkpoints, while open circles at step 0 mark each route's initial policy. RSFT and RL-only start from the base policy, whereas RSFT-RL starts from the selected RSFT checkpoint. Grey dotted lines show base accuracy. In the RSFT panels, the red open circle and dotted line mark the best checkpoint.

In Equation (3), s is the ladder score and the turn penalty is $\rho(T) = \min(0.15,\ 0.05\max(0, T-5))$. The required-tool condition holds when the server records at least one call to the dataset or climatology, as specified by the question.

B.3 Stage-1 RL and evaluation

Settings. Stage-1 RL uses the report-stage trainer but replaces the judge with the accuracy reward in Equation (3). It uses learning rate 10^{-5}. RSFT-RL starts from the RSFT checkpoint, while RL-only starts from the base policy. The 44 batches contain 704 draws from the 710-question pool. Report RL is initialized from RL-only step 44 for Qwen3.5-4B and RSFT-RL step 24 for Qwen3.8-27B.

Evaluation protocol. The held-out validation split contains 125 questions. We sample four rollouts per question at each checkpoint, giving 500 rollouts. Figure B.1 shows the checkpoint trajectories, and Table B.4 breaks down the selected results by answer type. These questions are used only for Stage-1 checkpoint selection and are separate from the report-stage test events.

Required-tool access ranges from 0.988 to 1.000, and the accuracy reward differs from verifier accuracy by at most 0.002. Differences between checkpoints therefore come mainly from answer quality. For Qwen3.5-4B, RL-only improves all four answer types over the base policy, with the largest gains on numerical (+0.215) and anomaly (+0.187) questions. For Qwen3.8-27B, RSFT-RL outperforms RL-only on numerical and location questions but is slightly worse on time and anomaly questions. The benefit of RSFT therefore depends on the model scale and question type.

B.4 Representative Stage-1 trajectories

Claude Opus 5.5 first read and summarized the complete logged trajectories. After reviewing its summaries, we manually selected one representative Qwen3.5-4B trajectory for each Stage-1 answer type and chose the turn-level information shown in the four boxes below. We checked every displayed

Table B.4: Stage-1 results on the held-out validation set. Acc. averages the per-answer score in Table B.3. Acc. reward averages Equation (3), including the tool check and turn penalty. Tool is the required-tool access rate. The remaining columns give mean scores by answer type.

	Verifier			Accuracy by answer type			
Checkpoint	**Acc.**	**Acc. reward**	**Tool**	**Num.**	**Loc.**	**Time**	**Anom.**
Qwen3.5-4B	0.394	0.394	0.996	0.334	0.400	0.392	0.522
RSFT (step 546)	0.498	0.496	0.988	0.467	0.521	0.457	0.595
RL-only (step 44)	**0.572**	**0.572**	0.998	**0.549**	**0.562**	**0.526**	**0.709**
RSFT-RL (step 32)	0.521	0.521	**1.000**	0.473	0.512	0.491	0.688
Qwen3.8-27B	0.648	0.648	0.998	0.571	0.637	0.685	0.785
RSFT (step 450)	0.675	0.675	**1.000**	0.587	0.667	0.694	0.865
RL-only (step 44)	0.687	0.687	**1.000**	0.592	0.650	**0.728**	**0.901**
RSFT-RL (step 24)	**0.708**	**0.708**	**1.000**	**0.636**	**0.679**	0.724	0.893

detail against the original traces. Brackets summarize the model's reasoning, code is abridged to the relevant lines, and observations are condensed from the logs. The examples cover a one-call numerical solution, a repaired indexing error, recovery from an unsuccessful location lookup, and climatology-based anomaly detection.

Stage-1 trajectory, Qwen3.5-4B, numerical task, one call

Question. What is the minimum surface temperature in Antarctica?

Required tool. `ds`. **Gold. 233.517 K.**

Turn 1. [Use `2m_temperature`, locate Antarctica, apply the mask, and reduce.]

```
def run(ds, geolocator, forecaster, climatology, tempestextremes, plotting):
    mask = geolocator.get_mask_by_geofeature('continent', 'Antarctica')
    t2m = ds[0]["2m_temperature"]
    value = t2m.where(mask == True).min().values
    print(f"Minimum 2m temperature: {value} K")
    return {"min_antarctic_temperature": value}
```

Observation. Minimum 2m temperature is 233.4048614501953 K.

Turn 2. [Return one number in Kelvin.] `<solution>233.40486</solution>`

Verifier. The relative error is 4.8×10^{-4}, so $s = 1$. The tool condition is satisfied. The rollout uses two turns and one code call with no execution error.

Stage-1 trajectory, Qwen3.5-4B, location task, error repaired

Question. Which continent experienced the lowest average specific humidity at 400 hPa?

Required tool. `ds`. **Gold. Antarctica.**

Turn 1. [Find the 400 hPa level, mask every continent with the geolocator, and average.]

```
def run(ds, geolocator, forecaster, climatology, tempestextremes, plotting):
    sh = ds[0]['specific_humidity']
    print("Level coordinates:", sh.level.values)
    level_400_idx = np.where(sh['level'].isel(index=0) == 400)[0][0]
    ...
```

Observation. The level coordinates are [50 100 150 200 250 300 400 500 600 700 850 925 1000]. The third line then raises `ValueError: Dimensions {'index'} do not exist. Expected one or more of ('level',).`

Turn 2. [The level coordinate has no time dimension, so index it directly.]

```
def run(ds, geolocator, forecaster, climatology, tempestextremes, plotting):
    sh = ds[0]['specific_humidity']
    sh_400 = sh.isel(level=int(np.where(sh.level == 400)[0][0])).isel(index=0)
    results = {}
    for continent in geolocator.get_all_locations('continent'):
        mask = geolocator.get_mask_by_geofeature('continent', continent)
        results[continent] = float(sh_400.where(mask).mean())
    lowest = min(results, key=results.get)
    print("Lowest average:", lowest, "with", results[lowest])
    return {"min_continent": lowest, "min_value": results[lowest]}
```

Observation. The lowest average is Antarctica with 0.000112. The other continental means range from 0.000114 in Europe to 0.001050 in South America.

Turn 3. [Return the name and the value in the required JSON form.] `<solution>{"name": "Antarctica", "value": 0.000112}</solution>`

Verifier. The normalized name matches the gold name, so $s = 1$. The tool condition is satisfied. The rollout uses three turns and two code calls, correcting one execution error.

Stage-1 trajectory, Qwen3.5-4B, time task, lookup recovered

Question. How many hours from the initial timeframe did Ostrobothnia, Finland experience its highest 10-meter V component of wind?

Required tool. `ds`. **Gold. 132 h.**

Turn 1. [Identify Finland, list its sublocations, and query the expected coordinates of Ostrobothnia.]

```
def run(ds, geolocator, forecaster, climatology,
        tempestextremes, plotting):
    finland = geolocator.get_geofeature_by_point(
        25.0, 62.0, "country")
    states = geolocator.get_sublocations("country", finland)
    name = geolocator.get_geofeature_by_point(
        22.0, 64.0, "state")
    print("States in Finland:", states)
    print("Ostrobothnia state:", name)
    return {"finland": finland, "states": states}
```

Observation. The direct coordinate lookup returns `None`. The sublocation list contains the registered name `Ostrobothnia, Finland`.

Turn 2. [Use the registered name and check that its mask aligns with the wind field.]

```
def run(ds, geolocator, forecaster, climatology,
        tempestextremes, plotting):
    name = "Ostrobothnia, Finland"
    mask = geolocator.get_mask_by_geofeature("state", name)
    wind = ds[0]["10m_v_component_of_wind"]
    print("Mask shape:", mask.shape)
    print("Wind dimensions:", wind.dims)
    print("Number of time steps:", wind.sizes["index"])
    return {"name": name, "mask_shape": mask.shape}
```

Observation. The mask has shape $(360, 181)$, matching the longitude and latitude axes of the wind field. The episode contains 28 time steps.

Turn 3. [Find the spatial maximum within the mask at every time step and retain the largest one.]

```
mask = geolocator.get_mask_by_geofeature(
    "state", "Ostrobothnia, Finland")
wind = ds[0]["10m_v_component_of_wind"]
max_v, max_idx = -1e99, -1
for idx in range(wind.sizes["index"]):
    masked = wind.isel(index=idx) * mask
    value = float(masked.max())
    print(idx, value)
    if value > max_v:
        max_v, max_idx = value, idx
return {"max_v": max_v, "time_index": max_idx,
        "hours": max_idx * 6}
```

Observation. The per-time output is long enough that the final lines are truncated, although the execution succeeds.

Turn 4. [Repeat the same reduction without printing every time step.]

```
mask = geolocator.get_mask_by_geofeature(
    "state", "Ostrobothnia, Finland")
wind = ds[0]["10m_v_component_of_wind"]
max_v, max_idx = -1e99, -1
for idx in range(wind.sizes["index"]):
    value = float((wind.isel(index=idx) * mask).max())
    if value > max_v:
        max_v, max_idx = value, idx
return {"max_v": max_v, "time_index": max_idx,
        "hours": max_idx * 6}
```

Observation. The maximum is 8.7056 m s^{-1} at index 22, which is 132 hours from the initial time.

Turn 5. [Return the requested hour.] `<solution>132</solution>`

Verifier. The time matches exactly, so $s = 1$. The dataset condition is satisfied. The rollout uses five turns and four code calls with no execution error.

Stage-1 trajectory, Qwen3.5-4B, anomaly task, one call

Question. Which water bodies contain time-mean sea-level pressure at least three standard deviations below the all-time climatology?

Required tool. Climatology. **Gold. Philippine Sea.**

Turn 1. [Standardize the time-mean field against the all-time climatology, then intersect cells below -3σ with each water-body mask.]

```
def run(ds, geolocator, forecaster, climatology,
        tempestextremes, plotting):
    field = ds[0]["mean_sea_level_pressure"].mean(dim="index")
    mu = climatology.get_climatology(
        "mean_sea_level_pressure", "all_time", "mean")
    sigma = climatology.get_climatology(
        "mean_sea_level_pressure", "all_time", "std")
    below = ((field.values - mu.values) / sigma.values) < -3
    answer = []
    for name in geolocator.get_all_locations("water_body"):
        mask = geolocator.get_mask_by_geofeature("water_body", name)
        if (below & mask).any():
            answer.append(name)
    return answer
```

Observation. The call returns `["Philippine Sea"]`.

Turn 2. [Return the anomaly set as JSON.] `<solution>["Philippine Sea"]</solution>`

Verifier. The set F1 is 1. The returned mean and standard deviation satisfy the climatology condition. The rollout uses two turns and one code call with no execution error.

C Method and Evaluation Details

C.1 Report-stage reward and training constants

The quality reward follows Equation (1). Direct RL and Curriculum RL use the same update rule, initial weights, smoothing, and bounds. The realized weights depend on the rollout scores in each run. At optimization step t, let G_t count prompt groups whose K rollouts all passed the contract and were scored. For component $c \in \{\mathrm{r}, \mathrm{f}\}$ (rubric or F1), the pooled within-group variance is

$$\hat{\sigma}^2_{c,t} = \frac{1}{G_t(K-1)} \sum_{g=1}^{G_t} \sum_{k=1}^{K} \left(s_{c,gk} - \bar{s}_{c,g}\right)^2, \tag{4}$$

where $s_{c,gk}$ is rollout k's score and $\bar{s}_{c,g}$ its prompt-group mean. An exponential moving average $v_{c,t} = \beta v_{c,t-1} + (1-\beta)\hat{\sigma}^2_{c,t}$ smooths the estimates.

With $\tilde{\sigma}_{c,t} = \max(\sqrt{v_{c,t}}, 10^{-6})$, the rubric weight is

$$w_t = \mathrm{clip}_{[w_{\min},\, 1-w_{\min}]}\left(\frac{\tilde{\sigma}_{\mathrm{f},t}}{\tilde{\sigma}_{\mathrm{r},t} + \tilde{\sigma}_{\mathrm{f},t}}\right). \tag{5}$$

Because CISPO compares rollouts for the same prompt, the estimate uses within-group variation rather than differences in difficulty across prompts. The inverse-spread rule gives the less variable component a larger coefficient, preventing the more variable score from dominating group-relative advantages. We use $\beta = 0.9$ to dampen changes caused by an atypical batch and $w_{\min} = 0.2$ to keep both quality signals active. A step updates the weights only when at least eight groups are fully scorable. Otherwise, it retains the previous weights. Each run therefore learns its own weight trajectory from its current rollout distribution. Before clipping, the weighted spreads satisfy $w_t\tilde{\sigma}_{\mathrm{r},t} = (1-w_t)\tilde{\sigma}_{\mathrm{f},t}$. We apply the blend before computing group-centered advantages. Groups with a contract or infrastructure failure are excluded so that fixed penalties and missing scores do not distort the estimated quality spread. The weighting changes only the training blend. The raw rubric and F1 scores remain unchanged and are reported separately. At the forced final turn, unusable and ungrounded reports receive -0.05. Usable but ungrounded reports retain judge metrics only for monitoring, while unusable reports skip judging. We subtract a 0.02 format penalty only when the report must be recovered from an untagged forced reply.

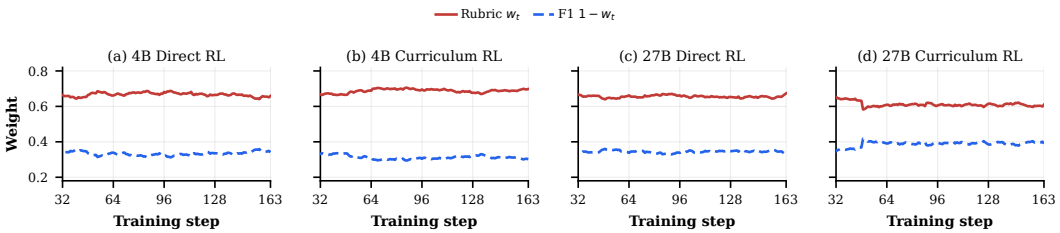

Figure C.1: Online reward weights during DIRECT RL and CURRICULUM RL report RL for both model scales. Each run extends through step 163, and the two weights sum to one.

Figure C.1 shows the realized weights. For Qwen3.5-4B, the rubric weight ranges from 0.54 to 0.69 under DIRECT RL and from 0.66 to 0.71 under CURRICULUM RL. The corresponding ranges for Qwen3.8-27B are 0.58 to 0.69 and 0.58 to 0.66. Claim-entailment F1 has greater within-group variation and therefore receives less weight throughout each run.

The 163 steps make one pass over the 2,608 report-training events and produce 10,432 rollouts. DIRECT RL and CURRICULUM RL share the same rollout budget, optimizer, weighting rule, and evaluation schedule. They differ only in their initial policy. Failed executions carry no retry penalty, so the model can repair its code within the turn and resource limits. Only usable reports that satisfy the evidence contract are scored for quality.

C.2 REPORT GROUNDING REQUIREMENT

At each turn, the model returns either Python code in an `<execute>` block or a report in a `<solution>` block. A response containing both tags is rejected without executing the code or ending the episode. A response containing neither tag receives a corrective observation. DIRECT RL and CURRICULUM RL use the same prompt, which provides the tool documentation, data layout, units, date context, budgets, evidence contract, and report style. Both routes allow 12 working turns followed by a forced final turn. Appendix F.1 reproduces the prompt and environment messages.

The grounding state is maintained by the execution server and reset at the start of every episode. It contains three cumulative Boolean witnesses. Dataset access passes after `ds` returns at least one valid result. Forecast access passes after the forecaster returns valid fields at a requested lead from the ground-truth backend. Evidence delivery passes after the server appends at least one verified finding to the model's observation. These witnesses may be satisfied on different turns, and the grounding flag is their conjunction.

$$g(\tau) = \mathbf{1}[\text{dataset access}]\,\mathbf{1}[\text{forecast access}]\,\mathbf{1}[\text{evidence delivered}]. \tag{6}$$

The server updates these witnesses from execution metadata rather than from the model's code, printed text, or use of an `<execute>` tag. Tool access requires a server-recorded successful return. Evidence delivery requires a server-generated finding that is actually included in the observation after output truncation.

Before the forced final turn, a `<solution>` with $g(\tau) = 0$ is rejected, consumes the turn, and receives a message requesting the missing evidence. The episode then continues. At the cutoff, the environment requests a final response regardless of the grounding state. If the contract remains incomplete, the rollout receives the fixed reward -0.05, and any judge metrics are retained only for monitoring rather than used for training. Unambiguous server or judge failures are handled separately through the group-invalidation rule in Appendix C.6.

C.3 REPRESENTATIVE REPORT-STAGE TRAJECTORY

The trajectory below produced the Qwen3.8-27B CURRICULUM RL report in Figure 4 and Appendix E. Claude Opus 5.5 first read and summarized the complete trace. After reviewing its summary, we manually selected the representative turns and information shown below and checked every displayed detail against the original log. The box shows the two evidence-acquisition calls, summarizes the remaining analysis, and reproduces the opening paragraph. The rollout contains six successful executions and no execution errors.

Report-stage trajectory, Qwen3.8-27B CURRICULUM RL

Request. Produce an approximately 300-word operational CONUS report covering systems and movement, temperature, precipitation and hazards, pressure evolution, fronts, and the jet stream.

Turn 1. Analysis acquisition. [Read the current pressure and 500 hPa fields over CONUS.]

```
def run(ds, geolocator, forecaster, climatology,
        tempestextremes, plotting):
    d = ds[0]
    mslp = d["mean_sea_level_pressure"].isel(index=0) / 100
    us_mslp = mslp.sel(latitude=slice(24, 50),
                       longitude=slice(235, 293))
    pmin = us_mslp.isel(
        us_mslp.argmin(dim=["longitude", "latitude"]))
    z500 = d["geopotential"].sel(level=500).isel(index=0) / 9.80665
    return {"mslp_min": float(pmin),
            "z500_min": float(z500.sel(latitude=slice(24, 50),
              longitude=slice(235, 293)).min())}
```

Observation. Current MSLP minimum 1008.33 hPa near 50°N, 110°W and 500 hPa minimum 5429.72 m. Server-delivered evidence additionally records 70.44 mm maximum 12-hour precipitation in the southern Plains, a 65.49 m/s 250 hPa jet maximum in the northern Plains, and a 5891.5 m 500 hPa maximum in the Southeast. Analysis access and evidence delivery now pass. Forecast acquisition does not yet pass.

Turn 2. [Plot current MSLP and 500 hPa height to inspect the trough–ridge structure.]

Turn 3. Forecast acquisition. [Generate 6–72 h oracle fields and detect extratropical tracks.]

```
def run(ds, geolocator, forecaster, climatology,
        tempestextremes, plotting):
    context = ds[0].isel(index=[0, 2])
    preds = [forecaster.forecast(context, horizon=h)
             for h in range(1, 13)]
    seq = xr.concat(preds, dim="index")
    tracks = tempestextremes.detect_with_profile(
        seq, mode="extratropical", strictness="medium",
        region={"minlat": 24, "maxlat": 52,
                "minlon": -125, "maxlon": -70}, top_k=4)
    return {"n_tracks": len(tracks)}
```

Observation. Twelve forecast leads and two tracks returned. Verified evidence includes a t + 36 h MSLP minimum of 1005.72 hPa and 500 hPa minimum of 5395.6 m in the northern Plains, plus 50.13 mm maximum 12-hour precipitation in the southern Plains. Analysis read, forecast obtained, and evidence delivered now all pass.

Turns 4–6. [Inspect track points, 850 hPa temperature, precipitation, and the t + 36 h 500 hPa field.] All three executions succeed. The agent interprets an eastward-moving northern trough and front, colder air behind it, and a persistent heavy-rain signal over the southern Plains.

Turn 7. Final report. [Synthesize the verified analysis and forecast evidence into continuous prose.]

Opening paragraph. A longwave 500 mb trough currently sits over the northern Plains and Canadian prairies, with the upper-level jet core (250 mb winds near 65 m/s) overhead and a surface low centered near the northern Plains–Canadian border around 1008 hPa. A cold front extends south from that low across the central Plains, marking a sharp 850 mb thermal gradient between sub-freezing air over the northern Plains and warm air (8–12 °C at 850 mb) over the South. A secondary low over the southern Plains, near the Oklahoma–Kansas–Texas Panhandle area, is tapping warm, moist Gulf air and is the main source of heavy rain, with 12-hour accumulations already reaching 50–70 mm and 6-hour totals near 45 mm.

The complete report appears with claim labels in Figure 4 and Appendix E. The episode ends after seven turns with six successful executions, no execution errors, a satisfied contract, and a native final report.

C.4 REFERENCE-ENTAILMENT SCORING

Claim-entailment F1 uses the claim extraction prompt and the fixed claim-definition evaluation prompt of Appendix F.3. The judge is queried separately for each claim at temperature zero. As defined in Section 3.2, NEUTRAL generated claims are excluded from the precision denominator. Precision and F1 are zero when no generated claim is SUPPORTED or REFUTED, including when no report claims are extracted. A reference claim that the report neither states nor contradicts counts as a recall miss. For offline evaluation, reports shorter than 40 characters are replaced by the no-report sentinel before claim extraction. The sentinel is processed through extraction and claim labeling in both directions, and the score follows the same precision and recall formulas. There is no fixed

Table C.1: The six meteorological rubric criteria supplied to the judge. Anchored integer scores are normalized to $[0, 1]$.

Criterion	Definition
Setup Fidelity & Key Feature Retention (Additions AND Omissions)	Accurately represents the specific synoptic/mesoscale features and stated hazards/corridors in the setup, without introducing unsupported details and without dropping any primary stated feature/hazard that changes the forecast story.
Spatiotemporal Specificity (Progression When Present; Persistence When Static)	Communicates where and when impacts occur with specificity appropriate to the setup: describes progression tied to moving features when the setup is progressive, or explicitly states persistence/limited movement when the setup is quasi-stationary.
Impact/Hazard Extraction & Prioritization (Reference-Weighted)	Identifies the setup's main hazards and impact corridors (and any stated relative maxima/minima) and ranks them appropriately, using practical, decision-relevant framing without adding new hazards.
Mechanism-to-Weather Reasoning (Sufficient, Setup-Consistent)	Uses meteorologically correct, setup-consistent causal links (forcing, moisture transport, instability, terrain/onshore/upslope, deformation/lake-effect) to explain why impacts occur where/when described. Depth is rewarded but not required for a high score if the forecast is a faithful paraphrase.
Thermal/Airmass & Precip-Type Plausibility (No Out-of-Pattern P-Type)	Maintains a coherent temperature/airmass narrative consistent with the setup and seasonally plausible precipitation type statements (snow/ice/convective vs stratiform), avoiding thermodynamic regime shifts not supported by the setup.
Constraint-Respecting Detail & Calibration (No Invented Timing/Numbers/Pattern Labels)	Uses specificity only when supported: avoids invented cutoffs ("ends by Saturday evening"), ungrounded pattern labels ("split flow") or pseudo-precision, and matches the setup's confidence level (possible/expected/likely).

zero-score override. During training, unusable reports bypass both judges and receive the no-report penalty of Appendix C.1.

C.5 Rubric construction and scoring

Construction. Table C.1 lists the six criteria in the frozen rubric. We construct the rubric from human-written references in the report-training split. With seed 0, we sample 100 references for positive examples and another 100 for negative examples. GPT-5.2 creates each positive by faithfully paraphrasing its reference. It creates each negative by removing weather claims, adding claims from an unrelated training reference, or applying both changes. Light, medium, and heavy corruptions alter one, two to three, and four to six claims. We then paraphrase the negatives in the same way as the positives to reduce stylistic cues. We also sample 25 positive and 25 negative examples with disjoint task identifiers for final rubric evaluation. All source reports come from the report-training split, but these examples are used only to construct the rubric and never to train the policy.

GPT-5.2 writes and scores the candidate rubrics with medium reasoning effort. The first round receives five positive and five negative seed examples that are separate from the refinement subset. We score each candidate on a fixed subset of 16 positive and 16 negative examples. In later rounds, GPT-5.2 sees the previous rubric, both score distributions, and the six examples with the most problematic scores. It then revises the criteria and anchors. Refinement runs for at most four rounds and stops when AUC reaches 0.9 and its improvement falls below 0.05. We select the candidate with the highest AUC and use the mean score gap to break ties. The selected rubric comes from round 3. We evaluate it once on the held-out 25+25 examples, review each criterion, and freeze it before policy training. Examples from the report test set and outputs from policy checkpoints are never used during rubric construction.

Scoring protocol. The production prompt contains the frozen rubric, task request, reference report, and generated report. At temperature zero, one judge call returns an integer score and a brief explanation for each of the six criteria, followed by an overall assessment. The rubric anchors scores at 1, 3, 5, 8, and 10 but allows the intermediate integers. We round returned values to integers and clip them to $[1, 10]$. A missing criterion receives the minimum score. The raw rubric average is the mean of the six scores. We report the normalized value $(\text{average} - 1)/9$.

Setup Fidelity & Key Feature Retention (Additions AND Omissions)

1	Multiple major errors: invents or contradicts key systems/hazards OR omits multiple primary hazards/corridors explicitly in the setup.
3	One major fault: a single invented/contradictory feature that materially changes the story (e.g., added cutoff/dry slot/Arctic front) OR omission of a primary stated hazard/corridor (e.g., ignores a named snowstorm area).
5	Broadly consistent, but includes noticeable unsupported embellishments or misses an important secondary detail (still preserves the main story).
8	Faithfully captures all primary features/hazards/corridors; only small, non-impactful deviations or compressions.
10	Near-verbatim fidelity: all primary features/hazards and their described relationships are retained; no unsupported additions.

Spatiotemporal Specificity (Progression When Present; Persistence When Static)

1	Vague in both time and place; no usable windows or corridors.
3	Some geography or some timing, but not both; progression/persistence is unclear or misleading.
5	Provides identifiable corridors and broad windows (e.g., through Thu/Fri, weekend), but progression/persistence linkage to the setup is incomplete.
8	Clear corridors and windows; correctly describes either (a) feature-tied progression or (b) persistent/unchanged pattern when that is the setup.
10	Consistently precise windows (day/night or 24–48h) and corridor evolution tied to named features, without inventing timing details.

Impact/Hazard Extraction & Prioritization (Reference-Weighted)

1	Misses or misidentifies the primary hazards, or prioritizes a non-existent hazard as the main threat.
3	Mentions hazards but misses a key high-impact corridor/maxima stated in the setup OR substantially downplays it into generic wording.
5	Captures the main hazard types and broad areas, but prioritization and 'where it's worst' localization are weak or generic.
8	Clearly highlights primary hazard corridor(s) with timing and relative emphasis consistent with the setup; separates main vs secondary impacts.
10	Decision-ready: includes the setup's key qualifiers where given (e.g., explicit 'few feet'/'south of X line' style maxima, 'possible vs expected') and avoids adding unstated threats.

Mechanism-to-Weather Reasoning (Sufficient, Setup-Consistent)

1	Mechanisms are largely incorrect or contradict the setup (e.g., wrong forcing for stated precip type/region).
3	Mostly outcome listing with weak/incorrect causal links, or mechanisms are used to justify details not present in the setup.
5	Basic correct links (e.g., ridge→heat, onshore/upslope→mountain precip, boundary→convection) but limited differentiation of intensity/coverage.
8	Multiple correct mechanisms tied to specific corridors/timing, explaining distribution and intensity differences without overreach.
10	Integrated, physically grounded explanation (moisture/forcing/thermodynamics/terrain) that clearly supports the stated impacts and evolution.

Thermal/Airmass & Precip-Type Plausibility (No Out-of-Pattern P-Type)

1	Thermal/ptype narrative is implausible or contradictory (e.g., widespread snow/ice or major cold outbreak implied without support).
3	Introduces a notable but unsupported thermodynamic/ptype implication (e.g., adds a cold-air push driving wintry precip outside stated areas) OR omits a key stated p-type transition (e.g., snow/wintry mix explicitly mentioned).
5	Mostly plausible; limited temperature evolution detail, but no major thermodynamic errors.
8	Coherent regional thermal contrasts/trends consistent with the setup; precip type aligned with elevation/airmass where relevant.
10	Strong, setup-faithful airmass evolution with clear implications for sensible weather and precip type where the setup supports it.

Constraint-Respecting Detail & Calibration (No Invented Timing/Numbers/Pattern Labels)

1	Frequent unjustified specifics (exact timing, magnitudes, or dynamics) or overconfident tone that is not supported by the setup.
3	At least one impactful unsupported embellishment (e.g., a specific weakening time, moisture cutoff, added structural diagnosis) that meaningfully changes interpretation or confidence.
5	Mostly calibrated; occasional minor over-specific or extra-structural wording, but not dominant.
8	Well-calibrated language; any numbers/timing/structure used are clearly supported by the setup; no fake precision.
10	Exemplary discipline: mirrors the setup's certainty, preserves stated qualifiers, and adds no extra structure beyond what is warranted.

Figure C.2: Complete scoring anchors for the six meteorological rubric criteria. The judge receives the displayed texts at levels 1, 3, 5, 8, and 10. Intermediate integer scores are allowed.

For offline evaluation, a report shorter than 40 characters is replaced by a no-report sentinel and passed through the same judge. During training, only grounded, usable reports receive judge-based rewards. Other outcomes receive the terminal failure reward under Appendix C.2, which also describes monitoring-only scores. Every reported system–judge score averages three complete passes over the same 197 frozen reports.

Complete scoring anchors. Figure C.2 gives the complete scoring anchors for the six rubric criteria.

High scores require faithful coverage of the main features, hazards, locations, timing, and uncertainty. Extra mechanisms or numerical detail help only when supported by the setup. The rubric therefore rewards accurate synthesis rather than detail alone.

C.6 Training details

Policy objective. For each prompt, the sampler draws K rollouts with continuous token-in token-out histories. Their group-centered advantages are $A_k = R(\tau_k) - K^{-1}\sum_j R(\tau_j)$, without standard-deviation normalization. Let $\mathcal{I}_k$ index policy-generated tokens, with environment tokens masked out. The sequence-mean CISPO loss is

$$\mathcal{L}(\theta) = -\frac{1}{K}\sum_{k=1}^{K} A_k \frac{1}{|\mathcal{I}_k|}\sum_{t\in\mathcal{I}_k} \mathrm{sg}\big[\,\mathrm{clip}(\rho_{k,t}, 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}})\big]\,\log \pi_\theta(a_{k,t} \mid s_{k,t}), \tag{7}$$

where $\rho_{k,t}$ is the current-policy to sampling-policy importance ratio and sg stops gradients through the clipped ratio. Invalidated prompt groups are masked out of the loss.

- **Parameters and masking.** LoRA adapters train the language blocks with the vision tower frozen and a zero KL coefficient. Observations, evidence summaries, and forced-turn messages are environment tokens and are masked out of Equation (7).
- **Tool witness.** The code server credits only returned, validated evidence-bearing calls. Invalid or non-evidence-bearing executions receive no contract credit.
- **Group invalidation.** An unambiguous infrastructure or judge failure excludes its four-rollout prompt group from the update. Policy-owned code and contract failures remain training outcomes.
- **Distributed placement.** Each training run uses two GPUs. For the 4B model, rollout inference and policy optimization run on separate devices. For the 27B model, a tensor-parallel inference engine (TP = 2) and two policy ranks are colocated across both devices.
- **Hardware.** Training uses two NVIDIA H200 GPUs. Standalone inference and evaluation use NVIDIA A100 GPUs.

C.7 Checkpoint selection and evaluation protocol

We select Stage-1 checkpoints by mean verifier credit on the 125-question held-out validation set. Ties are broken first by accuracy reward and then by the later step. Report-stage comparisons use the terminal step-163 policy. We also evaluate steps 32, 64, 96, 128, and 160 but omit step 160 from the curves. Each judge scores the same frozen reports at a given checkpoint. These diagnostics use the 197 test events, but they do not determine any checkpoint or hyperparameter. Appendix D.1 shows the complete trajectories.

D Additional results

This section presents checkpoint trajectories, judge comparisons, and the expert audit of claim labels.

D.1 Checkpoint trajectories

Figures D.1–D.4 show the report-RL trajectories.

Every evaluated checkpoint exceeds its base reference in F1 under all three judges. At 4B, Curriculum RL leads in F1 from step 96 onward, although the independent-judge rubric curves generally favor Direct RL. At 27B, the F1 curves converge midway through training, while Curriculum RL retains stronger rubric scores. Main comparisons use the terminal step-163 policies rather than intermediate peaks.

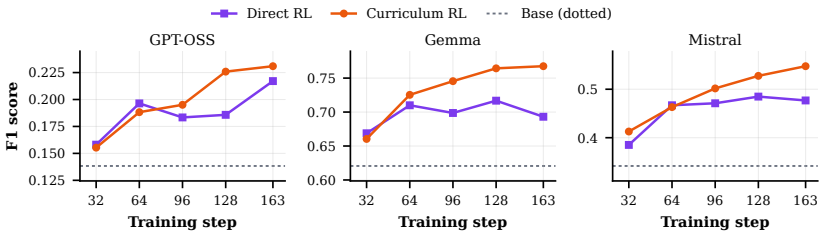

Figure D.1: Report-RL F1 trajectories for Qwen3.5-4B across the three evaluation judges. Each point averages three scoring passes on the 197 test events.

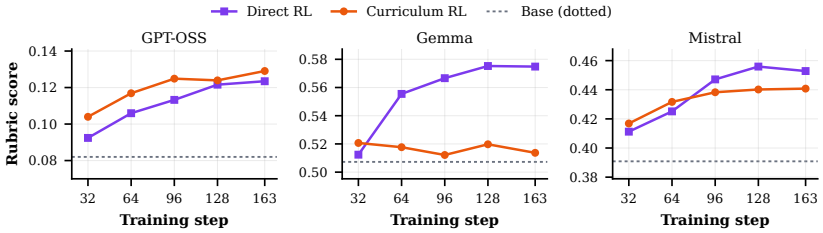

Figure D.2: Report-RL rubric-score trajectories for Qwen3.5-4B across the three evaluation judges. Each point averages three scoring passes on the 197 test events.

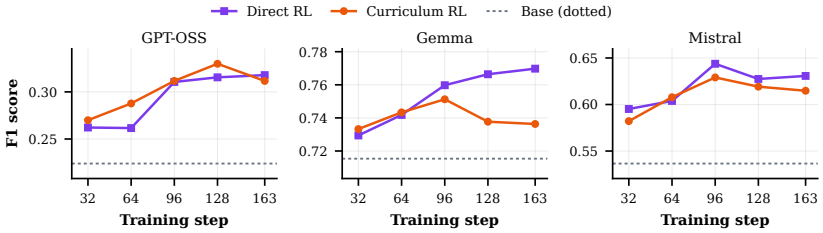

Figure D.3: Report-RL F1 trajectories for Qwen3.8-27B across the three evaluation judges. Each point averages three scoring passes on the 197 test events.

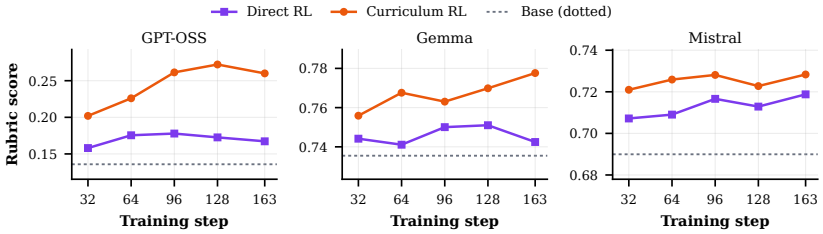

Figure D.4: Report-RL rubric-score trajectories for Qwen3.8-27B across the three evaluation judges. Each point averages three scoring passes on the 197 test events.

D.2 Judge diagnostics

The judges disagree more on individual reports than on overall system rankings (Table D.1). Gemma and Mistral agree most on both report-level metrics and on the rubric ranking. GPT-OSS and Mistral have the closest F1 ranking. We therefore focus on conclusions that are consistent across judges rather than comparing their absolute score levels.

Table D.2 explains some of these differences. Mistral extracts the most claims, Gemma assigns the largest supported fractions, and GPT-OSS is the most conservative in both directions. Each judge determines both the claims and their labels, so these choices change the F1 numerator and denominator together. Route comparisons within a judge are therefore more informative than raw scores across judges. Appendix E shows the three labeling patterns on one report.

Table D.1: Agreement between judges across 15 systems and 2,955 system–event reports. Per-report values are Spearman correlations between three-pass mean scores. Ranking values are Kendall's τ between system rankings.

	Per report		Ranking	
Judge pair	**Rubric**	**F1**	**Rubric**	**F1**
GPT-OSS / Gemma	0.46	0.48	0.66	0.71
GPT-OSS / Mistral	0.49	0.61	0.77	0.85
Gemma / Mistral	0.75	0.62	0.89	0.71

Table D.2: Claim extraction and support rates for the same 15 systems as Table D.1. Values pool all three scoring passes. Per rep. is the mean claim count, while Supp. and Cov. are supported fractions for report and reference claims.

	Report claims		Reference points	
Judge	**Per rep.**	**Supp.**	**Per rep.**	**Cov.**
GPT-OSS	11.7	0.15	12.8	0.20
Gemma	10.3	0.58	11.4	0.68
Mistral	16.6	0.28	14.6	0.49

D.3 Expert audit of claim labels

All three meteorologists label the same 100 claim–report pairs from five randomly sampled forecast tasks. The reports come from Qwen3.5-4B, Qwen3.8-27B, and GPT-5.6-Luna. Each task includes both comparison directions, with generated claims checked against the reference and reference claims against the generated report. Annotators see neither the direction nor the report generator.

The meteorologists use the same SUPPORTED, NEUTRAL, and REFUTED definitions as the LLM judges. Exact accuracy requires matching labels, while Cohen's κ adjusts for chance agreement. Within-1 agreement excludes only direct disagreements between SUPPORTED and REFUTED. We compute majority labels separately for the three meteorologists and three LLM judges, assigning NEUTRAL to a three-way tie. Table 1 reports pairwise and majority-vote agreement.

The audit covers claim labeling rather than claim extraction, rubric scoring, or complete-report preference. Its 100 pairs also come from only five tasks. The results support the practical use of the judge but do not establish human-level report evaluation.

E Judge claim extractions for one report

GPT-OSS-120B, 9 claims (4 supported, 4 neutral, 1 refuted)

[N1] A longwave 500 mb trough currently sits over the northern Plains and Canadian prairies, with the upper-level jet core (250 mb winds near 65 m/s) overhead and a surface low centered near the northern Plains–Canadian border around 1008 hPa. [N2] A cold front extends south from that low across the central Plains, marking a sharp 850 mb thermal gradient between sub-freezing air over the northern Plains and warm air (8–12 °C at 850 mb) over the South. [N3] A secondary low over the southern Plains, near the Oklahoma–Kansas–Texas Panhandle area, is tapping warm, moist Gulf air and is the main source of heavy rain, with 12-hour accumulations already reaching 50–70 mm and 6-hour totals near 45 mm.

[S1] Over the next 24–48 hours the upper trough and its surface low will track east across the northern Plains and into the Upper Midwest and Great Lakes, bringing a broad band of showers and storms to the northern Plains, the Dakotas, and the Lakes through Friday into Saturday. [S2] Cooler air behind the front will cap daytime highs and allow snow showers along the northern Rockies and higher terrain where 850 mb temperatures remain below freezing. [S3] Meanwhile the southern Plains low and front will persist into the weekend, maintaining a risk of scattered to frequent thunderstorms with locally heavy rain, flash-flood potential, and the threat of severe weather given the warm, moist airmass and strong mid-level flow.

[R1] The Southeast and much of the South remain under the eastern flank of a 500 mb ridge, where conditions stay warm, stable, and largely dry, with 2 m temperatures in the low to mid 30s °C and only isolated afternoon activity. [N4] The Pacific Northwest and northern Rockies will see intermittent precipitation from the western flank of the trough. [S4] The dominant hazard through the period is therefore the Plains frontal system: a moving band of precipitation and cool-air advection across the north, and a persistent, potentially severe and flood-prone storm track across the southern Plains, while the South enjoys warm, ridge-dominated weather.

Gemma-3-27B-IT, 9 claims (8 supported, 0 neutral, 1 refuted)

[S1] A longwave 500 mb trough currently sits over the northern Plains and Canadian prairies, with the upper-level jet core (250 mb winds near 65 m/s) overhead and a surface low centered near the northern Plains–Canadian border around 1008 hPa. [S2] A cold front extends south from that low across the central Plains, marking a sharp 850 mb thermal gradient between sub-freezing air over the northern Plains and warm air (8–12 °C at 850 mb) over the South. [S3] A secondary low over the southern Plains, near the Oklahoma–Kansas–Texas Panhandle area, is tapping warm, moist Gulf air and is the main source of heavy rain, with 12-hour accumulations already reaching 50–70 mm and 6-hour totals near 45 mm.

[S4] Over the next 24–48 hours the upper trough and its surface low will track east across the northern Plains and into the Upper Midwest and Great Lakes, bringing a broad band of showers and storms to the northern Plains, the Dakotas, and the Lakes through Friday into Saturday. [S5] Cooler air behind the front will cap daytime highs and allow snow showers along the northern Rockies and higher terrain where 850 mb temperatures remain below freezing. [S6] Meanwhile the southern Plains low and front will persist into the weekend, maintaining a risk of scattered to frequent thunderstorms with locally heavy rain, flash-flood potential, and the threat of severe weather given the warm, moist airmass and strong mid-level flow.

[R1] The Southeast and much of the South remain under the eastern flank of a 500 mb ridge, where conditions stay warm, stable, and largely dry, with 2 m temperatures in the low to mid 30s °C and only isolated afternoon activity. [S7] The Pacific Northwest and northern Rockies will see intermittent precipitation from the western flank of the trough. [S8] The dominant hazard through the period is therefore the Plains frontal system: a moving band of precipitation and cool-air advection across the north, and a persistent, potentially severe and flood-prone storm track across the southern Plains, while the South enjoys warm, ridge-dominated weather.

Mistral-Small-3.2-24B, 16 claims (4 supported, 9 neutral, 3 refuted)

[N1] A longwave 500 mb trough currently sits over the northern Plains and Canadian prairies, [N2] with the upper-level jet core (250 mb winds near 65 m/s) overhead [N3] and a surface low centered near the northern Plains–Canadian border around 1008 hPa. [N4] A cold front extends south from that low across the central Plains, [N5] marking a sharp 850 mb thermal gradient between sub-freezing air over the northern Plains and warm air (8–12 °C at 850 mb) over the South. [S1] A secondary low over the southern Plains, near the Oklahoma–Kansas–Texas Panhandle area, is tapping warm, moist Gulf air and is the main source of heavy rain, [N6] with 12-hour accumulations already reaching 50–70 mm and 6-hour totals near 45 mm.

[N7] Over the next 24–48 hours the upper trough and its surface low will track east across the northern Plains and into the Upper Midwest and Great Lakes, [N8] bringing a broad band of showers and storms to the northern Plains, the Dakotas, and the Lakes through Friday into Saturday. [S2] Cooler air behind the front will cap daytime highs and allow snow showers along the northern Rockies and higher terrain where 850 mb temperatures remain below freezing. [S3] Meanwhile the southern Plains low and front will persist into the weekend, maintaining a risk of scattered to frequent thunderstorms with locally heavy rain, flash-flood potential, and the threat of severe weather given the warm, moist airmass and strong mid-level flow.

[R1] The Southeast and much of the South remain under the eastern flank of a 500 mb ridge, where conditions stay warm, stable, and largely dry, [R2] with 2 m temperatures in the low to mid 30s °C and only isolated afternoon activity. [N9] The Pacific Northwest and northern Rockies will see intermittent precipitation from the western flank of the trough. [S4] The dominant hazard through the period is therefore the Plains frontal system: a moving band of precipitation and cool-air advection across the north, and a persistent, potentially severe and flood-prone storm track across the southern Plains, [R3] while the South enjoys warm, ridge-dominated weather.

Figure E.1: Independent claim extraction and labeling of the 27B CURRICULUM RL report in Figure 4. Tags mark supported (S), refuted (R), and neutral (N) claims. Each judge has its own numbering.

Figure E.1 shows both a stable report error and judge-dependent scoring. All three judges refute the claim that the Southeast and much of the South remain largely dry. This shared finding separates a robust qualitative error from judge-specific score calibration. The remaining variation concerns where claims are segmented and how much evidence counts as support. GPT-OSS and Gemma each extract nine claims, but GPT-OSS marks the three opening claims NEUTRAL while Gemma supports them. Mistral splits compound sentences into 16 narrower claims, producing nine NEUTRAL and three REFUTED labels. Claim granularity and support thresholds therefore change both the count and label mix entering precision, so absolute F1 reflects judge behavior as well as report quality. Table D.2 confirms this pattern in aggregate. We consequently compare routes within each judge and emphasize trends shared across judges.

F PROMPTS

F.1 REPORT-STAGE PROMPT

The report-stage prompt contains a system message, a user message, and a short cutoff message. The abridged version retains the action format, evidence contract, report style, and task request. Brackets mark omitted tool and data documentation. DIRECT RL and CURRICULUM RL use the same prompt for training and evaluation.

Report-stage prompt (abridged, with omissions in brackets)

[system message]

```
You are a weather expert agent.
At each turn, respond with exactly one block:  either Python code in <execute> tags or
the final answer in <solution> tags.  Use additional execution turns when they can
refine the answer.
```

[tool documentation, data layout, units, indexing rules, plotting guidance, and the execution signature]

```
TOOL-EVIDENCE CONTRACT for this task:
- Before a <solution> is accepted, successfully access the dataset and obtain forecast
fields with the forecaster.  A premature solution is rejected and costs a turn.
- Every response must contain exactly one tag, never both.
- Base the final report on observations from the current episode.
- After evidence-bearing calls, the environment appends a server-computed [verified
evidence] summary.  Optional tools should be used only when they materially change the
analysis.
FINAL REPORT STYLE - applies only inside <solution>:
Write approximately 250-350 words in 3-5 continuous prose paragraphs.
Do not use titles, headings, lists, or separator lines.  Begin with the dominant
forecast pattern, organise the report around regional and temporal evolution, connect
atmospheric mechanisms to impacts, and state uncertainty where evidence is limited.  Use
numbers selectively and do not describe the tool-use process.
```

[user message: atmospheric variables, time window, calendar context, interaction contract, tool documentation, data schema, units, execution rules, and the complete CONUS weather-report request]

[cutoff policy: at most 12 working turns followed by one forced final-answer turn]

The environment uses the following messages at cutoff or after an invalid response.

Forced final turn and rejection messages (opening sentences)

[contract passed] `Hard cutoff reached. Do not write more code. Your next response must be exactly one <solution> block containing your complete forecast discussion, closed with </solution>.`

[contract not passed] `Hard cutoff reached. Do not write more code. Required tool evidence was not completed this episode (successful dataset access and a successful forecast were both required), so a forecast discussion cannot be scored.`

[premature solution] `Your <solution> was not accepted yet: the final report must be based on evidence gathered in this episode.`

[neither tag] `No <execute> block and no <solution> block found.`

[no-answer sentinel] `Unable to provide a more detailed answer from the available analysis.`

F.2 Rubric judge prompt

At temperature zero, the rubric judge receives the system line and user prompt below. The inserted rubric contains the complete criteria and scoring anchors from Appendix C.5.

Rubric judge prompt

[system] `You are a meteorology evaluation expert. You score forecast discussions strictly against a rubric. Be terse, precise, and consistent.`

[user]
```
Score the following answer against the rubric.
Use the integer 1-10 scale defined by the anchors (1 = worst, 10 = best). Be strict and consistent.
For EACH criterion, give the integer score AND a 1-2 sentence explanation that
- cites concrete evidence from the answer (or its absence), and
- explains why that evidence places the answer at that anchor.
RUBRIC DESCRIPTION (what this rubric measures): {RUBRIC DESCRIPTION}
RUBRIC: {RUBRIC JSON}
QUESTION: {QUESTION}
REFERENCE (ground truth meteorologist discussion): {REFERENCE}
ANSWER TO SCORE: {ANSWER}
```

[output format: JSON with an integer score and a one-sentence explanation per criterion, plus one overall sentence; the keys must match the rubric criterion names]

F.3 Claim extraction and claim evaluation prompts

Claim-entailment F1 uses one prompt to extract claims and another to label each claim against the opposite text. Both prompts are reproduced verbatim, with braces marking the input slots.

Claim extraction

```
You are given a meteorological discussion. Please extract all the main claims and present it in compact bullet points.
Discussion: {}
Respond with a json list of strings as shown below:
{{
"points": [
"point_1",
"point_2",
```

```
...
]
}}
```

Claim evaluation

```
You are a climate expert that annotates if a given claim is either
SUPPORTED, REFUTED or NEUTRAL with respect to the presented report.
The evidence is in the form of a meteorological discussion.

Use the following definitions:

SUPPORTED: There is at least one statement in the report that
directionally agrees with the claim, i.e.  the claim could roughly be
used as a substitute for some statement(s) in the report.

NEUTRAL: There is not material information in the report related to
the claim.

REFUTED: There is at least one statement in the report that
directionally disagrees with the claim, i.e.  the report and the
claim can not both hold true simultaneously.

You will be provided the following as the input:

Claim:  {claim}

Evidence:  {report}

Respond with just one word:  SUPPORTED, NEUTRAL, or REFUTED.
```