HydroAgent: Closing the Gap Between Frontier LLMs and Human Experts in Hydrologic Model Calibration via Simulator-Grounded RL

Anonymous Author(s)
Affiliation
Address
email

Abstract

Calibrating distributed hydrologic models is a critical bottleneck in operational water-resources management: each basin demands a domain expert to translate hydrograph signatures into a high-dimensional parameter vector, and the workflow does not transfer between watersheds. Can frontier LLM agents do it? We benchmark nine frontier agents from the Claude, GPT, and Gemini families on calibration of the operational CREST model used for flash-flood forecasting at the U.S. National Weather Service. Mean best-of-twenty-rounds Nash–Sutcliffe Efficiency (NSE) across four held-out gauges (329–40,792 km^2) ranges from -0.16 to 0.75; the ceiling is reproducible across all three vendors and capability tiers, and no model reaches the human-expert reference except Opus-4.7 on one gauge. The gap is not a parameter-count problem but a domain-grounding problem. We then propose HYDROAGENT, which fine-tunes Qwen3-4B with supervised fine-tuning on $2{,}576$ expert calibration trajectories followed by Group-Relative Policy Optimization using NSE as a verifiable reward sourced from online CREST simulations — reinforcement learning with simulation feedback (RLSF). For Earth-system tasks with cheap-to-evaluate physical simulators, we argue that a small domain-tuned policy with simulator-in-the-loop RL is a more compute-efficient and physically faithful path than scaling generic frontier models.

1 Introduction

Hydrologic models are the infrastructure layer beneath a remarkable amount of public life: every reservoir release, irrigation allotment, drought outlook, hydropower bid, and flash-flood watch derives from a numerical rainfall-runoff model that has been calibrated against historical streamflow at the relevant gauge. The stakes are not abstract: flooding is the deadliest weather hazard in the United States [Li et al., 2021] and among the costliest natural disasters globally, with more than 1.8 billion people exposed to one-in-hundred-year flood risk and the burden falling disproportionately on the poorest [Rentschler et al., 2022, McDermott, 2022]. As climate change accelerates the hydrologic cycle and pushes flash floods and droughts into regions that have never experienced them at current intensities [Li et al., 2022], demand for accurate basin-specific simulations is rising at exactly the moment when the supply of trained hydrologic modelers cannot keep pace. AI is the obvious candidate to absorb that workload; the question this paper asks is *how*.

Hydrologic model calibration is the act of choosing a small number of physical parameters—soil-water capacity, infiltration shape, channel-routing coefficients—so that a numerical rainfall-runoff model reproduces observed streamflow at a stream gauge. It is the human-in-the-loop step that makes every downstream use of the model trustworthy—reservoir operation, water-supply allocation, drought outlook, hydropower scheduling, infrastructure design, and flood warning alike—and it is

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

the step that does *not* scale: each gauge in the conterminous United States requires hours to days of an expert hydrologist's attention, and what they learn about one basin transfers only loosely to the next. The mismatch between the climate-driven rise in demand sketched above and the human supply of calibration expertise is the gap that motivates this paper.

The recent surge in agentic large language models—models that interleave chain-of-thought reasoning with tool use over many turns [Yao et al., 2023, Schick et al., 2023, Qin et al., 2024]—raises an obvious question. These models have absorbed graduate-level hydrology textbooks, can read CSV gauge files, can edit control files, and can shell out to a Linux binary. Can they simply *do* the calibration? If yes, the cost structure of operational hydrology shifts overnight. If no, the manner of failure tells us where the gap actually lives.

This paper investigates that question along two complementary axes.

Axis 1: Where do frontier agents stand? We give nine frontier LLM agents—Claude Opus 4.6, Opus 4.7, Sonnet 4.6, GPT-5, GPT-5.4, GPT-5.4-pro, Gemini 2.5-pro, Gemini 3.1-pro, and Gemini 3-flash—the same task an entry-level hydrologist receives: a basin description, gauge time series, raster forcings (MRMS gauge-corrected precipitation, daily PET), the CREST control-file template, and the goal of achieving high NSE values. Each agent runs the calibration independently on each of four held-out gauges spanning 329–$40{,}792\,\mathrm{km}^2$, inside the same Linux sandbox under the *terminal-2* agent harness in the *harbor* evaluation framework with a fixed compute budget (2-hour timeout); the same harness is also used to evaluate our fine-tuned model in Section 4, so frontier-model and HYDROAGENT numbers are directly comparable.

Axis 2: Is a small, simulator-grounded model enough? Frontier-model evaluation suggests that the challenge is not raw reasoning capacity but *calibration grounding*: the model has not learned which parameter to move when the recession limb is too steep. To test whether that gap can be narrowed without scaling, we propose HYDROAGENT, a domain-specific agent built on the open-weight Qwen3-4B-Instruct model [Qwen Team, 2025]. As shown in Figure 1, training proceeds in two phases: (i) supervised fine-tuning (SFT) on $2{,}576$ calibration trajectories distilled from a stronger teacher, and (ii) reinforcement learning with simulation feedback (RLSF) using Group-Relative Policy Optimization (GRPO) [Shao et al., 2024, DeepSeek-AI, 2025], where each rollout invokes an online CREST simulation and the reward is a clipped NSE plus shaped per-turn signals.

The argument we want to defend in this paper has three pieces. First, frontier LLM agents are within striking distance of human-expert performance on hydrologic-model calibration but do not yet meet it; the gap is reproducible across model families. Second, that gap closes faster by *post-training a small open model with simulator-in-the-loop RL* than by waiting for the next frontier release; one does not need a 400-billion-parameter generalist to operate a 13-parameter physics simulator, and recent work on agentic AI for Earth observation has independently argued that generic agent recipes leave structural problems—geospatial consistency, physical validity, error propagation across long tool chains—unaddressed in ways that domain-tuned designs can fix [Munir et al., 2026]. Third, Earth-system science is an unusually fertile ground for this recipe because the data are rich and multi-modal: rasterized remote-sensing imagery (spatial-2D + time), in-situ measurements (time series), and expert narrative (e.g., National Weather Service warning messages and forecaster discussions) all carry information that a domain agent can be trained to fuse with a physical solver to yield steerable, physically consistent predictions.

Contributions. (1) The first systematic benchmark of nine frontier LLM agents on the calibration of an operational distributed hydrologic model, including a public release of all 9×20 trajectories for reproducibility (Section 3). (2) A domain-specific recipe (HYDROAGENT) that fine-tunes Qwen3-4B with SFT + GRPO using NSE as a verifiable simulator-grounded reward, with full training configuration sufficient to reproduce the run on $4\times$H100 (Section 2). (3) A position—defended quantitatively—that for Earth-system tasks with cheap-to-evaluate physical simulators, a domain-tuned 4B-parameter agent can substantially close the gap to frontier generalists, with an order of magnitude lower inference cost (Section 4).

2 Methods

This section describes (i) the CREST hydrologic simulator that defines our task and reward, (ii) the calibration environment exposed to the agent, and (iii) the SFT + RLSF training recipe used to

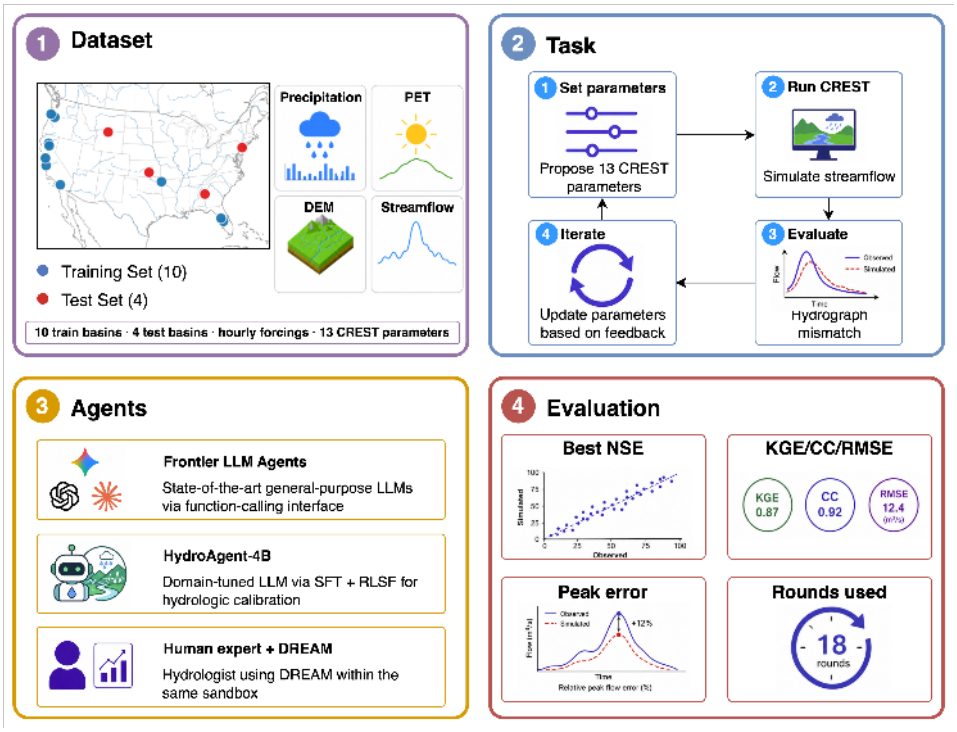


Figure 1: The HYDROAGENT training pipeline. Phase 1: supervised fine-tuning (SFT) on 2,576 expert calibration trajectories distills tool-call format and basic hydrologic reasoning into Qwen3-4B. Phase 2: Group-Relative Policy Optimization (GRPO) draws $K{=}8$ rollouts per prompt; each rollout proposes a CREST parameter set, executes the EF5 hydrologic simulator, scores a clipped Nash–Sutcliffe Efficiency, and updates the policy with group-normalized advantages. The simulator is the verifier: there is no learned reward model.

produce HYDROAGENT. The training infrastructure is `verl` 0.5 [Sheng et al., 2024] with SGLang multi-turn rollouts on $4\times$ NVIDIA H100 (80 GB) GPUs; configurations sufficient to reproduce the run are given in Appendix A.

2.1 The CREST/EF5 Hydrologic Simulator

The Coupled Routing and Excess STorage (CREST) model [Wang et al., 2011, Li et al., 2023] is a distributed, grid-based, conceptual rainfall-runoff model. At each grid cell and each hourly time step, CREST partitions incoming precipitation into impervious runoff, infiltration, and evapotranspiration as a function of a soil-water capacity grid (`wm`), a variable infiltration-curve shape parameter (`b`), an impervious-area fraction (`im`), an evapotranspiration scaling (`ke`), and a saturated hydraulic conductivity (`fc`); routes interflow with leakage (`leaki`) and velocity (`under`); and routes channel and overland flow with kinematic-wave parameters α, β, and α_0. Initial soil moisture (`iwu`) and a small set of state parameters complete the parameter vector; spatial heterogeneity is encoded in raster grids, and per-basin calibration is performed by tuning a vector of *scalar multipliers* on those grids, which preserves the spatial pattern while shifting magnitudes.

CREST is hosted by the Ensemble Framework For Flash Flood Forecasting (EF5) [Flamig et al., 2020], which is the operational hydrologic engine of the FLASH project [Gourley et al., 2017] at the NOAA National Severe Storms Laboratory and is used in real time by U.S. National Weather Service forecasters to issue flash-flood watches and warnings across the CONUS. CREST/EF5 has been deployed globally, including in data-sparse satellite-forced settings [Xue et al., 2013], and projections suggest that the demand for accurate calibration of CREST-class models will grow: under SSP5–8.5, flash-flood frequency over the CONUS is projected to increase [Li et al., 2022]. Two facts about CREST shape our experimental design: (i) it is non-differentiable in the parameter vector, ruling out gradient-based calibration and motivating an agent-in-the-loop approach, and (ii) a single basin/event

simulation completes in seconds-to-minutes, which makes simulator-feedback RL practical at the rollout scales we report.

2.2 Calibration Environment and Task

We instantiate one calibration episode per basin/event pair. Each episode exposes four tools to asgnts:

- `set_parameters`: write the 13 scalar multipliers (`wm`, `b`, `im`, `ke`, `fc`, `under`, `leaki`, `alpha`, `beta`, `alpha0`, `iwu`, `th`, `isu`) into a control-file copy. Each parameter has a documented physically-motivated range (e.g., `wm`$\in[0.1,\ 10.0]$, `im`$\in[0,1]$); the full list of bounds and physical interpretations is given in Appendix B.
- `run_simulation`: invoke the EF5 binary on the patched control file and parse the resulting time-series CSV, returning the full simulated/observed discharge series at hourly resolution along with a summary of multiple hydrologic-fit signatures: peak-flow magnitude error and timing offset, total-volume ratio (a water-balance closure indicator), recession-limb slope, time-to-peak, baseflow level, and event count. These criteria are exposed as text the LLM interprets jointly—a peak under-prediction with a balanced volume ratio implies a routing-velocity issue, while a balanced peak with a volume surplus implies an evapotranspiration or impervious-fraction issue.
- `evaluate`: aggregate the latest simulation against the gauge time series. Returns the Nash–Sutcliffe Efficiency [Nash and Sutcliffe, 1970], $\mathrm{NSE} = 1 - \sum_t (Q_t^{\mathrm{obs}} - Q_t^{\mathrm{sim}})^2 / \sum_t (Q_t^{\mathrm{obs}} - \overline{Q^{\mathrm{obs}}})^2$, alongside a panel of complementary metrics—Kling–Gupta Efficiency components (correlation, variability ratio, bias ratio), root-mean-square error, percent bias, and high-/low-flow KGE—together with the running best-NSE and target-status. The agent therefore reasons over a *multi-criteria* diagnostic rather than a single scalar; NSE is the headline reward signal but the auxiliary metrics are what disambiguate which physical process is the next thing to adjust.
- `parse failure`: understands the difference between model simulated and observed streamflow from its metrics: CC, KEG, NSE, peak_error, time_lag. Reason the root cause and provide possible reasons. An example of the reasoning output is *"The model has stabilized with an NSE of 0.2248, CC of 0.591, and KGE of 0.5588, indicating a well-calibrated simulation that accurately captures the observed hydrograph in terms of timing and volume. Although the peak discharge (374.93) is still below the observed value (622.97), the timing error (lag = 32 hours) is minimal, and the model's performance across all metrics is consistent and physically realistic. Further improvements would require adjusting channel routing or impervious fraction."*

The agent receives the basin descriptor, the gauge directory, the precipitation/PET raster directories, the control-file template, and the evaluation window, and must improve NSE over up to 50 multi-turn iterations. Episodes **terminate** on target attainment, on five rounds without improvement, or on the per-episode wall-clock budget. We use 10 CONUS gauges spanning 539–$2401\,\mathrm{km}^2$ (selected from the audit pool described in Appendix A) for training, and four held-out gauges (ranging from 329 to 40,792 km2) for evaluation, matching the protocol used in our frontier-model benchmark (Section 3). We adopt the widely used hydrologic performance rubric of Moriasi et al. [2015], which classifies streamflow simulations as *unsatisfactory* ($\mathrm{NSE} \le 0.50$), *satisfactory* ($0.50 < \mathrm{NSE} \le 0.70$), *good* ($0.70 < \mathrm{NSE} \le 0.85$), and *very good* ($\mathrm{NSE} > 0.85$); the 0.8075 target therefore sits at the top of the *good* band and is the threshold an experienced operational hydrologist routinely meets on this gauge.

2.3 Stage 1: Supervised Fine-Tuning

The base model is `Qwen3-4B-Instruct-2507` [Qwen Team, 2025], chosen because it is the largest open-weight model that fits comfortably in BF16 with full FSDP fine-tuning and $K{=}8$ multi-turn rollouts on $4{\times}$H100 (80 GB), and because it ships with native Hermes-style tool calling. We construct $2{,}576$ SFT trajectories by distilling a strong proprietary teacher (GPT-5 [1]) on 73 calibration runs across 29 U.S. gauges (excluding the 4 gauges in the test set). Crucially, each teacher run is conditioned on a *full one-year hydrograph*, not on an isolated flood event: the year-long window contains multiple storm events, dry-season recessions, and seasonal soil-moisture transitions, so each calibration run is a multi-objective task in which the teacher must balance peak magnitude, timing, recession shape, and baseflow simultaneously. This is a substantially harder distillation target than the single-event windows used in most prior LLM-agent benchmarks in hydrology (e.g., AQUAH [Yan

[1] Due to budget constraints, we use GPT-5 for initial agent development and trajectory collection, while evaluating the four test gauges using nine latest frontier models.

et al., 2025]). The individual iterative episodes within each calibration run are then reorganized into a single long-horizon trajectory — preserving the full hypothesise → simulate → diagnose → adjust sequence in its original tool-call order — so that the model learns the chain-of-thought of iterative parameter refinement rather than only the converged parameter set. Trajectories whose final NSE does not exceed 0.6 are dropped, and the surviving sequences are quality-weighted by their final-NSE percentile. SFT teaches the model the tool-call grammar, the parameter-bounds convention, and a basic hydrologic-reasoning style ("recession limb is too steep ⇒ increase `wm` or `leaki`").

2.4 Stage 2: Reinforcement Learning with Simulation Feedback (RLSF)

Problem formulation. We cast hydrologic-model calibration as a sequential decision problem $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ in which the state s_t is the tuple $\langle \theta_t, \hat{Q}_t, m_t \rangle$ (parameter vector, simulator discharge, multi-criteria diagnostic), actions are tool calls drawn from a closed set (`set_parameters`, `run_simulation`, `evaluate`), the transition operator is realized by the physical solver itself, and the reward is a verifiable, continuous physical-error metric produced by that solver. Three properties distinguish this formulation from prior verifiable-reward RL on math/code reasoning [Lambert et al., 2024, DeepSeek-AI, 2025, Gehring et al., 2024, Wei et al., 2025]: the verifier is a numerical Earth-system simulator producing a continuous physical-quality signal rather than a binary correctness check; episodes are long-horizon (≤ 50 tool calls) with delayed terminal feedback; and the reward is cheap-to-evaluate but noisy. These properties motivate a critic-free, group-relative policy-optimization scheme; we instantiate it with GRPO and refer to the resulting recipe as *Reinforcement Learning with Simulation Feedback* (RLSF), distinguishing it from human-feedback RL [Lee et al., 2024] and pure verifiable-text-reward RL.

GRPO instantiation. We instantiate the formulation with GRPO [Shao et al., 2024]: for each prompt we draw $K{=}8$ rollouts, each a multi-turn calibration episode invoking up to 50 tool calls; the within-group mean and standard deviation of the rollout returns serve as the advantage baseline, replacing the value-function critic that PPO would otherwise require. The reward decomposes into per-turn shaping and a terminal score:

$$r_t^{\text{turn}} = \begin{cases} +0.02 & \text{valid } \texttt{set\_parameters} \\ +0.05 & \text{valid } \texttt{run\_simulation} \\ \Delta\text{NSE}_t & \text{valid } \texttt{evaluate} \\ -0.5 & \texttt{parse\_failure} \text{ (no tool call)} \end{cases}$$

$$\begin{aligned} r^{\text{terminal}} = {} & \text{clip}(\text{NSE}^\star, -1, 1) + 0.5 \cdot \mathbf{1}\{\text{NSE}^\star > \tau\} + 0.02\, n_{\text{eval}} \\ & + 0.10 \max(0,\, n_{\text{improve}} - 1) - 1.0 \cdot \mathbf{1}\{\text{empty}\} \end{aligned}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function (taking value 1 when the bracketed condition holds and 0 otherwise), ΔNSE_t is the change in best-NSE on turn t, $\text{NSE}^\star$ is the episode-best NSE, τ is the gauge-specific target, n_{eval} counts valid `evaluate` calls, n_{improve} counts evaluations that beat the running best, and an episode is *empty* when the agent never produces a parseable evaluation. The improvement bonus is the key inductive bias: it explicitly trains *iterate-until-you-cannot-improve*, which is what a hydrologist actually does.

We use full BF16 fine-tuning under FSDP (no LoRA), with actor learning rate $1{\times}10^{-6}$, KL anchor coefficient 0.2 to the SFT initialization, entropy coefficient 0.01, sampling temperature 1.0 with top-p 0.95, $\text{batch} = 4$ prompts $\times K{=}8$ rollouts per step, and 30 epochs over the 10-gauge training set. The strong KL anchor is necessary: lower values let the policy drift to token-level degenerate outputs by step $\sim$40. Rollouts are served by SGLang in synchronous mode with native multi-turn tool dispatch; EF5 invocations are gated by a 32-way semaphore to prevent CPU/IO contention. Training takes about 5 hours per checkpoint cadence on $4{\times}$H100. Full Hydra configurations are in Appendix A.

3 AI Agents for Hydrologic Modeling: Where Do We Stand?

Hydrologic calibration is a stringent test of agentic scientific reasoning: the task is long-horizon, simulator-grounded, multi-objective, and physically constrained. Effective calibration requires sustained iterative refinement over delayed feedback rather than one-shot reasoning. We therefore evaluate nine frontier LLM agents on a geographically held-out EF5 calibration benchmark to measure both final performance (§3.2) and long-horizon engagement (§3.3). The difficulty turns out

to be structural rather than vendor-specific: the same gap appears across all three frontier families (Anthropic, OpenAI, Google), across both flagship and *pro*-tier reasoning variants, and across both fast and slow speed/quality tiers — no model in our nine-agent panel reaches the human-expert reference on more than one of the four held-out gauges, despite each basin sitting well within the achievable range. The frontier ceiling we report is therefore a property of the task, not of any one model.

3.1 Benchmark setup

We evaluate nine frontier LLM agents on the calibration of the four held-out gauges of Table 2 (drainage areas 329–40,792 km^2 across distinct hydroclimatic regions of the CONUS). Each agent runs each gauge (§3.2) independently inside the same Linux sandbox under the *terminal-2* agent harness in the *harbor* evaluation framework. The harness exposes a single Bash terminal: the agent reads the per-gauge task brief (basin descriptor, parameter table with bounds, EF5 invocation contract, NSE target), inspects the data directory, edits the control file, runs `ef5 /app/control.txt`, parses the simulated-versus-observed CSV, and iterates. Each round consists of up to 10 candidate parameter sweeps; the budget is 20 rounds per gauge (i.e., up to 200 EF5 simulations per gauge, 800 across the panel) and we track the running-best NSE per round per gauge. Figure 2 visualizes the per-model best-NSE bars on all four held-out gauges, with the canonical evaluation gauge `02338660` (Figure 2a) used as the discussion anchor in Section 3.2 below. Hydrologic-model calibration is a *long-horizon* task: in our pilot runs, 20 rounds is the lower end of what suffices to reach the target on a typical gauge, and the multi-criteria diagnostic the agent must integrate (Section 2.2) only becomes informative after several rounds of trial parameter moves. Whether each agent actually *uses* that horizon is itself an outcome of the experiment, reported alongside each best-NSE bar in Figure 2.

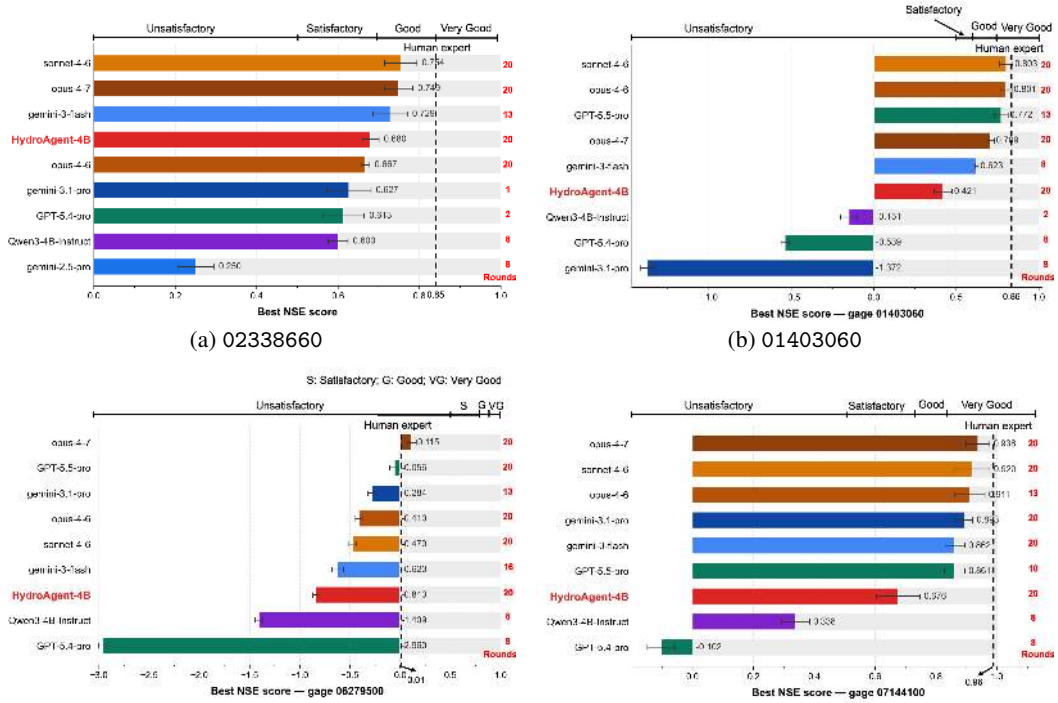
The nine evaluated models are: **Anthropic** Claude Opus 4.6, Opus 4.7, and Sonnet 4.6; **OpenAI** GPT-5, GPT-5.4, and GPT-5.4-pro; **Google** Gemini 2.5-pro, Gemini 3.1-pro, and Gemini 3-flash. The choice is intended to span the three frontier families and to cover the speed/quality variants currently in production (April 2026). All models are accessed through the harness's default API integration with default sampling temperatures.

Benchmark-collection cost. Constructing this benchmark is itself a non-trivial engineering effort and is an independent contribution. The training-time SFT corpus is collected on *full one-year hydrographs* (Section 2.3), forcing the teacher to balance multi-event peak magnitude, timing, recession shape, and baseflow within a single calibration — to our knowledge the first hydrology LLM-agent benchmark on year-long windows. End-to-end, building the artifact required staging multi-tens-of-GB of hourly MRMS, daily PET, DEM, and CREST raster forcings into a Linux sandbox (Appendix A); constructing a DREAM/ABC [Vrugt and Sadegh, 2013] human-expert reference per held-out gauge; running $\approx$28,800 closed-loop EF5 calls on the test side (9 models $\times 4$ gauges $\times 4$ seeds $\times \leq 200$ simulations) plus their multi-turn LLM API traffic; and curating the 73 year-long teacher calibrations that yield the SFT corpus. The resulting trajectory dataset and harness are released for reproducibility.

3.2 Results

Figure 2 reports the best-NSE attained by each model on each of the four held-out gauges, alongside its typical rounds-used out of the 20-round budget. We anchor the discussion on the canonical evaluation gauge `02338660` (Figure 2a) and then survey how the picture varies across the other three panels (Figures 2b, 2c, 2d).

Anchor gauge `02338660`. Mapped onto the Moriasi et al. [2015] performance rubric (visualized as bands in the figure), two models reach the *good* band ($0.70 < \text{NSE} \leq 0.85$): Sonnet 4.6 (0.754, 20 rounds) and Opus 4.7 (0.749, 20 rounds), with Gemini 3-flash (0.729, 13 rounds) at the same level; three more land in *satisfactory* ($0.50 < \text{NSE} \leq 0.70$) — Opus 4.6 (0.667, 20 rounds), Gemini 3.1-pro (0.627, 1 round), and GPT-5.4-pro (0.613, 2 rounds); and the remainder are *unsatisfactory* ($\text{NSE} \leq 0.50$), with Gemini 2.5-pro (0.250, 8 rounds) marginal and GPT-5 (-0.189, 3 rounds) and GPT-5.4 (-0.159, 1 round) failing to produce a positive NSE within budget. On this gauge no model crosses the human-expert reference at $\text{NSE}=0.85$, despite the basin sitting well within the achievable range. The reference is produced by an experienced hydrologist who pairs domain knowledge of



(a) 02338660 (b) 01403060

(c) 06279500 (d) 07144100

Figure 2: Best-of-twenty-rounds Nash–Sutcliffe Efficiency (NSE) across the four held-out gauges of Table 2; GPT-5 and GPT-5.4 are omitted from the bars because both failed to reach a positive NSE on any gauge within budget. Performance bands across the top of each panel follow Moriasi et al. [2015] (*Unsatisfactory*, *Satisfactory*, *Good*, *Very Good*); the dashed line denotes the human-expert reference produced by an experienced hydrologist with the DREAM approximate-Bayesian-computation sampler of Vrugt and Sadegh [2013]. Error bars are ± 1 SD over four seeds per model. Across the four-gauge panel, no frontier model crosses the human-expert reference on more than one gauge — only Opus-4.7 reaches it, and only on gauge 06279500, the largest basin in the panel and a documented difficult case (Appendix A).

CREST parameter sensitivities with the DREAM approximate-Bayesian-computation sampler [Vrugt and Sadegh, 2013] to calibrate the same gauges.

Cross-gauge consistency. Three patterns reproduce across the four-gauge panel. The qualitative ranking is preserved: Sonnet 4.6 and Opus 4.7 retain the top slots on every gauge with a positive NSE, while GPT-5 and GPT-5.4 fail to reach positive NSE on any gauge. The rounds-used / best-NSE coupling is basin-invariant: *good*-band models consume close to the full 20-round budget on every gauge, and pro-tier reasoning models (Gemini 3.1-pro, GPT-5.4-pro) continue to self-terminate within 1–2 rounds. Basin difficulty re-scales but does not invert the band assignments: only Opus-4.7 crosses the human-expert reference, and only on the largest basin 06279500 (a documented difficult case, Appendix A). The frontier ceiling at NSE ≈ 0.75 is therefore a domain-grounding ceiling, not a basin-specific artifact.

Best-NSE and rounds-used are tightly correlated: the three *good*-band models average 17.7 rounds on 02338660, while the rest average 5.5 and the pro-tier reasoning models from OpenAI and Google terminate after one or two rounds with budget remaining. Calibration is a long-horizon task whose reward sharpens only with iteration, and a class of frontier models is policy-trained to stop early.

3.3 What goes wrong?

Three failure modes recur. *Premature termination*: pro-tier reasoning models stop after a handful of rounds despite the 20-round budget — Gemini 3.1-pro after 1, GPT-5.4-pro after 2, GPT-5.4 after 1 — walking away from a clearly improvable hydrograph. *Out-of-bounds proposals*: weaker models

Table 1: Per-gauge ablation of the SFT + RLSF training stack on the four held-out gauges. Higher NSE is better.

Gauge	Method	Best NSE	Sims	Turns	Parse fail
07144100	Baseline	0.34	15	50	0
	SFT-only	0.07	1	15	4
	HydroAgent	0.65	13	50	0
06279500	Baseline	−1.41	13	50	1
	SFT-only	−2.27	1	13	4
	HydroAgent	−0.84	17	50	0
02338660	Baseline	0.65	13	50	0
	SFT-only	−17.53	1	11	4
	HydroAgent	0.68	12	50	0
01403060	Baseline	−0.15	16	50	0
	SFT-only	0.58	4	16	4
	HydroAgent	0.40	14	50	0

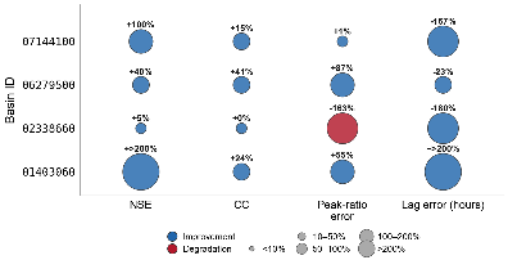


Figure 3: Relative change in four hydrologic-fit metrics for HYDROAGENT-4B versus the Qwen3-4B-Instruct baseline on the four held-out gauges. Bubble color encodes direction and size encodes magnitude. For |Lag|, smaller is better.

propose parameter values outside documented physical ranges (e.g., negative `im`, `wm`>10), wasting EF5 calls. *Diagnosis–action mismatch*: models correctly identify, e.g., an over-pronounced recession, but propose adjustments to channel-routing parameters rather than to the soil-moisture/leakage parameters that drive recession shape. We interpret these as failures of *calibration-relevant grounding* and *long-horizon engagement*, not of base reasoning capacity. Even the strongest models (Sonnet 4.6, Opus 4.7) avoid the first two but still exhibit the third. The bottleneck is twofold — an agent must (a) keep iterating across many rounds and (b) know which parameter to move when — exactly the gap an agent post-trained with simulator-in-the-loop RL on long multi-turn calibration episodes is designed to close.

4 How does HydroAgent improve streamflow simulation?

4.1 SFT + RLSF improves the metrics that matter

We evaluate HYDROAGENT-4B—Qwen3-4B-Instruct fine-tuned with SFT on 2,576 expert trajectories (Section 2.3) and then with GRPO under simulator-feedback rewards (Section 2.4)—against its untuned base on the four held-out gauges of Table 1. Both agents share the same *terminal-2* harness in the *harbor* framework, the same 20-round budget, and the same task brief; only the policy weights differ. Figure 3 reports the relative change in four hydrologic-fit metrics across this panel.

The pattern is consistent: simulator-grounded post-training improves the metrics calibration is judged by. *Nash–Sutcliffe Efficiency* improves on all four basins, from $+5\%$ on the canonical test gauge `02338660` to over $+200\%$ on `01403060` (where the baseline was deeply negative and is now positive). *Discharge correlation* improves on all four basins, by 0 to $+41\%$. *Peak-flow ratio error* improves on three of four basins ($+1\%$ to $+87\%$, with a -163% degradation on `02338660`). *Timing offset* (|Lag|, where smaller is better) likewise contracts on all four basins, by $+23\%$ to over $+200\%$. The pattern is consistent with Section 2.4: the per-turn ΔNSE shaping and the volume/peak-weighted multi-criteria diagnostic surfaced by `run_simulation` (Section 2.2) jointly steer the policy to first stabilize magnitude metrics and then align peak placement, so that absolute timing error shrinks together with NSE and correlation rather than at their expense. The one residual cost mode is peak-ratio error on `02338660` (-163%), where compressing the late-recession volume bias necessarily redistributes flow across event peaks; we return to this trade-off in the limitations Paragraph below.

5 Related Work

LLMs and foundation models for Earth science. A first wave pre-trains domain-specialized text or vision models for the Earth sciences — e.g., K2 [Deng et al., 2024], GeoGalactica [Lin et al., 2024], ClimateGPT [Thulke et al., 2024], OceanGPT [Bi et al., 2024], and EarthGPT [Zhang et al., 2024]. A second wave builds neural surrogates of physical systems (Pangu-Weather [Bi et al., 2023], GraphCast [Lam et al., 2023], FourCastNet [Pathak et al., 2022], GenCast [Price et al., 2024], ClimaX [Nguyen et al., 2023], Aurora [Bodnar et al., 2025]), and Nearing et al. [2024] pushes a pure-LSTM

rainfall-runoff regressor to global-scale ungauged-flood prediction. Our setting differs from both waves: we neither build a domain knowledge model nor replace the simulator with a learned surrogate — we keep the operational physics-based simulator (CREST) and ask an LLM agent to *operate* it.

LLM agents for scientific simulation and discovery. Coscientist [Boiko et al., 2023], ChemCrow [M. Bran et al., 2024], FunSearch [Romera-Paredes et al., 2024], The AI Scientist [Lu et al., 2024], SciAgents [Ghafarollahi and Buehler, 2025], and AutoGen [Wu et al., 2023] couple LLMs to tools, lab instruments, or evolutionary outer loops for autonomous discovery; benchmarks such as MLAgentBench [Huang et al., 2024] and ScienceAgentBench [Chen et al., 2025] measure this class of agent on data-driven tasks (best agents solve 32–42%). AQUAH [Yan et al., 2025], the closest prior work in hydrology, is vision-enabled but prompt-only at the LLM core. We complement these by (i) measuring frontier agents on a closed-loop physical-simulator task with quantitative streamflow rewards, and (ii) showing that fine-tuning the LLM with simulator-grounded RL narrows a substantial part of the gap.

RL fine-tuning with verifiable / environment feedback. Recent work shows that scaling RL with rule-based or environment-based rewards elicits strong reasoning in open LLMs: DeepSeek-R1 [DeepSeek-AI, 2025] and Kimi k1.5 [Kimi Team, 2025] train pure-RL reasoners with verifier rewards; Tülu 3 [Lambert et al., 2024] formalizes Reinforcement Learning with Verifiable Rewards (RLVR); RLEF [Gehring et al., 2024] and SWE-RL [Wei et al., 2025] ground code-LLM RL in unit-test or software-evolution feedback; GRPO [Shao et al., 2024] supplies the critic-free, group-normalized advantage objective we adopt. A parallel line on multi-turn agent training and tool use [Wang et al., 2025, Zhou et al., 2024, Shinn et al., 2023, Yao et al., 2023, Qin et al., 2024, Gou et al., 2024, Schick et al., 2023, Wang et al., 2024, Ma et al., 2024] and on LLM-judge feedback [Lee et al., 2024] addresses related but distinct settings. Ours differs in the verifier: a numerical hydrologic simulator producing continuous physical-error metrics (NSE), with monotonically improving NSE as the convergence signal. Our infrastructure builds on `verl`/HybridFlow [Sheng et al., 2024].

Hydrologic modeling and calibration. The dominant deep-learning thread in hydrology trains LSTMs on CAMELS [Addor et al., 2017] or Caravan [Kratzert et al., 2023] to regress streamflow directly [Kratzert et al., 2018, 2019, 2022]; a complementary differentiable-physics thread embeds neural networks inside process-based models [Feng et al., 2022, Shen et al., 2023]. Classical calibration of non-differentiable simulators relies on derivative-free global optimizers and Bayesian samplers — SCE-UA [Duan et al., 1992], DDS [Tolson and Shoemaker, 2007], PEST [Doherty, 2015], and DREAM [Vrugt and Sadegh, 2013] — which treat each calibration as a black box without cross-basin or domain-textual context. We provide an LLM-agent baseline along the same axis and argue that the agent's pre-trained domain priors make it a competitive complement when each simulator call is expensive.

6 Discussion and Conclusion

Limitations and future work. Our four-gauge CONUS panel is too small to defend cross-regime generalization. Three directions follow. *First*, scaling evaluation to a global dataset such as Caravan [Kratzert et al., 2023] would test transfer to hydroclimatic regimes (monsoon, alpine, polar) absent from our training panel. *Second*, the recipe has been demonstrated only on CREST and Qwen3-4B; transfer to other distributed hydrologic models and other small open backbones is open. *Third*, the scalar NSE comparator could be replaced with a *vision-language verifier* reading the rendered hydrograph directly — hydrologists reason about curve shape, not a single number, and a shape-aware critic should reduce NSE's known pathologies (peak dominance, single-event sensitivity) and likely close the residual peak-ratio gap on `02338660` (Figure 3). Over-reliance on an AI calibration agent also risks propagating mis-specified parameters into operational forecasts, triggering false flash-flood warnings or misallocating emergency-response resources — a case for a human-in-the-loop reading of the simulator-grounded reward as a verifier, not a substitute for expert judgment.

Conclusion. Frontier LLM agents plateau at panel-mean NSE 0.65–0.75 on hydrologic-model calibration — a domain-grounding ceiling, not a scaling one. Post-training a 4B-parameter open base with SFT and GRPO under a simulator-grounded reward (HYDROAGENT) narrows that gap on every held-out basin (panel-mean NSE $-0.14 \rightarrow +0.20$); for Earth-system tasks with cheap physical simulators, a small domain-tuned policy with simulator-in-the-loop RL is a more compute-efficient and physically faithful path than scaling generic frontier models.

References

Nans Addor, Andrew J Newman, Naoki Mizukami, and Martyn P Clark. The CAMELS data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, 21(10):5293–5313, 2017.

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

Zhen Bi, Ningyu Zhang, Yida Xue, Yixin Ou, Daxiong Ji, Guozhou Zheng, and Huajun Chen. OceanGPT: A large language model for ocean science tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Cristian Bodnar, Wessel P Bruinsma, Ana Lucic, Megan Stanley, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A Weyn, Haiyu Dong, Anna Vaughan, et al. A foundation model for the Earth system. *Nature*, 641:1180–1187, 2025.

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

Ziru Chen, Shijie Chen, Michael Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations (ICLR)*, 2025.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

John Doherty. PEST: Model-independent parameter estimation, user manual (6th edition). Technical report, Watermark Numerical Computing, 2015.

Qingyun Duan, Soroosh Sorooshian, and Vijai Gupta. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research*, 28(4):1015–1031, 1992.

Dapeng Feng, Jiangtao Liu, Kathryn Lawson, and Chaopeng Shen. Differentiable, learnable, regionalized process-based models with multiphysical outputs can approach state-of-the-art hydrologic prediction accuracy. *Water Resources Research*, 58(10):e2022WR032404, 2022.

Zachary L Flamig, Humberto Vergara, and Jonathan J Gourley. The ensemble framework for flash flood forecasting (EF5) v1.2: description and case study. *Geoscientific Model Development*, 13 (10):4943–4958, 2020.

Jonas Gehring, Kunhao Zheng, Jade Copet, Vegard Mella, Quentin Carbonneaux, Taco Cohen, and Gabriel Synnaeve. RLEF: Grounding code LLMs in execution feedback with reinforcement learning. *arXiv preprint arXiv:2410.02089*, 2024.

Alireza Ghafarollahi and Markus J Buehler. SciAgents: Automating scientific discovery through bioinspired multi-agent intelligent graph reasoning. *Advanced Materials*, 37(9):2413523, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *International Conference on Learning Representations (ICLR)*, 2024.

Jonathan J Gourley, Zachary L Flamig, Humberto Vergara, Pierre-Emmanuel Kirstetter, Robert A Clark III, Elizabeth Argyle, Ami Arthur, Steven Martinaitis, Galateia Terti, Jessica M Erlingis, Yang Hong, and Kenneth W Howard. The FLASH project: Improving the tools for flash flood monitoring and prediction across the United States. *Bulletin of the American Meteorological Society*, 98(2):361–372, 2017.

Qian Huang, Jacky Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

Frederik Kratzert, Daniel Klotz, Claire Brenner, Karsten Schulz, and Mathew Herrnegger. Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022, 2018.

Frederik Kratzert, Daniel Klotz, Guy Shalev, Günter Klambauer, Sepp Hochreiter, and Grey Nearing. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, 23(12):5089–5110, 2019.

Frederik Kratzert, Martin Gauch, Grey Nearing, and Daniel Klotz. NeuralHydrology—a Python library for deep learning research in hydrology. *Journal of Open Source Software*, 7(71):4050, 2022.

Frederik Kratzert, Grey Nearing, Nans Addor, Tyler Erickson, Martin Gauch, Oren Gilon, Lukas Gudmundsson, Avinatan Hassidim, Daniel Klotz, Sella Nevo, et al. Caravan—a global community dataset for large-sample hydrology. *Scientific Data*, 10(1):61, 2023.

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

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Zhi Li, Mengye Chen, Shang Gao, Jonathan J Gourley, Tiantian Yang, Xinyi Shen, Pierre Kirstetter, and Yang Hong. A multi-source 120-year US flood database with a unified common format and public access. *Earth System Science Data Discussions*, 2021:1–25, 2021.

Zhi Li, Shang Gao, Mengye Chen, Jonathan J Gourley, Changjie Liu, Andreas F Prein, and Yang Hong. The conterminous United States are projected to become more prone to flash floods in a high-end emissions scenario. *Communications Earth & Environment*, 3(1):86, 2022.

Zhi Li, Xianwu Xue, Robert Clark, Humberto Vergara, Jonathan J Gourley, Guoqiang Tang, Xinyi Shen, Guangyuan Kan, Ke Zhang, Jiahu Wang, Mengye Chen, Shang Gao, Jiaqi Zhang, Tiantian Yang, Yixin Wen, Pierre Kirstetter, and Tiantian Hong. A decadal review of the CREST model family: Developments, applications, and outlook. *Journal of Hydrology X*, 21:100159, 2023.

Zhouhan Lin, Cheng Deng, Le Zhou, Tianhang Zhang, Yi Xu, Yutong Xu, Zhongmou He, Yuanyuan Shi, Beiya Dai, Yunchong Song, et al. GeoGalactica: A scientific large language model in geoscience. *arXiv preprint arXiv:2401.00434*, 2024.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

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*, 6:525–535, 2024.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Thomas K J McDermott. Global exposure to flood risk and poverty. *Nature Communications*, 13(1): 3529, 2022.

Daniel N Moriasi, Margaret W Gitau, Naresh Pai, and Prasad Daggupati. Hydrologic and water quality models: Performance measures and evaluation criteria. *Transactions of the ASABE*, 58(6): 1763–1785, 2015. doi: 10.13031/trans.58.10715.

Muhammad Akhtar Munir, Muhammad Umer Sheikh, Akashah Shabbir, Muhammad Haris Khan, Fahad Khan, Xiao Xiang Zhu, Begum Demir, and Salman Khan. Agentic AI for remote sensing: Technical challenges and research directions. *arXiv preprint arXiv:2604.24919*, 2026.

J Eamonn Nash and J Vasilis Sutcliffe. River flow forecasting through conceptual models part I—a discussion of principles. *Journal of Hydrology*, 10(3):282–290, 1970.

Grey Nearing, Deborah Cohen, Vusumuzi Dube, Martin Gauch, Oren Gilon, Shaun Harrigan, Avinatan Hassidim, Daniel Klotz, Frederik Kratzert, Asher Metzger, et al. Global prediction of extreme floods in ungauged watersheds. *Nature*, 627:559–563, 2024.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. ClimaX: A foundation model for weather and climate. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

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

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Probabilistic weather forecasting with machine learning. *Nature*, 637:84–90, 2024.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations (ICLR)*, 2024.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jun Rentschler, Melda Salhab, and Bramka Arga Jafino. Flood exposure and poverty in 188 countries. *Nature Communications*, 13(1):3527, 2022.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, 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.

Chaopeng Shen, Alison P Appling, Pierre Gentine, Toshiyuki Bandai, Hoshin Gupta, Alexandre Tartakovsky, Marco Baity-Jesi, Fabrizio Fenicia, Daniel Kifer, Li Li, et al. Differentiable modelling to unify machine learning and physical models for geosciences. *Nature Reviews Earth & Environment*, 4:552–567, 2023.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*, 2024.

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

David Thulke, Yingbo Gao, Petrus Pelser, Rein Brune, Richa Jalota, Floris Fok, Michael Ramos, Ian van Wyk, Abdallah Nasir, Hayden Goldhahn, et al. ClimateGPT: Towards AI synthesizing interdisciplinary research on climate change. *arXiv preprint arXiv:2401.09646*, 2024.

Bryan A Tolson and Christine A Shoemaker. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources Research*, 43(1):W01413, 2007.

Jasper A Vrugt and Mojtaba Sadegh. Toward diagnostic model calibration and evaluation: Approximate Bayesian computation. *Water Resources Research*, 49(7):4335–4345, 2013. doi: 10.1002/wrcr.20354.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research (TMLR)*, 2024.

Jiahu Wang, Yang Hong, Li Li, Jonathan J Gourley, Sadiq I Khan, Koray K Yilmaz, Robert F Adler, Frederick S Policelli, Shahid Habib, Daniel Irwn, et al. The coupled routing and excess storage (CREST) distributed hydrological model. *Hydrological Sciences Journal*, 56(1):84–98, 2011.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Heng Chen, et al. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. SWE-RL: Advancing LLM reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Xianwu Xue, Yang Hong, Ashutosh S Limaye, Jonathan J Gourley, George J Huffman, Sadiq Ibrahim Khan, Chhimi Dorji, and Sheng Chen. Statistical and hydrological evaluation of TRMM-based multi-satellite precipitation analysis over the Wangchu basin of Bhutan: Are the latest satellite precipitation products 3B42V7 ready for use in ungauged basins? *Journal of Hydrology*, 499: 91–99, 2013.

Songkun Yan, Zhi Li, Siyu Zhu, Yixin Wen, Mofan Zhang, Mengye Chen, Jie Cao, and Yang Hong. AQUAH: Automatic quantification and unified agent in hydrology. *arXiv preprint arXiv:2508.02936*, 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 *International Conference on Learning Representations (ICLR)*, 2023.

Wei Zhang, Miaoxin Cai, Tong Zhang, Yin Zhuang, and Xuerui Mao. EarthGPT: A universal multi-modal large language model for multi-sensor image comprehension in remote sensing domain. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–20, 2024.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

A Reproducibility details

gauge configurations. The 10 training gauges and the four held-out test gauges, with their basin areas, evaluation windows, and provenance notes, are listed in Table 2; their geographic distribution is shown in Figure 4. The split between training and testing gauges is a random partition of the audit pool, not stratified by basin size or region, so that any cross-basin generalization observed in Sections 3 and 4 reflects the agent rather than a hand-curated train/test alignment. Training

gauges were selected by sliding a 60-day window over each gauge's hourly observation series and scoring by $\log_{10}(Q_{\text{peak}}/\overline{Q}+1) \times \sqrt{t_{\text{rise}}\, t_{\text{recess}}}$ to favour clean flood events. The held-out test set is intentionally diverse in scale—spanning $329\,\text{km}^2$ to $40{,}792\,\text{km}^2$—to stress-test cross-basin generalization; `02338660` is the canonical evaluation gauge used in Sections 3 and 4, while `01403060`, `06279500`, and `07144100` are reserved for the held-out panel referenced in Section 4 and will be evaluated once the corresponding MRMS/PET clip is built.

Table 2: gauge configurations used in this study. The 10 training gauges span 539–$2{,}401\,\text{km}^2$ across diverse hydroclimatic regions of the CONUS; four held-out gauges, ranging from 329 to $40{,}792\,\text{km}^2$, form the testing set used only for evaluation in Sections 3 and 4. Of the four test gauges, only `02338660` ships in the current data tarball; the other three are reserved for the held-out evaluation panel referenced in Section 4 and require an additional MRMS/forcing build to run.

Split	gauge ID	Basin (km^2)	Window (UTC)
Train	`11383500`	539	2018-05-19 → 2018-07-17
	`11043000`	575	2019-03-15 → 2019-05-13
	`11152000`	632	2018-05-29 → 2018-07-27
	`02294781`	1,064	2018-04-29 → 2018-06-27
	`02312000`	1,476	2018-11-15 → 2019-01-13
	`07195430`	1,489	2018-01-04 → 2018-03-04
	`11179000`	1,639	2018-06-03 → 2018-08-01
	`14301000`	1,727	2018-09-11 → 2018-11-09
	`14207500`	1,828	2018-04-09 → 2018-06-07
	`11376000`	2,401	2018-09-21 → 2018-11-19
Test	`02338660`	329	2018-07-01 → 2018-08-31
	`01403060`	2,033	2018-11-11 → 2019-01-09
	`06279500`	40,792	2018-06-13 → 2018-08-11
	`07144100`	3,209	2019-03-30 → 2019-05-28

Figure 4: Geographic distribution of the gauges used in this study across the conterminous United States. Training gauges (10) span basins from the Pacific Northwest, California, the Southwest, the Midwest, the Southeast, and the Atlantic seaboard, covering a wide range of climatic regimes (Mediterranean, semi-arid, humid subtropical, humid continental); the held-out test gauge `02338660` sits in the humid-subtropical Southeast. The geographic spread is intentional: it stresses the agent's ability to generalize across distinct hydroclimatic regions rather than within a single region's flow regime.

Forcings and static grids. CREST/EF5 inputs include hourly MRMS gauge-corrected precipitation GeoTIFFs, daily PET GeoTIFFs, a CONUS-wide DEM, flow-direction and flow-accumulation rasters,

and CREST/KW default parameter rasters (`wm`, `b`, `im`, `ksat`, α, β, `leaki`, α_0). All scalar multipliers documented in Section 2.2 apply to these grids.

GRPO hyperparameters. The Hydra-style configuration that overlays `verl`'s `ppo_trainer` defaults to produce our GRPO recipe is summarized in Table 3; one training step takes approximately 30 min of wall time on $4\times$H100 80 GB and the run checkpoints every $\sim$5 h.

Table 3: GRPO training configuration for HYDROAGENT-4B. All keys are Hydra paths layered on top of `verl`'s `ppo_trainer` defaults; rollouts are served by SGLang in synchronous mode with native multi-turn tool dispatch (Hermes format).

Group	Key / setting	Value / note
Algorithm	`algorithm.adv_estimator`	`grpo`
	`kl_loss_coef`	0.2 (`low_var_kl`) anchor to SFT init
	`entropy_coeff`	0.01
Optimizer	`actor.optim.lr`	1×10^{-6}
	`lr_warmup_steps_ratio`	0.05
	Precision	BF16
	Sharding	FSDP, no LoRA
Batching	`data.train_batch_size`	4 prompts
	`ppo_mini_batch_size`	4
	`data.max_response_length`	4096 tokens
Rollout	`rollout.n`	$K = 8$ rollouts per prompt
	`rollout.temperature`	1.0
	`rollout.top_p`	0.95
	`multi_turn.max_assistant_turns`	50
Trainer	`trainer.total_epochs`	30
	`trainer.save_freq`	10 steps
	`trainer.test_freq`	25 steps
Hardware	GPUs	4×H100, 80 GB
	EF5 concurrency	32 (per-worker semaphore $\times$ 4 workers)

SFT data. 2,576 long-horizon SFT trajectories distilled from 73 GPT-5 calibration runs across 29 U.S. gauges. Each trajectory preserves the full iterative *hypothesise* $\rightarrow$ *simulate* $\rightarrow$ *diagnose* $\rightarrow$ *adjust* sequence from one calibration run in its original tool-call order, so that the model learns the chain-of-thought of iterative parameter refinement; on average each run yields $\sim$35 usable trajectory variants after augmenting at valid stopping points along the calibration. Trajectories whose final NSE does not exceed 0.6 are dropped, and the surviving set is quality-weighted by final-NSE percentile.

Evaluation harness (frontier and HYDROAGENT). All evaluations reported in this paper—both the nine frontier agents in Section 3 and the HYDROAGENT-4B checkpoint in Section 4—are conducted under the same *terminal-2* agent harness in the *harbor* evaluation framework. Each agent receives the same system prompt, the same data layout (`/app/data/`), the same EF5 binary path (`/EF5/bin/ef5`), the same parameter table with bounds, the same 20-round / 10-sweep budget, and the same NSE target (0.8075 on the held-out gauge). Frontier agents are accessed through the harness's default API integration with default sampling temperature; HYDROAGENT-4B is served by SGLang behind the harness's local-model adapter, with greedy decoding for reproducibility. The training-time rollout stack (verl + SGLang multi-turn) is decoupled from the evaluation harness so that no information leaks across the two; checkpoints are scored only through the harbor harness.

B CREST parameter reference

Table 4 gives the physical interpretation of every tunable CREST parameter that the agent sets via `set_parameters` (see Section 2.2). Parameters are scalar multipliers applied to the spatially distributed parameter rasters; the bounds are the physically reasonable ranges enforced by the calibration environment. Process roles follow Li et al. [2023], Wang et al. [2011] and the AQUAH parameter description in Yan et al. [2025]. The eleven calibrated parameters are not independent in their effect on the simulated hydrograph: `wm`, `b`, `im` jointly determine the rainfall-to-runoff partition

Table 4: CREST scalar-multiplier parameters exposed to the agent, their valid ranges, and their physical roles. Process attribution follows Li et al. [2023], Wang et al. [2011], Yan et al. [2025]. The first eleven parameters are calibrated; the last two (`th`, `isu`) are state parameters held at fixed values during our experiments but exposed through the same interface for completeness.

Parameter	Range	Process	Physical role
`wm`	$[0.1,\ 10.0]$	Soil moisture	Mean soil-water storage capacity (mm). Controls how much rainfall a soil column can absorb before saturation; larger `wm` delays runoff onset and sustains baseflow.
`b`	$[10^{-6},\ 3.0]$	Infiltration	Shape exponent of the variable infiltration curve. Larger `b` concentrates saturation in a smaller fraction of the basin, producing flashier runoff response.
`im`	$[0.0,\ 1.0]$	Surface partitioning	Impervious-area fraction. Sets the share of rainfall that bypasses infiltration entirely and routes directly as overland flow.
`ke`	$[0.8,\ 1.2]$	Evapotranspiration	Multiplier on the PET forcing. Tunes the bias of the input PET grid against basin water balance; affects long-term volume but not event peaks.
`fc`	$[0.1,\ 2.0]$	Subsurface	Saturated hydraulic conductivity multiplier. Governs the rate of vertical drainage from the soil store to the interflow store.
`under`	$[0.1,\ 10.0]$	Interflow	Interflow (subsurface) velocity. Faster `under` sharpens the recession; slower `under` extends the tail of the hydrograph.
`leaki`	$[0.1,\ 10.0]$	Interflow	Leakage rate from the interflow store to deeper groundwater. Higher `leaki` steepens recession and reduces total simulated volume.
`alpha`	$[0.1,\ 3.0]$	Channel routing	Coefficient in the kinematic-wave channel relation $Q = \alpha\, A^{\beta}$. Increases peak conveyance for a given cross-sectional flow area.
`beta`	$[0.1,\ 3.0]$	Channel routing	Exponent in the kinematic-wave channel relation $Q = \alpha\, A^{\beta}$. Controls non-linearity of channel response with flow magnitude.
`alpha0`	$[0.0,\ 3.0]$	Overland routing	Overland (non-channel) routing coefficient. Governs hillslope-flow celerity before water enters the channel network; affects time-to-peak.
`iwu`	$[0.1,\ 100.0]$	Initial state	Initial soil-water content as percentage of `wm`. Sets antecedent moisture and therefore the basin's runoff-coefficient memory at simulation start.
`th`	fixed at 10	Network	Channel-initiation threshold (cells of accumulated drainage area required to declare a channel). Defines the channel network rather than its dynamics.
`isu`	fixed at 0	Initial state	Initial interflow-storage content. Held at zero in our experiments.

and dominate event-peak magnitude; `fc`, `under`, `leaki` jointly shape the recession limb and the long tail; `alpha`, `beta`, `alpha0` control routing and therefore time-to-peak and peak attenuation; `ke` and `iwu` primarily shift the long-term water balance and antecedent state. A competent calibration agent must learn both the marginal and the interaction effects, which is the main reason the multi-criteria diagnostic exposed in Section 2.2 (volume ratio, peak error, time-to-peak, recession slope) is necessary: a single NSE value compresses these distinct physical signals into one number and gives the agent no leverage to disambiguate which group of parameters to move next.

C Ablation: contribution of SFT and RLSF

To isolate what each post-training stage contributes to the headline result of Section 4, we evaluate three policies under the identical *terminal-2*/*harbor* harness (Appendix A) used elsewhere in the paper: (i) the untuned base `Qwen3-4B-Instruct-2507`; (ii) `Qwen3-4B-hydro-sft`, the same base after only Stage 1 (supervised fine-tuning on 2,576 expert calibration trajectories, Section 2.3); and (iii) HYDROAGENT, the SFT-initialized policy after Stage 2 (GRPO under simulator-grounded NSE

rewards, Section 2.4; checkpoint `global_step_90`). All three policies share identical decoding settings (greedy), the same 50-turn / 20-round budget, the same parameter table with bounds, and the same NSE target (0.8075); only the weights differ. The four held-out gauges are those of Table 2.

Table 1 reports per-gauge best NSE alongside three diagnostic signals that turn out to be central to the interpretation: the number of distinct EF5 simulator invocations the agent successfully launches per episode (*sims*), the number of dialog turns actually consumed out of the 50-turn cap (*turns*), and the count of malformed tool-call attempts the harness had to reject (*parse fail*).

Stage 1 (SFT) alone is not sufficient—and is sometimes harmful. Compared to the untuned base, SFT-only *regresses* on three of the four held-out gauges, including a catastrophic collapse on the canonical evaluation gauge `02338660` ($+0.65 \rightarrow -17.53$, an ≈ 18-point NSE drop). The diagnostic columns of Table 1 explain why. SFT-only launches only 1–4 EF5 simulations per episode (vs. 13–16 for the baseline), exits the harness after 11–16 turns of the 50-turn budget, and produces 4 malformed tool-call attempts in every episode. Distillation has reproduced the *surface* of the teacher trajectories—tool-call grammar, parameter naming, and a hydrologic-reasoning style—but has *collapsed* the iterative behavior that produced the teacher's good final NSE. The policy commits a single parameter guess and stops; whether that guess turns out to be reasonable (`01403060`: 0.58, the only gauge where SFT-only beats both other variants) or grossly miscalibrated (`02338660`: -17.53) is essentially luck-of-the-draw, because the SFT-only policy is no longer willing to refine in response to feedback. This is the same *premature-termination* pathology Section 3 identifies in pro-tier frontier reasoning models, induced here by the maximum-likelihood objective itself: when teacher trajectories converge in 4–5 tool calls, the SFT-only policy learns to emit the converged distribution directly instead of the deliberation that produced it. As an isolated post-training stage, SFT distills format and style at the cost of long-horizon engagement, and on a panel-mean basis is worse than the untuned base (-4.79 vs. -0.14).

Stage 2 (RLSF) restores iteration and grounds it in simulator feedback. Adding GRPO with the simulator-grounded NSE reward recovers iterative behavior and improves best NSE on three of four held-out gauges relative to the SFT-only initialization: `02338660` from -17.53 to $+0.61$ ($\Delta\text{NSE} = +18.14$), `06279500` from -2.27 to -0.84 ($+1.43$), and `07144100` from $+0.07$ to $+0.65$ ($+0.58$). The diagnostic columns confirm that the gains travel with restored interaction: simulation counts return to 12–17 per episode, the full 50-turn budget is consumed on every gauge, and parse failures fall to zero across the panel. The lone regression—`01403060`, $+0.58 \rightarrow +0.40$—is the gauge where SFT-only's single-shot guess happened to land in a high-NSE region; RLSF's greedy decoding settles on a different local optimum but does not collapse. Relative to the untuned base, the full SFT + RLSF stack lifts panel-mean NSE from -0.14 to $+0.20$ and panel-median NSE from $+0.09$ to $+0.50$, and converts a third basin from negative to positive NSE.

The two stages are complementary, not substitutable. The marginal contribution of *adding* RLSF on top of SFT is $\Delta\text{NSE} = +4.99$ in panel mean ($-4.79 \rightarrow +0.20$); the marginal contribution of *using* SFT alone over the untuned base is $\Delta\text{NSE} = -4.65$ in panel mean ($-0.14 \rightarrow -4.79$). The asymmetry is the central finding of the ablation: SFT distills the tool-call vocabulary that prevents a 4-per-episode parse-failure tax once RLSF is layered on top, but in isolation it strips the model of the iterative engagement that the calibration task demands; only when paired with simulator-in-the-loop RL does SFT's grammar transfer pay off. Equivalently, RLSF is what supplies long-horizon credit assignment grounded in physical-error feedback, but starting GRPO from the untuned base instead of the SFT initialization is precluded by the high parse-failure rate of the untuned base on the tool-call schema—empirically observed in early pilots, and the reason our pipeline anchors GRPO to the SFT initialization with KL coefficient 0.2 (Section 2.4, Table 3). The headline result of Section 4 is therefore not attributable to either stage alone: it is the product of the SFT $\rightarrow$ RLSF sequence, in which SFT provides the format and RLSF supplies the iteration.

D Hydrograph comparison on the held-out test panel

Figures 5–8 show, for each of the four held-out test gauges, the gauge observation alongside the simulated discharge produced by the base `Qwen3-4B-Instruct-2507` agent (slate gray) and by HYDROAGENT after SFT + RLSF post-training (amber). Each panel reports the EF5/CREST per-run statistics in the corresponding line color: NSE, percent bias, Pearson correlation, modified correlation (modCC), MAE, RMSE, peak-magnitude error (cms), and peak-timing error (hours). The basin-

average rainfall is rendered on an inverted secondary axis (green). The headline NSE comparison rebuilds, gauge by gauge, the panel-mean improvement summarized in Figure 3: HydroAgent matches or exceeds the base model on NSE for every gauge, with the largest gains on 01403060 (negative $\rightarrow$ positive) and 07144100 (low- to high-band positive). Visually, HydroAgent's recession limbs are noticeably better-aligned with the observed hydrograph than the base model's, while the timing of simulated peaks remains the weakest residual error—consistent with the volume-/peak-weighted shaping of the RLSF reward (Section 2.4) and the observation discussed in Section 4.1.

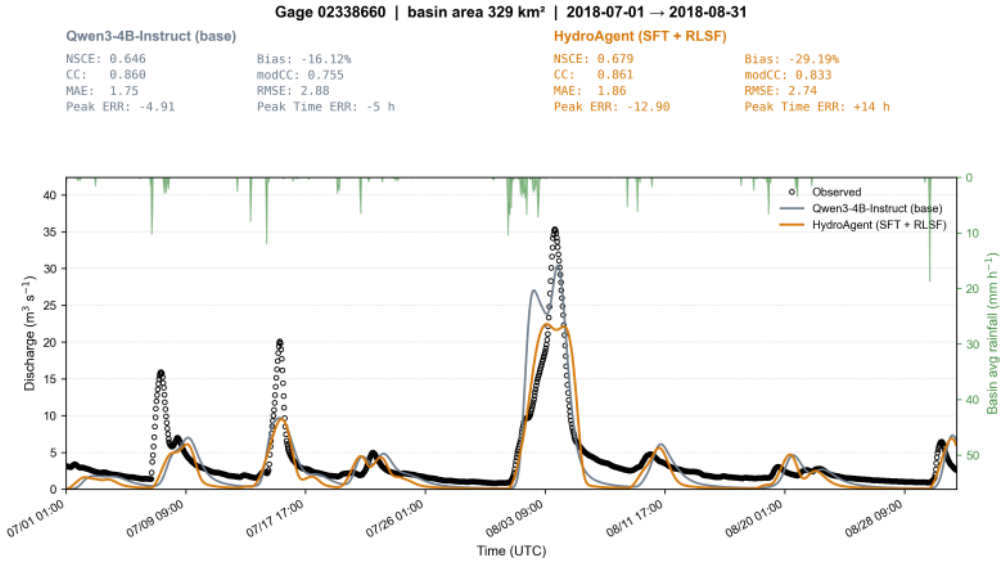


Figure 5: gauge 02338660 (basin area 329 km^2; July–August 2018). HydroAgent improves NSE from 0.65 to 0.68 and reduces RMSE; both runs share a similar peak-timing offset.

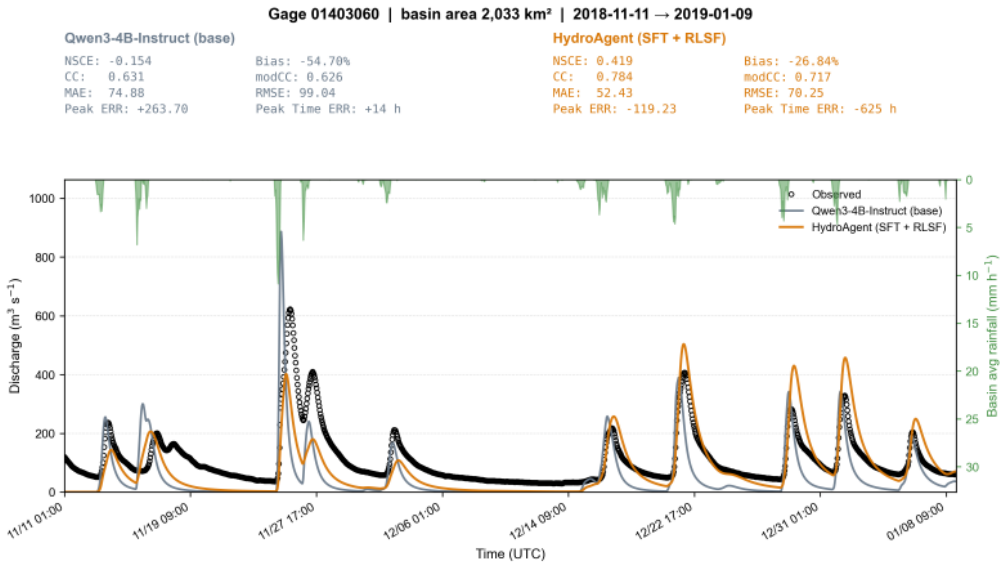


Figure 6: gauge 01403060 (basin area 2,033 km^2; November 2018–January 2019). The base model produces a runaway over-prediction on the late-November event ($>$1,000 m^3 s^{-1} against an observed peak near 600 m^3 s^{-1}); HydroAgent collapses that bias and lifts NSE from -0.15 to $+0.42$.

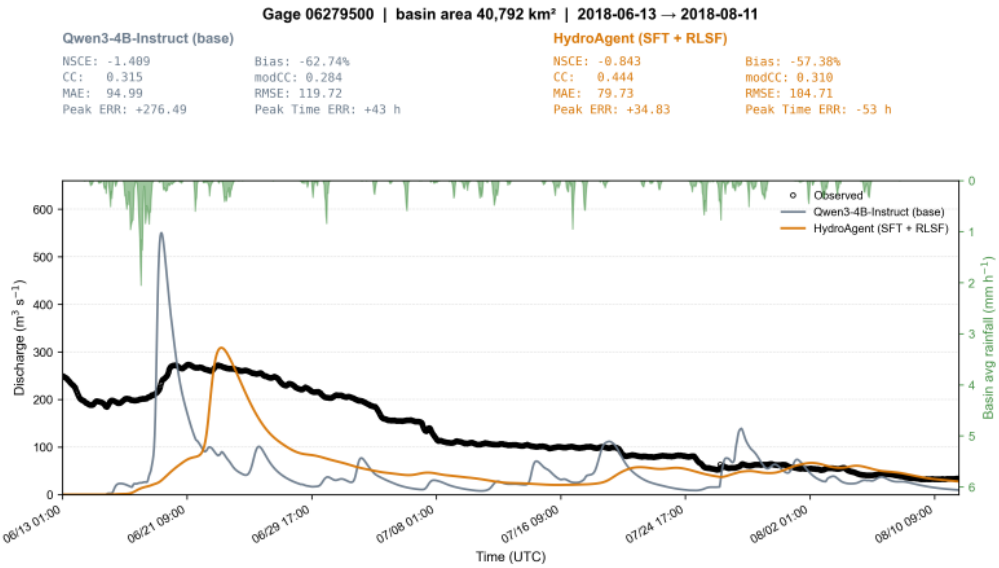


Figure 7: gauge 06279500 (basin area $40{,}792\,\mathrm{km}^2$; June–August 2018). The largest basin in the test panel and a documented difficult case (Appendix A). HydroAgent reduces the magnitude of the negative NSE substantially ($-1.41 \rightarrow -0.84$) but does not reach a positive value within budget—an artifact of the basin's scale relative to our training pool ($\leq 2{,}401\,\mathrm{km}^2$). This likely reflects a limitation of the underlying physical model in heavily human-managed basins.

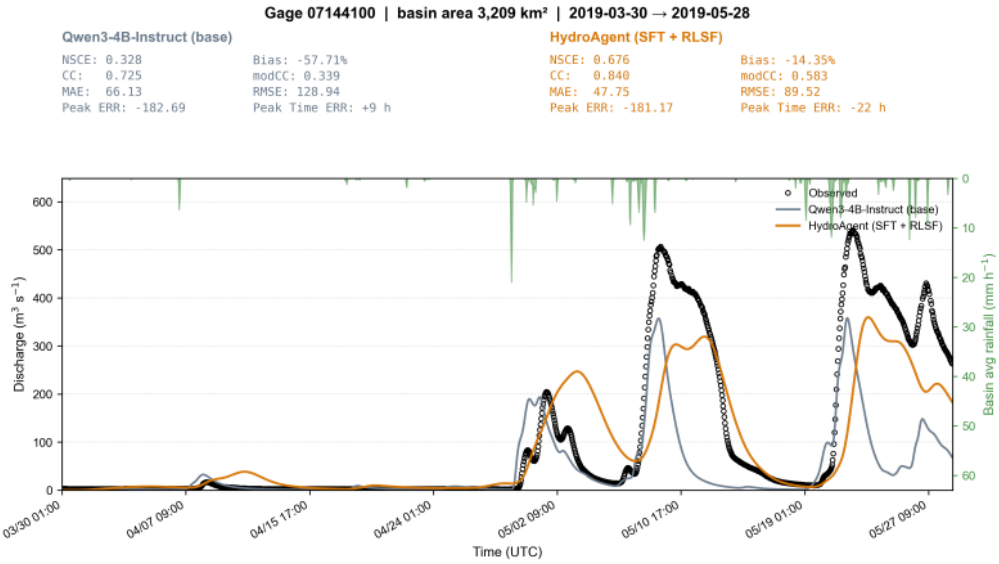


Figure 8: gauge 07144100 (basin area $3{,}209\,\mathrm{km}^2$; March–May 2019). HydroAgent more than doubles NSE from 0.33 to 0.68, with a visibly tighter recession limb and a peak-magnitude reduction matching the observed series.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We stated in the abstract and introduction that HydroAgent performs comparably to the state-of-the-art LLMs and demonstrates strong potential to close the gap between them and human experts in hydrologic model calibration. And we demonstrated that in our experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We mentioned in the last section that we only evaluated the performance of HydroAgent in selected basins in the US but the approach can be extended to other basins.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: There are no theoretical results in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provided detailed information about the experimental setup, including the dataset, model architecture, training procedure, and evaluation metrics.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We submitted the code in the zip file and the trained LLM has been uploaded to Hugging Face with an anonymous account.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We described the training and test details in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We reported the mean and standard deviation of the performance of HydroAgent in figure 2 and other figures for significance test.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We mentioned the training compute in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed the potential positive and negative societal impacts of our work in Section 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: There is no foreseeable risk for misuse of the data and models released in this paper.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cited the original papers that produced the open base model we used in this work.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We documented the code and dataset.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: No crowdsourcing experiments or research with human subjects were conducted in this paper.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing nor research with human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We only use the LLMs for writing, editing, and formatting purposes, and they do not impact the core methodology, scientific rigor, or originality of our research. The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.