AgriWorld: A World–Tools–Protocol Framework for Verifiable Agricultural Reasoning

Anonymous ACL submission

Abstract

Agricultural foundation models trained on remote sensing, soil, weather, and management data have improved forecasting and monitoring, but remain hard to use for language-driven agronomic analysis: they neither expose interactive reasoning nor support verifiable multi-step workflows over heterogeneous data. We present AGRIWORLD, a Python-based environment that unifies tools for parcel-level geospatial queries, remote-sensing time-series analysis, crop-growth simulation, and task-specific predictors for yield, stress, and disease risk. Building on AGRIWORLD, we introduce AGROREFLECTIVE, a multi-turn LLM agent that plans analyses, writes executable code, observes intermediate results, and refines its reasoning through an execute–observe–refine loop. We further propose AGROBENCH, a scalable benchmark for agricultural question answering across lookup, forecasting, anomaly detection, and counterfactual analysis. Experiments show that execution-driven reflection consistently improves over text-only and direct tool-use baselines, demonstrating a practical route toward verifiable agricultural reasoning.

1 Introduction

Large language models (LLMs) are increasingly used as general-purpose assistants for scientific work: they can follow complex instructions, explain concepts, summarize documents, and generate code that automates parts of an analysis (Luo et al., 2025; Prabhakar et al., 2025; Baier et al., 2002). Yet, in many scientific domains, answering even a seemingly simple question requires operating over *high-dimensional, heterogeneous, and spatiotemporal numerical data*—not purely textual context. This mismatch has motivated tool-augmented and code-executing agents that connect LLMs to external programs and datasets, enabling conclusions to be grounded in computation and verified through intermediate results rather than surface-form text (M. Bran et al., 2024; Ma et al., 2024; Wei et al., 2022; Zhou et al., 2022; Gao et al., 2023; Chen et al., 2023; Madaan et al., 2023; Paranjape et al., 2023). Recent agent research further shows that reliable problem solving benefits from explicit tool use, reflective refinement, and benchmarked interaction with external environments rather than one-shot text generation alone (Qin et al., 2024; Shen et al., 2023; Patil et al., 2023; Wang et al., 2024a; Liu et al., 2024; Wang et al., 2022; Xie et al., 2024). Our work builds on this trend, but focuses on a domain where correctness depends on numerically grounded execution over heterogeneous earth-observation and agronomic data rather than API orchestration alone.

Agriculture is a particularly compelling and challenging setting for executable scientific assistants. Modern agronomic decision making relies on diverse data products, including multi-spectral remote sensing time series, soil and terrain grids, field-parcel boundaries, field-level management logs, and local weather streams. Practical questions are inherently spatiotemporal and unit-sensitive: they require parcel–pixel–grid alignment, stage-specific temporal windows, and careful handling of measurement scales. Decades of work in vegetation monitoring and agricultural simulation have established the importance of physically meaningful indices and executable models for farm decision support (Tucker, 1979; Huete et al., 2002; Zhang et al., 2003; JONES et al., 2003; Holzworth et al., 2014). Meanwhile, modern digital agriculture increasingly combines remote sensing, smart-farming data platforms, and machine learning pipelines (Mulla, 2013; Wolfert et al., 2017; Kamilaris and Prenafeta-Boldú, 2018; Liakos et al., 2018; Kussul et al., 2017; Russwurm and Korner, 2018). However, foundation models trained on large agronomic corpora still often lack language-facing interfaces for interactive analysis (Yang et al., 2025; Chen et al., 2025), "what-if" exploration, and transparent, in-

spectable reasoning steps. Conversely, LLMs excel at natural-language interaction but cannot natively manipulate structured agronomic datasets, and thus may produce answers that are plausible yet difficult to verify (Yan et al., 2026). The gap becomes evident in everyday workflows. Consider a query such as: *"Which parcels are likely to experience water stress next week, and what irrigation adjustment would mitigate the risk?"* A reliable solution requires more than recalling facts. It must (i) perform geospatial filtering and joins across entities (fields, administrative regions, sensor footprints), (ii) compute and visualize remote-sensing indicators over time with correct aggregation windows, (iii) incorporate soil properties and exogenous drivers such as weather, and (iv) run a simulator or predictor to evaluate counterfactual interventions. In practice, agronomists orchestrate an ecosystem of tools—GIS operations, time-series analytics, crop growth simulators, and task-specific models—and translate computational artifacts into actionable recommendations. Without a programmatic interface and an execution loop, an LLM has no principled way to check coordinate systems, validate intermediate computations, or correct subtle errors in spatial joins, temporal windows, and unit conversions, making purely text-based responses brittle and hard to trust for scientific use (Zhang et al., 2024). We propose an agentic framework for agricultural science that makes LLM-based analysis *executable, auditable, and reproducible* (Jin et al., 2025; Wang et al., 2024b). Our design follows a **World–Tools–Protocol** abstraction. **World** provides a Python execution environment, AGRIWORLD, which exposes unified APIs for core agricultural operations: (1) geospatial querying over field parcels and administrative regions, (2) remote-sensing time-series analytics and anomaly statistics, (3) crop growth simulation with support for counterfactual interventions, and (4) task-specific predictors such as yield, stress, and disease risk. **Tools** in AGRIWORLD return inspectable intermediate artifacts (e.g., tables, plots, and derived masks), allowing analyses to be grounded in concrete computations and audited step-by-step. **Protocol** standardizes task specification and evaluation through *executable references and checkers* whenever the underlying question admits programmatic validation, enabling scalable and reproducible comparisons. On top of AGRIWORLD, we introduce AGRO-REFLECTIVE, a multi-turn LLM agent that alternates between writing code, executing it, and refining its analysis based on execution feedback. This execution-driven reflection is crucial in agriculture, where minor mistakes in coordinate systems, aggregation windows, or units can invalidate the final conclusion. Rather than treating tool calls as one-shot add-ons to generation, AGRO-REFLECTIVE uses intermediate artifacts as first-class signals to diagnose errors, revise hypotheses, and converge to a checkable solution. To evaluate such agents systematically, we build a verifiable evaluation suite *on top of an existing agricultural benchmark* by converting questions into tool-grounded instances with deterministic reference code and executable checker functions. This suite covers a broad spectrum of capabilities: basic lookups (indices, phenology, and resource use), forecasting and monitoring, anomaly detection, and counterfactual intervention analysis. Importantly, the executable design enables fine-grained diagnosis of failure modes (e.g., spatial misalignment, incorrect temporal windows, unit errors, and ungrounded claims), beyond coarse end-answer accuracy (Li et al., 2026). The main **contributions** are three-fold:

- **AGRIWORLD: an executable agricultural environment.** We introduce a Python environment that unifies geospatial querying, remote-sensing time-series analytics, crop simulation, and task-specific predictors through consistent APIs, enabling grounded and auditable agronomic analyses.
- **AGRO-REFLECTIVE: an execution-driven reflective agent.** We propose a multi-turn agent that iterates *execute–observe–refine*, using intermediate artifacts to self-correct and produce tool-grounded analysis traces rather than one-shot text.
- **A verifiable evaluation suite with executable checkers.** Built on top of an existing benchmark, we provide a scalable task generation and scoring pipeline with deterministic reference programs and executable checkers spanning lookups, forecasting, anomaly detection, and counterfactual interventions, enabling reproducible comparisons against text-only and direct tool-use baselines.

2 Related Work

Tool-augmented LLM agents and executable reasoning. A growing line of work augments LLMs with external tools (APIs, code interpreters, and symbolic modules) to ground answers in com-

putation rather than purely textual generation. Such systems typically couple a planner with tool invocation, enabling multi-step problem solving over structured resources and producing intermediate results that can be inspected (Schick et al., 2023; Yao et al., 2022). Program-centric reasoning methods such as chain-of-thought prompting, least-to-most decomposition, PAL, and Program-of-Thoughts further highlight the benefit of separating symbolic computation from surface-form generation (Wei et al., 2022; Zhou et al., 2022; Gao et al., 2023; Chen et al., 2023). More recent systems extend this perspective with self-refinement, automated tool selection, and large tool libraries (Madaan et al., 2023; Paranjape et al., 2023; Qin et al., 2024; Shen et al., 2023; Patil et al., 2023). Our work aligns with this direction but targets a domain where correctness hinges on spatiotemporal alignment, unit consistency, and reproducible execution traces. We thus emphasize (i) a domain-grounded environment that exposes typed operations and auditable artifacts, and (ii) a protocol that makes evaluation executable whenever possible, beyond text-only answer matching (Gupta and Kembhavi, 2023).

Scientific and data-centric agents for domain workflows. Recent scientific assistants and data-analysis agents aim to automate parts of research workflows by writing code, operating over datasets, and iteratively refining hypotheses using execution feedback (Jin et al., 2025; Wang et al., 2024b,a). Parallel efforts in geospatial analytics, remote sensing pipelines, and decision-support systems have developed mature toolchains for spatial joins, raster processing, and time-series modeling, but these pipelines are typically engineered for experts and are not language-interactive (Xu et al., 2024).

Benchmarks and verifiable evaluation for agentic systems. Evaluating agents remains challenging because free-form responses can be plausible yet incorrect, and failures often stem from latent mistakes in intermediate steps (Li et al., 2026). Existing benchmarks commonly score final textual answers, sometimes with reference solutions, but often without executable checkers that can validate constraints, tolerances, and counterfactual claims. Complementary work has proposed environment-based benchmarks such as ScienceWorld, AgentBench, GAIA, and OSWorld to enable reproducible assessment and failure diagnosis (Wang et al., 2022; Liu et al., 2024; Xie et al., 2024).

Agricultural remote sensing and decision support. Our problem setting is grounded in a long tradition of agricultural sensing and simulation. Satellite vegetation indices such as NDVI and EVI provide compact summaries of canopy status across space and time (Tucker, 1979; Huete et al., 2002; Zhang et al., 2003), while agronomic simulators such as DSSAT and APSIM support executable scenario analysis over weather, soil, and management interventions (JONES et al., 2003; Holzworth et al., 2014). Recent digital-agriculture research further integrates remote sensing, smart-farming infrastructure, and machine learning to support crop monitoring and yield prediction at scale (Mulla, 2013; Wolfert et al., 2017; Kamilaris and Prenafeta-Boldú, 2018; Liakos et al., 2018; Kussul et al., 2017; Russwurm and Korner, 2018).

3 Methodology

3.1 World–Tools–Protocol

Current LLMs lack the intrinsic capacity to reason over high-dimensional spatiotemporal data. To bridge this gap, we cast an agricultural assistant not as a chatbot but as an agent $\mathcal{A}$ operating within a *World–Tools–Protocol* framework (Figure 1). The framework comprises three components: an executable environment $\mathcal{E}$ (AGRIWORLD), a set of grounded functional tools $\mathcal{F}$, and a verifiable specification protocol $\mathcal{V}$.

As in Figure 1, the workflow originates from the state space $\mathcal{S}$, which encapsulates diverse agricultural inputs such as parcel geometries and wind fields. These inputs are routed through specialized modules for spatial querying, spatio-temporal analysis, crop simulation, and task-level prediction. To reconcile heterogeneous sources, a *coordinate alignment* step converts the raw pre-training sequence into a standardized aligned sequence (e.g., harmonizing units g $\rightarrow$ kg). The reasoning trace is then audited by the agent itself: temporal anomalies are detected and patched, and the final output is validated against pattern, conservation, and semantic constraints before being released.

Task instance definition. A task instance is a tuple $\mathcal{I} = \langle q,\ \mathcal{B},\ \mathcal{S}_{\text{out}},\ \mathcal{V},\ B \rangle$, where q is the natural-language query; $\mathcal{B}$ denotes the *bindings* to concrete data handles (parcel identifiers $\mathcal{P}_{id}$, time intervals $[t_{\text{start}}, t_{\text{end}}]$, sensor products, etc.); $\mathcal{S}_{\text{out}}$ specifies the output schema required by downstream applications; $\mathcal{V}$ is an executable checker; and B is the computational budget (maximum number of steps).

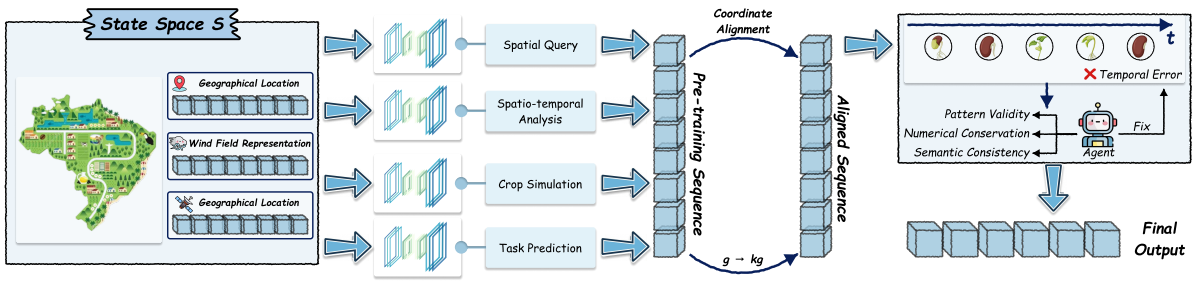

Figure 1: Overview of our proposed framework. The pipeline begins with a heterogeneous state space $\mathcal{S}$ (e.g., geographical locations, wind fields). Data flows through functional modules, including spatial query, spatio-temporal analysis, and crop simulation. A coordinate alignment step then maps the raw pre-training sequence onto a unified aligned sequence (e.g., g→kg). Finally, the agent audits the execution timeline for temporal errors and enforces pattern validity, numerical conservation, and semantic consistency before emitting the final output.

Tool Family	Illustrative Signatures
Geospatial Querying	`geo.filter_parcels(region, crop, area_range)` `geo.sjoin(parcels, sensor_footprints)`

Remote-Sensing Analytics	`rs.load(index, time_range, region)`
	`rs.zonal_stats(raster_ts, parcels, stat={mean,. . . })`
	`rs.anomaly(ts, baseline_window, method)`

Soil/Terrain Access	`grid.sample(gridfield, parcels)` `grid.aggregate(gridfield, region)`
Weather Variables	`wx.get(station/region, time_range)` `wx.rolling_sum/degree_days(. . . )`
Simulation & Predictors	`sim.run(parcels, wx, soil, mgmt, intervention)` `pred.stress/yield/disease(. . . )`

Table 1: **The AGRIWORLD toolset.** Core APIs grouped by functional domain. The environment exposes unified interfaces over heterogeneous spatiotemporal data.

Validity maximization. The agent interacts with $\mathcal{E}$ to produce a final answer $\hat{y}$ and a set of auditable artifacts $\mathcal{O}$. The protocol evaluates the solution as

$$(z, d) = \mathcal{V}(\hat{y}, \mathcal{O} \mid \mathcal{I}), \quad z \in \{0, 1\}, \tag{1}$$

where z is a binary correctness indicator and d is a structured diagnostic report enumerating violated constraints. The agent seeks a policy π that maximizes the expected validity $\mathbb{E}[z]$ subject to the budget B.

3.2 AGRIWORLD: Executable Environment

Data Model and State Space. Agricultural data are inherently heterogeneous. AGRIWORLD abstracts this complexity into a unified state space $\mathcal{S}$ composed of five typed entities:

- **Parcels ($\mathcal{P}$):** vector entities, where each $p \in \mathcal{P}$ carries a polygon geometry g_p, a coordinate reference system (CRS), and attribute metadata (e.g., crop type).
- **Raster time series ($\mathcal{R}$):** 4D tensors representing remote sensing observations. A raster $r \in \mathcal{R}$ is defined by a grid mapping $M : \mathbb{R}^2 \times T \to \mathbb{R}^k$ together with spectral bands and a validity mask (e.g., cloud cover).
- **Grid fields ($\mathcal{G}$):** static spatial layers (e.g., soil texture, DEM) aligned to a canonical grid.
- **Weather streams ($\mathcal{W}$):** time series of exogenous drivers $w = \{(t, \mathbf{v}_t)\}_{t=1}^{T}$, where $\mathbf{v}_t \in \mathbb{R}^d$ contains variables such as precipitation and radiation.
- **Management logs ($\mathcal{M}$):** discrete event sequences $m = \{(t_i, a_i, q_i)\}$ recording actions a_i (e.g., irrigation) and quantities q_i.

Unified Tool Interface and Artifacts. To ground reasoning in computation, AGRIWORLD exposes a set of functional tools $\mathcal{F}$. We treat each tool execution not as an opaque query but as a state transition that emits an *artifact*:

$$(\hat{v}, \alpha) = f(s_{\text{in}}; \theta), \quad f \in \mathcal{F}, \tag{2}$$

where $s_{\text{in}} \subseteq \mathcal{S}$ is the input data subset, $\hat{v}$ is the returned value (e.g., a time-series DataFrame), and α is the *inspectable artifact*.

We define an artifact as $\alpha = \langle \text{type}, \text{payload}, \text{meta}, \text{prov} \rangle$. Here meta

records physical constraints (units, CRS, resolution), and $\mathsf{prov} = H(f, s_{\text{in}}, \theta)$ is a cryptographic hash of the inputs and tool version that guarantees reproducibility of every claim. In practice, this environment view is implemented on top of mature scientific-computing primitives for tabular, array, optimization, plotting, and labelled spatiotemporal data (McKinney, 2010; Harris et al., 2020; Virtanen et al., 2020; Hoyer and Hamman, 2017; Hunter, 2007), allowing the agent to operate on auditable intermediate states rather than black-box tool outputs.

Spatiotemporal Alignment and Unit Safety. A core difficulty in agricultural AI is the misalignment of heterogeneous data sources. AGRIWORLD addresses this with a canonical alignment operator.

Canonical alignment operator. We define a composite operator Π that maps arbitrary inputs into a unified working space:

$$\tilde{x} = \Pi(x) = (\mathcal{T}_{\text{resample}} \circ \mathcal{S}_{\text{reproject}})(x). \quad (3)$$

Here $\mathcal{S}_{\text{reproject}}$ transforms vector and raster geometries into a task-specific CRS (e.g., EPSG:3857 for metric calculations), while $\mathcal{T}_{\text{resample}}$ aligns temporal data to a canonical frequency (e.g., daily) using explicit interpolation policies (nearest or linear).

Robust zonal statistics. To handle noise in satellite observations (e.g., clouds), we formalize the zonal-statistics operation. For a parcel p and a raster r with pixel values r_i and validity mask $m_i \in \{0, 1\}$, the aggregated signal μ and the *validity ratio* ν are

$$\mu(p, r) = \frac{\sum_{i \in Z(p)} m_i \cdot r_i}{\sum_{i \in Z(p)} m_i}, \quad \nu(p, r) = \frac{\sum_{i \in Z(p)} m_i}{|Z(p)|}, \quad (4)$$

where $Z(p)$ is the set of pixels spatially intersecting p. The system raises a `LowCoverageError` whenever $\nu(p, r) < \tau_{\min}$, preventing the agent from hallucinating on corrupted inputs.

3.3 AGRO-REFLECTIVE: Execution-Driven Agent

Generic agents tend to fail in scientific domains because of *silent* errors (e.g., wrong units that still produce plausible numbers). We introduce AGRO-REFLECTIVE, which couples an *Execute–Observe–Refine* loop with structured diagnostic feedback (Shinn et al., 2023).

Algorithm 1 AGRO-REFLECTIVE Execution Loop

Require: Task $\mathcal{I}$, environment $\mathcal{E}$, budget $T_{\max}$

$m_0 \leftarrow \text{ENCODE}(\mathcal{I})$
for $t = 1$ **to** $T_{\max}$ **do**
  $c_t \sim \pi_\theta(\cdot \mid m_{t-1})$ {propose code}
  $(o_t, \mathcal{O}_t) \leftarrow \mathcal{E}.\text{exec}(c_t)$ {execute}
  $m_t \leftarrow \text{UPDATE}(m_{t-1}, c_t, o_t)$
  $(z, d) \leftarrow \mathcal{V}(\hat{y}_t, \mathcal{O}_{1:t} \mid \mathcal{I})$ {protocol check}
  if $z = 1$ **then**
    return $\text{FINALIZE}(m_t)$
  end if
  $c_{\text{patch}} \leftarrow \rho(m_t, d)$ {self-correction}
  $m_t \leftarrow \text{APPEND}(m_t, c_{\text{patch}})$
end for
return $\text{BESTEFFORT}(m_{T_{\max}})$

Policy and trace. At step t, the agent acts under a policy $\pi_\theta(c_t \mid m_{t-1})$ conditioned on its memory m_{t-1}. Execution yields an observation tuple $(o_t, \mathcal{O}_t) \leftarrow \mathcal{E}(c_t)$, and the interaction trace is recorded as $\tau_t = \langle (c_1, o_1, \mathcal{O}_1), \ldots, (c_t, o_t, \mathcal{O}_t) \rangle$.

Reflection mechanism. Reflection is modelled as a diagnostic function ρ. When a runtime error or protocol violation occurs, the agent inspects the artifact metadata $\mathcal{M}(\alpha)$ and synthesizes a corrective patch:

$$c_{t+1} = \rho(c_t,\ d_t,\ \mathcal{M}(\alpha_t)). \quad (5)$$

ρ classifies errors into typed categories (e.g., *SpatialMisalignment*, *UnitError*) and proposes targeted fixes. The value of this loop, alongside the alignment operator Π, is empirically validated by our design-space study (Table 3): removing either component significantly degrades performance.

3.4 Protocol: Verifiable Specification

The protocol $\mathcal{V}$ grounds the agent's reasoning through a hierarchy of executable checkers, each enforcing a distinct class of constraint.

Schema validity. The output $\hat{y}$ must conform to the rigid JSON schema $\mathcal{S}_{\text{out}}$, exposing canonical fields such as `parcel_id`, `value`, and `unit`.

Numeric tolerance. For quantitative queries, the result must lie within a dynamic tolerance δ of the reference simulator output y^*, i.e., $|\hat{y} - y^*| < \delta$.

Counterfactual consistency. For "what-if" analyses, e.g., *does irrigation I alleviate water stress?*, the checker verifies the causal direction. Let $M(s, a)$ denote the simulator function evaluated

on state s under action a. The proposed intervention must satisfy

$$M(s, a_{\text{intervene}}) - M(s, a_{\text{baseline}}) \leq -\Delta, \quad (6)$$

ensuring that the agent identifies a strategy with non-trivial effect of magnitude at least Δ.

Physical sanity. All intermediate artifacts are inspected for unit consistency (dimensional homogeneity), CRS coherence, and sufficient valid-pixel coverage ($\nu > \tau_{\min}$), which prevents silent corruption from propagating downstream. Together, these checks transform vague agricultural recommendations into verifiable, reproducible computational claims.

4 Experiments

To verify whether executable, reflective agents can fundamentally improve agricultural reasoning over heterogeneous spatiotemporal data, we aim to answer the following questions:

- **RQ1 (Efficacy):** Does tool-grounding and reflective execution outperform state-of-the-art text-only LLMs in domain-specific tasks?
- **RQ2 (Ablation):** What is the individual contribution of each component (e.g., remote sensing tools, alignment protocol, reflection loop) to the overall system performance?
- **RQ3 (Robustness):** Can the proposed agent generalize to unseen geographic regions and temporal windows with extreme weather conditions?

4.1 Experimental Setup

Benchmark and Metrics. We utilize AGROBENCH, a verifiable evaluation suite constructed from real-world agronomic workflows. Unlike standard QA datasets, AGROBENCH requires agents to interact with the AGRIWORLD environment to derive answers. We employ a rigorous set of metrics for different task types: **Quantitative Accuracy:** For lookup tasks, we measure exact matches on retrieved values. **Forecasting Error:** For time-series prediction (e.g., yield, moisture), we use the Normalized Root Mean Square Error (NRMSE), defined as $NRMSE = \frac{\sqrt{\frac{1}{N}\sum(\hat{y}_i - y_i)^2}}{\bar{y}}$. **Spatial IoU:** For anomaly detection (e.g., identifying water stress zones), we calculate the Intersection over Union (IoU) between the predicted polygon mask and the ground truth derived from Sentinel-2 satellite imagery. **Causal Consistency:** For counterfactual analysis, we verify the causal direction using the simulator function $M(s, a)$. A success is recorded only if the agent's proposed intervention $a_{intervene}$ satisfies the physical constraint $M(s, a_{intervene}) - M(s, a_{baseline}) \leq -\Delta$, ensuring the decision is physically grounded.

Baselines and Implementation. To isolate the impact of our framework, we control for the underlying Large Language Model (LLM). We finetune the **Qwen3** series (8B and 32B parameters) using Low-Rank Adaptation (LoRA) on the training split. We compare three distinct interaction paradigms: **Text-only Baseline:** The model answers queries using internal parametric knowledge only. This serves as a baseline to measure the hallucination rate of pure LLMs. **AgriWorld-Direct (One-shot):** A standard "Code Interpreter" approach where the agent generates and executes Python code in a single pass without error handling. **Agro-Reflective (Ours):** The proposed agent equipped with the *Execute–Observe–Refine* loop. We set the maximum interaction budget $T = 20$ steps. The agent receives feedback from standard error streams and our custom validator $\mathcal{V}$. Additionally, we benchmark against strong open-source and proprietary baselines, including Qwen3 ,DeepSeek-V3 , Yi-1.5-34B , LLaMA 3 ,GPT-4o , and Gemini-2.0 to assess whether model scale can substitute for tool grounding.

4.2 Main Results and Analysis

Overall Performance Comparison. Table 2 presents the main evaluation results, including domain-specific QA performance on *Animal*, *Plant*, *Aquatic*, and *Herb*. The proposed **Qwen3-32B-LoRA-Reflective** achieves an aggregate QA score of **7.72/541** and an aggregate choice accuracy of **73.84%**, establishing a new state-of-the-art on this benchmark. **Beating the Giants with Grounding.** A critical finding is that our specialized, grounded agent significantly outperforms general-purpose giant models. For instance, **GPT-4o**, despite its massive parameter count and RLHF training, only achieves 36.75% accuracy. This is primarily because general-purpose models lack the specific API knowledge of AGRIWORLD and often hallucinate plausible but incorrect parameter names (e.g., inventing a nonexistent `get_soil_moisture()` function instead of using the correct `grid.sample()` API). In contrast, our agent, through tool grounding, strictly adheres to

Table 2: **Main results on agricultural reasoning benchmarks.** QA performance is reported as score/#tasks, and multiple-choice performance is reported as accuracy. Boldface marks the best full-benchmark result.

Model	Domain QA Score / #Tasks					MCQ Acc. (%)
	Animal	Plant	Aquatic	Herb	Total	
Open-source baselines						
Qwen3-32B	5.87/118	5.55/325	5.58/74	6.58/24	5.67/541	39.67
DeepSeek-V3	3.36/33	4.31/90	4.08/20	5.21/7	4.17/150	57.43
Yi-1.5-34B-Chat	5.79/33	5.32/90	6.60/20	6.86/7	5.67/150	35.38
Proprietary models						
GPT-4o	7.67/33	6.76/90	7.05/20	8.43/7	7.07/150	36.75
Gemini-2.0-Flash	2.88/33	4.02/90	4.30/20	6.07/7	3.90/150	55.33
LoRA-tuned baselines						
Qwen3-32B-LoRA	7.21/118	7.45/325	7.28/74	8.46/24	7.42/541	67.25
Qwen3-8B-LoRA	5.97/118	5.71/325	5.82/74	7.04/24	5.84/541	57.55
Agro-Reflective (ours)						
Qwen3-8B-LoRA-Reflective	6.08/118	6.41/325	5.92/74	5.98/24	6.25/541	41.34
Qwen3-32B-LoRA-Reflective	**7.45/118**	**7.76/325**	**7.68/74**	**8.66/24**	**7.72/541**	**73.84**

Table 3: **Ablation and design-space exploration of AGRO-REFLECTIVE.** We compare against text-only and direct-execution baselines. "Align." denotes spatiotemporal canonical alignment. **Bold** marks the best result.

Method	Inter.	Scope	Align.	Turns	Acc.
Baselines					
(1) Text-Only	Chat	None	–	1.0	21.9
(2) AgriWorld-Direct	One-shot	All	✓	1.0	48.2
Agro-Reflective Variants (Ours)					
(3) w/o Remote Sensing	Reflective	w/o RS	✓	2.4	41.5
(4) w/o Alignment	Reflective	All	×	3.1	51.3
(5) w/o Reflection	One-shot	All	✓	1.0	50.6
(6) **AGRO-REFLECTIVE (Full)**	**Reflective**	**All**	✓	3.5	**57.6**

Table 4: **Task-specific performance breakdown.** AGRO-REFLECTIVE improves tool-based lookup and complex multi-step reasoning.

Model	Lookup *Acc.* ↑	Forecast. *NRMSE* ↓	Anomaly *IoU* ↑	Counterf. *Succ.* ↑
Text-only	41.2%	0.89	0.12	14.5%
One-shot	82.5%	0.34	0.51	43.8%
OURS	**86.7%**	**0.18**	**0.68**	**71.4%**

the environment's schema.**The Failure of Text-Only Models.** Text-only baselines (Qwen3-32B, Llama3-8B) achieve only 30–40% accuracy, confirming that agricultural reasoning requires retrieving dynamic soil/weather data and aggregating time series rather than relying on parametric memory.

4.2.1 Fine-grained Task Analysis

To better understand where the improvements come from, we analyze the performance breakdown by task type (Table 4). **Lookups vs. Reasoning.** The gap between baselines and our method is smallest in the *Lookup* tasks (82.5% vs 86.7%). This is expected, as lookups (e.g., "What is the crop type of parcel X?") require only a single API call. Most "Code Interpreter" models can handle this without complex reasoning. **The Challenge of Forecasting.** In *Forecasting* tasks, however, the gap widens significantly (NRMSE 0.34 vs 0.18). Forecasting requires multi-step data processing: fetching weather data, resampling it to a daily frequency, aligning it with crop growth stages, and running a predictive model. The One-shot baseline often fails at the alignment step—for example, mixing up monthly aggregated precipitation with daily temperature—leading to poor predictions. AGRO-REFLECTIVE detects these dimensional mismatches via the execution feedback loop and corrects the resampling code, resulting in a 47% reduction in error. **Causal Reasoning.** The most dramatic improvement is seen in *Counterfactual Analysis* (Success Rate 43.8% vs 71.4%). These questions ask "What if we reduce irrigation by 10%?". To answer this, the agent must run a simulation, observe the outcome, and verify if the change is significant. The Text-only models fail completely here (<15%), as they cannot simulate physical processes. Our reflective agent excels because it treats the simulation as an experimental sandbox, iteratively adjusting inputs until a valid causal inference is drawn.

4.3 Ablation Study (Table 3)

Impact of Remote Sensing Tools. When we removed the remote sensing module (*w/o RS*), the agent lost the ability to "see" the fields. Performance on crop monitoring tasks dropped by over

40%. Text-only descriptions of fields (e.g., "a green field") are insufficient for calculating precise indices like NDVI, proving that pixel-level data access is non-negotiable for agricultural AI.

Impact of Canonical Alignment. Removing the canonical alignment operator (*w/o Alignment*) caused a surge in runtime errors. Without the Coordinate Reference System (CRS) transformation, the agent attempted to join GPS coordinates (WGS84) with projected satellite rasters (UTM), resulting in empty intersection sets. This confirms the spatiotemporal alignment protocol is the "glue" that makes heterogeneous data usable. **Impact of Reflection Loop.** Comparing *AgriWorld-Direct* with *Agro-Reflective* shows the value of self-correction. Although 35% of initial code generations contain bugs, the reflection loop recovers 92% of them, turning most execution failures into successful retrievals.

4.4 Efficiency Analysis

We further investigate the cost-performance tradeoff. Figure 2 illustrates the performance scaling with respect to the inference budget T. The curve follows a logarithmic saturation pattern. The steepest performance gain occurs in the first few turns ($T = 1 \rightarrow 4$), where the accuracy jumps from 48.2% to 68.5%. This initial boost corresponds to the agent fixing "low-hanging fruits"—simple syntax errors and API parameter mismatches. As T increases from 4 to 20, the gains become marginal (+5.3%), representing the correction of deeper semantic errors. Importantly, although we allocate a budget of $T = 20$, the *average* turns required for convergence is only **3.5**. This indicates that our verifiable protocol $\mathcal{V}$ is highly effective at early stopping: once the agent produces an output that satisfies the schema and physical constraints, the loop terminates, avoiding wasteful computation.

4.5 Robustness to OOD Scenarios

Spatial OOD (Brazil vs. California). As in Table 5, we test on fields in Brazil, which were excluded from the LoRA training set. Text-only models see a catastrophic performance drop ($\Delta = -55\%$), likely because they overfit to the specific crop calendars of the US (e.g., assuming harvest in October, which is incorrect for the Southern Hemisphere). AGRO-REFLECTIVE, however, maintains robustness ($\Delta = -12\%$) because it does not guess; it queries the local weather stations and satellite data dynamically. **Temporal OOD (The El Niño Year).** Testing on data from 2023, a year marked by extreme weather anomalies, reveals that text models relying on historical averages fail to predict the unusual yield losses. Our agent, by accessing real-time sensor streams, captures the signal of the anomaly and adjusts its predictions accordingly ($\Delta = -6\%$).

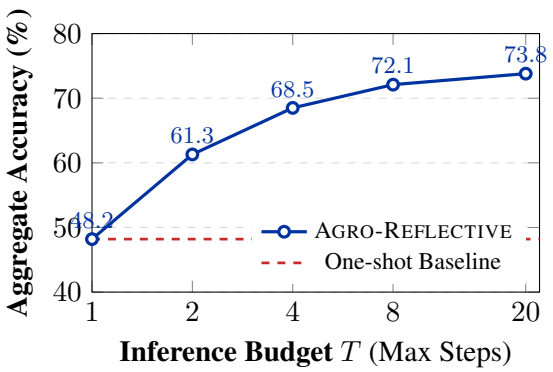

Figure 2: **Performance Scaling.** Accuracy saturates logarithmically, showing efficient self-correction.

Table 5: **OOD Robustness Analysis.** Comparison of accuracy degradation (Δ) on unseen regions (Spatial) and years (Temporal). Text models fail to generalize, while tool-grounded agents remain robust.

Method	IID	Spatial OOD		Temporal OOD	
	Acc.	Acc.	Δ	Acc.	Δ
Text-Only	41.2	18.5	-55%	22.1	-46%
One-shot	48.2	26.4	-45%	35.8	-26%
Agro-Reflective	**73.8**	**64.5**	**-12%**	**69.2**	**-6%**

5 Conclusion

In this work, we introduced AGRIWORLD, a World-Tools-Protocol framework that bridges Large Language Models with verifiable agricultural reasoning. Grounded in a Python-based execution environment, AGRIWORLD handles heterogeneous spatiotemporal data through automatic spatial alignment and temporal aggregation. Building on this environment, we developed AGRO-REFLECTIVE, an agent that follows an execute-observe-refine loop to self-correct errors using inspectable intermediate artifacts. Evaluations on the verifiable AGROBENCH dataset show that execution-driven reflection substantially outperforms both text-only models and direct one-shot tool-use baselines. Overall, this framework provides a robust, reproducible, and verifiable pathway for deploying AI assistants in real-world agronomic and scientific workflows.

6 Limitations

While AGRIWORLD and AGRO-REFLECTIVE significantly improve verifiable reasoning, several limitations remain. First, the framework's performance is intrinsically tied to the availability and quality of external data streams. In geographic regions with sparse meteorological stations or persistent cloud cover that disrupts remote sensing observations, the agent may face low-coverage errors or degraded forecasting capabilities, despite robust zonal statistics mechanisms.

Second, the fidelity of counterfactual and predictive reasoning is fundamentally bounded by the underlying agronomic simulators (e.g., DSSAT, APSIM). If these mechanistic models fail to account for complex biological nuances—such as undocumented pest outbreaks or compounded extreme weather events—the agent's proposed interventions, while mathematically and causally valid within the sandbox environment, may not perfectly translate to real-world efficacy.

Finally, the execute-observe-refine loop introduces a natural computational overhead. Although the verifiable protocol efficiently terminates early with an average of 3.5 turns to convergence, multi-step spatial joins, dynamic temporal resampling, and iterative code execution are inherently more latent and resource-intensive than single-pass text generation. Scaling this framework to continuous, real-time monitoring across millions of parcels will require further systems-level optimization in both the agent's planning efficiency and the environment's execution speed.

References

Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. 2002. Automated performance and dependability evaluation using model checking. In *IFIP International Symposium on Computer Performance Modeling, Measurement and Evaluation*, pages 261–289. Springer.

Jiabin Chen, Haiping Wang, Jinpeng Li, Yuan Liu, Zhen Dong, and Bisheng Yang. 2025. Spatialllm: From multi-modality data to urban spatial intelligence. *arXiv preprint arXiv:2505.12703*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

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

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14953–14962.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, and 1 others. 2020. Array programming with numpy. *nature*, 585(7825):357–362.

Dean P Holzworth, Neil I Huth, Peter G deVoil, Eric J Zurcher, Neville I Herrmann, Greg McLean, Karine Chenu, Erik J van Oosterom, Val Snow, Chris Murphy, and 1 others. 2014. Apsim–evolution towards a new generation of agricultural systems simulation. *Environmental Modelling & Software*, 62:327–350.

Stephan Hoyer and Joseph J. Hamman. 2017. xarray: N-d labeled arrays and datasets in python. *Journal of Open Research Software*, 5(1):10.

Alfredo Huete, Kamel Didan, Tsuguki Miura, Epifanio P. Rodriguez, Xiang Gao, and Leeuwen Ferreira. 2002. Overview of the radiometric and biophysical performance of the modis vegetation indices. *Remote Sensing of Environment*, 83(1–2):195–213.

John D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Daoyuan Jin, Nick Gunner, Niko Carvajal Janke, Shivranjani Baruah, Kaitlin M Gold, and Yu Jiang. 2025. Aleks: Ai powered multi agent system for autonomous scientific discovery via data-driven approaches in plant science. *arXiv preprint arXiv:2508.19383*.

JW JONES, G HOOGENBOOM, CH PORTER, KJ BOOTE, WD BATCHELOR, and LA HUNT. 2003. The dssat cropping system model. *European journal of agronomy*, 18(3-4):235–265.

Andreas Kamilaris and Francesc X. Prenafeta-Boldú. 2018. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70–90.

Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. 2017. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782.

Keyu Li, Junhao Shi, Yang Xiao, Mohan Jiang, Jie Sun, Yunze Wu, Shijie Xia, Xiaojie Cai, Tianze Xu, Weiye Si, and 1 others. 2026. Agencybench: Benchmarking the frontiers of autonomous agents in 1m-token real-world contexts. *arXiv preprint arXiv:2601.11044*.

Konstantinos G. Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, and Dionysis Bochtis. 2018. Machine learning in agriculture: A review. *Sensors*, 18(8):2674.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2024. Agentbench: Evaluating llms as agents. In *The Twelfth International Conference on Learning Representations*.

Ziming Luo, Zonglin Yang, Zexin Xu, Wei Yang, and Xinya Du. 2025. Llm4sr: A survey on large language models for scientific research. *arXiv preprint arXiv:2501.04306*.

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

Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, and Aixin Sun. 2024. Sciagent: Tool-augmented language models for scientific reasoning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 15701–15736.

Aman Madaan, Niket Tandon, Peter Clark, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Wes McKinney. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 56–61.

David J Mulla. 2013. Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems engineering*, 114(4):358–371.

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*.

Shishir G. Patil, Tianjun Zhang, Xin Wang, Joseph Gonzalez, and 1 others. 2023. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*.

Vignesh Prabhakar, Md Amirul Islam, Adam Atanas, Yao-Ting Wang, Joah Han, Aastha Jhunjhunwala, Rucha Apte, Robert Clark, Kang Xu, Zihan Wang, and 1 others. 2025. Omniscience: A domain-specialized llm for scientific reasoning and discovery. *arXiv preprint arXiv:2503.17604*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations*.

Marc Russwurm and Marco Korner. 2018. Multi-temporal land cover classification with sequential recurrent encoders. *ISPRS International Journal of Geo-Information*, 7(4):129.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Compton J. Tucker. 1979. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8(2):127–150.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, and 1 others. 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

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

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024b. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Sjaak Wolfert, Lan Ge, Cor Verdouw, and Marc-Jeroen Bogaardt. 2017. Big data in smart farming—a review. *Agricultural Systems*, 153:69–80.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Wenjia Xu, Zijian Yu, Boyang Mu, Zhiwei Wei, Yuanben Zhang, Guangzuo Li, Jiuniu Wang, and Mugen Peng. 2024. Rs-agent: Automating remote sensing tasks through intelligent agent. *arXiv preprint arXiv:2406.07089*.

Lian Yan, Haotian Wang, Chen Tang, Haifeng Liu, Tianyang Sun, Liangliang Liu, Yi Guan, and Jingchi Jiang. 2026. Agrieval: A comprehensive chinese agricultural benchmark for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34205–34213.

Bo Yang, Yu Zhang, Lanfei Feng, Yunkui Chen, Jianyu Zhang, Xiao Xu, Nueraili Aierken, Yurui Li, Yuxuan Chen, Guijun Yang, and 1 others. 2025. Agrigpt: A large language model ecosystem for agriculture. *arXiv preprint arXiv:2508.08632*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

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

Xiaoyang Zhang, Mark A. Friedl, Crystal B. Schaaf, Alan H. Strahler, and 1 others. 2003. Monitoring vegetation phenology using modis. *Remote Sensing of Environment*, 84(3):471–475.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.