SIMULATE, REASON, DECIDE: SCIENTIFIC REASONING WITH LLMS FOR SIMULATION-DRIVEN DECISION MAKING

Anonymous authors
Paper under double-blind review

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 traceability, systematic verification, 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 using a shared structured schema that captures 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/`.

1 INTRODUCTION

LLM agents provide a natural interface for automating decision-making workflows that ingest multimodal information from diverse sources and generate recommendations (Ma et al., 2025; Xiao et al., 2024). 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 (Kim et al., 2024; Datta et al., 2026; Choukhmane et al., 2026). 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 (Marathe & Vullikanti, 2013; Banks et al., 2010; Carson, 2005).

Despite this interest, existing approaches primarily use LLMs to generate, calibrate, or execute simulation models (Holt et al., 2025; Datta et al., 2026; Wahl et al., 2026), with limited support for reasoning about the simulators themselves. The central obstacle is not access. Given a simulator repository and an execution environment, a contemporary coding agent can inspect equations, trace code, run perturbation experiments, and cite implementation details. Such analyses, however, typically yield query-specific explanations in which the links among assumptions, simulator operations, and observed outcomes remain implicit. Consequently, a mechanistically plausible explanation may attribute an effect to the wrong rate law, transition, or contact rule without specifying the simulator intervention that would confirm or refute the claim. Moreover, the links established among equations, code, configuration, documentation, and execution traces remain dispersed across the agent's transcript and generated response. Subsequent tasks must therefore reconstruct which variables an intervention modifies, which mechanisms propagate its effects, which claims are supported by execution evidence, and which alternatives have been tested.

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

What such reconstruction involves is easiest to see in a concrete case. Consider a public-health decision involving an intervention intended to reduce disease transmission. An epidemic simulator may predict that the intervention lowers peak infections, yet the output alone does not reveal whether this effect arises from reduced contact frequency, changes in mobility, or assumptions about the contact-network structure. Moreover, the same intervention may enter different simulators through different mechanisms—for example, as a reduction in the aggregate contact rate in a compartmental model or as the removal of selected contact-layer edges in a network model. Assessing the resulting recommendation therefore requires determining how the intervention is represented, which mechanisms produce the predicted effect, and whether the simulator's assumptions are appropriate for the deployment context. These are exactly the relationships that a query-specific argument leaves implicit. Leaving them implicit is untenable when models inform consequential decisions, where transparency, provenance, and critical scrutiny are essential (Rudin, 2019; Saltelli et al., 2020; Holmdahl & Buckee, 2020). What is needed instead is a structured representation that makes generated claims traceable to simulator structure, assumptions, executions, and supporting evidence, so that the structural and empirical support for a mechanistic explanation can be checked and the decisions it informs can be audited. We therefore ask: *How can these heterogeneous sources be integrated into a structured representation for verifiable, mechanism-grounded LLM reasoning?*

To address this question, we introduce MechSim, a neuro-symbolic framework that represents executable simulators as structured mechanistic systems and constrains LLM reasoning over those representations. Our main contributions are threefold.

1. We introduce the **Simulation Reasoning Graph (SRG)**, a structured representation that organizes simulator variables, assumptions, mechanism dependencies, execution traces, and sensitivity-analysis artifacts. By representing heterogeneous simulators through a common schema, SRGs provide a shared symbolic substrate for comparative reasoning.
2. We develop a **mechanism-grounded reasoning and verification framework** that constructs explanations by tracing simulator outcomes through the SRG and linking individual claims to simulator structure, execution behavior, or scientific evidence. This approach makes the resulting reasoning explicit and traceable, enabling claims to be checked against the executable simulator.
3. We conduct **extensive experiments across three high-stakes domains** with multiple LLM backbones. The results (including an evaluation by 23 domain experts) show that MechSim consistently improves the quality of explanations and generally achieves stronger downstream decision performance than representative causal, logical, and graph-based reasoning baselines.

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 (Uhrmacher et al., 2024). 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 (Holmdahl & Buckee, 2020). 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) **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. (2) **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.

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 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. We address 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 three tightly coupled components: (1) contextual understanding, (2) simulator execution and grounding, and (3) mechanism reasoning and verification. The symbolic layer of MechSim is built around a structured simulator representation, which we call the *Simulation Reasoning Graph (SRG)*. The SRG encodes simulator assumptions, execution traces, and sensitivity-analysis artifacts. The neural layer consists of LLM-based contextual interpretation, scientific evidence synthesis, explanation generation, and decision reasoning. SRGs 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, SRG construction, 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 Understanding

The *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 $\mathcal{K}(q, \mathcal{D}) \to \mathcal{Z}$ retrieves relevant scientific evidence using a dense retrieval and reranking pipeline inspired by OpenScholar (Asai et al., 2026). The resulting evidence set $\mathcal{Z}$ provides scientific grounding for downstream reasoning and reduces unsupported or hallucinated explanations (Ji et al., 2023; Huang et al., 2025).

3.2 Simulator Execution and Grounding

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) (Sisson et al., 2018; Temfack & Wyse, 2025) 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

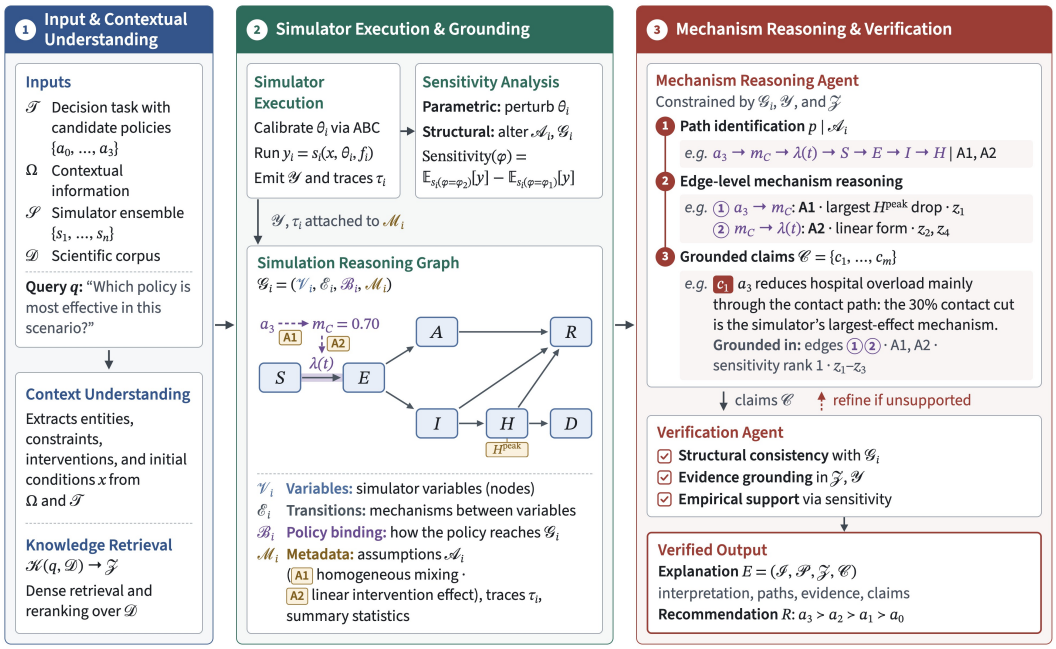


Figure 1: Overview of MechSim, a neuro-symbolic framework for mechanism-grounded reasoning over scientific simulators. LLM agents (1) extract decision-relevant context and retrieve scientific evidence; (2) calibrate, execute, and perturb simulators to generate a simulation reasoning graph $\mathcal{G}_i$ that binds policies to the variables they affect; and (3) reason over mechanism paths in $\mathcal{G}_i$ to produce claims, which are verified before emitting the explanation E and recommendation R.

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 (Saltelli et al., 2008). 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 $\phi \in \{\theta_i, \mathcal{A}_i\}$ and computes $\text{Sensitivity}(\phi) = \mathbb{E}_{s_i(\phi=\phi_2)}[y] - \mathbb{E}_{s_i(\phi=\phi_1)}[y]$, where ϕ_1 and ϕ_2 denote alternative parameter settings or assumptions. These interventions provide empirical evidence supporting, rejecting, or refining candidate explanations.

Simulation Reasoning Graph (SRG) Construction. For each simulator $s_i \in \mathcal{S}$, MechSim constructs a SRG $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{B}_i), \mathcal{M}_i)$, 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. To help users better understand how a policy acts in a given case, the policy binding $\mathcal{B}_i$ explains how the policy affects the simulator and its outcomes. The metadata $\mathcal{M}_i$ includes simulator assumptions $\mathcal{A}_i$, execution traces τ_i, and summary statistics derived from simulator execution. The simulation reasoning graph serves as the symbolic representation that constrains 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 $\mathcal{G}_i$, we have a *Mechanism Modeling Agent* that 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.

3.3 Mechanism Reasoning and Verification

SRG-constrained Reasoning. Our *Mechanism Reasoning Agent* performs constrained reasoning over the SRG, which includes 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 simulation reasoning graph $\mathcal{G}$, 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 simulation reasoning 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) (Miller, 2019; Toulmin, 2003), 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 $p \mid \mathcal{A}_i = (v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_n)$ 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 (Machamer et al., 2000).
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 $\mathcal{C}$ (Xu et al., 2026).
4. **Claims ($\mathcal{C}$):** $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ 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.

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. Therefore, building on previous work in verification (Kleijnen, 1995; Sargent, 2010), we require a valid explanation $E = (\mathcal{I}, \mathcal{P}, \mathcal{Z}, \mathcal{C})$ and recommendation R must satisfy three properties: (1) **Structural Consistency:** each claim $c \in \mathcal{C}$ must be traceable to valid reasoning paths within simulation reasoning 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 our verification checks detect 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 Experimental Setup

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/`, and the supplementary materials are provided alongside.

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 (Howerton et al., 2023). We use a stochastic, multi-compartment, discrete-time simulator as the oracle simulator and ground truth (Kain et al., 2021). 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 (Cramer et al., 2022).

(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 (Sterman, 2000). For the dataset, we use the M5 dataset, which includes hierarchical point-of-sale records at the item–store level in Walmart (Makridakis et al., 2022).

(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 (Allotey, 2017; Masters, 2024; González-Parra et al., 2025), 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 (Kuddus et al., 2021). The oracle simulator is a stochastic agent-based `measles::ModelMeaslesMixing` model accessed through `rpy2`/epiworldR (Meyer & Vega Yon, 2023; Vega Yon, 2026). Although measles and COVID-19 are both respiratory viral infections, they differ substantially in transmission dynamics and intervention strategies (Gaythorpe et al., 2021). 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*** (Wang et al., 2025) is an autonomous agent that operationalizes expert-level causal analysis within an LLM framework. (2) ***Logic-LM*** (Pan et al., 2023) integrates symbolic reasoning with LLMs to support structured, logically grounded analysis. (3) ***Graph of Thoughts (GoT)*** (Besta et al., 2024) 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 and the selection of baselines is available in Appendix B.1.

4.3 TASKS

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).

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

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 (Cesa-Bianchi & Lugosi, 2006): Regret $=$ RMSE$(\hat{s})$ $-$ $\min_{s\in\mathcal{S}}$ 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.

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. We evaluate the recommended ranking with Precision@K and Recall@K ($K \in \{3, 5\}$) (Manning, 2008; Liu, 2009), which compare each method's top-K policies with the oracle-preferred set: Precision@K is the fraction of the top-K recommendations that are oracle-preferred, and Recall@K is the fraction of the oracle-preferred set recovered by the top-K recommendations. To summarize both in a single number, we report F1@K, the harmonic mean of Precision@K and Recall@K, averaged over $K \in \{3, 5\}$ in the main text; full precision and recall results are given in Appendix C.2. Higher values indicate stronger alignment with the oracle ranking.

Explanation Evaluation Task: This task assesses whether a method can reliably explain simulator behavior in realistic decision-making scenarios. Given outputs from multiple simulators, each method identifies and explains the mechanisms underlying the observed outcomes. We first conduct an independent evaluation with 23 human domain experts. We then complement this assessment with an LLM-based evaluation, repeated over five random seeds with scores averaged across runs. For each evaluation, we use a different judge model family from the LLM used for generation to ensure a fair assessment.

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 (DeYoung et al., 2020; Liu et al., 2023). These metrics are also used to evaluate generated explanations in the simulator and policy selection tasks. For more details on the metrics, please refer to the LLM Judge Prompt in Appendix E.

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 Simulator Selection for Forecasting Results

Figure 2(a) relates these regret values to the quality of the explanations produced along the way. For every domain and LLM backbone, MechSim lies in the lower-right region, combining the lowest regret with the highest explanation score, while the baselines spread across the upper-left. The separation is largest in Supply Chain, where MechSim is the only method whose regret stays below 0.5 at an explanation score above 4. Figure 2(b) shows that this advantage holds on all four criteria: averaged over backbones, MechSim forms the outer envelope in each domain, indicating that its selections are not only closer to the post-cutoff optimum but also better justified. Full results with standard deviations are in Appendix C.3.

5.2 Policy Selection Results

Figure 2(c) plots F1@k against explanation score for the same runs. MechSim again clusters in the upper-right corner, reaching F1 above 0.65 with explanation scores between 4.2 and 4.9 in all three domains, whereas the baselines with comparable F1, such as Logic-LM on Measles, do so with noticeably lower explanation scores. Figure 2(d) confirms that the gain is uniform across criteria rather than driven by a single dimension: MechSim scores highest on completeness, mechanistic consistency, scientific soundness, and faithfulness in every domain. Together, these results suggest

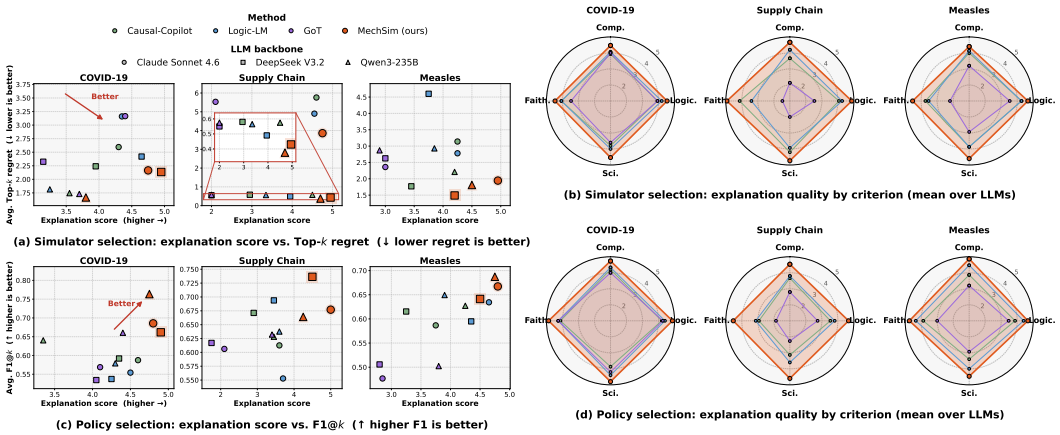


Figure 2: Task performance and explanation quality across three domains. (a, c) Explanation score (mean of four criteria) and task metric for each method–LLM pair. (b, d) Per-criterion explanation scores averaged over LLM backbones.

Table 1: Human expert evaluation scores using Claude Sonnet 4.6.

Method	**COVID-19**					**Supply Chain**					**Measles**				
	Comp.	Logic	Sci.	Faith.	Mean	Comp.	Logic	Sci.	Faith.	Mean	Comp.	Logic	Sci.	Faith.	Mean
Causal	3.33	2.54	2.67	3.67	3.05	3.88	3.75	**4.25**	3.88	3.94	3.50	3.50	4.00	3.50	3.63
Logic-LM	3.33	3.00	2.33	2.67	2.83	**4.38**	3.88	3.63	4.00	3.97	3.50	3.50	4.00	3.50	3.63
GoT	3.00	3.12	3.67	3.00	3.20	4.00	3.75	4.00	**4.38**	4.03	**4.00**	**4.32**	4.10	**4.00**	4.11
MechSim	**3.86**	**3.54**	**3.92**	**3.88**	**3.80**	4.13	**4.32**	**4.25**	4.25	**4.24**	**4.00**	4.00	**4.50**	**4.00**	**4.13**

that grounding the ranking in mechanism paths improves both the decision and the explanation that supports it. Full results with standard deviations are in Appendix C.2.

5.3 Explanation Evaluation Results

We conducted an independent human evaluation with 23 human domain experts, including 15 in epidemiology and 8 in supply chain. The experts evaluated explanations generated using Claude Sonnet 4.6 without relying on LLM judges. As shown in Table 1, MechSim achieves the highest mean score across all three domains and leads or ties on most metrics. We also have a LLM judge evaluation in Appendix: Table C.4 evaluates explanation quality using LLM judge across four dimensions: completeness, mechanism-logic consistency, scientific soundness, and faithfulness. MechSim obtains the best score in $31/36$ cells.

5.4 Simulation Reasoning Graph Evaluation

Since all downstream reasoning is constrained by $\mathcal{G}_i$, we asked 23 domain experts to rate the constructed graphs on a 1–5 scale, following prior evaluations of LLM-generated causal diagrams and knowledge graphs (Hosseinichimeh et al., 2024; Kıcıman et al., 2024). We assess *overall correctness*, *variable coverage*, and *relationship correctness*, measuring the fidelity of the simulator's mechanisms, coverage of variables and policy levers, and correctness of graph edges, respectively. As shown in Table 2, all three scores exceed 4 across domains, supporting the reliability of $\mathcal{G}_i$ for mechanism reasoning.

5.5 More Experimental Results

To connect with the motivation in the introduction, we ablate mechanism-based reasoning. This variant uses a coding agent that generates query-specific arguments rather than reasoning over a verified representation.Averaged across domains and metrics, the full MechSim outperforms this variant on all three tasks: 0.70 vs. 0.53 in P/R@k, 2.66 vs. 4.40 in Top-k regret, and 4.4 vs. 3.6 in explanation score. Additional results, including the full ablation, graph reliability and perturbation analysis, statistical significance tests, and standard deviations, are provided in Appendix C.

Table 2: Human expert evaluation of automatically constructed Simulation Reasoning Graphs.

Domain	Backbone	Simulator	Overall Correctness	Variable Coverage	Relationship Correctness
COVID-19	Claude Sonnet 4.6	SEIR-HD	4.12	4.53	4.56
Measles	DeepSeek V3.2	SVEIR	4.50	4.89	4.45
Supply Chain	Qwen 3-235B	Beer Game	4.35	4.25	4.50

6 Related Work

LLMs for Scientific Simulation and Analysis. Recent studies increasingly apply LLMs to scientific simulation and analysis (Lu et al., 2026; Baek et al., 2025). GenSim exploits LLMs' grounding and code-generation abilities to construct simulation environments and expert demonstrations (Wang et al., 2024). G-Sim presents a hybrid approach that integrates LLM-driven structural design with empirical calibration for simulator development (Holt et al., 2025). 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 (Zečević et al., 2023).

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 (Kon et al., 2025). In public health, agentic systems have been successfully deployed to automate epidemiological model construction (Datta et al., 2026) and act as autonomous policymakers to optimize intervention strategies (Aoki & Ghaffarzadegan, 2026). The AI Scientist introduces an end-to-end pipeline capable of independent idea generation, experimental execution, and automated manuscript preparation (Lu et al., 2026). Although these systems automate substantial portions of scientific workflows, they do not explicitly support comparative, mechanism-level reasoning over heterogeneous executable simulators.

Neurosymbolic Reasoning with LLMs. Neurosymbolic methods combine neural models with symbolic structure so that reasoning steps are constrained and checkable (Garcez & Lamb, 2023). With LLMs, the common recipe is to translate a problem into a formal representation and delegate inference to an external engine. Logic-LM and LINC map problems to logic programs or first-order logic and call symbolic solvers (Pan et al., 2023; Olausson et al., 2023). SatLM encodes constraints declaratively and uses a SAT solver (Ye et al., 2023). These methods improve faithfulness on logical and arithmetic tasks, but their symbolic component is a static formula or program.

7 Conclusion

This paper argued that the obstacle to LLM reasoning about scientific simulators is not access to their code or outputs but the form the reasoning takes: a query-specific argument whose claims cannot be checked, reused, or audited. MechSim addresses this by representing each simulator as a Simulation Reasoning Graph that reconciles its variables, assumptions, mechanisms, and execution traces, and by constraining LLM agents to build explanations as paths over that graph, grounded in scientific evidence and tested through sensitivity analysis on the executable simulator. Across three domains, three decision tasks, and three LLM backbones, this yields explanations that domain experts rate above those of causal, logical, and graph-based baselines and, in most settings, better simulator and policy selection. The exceptions, where a symbolic solver matches MechSim on rankings that follow directly from output magnitudes, indicate when the additional machinery is unnecessary.

Two limitations bound these claims. First, we evaluate on scientific simulators with explicit compartments and transitions, where mechanisms can be read from equations and code; simulators whose dynamics are learned, large agent-based models, or systems spread across many coupled modules would require SRG construction methods we have not developed. Second, decision tasks are evaluated offline against oracle simulators and fixed candidate ensembles, which leaves open how the framework performs when ground truth is unavailable and the ensemble changes. Extending MechSim to large coupled systems such as climate or infrastructure models, attaching uncertainty estimates to mechanism-level claims, and reducing the cost of verification for time-critical decisions are the directions we consider most important.

Ethics Statement

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.

AI Use Statement

Generative AI tools were used to assist with polishing the writing of this paper and with code implementation. All scientific ideas, experimental design, and reported results are the authors' own. The authors reviewed all AI-assisted text and code and take full responsibility for the content of the manuscript.

References

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

Goshi Aoki and Navid Ghaffarzadegan. AI agents as policymakers in simulated epidemics. *arXiv preprint arXiv:2601.04245*, 2026.

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

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 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)*, pp. 6709–6738, 2025.

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

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

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

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

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

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

Centers for Disease Control and Prevention. Cdc covid-19 travel-associated infections and diseases. https://www.cdc.gov/yellow-book/hcp/travel-associated-infections-diseases/covid-19.html, 2024. Accessed: 2026-05-06.

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

Taha Choukhmane, Tim de Silva, Weidong Lin, and Matthew Akuzawa. AI financial advice: Supply, demand, and life cycle implications. *Demand, and Life Cycle Implications (March 19, 2026)*, 2026.

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

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. The united states covid-19 forecast hub dataset. *Scientific data*, 9(1):462, 2022.

Rituparna Datta, Zihan Guan, Baltazar Espinoza, Yiqi Su, Priya Pitre, Srini Venkatramanan, Naren Ramakrishnan, and Anil Vullikanti. Agentic framework for epidemiological modeling. In *Forty-third International Conference on Machine Learning*, 2026.

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

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

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

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. *Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand*, volume 16. Imperial College London London, 2020.

Artur d'Avila Garcez and Luis C Lamb. Neurosymbolic ai: the 3 rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, 2023.

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

Gilberto González-Parra, Annika Vestrand, and Remy Mujynya. Modeling and characterizing the growth of the texas–new mexico measles outbreak of 2025. *Epidemiologia*, 6(4):60, 2025.

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

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

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. Temporal dynamics in viral shedding and transmissibility of covid-19. *Nature medicine*, 26(5):672–675, 2020.

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

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

Samuel Holt, Max Ruiz Luyten, et al. G-Sim: Generative simulations with large language models and gradient-free calibration. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Niyousha Hosseinichimeh, Aritra Majumdar, Ross Williams, and Navid Ghaffarzadegan. From text to map: a system dynamics bot for constructing causal loop diagrams. *System Dynamics Review*, 40(3):e1782, 2024.

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. Evaluation of the us covid-19 scenario modeling hub for informing pandemic response under uncertainty. *Nature communications*, 14(1):7260, 2023.

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

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

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

William Ogilvy Kermack and Anderson G McKendrick. 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):700–721, 1927.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2024.

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

Jack PC Kleijnen. Verification and validation of simulation models. *European journal of operational research*, 82(1):145–162, 1995.

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

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

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

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

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

Tianmi Ma, Jiawei Du, Wenxin Huang, Wenjie Wang, Liang Xie, Xian Zhong, and Joey Tianyi Zhou. Agent trading arena: A study on numerical understanding in llm-based agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5496–5514, 2025.

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

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

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

Madhav Marathe and Anil Kumar S Vullikanti. Computational epidemiology. *Communications of the ACM*, 56(7):88–96, 2013.

Nina B Masters. 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.

Derek Meyer and George Vega Yon. epiworldr: Fast agent-based epi models. *The Journal of Open Source Software*, 8(90), oct 2023. doi: 10.21105/joss.05781.

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

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. Projecting hospital utilization during the covid-19 outbreaks in the united states. *Proceedings of the National Academy of Sciences*, 117(16):9122–9126, 2020.

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

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5153–5176, 2023.

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

Evan L Porteus. *Foundations of stochastic inventory theory*. Stanford University Press, 2002.

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

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

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. Five ways to ensure that models serve society: a manifesto. *Nature*, 582(7813):482–484, 2020.

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

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

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

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

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

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

Dhorasso Temfack and Jason Wyse. Sequential monte carlo squared for online inference in stochastic epidemic models. *Epidemics*, pp. 100847, 2025.

Stephen E Toulmin. *The uses of argument*. Cambridge university press, 2003.

Adelinde M Uhrmacher, Peter Frazier, Reiner Hähnle, Franziska Klügl, Fabian Lorig, Bertram Ludäscher, Laura Nenzi, Cristina Ruiz-Martin, Bernhard Rumpe, Claudia Szabo, et al. Context, composition, automation, and communication: The c2ac roadmap for modeling and simulation. *ACM Transactions on Modeling and Computer Simulation*, 34(4):1–51, 2024.

George Vega Yon. *measles: Measles Epidemiological Models*, 2026. URL `https://github.com/UofUEpiBio/measles`. R package version 0.3.1-0.

Stefan Wahl, Raphaela Schenk, Ali Farnoud, Jakob H Macke, and Daniel Gedon. A probabilistic framework for llm-based model discovery. *arXiv preprint arXiv:2602.18266*, 2026.

Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

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

World Health Organization. Who covid-19 dashboard. `https://data.who.int/dashboards/covid19`, 2026. Accessed: 2026-05-06.

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

Haozhou Xu, Dongxia Wu, Matteo Chinazzi, Ruijia Niu, Rose Yu, and Yian Ma. SimulRAG: Simulator-based RAG for grounding LLMs in long-form scientific QA. In *Third Conference on Language Modeling*, 2026.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. Satlm: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems*, 36:45548–45580, 2023.

Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. Causal parrots: Large language models may talk causality but are not causal. *arXiv preprint arXiv:2308.13067*, 2023.

APPENDIX CONTENTS

A Dataset and Environment Details **16**

A.1 Datasets . . . 16

A.2 Simulator List . . . 17

A.3 Oracle Models . . . 18

A.4 Intervention Policy Lists . . . 20

B Experiment Details **21**

B.1 Baseline Method Details . . . 21

B.2 Calibration Methods . . . 21

B.3 Parameter Settings . . . 23

C Additional Experimental Results **23**

C.1 Ablation Study Results . . . 23

C.2 Comprehensive Policy Selection Task Result . . . 23

C.3 Comprehensive Simulator Selection Task Result . . . 24

C.4 Statistical Significance Analysis . . . 25

C.5 General LLM Judge Evaluation Result . . . 26

C.6 LLM Judge Evaluation for Simulator Selection Task . . . 27

C.7 LLM Judge Evaluation for Policy Selection Task . . . 27

C.8 Simulation Reasoning Graph Reliability and Perturbation Analysis . . . 28

D Example in our method **29**

D.1 Example Simulation Reasoning Graph . . . 29

D.2 Example about MechSim reasoning process . . . 29

D.3 Example Context Information . . . 29

D.4 Example Explanation Output . . . 30

E Prompt Templates **32**

E.1 Context Understanding Agent . . . 33

E.2 Mechanism Modeling Agent . . . 33

E.3 Mechanism Reasoning Agent . . . 34

E.4 Policy Selection Prompt . . . 35

E.5 Verification Agent Prompt . . . 35

E.6 LLM Judge Prompt . . . 36

F Simulation Results **37**

F.1 COVID-19 Forecast Task. . . . 37

F.2 Supply Chain Policy Selection Task. . . . 37

F.3 Sensitivity Analysis Results . . . 37

A Dataset and Environment Details

A.1 Datasets

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

Table A.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.	Dong et al. (2020)
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.	Makridakis et al. (2022)
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.	Masters (2024); Allotey (2017); González-Parra et al. (2025)

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 Cramer et al. (2022). For the forecast task, the pre-cutoff period is used to calibrate each simulator via particle filtering Temfack & Wyse (2025), 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 A.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 Grenfell et al. (2001).

Table A.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

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 A.3The 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 Temfack & Wyse (2025), and the post-cutoff period serves as the held-out

Table A.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.	Kermack & McKendrick (1927); Hethcote (2000)
SEIRD	S, E, I, R, D	Exposed latency ($\sigma = 1/$incubation) and disease-induced mortality (μ); homogeneous mixing.	Hethcote (2000)
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.	Ferguson et al. (2020); Moghadas et al. (2020)
SEIQRD	S, E, I, Q, R, D	Explicit quarantine compartment (Q); quarantine rate q directly reduces effective transmission.	Tang et al. (2020)
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.	He et al. (2020)
SIRV	S, I, R, V, D	Vaccination compartment (V) with daily coverage rate ν; vaccinated individuals fully protected.	Diekmann et al. (2013)

A.2.2 Supply Chain Simulators

Four supply chain simulators are used in the supply chain domain (Table A.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.

Table A.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.	Sterman (1989)
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.	Porteus (2002)
(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.	Scarf (1960)
Multi-Echelon (ME)	Decentralised three-level chain with independent (r, Q) policies at each level; deterministic lead times. Captures bullwhip amplification across echelons.	Porteus (2002)

A.2.3 Measles Simulators

Five compartmental baseline models are used as candidate simulators for the measles environment (Table A.5), implemented via the epydemix platform Gozzi et al. (2025) and calibrated with Approximate Bayesian Computation (ABC) (Beaumont et al., 2002).

Table A.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.	Kermack & McKendrick (1927); Hethcote (2000)
SEIR	S, E, I, R	Explicit latent period (σ); captures the ∼8–12 day measles incubation.	Diekmann et al. (2013)
SIRV	S, I, R, V	Single-dose vaccination (ν); susceptibles move to protected class.	Diekmann et al. (2013)
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.	Diekmann et al. (2013)
SVVIR	S, V_1, V_2, I, R	Compressed two-dose structure without explicit exposed state; used when latent-period data are unavailable.	Bauch & Earn (2004)

A.3 Oracle Models

A.3.1 COVID-19 Oracle

The COVID-19 Oracle is a stochastic, multi-compartment discrete-time simulator Kain et al. (2021) 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 scheme with sub-day step $\Delta t = 1/6$:

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

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.

$$\text{Harm} = C_{\text{death}} \cdot D_{\text{total}} + C_{\text{overload}} \cdot H_{\text{overload}} + \lambda_{\text{policy}} \cdot c_{\text{policy}} \cdot T, \tag{2}$$

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 Chu et al. (2015) 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.

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 $\geq 95\%$ 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 (Meyer & Vega Yon, 2023; Vega Yon, 2026), matching the oracle definition used in the main benchmark description. For policy evaluation, we use precomputed 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 (Brauner et al., 2021; Moore et al., 2021).

Table A.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 ($\times 0.95$ contact).
A3	Strict Lockdown	0.80	0.7	Full contact reduction ($\times 0.25$) 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 ($\times 0.65$); no isolation.
A6	Testing + Isolation	0.50	1.1	SIP ($\times 0.55$) + isolation of mild ($\times 0.15$) and severe ($\times 0.08$) cases.
Stress-test policies:				
A7	Reduce Trans. + Hospital Overload	0.35	1.2	Transmission $\downarrow$ but Oracle raises hospitalisation fraction (δ: $0.12 \rightarrow 0.20$).
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.

A.4.2 SUPPLY CHAIN POLICIES

Table A.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.

A.4.3 MEASLES POLICIES

Table A.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. Since this is a new problem, to the best of our knowledge, no prior work has studied it, and no domain-specific baseline is available for our setting. EpiAgent (Datta et al., 2026) is the closest related work, but it does not provide code for reproducing results, so we use general-purpose baselines for reasoning and decision-making instead.

Causal-Copilot (Wang et al., 2025). 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 Simulation Reasoning Graphs.

Logic-LM (Pan et al., 2023). 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) (Besta et al., 2024). 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 B.1); parameters are set from published epidemiological estimates. We sampled the parameters across 100 scenarios to test the robustness of our method.

Table B.1: 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: World Health Organization (2026) and Centers for Disease Control and Prevention (2024).

COVID-19 forecast task. Each ODE simulator is calibrated to observed US weekly death data using particle filtering Temfack & Wyse (2025). 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 (Makridakis et al., 2022); *(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) (Beaumont et al., 2002) using `epydemix`'s `ABCSampler`. Candidate simulators are SIR, SEIR, SIRV, SVEIR, and SVVIR, implemented with Epydemix (Kuddus et al., 2021). 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

Table B.2: 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 (Asai et al., 2026))
	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	$15\times$ (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

C Additional Experimental Results

C.1 Ablation Study Results

Figure C.1 shows the ablation results for the two core module groups across all three experiments. Both are important: removing either group degrades every metric in every domain.

For the two decision tasks, mechanism reasoning has the larger impact. In policy selection (Figure C.1a), removing it reduces ranking quality by 24%, compared with 16% for context and knowledge. In simulator selection (Figure C.1b), it increases Top-k regret by 69%, compared with 32%. Notably, "w/o Mechanism Reasoning" directly feeds context and retrieved evidence to the LLM without an explicit simulation graph, which is a common setup for LLM-based agents. Its weaker performance suggests that context and evidence alone are not sufficient; the gains come from reasoning over the explicit mechanism graph.

The pattern reverses for explanation evaluation (Figure C.1c). Removing context understanding and knowledge retrieval reduces scores by 39%–63% (mean 49%), with faithfulness affected most, while removing mechanism reasoning reduces scores by about 10%–24%. This reflects the different roles of the two module groups: retrieved evidence supports trustworthy and faithful explanations, while the explicit mechanism graph is more important for distinguishing candidate policies and simulators.

C.2 Comprehensive Policy Selection Task Result

We put the Comprehensive Policy Selection Task Result with standard deviation here in C.1. Table C.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 backbones: MechSim is best in $11/12$ Claude entries, $10/12$ DeepSeek entries, and Qwen entries.

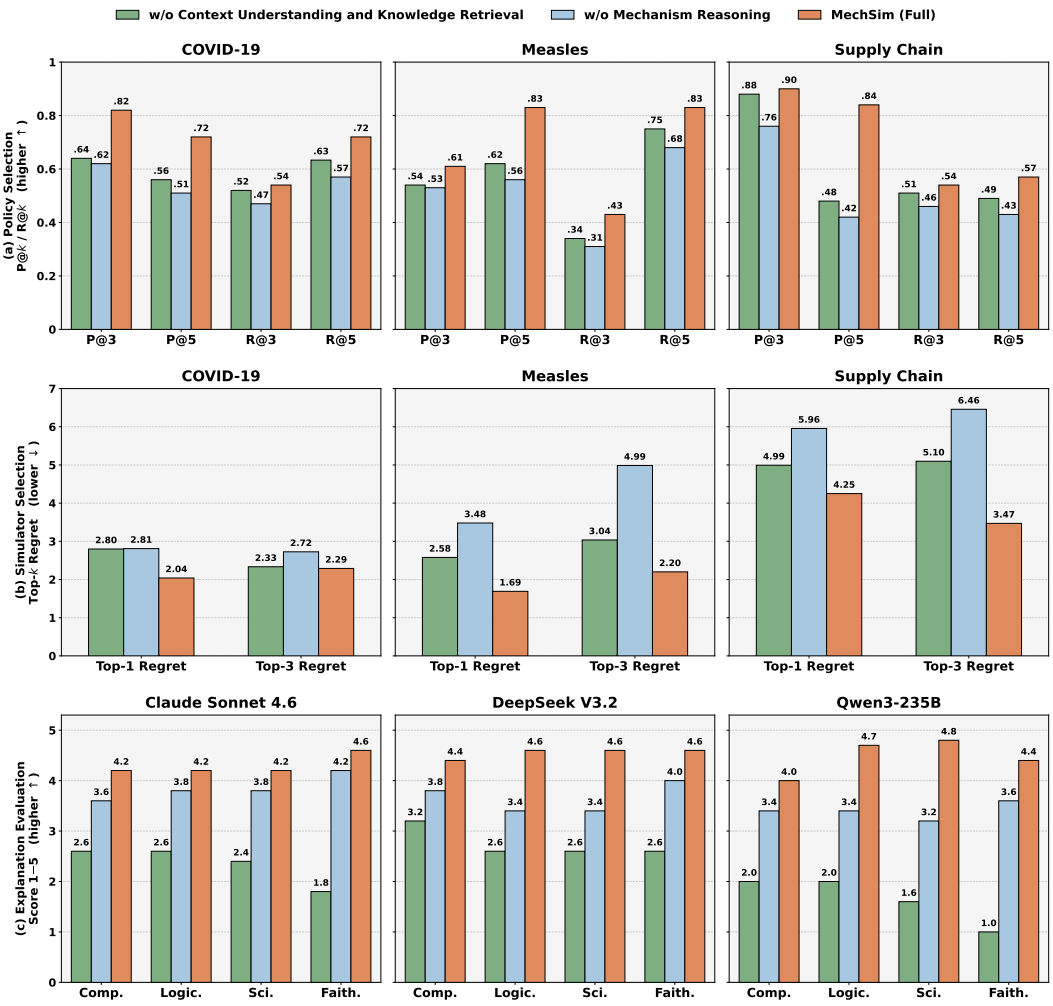


Figure C.1: Ablation results for all three experiments. "w/o" denotes removal of the module. (a) Policy selection and (b) simulator selection, both on the Claude Sonnet 4.6 backbone across the three domains. (c) Explanation evaluation on COVID-19 across three LLM backbones, scored 1–5 on completeness, mechanism-logic consistency, scientific soundness and faithfulness. Higher is better in (a) and (c); *lower* is better in (b), where the metric is Top-k regret.

On Measles, while MechSim performs well in most cases, Logic-LM remain competitive on several entries. This may be because Logic-LM uses simulator outputs as facts and domain priors as Horn rules, allowing a deterministic solver to directly derive the ranking. This works when the ranking follows from output magnitudes or simple relations, such as the clear inclusion order among Measles policies, where one policy combines the interventions of the other two.

C.3 Comprehensive Simulator Selection Task Result

We put the Comprehensive Simulator Selection Task Result with standard deviation here C.2.

Table C.2 evaluates simulator selection for forecasting using Top-k regret, where lower regret indicates a closer match 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. In the two settings where MechSim

Table C.1: 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	$\mathbf{0.778}_{\pm.369}$	$\mathbf{0.833}_{\pm.180}$	$0.433_{\pm.139}$	$\mathbf{0.833}_{\pm.153}$

is not best, it trails Causal-Copilot by only 0.01 on COVID-19 Top-3 regret and 0.15 on Measles Top-1 regret with DeepSeek V3.2.

C.4 Statistical Significance Analysis

To assess whether the observed gains are consistent across scenarios rather than driven by random variation, we compare MechSim with each baseline using paired per-scenario results on both decision tasks and all three domains. For policy selection we test Precision@3; for simulator selection we test

Table C.2: 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**

normalized Top-3 regret. For each task and domain, we apply the Wilcoxon signed-rank test against each of the three baselines. We additionally report 95% bootstrap confidence intervals for the mean paired improvement, and significance is determined by the adjusted p-value at $\alpha = 0.05$. For higher-is-better metrics, the improvement is defined as $\Delta = MechSim -$ Baseline; for lower-is-better metrics, as $\Delta =$ Baseline $- MechSim$. A positive Δ therefore always favors MechSim.

Table C.3 reports all 18 paired comparisons. MechSim achieves a positive mean improvement in every case, with 14 of 18 significant; all comparisons against Causal-Copilot and most against GoT are significant. The largest gains occur on Measles policy selection ($p < 10^{-5}$) and Supply Chain simulator selection ($p < 0.01$).

Four comparisons are not significant, and in three of them the competitive baseline is Logic-LM. Logic-LM translates simulator outputs into facts and domain priors into Horn rules, then derives pairwise preferences with a deterministic solver that defines the ranking directly. This suffices when the correct ranking follows from output magnitudes alone: supply chain policies differ mainly in how much they scale a single parameter, and measles policies form a clear inclusion order. The two methods differ in what they constrain. Logic-LM enforces logical consistency among output-level statements, whereas MechSim restricts reasoning to mechanism paths in the simulator and verifies each claim. This matters when policies act through different mechanisms, or when the decision scenerio is complex, which is where MechSim's gains are largest and significant. Although several results are not statistically significant, the overall results demonstrate clear advantages over the other baselines.

C.5 General LLM Judge Evaluation Result

Table C.4 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

Table C.3: Paired statistical significance tests of MechSim against each baseline on policy selection and simulator selection in 100 scenario, using Claude Sonnet 4.6. Δ is the mean paired improvement, oriented so that positive values favor MechSim; CIs are 95% bootstrap intervals. ✓ marks $p < 0.05$.

Domain	Baseline	Δ	95% CI	p value	Sig.
Policy Selection (Precision@3 ↑)					
COVID-19	Causal-Copilot	0.230	[0.140, 0.330]	1.6×10^{-4}	✓
	Logic-LM	0.100	[0.010, 0.200]	0.173	–
	GoT	0.180	[0.070, 0.290]	0.010	✓
Supply Chain	Causal-Copilot	0.060	[0.027, 0.093]	0.016	✓
	Logic-LM	0.033	[−0.007, 0.073]	0.527	–
	GoT	0.053	[0.027, 0.087]	0.023	✓
Measles	Causal-Copilot	0.370	[0.250, 0.490]	4.4×10^{-7}	✓
	Logic-LM	0.102	[0.064, 0.140]	5.6×10^{-6}	✓
	GoT	0.327	[0.257, 0.393]	1.0×10^{-11}	✓
Simulator Selection (normalized Top-3 regret ↓)					
COVID-19	Causal-Copilot	0.086	[0.066, 0.095]	7.1×10^{-7}	✓
	Logic-LM	0.048	[0.012, 0.084]	0.035	✓
	GoT	0.018	[0.014, 0.050]	0.274	–
Supply Chain	Causal-Copilot	0.317	[0.187, 0.449]	0.006	✓
	Logic-LM	0.317	[0.153, 0.479]	0.009	✓
	GoT	0.286	[0.161, 0.419]	0.009	✓
Measles	Causal-Copilot	0.063	[0.020, 0.107]	0.035	✓
	Logic-LM	0.050	[0.003, 0.103]	0.135	–
	GoT	0.067	[0.023, 0.110]	0.035	✓

Table C.4: Explanation evaluation across domains. Metrics: Completeness (Comp.), Mechanism Logic Consistency (Logic.), Scientific Soundness (Sci.), and Faithfulness (Faith.).

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

criterion for all three LLM backbones. In COVID-19, MechSim remains competitive but is usually second on completeness and faithfulness.

C.6 LLM Judge Evaluation for Simulator Selection Task

Here in table C.5, we use an LLM judge to evaluate the explanations for the simulator selection task across the four metrics described above.

C.7 LLM Judge Evaluation for Policy Selection Task

Here in table C.6, we use an LLM judge to evaluate the explanations for the policy selection task across the four metrics described above.

Table C.5: Evaluation of simulator-selection explanations across domains. Metrics: Completeness (Comp.), Mechanism Logic Consistency (Logic.), Scientific Soundness (Sci.), and Faithfulness (Faith.). **Bold**: best; underline: second best.

LLM	Method	COVID-19				Supply Chain				Measles			
		Comp.	Logic.	Sci.	Faith.	Comp.	Logic.	Sci.	Faith.	Comp.	Logic.	Sci.	Faith.
Claude Sonnet 4.6	Causal	4.2	4.2	4.0	4.8	4.6	4.6	4.4	**4.8**	4.2	4.4	4.4	4.0
	Logic	4.4	**4.6**	4.0	4.4	4.6	4.6	**4.6**	4.4	4.4	4.4	4.4	3.8
	GoT	**4.6**	4.4	4.2	4.4	2.2	2.4	2.0	1.8	3.2	3.4	3.2	2.2
	MechSim	**4.6**	**4.6**	**4.8**	**5.0**	**4.8**	**4.8**	**4.6**	**4.8**	**4.8**	**5.0**	**5.0**	**5.0**
DeepSeek V3.2	Causal	3.8	3.8	4.0	4.2	2.6	3.2	3.4	2.6	3.8	4.0	3.0	3.0
	Logic	4.6	4.4	4.8	4.8	4.2	4.2	3.8	3.6	3.8	4.2	3.6	3.4
	GoT	3.2	3.4	3.0	3.0	2.0	2.4	2.0	1.6	3.2	3.8	2.8	2.2
	MechSim	**4.8**	**5.0**	**5.0**	**5.0**	**4.8**	**5.0**	**5.0**	**5.0**	**4.0**	**4.4**	**4.4**	**4.0**
Qwen 3-235B	Causal	3.8	3.8	3.4	3.2	3.8	4.4	**4.8**	**5.0**	4.0	4.4	4.2	4.2
	Logic	3.2	3.6	3.2	3.0	3.8	4.0	3.4	2.2	4.2	4.2	3.6	3.4
	GoT	**4.2**	**4.0**	3.6	3.0	2.2	2.8	2.0	1.0	3.2	3.4	2.8	2.2
	MechSim	4.0	**4.0**	**3.8**	**3.4**	**4.4**	**4.8**	4.6	**5.0**	**4.4**	**4.6**	**4.4**	**4.6**

Table C.6: Evaluation of policy-selection explanations across domains.

LLM	Method	COVID-19				Supply Chain				Measles			
		Comp.	Logic.	Sci.	Faith.	Comp.	Logic.	Sci.	Faith.	Comp.	Logic.	Sci.	Faith.
Claude Sonnet 4.6	Causal	4.6	4.6	4.6	4.6	4.0	3.6	3.4	3.4	4.0	4.0	3.4	3.6
	Logic	4.4	4.6	4.6	4.4	3.8	3.8	3.8	3.4	4.6	4.6	**4.6**	**4.8**
	GoT	4.0	4.2	4.2	4.0	2.6	2.4	2.0	1.4	3.2	3.2	2.6	2.4
	MechSim	**4.8**	**4.8**	**4.8**	**4.8**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**4.8**	**4.6**	**4.8**
DeepSeek V3.2	Causal	4.4	4.4	4.2	4.4	3.2	3.4	2.8	2.2	3.2	3.6	2.8	3.4
	Logic	4.0	4.4	4.4	4.2	3.6	3.6	3.4	3.2	4.4	**4.6**	4.0	**4.4**
	GoT	4.0	4.2	4.0	4.0	2.0	2.2	1.6	1.2	2.6	3.2	2.6	2.8
	MechSim	**4.8**	**5.0**	**5.0**	**4.8**	**4.6**	**4.6**	**4.4**	**4.4**	**4.8**	**4.6**	**4.2**	**4.4**
Qwen 3-235B	Causal	3.6	3.6	2.8	3.4	3.8	3.6	3.2	3.2	4.4	4.0	4.0	4.6
	Logic	**4.6**	4.2	4.2	4.2	**4.0**	4.0	3.6	2.8	4.4	4.0	3.4	3.8
	GoT	4.0	4.4	4.4	4.8	3.8	3.6	3.2	3.0	3.8	4.0	3.6	3.8
	MechSim	**4.6**	**4.8**	**4.6**	**5.0**	**4.0**	**4.4**	**4.4**	**4.2**	**4.8**	**4.6**	**4.6**	**5.0**

C.8 Simulation Reasoning Graph Reliability and Perturbation Analysis

To evaluate the quality and correctness of the generated Simulation Reasoning Graphs, we measured the Verification Agent's error-detection accuracy and examined how graph errors affect downstream performance through controlled perturbation experiments.

Verification and correction. The Verification Agent checks each graph and its resulting explanation for structural and logical inconsistencies. Outputs that fail these checks are returned to the preceding agent for revision before the pipeline continues, limiting the propagation of graph-construction errors.

Controlled graph perturbations. We further perturb the graphs by corrupting assumptions and removing nodes or edges at mild and moderate severity levels. Table C.7 reports downstream performance on the COVID-19 policy-selection task.

Ranking performance remains stable under moderate corruption: Precision@3 and Recall@3 decrease by only 0.007 and 0.001, respectively, relative to the original graph. Faithfulness and mechanistic correctness decline more noticeably, by 0.224 and 0.373, but remain above 4.0 on average. Moreover, the Verification Agent flags 99.7% of mildly corrupted cases and 98.4% of moderately corrupted cases for correction. Together with the expert ratings, these results indicate that the generated graphs contain relatively few errors and that MechSim is robust to moderate graph-construction noise.

Table C.7: Robustness to controlled mechanism-graph perturbations. Mild and Moderate levels perturb approximately 20% and 40% of the affected graph elements, respectively. Detection denotes the percentage of corrupted cases flagged and corrected by the Verification Agent.

Corruption	Level	Precision@3	Recall@3	Detection
None	Original	0.603	0.564	–
Assumption	Mild	0.586	0.564	–
Edge removal	Mild	0.603	0.564	–
Node removal	Mild	0.602	0.564	–
Assumption	Moderate	0.585	0.563	–
Edge removal	Moderate	0.603	0.564	–
Node removal	Moderate	0.600	0.563	–
Aggregate results				
All corruptions	Mild	0.597	0.564	99.7%
All corruptions	Moderate	0.596	0.563	98.4%

D EXAMPLE IN OUR METHOD

D.1 EXAMPLE SIMULATION REASONING GRAPH

Here shows the Simulation Reasoning Graph D.2, it includes $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{M}_i)$ automatically extracted by the Mechanism Modeling Agent for the SEAIHRD simulator. Nodes represent compartments ($\mathcal{V}_i$); directed edges represent mechanistic transitions ($\mathcal{E}_i$); and policy binding shows how the policy acts in the case; also graph-level metadata ($\mathcal{M}_i$) includes simulator assumptions and execution statistics.

D.2 EXAMPLE ABOUT MECHSIM REASONING PROCESS

Figure D.3 in illustrates this process on a COVID-19 policy-selection case. Given policy a_3 (contact reduction), the agent first identifies the propagation path $a_3 \to m_C \to \lambda(t) \to S \to E \to I \to H$ in $\mathcal{G}_i$, then reasons over each edge by combining its mechanism, the activating assumption, sensitivity results, and retrieved evidence, and finally produces claims that cite the edges, assumptions, and evidence they rest on, which are aggregated into the recommendation R.

D.3 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

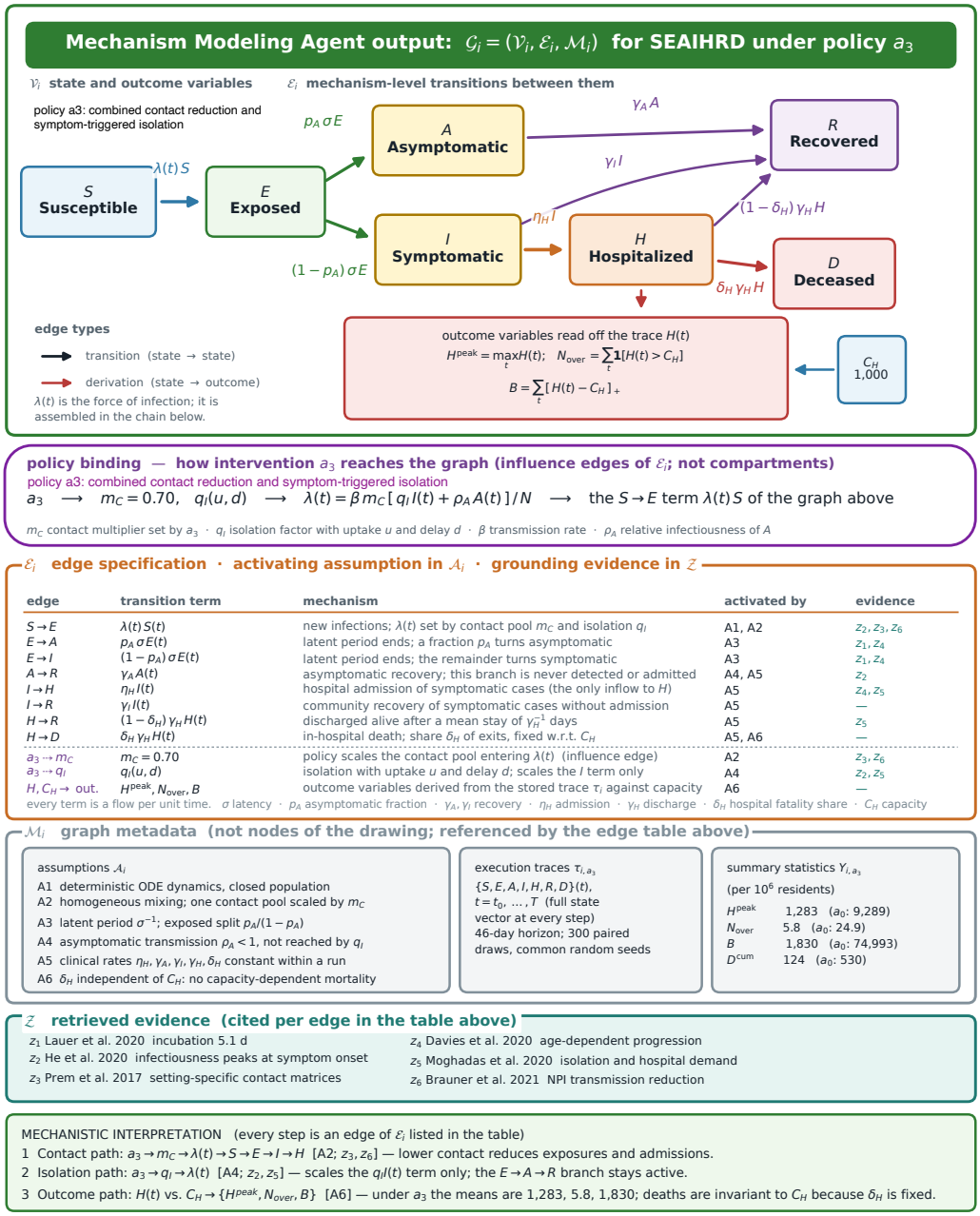


Figure D.2: Example of the Simulation Reasoning Graph

- 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.

D.4 Example Explanation Output

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

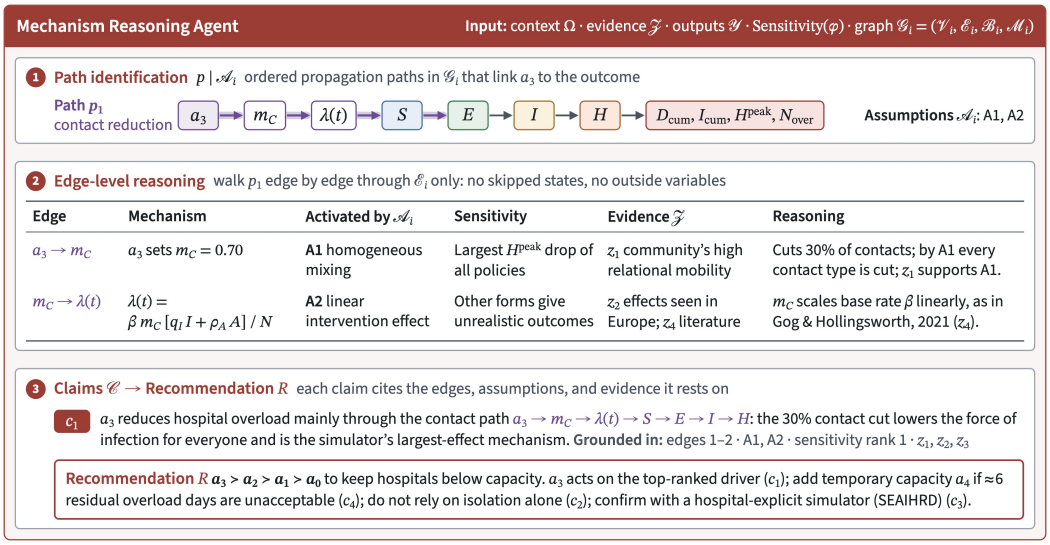


Figure D.3: Worked example of the Mechanism Reasoning Agent on COVID-19 policy selection, where policy a_3 (contact reduction) sets the contact multiplier m_C. The agent (1) identifies a propagation path p_1 from a_3 to the outcomes, (2) reasons over each edge, (3) produces grounded claims $\mathcal{C}$.

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 $4\times$ 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.

E PROMPT TEMPLATES

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

E.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`.

E.2 Mechanism Modeling Agent

Prompt: Simulation Reasoning Graph Extraction

You are an expert in scientific simulation, compartmental modeling, and causal inference. Your task is to extract a complete, structured Simulation Reasoning 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 Simulation Reasoning 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`.

E.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 Simulation Reasoning 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]}`
[Simulation Reasoning 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 Simulation Reasoning 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.

E.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, Simulation Reasoning Graphs, sensitivity analysis findings, and retrieved scientific evidence.

[Context Information] `{context_summary}`
[Candidate Policies] `{policy_list_with_descriptions}`
[Calibrated Simulator Predictions] `{simulation_result}`
[Simulation Reasoning 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 $\rightarrow$ contact rate $\beta \downarrow \rightarrow$ force of infection $\downarrow \rightarrow$ 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.

E.5 VERIFICATION AGENT PROMPT

Prompt: Verification Agent

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

[Simulation Reasoning 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$\rightarrow$S, S$\rightarrow$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.

E.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}`
[Simulation Reasoning 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 Simulation Reasoning 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).

F SIMULATION RESULTS

F.1 COVID-19 FORECAST TASK.

Table F.8 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 F.8: 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

F.2 SUPPLY CHAIN POLICY SELECTION TASK.

Table F.9 reports simulator-predicted harm scores across six candidate policies for a representative scenario (Scenario 2). The oracle identifies Policy 4 (Expedited Replenishment) as optimal.

Table F.9: 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

F.3 SENSITIVITY ANALYSIS RESULTS

Table F.10 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 F.11 summarises structural sensitivity results from assumption-level interventions.

Table F.10: 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 $\rightarrow$ higher, earlier peak
Contact rate	2.35	2	Positive	Higher contact rate $\rightarrow$ faster spread
Behavioral reduction (α)	0.85	3	Negative	Stronger NPI $\rightarrow$ lower peak
Recovery rate (γ)	0.81	4	Negative	Faster recovery $\rightarrow$ lower, later peak

Table F.11: 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	$+8.1\%$	$+31{,}100$