MÍMIR: PHYSICS-GROUNDED LLM AGENTS FOR LONG-HORIZON IRRIGATION CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents increasingly combine reasoning, tool use, and action, but most evidence comes from episodic tasks with relatively immediate feedback and reset failures. Long-running physical control operates in a different regime: actions alter future states, errors compound across decisions, and an agent must improve from experience without being allowed to rewrite the physical rules that make execution safe. We study this regime through irrigation, where daily decisions interact with soil-water dynamics over entire growing seasons. We present Mímir, a physics-grounded LLM agent organized around two repair timescales. At the fast timescale, a structured physical interface and deterministic simulator turn an LLM output into a proposal that we numerically check, revise, and subject to bounded deterministic action selection before execution. At the slow timescale, recurrent failure patterns are consolidated into persistent contextual principles that condition future proposals, while the physical model, evaluator, and execution constraints remain immutable. Under a common retrospective evaluator across multiple sites, crops, and years, Mímir attains the lowest reported aggregate control cost among the evaluated references and uses about 51% less irrigation than the historical schedule replay. The ablation study show higher control cost when forward simulation, verified revision, or persistent context is removed; model-scale and model-family studies show no monotonic gain from increasing LLM size. The resulting lesson show that persistent physical agents can combine semantic reasoning with bounded, evidence-driven self-improvement while reserving physical truth and actuator authority for explicit numerical mechanisms.

1 INTRODUCTION

Large language models (LLMs) are increasingly used as agents that interleave reasoning, tool use, memory, and action Yao et al. (2022). In embodied settings, language models can ground high-level decisions in perception, affordances, and external tools Ahn et al. (2022); Driess et al. (2023); Huang et al. (2022; 2023). However, most agent evaluations remain episodic: a task starts, the agent acts for a finite horizon, and the episode terminates. Persistent physical control is different. Each action changes the state on which later decisions depend; consequences can be delayed; and a locally plausible mistake can persist through the physical system long after the language-model call that produced it. A physical agent must therefore do more than generate a reasonable action. It need to remain grounded in numerical dynamics, repair bad decisions before actuation, and accumulate useful experience without allowing that experience to erode execution safety.

Irrigation provides a concrete instance of this regime. A controller repeatedly decides how much water to apply under evolving soil, crop, weather, and forecast conditions. Irrigation changes root-zone water storage and therefore the context for future decisions; excessive application consumes water and can increase drainage, whereas insufficient application can accumulate crop-water deficit Allen et al. (1998). The effect of today's decision may only become evident after subsequent evapotranspiration, rainfall, and additional control actions. Physics-based scheduling, model predictive control (MPC), sensor-driven irrigation, and reinforcement learning already provide strong solutions when the state, model, objective, and controller are sufficiently specified Allen et al. (1998); Delgoda et al. (2016); Garcia et al. (1989); Dong et al. (2024); Yu et al. (2026); Ding & Du (2024). We therefore do *not* ask whether an LLM should replace these numerical controllers. We ask a narrower systems-and-learning

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

question: *if an LLM is introduced as a reasoning component of a long-running physical controller, which responsibilities can be adaptive and which must remain explicit, numerical, and externally enforceable?*

This question exposes three coupled challenges. **Q1: physical grounding.** Language models operate naturally over semantic representations, whereas irrigation is governed by quantities with units, conservation relationships, state transitions, and admissible action bounds. The agent should not be asked to reconstruct these relationships implicitly from prose. **Q2: decision repair.** A physically feasible action is not necessarily a desirable action. Multiple feasible actions can induce different delayed stress, drainage, and water-use consequences, so a first language-model proposal should be treated as a hypothesis to verify but not a device command. **Q3: experience-driven adaptation.** Recurrent decision errors may become visible only after repeated operation. A persistent agent should learn from those errors, but learning in a physical system raises an additional governance question: what is allowed to change, and what must remain invariant?

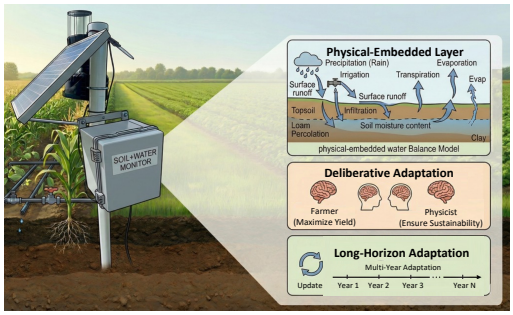


Figure 1: Mímir Physical-Embedded Irrigation Control.

We present Mímir, a two-timescale, physics-grounded LLM agent that answers these questions by separating *reasoning authority* from *physical authority*. At the fast timescale, a deterministic soil-water model converts observations into a structured physical state and feasible action region. The LLM proposes and revises a decision; programmatic simulation exposes the consequences of that proposal; a deterministic selector ranks a bounded candidate set; and runtime assurance retains final execution authority. At the slow timescale, recurring corrections and outcome patterns are distilled into persistent contextual principles that can bias future proposals. Importantly, this adaptation don't need to modify the transition model, evaluator, action bounds, or execution gate. The LLM therefore contributes contextual interpretation, proposal generation, and revision.

Recent continual and self-improving agents increasingly treat memory, prompts, and other framework components as objects that can change from experience without updating model weights Kim et al. (2024); Sridhar et al. (2026); Ren et al. (2026); Feng et al. (2026). Mímir adopts a deliberately bounded version of this idea: experience may repair future reasoning context, but not the trusted physical substrate. Fast-loop repair ends when verification no longer changes the proposal or its fixed budget is exhausted; slow-loop memory changes only when recurring evidence crosses a configured trigger. We therefore use *bounded adaptation*. This boundary distinguishes Mímir from language-only refinement Madaan et al. (2023); Du et al. (2024); Wu et al. (2026) and from broader self-adaptive physical agents Kaushik et al. (2026): language agreement is not physical evidence, memory cannot relax execution constraints, and an LLM proposal never directly becomes an actuator command.

We evaluate Mímir retrospectively using field-observed data from Michigan, Texas, and Colorado spanning multiple crops and years. In the reported multi-site comparison, Mímir attains an aggregate control cost of 100.94 ± 1.80 with 2.58 ± 0.12 mm irrigation under the common evaluator, while the historical schedule replay uses 5.32 ± 0.57 mm. Mechanism-specific matched ablations show higher control cost when forward simulation, verified proposal revision, or persistent context is removed. A separate nine-season sequence exhibits substantial seasonal variation without sustained divergence. That shows the ability of long-horizon operational stability. Finally, performance across the LLM sizes remains in the same broad range across several model families, indicating that the control architecture surrounding the language model materially shapes behavior.

Our contributions are threefold:

- **Physics-grounded agent interface.** We formulate and instantiate an interface in which language reasoning operates over explicit physical state and deterministic transition tools, while state evolution, feasible actions, and execution constraints remain outside the language model.

Representative Paradigm	Physical Grounding	Deliberative Reasoning	Persistent Adaptation	Independent Runtime Safety
Model-based irrigation Allen et al. (1998); Delgoda et al. (2016)	✓	–	○	✓
Learning-based irrigation Yang et al. (2020); Ding & Du (2024)	○	–	○	○
Embodied LLM agents Ahn et al. (2022); Driess et al. (2023); Yao et al. (2022)	○	○	–	–
Self-improving agents Shinn et al. (2023); Wang et al. (2023); Zhang et al. (2025)	–	✓	✓	–
Runtime-assured control Ames et al. (2019); Mehmood et al. (2022); Wachi & Sui (2020)	✓	–	–	✓
Mímir	✓	✓	✓	✓

Table 1: Positioning of Mímir relative to representative research paradigms. ✓denotes a primary mechanism, ○ denotes partial support, and – denotes that the capability is not a primary focus.

- **Simulator-grounded decision repair.** We develop a bounded propose-verify-revise mechanism in which an LLM proposal is confronted with numerical rollout evidence before a deterministic selector ranks executable candidates.
- **Bounded experience-driven adaptation.** We integrate evidence-triggered persistent context that can repair future proposal priors from recurring experience while leaving the physical model, evaluator, and runtime constraints immutable.

2 Related Work

Irrigation and long-horizon control. Irrigation scheduling is commonly formulated as a physical control problem linking soil-water availability, crop demand, weather forcing, and irrigation actions. FAO-56 provides widely used evapotranspiration and crop-water formulations Allen et al. (1998); MPC optimizes future actions under explicit dynamics and constraints Delgoda et al. (2016); Garcia et al. (1989); and learning-based approaches optimize policies from simulated or observed interaction Yang et al. (2020); Ding & Du (2024); Kelly & Foster (2021). These approaches are strong control baselines and, when the control problem is fully specified, do not require language reasoning. Agricultural LLM systems increasingly incorporate sensor and weather context for irrigation advice or decision support Kandamali et al. (2025); Yu et al. (2026). Our question is different: how should a language-model agent participate in repeated closed-loop control when numerical prediction and execution authority must remain explicit?

Grounded, embodied, and verifiable LLM agents. Grounded agents couple language reasoning to external state, tools, or executable capabilities. SayCan grounds plans in robot affordances Ahn et al. (2022), PaLM-E incorporates embodied observations Driess et al. (2023), and ReAct interleaves reasoning with actions and observations Yao et al. (2022). Iterative methods such as Self-Refine and multi-agent debate revisit an initial generation through critique or interaction Madaan et al. (2023); Du et al. (2024), while VIRF couples an LLM planner to a deterministic logic tutor for verifiable plan repair Wu et al. (2026). Mímir shares the principle that generative proposals should be externally checked, but continuous physical control imposes a different contract: numerical simulation determines state consequences, deterministic logic ranks bounded actions, and runtime assurance remains independent of language-model agreement.

Persistent memory, continual learning, and self-improving agents. A growing literature studies agents that accumulate experience. Reflexion stores verbal feedback Shinn et al. (2023), Voyager accumulates reusable skills Wang et al. (2023), online continual learning has been studied for

embodied instruction-following agents Kim et al. (2024), and experience retrieval improves long-horizon robotic control Sridhar et al. (2026). Agentic Context Engineering (ACE) treats context as an evolving playbook Zhang et al. (2025). More recent work frames changes to prompts, memory, tools, or controller frameworks as forms of agent self-improvement Ren et al. (2026), and proposes broader self-evolving embodied systems that adapt memory, task knowledge, or models over time Feng et al. (2026). Concurrent physical-agent work similarly investigates self-adaptive crop management with verification and reflection Kaushik et al. (2026).

Mímir occupies a deliberately narrower point in this design space. It performs no model-weight update, no online world-model revision, and no autonomous relaxation of safety rules. Experience can change only the contextual framework used for future proposals. This boundedness is important in physical control because continual adaptation and runtime assurance solve different problems: the former determines what the agent should remember or reconsider, while the latter determines what it is allowed to execute. Control barrier functions, Simplex-style runtime assurance, and safe RL provide complementary mechanisms for maintaining explicit safety conditions Ames et al. (2019); Mehmood et al. (2022); Wachi & Sui (2020). The contribution of Mímir is a two-timescale architecture that permits experience-driven reasoning repair while preserving an immutable numerical control boundary.

3 Problem Formulation

We study persistent physical control in which an agent repeatedly observes an evolving environment, proposes an action, and receives feedback only through subsequent state evolution. At decision step t, observations o_t are mapped by a fixed physical abstraction to state

$$s_t = \phi(o_t; \theta_{\text{phys}}), \tag{1}$$

and the process evolves as

$$s_{t+1} = f(s_t, a_t, w_t), \qquad a_t \in \mathcal{A}(s_t), \tag{2}$$

where w_t denotes exogenous forcing and $\mathcal{A}(s_t)$ is the feasible action set. Over horizon T, the controller seeks low cumulative physical cost,

$$J(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} c(s_t, a_t)\right], \qquad \text{s.t. Eq. 2.} \tag{3}$$

Irrigation instantiates this problem with root-zone water state, precipitation, and atmospheric demand as forcing, and irrigation depth as the action.

Our objective is to decompose control into an adaptive semantic layer and a trusted numerical layer. Let x_t denote the structured decision context and m_e the persistent context for episode or season e. The LLM produces a proposal

$$z_t = g_\theta(x_t, m_e), \tag{4}$$

while the executed action is determined only after physical evaluation and assurance,

$$a_t = \mathcal{D}(s_t, z_t; f, \mathcal{A}), \tag{5}$$

where $\mathcal{D}$ denotes bounded rollout, candidate selection, runtime assurance, and actuation gating. This decomposition makes the allocation of authority explicit: g_θ can influence what is proposed or reconsidered, but f, $\mathcal{A}$, and $\mathcal{D}$ determine physical consequence and executable action.

Persistent experience acts on a slower timescale. Let τ_e denote the completed trajectory and outcome summary for episode e, and let q_e be the configured evidence trigger extracted from that experience. We write

$$m_{e+1} = \begin{cases} \mathcal{G}(m_e, \tau_e), & q_e = 1, \\ m_e, & q_e = 0, \end{cases} \tag{6}$$

where $\mathcal{G}$ modifies only the contextual framework supplied to future LLM calls. In particular, θ, f, θ_{phys}, the evaluator, and runtime constraints are fixed by construction. Equation 6 formalizes the limited sense in which the agent can improve from experience: the reasoning context can be repaired when recurring evidence warrants an update, but the trusted physical substrate cannot self-modify.

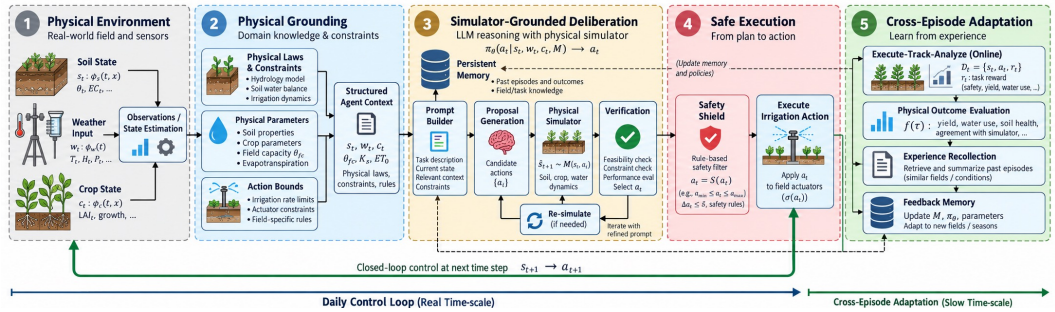


Figure 2: Mímir as a two-timescale physical agent. In the daily loop, observations are converted to a structured physical state; the LLM proposes or revises a decision; runtime assurance gates execution. In the slower loop, recurring failures are consolidated into persistent context. Adaptation only changes the reasoning context.

This exposes three questions. **Q1** How should x_t show physically meaningful state and constraints to language reasoning? **Q2** how should a proposal z_t be checked and repaired before it can influence a_t? **Q3** how should m_e accumulate experience without changing physical truth or execution authority? Mímir addresses these questions through a physics-grounded interface, simulator-grounded decision repair, and bounded cross-episode adaptation.

4 Mímir Design

Figure 2 shows the central design principle: adaptive reasoning and physical authority are intentionally asymmetric. The language model may propose, explain, and revise. Numerical tools determine predicted consequences. Deterministic logic constrains and ranks actions. Runtime assurance retains final actuator authority. Persistent memory may influence future reasoning, but it cannot expand what the physical controller permits. This creates two repair loops: a fast loop corrects individual proposals using physical evidence, while a slow loop consolidates recurring experience into contextual principles. Neither loop can alter physical state-transition rules or execution constraints.

4.1 Physics-Grounded Agent Interface

Mímir exposes a structured numerical state. Let S_t denote root-zone water storage, P_t precipitation, ET_t evapotranspiration, I_t irrigation, and D_t drainage. The controller-side model uses the lightweight balance

$$S_{t+1} = S_t + P_t + I_t - ET_t - D_t. \tag{7}$$

The reasoning context contains root-zone storage, field capacity (FC), wilting point, crop stage, evapotranspiration, and decision-time forecast information. This representation is an interface contract in which units, state variables, and admissible quantities are fixed before querying the LLM. Appendix A specifies the physical abstraction and its assumptions.

The state also induces an execution-feasible action interval. With anticipated retained precipitation $\widehat{P}_{\mathrm{eff},t+1}$,

$$I_t^{\max} = \max\left(0, FC - S_t - \widehat{P}_{\mathrm{eff},t+1}\right), \qquad \mathcal{A}_t = [0, I_t^{\max}]. \tag{8}$$

The final command is projected into this interval,

$$a_t = \Pi_{\mathcal{A}_t}(\widetilde{a}_t). \tag{9}$$

This is an upper-bound execution constraint. The LLM cannot expand $\mathcal{A}_t$, alter units, or change the transition parameters used by the controller.

4.2 Simulator-Grounded Decision Repair

Physical feasibility alone does not determine which feasible action is preferable. Mímir therefore treats language output as a proposal to be repaired if physical evidence contradicts its rationale. Given

structured context x_t, the proposer generates

$$\widetilde{a}_t^{(0)} = \mathrm{Propose}(x_t). \tag{10}$$

At repair round k, programmatic tools evaluate the proposed consequence,

$$e_t^{(k)} = \mathrm{Verify}(s_t, \widetilde{a}_t^{(k)}), \tag{11}$$

and the proposal can be revised using the resulting numerical evidence,

$$\widetilde{a}_t^{(k+1)} = \mathrm{Revise}(x_t, \widetilde{a}_t^{(k)}, e_t^{(k)}). \tag{12}$$

The implementation names the proposal and verification roles "Farmer" and "Physicist". The distinction is between generative proposal and programmatic verification. The fast repair loop terminates when verification no longer changes the proposal or when the fixed round budget is reached, so the agent cannot deliberate indefinitely.

The repaired proposal remains only an *anchor*. A deterministic selector constructs a finite set $\mathcal{C}_t \subseteq \mathcal{A}_t$ around the anchor while retaining reference candidates such as zero irrigation and the admissible refill bound. Each candidate is rolled out over the control horizon,

$$\widehat{\tau}_t(a) = \mathrm{Rollout}(s_t, a, w_{t:t+H}), \tag{13}$$

and assigned fixed rollout cost $J_t(a)$. The selector chooses

$$a_t^\star = \arg\min_{a \in \mathcal{C}_t} \left(N_{\mathrm{viol}}(a), J_t(a) \right), \tag{14}$$

using lexicographic priority on modeled physical violations and then rollout cost. The selected candidate passes runtime assurance and a trusted fallback path before the final projection in Eq. 9.

Language can therefore change the search prior, but not physical evidence or execution authority. The corresponding ablation tests whether numerical feedback improves the LLM-conditioned proposal, not whether language agents replace optimization.

4.3 Bounded Experience-Driven Adaptation

The fast loop repairs an individual decision, but persistent agents also encounter repeated failure modes. A proposal may repeatedly overreact to forecast uncertainty, require the same verifier correction, or reach the same runtime safeguard across multiple episodes. Mímir therefore maintains a slower adaptation loop that converts recurring experience into reusable contextual knowledge.

Let τ_e be the completed trajectory and outcome summary for episode or season e. We implement $\mathcal{G}$ in Eq. 6 using Agentic Context Engineering Zhang et al. (2025). Deliberation traces, repeated verifier corrections, control-cost patterns, and runtime interventions provide evidence from which we can distill compact principles. A principle records the context in which a pattern occurred, the corrective tendency, relevant exceptions, and its supporting evidence. When the configured recurrence trigger is not met, the slow loop performs no update; thus adaptation is event-driven rather than an unconditional rewrite after every season.

Memory can condition later proposals or provide a soft anchor, but cannot rank or execute actions. Across updates,

$$m_e \to m_{e+1}, \tag{15}$$

while

$$g_\theta\text{'s weights}, \quad f_{\mathrm{phys}}, \quad \theta_{\mathrm{phys}}, \quad \mathcal{A}_t, \quad \text{Evaluator}, \quad \text{Runtime Assurance} \tag{16}$$

remain fixed. Memory cannot access future held-out outcomes, relax a safety bound, change the scoring rule, or bypass deterministic selection.

We call this *bounded experience-driven adaptation*: framework-level self-improvement in which accumulated experience can change future reasoning context, but not model weights, the physical world model, or runtime constraints. The central question is therefore not only whether a physical agent can learn, but *which parts are permitted to learn and under what evidence*. Appendix H gives the full memory contract.

5 EXPERIMENTAL SETUP

5.1 DATASETS AND PROTOCOL

The corpus contains approximately 6,938 days, or 19 crop-years, spanning four environments: Michigan potato under humid, variable rainfall Dong et al. (2024); Bushland, Texas maize under high evaporative demand Evett et al. (2024); Colorado LIRF maize under limited-irrigation treatments Trout & Bausch (2017); and a separate Colorado sunflower sequence for the nine-season analysis Trout et al. (2025). Available streams include hourly weather forcing, irrigation records, evapotranspiration-related variables, and soil-water observations. Appendix B summarizes their role and describes provenance and preprocessing.

For each eligible chronological segment, the first 60% is reserved for calibration or baseline training and the remaining 40% for evaluation; samples are never shuffled. Decisions are issued every 24 h using a 168 h history and a 24 h forward horizon, while the physical model integrates hourly. At time t, a method can use only information available by that decision time. All methods share the same forecast construction, actuator limits, feasible action bounds, and evaluation window. The protocol forbids future reads, initialization or repair of persistent context from future held-out outcomes, test-based seed selection, and overlap between calibration and evaluation windows. The nine-season sequence preserves chronological order and is analyzed separately from the main aggregate.

5.2 REFERENCE CONTROLLERS AND COMPARISON CONTRACT

We report four reference classes. **Historical irrigation** replays recorded irrigation actions under the common evaluator and serves as a management reference; it is not a comparison of realized crop yield. **MAD-50** irrigates after 50% management-allowable depletion and refills toward field capacity subject to the common action limit FAO (2017). **ET-Replace** replaces accumulated evapotranspiration loss using the same causal weather information. **DRLIC** provides the reinforcement-learning reference Ding & Du (2024). Appendix E additionally describes zero irrigation, fixed-threshold control, and a deterministic physics-search/MPC reference used diagnostically to separate language-conditioned anchoring from numerical action search.

5.3 METRICS

The primary metric is seasonal physical control cost,

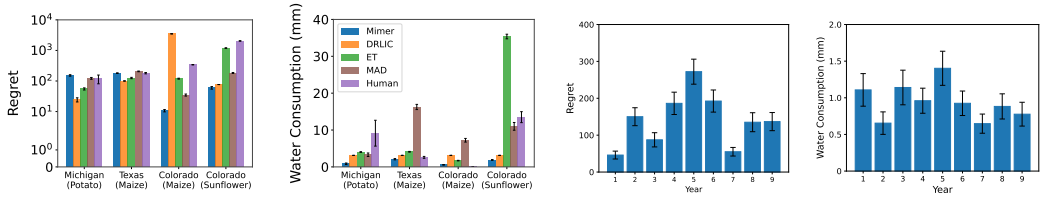
$$C = w_W W + w_S S + w_R R + w_L L, \tag{17}$$

where W is irrigation applied, S accumulated soil-water stress deficit, R runoff, and L deep percolation/leaching. All terms are computed from the same hourly trajectory and use a single fixed set of weights across sites, controllers, and language-model backbones. Lower C is better. We call this quantity *control cost*. It is an absolute trajectory objective. We separately report irrigation applied in millimeters; lower water use is interpreted only jointly with control cost, because a controller can save water by accepting larger stress. Appendix D defines the components and aggregation protocol.

5.4 IMPLEMENTATION DETAILS

Unless stated otherwise, Mímir uses a DeepSeek-14B backbone served locally through Ollama. Fast-loop decision repair is capped at three proposal-verification-revision rounds. Every proposal is followed by deterministic candidate construction, rollout scoring, runtime assurance, fallback if required, and the final actuation gate; the LLM never writes directly to the actuator.

The slow loop changes only persistent context. Recurrence triggers, memory budget, activation/reset rules, and the representation of stored principles are fixed by configuration within each matched experiment. A season with no qualifying recurring pattern produces no contextual update. For model-scale and model-family studies, only the language-model checkpoint changes; prompts, physical state, simulator, action bounds, repair budget, candidate selection, and evaluation protocol remain fixed. Appendix F reports prompting and parsing details, and Appendix H details the persistent-context update contract.



(a) Regional control cost. (b) Regional irrigation. (c) Annual control cost. (d) Annual irrigation.

Figure 3: Regional results for Michigan potato, Texas maize, Colorado maize, and Colorado sunflower environments. Nine-season Colorado sequence. This experiment evaluates operational stability, not monotonic self-improvement.

Table 2: Aggregate performance under the common multi-site evaluation protocol. Control cost is the primary objective; irrigation is reported separately because lower water alone can correspond to higher stress.

Controller	**Control cost ↓**	**Irrigation (mm)**
Mímir	$\mathbf{100.94 \pm 1.80}$	2.58 ± 0.12
MAD-50	134.33 ± 3.10	10.58 ± 0.44
ET-Replace	440.74 ± 12.48	12.70 ± 0.31
DRLIC	924.47 ± 27.50	0.35 ± 0.00
Historical irrigation	790.51 ± 17.76	5.32 ± 0.57

6 Evaluation

In the evaluation We first ask whether the complete controller attains useful long-horizon performance under a common protocol. We then test the three proposed capabilities in order: physical grounding (Q1), verified decision repair (Q2), and experience-driven adaptation (Q3). Because several component ablations were produced on different matched evaluation slices, each effect is interpreted only against the corresponding full configuration.

6.1 Multi-Site and Multi-Year Control Performance

Table 2 reports the multi-site aggregate. Mímir obtains control cost 100.94 ± 1.80 with 2.58 ± 0.12 mm irrigation. MAD-50 records 134.33 ± 3.10 at 10.58 ± 0.44 mm, and ET-Replace records 440.74 ± 12.48 at 12.70 ± 0.31 mm. DRLIC applies the least water (0.35 ± 0.00 mm) but incurs the largest reported cost (924.47 ± 27.50), illustrating why water use cannot be interpreted independently of stress and loss terms. Historical irrigation replay uses 5.32 ± 0.57 mm and has cost 790.51 ± 17.76 under the same evaluator. Within this reported set, Mímir therefore occupies the lowest-cost operating point; this comparison is not a claim of dominance over all model-based irrigation controllers.

The regional view in Fig. 3a and Fig. 3b checks whether the aggregate behavior is concentrated in one environment. Mímir records 152.25 ± 5.91 cost and 1.54 ± 0.38 mm irrigation in Michigan, 129.16 ± 2.71 and 3.80 ± 0.31 mm in Texas, and 59.82 ± 4.07 and 3.43 ± 0.19 mm in the reported Colorado evaluation. Because horizon length and forcing differ across crops and sites, we do not use raw regional cost to rank environments; the figure is used only to assess whether the same controller remains viable under distinct environmental conditions.

6.2 Performance of Explicit Physical Grounding

We ablate two distinct responsibilities of the physical substrate: predicting delayed consequences and enforcing executable bounds. Removing forward simulation increases control cost from 149.5 to 730.5 on its matched slice, while irrigation falls from 3.29 to 0.45 mm. The joint decrease in water and increase in cost is consistent with under-irrigation whose delayed deficit is not represented during proposal evaluation. Removing the execution shield increases cost from 147.2 to 259.9 while mean irrigation changes comparatively little. The second result indicates a different role: execution

Table 3: Paired component ablations. Each row has its own matched full configuration; absolute full values differ across slices and must not be compared between rows. Water is in mm.

	Full		Ablated	
Removed component	**Cost**	**Water**	**Cost**	**Water**
Persistent context (ACE)	103.0	3.40	157.5	4.41
Decision repair	109.1	2.47	180.5	3.19
Execution shield	147.2	–	259.9	$\approx$ unchanged
Forward simulator	149.5	3.29	730.5	0.45

bounding can alter costly tail behavior without determining the controller's mean irrigation level. Together, the two comparisons support the bounded claim that explicit physical prediction and execution constraints materially affect behavior under the tested conditions. Neither comparison guarantees future agronomic safety. The direct/zero-shot LLM diagnostic used a different metric configuration and is therefore reported only in Appendix J; its scalar value is not combined with the matched ablations.

6.3 Performance of Simulator-Grounded Decision Repair

We remove the proposal–verification–revision stage while retaining the downstream simulator, candidate selector, and execution constraints. On the matched slice, control cost increases from 109.1 to 180.5 and irrigation from 2.47 to 3.19 mm. Because the two variants share the same downstream physical substrate, the difference isolates the value of revising the language-conditioned anchor using explicit numerical feedback. The result supports a narrow conclusion: verified repair improves the evaluated objective on this slice relative to the corresponding single-pass proposal. It does not establish that free-form multi-agent debate is superior to every alternative inference-time strategy, nor that the language model replaces deterministic search.

6.4 Performance of Cross-Year Experience

Removing persistent ACE context increases control cost from 103.0 to 157.5 and irrigation from 3.40 to 4.41 mm on its matched slice (Table 3). The physical simulator, candidate selector, and execution shield are unchanged. The comparison therefore supports the statement that information carried across episodes improves later proposal quality beyond the instantaneous physical state under this protocol. It is evidence for useful experience accumulation, not for monotonic continual learning.

We separately examine nine consecutive Colorado seasons. Figure 3c and Figure 3d shows substantial year-to-year variation, including a maximum reported annual cost of 272.15 in Year 5, followed by lower values rather than sustained divergence. We interpret this experiment as evidence of operational stability across changing seasonal forcing. It does not establish that the agent improves every year, that every memory update is beneficial, or that the memory mechanism is immune to forgetting. Those stronger properties would require dedicated continual-learning evaluation beyond the current experiments.

7 Conclusion

We presented Mímir, a physics-grounded LLM agent for long-horizon irrigation control. The design begins from a simple premise: persistent physical agents need the ability to adapt, but not every part of a physical controller should be equally free to change. Mímir therefore separates three responsibilities. A structured physical interface defines state and feasible actions; simulator-grounded decision repair lets language-model proposals be challenged with numerical evidence before deterministic selection; and bounded cross-episode context carries recurring experience forward without changing the physical model or execution constraints. Across the reported multi-site and multi-year evaluation, matched ablations show that physical simulation, verified repair, and persistent context each affect control behavior. The resulting contribution shows one way to build a persistent physical agent in which reasoning can evolve from experience while physical truth and actuator authority remain explicitly governed.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Richard G Allen, Luis S Pereira, Dirk Raes, Martin Smith, et al. Crop evapotranspiration-guidelines for computing crop water requirements-fao irrigation and drainage paper 56. *Fao, rome*, 300(9): D05109, 1998.

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pp. 3420–3431. Ieee, 2019.

Dilini Delgoda, Hector Malano, Syed K Saleem, and Malka N Halgamuge. Irrigation control based on model predictive control (mpc): Formulation of theory and validation using weather forecast data and aquacrop model. *Environmental Modelling & Software*, 78:40–53, 2016.

Xianzhong Ding and Wan Du. Optimizing irrigation efficiency using deep reinforcement learning in the field. *ACM Transactions on Sensor Networks*, 20(4):1–34, 2024.

Younsuk Dong, Benjamin Werling, Zhichao Cao, and Gen Li. Implementation of an in-field iot system for precision irrigation management. *Frontiers in Water*, 6:1353597, 2024.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024. URL `https://proceedings.mlr.press/v235/du24e.html`.

Steven R. Evett, Gary W. Marek, Paul D. Colaizzi, Karen S. Copeland, and Brice B. Ruthardt. Spreadsheet for lysimeter data analysis, bushland, texas, 2024. Dataset.

FaAOotUN FAO. *Water for Sustainable Food and Agriculture: A report produced for the G20 Presidency of Germany*. 2017.

Tongtong Feng, Xin Wang, and Wenwu Zhu. Self-evolving embodied ai. *arXiv preprint arXiv:2602.04411*, 2026.

Carlos E Garcia, David M Prett, and Manfred Morari. *Model predictive control: Theory and practice—A survey*, volume 25. Elsevier, 1989.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Deus F. Kandamali, Wesley M. Porter, Erin Porter, Alex McLemore, and Glen C. Rains. Cottonbot: An ai-driven cotton farming assistant and irrigation advisor using llm-rag and agentic ai tools. *Smart Agricultural Technology*, 12:101640, 2025. doi: 10.1016/j.atech.2025.101640.

Varun Kaushik, Yayun Tan, and Xiaofan Yu. Toward self-adaptive physical ai: Can llm agents manage long-horizon physical tasks? *arXiv preprint arXiv:2609.13436*, 2026.

TD Kelly and Timothy Foster. Aquacrop-ospy: Bridging the gap between research and practice in crop-water modeling, 2021.

Byeonghwi Kim, Minhyuk Seo, and Jonghyun Choi. Online continual learning for interactive instruction following agents. In *International Conference on Learning Representations*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. volume 36, pp. 46534–46594, 2023.

Usama Mehmood, Sanaz Sheikhi, Stanley Bak, Scott A Smolka, and Scott D Stoller. The black-box simplex architecture for runtime assurance of autonomous cps. In *NASA formal methods symposium*, pp. 231–250. Springer, 2022.

Zhe Ren, Yimeng Chen, Dandan Guo, Guowei Rong, Tonghui Li, RB Xiong, Qingfeng Lan, Wenyi Wang, Li Nanbo, Yibo Yang, et al. Self-improvements in modern agentic systems: A survey. *arXiv preprint arXiv:2607.13104*, 2026.

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

Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. Scaling up memory for robotic control via experience retrieval. In *International Conference on Learning Representations*, volume 2026, pp. 97142–97166, 2026.

Thomas J. Trout and Walter C. Bausch. Usda-ars colorado maize water productivity data set. *Irrigation Science*, 35(3):241–249, 2017.

Thomas J Trout, Louise H Comas, Kendall C DeJonge, and Huihui Zhang. Colorado sunflower water use, physiology and productivity dataset. *Data in Brief*, pp. 111959, 2025.

Akifumi Wachi and Yanan Sui. Safe Reinforcement Learning in Constrained Markov Decision Processes. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research (PMLR)*, pp. 9797–9806. PMLR, 2020.

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. *arXiv preprint arXiv:2305.16291*, 2023.

Feiyu Wu, Xu Zheng, Yue Qu, Zhuocheng Wang, Zicheng Feng, and Hui Li. Grounding generative planners in verifiable logic: A hybrid architecture for trustworthy embodied ai. In *International Conference on Learning Representations*, 2026.

Yanxiang Yang, Jiang Hu, Dana Porter, Thomas Marek, Kevin Heflin, and Hongxin Kong. Deep reinforcement learning-based irrigation scheduling. *Transactions of the ASABE*, 63(3):549–556, 2020.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. 2022.

Lei Yu, Zhikai Li, Kai Huang, Fulin Li, Zixiang Lu, Yurui Fan, Yanan Jiang, Chenglong Zhang, and Guangtao Fu. Integrating llm-based agents with uncertainty-aware optimization for water-energy-carbon nexus management in irrigation districts. *Water Research*, pp. 126740, 2026.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. *arXiv preprint arXiv:2510.04618*, 2025.

APPENDIX OVERVIEW

This appendix provides additional methodological, implementation, evaluation, and reproducibility details supporting the main paper. Each entry below is hyperlinked to the corresponding section.

Sec.	Content	Purpose
A	**Physical Model and Execution Constraints**	Defines the physical state, root-zone water balance, feasible actions, and execution-time constraints.
B	**Dataset Provenance and Preprocessing**	Describes the multi-site datasets, preprocessing, temporal resolution, unit normalization, and causal data handling.
C	**Evaluation Protocol and Leakage Controls**	Details chronological splitting, causal-information constraints, matched comparisons, and safeguards against evaluation leakage.
D	**Metric Definition**	Specifies the seasonal physical control cost, component metrics, aggregation, and uncertainty reporting.
E	**Reference Controller Implementations**	Documents historical, heuristic, model-based, and learning-based irrigation references under the common evaluation interface.
F	**Language-Model Interface**	Reports the LLM configuration, structured prompting interface, decoding settings, parsing, fallback behavior, and telemetry.
G	**Simulator-Grounded Decision Repair**	Provides the proposal–verification–revision loop, bounded candidate search, deterministic selection, and a representative trace.
H	**Bounded Experience-Driven Adaptation**	Describes persistent contextual memory, evidence-triggered updates, stopping conditions, and immutable adaptation boundaries.
I	**Full Quantitative Results**	Reports aggregate and regional results underlying the main-paper comparisons.
J	**Component Ablations**	Provides matched component ablations and additional diagnostics.
K	**Foundation-Model Robustness**	Reports sensitivity to language-model scale and model family under an otherwise fixed physical-control architecture.
L	**Nine-Season Long-Horizon Analysis**	Provides year-by-year results used to assess long-horizon operational stability.
M	**Reproducibility and Provenance**	Summarizes configuration freezing, experiment provenance, artifact logging, and reproduction requirements.
N	**Discussion and Limitations**	Clarifies claim scope, adaptation boundaries, evidence limitations, and implications for persistent physical agents.
O	**AI Use Disclosure**	Discloses the use and verification of generative AI tools in the research workflow.

A PHYSICAL MODEL AND EXECUTION CONSTRAINTS

A.1 ROOT-ZONE STATE AND UNITS

All water quantities are represented as equivalent water depth in millimeters. When volumetric soil moisture θ_t (m^3/m^3) is available, the controller maps it to root-zone storage as

$$S_t = \theta_t Z_{\mathrm{root}}, \tag{18}$$

with $Z_{\mathrm{root}} = 600\,\mathrm{mm}$ in the reported configuration. This fixed depth is part of the controller abstraction.

The structured controller state contains root-zone storage S_t, field capacity FC, wilting point WP, crop-stage information, evapotranspiration, observed precipitation, and the decision-time precipitation input. We keep three quantities conceptually separate throughout the system: (i) physical storage limits, (ii) the agronomic stress threshold used by the evaluator, and (iii) the feasible action bound used by the controller.

Table 4: Dataset inventory used by the evaluation protocol.

Environment	Crop	Resolution	Sequence	Use
Michigan/LOCOSMOS	Potato	Hourly	2021–2023 entries	Main evaluation
Bushland, Texas	Maize	Hourly	Multi-year	Main evaluation
Colorado LIRF	Maize	Hourly	5-year sequence	Main evaluation
Colorado sunflower	Sunflower	Hourly	9-season sequence	Long-horizon analysis

A.2 Water-Balance Transition

The controller-side model evolves root-zone storage according to

$$S_{t+1} = S_t + P_t + I_t - ET_t - D_t, \tag{19}$$

where P_t, I_t, ET_t, and D_t denote precipitation, irrigation, evapotranspiration, and drainage/deep percolation, respectively. The bucket approximation uses

$$D_t = \max(0,\, S_t + P_t + I_t - ET_t - FC)\,. \tag{20}$$

The model is intentionally lightweight. It is used to construct and verify bounded irrigation decisions. Reference evapotranspiration is computed from the available meteorological forcing using FAO-56 Penman–Monteith where the required variables are available Allen et al. (1998); otherwise, the implementation follows a deterministic fallback path fixed before evaluation and shared by all controllers.

A.3 Effective Precipitation and Causal Forecast Input

The controller can map forecast precipitation to retained root-zone input using the fixed transform

$$P_{\text{eff}} = \begin{cases} P\frac{125-0.2P}{125}, & P \le 250, \\ 125 + 0.1P, & P > 250. \end{cases} \tag{21}$$

Here P denotes the precipitation quantity supplied through the controller's forecast interface, expressed in millimeters for the corresponding decision interval. We treat Eq. 21 strictly as a fixed implementation heuristic in the controller interface. Every controller receives the same transformed causal input, and no controller may access realized future rainfall beyond the frozen forecast interface.

A.4 Feasible Action and Execution Gate

Before actuation, the controller computes the refill upper bound

$$I_t^{\max} = \max\Big(0,\, FC - S_t - \widehat{P}_{\text{eff},t+1}\Big)\,, \tag{22}$$

which induces the legal action interval

$$\mathcal{A}(S_t) = [0, I_t^{\max}]. \tag{23}$$

Given a proposal $\widetilde{I}_t$, the final actuation gate applies

$$I_t = \Pi_{\mathcal{A}(S_t)}(\widetilde{I}_t) = \min\Big(\max(0, \widetilde{I}_t), I_t^{\max}\Big)\,. \tag{24}$$

This mechanism provides execution-feasibility and upper-bound protection. It does not guarantee avoidance of future crop-water deficit under arbitrary forecast error. Mass-balance checks and the device-level fallback/watchdog path remain external to the LLM and cannot be modified by persistent context.

B Dataset Provenance and Preprocessing

The combined corpus contains approximately 6,938 days, corresponding to roughly 19 crop-years. The environments span humid, semi-arid, and continental agricultural conditions and include multiple

crop systems. All input streams are converted to canonical units before entering the controller. Hourly weather and soil trajectories are retained for physical integration, whereas control decisions are issued at daily boundaries.

Missing values are handled before window construction. No imputation step may use observations that occur after the current decision time. Soil parameters are calibrated only on the calibration portion of each chronological sequence and are then frozen for held-out evaluation. Crop-stage and weather variables exposed to a controller are restricted to information available at the corresponding decision time.

The nine-season Colorado sunflower sequence is retained in chronological order for long-horizon analysis and is not pooled into the primary multi-site held-out aggregate.

C Evaluation Protocol and Leakage Controls

C.1 Chronological Split

For the main comparison, the first 60% of each eligible chronological segment is reserved for calibration or baseline training and the remaining 40% for evaluation. Time series are never shuffled. Each control decision uses a 168-hour history and a 24-hour control horizon. The physical process is integrated at hourly resolution, and the next controller invocation occurs after 24 hours.

The nine-season sequence is analyzed as a preserved chronological sequence.

C.2 Causal Information Policy

The evaluation protocol enforces the following leakage guards:

- no controller may access observations after the current decision time;
- persistent context may not be initialized or updated from future held-out outcomes;
- held-out results may not be used to select a favorable random seed;
- calibration/training and evaluation windows do not overlap.

The forecast interface is identical for all methods. In retrospective replay, all controllers receive the same causal precipitation input and the same precipitation-persistence construction where that forecast rule is used.

C.3 Matched Controller and Component Comparisons

For every controlled comparison, the initial state, weather sequence, observation history, physical action range, metric definition, and random seed are held fixed. Changing the controller does not change the environmental forcing or the evaluation rule.

Each ablation is interpreted only against the full configuration evaluated on the same matched slice. Several ablations were produced on different matched slices, so their absolute full-system values differ. Consequently, ablation effect sizes are interpreted within row and are not ranked across rows.

D Metric Definition

D.1 Seasonal Physical Control Cost

The frozen evaluator computes

$$C = w_W W + w_S S + w_R R + w_L L, \tag{25}$$

where

$$W = \sum_t I_t, \tag{26}$$

$$S = \sum_t \max(0, S_{\text{stress}} - S_t), \tag{27}$$

$$R = \sum_t R_t, \tag{28}$$

$$L = \sum_t L_t. \tag{29}$$

Here W is irrigation applied, S is accumulated storage deficit below the configured stress threshold, R is runoff, and L is deep percolation/leaching. All components are computed from the same hourly trajectory. The weights (w_W, w_S, w_R, w_L) are fixed globally across sites, controllers, and language-model backbones.

Because Eq. 25 is an absolute trajectory objective.

Frozen numerical weights. The exact numerical values of (w_W, w_S, w_R, w_L) are taken from the frozen experiment configuration associated with the reported results. They should be inserted here verbatim from that configuration in the final archival version.

D.2 Irrigation Applied

We separately report

$$W = \sum_t I_t \tag{30}$$

in millimeters. Lower irrigation is not interpreted as better control in isolation because a zero-action controller can minimize water use while accumulating substantial stress-related cost.

D.3 Aggregation and Uncertainty

Tables report mean and standard deviation over the repeated evaluation units defined by the corresponding frozen experiment configuration. The statistical unit and sample count n are retained with the result artifact for each table/figure and should be reported directly from that artifact in the final archival version. No claim of statistical significance is made without an explicit hypothesis test or confidence interval.

E Reference Controller Implementations

E.1 Historical Irrigation Replay

The historical reference replays recorded irrigation actions under the same evaluator used for algorithmic controllers. It is included as a practical management reference and is not interpreted as a comparison to realized grower yield.

E.2 Zero Irrigation and Fixed Threshold

A zero-action controller provides a lower-water diagnostic for identifying degenerate water-saving behavior. A fixed-threshold controller irrigates from the current soil-water state using a pre-specified threshold and has no learned or language-model state.

E.3 MAD-50

MAD-50 computes available water from field capacity and the configured lower agronomic threshold and triggers irrigation after 50% allowable depletion. When triggered, it refills toward field capacity subject to the shared actuator limit. It receives no privileged future information.

E.4 ET Replacement

ET-Replace accumulates evapotranspiration loss since the previous irrigation event and requests replacement of that loss, clipped to the same feasible action interval as the other controllers. Weather input and ET computation follow the same causal interface used by Mímir.

E.5 Deterministic Physics-Search/MPC Diagnostic

A deterministic physics-search controller enumerates a finite candidate action set under the same controller-side physical model and selects the candidate with the minimum rollout objective. We use this controller diagnostically to distinguish language-conditioned anchoring from numerical action search.

E.6 DRLIC and Learning References

The reproduced DRLIC reference uses the common state/action interface and the same physical action limits as Mímir in the reported comparison. Learning-based values are treated as comparable only when the trained policy is replayed through the common evaluator; externally reported rewards are not mixed with the manuscript's control-cost metric.

F Language-Model Interface

The main configuration uses a frozen DeepSeek-14B model through a local Ollama client. The prompt contains the structured physical state, actuator limits, forecast information, and active persistent principles and requests structured JSON output.

The primary decoding configuration uses greedy decoding with temperature zero where applicable, a fixed software seed, `think=False`, `format=json`, and `num_predict=2048`. No model-weight update, fine-tuning, or gradient step occurs during control or across seasons.

Only designated action fields are parsed from the model response. Malformed responses enter a deterministic fallback path and cannot bypass candidate selection, runtime assurance, or the final actuation gate. Per-call telemetry records success/error state, retry count, latency, prompt length, response length, and model identifier. Experiment-level metadata additionally record the software revision and data hash.

G Simulator-Grounded Decision Repair

G.1 Proposal–Verification–Revision Loop

At decision time, Mímir executes the bounded repair process

$$a_t^{(0)} = \mathrm{Propose}(x_t), \tag{31}$$

$$e_t^{(k)} = \mathrm{Verify}(s_t, a_t^{(k)}), \tag{32}$$

$$a_t^{(k+1)} = \mathrm{Revise}(x_t, a_t^{(k)}, e_t^{(k)}). \tag{33}$$

The loop terminates when physical verification no longer changes the proposed action or after the fixed three-round budget.

"Farmer" and "Physicist" are implementation role names. The former proposes an irrigation action and rationale; the latter invokes numerical tools and returns structured physical feedback. The methodological contribution is the separation of proposal generation from programmatic physical verification, not the role names themselves or free-form debate in isolation.

G.2 Bounded Candidate Selection

The deliberation output does not directly determine the executed action. Instead, it provides an anchor for a deterministic candidate selector.

Without a language-conditioned anchor, the selector constructs candidates over the legal action interval and retains physically meaningful references, including zero irrigation and refill toward field capacity. With an LLM/ACE anchor, the selector searches a bounded neighborhood around that anchor while retaining the same reference actions. A large raw LLM output cannot enlarge the legal action interval.

Each daily candidate is expanded over the 24-hour horizon, rolled out by the fixed controller-side model, and evaluated by the frozen rollout objective. Selection minimizes

$$(N_{\mathrm{viol}}(a), J(a)), \tag{34}$$

lexicographically, where N_{viol} is the modeled physical-violation count and J is the rollout cost. The selected action is subsequently passed to runtime assurance, fallback logic if required, and the final actuation gate.

The fast-loop execution sequence is therefore:

proposal → verification/revision → anchor → bounded search → physical rollout → selection → runtime assurance → actuat

G.3 Representative Trace

A representative trace illustrates the interaction:

> State: $S_t = 19.7\,\mathrm{mm}$, $WP = 10.5\,\mathrm{mm}$, $FC = 71.8\,\mathrm{mm}$, $ET_t = 3.8\,\mathrm{mm}$, $\widehat{P}_{\mathrm{eff},t+1} = 0\,\mathrm{mm}$.
> Proposal: $8.0\,\mathrm{mm}$.
> Verification: the simulated next state remains above the configured lower threshold; additional irrigation raises modeled drainage risk.
> Revision: $4.0\,\mathrm{mm}$.
> Verification: the zero-irrigation rollout remains feasible under the same short-horizon model.
> Final revision: $0.0\,\mathrm{mm}$.

This trace is illustrative only. Quantitative conclusions rely on matched component comparisons.

H Bounded Experience-Driven Adaptation

The slow loop modifies the contextual framework supplied to future language-model calls without changing model weights or the physical controller. Let τ_e denote the completed trajectory and outcome summary for episode/season e, m_e the persistent context, and q_e a configuration-controlled recurrence/evidence trigger. The update takes the form

$$m_{e+1} = \begin{cases} \mathcal{G}(m_e, \tau_e), & q_e = 1, \\ m_e, & q_e = 0. \end{cases} \tag{35}$$

The system is not forced to rewrite its context after every episode: if no recurring pattern satisfies the configured evidence condition, no contextual update occurs.

ACE implements $\mathcal{G}$ as a soft principle/anchor memory used by the reasoning layer. Deliberation traces, repeated verifier corrections, control-cost patterns, and runtime-assurance interventions can provide evidence for a candidate principle. A stored principle contains an identifier, source context, trigger, suggested action tendency, exception, supporting evidence, version, and activation state.

The adaptation pathway preserves the following invariants:

- no language-model weight update;
- no modification of physical parameters, transition equations, or evaluation rules;
- no relaxation of action or runtime-assurance bounds;
- no future held-out observations in persistent context;
- no bypass of candidate selection, runtime assurance, fallback, or the actuation gate.

Accordingly, "adaptation" denotes a change in future reasoning context. This is scaffold/framework-level.

Table 5: Aggregate results under the common multi-site evaluator. Irrigation is reported separately from control cost.

Controller	Control cost	Irrigation (mm)
Mímir	100.94 ± 1.80	2.58 ± 0.12
MAD-50	134.33 ± 3.10	10.58 ± 0.44
ET-Replace	440.74 ± 12.48	12.70 ± 0.31
DRLIC	924.47 ± 27.50	0.35 ± 0.00
Historical irrigation	790.51 ± 17.76	5.32 ± 0.57

Table 6: Paired component ablations. Each row uses its own matched evaluation slice; effect sizes must not be compared across rows. Water is in mm.

	Full		Ablated	
Removed component	**Cost**	**Water**	**Cost**	**Water**
Persistent context (ACE)	103.0	3.40	157.5	4.41
Decision repair	109.1	2.47	180.5	3.19
Execution shield	147.2	–	259.9	$\approx$ unchanged
Forward simulator	149.5	3.29	730.5	0.45

H.1 Stopping Conditions

At the fast timescale, proposal repair stops when physical verification no longer changes the proposal or the fixed three-round budget is reached. At the slow timescale, no persistent memory update occurs unless the recurrence/evidence trigger is satisfied.

Frozen adaptation configuration. Memory-update thresholds, recurrence rules, memory budget, and reset policy are fixed within each matched comparison and recorded in the corresponding frozen experiment configuration. Their exact values should be copied verbatim from the frozen configuration into the final archival supplement.

I Full Quantitative Results

The explicitly reported regional Mímir values are: Michigan, 152.25 ± 5.91 control cost and $1.54 \pm 0.38\,\mathrm{mm}$ irrigation; Texas, 129.16 ± 2.71 and $3.80 \pm 0.31\,\mathrm{mm}$; and Colorado, 59.82 ± 4.07 and $3.43 \pm 0.19\,\mathrm{mm}$.

Absolute control-cost magnitudes are not interpreted as directly comparable across crops or sites when the evaluated horizons and environmental forcings differ. The regional results are used to assess consistency across heterogeneous environments.

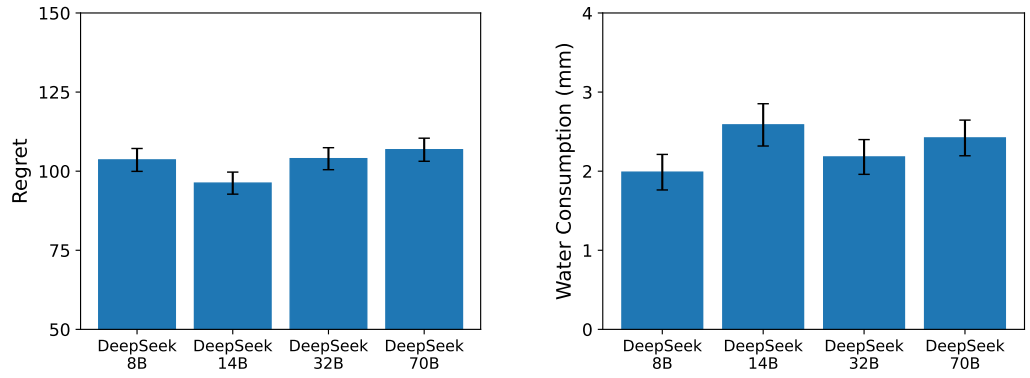
J Component Ablations

Direct-LLM diagnostic. An earlier zero-shot/direct-LLM run applied more than $20\,\mathrm{mm}$ under a metric configuration that predates the control-cost definition used for the matched ablations. Its scalar score is therefore not numerically comparable to Table 6 and is not used as quantitative evidence in the main paper.

K Foundation-Model Robustness

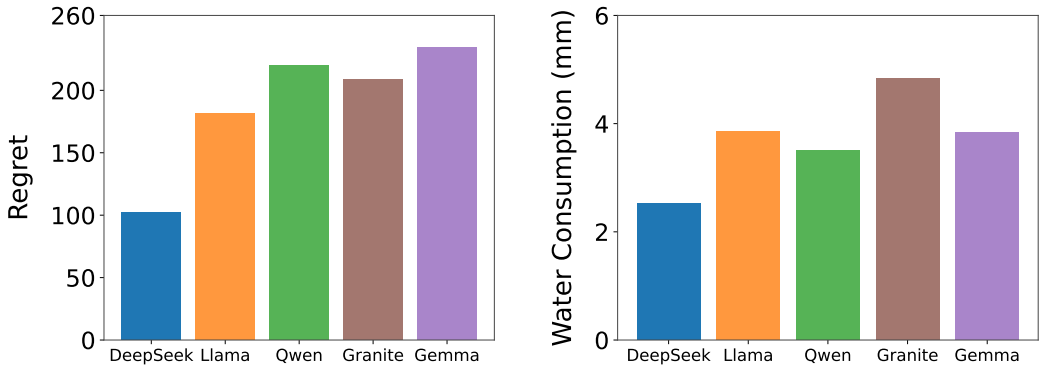
K.1 Model Scale

DeepSeek-14B records 96.23 ± 12 control cost with $2.585\,\mathrm{mm}$ irrigation in the reported sweep, whereas DeepSeek-8B records 103.54 ± 18 with lower irrigation. The 32B and 70B variants remain



(a) Control cost. (b) Irrigation applied.

Figure 4: DeepSeek scale sensitivity under an otherwise fixed Mímir pipeline.



(a) Control cost. (b) Irrigation applied.

Figure 5: Model-family sensitivity under a shared physical interface and evaluation sequence.

in a comparable range. We therefore report a non-monotonic relationship between parameter count and control performance within these tested checkpoints.

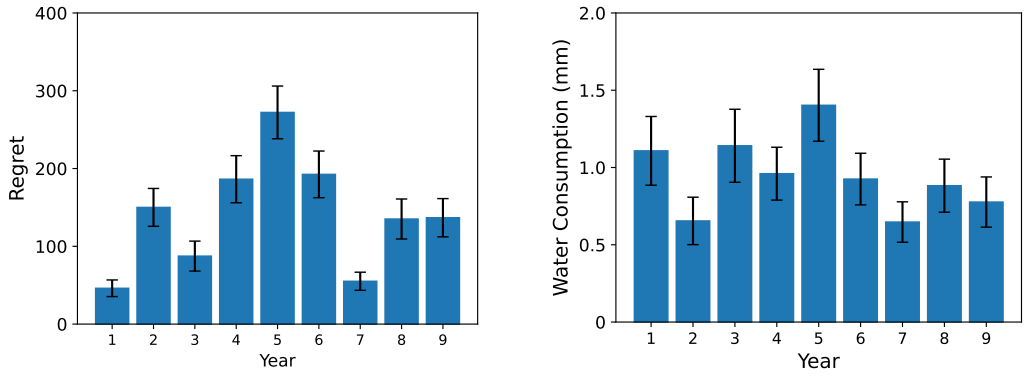
K.2 Model Family

The tested families are DeepSeek, Qwen, Llama, Gemma, and Granite. DeepSeek has the lowest explicitly reported value in this sweep (101.92 control cost, 4.26 mm irrigation), while the remaining families remain in the same broad numerical range. These results are interpreted as robustness/sensitivity within the tested set.

L Nine-Season Long-Horizon Analysis

The largest explicitly reported annual cost is 272.15 in Year 5. The nine-season sequence is used as an operational-stability analysis: control cost varies with seasonal forcing, but the sequence does not show sustained divergence after the high-cost season. Individual years are not assigned a causal weather label unless the corresponding weather evidence is reported alongside the control result.

This analysis is intentionally separated from the adaptation ablation. The sequence supports the claim that the complete system can continue operating over heterogeneous seasons without persistent divergence; it does not by itself demonstrate monotonic year-by-year learning. Evidence for the contribution of persistent context comes from the matched ACE ablation in Appendix J.



(a) Annual control cost.

(b) Annual irrigation applied.

Figure 6: Nine-season Colorado evaluation sequence.

M Reproducibility and Provenance

For each reported experiment, the evaluation pipeline records the dataset/configuration identifier, software revision, language-model checkpoint, random seed, and data hash. Non-LLM components are deterministic under a fixed environment and configuration. Language-model calls use the frozen decoding settings reported in Appendix F.

The controller operates once per day while the simulator integrates the physical state at hourly resolution. Reproduction should use the exact frozen configuration associated with each reported result, including metric weights, candidate-search span and step size, persistent-context thresholds and reset policy, seed set, and learning-baseline hyperparameters.

The source code is intended for public release upon publication. Experiment artifacts should be treated as the authoritative record for run-specific configuration values.

N Discussion and Limitations

What the LLM contributes. The LLM in Mímir is neither the physical model nor the final optimizer. It maps structured state, forecast, crop-stage information, and persistent guidance to a context-conditioned proposal and can revise that proposal after numerical feedback. Physical rollout evaluates consequences, deterministic selection ranks bounded candidates, and runtime assurance retains final execution authority. The paper therefore studies how adaptive semantic reasoning can be composed with trusted numerical control; it does not establish that an LLM is necessary for every irrigation problem or that it is superior to the best deterministic optimizer.

Bounded self-improvement and stopping conditions. Mímir implements scaffold-level, experience-driven adaptation. Persistent principles can change future proposals, but updates are evidence-triggered and cannot modify model weights, the physical transition model, the evaluator, or the feasible-action set. Fast-loop repair terminates when physical verification no longer changes the proposal or the fixed round budget is reached. Slow-loop adaptation is a no-op when no recurring pattern satisfies the configured update trigger. These semantics prevent open-ended reflection from becoming part of actuation.

We do not claim that persistent memory converges, improves monotonically, or is immune to forgetting. Mímir should therefore be viewed as a bounded step toward persistent or lifelong physical agency.

Implication for embodied physical agents. Irrigation is a persistent embodied-control problem: actions modify a physical state, feedback is delayed, environmental forcing changes over time, and experience accumulates across repeated interaction. The transferable design lesson is a separation of

authority across timescales: semantic reasoning may propose and adapt; numerical models predict consequences; deterministic logic constrains execution; and persistent context can repair future reasoning without rewriting the trusted physical substrate. Whether the same decomposition transfers to other long-horizon physical domains remains an empirical question for future work.

O AI USE DISCLOSURE

We used generative AI tools, including OpenAI ChatGPT, OpenAI Codex, and Google Gemini, during the development and preparation of this work.

These tools were used for four main purposes. First, they assisted with writing and editing, including improving the clarity, readability, organization, and presentation of author-provided technical content. Second, they supported literature retrieval and discovery by helping identify potentially relevant references and organize related-work discussions. Third, they assisted with research ideation and execution, including organizing research logic, translating author-defined objectives and conceptual designs into implementations, providing feedback on experimental design, and supporting code development and debugging. Fourth, they were used to help draft and revise portions of the manuscript based on research objectives, outlines, methods, experimental results, and technical content specified by the authors.

Generative AI was not used to fabricate experimental observations, numerical results, or evaluation outcomes. All empirical data reported in this work originate from the real-world measurements or research datasets described in the paper, and all reported quantitative results were obtained by executing the implemented experimental pipeline.

All AI-assisted code, experimental configurations, numerical results, citations, technical statements, and scientific claims were reviewed and verified by the authors. Suggested references were checked against their original sources, and AI-generated or AI-assisted code used in the reported experiments was inspected and tested before use.

Generative AI tools served only as research and writing assistants. The authors determined the research questions, methodological choices, experimental protocols, analyses, interpretations, claims, and conclusions. The authors take full responsibility for the final content of this work, including all text, code, results, citations, claims, and other artifacts produced with the assistance of generative AI.