Simulate, Reason, Decide: Scientific Reasoning with LLMs for Simulation-Driven Decision Making

Yuhan Yang
yuhanyyh@umich.edu
University of Michigan
Computer Science and Engineering
Ann Arbor, Michigan, USA

Ruipu Li
liruipu@umich.edu
University of Michigan
Computer Science and Engineering
Ann Arbor, Michigan, USA

Alexander Rodríguez
alrodri@umich.edu
University of Michigan
Computer Science and Engineering
Ann Arbor, Michigan, USA

Abstract

Scientific simulators are increasingly being integrated into LLM-driven systems for high-stakes simulation-driven decision-making. However, existing frameworks primarily use LLMs to generate, calibrate, or execute simulators, treating them as black-box interfaces rather than as structured mechanistic systems that can be reasoned about. As a result, current approaches lack the ability to identify, represent, and reason about the assumptions and mechanisms underlying simulator behavior, limiting transparency, auditability, and decision justification. We introduce MechSim, a mechanism-grounded neuro-symbolic reasoning framework for executable scientific simulators. Unlike prior neuro-symbolic approaches that primarily reason over static symbolic structures, MechSim enables LLM agents to reason about the mechanisms, assumptions, and execution behavior of scientific simulators. Our framework represents simulators through a shared structured schema capturing assumptions, variables, mechanism dependencies, and execution traces. On top of this representation, LLM agents operate as constrained reasoning engines that generate structured, evidence-grounded explanations linking simulator outcomes to their underlying mechanisms. We evaluate our approach across multiple high-stakes domains and show that it improves mechanism-level explanation quality, simulator analysis, and downstream decision-making reliability. The code is available at https://anonymous.4open.science/r/MechSim-2026/.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; *Decision Making*.

Keywords

large language models, scientific simulation, neuro-symbolic reasoning, decision making, explainability

ACM Reference Format:
Yuhan Yang, Ruipu Li, and Alexander Rodríguez. 2026. Simulate, Reason, Decide: Scientific Reasoning with LLMs for Simulation-Driven Decision Making. In *Proceedings of ACM KDD 2026 (KDD'26).* ACM, New York, NY, USA, 27 pages. https://doi.org/XXXXXXX.XXXXXXX

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD'26, Jeju, Korea
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2026/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

LLM agents provide a natural interface for automating decision-making workflows that ingest multimodal information from diverse sources and generate recommendations [40, 69]. More recently, LLM-driven systems have begun incorporating scientific simulators to support simulation-driven decision-making in high-stakes domains such as healthcare, finance, logistics, and public policy [13, 16, 34]. Scientific simulators are particularly valuable in these settings because they enable the evaluation of interventions and counterfactual scenarios when real-world experimentation is expensive, risky, or ethically infeasible [5, 10, 44].

Despite growing interest in LLM-integrated simulation systems, existing approaches primarily use LLMs to generate, calibrate, or execute simulation models [16, 28, 65]. However, they provide limited support for reasoning about the simulators themselves. In particular, current frameworks lack structured representations that allow LLM agents to analyze simulator assumptions, mechanism-level dependencies, and execution behavior in a grounded and verifiable manner. Consequently, these systems remain limited to descriptive summaries and surface-level interpretations of simulator outputs, often producing explanations that appear plausible but are not grounded in simulator structure, execution traces, or scientific evidence.

This limitation is particularly important because scientific simulators explicitly encode assumptions, variables, and mechanism-level dependencies that fundamentally shape their outputs. However, these assumptions are often embedded within specialized scientific formalisms, making it difficult to determine why particular outcomes arise and whether simulator behavior is aligned with the deployment context or evaluated policies. For example, during a public health crisis, an epidemic simulator may predict that school closures reduce peak infections, yet it may remain unclear whether this effect is primarily driven by reduced child-to-child transmission, mobility changes, or assumptions embedded in the contact-network structure. More importantly, the validity of these conclusions may depend on whether the simulator assumptions align with the real-world context, such as local social-contact patterns, behavioral responses, or intervention compliance. Without a framework capable of tracing outcomes back to simulator assumptions and mechanisms, simulation-driven recommendations become difficult to interpret, audit, and validate, all of which are critical for high-stakes decision-making [27, 52, 53].

To address these challenges, we propose MechSim, a neuro-symbolic framework for reasoning about scientific simulators in simulation-driven decision-making. MechSim combines structured simulator representations with LLM-based reasoning to enable

grounded analysis of simulator behavior and outcomes. Our main contributions are threefold.

(1) We introduce a **mechanism-grounded reasoning framework** that enables LLM agents to analyze how and why simulator outcomes arise, rather than operating solely on surface-level outputs. Our framework represents simulators through a shared structured schema capturing assumptions, variables, mechanism-level dependencies, and execution traces, enabling LLM agents to reason over simulator structure in a grounded and constrained manner.
(2) We introduce a **verifiable explanation framework** in which generated claims must be grounded in simulator structure, execution traces, or retrieved scientific evidence. This transforms explanation generation from post-hoc summarization into a process of constructing and validating mechanism-grounded hypotheses about simulator behavior.
(3) We conduct extensive experiments across three high-stakes domains, demonstrating that MechSim improves explanation quality, simulator analysis, and downstream decision-making reliability across multiple scenarios.

2 Problem Formulation

We study the problem of enabling LLM agents to guide simulation-driven decision-making. In this setting, LLM agents must analyze simulator behavior, assumptions, outputs, generate recommendations, and provide grounded justifications for real-world decision tasks. We consider a setting consisting of a decision task $\mathcal{T}$, a set (ensemble) of structurally heterogeneous scientific simulators $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, contextual information Ω describing the deployment environment, and a scientific corpus $\mathcal{D}$ containing scientific literature, technical reports, or policy guidelines.

The contextual information Ω captures properties of the real-world environment in which decisions are made, including population characteristics, institutional constraints, geographic conditions, time horizons, behavioral patterns, and cultural factors that influence whether simulator assumptions are valid for the target setting [63]. Importantly, simulators within $\mathcal{S}$ may differ substantially in their assumptions, internal mechanisms, and computational abstractions, potentially producing divergent outputs under the same deployment context [27]. As a result, simulator outputs alone are often insufficient for reliable decision-making, since recommendations may depend critically on latent simulator assumptions and their alignment with the deployment environment.

The task specification $\mathcal{T}$ defines the intended use of simulator outputs for decision-making. In this work, we focus on two primary scenarios. (1) **Policy Selection**: the agent evaluates candidate interventions based on simulator outputs, such as public-health interventions in epidemiology or inventory-management strategies in supply-chain systems. In this setting, $\mathcal{T}$ includes the set of candidate interventions to be evaluated. (2) **Simulator Selection for Forecasting**: prior to making predictions about future events, such as hospitalization forecasts or demand forecasts, the agent selects the most reliable simulator(s) from a set of available models.

Each simulator $s_i \in \mathcal{S}$ maps an initial condition x to an output y_i according to $y_i = s_i(x, \theta_i, f_i)$, where θ_i denotes simulator parameters and f_i denotes the simulator's computational logic. In many scientific simulators, x and θ_i are themselves uncertain and are therefore modeled as probability distributions rather than fixed values, an assumption we adopt in this work. The initial conditions x and parameters θ_i are instantiated from the contextual information Ω and the task specification $\mathcal{T}$. Our goal is therefore to enable LLM agents to generate recommendations and explanations that are grounded in simulator mechanisms, execution behavior, and contextual validity.

3 MechSim: Mechanism-Aware Reasoning for Scientific Simulators

MechSim is a neuro-symbolic framework for mechanism-grounded constrained reasoning over executable simulators. The key challenge in our problem setting is that simulator outputs alone do not expose which assumptions, mechanisms, or contextual factors drive the resulting predictions. MechSim addresses this challenge by explicitly representing simulator structure and constraining LLM reasoning using simulator mechanisms, scientific evidence, and execution behavior, defined as the evolution of simulator states and mechanism activations during simulation.

Our framework decomposes simulator reasoning into four tightly coupled components: (1) contextual grounding, (2) mechanism-level grounding, (3) constrained mechanism-level reasoning, and (4) verification checks. The symbolic layer of MechSim consists of structured simulator representations, including mechanism graphs, assumptions, execution traces, and sensitivity-analysis artifacts. The neural layer consists of LLM-based contextual interpretation, scientific evidence synthesis, explanation generation, and decision reasoning. Mechanism graphs serve as the symbolic substrate over which neural reasoning operates, enabling grounded reasoning over executable simulators rather than unconstrained reasoning over textual summaries alone.

Each component of MechSim is instantiated through specialized modules implemented as cooperative LLM agents, as illustrated in Figure 1. These agents are responsible for contextual understanding, simulator calibration and execution, mechanism modeling, constrained reasoning, sensitivity analysis, and verification. While the framework is conceptually organized around grounded neuro-symbolic reasoning, the agent decomposition enables modular execution and iterative refinement across stages.

Formally, given contextual information Ω, task specification $\mathcal{T}$, simulator ensemble $\mathcal{S}$, and scientific corpus $\mathcal{D}$, MechSim constructs structured simulator representations and generates a verified explanation E and recommendation R subject to structural consistency, evidence grounding, and empirical support constraints. It is noteworthy that unlike prior neuro-symbolic reasoning approaches that primarily operate over static symbolic structures, MechSim performs constrained reasoning over executable simulators whose behavior emerges dynamically through simulation and intervention.

3.1 Contextual Grounding

The contextual grounding component, implemented through a Context Understanding Agent, transforms heterogeneous contextual information into structured representations suitable for downstream reasoning. Given contextual information Ω, task specification $\mathcal{T}$,

and scientific corpus $\mathcal{D}$, MechSim extracts decision-relevant entities, constraints, interventions, and initial conditions required for simulator execution and reasoning. In parallel, the framework retrieves scientific evidence $\mathcal{Z}$ to ground downstream explanation generation and verification. Specifically, given a query q derived from Ω and $\mathcal{T}$, the retrieval function $\mathcal{K}(q, \mathcal{D}) \rightarrow \mathcal{Z}$ retrieves relevant scientific evidence using a dense retrieval and reranking pipeline inspired by OpenScholar [3]. The resulting evidence set $\mathcal{Z}$ provides scientific grounding for downstream reasoning and reduces unsupported or hallucinated explanations [30, 31].

3.2 Mechanism-Level Grounding

Mechanism-level grounding constructs structured representations of simulator behavior and execution dynamics. The central goal is to expose the mechanisms, assumptions, and execution patterns that produce simulator outcomes, enabling grounded reasoning over simulator behavior rather than reasoning solely over outputs.

Mechanism Graph Construction. For each simulator $s_i \in \mathcal{S}$, MechSim constructs a mechanism graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{M}_i)$, where $\mathcal{V}_i$ denotes simulator variables, $\mathcal{E}_i$ denotes mechanism-level transitions between variables, and $\mathcal{M}_i$ denotes graph metadata associated with the simulator. The metadata $\mathcal{M}_i$ includes simulator assumptions $\mathcal{A}_i$, execution traces τ_i, and summary statistics derived from simulator execution. The mechanism graph serves as the symbolic representation used to constrain downstream reasoning. Nodes correspond to simulator entities or state variables, while edges represent mechanistic transitions, interactions, or propagation pathways encoded by the simulator. Assumptions $\mathcal{A}_i$ determine which mechanisms are active under a given deployment context and intervention setting.

To construct $\mathcal{G}_i$, the Mechanism Modeling Agent extracts structured simulator representations from simulator definitions, execution traces, and domain documentation. Simulation outputs such as peak infection, mortality, or system overload are attached as graph-level metadata, enabling joint reasoning over simulator structure and observed outcomes.

Simulator Calibration and Execution. Given contextual information Ω and task specification $\mathcal{T}$, MechSim instantiates and executes each simulator in the ensemble $\mathcal{S}$. Initial conditions and intervention settings are derived from the deployment context, observational data, and task constraints.

Because simulator parameters are often uncertain, we model them as probability distributions rather than fixed values. We use approximate Bayesian computation (ABC) [57, 61] to estimate posterior parameter distributions (calibration) for each simulator, enabling uncertainty-aware execution and robust comparison across simulators. The Simulation Execution Agent runs each simulator and produces outputs $\mathcal{Y} = \{y_i\}_{i=1}^{n}$, along with execution traces (observable dynamical trajectories and intermediate states generated during simulation) and summary statistics used for downstream reasoning and verification. These artifacts provide empirical grounding for mechanism-level reasoning.

Sensitivity Analysis as Executable Verification. Sensitivity analysis operationalizes executable hypothesis testing over simulator mechanisms. Rather than treating explanations as purely linguistic artifacts, our Sensitivity Analysis Agent evaluates whether proposed explanations are empirically supported through controlled interventions on simulator parameters and assumptions [54].

For parametric sensitivity analysis, numerical parameters θ_i are perturbed to identify variables that most strongly influence simulator outputs. For structural sensitivity analysis, simulator assumptions $\mathcal{A}_i$ and mechanism components within $\mathcal{G}_i$ are modified to evaluate how changes in simulator structure alter observed outcomes. To quantify these effects, MechSim re-executes simulator S_i under alternative configurations of ϕ:

$$\text{Sensitivity}(\phi) = \mathbb{E}_{s_i(\phi=\phi_2)}[y] - \mathbb{E}_{s_i(\phi=\phi_1)}[y], \tag{1}$$

where $\phi \in \{\theta_i, \mathcal{A}_i\}$ and ϕ_1, ϕ_2 denote alternative parameter settings or assumptions. These interventions provide empirical evidence supporting, rejecting, or refining candidate explanations.

3.3 Constrained Mechanism-Level Reasoning

Our Mechanism Reasoning Agent performs constrained reasoning over mechanism graphs, simulator outputs, execution traces, sensitivity-analysis artifacts, and retrieved scientific evidence. Unlike unconstrained LLM reasoning, the framework restricts explanation generation to reasoning paths supported by simulator structure, execution behavior, and empirical evidence.

Given mechanism graphs $\mathcal{G}$, simulator outputs $\mathcal{Y}$, sensitivity-analysis results, and retrieved evidence $\mathcal{Z}$, we construct explanations by tracing how simulator assumptions activate mechanism-level propagation paths that produce decision-relevant outcomes. Rather than relying on surface-level metric comparison, the framework reasons over node-to-node transitions within mechanism graphs and explicitly links generated claims to simulator mechanisms, assumptions, evidence, and execution behavior. Importantly, when evidence is incomplete, conflicting, or insufficient to support a claim, the framework explicitly represents the corresponding uncertainty in its explanations.

Drawing on foundational work in explanation and argumentative reasoning (Toulmin model) [47, 62], MechSim constructs a structured explanation: $E = (\mathcal{I}, \mathcal{P}, \mathcal{Z}, \mathcal{C})$, where:

(1) **Output Interpretation ($\mathcal{I}$):** synthesizes the context, task, and simulator outputs, identifying decision-relevant output patterns for downstream reasoning and decision-making.
(2) **Reasoning Paths ($\mathcal{P}$):** $\mathcal{P} = \{p_1, p_2, \ldots, p_k\}$ denotes a set of mechanism propagation paths under a simulator-specific assumption set $\mathcal{A}_i$. Each path $p \mid \mathcal{A}_i = (v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} \ldots \xrightarrow{e_{n-1}} v_n)$ represents an ordered sequence of variables connected by mechanisms activated under $\mathcal{A}_i$, showing how assumptions and mechanisms lead to outcomes relevant to the task [41].
(3) **Supporting Evidence ($\mathcal{Z}$):** contains scientific evidence retrieved from the domain corpus $\mathcal{D}$. The evidence provides scientific grounding and is explicitly referenced to support the claims in C [70].
(4) **Claims (C):** $C = \{c_1, c_2, \ldots, c_m\}$ provides mechanism-grounded claims about the simulator results. Each claim integrates mechanism paths and external evidence to translate simulator outcomes into interpretable explanations that justify decisions and recommendations.

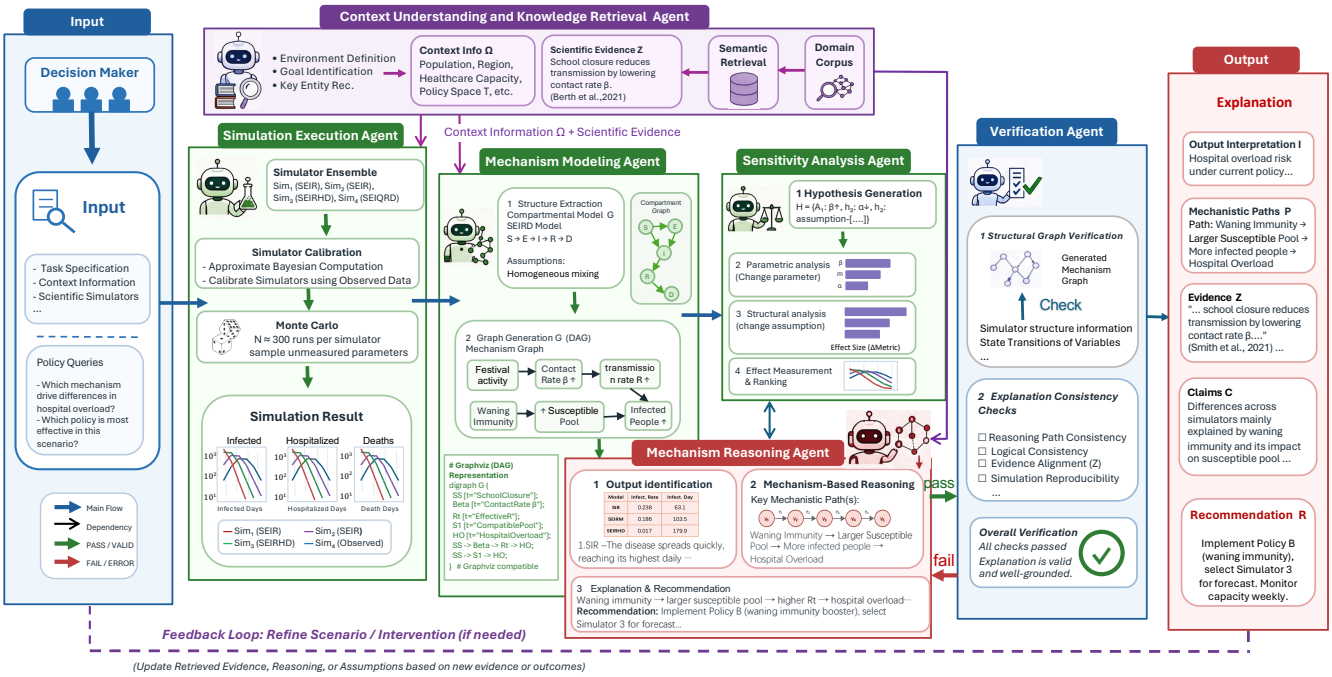

Figure 1: Overview of MechSim, a neuro-symbolic framework for mechanism-grounded reasoning over scientific simulators in simulation-driven decision-making. MechSim coordinates specialized LLM agents for contextual grounding, simulator execution, mechanism modeling, sensitivity analysis, constrained reasoning, and verification to generate empirically grounded explanations and decision recommendations from simulator outputs, mechanism graphs, and scientific evidence.

3.4 Verification Checks

The final stage is implemented by a Verification Agent that verifies that explanations and recommendations are faithful to simulator mechanisms, execution behavior, scientific evidence, and empirical validation results. The goal is to ensure that generated explanations are not merely linguistically plausible but are structurally and empirically grounded [55].

A valid explanation $E = (\mathcal{I}, \mathcal{P}, \mathcal{Z}, C)$ and recommendation R must satisfy three properties: (1) **Structural Consistency:** each claim $c \in C$ must be traceable to valid reasoning paths within mechanism graphs $\mathcal{G}_i$; (2) **Evidence Grounding:** generated claims must be logically consistent with retrieved scientific evidence $\mathcal{Z}$ and simulator outputs $\mathcal{Y}$; (3) **Empirical Support:** reasoning paths and claims must be supported by executable verification through sensitivity analysis when available.

If verification detects unsupported claims, invalid reasoning paths, or inconsistencies between explanations and simulator behavior, the framework initiates iterative refinement before producing final outputs. After verification, MechSim produces a verified explanation E and decision recommendation R. For policy selection, recommendations may identify interventions supported by simulator evidence and contextual assumptions. For simulator selection for forecasting, recommendations may identify which simulator is most reliable under the target deployment context. Importantly, recommendations are generated jointly with explanations and finalized only after verification.

4 Experiments and Evaluation

We evaluate MechSim across three high-stakes domains and three tasks: explanation generation, policy selection, and simulator selection for forecasting. We compare MechSim with representative LLM-based reasoning baselines. Additional experimental details are provided in the Appendix B. The code is available at https://anonymous.4open.science/r/MechSim-2026/.

4.1 Benchmark Environments

We use three simulation environments constructed from, or inspired by, real-world scenarios. Each environment includes an oracle simulator that generates ground-truth outcomes under different policies, along with a set of candidate simulators for modeling the dynamics. For each environment, we run 100 simulations by sampling the oracle simulator parameters and initial conditions from a predefined distribution and report the aggregated metrics.

(1) ***COVID-19***: We study COVID-19 transmission under alternative intervention policies using six compartmental epidemic simulators: SIR, SEIRD, SEIRHD, SEIQRD, SEAIHRD, and SIRV. This environment is motivated by COVID-19 scenario modeling, where compartmental simulators are calibrated to observed data and used

to project epidemic trajectories under different interventions [29]. We use a stochastic, multi-compartment, discrete-time simulator as the oracle simulator and ground truth [32]. For the forecast task, we use real-world COVID-19 data, including daily confirmed cases and deaths aggregated at the US state and national level [15].

(2) ***Supply Chain***: We analyze inventory management and demand fluctuations in multi-tier logistics systems using four candidate simulators: base-stock inventory control, (s, S) inventory control[1], decentralized multi-echelon (r, Q) control, and the Beer Game supply-chain model. The oracle simulator is a stochastic multi-echelon supply-chain simulator with intervention policies that alter managerial levers such as safety stock, lead times, demand sharing, order quantities, and expedited replenishment [59]. For the dataset, we use the M5 dataset, which includes hierarchical point-of-sale records at the item–store level in Walmart [42].

(3) ***Measles***: We study measles transmission and vaccination dynamics across six configured outbreak scenarios, including three source-informed configurations based on London, Chicago, and Texas–New Mexico outbreaks [1, 22, 45], and three synthetic configurations that simulate outbreaks in small populations such as schools. We use compartmental mechanistic baseline simulators implemented with Epydemix, including SIR, SEIR, SIRV, SVEIR, and SVVIR models [36]. The oracle simulator is a stochastic agent-based `measles::ModelMeaslesMixing` model accessed through `rpy2`/epiworldR [46, 64]. Although measles and COVID-19 are both respiratory viral infections, they differ substantially in transmission dynamics and intervention strategies [21]. As a result, their simulators rely on distinct assumptions and encode different mechanisms.

4.2 Baseline Methods

We compare MechSim with three baselines. For a fair comparison, all baselines reason over the same simulation outputs as MechSim and receive the same task information. (1) ***Causal-Copilot*** [67] is an autonomous agent that operationalizes expert-level causal analysis within an LLM framework. (2) ***Logic-LM*** [50] integrates symbolic reasoning with LLMs to support structured, logically grounded analysis. (3) ***Graph of Thoughts (GoT)*** [8] organizes inference as a graph, allowing the model to explore and aggregate multiple reasoning paths. These baselines were selected because they represent three distinct categories of reasoning methods and are both representative and reproducible. More information about baseline methods in available is Appendix B.1.

4.3 Tasks

Policy Selection Task: This task mimics the real-world decision-making process of selecting an optimal policy. For each simulation environment, we use a sophisticated, realistic simulator as the oracle model. The oracle model first generates ground-truth observations, then generates that under different policies, allowing us to evaluate the effect of policies and obtain the ground-truth policy ranking. In each experiment, we test 6–9 candidate policies, with specific details provided in the Appendix A.4. To test the robustness of our method, we sampled the parameters across 100 scenarios and recorded the average metrics. Each method then recommends a ranking of the candidate policies, and evaluation metrics are averaged over domain-specific sampled scenarios or stochastic replicates, as detailed in the Appendix B.

Evaluation Metrics. In the main results, we report Precision@3, Precision@5, Recall@3, and Recall@5 [37, 43]. These ranking metrics compare each method's top-K recommended policies with the oracle-preferred policy set: Precision@K measures the fraction of recommended top-K policies that are oracle-preferred, while Recall@K measures how much of the oracle-preferred set is recovered by the method's top-K recommendations. Higher precision and recall indicate stronger alignment with the oracle policy ranking.

Simulator Selection for Forecasting Task: This task requires selecting the most suitable simulator for forecasting under varying conditions. In practice, a decision-maker may have access to multiple calibrated simulators, each fit to real-world data but differing in its assumptions, mechanisms, and predictive behavior. During the pre-cutoff period, each simulator generates forecasts, and its predictive error is measured using Root Mean Square Error (RMSE). Each method then recommends a ranking of the candidate simulators given the pre-cutoff observations. The evaluation metrics are averaged over domain-specific forecast instances. Each domain evaluates simulator rankings on held-out forecasts: COVID-19 uses 4-week-ahead rolling forecasts over weekly death data across a 60-week held-out period, Supply Chain uses a 28-day demand forecast window, and Measles uses the first 14 days of incidence for fitting before forecasting the remaining held-out horizon (around 1 year).

Evaluation Metrics. We compute *regret* to evaluate the recommended simulators using ground-truth observations from the post-cutoff period. Regret is the gap between the RMSE of the selected simulator and that of the optimal simulator, i.e., the simulator with the lowest post-hoc RMSE [12]: $\text{Regret} = \text{RMSE}(\hat{s}) - \min_{s \in \mathcal{S}} \text{RMSE}(s)$, where $\hat{s}$ denotes the selected simulator and $\mathcal{S}$ is the set of candidate simulators. For more than one selected simulators, we compute the average of the regret. Lower regret indicates better performance.

Explanation Evaluation Task: This task evaluates whether a method can provide reliable explanations of simulator behavior in realistic decision-making scenarios. Given outputs from multiple simulators, each method explains the mechanisms that drive the observed outcomes. All evaluations are performed over five rounds with different random seeds, and the metrics are averaged.

Evaluation Metrics. To evaluate the explanations, we use multiple LLMs as independent evaluators and conduct multi-round scoring based on four criteria: completeness, mechanistic consistency, scientific soundness, and faithfulness, inspired by prior work [17, 38].

5 Main Results

We report the performance of our framework in three evaluation tasks across three benchmark domains: COVID-19, Supply Chain, and Measles.

5.1 Policy Selection Results

Table 1 evaluates policy selection as a ranked decision problem, using Precision@k and Recall@k to measure whether each method identifies the oracle-preferred policies. MechSim obtains the best result in almost all settings, with especially consistent gains on COVID-19 and Supply Chain. This pattern is stable across LLM

[1]Here, s refers to the reorder point in the inventory policy, which is different from the s used to denote simulators earlier.

Table 1: Policy selection results (Precision@k and Recall@k) across three domains and three LLM backbones. Bold = best per column within each domain. "Causal" denotes Causal-Copilot, "Logic" refers to Logic-LM, and "GoT" denotes Graph of Thoughts.

		Claude Sonnet 4.6				DeepSeek V3.2				Qwen 3-235B			
Domain	**Method**	P@3	P@5	R@3	R@5	P@3	P@5	R@3	R@5	P@3	P@5	R@3	R@5
COVID-19	Causal.	0.67	0.65	0.40	0.70	0.63	**0.72**	0.38	0.70	0.67	0.68	0.64	0.58
	Logic	0.63	0.62	0.38	0.65	0.60	0.61	0.36	0.64	0.70	0.60	0.42	0.67
	GoT	0.71	0.60	0.42	0.62	0.67	0.59	0.42	0.52	0.80	0.52	0.65	0.72
	MechSim	**0.82**	**0.72**	**0.54**	**0.72**	**0.83**	0.68	**0.50**	**0.72**	**0.83**	**0.70**	**0.76**	**0.77**
Supply Chain	Causal.	0.77	0.79	0.47	0.54	0.91	0.79	0.57	0.54	0.86	0.83	0.51	0.49
	Logic	0.82	0.83	0.49	0.35	0.87	0.85	0.52	0.65	0.87	**0.86**	0.52	0.49
	GoT	0.80	0.74	0.51	0.49	0.83	0.61	0.50	0.61	0.85	0.84	0.51	0.50
	MechSim	**0.90**	**0.84**	**0.54**	**0.57**	**0.97**	**0.86**	**0.58**	**0.66**	**0.91**	0.84	**0.55**	**0.52**
Measles	Causal.	0.56	0.73	0.36	0.74	0.61	0.77	0.37	0.77	0.67	0.72	0.40	0.79
	Logic	**0.72**	0.77	0.37	0.79	0.69	0.67	**0.43**	0.65	0.72	0.73	0.47	0.73
	GoT	0.33	0.63	0.29	0.66	0.58	0.43	0.43	0.65	0.33	0.65	**0.58**	0.53
	MechSim	0.61	**0.83**	**0.43**	**0.83**	**0.72**	**0.78**	0.38	**0.79**	**0.78**	**0.83**	0.43	**0.81**

Table 2: Simulator selection results for the forecasting task: Top-k regret ($k \in \{1, 3\}$; *lower is better*) across three domains and three LLM backbones. Bold denotes the best (lowest) result per row within each LLM block. "Causal" denotes Causal-Copilot, "Logic" refers to Logic-LM, and "GoT" denotes Graph of Thoughts.

		Claude Sonnet 4.6				DeepSeek V3.2				Qwen3-235B			
Domain	**Metric**	Causal	Logic	GoT	MechSim	Causal	Logic	GoT	MechSim	Causal	Logic	GoT	MechSim
COVID-19	Top-1 Regret	2.59	3.91	3.80	**2.04**	2.38	2.70	2.47	**2.16**	1.79	1.78	1.76	**1.65**
	Top-3 Regret	2.60	2.41	2.53	**2.29**	**2.10**	2.14	2.18	2.11	1.70	1.85	1.70	**1.68**
Supply Chain	Top-1 Regret	6.17	5.42	5.92	**4.25**	0.62	0.54	0.59	**0.47**	0.59	0.62	0.59	**0.24**
	Top-3 Regret	5.35	4.38	5.14	**3.47**	0.54	0.44	0.51	**0.39**	0.56	0.51	0.56	**0.51**
Measles	Top-1 Regret	2.88	2.35	1.93	**1.69**	**0.55**	5.94	2.49	0.70	1.87	2.69	3.04	**1.13**
	Top-3 Regret	3.40	3.21	2.79	**2.20**	2.99	3.26	2.76	**2.28**	2.55	3.17	2.70	**2.49**

backbones: MechSim is best in 11/12 Claude Sonnet 4.6 entries, 10/12 DeepSeek V3.2 entries, and 10/12 Qwen 3-235B entries. Compared with baselines emphasizing causal analysis [67], symbolic reasoning [50], and graph-structured deliberation [8], MechSim combines mechanism abstraction, empirical sensitivity analysis, retrieval, reasoning, and verification. The policy-selection results suggest a benefit from the framework's integration of these components.

On Measles, while MechSim excels in P@5 and R@5, Logic-LM or GoT is competitive on several P@3 or R@3 entries. This variability may reflect the greater heterogeneity of the Measles outbreak configurations and a smaller population size in these scenarios. These factors can make the oracle-preferred policy ranking less stable across top-k metrics. Full results with standard deviations are in Appendix C.7.

5.2 Simulator Selection for Forecasting Results

Table 2 evaluates simulator selection for forecasting using Top-k regret, where lower regret means that the recommended simulator is closer to the post-cutoff RMSE-optimal simulator. MechSim obtains the lowest or tied-lowest regret in almost all settings. The gains are especially pronounced in Supply Chain, where MechSim reduces regret relative to the best baseline by 21.6% for Claude Sonnet 4.6 Top-1, 20.8% for Claude Sonnet 4.6 Top-3, and 59.3% for Qwen 3-235B Top-1. COVID-19 shows smaller but consistent improvements for Top-1 regret across all three LLM backbones, consistent with the intended role of mechanism-grounded reasoning in identifying which simulator best captures the relevant dynamics. In the two settings where MechSim is not best, it remains close to the best score in absolute regret: it trails Causal-Copilot by 0.01 on COVID-19 DeepSeek V3.2 Top-3 regret and by 0.15 on Measles DeepSeek V3.2 Top-1 regret. In addition, similar to the policy selection results,

Table 3: Comparison of different LLMs and reasoning methods across domains. Metrics include Completeness (Comp.), Mechanism Logic Consistency (Logic.), Scientific Soundness (Sci.), and Faithfulness (Faith.). "Causal" denotes Causal-Copilot, "Logic" refers to Logic-LM, and "GoT" denotes Graph of Thoughts.

LLM	Method	COVID-19				Supply Chain				Measles			
		Comp.	Logic.	Sci.	Faith.	Comp.	Logic.	Sci.	Faith.	Comp.	Logic.	Sci.	Faith.
Claude Sonnet 4.6	Causal	2.2	2.6	2.4	2.8	2.2	2.2	2.2	2.8	2.8	3.2	2.2	2.4
	Logic	3.4	3.4	3.4	3.4	2.2	1.6	1.6	1.8	3.6	3.4	3.4	4.4
	GoT	2.4	2.4	2.4	2.6	3.0	3.2	3.2	3.2	2.6	2.6	2.6	2.8
	MechSim	**3.8**	**3.8**	**3.8**	**3.4**	**4.4**	**4.6**	**4.6**	**4.6**	**4.2**	**4.2**	**4.2**	**4.6**
DeepSeek V3.2	Causal	**4.4**	3.8	3.8	**4.6**	2.4	2.2	2.2	2.4	2.0	2.0	2.0	2.2
	Logic	1.8	2.0	2.0	1.8	1.8	3.6	2.4	2.6	3.4	3.0	3.0	3.2
	GoT	3.0	3.0	3.0	3.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.6
	MechSim	3.6	**4.4**	**4.2**	4.2	**4.6**	**4.4**	**4.4**	**4.6**	**4.0**	**4.0**	**4.0**	**4.0**
Qwen 3-235B	Causal	2.8	3.0	2.8	3.0	3.4	2.2	2.4	3.6	2.6	2.2	2.2	2.6
	Logic	**3.6**	3.0	3.0	**3.8**	2.4	2.4	3.4	2.2	3.6	4.6	4.6	**4.8**
	GoT	1.8	2.0	2.0	2.4	3.2	3.4	3.4	3.2	2.4	3.2	2.6	3.4
	MechSim	2.8	**3.2**	**3.2**	3.2	**3.8**	**3.8**	**3.8**	**3.8**	**4.0**	**4.7**	**4.8**	4.4

MechSim yields consistent gains across LLM backbones, demonstrating that the framework's benefits are not specific to a particular LLM. Full results with standard deviations are in Appendix C.8.

5.3 Explanation Evaluation Results

Table 3 evaluates explanation quality using a 1–5 judge score across four dimensions: completeness, mechanism-logic consistency, scientific soundness, and faithfulness. MechSim obtains the best score in 31/36 cells. The gains are most consistent in Supply Chain, where MechSim ranks first on every criterion for all three LLM backbones. In COVID-19, MechSim remains competitive but is usually second on completeness and faithfulness.

5.4 Ablation Study Results

The ablation study (full results in Appendix C.6) shows that removing mechanism reasoning causes the largest drop (50%), while removing context understanding or knowledge retrieval leads to smaller decreases (18%), indicating all components contribute to high-quality explanations, with mechanism reasoning being central, especially for weaker LLMs.

6 Related Work

LLMs for Scientific Simulation and Analysis. Recent studies increasingly apply LLMs to scientific simulation and analysis [4, 39]. GenSim exploits LLMs' grounding and code-generation abilities to construct simulation environments and expert demonstrations [66]. G-Sim presents a hybrid approach that integrates LLM-driven structural design with empirical calibration for simulator development [28]. However, there is a notable gap in utilizing LLMs as comparative analysts that can synthesize domain knowledge to diagnose logic inconsistencies across multiple scientific simulators [71].

Agentic Scientific Modeling Systems. Scientific modeling is increasingly driven by autonomous agentic systems that automate the simulation and experimentation lifecycle. For instance, Curie establishes a rigorous framework to manage scientific hypotheses and experimental protocols systematically [35]. In public health, agentic systems have been successfully deployed to automate epidemiological model construction [16] and act as autonomous policymakers to optimize intervention strategies [2]. The AI Scientist introduces an end-to-end pipeline capable of independent idea generation, experimental execution, and automated manuscript preparation [39].

7 Conclusion

This paper introduces MechSim, a neuro-symbolic framework for mechanism-grounded reasoning over executable simulators in simulation-driven decision-making. MechSim combines structured simulator representations, scientific evidence retrieval, sensitivity analysis, and verification-oriented reasoning to generate empirically grounded explanations and recommendations. While MechSim demonstrates strong performance in our benchmarks, extending the framework to increasingly complex scientific domains presents opportunities for future research. Domains such as climate modeling, financial systems, and large-scale infrastructure or biological simulations involve highly interconnected dynamics, substantial uncertainty, and large mechanism spaces that require more scalable

reasoning and verification capabilities. Future work could therefore focus on improving MechSim's ability to reason over large-scale dynamical systems, strengthen uncertainty-aware mechanistic inference, and enhance computational efficiency for real-time decision-support applications.

Potential Negative Impacts. Although our work aims to improve transparency and grounded reasoning in simulation-driven decision-making, incorrect assumptions, biased observational data, or flawed evidence may still lead to misleading explanations or harmful recommendations. Because the framework may be used in high-stakes domains, overreliance on automatically generated explanations could create risks if users interpret them as guarantees rather than uncertainty-aware decision-support tools. MechSim should therefore be deployed with human oversight, domain-expert validation, and clear communication of uncertainty and system limitations.

References

[1] Clifford Kwei-Ann Allotey. 2017. *A comparison of existing measles models.* Master's thesis. University of Manitoba, Winnipeg, Canada. https://mspace.lib.umanitoba.ca/handle/1993/32135

[2] Goshi Aoki and Navid Ghaffarzadegan. 2026. AI Agents as Policymakers in Simulated Epidemics. *arXiv preprint arXiv:2601.04245* (2026).

[3] Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'Arcy, et al. 2026. Synthesizing scientific literature with retrieval-augmented language models. *Nature* (2026), 1–7.

[4] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2025. Researchagent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 6709–6738.

[5] Jerry Banks, John S. Carson, Barry L. Nelson, and David M. Nicol. 2010. *Discrete-Event System Simulation* (5th ed.). Prentice Hall.

[6] Chris T Bauch and David JD Earn. 2004. Vaccination and the theory of games. *Proceedings of the National Academy of Sciences* 101, 36 (2004), 13391–13394.

[7] Mark A Beaumont, Wenyang Zhang, and David J Balding. 2002. Approximate Bayesian computation in population genetics. *Genetics* 162, 4 (2002), 2025–2035.

[8] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 17682–17690.

[9] Jan M Brauner, Sören Mindermann, Mrinank Sharma, David Johnston, John Salvatier, Tomáš Gavenčiak, Anna B Stephenson, Gavin Leech, George Altman, Vladimir Mikulik, et al. 2021. Inferring the effectiveness of government interventions against COVID-19. *Science* 371, 6531 (2021), eabd9338.

[10] John S Carson. 2005. Introduction to modeling and simulation. In *Proceedings of the Winter Simulation Conference, 2005*. IEEE, 8–pp.

[11] Centers for Disease Control and Prevention. 2024. CDC COVID-19 Travel-Associated Infections and Diseases. https://www.cdc.gov/yellow-book/hcp/travel-associated-infections-diseases/covid-19.html. Accessed: 2026-05-06.

[12] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, learning, and games*. Cambridge university press.

[13] Taha Choukhmane, Tim de Silva, Weidong Lin, and Matthew Akuzawa. 2026. AI Financial Advice: Supply, Demand, and Life Cycle Implications. *Demand, and Life Cycle Implications (March 19, 2026)* (2026).

[14] Yunfei Chu, Fengqi You, John M Wassick, and Anshul Agarwal. 2015. Simulation-based optimization framework for multi-echelon inventory systems under uncertainty. *Computers & Chemical Engineering* 73 (2015), 1–16.

[15] Estee Y Cramer, Yuxin Huang, Yijin Wang, Evan L Ray, Matthew Cornell, Johannes Bracher, Andrea Brennen, Alvaro J Castro Rivadeneira, Aaron Gerding, Katie House, et al. 2022. The United States COVID-19 forecast hub dataset. *Scientific data* 9, 1 (2022), 462.

[16] Rituparna Datta, Zihan Guan, Baltazar Espinoza, Yiqi Su, Priya Pitre, Srini Venkatramanan, Naren Ramakrishnan, and Anil Vullikanti. 2026. Agentic Framework for Epidemiological Modeling. *arXiv preprint arXiv:2602.00299* (2026).

[17] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 4443–4458.

[18] Odo Diekmann, Hans Heesterbeek, and Tom Britton. 2013. *Mathematical tools for understanding infectious disease dynamics*. Vol. 7. Princeton University Press.

[19] Ensheng Dong, Hongru Du, and Lauren Gardner. 2020. An interactive web-based dashboard to track COVID-19 in real time. *The Lancet infectious diseases* 20, 5 (2020), 533–534.

[20] Neil M Ferguson, Daniel Laydon, Gemma Nedjati-Gilani, Natsuko Imai, Kylie Ainslie, Marc Baguelin, Sangeeta Bhatia, Adhiratha Boonyasiri, Zulma Cucunubá, Gina Cuomo-Dannenburg, et al. 2020. *Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand*. Vol. 16. Imperial College London London.

[21] Katy AM Gaythorpe, Kaja Abbas, John Huber, Andromachi Karachaliou, Niket Thakkar, Kim Woodruff, Xiang Li, Susy Echeverria-Londono, Matthew Ferrari, et al. 2021. Impact of COVID-19-related disruptions to measles, meningococcal A, and yellow fever vaccination in 10 countries. *Elife* 10 (2021), e67023.

[22] Gilberto González-Parra, Annika Vestrand, and Remy Mujynya. 2025. Modeling and Characterizing the Growth of the Texas–New Mexico Measles Outbreak of 2025. *Epidemiologia* 6, 4 (2025), 60.

[23] Nicolò Gozzi, Matteo Chinazzi, Jessica T Davis, Corrado Gioannini, Luca Rossi, Marco Ajelli, Nicola Perra, and Alessandro Vespignani. 2025. Epydemix: An open-source Python package for epidemic modeling with integrated approximate Bayesian calibration. *PLOS Computational Biology* 21, 11 (2025), e1013735.

[24] Bryan T Grenfell, Ottar N Bjørnstad, and Jens Kappey. 2001. Travelling waves and spatial hierarchies in measles epidemics. *Nature* 414, 6865 (2001), 716–723.

[25] Xi He, Eric HY Lau, Peng Wu, Xilong Deng, Jian Wang, Xinxin Hao, Yiu Chung Lau, Jessica Y Wong, Yujuan Guan, Xinghua Tan, et al. 2020. Temporal dynamics in viral shedding and transmissibility of COVID-19. *Nature medicine* 26, 5 (2020), 672–675.

[26] Herbert W Hethcote. 2000. The mathematics of infectious diseases. *SIAM review* 42, 4 (2000), 599–653.

[27] Inga Holmdahl and Caroline Buckee. 2020. Wrong but useful—what covid-19 epidemiologic models can and cannot tell us. *New England Journal of Medicine* 383, 4 (2020), 303–305.

[28] Samuel Holt, Max Ruiz Luyten, et al. 2025. G-Sim: Generative Simulations with Large Language Models and Gradient-Free Calibration. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

[29] Emily Howerton, Lucie Contamin, Luke C Mullany, Michelle Qin, Nicholas G Reich, Samantha Bents, Rebecca K Borchering, Sung-mok Jung, Sara L Loo, Claire P Smith, et al. 2023. Evaluation of the US COVID-19 Scenario Modeling Hub for informing pandemic response under uncertainty. *Nature communications* 14, 1 (2023), 7260.

[30] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43, 2 (2025), 1–55.

[31] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys* 55, 12 (2023), 1–38.

[32] Morgan P Kain, Marissa L Childs, Alexander D Becker, and Erin A Mordecai. 2021. Chopping the tail: how preventing superspreading can help to maintain COVID-19 control. *Epidemics* 34 (2021), 100430.

[33] William Ogilvy Kermack and Anderson G McKendrick. 1927. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character* 115, 772 (1927), 700–721.

[34] Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W Park. 2024. MDAgents: An adaptive collaboration of LLMs for medical decision-making. *Advances in Neural Information Processing Systems* 37 (2024), 79410–79452.

[35] Patrick Tser Jern Kon, Jiachen Liu, Qiuyi Ding, Yiming Qiu, Zhenning Yang, Yibo Huang, Jayanth Srinivasa, Myungjin Lee, Mosharaf Chowdhury, and Ang Chen. 2025. Curie: Toward rigorous and automated scientific experimentation with ai agents. *arXiv preprint arXiv:2502.16069* (2025).

[36] Md Abdul Kuddus, M Mohiuddin, and Azizur Rahman. 2021. Mathematical analysis of a measles transmission dynamics model in Bangladesh with double dose vaccination. *Scientific reports* 11, 1 (2021), 16571.

[37] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[38] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*. 2511–2522.

[39] Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. 2026. Towards end-to-end automation of AI research. *Nature* 651, 8107 (2026), 914–919.

[40] Tianmi Ma, Jiawei Du, Wenxin Huang, Wenjie Wang, Liang Xie, Xian Zhong, and Joey Tianyi Zhou. 2025. Agent Trading Arena: A Study on Numerical Understanding in LLM-Based Agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*. 5496–5514.

[41] Peter Machamer, Lindley Darden, and Carl F Craver. 2000. Thinking about mechanisms. *Philosophy of science* 67, 1 (2000), 1–25.

[42] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2022. M5 accuracy competition: Results, findings, and conclusions. *International journal of forecasting* 38, 4 (2022), 1346–1364.

[43] Christopher D Manning. 2008. *Introduction to information retrieval*. Syngress Publishing,.

[44] Madhav Marathe and Anil Kumar S Vullikanti. 2013. Computational epidemiology. *Commun. ACM* 56, 7 (2013), 88–96.

[45] Nina B Masters. 2024. Real-time use of a dynamic model to measure the impact of public health interventions on measles outbreak size and duration—Chicago, Illinois, 2024. *MMWR. Morbidity and Mortality Weekly Report* 73 (2024).

[46] Derek Meyer and George Vega Yon. 2023. epiworldR: Fast Agent-Based Epi Models. *The Journal of Open Source Software* 8, 90 (oct 2023). doi:10.21105/joss.05781

[47] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.

[48] Seyed M Moghadas, Affan Shoukat, Meagan C Fitzpatrick, Chad R Wells, Pratha Sah, Abhishek Pandey, Jeffrey D Sachs, Zheng Wang, Lauren A Meyers, Burton H Singer, et al. 2020. Projecting hospital utilization during the COVID-19 outbreaks in the United States. *Proceedings of the National Academy of Sciences* 117, 16 (2020), 9122–9126.

[49] Sam Moore, Edward M Hill, Michael J Tildesley, Louise Dyson, and Matt J Keeling. 2021. Vaccination and non-pharmaceutical interventions for COVID-19: a mathematical modelling study. *The lancet infectious diseases* 21, 6 (2021), 793–802.

[50] Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 3806–3824.

[51] Evan L Porteus. 2002. *Foundations of stochastic inventory theory*. Stanford University Press.

[52] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence* 1, 5 (2019), 206–215.

[53] Andrea Saltelli, Gabriele Bammer, Isabelle Bruno, Erica Charters, Monica Di Fiore, Emmanuel Didier, Wendy Nelson Espeland, John Kay, Samuele Lo Piano, Deborah Mayo, et al. 2020. Five ways to ensure that models serve society: a manifesto. *Nature* 582, 7813 (2020), 482–484.

[54] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. 2008. *Global sensitivity analysis: the primer*. John Wiley & Sons.

[55] Robert G Sargent. 2010. Verification and validation of simulation models. In *Proceedings of the 2010 winter simulation conference*. IEEE, 166–183.

[56] Herbert Scarf. 1960. The optimality of (S, s) policies in the dynamic inventory problem. In *Mathematical Methods in the Social Sciences*, Kenneth J. Arrow, Samuel Karlin, and Patrick Suppes (Eds.). Stanford University Press, Stanford, CA, 196–202.

[57] Scott A Sisson, Yanan Fan, and Mark Beaumont. 2018. *Handbook of approximate Bayesian computation*. CRC press.

[58] John D Sterman. 1989. Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. *Management science* 35, 3 (1989), 321–339.

[59] John D. Sterman. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. McGraw-Hill.

[60] Biao Tang, Xia Wang, Qian Li, Nicola Luigi Bragazzi, Sanyi Tang, Yanni Xiao, and Jianhong Wu. 2020. Estimation of the transmission risk of the 2019-nCoV and its implication for public health interventions. *Journal of clinical medicine* 9, 2 (2020), 462.

[61] Dhorasso Temfack and Jason Wyse. 2025. Sequential Monte Carlo Squared for online inference in stochastic epidemic models. *Epidemics* (2025), 100847.

[62] Stephen E Toulmin. 2003. *The uses of argument*. Cambridge university press.

[63] Adelinde M Uhrmacher, Peter Frazier, Reiner Hähnle, Franziska Klügl, Fabian Lorig, Bertram Ludäscher, Laura Nenzi, Cristina Ruiz-Martin, Bernhard Rumpe, Claudia Szabo, et al. 2024. Context, composition, automation, and communication: The C2AC roadmap for modeling and simulation. *ACM Transactions on Modeling and Computer Simulation* 34, 4 (2024), 1–51.

[64] George Vega Yon. 2026. *measles: Measles Epidemiological Models*. https://github.com/UofUEpiBio/measles R package version 0.3.1-0.

[65] Stefan Wahl, Raphaela Schenk, Ali Farnoud, Jakob H Macke, and Daniel Gedon. 2026. A Probabilistic Framework for LLM-Based Model Discovery. *arXiv preprint arXiv:2602.18266* (2026).

[66] Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. 2024. GenSim: Generating Robotic Simulation Tasks via Large Language Models. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=OI3RoHoWAN

[67] Xinyue Wang, Kun Zhou, Wenyi Wu, Har Simrat Singh, Fang Nan, Songyao Jin, Aryan Philip, Saloni Patnaik, Hou Zhu, Shivam Singh, et al. 2025. Causal-copilot: An autonomous causal analysis agent. *arXiv preprint arXiv:2504.13263* (2025).

[68] World Health Organization. 2026. WHO COVID-19 Dashboard. https://data.who.int/dashboards/covid19. Accessed: 2026-05-06.

[69] Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. 2024. TradingAgents: Multi-agents llm financial trading framework. *arXiv preprint arXiv:2412.20138* (2024).

[70] Haozhou Xu, Dongxia Wu, Matteo Chinazzi, Ruijia Niu, Rose Yu, and Yi-An Ma. 2025. SimulRAG: Simulator-based RAG for Grounding LLMs in Long-form Scientific QA. *arXiv preprint arXiv:2509.25459* (2025).

[71] Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. 2023. Causal Parrots: Large Language Models May Talk Causality But Are Not Causal. *arXiv preprint arXiv:2308.13067* (2023).

Appendix

Contents

Abstract		1
1	Introduction	1
2	Problem Formulation	2
3	MechSim: Mechanism-Aware Reasoning for Scientific Simulators	2
3.1	Contextual Grounding	2
3.2	Mechanism-Level Grounding	3
3.3	Constrained Mechanism-Level Reasoning	3
3.4	Verification Checks	4
4	Experiments and Evaluation	4
4.1	Benchmark Environments	4
4.2	Baseline Methods	5
4.3	Tasks	5
5	Main Results	5
5.1	Policy Selection Results	5
5.2	Simulator Selection for Forecasting Results	6
5.3	Explanation Evaluation Results	7
5.4	Ablation Study Results	7
6	Related Work	7
7	Conclusion	7
References		9
Contents		11
A	Dataset and Environment Details	11
A.1	Datasets	11
A.2	Simulator List	11
A.3	Oracle Models	11
A.4	Intervention Policy Lists	13
B	Experiment Details	14
B.1	Baseline Method Details	14
B.2	Calibration Methods	15
B.3	Parameter Settings	15
B.4	Prompt Templates	15
C	Additional Experimental Results	21
C.1	Example Context Information	21
C.2	Example Explanation Output	21
C.3	Example Mechanism Graph	22
C.4	Simulation Result	22
C.5	Sensitivity Analysis Results	23
C.6	Ablation Study Results	23
C.7	Comprehensive Policy Selection Task Result	23
C.8	Comprehensive Simulator Selection Task Result	25

A Dataset and Environment Details

A.1 Datasets

We evaluate MechSim across three domains. Table 1 summarises the data sources and their key characteristics.

COVID-19 data processing. we use real-world COVID-19 data from COVID-19 forecast hub, including daily confirmed cases and deaths aggregated at the US state and national level [15]. For the forecast task, the pre-cutoff period is used to calibrate each simulator via particle filtering [61], and the post-cutoff period serves as the held-out evaluation window. We sampled model parameters with random seed and conducted experiments across 50 different locations.

Supply chain data processing. The M5 dataset provides hierarchical point-of-sale records at the item–store level for Walmart. We aggregate daily sales to the product-category and state levels for the supply-chain experiments.

Measles data processing. Each scenario is parameterised by a dedicated YAML configuration file specifying population size N, population structure and immunity/vaccination inputs, contact-matrix structure (flat homogeneous mixing or a Short Creek–calibrated age-structured matrix), and school-term forcing. The six scenarios are summarised in Table 2. Oracle traces were loaded as pre-computed aggregate outputs, with one trace for each (config, policy) pair. Incidence was derived from the daily change in susceptibles ΔS_t [24].

A.2 Simulator List

A.2.1 COVID-19 Simulators. We employ six ODE-based compartmental simulators for the COVID-19 domain, listed in Table 3. The simulator dynamics were integrated numerically for a 180-day period to produce trajectories. For the forecast task, the pre-cutoff period is used to calibrate each simulator via particle filtering [61], and the post-cutoff period serves as the held-out

A.2.2 Supply Chain Simulators. Four supply chain simulators are used in the supply chain domain (Table 4). All operate in discrete time with Poisson external demand. Costs include per-unit holding cost c_h and per-unit backlog cost c_b.

A.2.3 Measles Simulators. Five compartmental baseline models are used as candidate simulators for the measles environment (Table 5), implemented via the epydemix platform [23] and calibrated with Approximate Bayesian Computation (ABC) [7].

A.3 Oracle Models

A.3.1 COVID-19 Oracle. The COVID-19 Oracle is a stochastic, multi-compartment discrete-time simulator [32] designed to capture epidemic dynamics beyond the representational capacity of the simple ODE models evaluated by MechSim. It serves as the ground-truth reference for the policy selection task.

Compartment structure. S (susceptible), E (latent), I_a (asymptomatic), I_p (pre-symptomatic), I_m (mild symptomatic), I_s (severe symptomatic), H_r (hospitalised, recovering), H_d (hospitalised, dying), R (recovered), D (dead). Daily incident counters D_{new}, H_{new}, $I_{\text{new,sympt}}$ reset each period.

Transition dynamics. All transitions are stochastic, drawn from Binomial/Multinomial distributions using an Euler-Multinomial

Table 1: Summary of datasets used across the three experimental domains.

Domain	Dataset	Description	Reference
COVID-19	JHU CSSE COVID-19 Time Series	Daily confirmed cases and deaths aggregated at the US county and national level, covering the period from January 2020 onward.	[19]
Supply Chain	M5 Forecasting – Accuracy	Hierarchical daily unit-sales records for 3,049 Walmart products across ten stores in three US states (CA, TX, WI), spanning 1,941 days (2011–2016). Features include calendar events and sell prices.	[42]
Measles	Measles Scenario Ensemble	We conducted our experiment across six configured scenarios (default synthetic, perfect-mixing with and without school-term forcing, London pre-vaccine era, Chicago 2024 migrant shelter, and Texas–New Mexico 2025 outbreak). Population sizes range from 1,877 to 2,445,368; simulation horizons from 35 to 365 days.	[1, 22, 45]

Table 2: Measles scenario configurations used in the scenario ensemble.

Scenario	N	R_0	Horizon (d)	Notes
Default synthetic	5,630	12	365	Short Creek–calibrated age-structured matrix
Perfect mixing	5,630	15	365	Flat 1×1 matrix; no term forcing
Perfect mixing + school term	5,630	15	365	Flat matrix; spring/summer/winter closures
London pre-vaccine (1944–64)	2,445,368	28	365	Natural-immunity proxy; UK term forcing
Chicago 2024 shelter	1,877	25	365	5-group demography; flat contact mixing
Texas–NM 2025	128,924	30	35	5-county region represented as one group

Table 3: COVID-19 simulators used in the evaluate and forecast tasks.

Model	Compartments	Key Mechanisms & Assumptions	Reference
SIR	S, I, R	Homogeneous mixing; constant β and γ; permanent immunity; no latency, mortality, or hospitalisation.	[26, 33]
SEIRD	S, E, I, R, D	Exposed latency ($\sigma = 1/\text{incubation}$) and disease-induced mortality (μ); homogeneous mixing.	[26]
SEIR-HD	S, E, I, R, H, D	Explicit hospitalisation compartment (H) with capacity threshold; separates community mortality (μ_I) from hospital CFR (δ); enables overload tracking.	[20, 48]
SEIQRD	S, E, I, Q, R, D	Explicit quarantine compartment (Q); quarantine rate q directly reduces effective transmission.	[60]
SEAIHRD	S, E, A, I, H, R, D	Asymptomatic transmission (A) with relative transmissibility $\rho \approx 0.5$; fraction $f \approx 0.3$ of exposed become asymptomatic.	[25]
SIRV	S, I, R, V, D	Vaccination compartment (V) with daily coverage rate ν; vaccinated individuals fully protected.	[18]

scheme with sub-day step $\Delta t = 1/6$:

$$\Delta S \to E \sim \text{Binomial}\Big(S,\ 1 - e^{-\beta_t \Lambda \Delta t}\Big),$$
$$\Lambda = \frac{C_a I_a + C_p I_p + \text{iso}_m C_m I_m + \text{iso}_s C_s I_s}{N}. \quad (2)$$

Key parameters. Fraction asymptomatic $\alpha = 0.30$; relative contact weights C_a=0.50, C_p=0.80, C_m=0.60, C_s=1.00; latent period $T_{\text{lat}} = 4$ days; probability of mild given symptomatic $\mu = 0.50$; hospitalisation fraction from severe $\delta = 0.12$; hospital CFR $\rho_d^{-1} = 9$ days; hospital recovery $\rho_r^{-1} = 14$ days.

Harm function.

$$\text{Harm} = C_{\text{death}} \cdot D_{\text{total}} + C_{\text{overload}} \cdot H_{\text{overload}} + \lambda_{\text{policy}} \cdot c_{\text{policy}} \cdot T, \quad (3)$$

where $C_{\text{death}} = 10{,}000$, $C_{\text{overload}} = 100$ (per overload day), $\lambda_{\text{policy}} = 5$. Harm is normalised per 100,000 population. Monte Carlo: $M = 500$ replicates per (scenario, intervention) pair.

A.3.2 Supply Chain Oracle. The Supply Chain Oracle is an agent-based, discrete-time stochastic simulator [14] designed to capture the complex, dynamic behaviors of a distributed inventory network. It evaluates the performance of decentralised (r, Q) inventory control policies across interacting facilities under demand and lead-time uncertainties.

Table 4: Supply chain simulators used in the evaluate and forecast tasks.

Model	Key Mechanisms & Assumptions	Reference
Beer Game (BG)	Four-echelon chain (retailer → wholesaler → distributor → factory) with Sterman anchor-and-adjust heuristic ordering. Information and shipping delays of 2 periods each; captures the bullwhip effect.	[58]
Base Stock (BS)	Single-stage, periodic review. Orders up to a fixed base-stock level S each period; deterministic lead time L. Optimal under linear holding and backlog costs.	[51]
(s, S) Inventory	Orders $S - I_t$ only when inventory falls below reorder point s; fixed ordering cost K. Captures economies of scale in ordering.	[56]
Multi-Echelon (ME)	Decentralised three-level chain with independent (r, Q) policies at each level; deterministic lead times. Captures bullwhip amplification across echelons.	[51]

Table 5: Measles simulators used in the evaluate and forecast tasks.

Model	Compartments	Key Mechanisms & Assumptions	Reference
SIR	S, I, R	Permanent immunity post-infection; $R_0 \approx 12$–18 for measles.	[26, 33]
SEIR	S, E, I, R	Explicit latent period (σ); captures the ~8–12 day measles incubation.	[18]
SIRV	S, I, R, V	Single-dose vaccination (ν); susceptibles move to protected class.	[18]
SVEIR	S, V_1, V_2, E, I, R	Two-dose vaccination schedule with waning immunity; partial protection after dose 1, full after dose 2.	[18]
SVVIR	S, V_1, V_2, I, R	Compressed two-dose structure without explicit exposed state; used when latent-period data are unavailable.	[6]

Network structure. The system models a divergent three-echelon distribution network. It consists of one plant (acting as an infinite-inventory root), one central warehouse, two distribution centers, and four customer-facing distributors. Facilities interact via forward material flows (shipments) and backward information flows (orders).

Transition dynamics. The simulator operates on a discrete daily time step ($t = 1, \ldots, N_T$). Each facility agent autonomously monitors its inventory position, defined as on-hand inventory minus backorders plus on-order inventory. The transition logic dictates that when a facility's inventory position falls to or below its reorder point r_i, it places a replenishment order of quantity Q_i ($ROI_{it} = 1$). Unmet demands are strictly backordered. Order processing times follow truncated uniform distributions, but the actual cumulative lead time dynamically depends on upstream inventory availability: if the upstream facility is stocked out, it must first replenish its own inventory before processing the downstream order.

Key parameters. Daily customer order demands follow Gaussian distributions (with means ranging from 400 to 1000, and standard deviations from 200 to 500). Order processing times across edges are bounded uniform distributions (e.g., $[10, 15]$, $[8, 12]$, $[4, 6]$ days). Facility-level cost heterogeneity is modeled with a uniform holding cost of 1 m.u./(unit·day) , and varying fixed reorder costs: 300 m.u. for the warehouse, 200 m.u. for distribution centers, and 100 m.u. for distributors.

Objective function (Harm). While the original framework enforces a ≥ 95% fill rate as a probability constraint, the Oracle adapts this into a penalised composite scalar, *Harm*:

$$\text{Harm} = \text{INVC} + C_{\text{stockout}} \cdot \text{UnmetDemand} + \lambda_{\text{policy}} \cdot c_{\text{policy}} \cdot T,$$

where the base inventory cost (INVC) over the horizon N_T is calculated as the sum of average on-hand holding costs and discrete reorder setup costs:

$$\text{INVC} = \sum_{i=1}^{N_F} c_i^H \text{AIO}_i + \sum_{i=1}^{N_F} c_i^R \text{SRO}_i.$$

The C_{stockout} penalty replaces the strict service-level constraints for continuous policy evaluation. The expectations of these stochastic performance functions are estimated via the Monte Carlo method with $M = 300$ replicates per intervention.

A.3.3 Measles Oracle. The measles Oracle is a stochastic agent-based `measles::ModelMeaslesMixing` simulator accessed through `rpy2`/epiworldR [46, 64], matching the oracle definition used in the main benchmark description. For policy evaluation, we use pre-computed aggregate traces from this R-backed Oracle for each configuration–policy pair; horizons follow the scenario YAMLs, with five configurations running days 0–365 and Texas–NM 2025 running days 0–35. The incidence target is stored in the `target` column and is computed from daily depletion of `Susceptible` plus `Quarantined_Susceptible`, clipped at zero.

A.4 Intervention Policy Lists

A.4.1 COVID-19 Policies. The table summarizes the intervention policies used in the policy selection task, with each parameter defined following [9, 49].

A.4.2 Supply Chain Policies.

A.4.3 Measles Policies.

Table 6: COVID-19 intervention policies for the policy selection task.

ID	Title	Cost	R_0	Mechanism
A1	No Intervention	0.0	2.6	Baseline; no modifications.
A2	Mass Testing Only	0.10	2.4	Detection without isolation; mild WFH (×0.95 contact).
A3	Strict Lockdown	0.80	0.7	Full contact reduction (×0.25) with case isolation.
A4	Elderly Vaccination	0.30	1.8	Targeted high-risk vaccination; reduces mortality not transmission.
A5	Masking + WFH	0.20	1.6	Moderate contact reduction (×0.65); no isolation.
A6	Testing + Isolation	0.50	1.1	SIP (×0.55) + isolation of mild (×0.15) and severe (×0.08) cases.
Stress-test policies:				
A7	Reduce Trans. + Hospital Overload	0.35	1.2	Transmission ↓ but Oracle raises hospitalisation fraction (δ: 0.12→0.20).
A8	Delay Peak + Increase Deaths	0.25	1.5	Slows recovery; Oracle accelerates hospital death rate.
A9	Mild Cases Only	0.15	2.2	Isolates mild cases only; severe transmission unchanged in Oracle.

Table 7: Supply chain intervention policies for the policy selection task.

ID	Title	Cost	Mechanism
P1	No Intervention (Baseline)	0.0	Maintain calibrated (r, Q) baseline.
P2	Safety-Stock Boost (+50%)	0.25	Reorder point r raised by 50% at all facilities.
P3	Lead-Time Reduction (−30%)	0.40	Processing times on all edges cut by 30%.
P4	Demand Sharing / VMI	0.20	Demand std reduced 20% at distributors via information sharing.
P5	Order Quantity +30% (EOQ)	0.15	Order quantity Q raised 30%; reduces ordering frequency.
P6	Expedited Replenishment	0.55	Emergency reorder at $2 \times r$; near-eliminates stockouts.

Table 8: Measles intervention policies for the policy selection task.

ID	Policy	Cost	Mechanism
A1	No Intervention	0.0	Baseline with quarantine, isolation, and contact tracing disabled.
A2	Isolation Only	0.10	Symptomatic individuals self-isolate; quarantine and contact tracing are disabled.
A3	Quarantine + Contact Tracing	0.45	Contact tracing identifies exposures and quarantines traced contacts; self-isolation is disabled.
A4	School Closure	0.35	Scales the contact matrix to 50% of baseline contacts without quarantine, isolation, or tracing.
A5	Full Response	0.85	Combines isolation, quarantine, contact tracing, and school-closure contact reduction.
A6	Enhanced Full Response	1.10	Full response with no detection delay and a longer symptomatic isolation period.
A7	Vaccination Campaign	0.60	Increases configured per-group vaccination rates by a 1.10 multiplier, capped at 1.0.

B Experiment Details

B.1 Baseline Method Details

All baselines receive identical simulator outputs and produce structured explanations in the $\mathcal{E} = (\mathcal{I}, \mathcal{P}, \mathcal{Z}, \mathcal{C})$ schema.

Causal-Copilot [67]. An autonomous causal analysis agent that operationalises expert-level causal discovery within an LLM framework. We faithfully reimplement its four-phase pipeline: *(i)* Algorithm selection — an LLM characterises the data regime and selects from 20+ causal discovery algorithms (PC, FCI, GES, LiNGAM, NOTEARS, etc.); *(ii)* Causal graph construction — the selected algorithm is run on simulation output data; *(iii)* Causal effect estimation via do-calculus and regression adjustment; *(iv)* LLM interpretation grounded in the learned graph. This differs from MechSim in that Causal-Copilot discovers causal structure from observational data, whereas MechSim formalises the known mechanistic structure of each simulator via Mechanism Graphs.

Logic-LM [50]. Integrates symbolic reasoning with LLMs for faithful logical inference via three stages: *(i)* Symbolic formalisation — the LLM translates simulation summaries into first-order-logic predicates and transition rules; *(ii)* Constraint checking — a lightweight Python solver evaluates the rule set, derives a canonical simulator ranking, and flags logical contradictions; *(iii)* LLM interpretation grounded in verified logical derivations. Logic-LM provides structured conclusions but lacks grounding in domain-specific mechanistic assumptions.

Graph of Thoughts (GoT) [8]. Structures inference as a directed graph of thought nodes. We implement three GoT operations: *Generate* — spawn independent thought nodes per simulator and per outcome metric; *Aggregate* — merge compatible thoughts to identify common mechanistic drivers; *Refine* — improve the aggregated thought using retrieved scientific evidence. Unlike MechSim, GoT does not constrain reasoning paths to verified mechanistic graph transitions.

B.2 Calibration Methods

COVID-19 Policy Selection Task. Simulators are parameterised using the default scenario schema (Table 9); parameters are set from published epidemiological estimates. We sampled the parameters across 100 scenarios to test the robustness of our method.

COVID-19 forecast task. Each ODE simulator is calibrated to observed US weekly death data using particle filtering [61]. The objective minimises MAE normalised by mean observed cases on the training split. Settings: strategy `best1bin`; population $15 \times$ (num. params); max iterations 200; tolerance 10^{-7}; seed 42; polish enabled. Candidate simulators are SIR, SEIRD, SEIRHD, SEIQRD, SEAIHRD, and SIRV, each with their respective free parameters and logistic time-varying transmission rate $\beta(t)$.

Supply chain policy task. Two-step calibration: *(i)* run the Oracle for 28 periods with 5 Monte Carlo replicates under no-intervention (P1) to obtain "historical" demand and stockout series from the M5 dataset [42]; *(ii)* fit each simulator's parameters to match observed demand mean, standard deviation, and lead-time distribution. Candidate simulators include base-stock inventory control, (s, S) inventory control, decentralised multi-echelon (r, Q) control, and the Beer Game supply-chain model. Policy effects are modelled as covariate overrides (reorder-point multipliers, lead-time multipliers, demand multipliers) applied on top of the calibrated baseline.

Measles forecast task. Simulators are calibrated via Approximate Bayesian Computation (ABC) [7] using `epydemix`'s `ABCSampler`. Candidate simulators are SIR, SEIR, SIRV, SVEIR, and SVVIR, implemented with Epydemix [36]. The first 14 days of incidence are used for fitting; the remaining held-out horizon is then forecast. Prior distributions (all uniform): initial infected, bounded by early incidence; initial exposed for SEIR and SVEIR; transmission rate $\mathcal{U}(0.05, 1.50)$; recovery rate $\mathcal{U}(0.05, 0.50)$; incubation rate (SEIR, SVEIR) $\mathcal{U}(0.03, 0.33)$; vaccination rate, second-dose rate, and protection rate (SIRV, SVEIR, SVVIR) $\mathcal{U}(0.0, 0.08)$; waning rate $\mathcal{U}(0.0, 0.02)$.

B.3 Parameter Settings

B.4 Prompt Templates

Table 9: Default COVID-19 scenario parameter ranges for the evaluation and policy selection task. Values indicate the ranges explored in simulations.

Parameter	Value Range
Population N	log-uniform between 100,000 and 2,000,000
Basic reproduction number R_0	uniform between 2.0 and 3.5
Infection duration (days)	uniform between 1.0 and 6.0
Latent period (days)	uniform between 3.0 and 12.0
Initial infected I_0	log-uniform between 50 and 2,000
Hospital bed rate (per 1k population)	uniform between 1.5 and 10.0
Simulation horizon (days)	choice among 90, 120, 180, or 365

Parameters were set according to publicly available data and guidelines: [68] and [11].

Table 10: Key hyperparameter settings across all tasks and domains.

Component	Parameter	Value
LLM	Main model	claude-sonnet-4.6, DeepSeek V3.2, Qwen3-235B
	Max tokens (explanations)	2,000
	Temperature (reasoning / judge)	0.7 / 0.0
Knowledge Retrieval	API	arXiv, PubMed
	Re-ranker	Cross-encoder (two-stage, inspired by OpenScholar [3])
	Top-K papers	5; maximum abstract 800 characters
Simulation	Number of environments	3
	Samples per environment	100 per intervention
Calibration	Max iterations	200
	Population size	15× (number of parameters)
	Tolerance	10^{-7}
Evaluation	Judge LLM rounds	5 independent rounds
	Explanation dimensions	4 (completeness, mechanistic logic consistency, scientific soundness, faithfulness)
	Scoring scale	1–5 per dimension
	Policy metrics	Precision@3, Precision@5, Recall@3, Recall@5
Sensitivity Analysis	Parametric variants	4–5 values per parameter
	Structural variant types	Mixing pattern, immunity model, severity structure

We present the key prompt templates used by each MechSim agent. Placeholders `{...}` are populated at runtime.

B.4.1 *Context Understanding Agent.*

Prompt: Context Understanding Agent (Epidemic Domain)

System: You are an expert in epidemiological modeling and public health policy. Analyze the following epidemic scenario to construct a structured semantic context.

Task: Perform three functions simultaneously:

(1) **Environment Definition**: Identify real-world factors (population traits, healthcare capacity, geographic context) that constrain model assumptions and mechanisms.
(2) **Goal Identification**: Specify the decision-making objective (policy evaluation or forecasting).
(3) **Key Entity Recognition**: Extract critical variables from the scenario (R_0, β, γ, hospital beds, population).

[Scenario Specification]
Population: `{N}`; Initial Infected: `{I0}`; R0: `{R0}`; Hospital Beds: `{hospital_beds}`; Horizon: `{horizon}` days; Task: `{task}`

Return ONLY valid JSON with keys: `environment` (geographic_context, healthcare_capacity, real_world_factors), `goal` (primary_objective, decision_horizon, key_outcomes), `key_entities`, `assumption_validity_notes`.

B.4.2 *Mechanism Modeling Agent.*

Prompt: Mechanism Graph Extraction

You are an expert in scientific simulation, compartmental modeling, and causal inference. Your task is to extract a complete, structured mechanism graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{M}_i)$ for the `{model_name}` simulator, where nodes represent state variables or simulator entities, edges represent mechanistic transitions, and metadata captures assumptions, execution traces, and summary statistics.

[Simulator Definition] `{simulator_source_or_description}`
[Execution Traces] `{execution_traces}`
[Simulation Summary Statistics] Peak infection rate: `{peak_infection_rate}`; Total deaths: `{total_deaths}`; Hospital overload days: `{hospital_overload_days}`; Additional outputs: `{extra_summary_stats}`
[Scientific Evidence] `{retrieved_evidence}`

Extract the mechanism graph according to the following schema:

(1) **State nodes ($\mathcal{V}_i$):** List all simulator compartments or state variables (e.g., S, E, I, R, H, D, V). Each node must be a plain string matching the simulator's variable names exactly.
(2) **Mechanistic edges ($\mathcal{E}_i$):** For each transition, specify: `from`, `to` (plain strings); `mechanism` (the rate or process driving the transition, e.g., $\beta \cdot S \cdot I/N$); `activated_by` (the simulator assumption in $\mathcal{A}_i$ that enables this transition, e.g., homogeneous mixing, waning immunity).
(3) **Graph metadata ($\mathcal{M}_i$):** Extract the following:
   - `assumptions` $\mathcal{A}_i$: list all structural assumptions encoded by this simulator (e.g., permanent immunity, constant β, homogeneous mixing);
   - `execution_summary`: key statistics attached as graph-level metadata (peak infections, mortality, overload days);
   - `reasoning_path`: a topologically ordered sequence of nodes representing the primary propagation pathway from initial state to decision-relevant outcome;
   - `dot_code`: a Graphviz-compatible DAG representation of the full graph;
   - `scientific_summary`: a brief statement grounding the simulator's assumptions and mechanisms in retrieved scientific evidence.

Important constraints:

- Every `from`/`to` field must be a plain string; no nested objects or numeric indices.
- Do not introduce nodes or edges absent from the simulator definition or execution traces.
- If an assumption is absent or cannot be verified from the simulator source, explicitly mark it as `unverified` in $\mathcal{A}_i$.

Return ONLY valid JSON with the following top-level keys: `assumptions`, `nodes`, `edges`, `execution_summary`, `reasoning_path`, `dot_code`, `scientific_summary`.

B.4.3 *Mechanism Reasoning Agent.*

Prompt: Mechanism Reasoning Agent

Task: You are an expert in scientific simulation and mechanism-grounded reasoning. Your goal is to produce a structured, verifiable explanation of the simulation results by reasoning strictly over the extracted mechanism graphs, execution traces, and retrieved scientific evidence. Do **not** summarize outputs superficially; instead, trace how simulator assumptions activate mechanistic propagation paths that produce the observed outcomes.

[Simulation Results] `{sim_summaries["models"]}`

[Scenario Context] Environment: `{geographic_context}`; Objective: `{primary_objective}`; Assumption validity notes: `{assumption_validity_notes[:2]}`
[Mechanism Graphs] `{mechanism_graphs}`
[Execution Traces] `{execution_traces}`
[Scientific Evidence (RAG)] `{retrieved_evidence}`
[Sensitivity Analysis Findings] Top drivers: `{top_3_drivers}`; Validated hypotheses: `{validated_hypotheses[:2]}`. Use these findings to confirm, refine, or qualify each mechanistic claim.

[Mechanism Reasoning Constraints]

- You MUST follow the mechanism graph strictly. Each reasoning step must correspond to a valid node-to-node transition in the graph.
- Do NOT skip intermediate states or introduce variables absent from the graph.
- When evidence is incomplete, conflicting, or insufficient to support a claim, explicitly represent the uncertainty rather than asserting unsupported conclusions.
- Cross-reference sensitivity analysis findings at each step: if a transition corresponds to a top-ranked driver, explicitly note its empirical effect size.

For each simulator, trace the mechanism path step by step, e.g.:

```
[Model: SEIR-HD — Mechanism Path]
  Step 1 (S): Explain initial susceptible pool and inflow conditions.
  Step 2 (E): Describe S→E transition; identify the activating assumption.
  Step 3 (I): Describe E→I transition; link to sensitivity-ranked drivers.
  ...
  Step N (H/D): Explain terminal state; assess real-world implications.
```

[Structured Explanation — $\mathcal{E} = (\mathcal{I}, \mathcal{P}, \mathcal{Z}, \mathcal{C})$]
1. Output Interpretation ($\mathcal{I}$): Synthesize the scenario context, task objective, and simulator outputs. Identify decision-relevant patterns (e.g., peak divergence, mortality gaps, capacity breaches) and connect them to real-world implications for the deployment context.
2. Mechanism Reasoning Paths ($\mathcal{P}$): For each simulator, trace the full propagation path node-by-node. For each transition, explicitly state: (a) the mechanism label on the edge, (b) the assumption in $\mathcal{A}_i$ that activates it, and (c) whether sensitivity analysis confirms it as a key driver.
3. Supporting Evidence ($\mathcal{Z}$): For each claim, cite retrieved scientific evidence with specific quantitative findings. Where evidence conflicts with simulator predictions, explicitly flag the discrepancy and assess its impact on reliability.
4. Claims ($\mathcal{C}$): State 3–5 mechanism-grounded claims. Each claim must: (a) identify the responsible simulator assumption, (b) trace the full propagation path through $\mathcal{P}$, (c) cite a specific evidence reference from $\mathcal{Z}$, and (d) note any uncertainty or assumption-context mismatch that limits confidence.
5. Decision Recommendation ($\mathcal{R}$): Provide actionable, mechanism-grounded recommendations for the decision maker. All recommendations must be consistent with the verified explanation and finalized only after the full reasoning chain is complete.

B.4.4 Policy Selection Prompt.

Prompt: Policy Selection

You are an expert scientific advisor specializing in simulation-driven decision-making. Your task is to recommend the **top-3 intervention policies** from the candidate list, ranked from most to least effective, based on simulator outputs, mechanism graphs, sensitivity analysis findings, and retrieved scientific evidence.

[Context Information] `{context_summary}`
[Candidate Policies] `{policy_list_with_descriptions}`
[Calibrated Simulator Predictions] `{simulation_result}`
[Mechanism Graphs] `{mechanism_graphs}`
[Sensitivity Top Drivers] `{sensitivity_ranking[:3]}`
[Scientific Evidence] `{retrieved_evidence}`

Reason step by step for each candidate policy:

(1) **Mechanism tracing**: Identify which nodes and edges in the mechanism graph this policy activates or suppresses. Trace the full propagation path from intervention to outcome (e.g., policy → contact rate $\beta \downarrow$ → force of infection $\downarrow$ → peak infections $\downarrow$).
(2) **Simulator faithfulness**: Assess whether each candidate simulator structurally captures the mechanisms this policy targets. Flag simulators that lack the relevant compartments or assumptions needed to model the policy's effects.
(3) **Sensitivity alignment**: Cross-reference the top sensitivity drivers with the policy's mechanism. Prioritize policies that act on the highest-ranked drivers.
(4) **Evidence grounding**: Cite retrieved scientific evidence that supports or challenges the expected effectiveness of each policy under the given deployment context.
(5) **Assumption validity**: Explicitly note whether any simulator assumptions (e.g., homogeneous mixing, permanent immunity) may limit the reliability of predictions for this policy.

Return ONLY valid JSON with the following fields:
`ranking`: ordered list of top-3 policy IDs (best to worst);
`reasoning`: per-policy justification grounded in mechanism paths and evidence;
`mechanism_paths`: key propagation paths activated by each ranked policy;
`assumption_flags`: simulator assumptions that may affect prediction reliability;
`confidence`: per-policy confidence score (low / medium / high) with rationale.

B.4.5 *Verification Agent Prompt.*

Prompt: Verification Agent

You are a scientific validator for simulation-driven explanations. Given the mechanism graph $\mathcal{G}_i$, execution traces, and generated explanation $\mathcal{E} = (\mathcal{I}, \mathcal{P}, \mathcal{Z}, \mathcal{C})$ below, perform three verification checks.

[Mechanism Graph] `{mechanism_graph}` **[Execution Traces]** `{execution_traces}` **[Generated Explanation]** `{explanation}` **[Sensitivity Results]** `{sensitivity_results}`

1. Structural Consistency: (a) Do all nodes match valid compartments for `{model_name}`? (b) Are edges mechanistically valid with no impossible transitions (e.g., D→S, S→R without infection)? (c) Is each reasoning path in $\mathcal{P}$ a valid topological ordering of $\mathcal{G}_i$?
2. Evidence Grounding: (a) Are all $\mathcal{E}$-components present and substantive? (b) Is each claim in $\mathcal{C}$ traceable to a valid path in $\mathcal{P}$ and grounded in $\mathcal{Z}$, with no hallucinated variables or unsupported effect directions? (c) Are empirical findings from sensitivity analysis reflected where relevant?
3. Empirical Support: Are recommendations in $\mathcal{R}$ actionable, consistent with simulator outputs and verified mechanistic paths, and free of contradictions with retrieved evidence?

Return ONLY valid JSON: `status` (`PASS`|`WARN`|`FAIL`) per check; `correction_feedback` with concrete, targeted corrections for any failed check. Failed checks trigger iterative refinement by the Mechanism Reasoning Agent before final output is produced.

B.4.6 *LLM Judge Prompt.*

Prompt: Multi-Dimensional LLM Judge

You are an expert scientific judge evaluating simulation-driven explanations produced by different reasoning methods. Your evaluation must be rigorous, independent across dimensions, and grounded in the provided simulation results. Score each explanation on a **1–5 integer scale** across the following four dimensions.

[Context Information] `{context_summary}`
[Simulation Results] `{sim_summaries}`
[Mechanism Graphs] `{mechanism_graphs}`
[Explanations] **A** (Causal-Copilot), **B** (Logic-LM), **C** (GoT), **D** (MechSim): `{explanations}`

1. Completeness — Are all $\mathcal{E}$-components ($\mathcal{I}, \mathcal{P}, \mathcal{Z}, \mathcal{C}$) present with sufficient depth?

- **5**: All four components present; each is substantive, specific, and mutually consistent.
- **4**: All components present; one is underdeveloped but does not undermine overall coherence.
- **3**: At least three components present; missing or shallow component noticeably limits the explanation.
- **2**: One or two components missing or largely superficial.
- **1**: Explanation is a generic summary with no meaningful $\mathcal{E}$-structure.

2. Mechanistic Logic Consistency — Are reasoning steps logically consistent with the simulator's mechanism structure and free of invalid transitions?

- **5**: Every reasoning step corresponds to a valid node-to-node transition in $\mathcal{G}_i$; no skipped states or spurious variables.
- **4**: Reasoning is largely graph-consistent; at most one minor transition error that does not affect the conclusion.
- **3**: Core propagation path is correct, but intermediate steps are incomplete or loosely connected to the graph.
- **2**: Multiple invalid transitions or variables introduced outside the mechanism graph.
- **1**: Reasoning ignores the mechanism structure entirely or contains contradictory transitions.

3. Scientific Soundness — Are claims grounded in retrieved scientific evidence and established domain knowledge?

- **5**: All major claims are supported by specific cited evidence; no assertions contradict established scientific consensus.
- **4**: Most claims are evidence-grounded; one claim is asserted without citation but is plausible and consistent with domain knowledge.
- **3**: Some claims are evidence-grounded; others are generic or imprecise; no direct contradictions with established science.
- **2**: Claims are mostly unsupported or rely on vague domain knowledge without specific grounding.
- **1**: Claims contradict established scientific evidence or are entirely fabricated without reference to retrieved literature.

4. Faithfulness — Do claims accurately reflect simulation outputs without hallucination or directional errors?

- **5**: All quantitative and directional claims are consistent with the simulation outputs; no hallucinated values or reversed effects.
- **4**: Claims are faithful overall; at most one minor imprecision that does not affect the recommendation.

- **3**: Most claims are directionally correct, but some lack quantitative grounding or overstate simulation findings.
- **2**: Noticeable hallucinations or directional errors in one or more claims.
- **1**: Claims systematically misrepresent simulation outputs or invert effect directions.

Scoring instructions: Evaluate each dimension independently for all methods before moving to the next dimension. Do not anchor on the first explanation read or on overall impression. Assign integer scores only. If two methods are indistinguishable on a dimension, they may receive the same score.

Return ONLY valid JSON with the following structure: per-method integer scores (1–5) for each of the four dimensions; `winner` (method with highest average score); `ranking` (all methods ordered by average score, descending); `score_rationale` (one sentence per method per dimension justifying the assigned score).

C Additional Experimental Results

C.1 Example Context Information

Below is a representative context information block Ω extracted by the Context Understanding Agent for a COVID-19 policy evaluation scenario. This structured output is passed downstream to the Mechanism Modeling Agent, Sensitivity Analysis Agent, and Mechanism Reasoning Agent.

Context Information Ω: COVID-19 Policy Evaluation Scenario

Environment. Closed, well-mixed community of $N = 1{,}000{,}000$ individuals over a 180-day horizon. Hospital capacity: 1,000 acute-care beds ($\approx 0.1\%$ of population), representing a hard ceiling for simultaneous severe cases. Key real-world factors include age-dependent severity, heterogeneous contact patterns (households, workplaces, schools), and behavioral adaptation over time.

Goal. Task: *policy evaluation*. decision maker's question: determine why SIR, SEIRD, and SEIR-HD yield divergent epidemic trajectories under the same inputs, and identify the mechanisms driving those differences. Decision context: public-health officials responsible for surge-capacity planning and intervention timing.

Key Entities.
State variables: $S,\ E,\ I,\ H,\ R,\ D$
Key parameters:

- $R_0 = 2.6$; $\beta = 0.26$; $\gamma = 0.10$; $\sigma = 0.25$
- Hospital beds = 1,000; simulation horizon = 180 days
- Initial infectious $I_0 = 100$

Intervention targets: transmission rate β, contact rate, vaccination rate, hospital capacity expansion.
Outcome metrics: peak infection rate, total deaths, hospital overload days, time to peak, cumulative cases.

C.2 Example Explanation Output

We present a representative MechSim explanation generated for the COVID-19 evaluate task.

MechSim Output: Example Structured Explanation $\mathcal{E} = (\mathcal{I}, \mathcal{P}, \mathcal{Z}, \mathcal{C})$

Context. US national COVID-19 outbreak ($N = 500{,}000$; $R_0 = 2.8$; hospital capacity = 3,000 beds). Task: rank candidate intervention policies using three simulators — SIR, SEIRD, and SEIR-HD — calibrated to observed weekly death data. Evaluated policy: *school closure + work-from-home* (contact multiplier $\times 0.50$; cost = 0.35).

1. Output Interpretation ($\mathcal{I}$). All three simulators predict a reduction in peak daily infections under the evaluated policy, but diverge substantially in magnitude and in the outcomes they can represent. SIR projects $I_{\max} \approx 180{,}000$ (day 55); SEIRD projects $I_{\max} \approx 155{,}000$ (day 63), with 8,900 cumulative deaths versus SIR's 11,200. Critically, neither SIR nor SEIRD can represent hospital occupancy, leaving the primary capacity-planning bottleneck unobserved. SEIR-HD projects peak hospital occupancy of 12,400 beds — exceeding the 3,000-bed threshold by $4\times$ — sustained for 28 days, a signal invisible to the other two simulators. This divergence has direct decision relevance: a policy that appears effective under SIR or SEIRD may still produce system overload under SEIR-HD.

2. Mechanism Reasoning Paths ($\mathcal{P}$).
SIR ($\mathcal{A}_i$: homogeneous mixing; constant β; permanent immunity):

$$S \xrightarrow{\beta(t)\downarrow} I \xrightarrow{\gamma} R$$

School closure reduces the effective contact rate ($\beta : 0.26 \to 0.13$), directly lowering the force of infection and suppressing the $S \to I$ transition. The absence of a latent compartment causes the peak to arrive earlier and be overestimated relative to models with exposed states.
SEIRD ($\mathcal{A}_i$: exposed latency $\sigma^{-1} = 4$ days; disease-induced mortality μ):

$$S \xrightarrow{\beta} E \xrightarrow{\sigma} I \xrightarrow{\gamma} R \ / \ \xrightarrow{\mu} D$$

The exposed compartment buffers the $S \to I$ flow, smoothing the epidemic curve and shifting the peak by +8 days. Lower peak infections (13% below SIR) reflect latency damping. Mortality is modelled explicitly, but hospital dynamics remain outside this simulator's scope.
SEIR-HD ($\mathcal{A}_i$: hospitalisation branch; capacity threshold; separate community vs. hospital CFR):

$$S \xrightarrow{\beta} E \xrightarrow{\sigma} I \xrightarrow{\alpha} H \xrightarrow{\gamma_H} R \ / \ \xrightarrow{\delta} D$$

With hospitalisation rate $\alpha = 0.02$ and mean hospital stay $\gamma_H^{-1} = 10$ days, the $I \to H$ branch sustains occupancy above capacity for 28 days. Crucially, α is not modified by the school-closure covariate, meaning adult severe cases accumulate at baseline rates even as child-to-child transmission declines.

3. Supporting Evidence ($\mathcal{Z}$).

- **[Z1]** Cauchemez et al. (2008, *Nature*). *Estimating the impact of school closure on influenza transmission from sentinel data.* Reports 20–40% reduction in transmission during school holidays in high-density settings. Supports the mechanistic effect of the $\beta \downarrow$ transition under school closure, though the upper bound of our assumed 50% reduction warrants caution in adult-dominated contact networks.
- **[Z2]** Holmdahl & Buckee (2020, *New England Journal of Medicine*). *Wrong but useful — what COVID-19 epidemiologic models can and cannot tell us.* Demonstrates that hospital-explicit compartmental models are necessary for reliable capacity-planning forecasts, and that simpler compartmental models systematically underestimate overload risk. Directly supports the claim that SIR and SEIRD are structurally blind to the $I \to H$ pathway.
- **[Z3]** Lauer et al. (2020, *Annals of Internal Medicine*). *The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases.* Reports a median incubation period of 5.1 days (95% CI: 4.5–5.8 days). Validates SEIRD's latency assumption ($\sigma^{-1} = 4$ days) and supports the 8-day peak shift relative to SIR.
- **[Z4]** Ferguson et al. (2020, *Imperial College London Report 9*). *Impact of non-pharmaceutical interventions to reduce COVID-19 mortality and healthcare demand.* Projects that combined school closure and social distancing can suppress peak ICU demand by up to 67%, but emphasizes that hospital surge capacity remains a binding constraint even under aggressive NPIs. Corroborates

SEIR-HD's prediction of sustained overload despite policy implementation.

4. Claims ($\mathcal{C}$).

(1) The *homogeneous mixing* assumption shared by all three simulators uniformly scales all contact types, causing school closure to reduce adult workplace contacts in addition to child school contacts. This overestimates the policy effect for working-age adults. *[Path: $\beta(t) \to S \to I$; Evidence: Z1]*

(2) SIR overestimates peak infections by 13% relative to SEIRD because it lacks a latent compartment, causing exposed individuals to become immediately infectious. This compresses the epidemic curve and advances the peak by 8 days. *[Path: $S \to I$ vs. $S \to E \to I$; Evidence: Z3]*

(3) SIR and SEIRD structurally cannot detect hospital overload: the $I \to H$ branch is absent, making these simulators unreliable for capacity-planning decisions under any policy. *[Evidence: Z2]*

(4) Under SEIR-HD, school closure does not reduce the hospitalisation rate α, so severe adult cases accumulate at baseline rates. Even with infection suppression, hospital occupancy exceeds capacity by 4× for 28 days, indicating that transmission reduction alone is insufficient. *[Path: $I \xrightarrow{\alpha} H$; α invariant to policy covariate; Evidence: Z4]*

5. Decision Recommendation ($\mathcal{R}$). Simulator selection: Use SEIR-HD for forecasting and policy evaluation in this scenario; it is the only simulator that exposes hospital overload risk, which is the binding constraint under the current deployment context. **Policy recommendation**: School closure should be implemented but is insufficient alone. SEIR-HD predicts 28 days of capacity breach even under full compliance; complementary measures targeting adult severe cases (e.g., targeted elderly vaccination, hospital surge capacity expansion) are required. **Uncertainty note**: The homogeneous mixing assumption shared by all simulators may overstate the policy's effect on adult transmission; network-based contact models should be considered if heterogeneous contact data are available.

C.3 Example Mechanism Graph

Here shows the mechanism graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{M}_i)$ automatically extracted by the Mechanism Modeling Agent for the SEIR-HD simulator. Nodes represent compartments ($\mathcal{V}_i$); directed edges represent mechanistic transitions ($\mathcal{E}_i$), labelled with the governing rate; and graph-level metadata ($\mathcal{M}_i$) includes simulator assumptions and execution statistics.

Graphviz DOT Code: SEIR-HD Mechanism Graph

```
digraph SEIR_HD {
    rankdir=LR;
    graph [
        label="SEIR-HD Mechanism Graph\n\
Assumptions:\n\
  A2: homogeneous mixing\n\
  A3: latent (incubation) period\n\
  A4: hospitalisation branch\n\
  A5: hospital CFR (recovery vs. death)\n\
  A6: permanent immunity post-recovery\n\
  A7: deterministic ODE dynamics\n\
-------------------------------------------\n\
Execution Statistics (calibrated run):\n\
  Peak Infection Rate : 0.0128\n\
  Total Deaths        : 2764\n\
  Hospital Overload   : 69 days",
        labelloc=b,
        fontsize=11,
        fontname="Helvetica"
    ];

    /* -- Node declarations -- */
    node [shape=circle, style=filled,
          fontname="Helvetica", fontsize=12];

    S [label="S\nSusceptible",  fillcolor="#AED6F1"];
    E [label="E\nExposed",      fillcolor="#A9DFBF"];
    I [label="I\nInfectious",   fillcolor="#F9E79F"];
    H [label="H\nHospitalised", fillcolor="#F0B27A"];
    R [label="R\nRecovered",    fillcolor="#D7BDE2"];
    D [label="D\nDeceased",     fillcolor="#F1948A"];

    /* -- Mechanistic edges -- */
    edge [fontname="Helvetica", fontsize=10];

    S -> E [label="beta*S*I/N  [A2]",
            color="#2E86C1", fontcolor="#2E86C1"];

    E -> I [label="sigma (1/incubation)  [A3]",
            color="#1E8449", fontcolor="#1E8449"];

    I -> R [label="gamma (community recovery)  [A4]",
            color="#7D3C98", fontcolor="#7D3C98"];

    I -> H [label="theta (hospitalisation rate)  [A4]",
            color="#CA6F1E", fontcolor="#CA6F1E",
            style=bold];

    H -> R [label="gamma_H (hospital recovery)  [A5]",
            color="#7D3C98", fontcolor="#7D3C98",
            style=dashed];

    H -> D [label="mu_H (hospital CFR)  [A5]",
            color="#C0392B", fontcolor="#C0392B",
            style=bold];

    /* -- Primary propagation path (S->E->I->H->D) -- */
    {rank=same; S; E; I; H; D}
}
```

C.4 Simulation Result

COVID-19 Forecast Task. Table 11 reports simulator performance on held-out US COVID-19 data. The task requires selecting the best simulator prior to the forecast cutoff using training-period evidence alone.

Table 11: COVID-19 simulator forecast performance on held-out period (US national). Lower WIS, CRPS, nRMSE are better; higher coverage is better.

Simulator	Mean WIS	Mean CRPS	nRMSE	Cov. (90%)
SIR (Dynamic)	87,312	14,821	1.031	0.712
SEIRD (Dynamic)	79,450	13,642	0.924	0.764
SEIR-HD	68,234	11,508	0.847	0.829
SEIQRD	73,891	12,201	0.889	0.801
SEAIHRD	71,105	11,944	0.872	0.815
SIRV	80,117	13,299	0.951	0.758
ParticleFilter-SEIRH (ref.)	73,485	12,354	0.964	0.890
ARIMA (ref.)	52,688	7,216	0.438	0.829
Constant (ref.)	115,269	10,978	0.484	0.000

Supply Chain Policy Selection Task. Table 12 reports simulator-predicted harm scores across six candidate policies for a representative scenario (Scenario 2). The oracle identifies Policy 4 (Expedited Replenishment) as optimal.

C.5 Sensitivity Analysis Results

Table 13 shows parametric sensitivity scores for the COVID-19 default scenario. Scores measure the interventional effect $\mathbb{E}[y \mid \mathrm{do}(\phi{=}\phi_2)] - \mathbb{E}[y \mid \mathrm{do}(\phi{=}\phi_1)]$. Table 14 summarises structural sensitivity results from assumption-level interventions.

Table 14: Structural sensitivity analysis: impact of modifying key modeling assumptions.

Assumption	Variant A	Variant B	Δ Peak	Δ Deaths
Mixing pattern	Homogeneous	Network-based	+14.9%	+8,855
Immunity	Permanent	Waning (180d)	+22.3%	+15,240
Severity dist.	Mild-dominant	Severe-dominant	+8.1%	+31,100

C.6 Ablation Study Results

Figure 1 shows selected COVID-19 ablation results. Removing mechanism reasoning causes the largest degradation, with relative score reductions of around 50% on completeness, mechanism-logic consistency, scientific soundness, and faithfulness. This shows that explicit mechanism reasoning contributes to the framework's explanation gains. Removing context understanding and knowledge retrieval has a smaller but consistent effect, with relative reductions of around 18% on the same four criteria. Overall, the ablation results suggest that the framework's components work together to support high-quality explanations, with mechanism reasoning playing a central role. In addition, the degradation is larger for less powerful LLMs, suggesting that the framework's structured reasoning may be especially beneficial for weaker LLMs.

C.7 Comprehensive Policy Selection Task Result

We put the Comprehensive Policy Selection Task Result with standard deviation here in Table 15.

Table 12: Simulator-predicted harm across six candidate policies (Scenario 2, Supply Chain). Values in $k. Bold = oracle-optimal policy (P4).

Simulator	P1	P2	P3	P4	P5	P6
Oracle	74.0	102.7	96.2	**58.1**	96.5	99.7
Base Stock	0.6	135.1	225.0	113.1	81.0	315.0
(s, S) Inventory	0.9	135.9	225.6	113.4	81.9	315.9
Multi-Echelon	12.1	12.5	12.3	12.2	12.2	12.7
Beer Game	21.1	156.1	246.1	133.6	102.1	336.1

Table 13: Parametric sensitivity analysis: COVID-19 evaluate task, outcome = peak infection rate. Rank 1 is most influential.

Parameter	Score	Rank	Direction	Interpretation
Transmission rate (R_0)	5.57	1	Positive	Higher R_0 → higher, earlier peak
Contact rate	2.35	2	Positive	Higher contact rate → faster spread
Behavioral reduction (α)	0.85	3	Negative	Stronger NPI → lower peak
Recovery rate (γ)	0.81	4	Negative	Faster recovery → lower, later peak

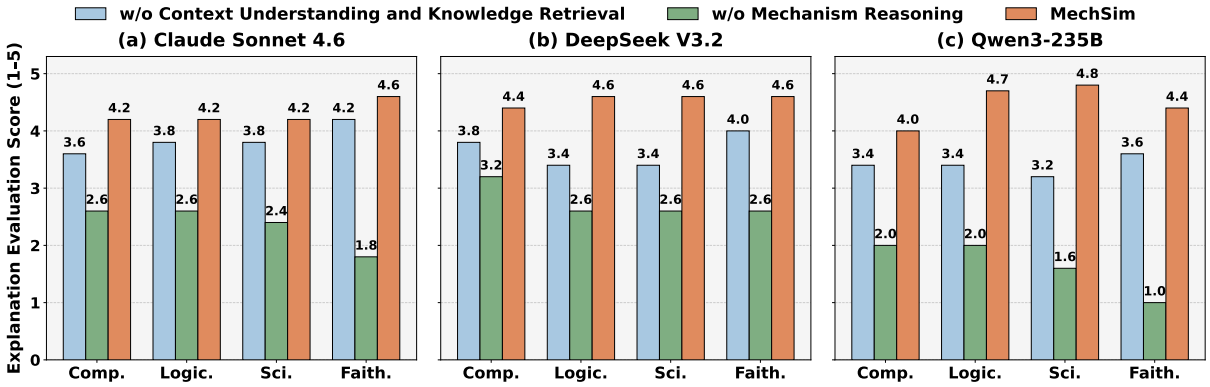

Figure 1: Selected COVID-19 ablation results for the explanation evaluation task across three LLM backbones. "w/o" denotes removal of the module or module group named in the legend.

C.8 Comprehensive Simulator Selection Task Result

We put the Comprehensive Simulator Selection Task Result with standard deviation here in Table 16.

Table 15: Policy selection results (Precision@k and Recall@k) across three domains and three LLM backbones. Bold = best per metric within each domain/LLM group. "GoT" denotes Graph of Thoughts.

Domain	LLM	Method	P@3	P@5	R@3	R@5
COVID-19	Claude	Causal-Copilot	$0.667_{\pm.000}$	$0.720_{\pm.103}$	$0.400_{\pm.024}$	$0.701_{\pm.105}$
		Logic-LM	$0.633_{\pm.105}$	$0.620_{\pm.114}$	$0.380_{\pm.063}$	$0.645_{\pm.138}$
		GoT	$0.633_{\pm.105}$	$0.600_{\pm.133}$	$0.380_{\pm.063}$	$0.624_{\pm.133}$
		MechSim	$\mathbf{0.823}_{\pm.161}$	$\mathbf{0.714}_{\pm.105}$	$\mathbf{0.540}_{\pm.097}$	$\mathbf{0.720}_{\pm.103}$
	DeepSeek	Causal-Copilot	$0.633_{\pm.105}$	$0.500_{\pm.105}$	$0.380_{\pm.063}$	$0.700_{\pm.105}$
		Logic-LM	$0.600_{\pm.141}$	$0.640_{\pm.084}$	$0.360_{\pm.084}$	$0.640_{\pm.084}$
		GoT	$0.667_{\pm.000}$	$0.520_{\pm.103}$	$0.423_{\pm.036}$	$0.520_{\pm.103}$
		MechSim	$\mathbf{0.833}_{\pm.176}$	$\mathbf{0.720}_{\pm.103}$	$\mathbf{0.500}_{\pm.105}$	$\mathbf{0.720}_{\pm.103}$
	Qwen	Causal-Copilot	$0.667_{\pm.000}$	$0.680_{\pm.103}$	$0.640_{\pm.097}$	$0.580_{\pm.132}$
		Logic-LM	$0.700_{\pm.105}$	$0.600_{\pm.094}$	$0.420_{\pm.105}$	$0.667_{\pm.086}$
		GoT	$0.804_{\pm.142}$	$0.520_{\pm.103}$	$0.650_{\pm.105}$	$0.721_{\pm.124}$
		MechSim	$\mathbf{0.833}_{\pm.176}$	$\mathbf{0.700}_{\pm.105}$	$\mathbf{0.760}_{\pm.084}$	$\mathbf{0.766}_{\pm.132}$
Supply Chain	Claude	Causal-Copilot	$0.783_{\pm.163}$	$0.810_{\pm.045}$	$0.470_{\pm.098}$	$0.543_{\pm.231}$
		Logic-LM	$0.817_{\pm.170}$	$0.830_{\pm.073}$	$0.490_{\pm.102}$	$0.350_{\pm.408}$
		GoT	$0.783_{\pm.163}$	$0.812_{\pm.089}$	$0.470_{\pm.098}$	$0.544_{\pm.357}$
		MechSim	$\mathbf{0.900}_{\pm.157}$	$\mathbf{0.840}_{\pm.082}$	$\mathbf{0.540}_{\pm.094}$	$\mathbf{0.569}_{\pm.183}$
	DeepSeek	Causal-Copilot	$0.833_{\pm.171}$	$0.810_{\pm.045}$	$0.500_{\pm.103}$	$0.810_{\pm.045}$
		Logic-LM	$0.867_{\pm.168}$	$0.850_{\pm.089}$	$0.520_{\pm.101}$	$0.850_{\pm.089}$
		GoT	$0.833_{\pm.171}$	$0.810_{\pm.045}$	$0.500_{\pm.103}$	$0.810_{\pm.045}$
		MechSim	$\mathbf{0.967}_{\pm.103}$	$\mathbf{0.860}_{\pm.094}$	$\mathbf{0.580}_{\pm.062}$	$\mathbf{0.860}_{\pm.094}$
	Qwen	Causal-Copilot	$0.856_{\pm.168}$	$0.833_{\pm.076}$	$0.513_{\pm.101}$	$0.488_{\pm.170}$
		Logic-LM	$0.867_{\pm.166}$	$\mathbf{0.856}_{\pm.186}$	$0.520_{\pm.100}$	$0.490_{\pm.241}$
		GoT	$0.845_{\pm.169}$	$0.840_{\pm.081}$	$0.507_{\pm.102}$	$0.501_{\pm.188}$
		MechSim	$\mathbf{0.911}_{\pm.150}$	$0.842_{\pm.046}$	$\mathbf{0.547}_{\pm.090}$	$\mathbf{0.524}_{\pm.192}$
Measles	Claude	Causal-Copilot	$0.556_{\pm.248}$	$0.733_{\pm.149}$	$0.363_{\pm.124}$	$0.742_{\pm.109}$
		Logic-LM	$\mathbf{0.722}_{\pm.299}$	$0.767_{\pm.180}$	$0.367_{\pm.143}$	$\mathbf{0.833}_{\pm.180}$
		GoT	$0.333_{\pm.038}$	$0.631_{\pm.187}$	$0.293_{\pm.902}$	$0.657_{\pm.103}$
		MechSim	$0.611_{\pm.299}$	$\mathbf{0.833}_{\pm.180}$	$\mathbf{0.433}_{\pm.180}$	$0.767_{\pm.129}$
	DeepSeek	Causal-Copilot	$0.611_{\pm.299}$	$0.767_{\pm.180}$	$0.367_{\pm.146}$	$0.767_{\pm.197}$
		Logic-LM	$0.722_{\pm.404}$	$0.667_{\pm.249}$	$\mathbf{0.433}_{\pm.243}$	$0.667_{\pm.249}$
		GoT	$0.579_{\pm.196}$	$0.425_{\pm.127}$	$0.428_{\pm.579}$	$0.648_{\pm.209}$
		MechSim	$\mathbf{0.767}_{\pm.180}$	$\mathbf{0.767}_{\pm.180}$	$0.379_{\pm.180}$	$\mathbf{0.767}_{\pm.180}$
	Qwen	Causal-Copilot	$0.667_{\pm.272}$	$0.767_{\pm.180}$	$0.400_{\pm.163}$	$0.767_{\pm.180}$
		Logic-LM	$0.722_{\pm.299}$	$0.733_{\pm.275}$	$0.467_{\pm.221}$	$0.733_{\pm.275}$
		GoT	$0.333_{\pm.138}$	$0.648_{\pm.128}$	$\mathbf{0.583}_{\pm.256}$	$0.529_{\pm.108}$
		MechSim	$\mathbf{0.778}_{\pm.369}$	$\mathbf{0.833}_{\pm.180}$	$0.433_{\pm.139}$	$\mathbf{0.833}_{\pm.153}$

Table 16: Full simulator selection results with standard deviations (Top-k regret, $k \in \{1, 3\}$; *lower is better*). Values are reported as mean±std. Bold denotes the best (lowest) mean within each (LLM, Domain, Metric) group.

LLM	Domain	Metric	Causal-Copilot	Logic-LM	GoT	MechSim (Ours)
Claude Sonnet 4.6	COVID-19	Top-1 Regret	2.590±2.340	3.910±3.180	3.800±3.000	**2.040±2.000**
		Top-3 Regret	2.600±1.760	2.410±1.730	2.530±1.690	**2.290±1.750**
	Supply Chain	Top-1 Regret	6.170±3.480	5.420±4.020	5.920±0.370	**4.250±4.390**
		Top-3 Regret	5.350±3.650	4.380±4.310	5.140±0.391	**3.470±4.140**
	Measles	Top-1 Regret	2.880±3.900	2.350±3.090	1.930±3.210	**1.690±1.140**
		Top-3 Regret	3.400±4.530	3.210±2.060	2.790±2.830	**2.200±3.530**
DeepSeek V3	COVID-19	Top-1 Regret	2.380±2.690	2.700±3.020	2.470±2.730	**2.160±2.620**
		Top-3 Regret	2.100±1.800	2.140±1.810	2.180±1.880	**2.110±1.790**
	Supply Chain	Top-1 Regret	0.617±0.348	0.542±0.402	0.592±0.370	**0.467±0.407**
		Top-3 Regret	0.535±0.365	0.438±0.431	0.514±0.391	**0.389±0.393**
	Measles	Top-1 Regret	**0.548±1.217**	5.940±8.936	2.486±4.204	0.702±1.181
		Top-3 Regret	2.994±4.146	3.261±4.235	2.764±3.469	**2.278±2.877**
Qwen3-235B	COVID-19	Top-1 Regret	1.790±0.850	1.780±0.870	1.760±0.860	**1.650±0.880**
		Top-3 Regret	1.700±0.750	1.850±0.910	1.700±0.750	**1.680±0.830**
	Supply Chain	Top-1 Regret	0.593±0.334	0.621±0.361	0.591±0.338	**0.240±0.306**
		Top-3 Regret	0.556±0.162	0.514±0.190	0.556±0.162	**0.513±0.216**
	Measles	Top-1 Regret	1.870±4.108	2.693±2.999	3.036±5.321	**1.128±2.008**
		Top-3 Regret	2.549±3.183	3.172±4.287	2.698±3.427	**2.488±4.187**