CAMO: An Agentic Framework for Automated Causal Discovery from Micro Behaviors to Macro Emergence in LLM Agent Simulations

Xiangning Yu[1,2,3]*, Yuwei Guo[1,2,3]*, Yuqi Hou[1,2,3], Xiao Xue[1,2,3]†, Qun Ma[1,2,3]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Tianjin Key Laboratory of Healthy Habitat and Smart Technology, Tianjin, China
[3]Laboratory of Computation and Analytics of Complex Management Systems, Tianjin University, Tianjin, China

{yxn9191, 2024244171, houyuqi, jzxuexiao, 1023244018}@tju.edu.cn

Abstract

LLM-empowered agent simulations are increasingly used to study social emergence, yet the micro-to-macro causal mechanisms behind macro outcomes often remain unclear. This is challenging because emergence arises from intertwined agent interactions and meso-level feedback and nonlinearity, making generative mechanisms hard to disentangle. To this end, we introduce **CAMO**, an automated **Ca**usal discovery framework from **Mi**cr**o** behaviors to **Ma**cr**o** Emergence in LLM agent simulations. CAMO converts mechanistic hypotheses into computable factors grounded in simulation records and learns a compact causal representation centered on an emergent target Y. CAMO outputs a computable Markov boundary and a minimal upstream explanatory subgraph, yielding interpretable causal chains and actionable intervention levers. It also uses simulator-internal counterfactual probing to orient ambiguous edges and revise hypotheses when evidence contradicts the current view. Experiments across four emergent settings demonstrate the promise of CAMO.[1]

1 Introduction

LLM-empowered agent simulations have emerged as a powerful laboratory for studying complex social phenomena (Park et al., 2023; Piao et al., 2025b). By instantiating populations of agents that make autonomous, LLM-driven decisions, these simulations generate rich, adaptive, and often nonlinear interaction patterns. Through decentralized interactions, macro-level emergent outcomes such as coordination, norm formation, and polarization naturally arise (Riedl, 2025; Chen et al., 2024; Ren et al., 2024; Takata et al., 2024; Piatti et al., 2024).

Although emergent patterns are frequently observed in such simulations, they provide limited insight into the causal mechanisms that generate them (Mou et al., 2024; Guo et al., 2024; Piao et al., 2025a). In particular, it often remains unclear how micro-level agent behaviors and meso-level interaction structures jointly give rise to a target emergent outcome, making interventions on prompts, agent policies, or interaction settings largely heuristic and difficult to generalize.

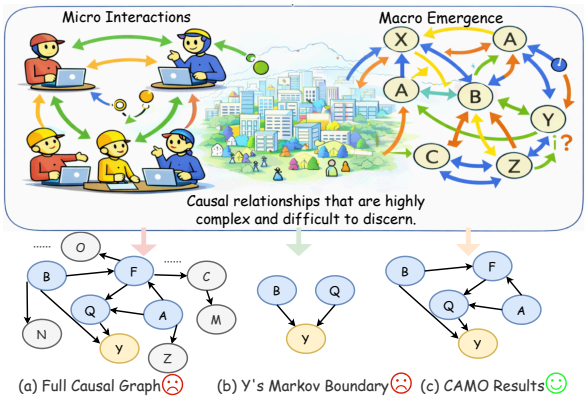


Figure 1: **Causal representations recovered by CAMO.** CAMO identifies a compact causal neighborhood around the target outcome Y that is sufficient for causal identification, and augments it with a minimal set of upstream pathways needed to explain and support intervention on micro-to-macro emergence.

Recent work has explored the integration of large language models with causal discovery and causal reasoning (Jiralerspong et al., 2024; Le et al., 2024; Takata et al., 2024; Kiciman et al., 2023; Zečević et al., 2023), but these methods are largely developed for static variable spaces and do not address how causal mechanisms produce emergent outcomes in adaptive multi-agent systems. In parallel, LLM-empowered agent simulations have been widely used to study emergence phenomena (Mao et al., 2025; Park et al., 2023; Ren et al., 2024; Gupta et al., 2025; Chuang et al., 2024), yet they primarily demonstrate or characterize emergent behavior rather than uncovering the causal mecha-

*Equal contribution
†Corresponding author
[1]The code is available at: https://github.com/RisingDate/CAMO.

Findings of the Association for Computational Linguistics: ACL 2026, pages 24447–24479

nisms by which it arises. Consequently, the micro-to-macro causal structures needed to explain and reliably intervene on a specific emergent outcome remain largely uncharacterized.

In this work, we ask: *given an LLM-empowered agent simulation, how can we automatically recover the causal mechanisms that link micro-level agent behaviors and meso-level interactions to an emergent macro-level outcome Y, in a way that supports both explanation and intervention?*

We introduce CAMO, a multi-agent framework for micro-to-macro causal discovery in LLM-empowered agent simulations. Rather than attempting to recover the full causal structure underlying emergence, which is often infeasible and unnecessary in complex agent-based systems, CAMO identifies a *minimal yet sufficient* set of upstream variables and causal pathways that explain a given emergent outcome Y.

The central idea of CAMO is to explicitly separate *causal identification* from *mechanistic explanation*. Causal identification focuses on recovering a minimal local causal structure around Y that supports prediction and intervention. Mechanistic explanation then traces this structure backward through upstream micro- and meso-level pathways, yielding an interpretable account of how emergence arises. This separation allows CAMO to remain statistically grounded while preserving clear micro-to-macro causal narratives.

Operationally, CAMO treats the LLMs driving agent behavior inside the simulation as a black-box stochastic data-generating process. A separate set of *workflow LLMs* operates outside the simulation to interpret textual descriptions, propose and revise mechanistic hypotheses, and coordinate discovery. Candidate causal relations are admitted, pruned, and oriented using constraints from observational records and targeted simulator-internal counterfactual interventions, organized through a fast–slow self-evolution loop.

Our contributions are threefold:

- We frame micro-to-macro causal discovery in LLM-empowered agent simulations as the problem of identifying a minimal but sufficient upstream causal explanation for an emergent outcome.
- We propose CAMO, the first multi-agent framework for studying *micro-to-macro emergence mechanisms* in LLM-based simulations that combines domain priors, observational data, and simulator-internal counterfactuals.
- We introduce a fast–slow self-evolving loop that curbs hallucinated priors and noisy interventions, improving robustness without assuming global identifiability.

2 Related Work

LLM-assisted causal discovery. Recent work explores using large language models to assist causal discovery and causal reasoning, including proposing causal relations, constraints, or latent factors to complement statistical methods (Chi et al., 2024; Jin et al., 2023; Liu et al., 2025; Kiciman et al., 2023; Wu et al., 2024; Feng et al., 2024). Several approaches integrate LLM-generated knowledge into classical structure learning pipelines under limited or noisy data (Takayama et al., 2024; Jiralerspong et al., 2024; Farooq et al., 2023). More recent systems formulate causal discovery as an agentic or tool-augmented workflow, including multi-agent refinement and autonomous causal modeling (Le et al., 2024; Liu et al., 2024; Han et al., 2024; Khatibi et al., 2025). These methods are primarily designed for fixed or predefined variable spaces and do not address causal discovery in adaptive, emergent multi-agent environments (Richens and Everitt, 2024; Yu et al., 2025a, 2018).

LLM-based simulation and emergence. LLM-empowered agent simulations have been widely used to study emergent social phenomena and collective dynamics (Mao et al., 2025; Park et al., 2023; Gao et al., 2024; Ashery et al., 2025; Anthis et al., 2025). Large-scale platforms demonstrate that thousands of interacting LLM agents can exhibit coordination, cooperation, and polarization (Gurcan, 2024; Piao et al., 2025b,a; Ren et al., 2024; Piatti et al., 2024). At the individual level, generative agent simulations validate the behavioral fidelity of LLM agents (Agrawal et al., 2024; Adornetto et al., 2025; Chuang et al., 2024; Dai et al., 2024). However, existing studies primarily describe or reproduce emergent patterns, while the underlying micro-to-macro causal mechanisms remain largely unexplored from a causal view (Zhang et al., 2025; Wang et al., 2025b; Xia et al., 2024).

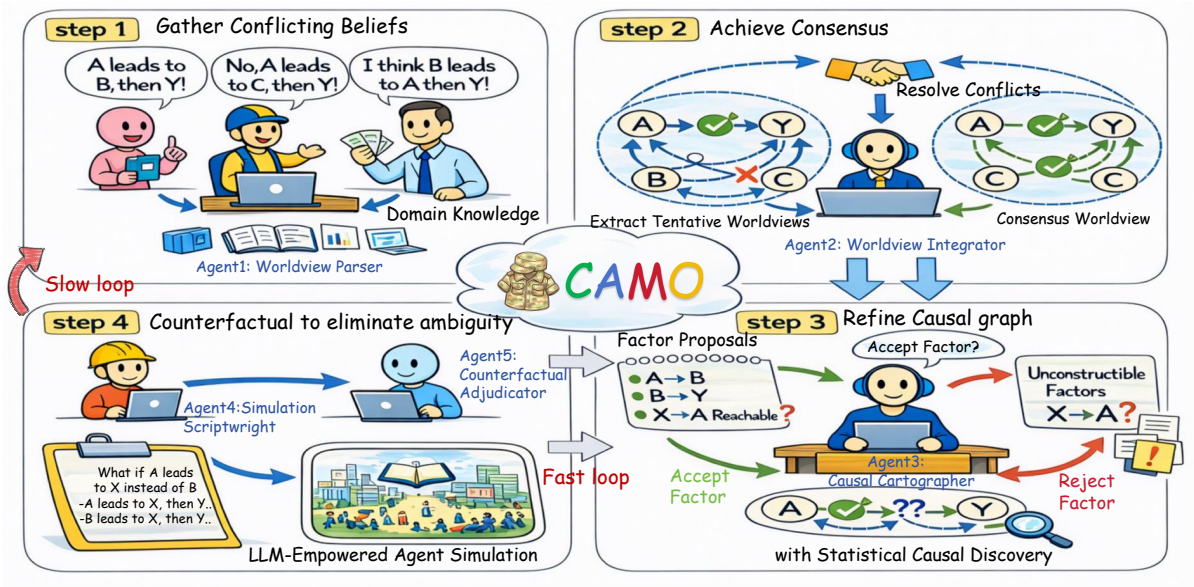


Figure 2: **Overview of CAMO.** A fast–slow loop integrates textual worldviews, causal discovery, and simulator-internal interventions to recover a minimal causal interface and micro-to-macro explanation for the target outcome.

3 Methodology

3.1 Problem Formulation and Objectives

We study causal discovery for emergent outcomes in *LLM-empowered agent simulations*, where populations of agents interact and collectively give rise to a macro-level target variable Y. Let X_{obs} denote the full set of logged simulation observables. Beyond logged signals, CAMO introduces additional variables as explicit, *computable nonlinear* transforms of X_{obs}, since many micro/meso mechanisms are not primitive logs but only become measurable through such compositions, yielding an induced factor space $\mathcal{H}$ built from a fixed set of interpretable templates (e.g., ratios, rolling statistics, and graph metrics).

Rather than recovering the full simulator causal DAG, we focus on a validated *local causal interface* around Y and a minimal mechanistic explanation that traces emergence from micro/meso-level causes to Y. Formally, CAMO outputs:

$$\Big(\mathrm{MB}^{\mathcal{H}}(Y),\ E_Y \Big). \tag{1}$$

Here, $\mathrm{MB}^{\mathcal{H}}(Y)$ denotes the Markov boundary of Y in the induced factor space, i.e., the minimal sufficient conditioning set for predicting and intervening on Y, while the explanatory subgraph E_Y augments this local causal interface with a minimal set of upstream micro- and meso-level variables and pathways, providing a mechanistic account of how emergence arises.

3.2 Overview of CAMO

CAMO is a multi-agent framework consisting of five LLM agents: the Worldview Parser (A1), Worldview Integrator (A2), Causal Cartographer (A3), Simulation Scriptwright (A4), and Counterfactual Adjudicator (A5). These agents coordinate in two loops: a fast refinement loop that proposes and prunes computable factors to fit a local causal model under a fixed hypothesis, followed by targeted interventions to resolve remaining ambiguities. A slow revision loop is triggered when interventional evidence contradicts the hypothesis, prompting hypothesis updates. Algorithm 1 summarizes the procedure; Figure 2 illustrates it.

A1: Worldview Parser. As a CAMO workflow agent, A1 converts unstructured domain knowledge into structured mechanistic causal hypotheses. Given a user query Q specifying the simulator setting and target outcome Y, A1 automatically retrieves a relevant text bundle T (e.g., domain references and background materials) and parses T into candidate variables and relations. It merges semantically equivalent mentions, tags variables by micro/meso/macro scale, and preserves conflicts from heterogeneous sources (e.g., opposite directions under different assumptions) as explicit competing alternatives. The result is a set of revisable causal worldviews that serve as priors for downstream discovery.

Algorithm 1: CAMO: Local Interface + Minimal Emergence Explanation

Input: Simulator $\mathcal{S}$, target Y, observations X_{obs}, user query Q
Output: $\text{MB}^{\mathcal{H}}(Y)$ and E_Y

A1: Retrieve domain texts T from Q; parse T into worldviews $\{W_a\}$;
repeat `// Slow loop`
  A2: Select decision worldview W and mechanism graph G_W;
  repeat `// Fast loop`
    A3: Add/prune factors $\rightarrow$ stabilize V and $\text{MB}^{\mathcal{H}}(Y)$;
    A3: Constrained discovery $\rightarrow$ CPDAG/PAG; rank ambiguities;
    A4–A5: Counterfactual queries; add confirmed constraints;
    A3: Update local graph;
  until *local graph stabilizes or counterfactuals become uninformative*;
  if *counterfactual evidence contradicts* W **then**
    Update/remove W; **continue**;
  A3: Extract E_Y from G_W s.t. $\mathcal{R} \Rightarrow \text{MB}^{\mathcal{H}}(Y)$;
until $\text{MB}^{\mathcal{H}}(Y)$ *and* E_Y *stabilize*;

A2: Worldview Integrator. A2 aligns the perspective-indexed worldviews into a shared, *computable* representation. It (i) unifies semantically equivalent variables, (ii) assigns canonical construction rules for computable factors (e.g., ratios, summary statistics, graph metrics), and (iii) makes cross-perspective conflicts explicit. A2 selects a decision worldview $W^{(k)}$ for the current refinement round (see Appendix C.2 for selection criteria), and also produces a retained *mechanism graph* $G_W^{(k)}$ that stores upstream causal pathways and alternative hypotheses for later revision.

3.3 Local Interface Learning by Refinement (A3: Causal Cartographer)

Data-grounded representation. Let $V^{(t)}$ be the representation variables retained at refinement step t, where $V^{(t)} \subseteq \mathcal{H}$ consists of variables that are either directly logged in X_{obs} or computably constructed from X_{obs}.

Add–prune refinement. Under the current decision worldview, A3 refines $V^{(t)}$ by testing each computable factor Z via its estimated information gain for Y:

$$\widehat{\Delta I}(Z) := \widehat{H}(Y \mid V^{(t)}) - \widehat{H}(Y \mid V^{(t)}, Z) \; > \; \tau_I. \tag{2}$$

That is, Z is added only if it reduces the conditional uncertainty of Y by at least τ_I. A3 then prunes any factor that becomes conditionally redundant for predicting Y given the remaining variables in $\mathcal{H}$. This add–prune loop yields a compact *computable Markov boundary* $\text{MB}^{\mathcal{H}}(Y)$; see Appendix C.1 for estimation details.

Constrained local discovery. Given the stabilized representation $V^{(t)}$, A3 performs constraint-based causal discovery under the structural constraints specified by $W^{(k)}$, selecting a suitable test/algorithm based on the data regime, and outputs a partially oriented equivalence-class graph (e.g., a CPDAG or PAG). Edges whose orientation remains ambiguous are prioritized (by estimated effect importance and orientation uncertainty) and forwarded to A4 for intervention design.

3.4 Minimal Connecting Explanatory Subgraph (A3: Causal Cartographer)

Root variables. We define root variables $\mathcal{R}$ as upstream variables in the retained mechanism graph $G_W^{(k)}$ that correspond to fundamental agent behaviors or environment settings, and serve as anchors for tracing micro-to-macro emergence.

Definition of E_Y via minimal connecting subgraph. After stabilizing the computable Markov boundary $\text{MB}^{\mathcal{H}}(Y)$ and incorporating simulation-confirmed constraints, A3 constructs an explanatory subgraph

$$\begin{aligned} E_Y \; &:= \{Y\} \cup \text{MB}^{\mathcal{H}}(Y) \\ &\quad \cup \; \text{Conn}_{\min}\Big(\mathcal{R} \Rightarrow \text{MB}^{\mathcal{H}}(Y)\Big). \end{aligned} \tag{3}$$

where $\text{Conn}_{\min}(\mathcal{R} \Rightarrow \text{MB}^{\mathcal{H}}(Y))$ denotes a connecting subgraph of $G_W^{(k)}$ that, for each boundary variable $B \in \text{MB}^{\mathcal{H}}(Y)$, retains all nodes and edges that lie on directed paths from some root $r \in \mathcal{R}$ to B, while enforcing all simulation-confirmed edges as hard constraints. It is *minimal*: removing any retained node/edge would break the required $r \rightarrow B$ reachability for some $B \in \text{MB}^{\mathcal{H}}(Y)$. Computation details are in Appendix G.

Intuitively, $\text{MB}^{\mathcal{H}}(Y)$ provides a minimal *local causal interface* sufficient for identifying and intervening on Y, while E_Y augments this interface

with the minimal set of upstream micro- and meso-level causal pathways needed to explain how the emergent outcome arises.

3.5 Interventions and Counterfactual Evidence (A4–A5)

A4: Simulation Scriptwright. A4 converts prioritized ambiguous edges into executable intervention scripts when the simulator admits a realizable control (e.g., prompt/policy override or environment parameter change); prioritization uses a computable importance–uncertainty score (Appendix C.3). Each script specifies the intervention endpoint, levels, paired-rollout protocol holding other randomness fixed, and replication budget across configurations.

A5: Counterfactual Adjudicator. A5 executes paired interventions (e.g., $\mathrm{do}(X{=}x)$ vs. $\mathrm{do}(X{=}x')$) to obtain simulator-internal counterfactual contrasts. An edge is marked *simulation-confirmed* if effects are consistently nonzero and orientations are consistent across configurations. Confirmed edges are treated as strong constraints in subsequent fast-loop updates.

4 Theoretical Analysis

This section analyzes theoretical properties of CAMO (§ 3). Proofs are given in Appendix A.

4.1 Convergence to a Computable Markov Boundary

Let $V^{(t)}$ denote the representation variables available at refinement step t, including both observed variables and computably constructed factors in the induced factor space $\mathcal{H}$. Let B_t denote the Markov boundary of Y within $V^{(t)}$ (Koller and Friedman, 2009).

Theorem 4.1 (One-step compactness of boundary refinement)**.** *Under the add–prune refinement procedure of Agent A3, suppose refinement step $t \to t{+}1$ admits a (possibly empty) set of new factors $\mathcal{Z}_{t+1}$ with $m_t := |\mathcal{Z}_{t+1}|$. Then the Markov boundary size satisfies*

$$|B_{t+1}| \le |B_t| + m_t, \tag{4}$$

with strict decrease possible whenever some previously boundary variable is pruned as conditionally redundant given the remaining variables.

Following Liu et al. (2024), we characterize convergence beyond boundary size by measuring the remaining unexplained dependence between Y and the full observation space:

$$F_t := I(Y; X_{\mathrm{obs}} \mid V^{(t)}). \tag{5}$$

Theorem 4.2 (Geometric decay under refinement capability)**.** *Assume that with probability at least $p > 0$, a refinement step admits at least one informative computable factor that reduces the residual dependence, and that conditioned on such a step the reduction is by a constant fraction $C > 0$, i.e.,*

$$F_{t+1} \le (1 - C)F_t, \tag{6}$$

and otherwise $F_{t+1} \le F_t$. Then the expected residual dependence decays geometrically:

$$\mathbb{E}[F_t] \le (1 - pC)^t F_0. \tag{7}$$

Consequently, refinement stabilizes (up to equivalence in the induced factor space $\mathcal{H}$) on a Y-sufficient representation, which can be pruned to yield a computable Markov boundary $\mathrm{MB}^{\mathcal{H}}(Y)$.

The constants p and C are used only to characterize refinement effectiveness; they are estimated a posteriori from the $\{F_t\}$ trajectory (Appendix C.1) and are not required by the algorithm.

Theorem 4.1 ensures the computable Markov boundary cannot expand uncontrollably, while Theorem 4.2 implies refinement converges (up to equivalence in $\mathcal{H}$) to a Y-sufficient representation whose pruning yields $\mathrm{MB}^{\mathcal{H}}(Y)$.

4.2 Identifiability Gains from Simulator Counterfactuals

Constraint-based discovery on observational records identifies at most the observational equivalence class of the local causal graph, leaving some edge orientations unresolved (Spirtes et al., 2000; Zhang, 2008; Chickering, 2002).

Proposition 1 (Resolving ambiguous edges via simulator counterfactuals)**.** *Simulator-internal counterfactual contrasts can resolve some CPDAG/PAG orientation ambiguities by ruling out directions inconsistent with the induced effect, making a subset of edges identifiable.*

4.3 Minimal Yet Sufficient Emergence Explanation via E_Y

CAMO augments the computable Markov boundary $\mathrm{MB}^{\mathcal{H}}(Y)$ with upstream causal pathways to form an explanatory subgraph E_Y (Eq. (3)), enabling micro-to-macro interpretation of emergence.

Proposition 2 (Sufficiency and minimality of E_Y)**.** *The subgraph E_Y is sufficient for local prediction and intervention on Y, since it contains $\mathrm{MB}^{\mathcal{H}}(Y)$. Moreover, E_Y is minimal for mechanistic explanation: it retains exactly the causal structure required to connect each boundary variable to upstream micro/meso-level root causes, and contains no superfluous nodes or edges.*

5 Experiments

We evaluate CAMO under the following research questions:

- **RQ1.** Can CAMO recover the causal structure linking micro- and meso-level mechanisms to the macro outcome Y?
- **RQ2.** Can the learned causal representation correctly identify effective interventions, namely actions on causes of Y that reliably change Y?

5.1 Factor Discovery and Causal Structure Recovery (RQ1)

Simulation setup. We construct a benchmark consisting of an interactive LLM-empowered multi-agent simulation of an online-to-offline (O2O) delivery platform, built upon a *minimal yet sufficient* ground-truth causal structure calibrated to real-world statistics from Meituan Research (Wang et al., 2025a). This ground-truth structure is used exclusively for evaluation and is never exposed to CAMO. We quantify emergence using the indicator of Yu et al. (2025b); full simulation details see Appendix H.

Baselines. *Factor discovery.* We compare CAMO against LLM-based factor discovery baselines, including single-round text-only, data-grounded, and chain-of-thought prompting, as well as multi-round prompting without causal feedback. Details and prompts see Appendices F.1 and F.2.

Causal structure recovery. We compare CAMO against three classes of baselines. *Statistical causal discovery (SCD)* methods[2] include PC (Spirtes and Glymour, 1991), FCI (Spirtes et al., 2000), GES (Chickering, 2002), and MMHC (Tsamardinos et al., 2006), all applied directly to observed variables. *Pure LLM-based methods* include Efficient-CDLMs (Jiralerspong et al., 2024), MAC (Le et al., 2024), and PAIRWISE (Kiciman et al., 2023), which rely on LLM reasoning to infer causal relations without explicit statistical tests. *Hybrid SCD+LLM methods* combine statistical discovery with LLM-based graph refinement, including SCD-LLM, ReAct (Yao et al., 2022), and LLM-KBCI (Takayama et al., 2024). Details are provided in Appendix F.3.

Metrics. *(i) Factor discovery for the local interface and upstream explanation.* We evaluate discovered factors by comparing them with ground truth, using the target's Markov boundary $B(Y)$ and ancestor set $\mathrm{An}(Y)$ as references. Discovered factors are categorized as Markov-boundary (MB), ancestor-but-non-boundary (AN), or off-target (OT), and we report category counts as well as Precision, Recall, and F1 for MB recovery. To assess whether the learned local interface and retained upstream structure cover the true causal ancestry relevant to emergence, we additionally report ancestor-level accuracy (Anc-F1). Formal definitions are provided in Appendix D.1.

(ii) Causal structure recovery for emergence explanation. To compare CAMO with methods operating solely on logged observables, we report standard structure metrics—Precision, F1, Ancestor F1, FPR, SHD, and Added/Missed/Reversed/Unoriented edge counts (Appendix D.3)—computed on the observed-variable projection $\Pi(\cdot)$, which removes constructed factors and maps their effects back to observed variables (Appendix D.2). This projection enables a fair comparison with statistical causal discovery methods, which typically operate on fixed observed variables and do not propose new factors. For completeness, we also report results without projection; see Appendix B.2.

Model. CAMO is evaluated with multiple LLM backbones: Qwen3-235B-A22B (Yang et al., 2025), DeepSeek-R1-Distill-Qwen-32B (DeepSeek-AI, 2025a), DeepSeek-V3.2 (DeepSeek-AI, 2025b), GPT-5 mini (OpenAI, 2025) and Gemma3-27B (Team et al., 2025).

Results. **Factor discovery.** Table 1 shows that CAMO best recovers the target's Markov boundary and its full ancestor set, validating (i) a correct local causal interface for predicting/intervening on Y and (ii) sufficient upstream coverage for emergence explanation; notably, it substantially outperforms all baselines in both MB-F1 and Anc-F1.

Structure recovery and minimal connecting

[2]We do *not* perform time-series causal discovery: each independent rollout is treated as a single observational unit and mapped to run-level features, yielding a cross-sectional dataset on which PC/FCI are applied as static baselines.

Table 1: Comparison of different causal discovery methods for factor discovery on the O2O delivery simulation (Markov boundary vs. full-ancestor targets). All LLM-based baselines are evaluated using DeepSeek-V3.2.

Method	MB↑	AN↑	OT↓	MB-Prec↑	MB-Rec↑	MB-F1↑	Anc-Prec↑	Anc-Rec↑	Anc-F1↑
Single-round (Text-only)	0	8	6	0.00	0.00	0.00	0.20	0.41	0.27
Single-round (Data-grounded)	1	8	6	0.20	0.50	0.29	0.25	0.61	0.35
Single-round (CoT)	0	8	5	0.00	0.00	0.00	0.26	0.41	0.32
Multi-round (No causal feedback)	0	8	4	0.00	0.00	0.00	0.27	0.41	0.33
CAMO (Ours)	**2**	**9**	**0**	**1.00**	**1.00**	**1.00**	**0.95**	**1.00**	**0.98**

Table 2: Comparison of different causal discovery methods for causal structure recovery on the O2O delivery simulation. All LLM baselines use DeepSeek-V3.2; CAMO uses the LLM specified in parentheses.

Method	Prc↑	Rec↑	F1↑	Acc↑	Anc-F1↑	FPR↓	SHD↓	Add.↓	Mis.↓	Rev.↓	Unor.↓
Statistical causal discovery (SCD)											
PC	0.20	0.38	0.26	0.61	0.42	0.34	23	15	3	5	**0**
FCI	0.13	0.23	0.17	0.58	0.19	0.34	29	19	**0**	6	4
GES	0.12	0.15	0.14	0.65	0.19	0.24	20	9	6	5	**0**
MMHC	0.12	0.15	0.13	0.64	0.12	0.25	20	9	5	6	**0**
Pure LLM-based methods											
Efficient-CDLMs	0.71	0.77	0.74	0.90	0.91	0.07	7	4	1	**0**	2
MAC	0.67	0.77	0.71	0.89	0.92	0.08	8	5	2	**0**	1
PAIRWISE	0.37	0.54	0.44	0.75	0.46	0.20	15	9	5	1	**0**
Hybrid SCD+LLM methods											
SCD (PC)-LLM	0.42	0.77	0.54	0.76	0.60	0.24	18	15	3	**0**	**0**
ReAct	0.60	0.69	0.64	0.86	0.83	0.10	10	6	2	**0**	2
LLM-KBCI	0.85	0.85	0.85	0.94	0.74	0.03	3	**1**	1	1	**0**
CAMO (Qwen3-235B-A22B)	0.80	0.92	0.86	0.94	0.97	0.05	4	3	1	**0**	**0**
CAMO (DeepSeek-R1-Qwen-32B)	0.59	**1.00**	0.74	0.88	0.76	0.15	9	9	**0**	**0**	**0**
CAMO (GPT-5 mini)	0.87	**1.00**	0.93	0.97	0.97	0.03	2	2	**0**	**0**	**0**
CAMO (DeepSeek-V3.2)	**0.93**	**1.00**	**0.96**	**0.99**	**1.00**	**0.02**	**1**	**1**	**0**	**0**	**0**
CAMO (Gemma3-27B)	0.71	0.92	0.80	0.92	0.82	0.08	6	5	1	**0**	**0**

explanatory subgraph. Against causal-structure-recovery baselines, CAMO achieves the strongest recovery in both metrics and qualitative comparisons (Table 2, Figure 3; after observed-variable projection). Crucially, this validates our Minimal Connecting Explanatory Subgraph objective: CAMO is *sufficient* in that it best preserves the full upstream explanatory pathways (highest Anc-F1), and *minimal* in that it introduces the fewest redundant edges (lowest Added, SHD, and FPR), yielding the most faithful and compact graph.

Analysis without projection. For completeness, we also report metrics (Table 5) and visualizations (Figure 7) without graph projection, and compare how different LLM backbones affect CAMO; notably, DeepSeek-V3.2 and GPT-5 mini yield higher Prc/F1 and more compact graphs (lower FPR/SHD). See Appendix B.2 for details.

Refinement effectiveness. We analyze the A2–A3 iterative add/prune refinement in Appendix B.1 (Figure 5). Motivated by Theorem 4.2, Figure 4 reports estimated (p, C); see Appendix B.4 and Appendix C.1 for analysis and estimation details.

Ablation Study. Table 4 and Figure 9 show that ablating any component reduces recovery, with the most pronounced degradation when removing A1&A2; see Appendix B.5 for details.

5.2 Identifying Effective Interventions without Ground Truth (RQ2)

We evaluate CAMO in settings without ground-truth causal graphs, asking whether the learned structure yields *actionable* guidance for intervening on emergent outcomes.

Simulation setup. We consider two LLM-agent simulation environments with complex emergent dynamics: **Smallville** (Park et al., 2023) and **AgentSociety** (Piao et al., 2025b). In Smallville, the target outcome is *agent coordination*; in AgentSociety, we study *opinion polarization* and the *spread of inflammatory messages*.

Metrics. Without ground-truth causal graphs, we evaluate whether the learned graph G yields actionable guidance by ranking effective interventions ahead of ineffective ones. Following the

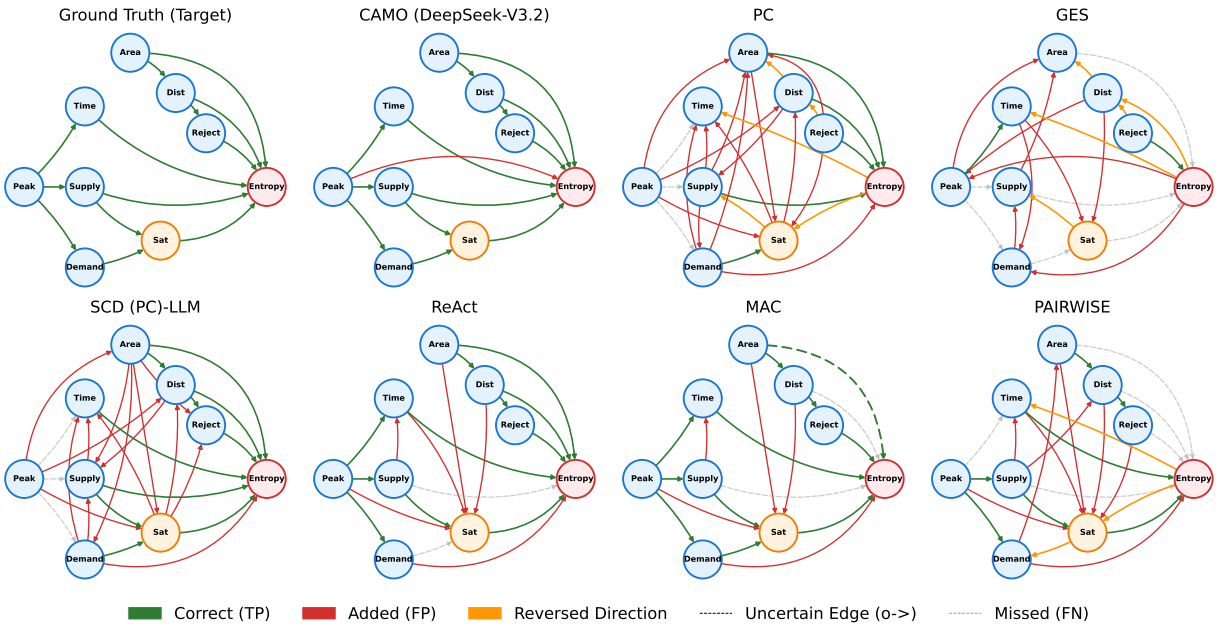


Figure 3: Qualitative comparison of recovered causal structures (O2O delivery simulation; projected). Full visualizations for all methods are in Figure 6.

Table 3: Intervention ranking without ground-truth causal graphs (higher is better). All LLM-based methods use DeepSeek-V3.2. Results are averaged over 3 runs.

	Smallville (Coord.)			AgentSociety (Polar.)			AgentSociety (Inflam.)		
Method	Precision@5	MAP@5	MRR	Precision@5	MAP@5	MRR	Precision@5	MAP@5	MRR
PC	0.40	0.22	0.25	0.30	0.32	0.75	0.17	0.07	0.21
MMHC	0.22	0.14	0.31	0.40	0.56	**1.00**	0.16	0.08	0.22
PAIRWISE	0.40	0.22	0.25	0.30	0.32	0.75	0.18	0.09	0.24
SCD (PC)+LLM	0.18	0.09	0.24	0.50	0.63	**1.00**	0.16	0.08	0.22
LLM-KBCI	0.40	0.28	0.33	0.50	0.46	0.50	0.26	0.16	0.27
CAMO(Ours)	**0.60**	**0.64**	**0.50**	**0.60**	**0.84**	**1.00**	**0.60**	**0.71**	**0.61**

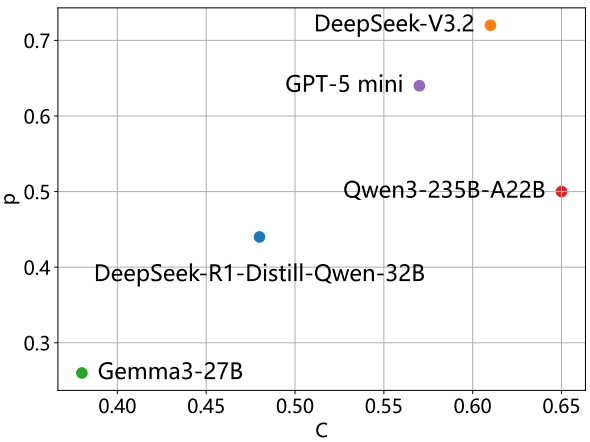


Figure 4: Estimated (p, C) across LLMs.

Table 4: Ablation on core components (O2O simulation); evaluated using GPT-5 mini.

Variant	MB-F1↑	Anc-F1↑	OT↓	FPR↓	SHD↓	Unor.↓
Full CAMO	**0.80**	0.90	1	**0.04**	**5**	**0**
w/o A1&A2	0.00	0.46	**0**	0.08	23	**0**
w/o A3-add	0.67	0.74	1	**0.04**	9	**0**
w/o A5 (no counterfactual)	**0.80**	0.79	3	0.05	10	**0**
w/o competing hypotheses	0.67	**0.91**	2	**0.04**	6	**0**

root-cause-analysis-style evaluation in Zheng et al. (2024); Shen et al. (2024), we run Random Walk with Restart from Y and use each candidate target node's RWR score (stationary visit probability) as the ranking score, where higher values indicate stronger graph-mediated relevance to Y. We report Precision@K, MAP@K (K=5), and MRR; detailed definitions are in Appendix E.

Results. Table 3 shows CAMO consistently achieves the best Precision@5 and MAP@5 on Polar., Inflam., and Smallville, indicating more accurate top-K *effectiveness judgments*. Some baselines attain high MRR (e.g., Polar MMHC/SCD: 1.00) but lower MAP@5, suggesting weaker ranking among the remaining interventions. Baselines perform worst on Inflammatory Spread; see Appendix B.3 for a detailed case study.

6 Conclusion

CAMO is a multi-agent framework for explaining emergence in LLM-agent simulations. It learns a computable Markov boundary and a minimal connecting explanatory subgraph. Experiments show strong factor recovery, accurate structure reconstruction, and actionable intervention guidance.

7 Limitations

This work is scoped to causal discovery in LLM-agent simulations. CAMO depends on LLM-derived priors (and thus on pretraining, prompts, and available context), and counterfactual refinement cannot guarantee completeness. As a *local causal interface* around Y, it may miss important confounders that are neither logged in X_{obs} nor expressible in $\mathcal{H}$, which can leave residual confounding. Results are validated only with respect to the simulator's logging/dynamics and may not transfer beyond it. The iterative loop also incurs a budget–accuracy trade-off from repeated simulator queries, and learned actionable levers could be misused or over-interpreted. We partially mitigate these issues by grounding updates in observational constraints and simulator counterfactual tests, evaluating in controlled simulation settings, and restricting claims to simulator-scoped, evidence-backed interventions.

8 Ethical considerations

Our study focuses on causal discovery and explanation in LLM-empowered multi-agent simulations, including an online-to-offline (O2O) delivery platform, where simulator mechanisms are calibrated using real-world statistics. We do not conduct experiments on human subjects, and we do not use, store, or release any personally identifiable information: the calibration relies only on non-identifying, aggregate statistics rather than individual-level records, and all reported analyses are performed on synthetic simulation rollouts and logs. Since CAMO can surface actionable "levers" in simulated systems, there is a risk that such interventions could be over-interpreted or misused outside the simulator; we therefore restrict claims to simulator-scoped, evidence-backed findings and evaluate only in controlled simulation settings.

Acknowledgments

This work has been supported in part by National Key Research and Development Program of China (No.2025YFE0216300), National Natural Science Foundation of China (No. 62472306, No. 62441221), Tianjin University's 2024 Special Project on Disciplinary Development (No. XKJS-2024-5-9), Tianjin University Talent Innovation Reward Program for Literature & Science Graduate Student (C1-2022-010), and Henan Province Key Research and Development Program (No.251111210500), Tianjin University Independent Innovation Project (No.2025XJ3-0043).

References

Carlo Adornetto, Adrian Mora, Kai Hu, Leticia Izquierdo Garcia, Parfait Atchade-Adelomou, Gianluigi Greco, Luis Alberto Alonso Pastor, and Kent Larson. 2025. Generative agents in agent-based modeling: Overview, validation, and emerging challenges. *IEEE Transactions on Artificial Intelligence*.

Amey Agrawal, Nitin Kedia, Jayashree Mohan, Ashish Panwar, Nipun Kwatra, Bhargav S Gulavani, Ramachandran Ramjee, and Alexey Tumanov. 2024. Vidur: A large-scale simulation framework for llm inference. *Proceedings of Machine Learning and Systems*, 6:351–366.

Jacy Reese Anthis, Ryan Liu, Sean M Richardson, Austin C Kozlowski, Bernard Koch, James Evans, Erik Brynjolfsson, and Michael Bernstein. 2025. Llm social simulations are a promising research method. *arXiv preprint arXiv:2504.02234*.

Ariel Flint Ashery, Luca Maria Aiello, and Andrea Baronchelli. 2025. Emergent social conventions and collective bias in llm populations. *Science Advances*, 11(20):eadu9368.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, and 1 others. 2024. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *ICLR*.

Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. 2024. Unveiling causal reasoning in large language models: Reality or mirage? *Advances in Neural Information Processing Systems*, 37:96640–96670.

David Maxwell Chickering. 2002. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554.

Yun-Shiuan Chuang, Agam Goyal, Nikunj Harlalka, Siddharth Suresh, Robert Hawkins, Sijia Yang, Dhavan Shah, Junjie Hu, and Timothy Rogers. 2024. Simulating opinion dynamics with networks of llm-based agents. In *Findings of the association for computational linguistics: NAACL 2024*, pages 3326–3346.

Gordon Dai, Weijia Zhang, Jinhan Li, Siqi Yang, Srihas Rao, Arthur Caetano, Misha Sra, and 1 others. 2024. Artificial leviathan: Exploring social evolution of llm agents through the lens of hobbesian social contract theory. *arXiv preprint arXiv:2406.14373*.

DeepSeek-AI. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI. 2025b. Deepseek-v3.2: Pushing the frontier of open large language models.

Mugariya Farooq, Shahad Hardan, Aigerim Zhumbhayeva, Yujia Zheng, Preslav Nakov, and Kun Zhang. 2023. Understanding breast cancer survival: Using causality and language models on multi-omics data. *arXiv preprint arXiv:2305.18410*.

Tao Feng, Lizhen Qu, Niket Tandon, Zhuang Li, Xiaoxi Kang, and Gholamreza Haffari. 2024. On the reliability of large language models for causal discovery. *arXiv preprint arXiv:2407.19638*.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2024. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1):1–24.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Prateek Gupta, Qiankun Zhong, Hiromu Yakura, Thomas Eisenmann, and Iyad Rahwan. 2025. The role of social learning and collective norm formation in fostering cooperation in llm multi-agent systems. *arXiv preprint arXiv:2510.14401*.

Onder Gurcan. 2024. Llm-augmented agent-based modelling for social simulations: Challenges and opportunities. *arXiv preprint arXiv:2405.06700*.

Kairong Han, Kun Kuang, Ziyu Zhao, Junjian Ye, and Fei Wu. 2024. Causal agent based on large language model. *arXiv preprint arXiv:2408.06849*.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and 1 others. 2023. Cladder: Assessing causal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:31038–31065.

Thomas Jiralerspong, Xiaoyin Chen, Yash More, Vedant Shah, and Yoshua Bengio. 2024. Efficient causal graph discovery using large language models. *arXiv preprint arXiv:2402.01207*.

Elahe Khatibi, Mahyar Abbasian, Zhongqi Yang, Iman Azimi, and Amir M. Rahmani. 2025. Alcm: Autonomous llm-augmented causal discovery framework. *Preprint*, arXiv:2405.01744.

Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*.

Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

Hao Duong Le, Xin Xia, and Zhang Chen. 2024. Multi-agent causal discovery using large language models. *arXiv preprint arXiv:2407.15073*.

Chenxi Liu, Yongqiang Chen, Tongliang Liu, Mingming Gong, James Cheng, Bo Han, and Kun Zhang. 2024. Discovery of the hidden world with large language models. *Advances in Neural Information Processing Systems*, 37:102307–102365.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, and 1 others. 2025. Large language models and causal inference in collaboration: A comprehensive survey. *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7668–7684.

Yuren Mao, Peigen Liu, Xinjian Wang, Rui Ding, Jing Miao, Hui Zou, Mingjie Qi, Wanxiang Luo, Longbin Lai, Kai Wang, Zhengping Qian, Peilun Yang, Yunjun Gao, and Ying Zhang. 2025. Agent-kernel: A microkernel multi-agent system framework for adaptive social simulation powered by llms. *Preprint*, arXiv:2512.01610.

Xinyi Mou, Xuanwen Ding, Qi He, Liang Wang, Jingcong Liang, Xinnong Zhang, Libo Sun, Jiayu Lin, Jie Zhou, Xuanjing Huang, and 1 others. 2024. From individual to society: A survey on social simulation driven by large language model-based agents. *arXiv preprint arXiv:2412.03563*.

OpenAI. 2025. GPT-5 mini (openai api model specs). `https://platform.openai.com/docs/models/gpt-5-mini`. Accessed: 2025-12-31.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Jinghua Piao, Zhihong Lu, Chen Gao, Fengli Xu, Qinghua Hu, Fernando P Santos, Yong Li, and James Evans. 2025a. Emergence of human-like polarization among large language model agents. *arXiv preprint arXiv:2501.05171*.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, and 1 others. 2025b. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. *arXiv preprint arXiv:2502.08691*.

Giorgio Piatti, Zhijing Jin, Max Kleiman-Weiner, Bernhard Schölkopf, Mrinmaya Sachan, and Rada Mihalcea. 2024. Cooperate or collapse: Emergence of sustainable cooperation in a society of llm agents. *Advances in Neural Information Processing Systems*, 37:111715–111759.

Siyue Ren, Zhiyao Cui, Ruiqi Song, Zhen Wang, and Shuyue Hu. 2024. Emergence of social norms in generative agent societies: principles and architecture. *arXiv preprint arXiv:2403.08251*.

Jonathan Richens and Tom Everitt. 2024. Robust agents learn causal world models. *arXiv preprint arXiv:2402.10877*.

Christoph Riedl. 2025. Emergent coordination in multi-agent language models. *arXiv preprint arXiv:2510.05174*.

ChengAo Shen, Zhengzhang Chen, Dongsheng Luo, Dongkuan Xu, Haifeng Chen, and Jingchao Ni. 2024. Exploring multi-modal integration with tool-augmented llm agents for precise causal discovery. *arXiv preprint arXiv:2412.13667*, 1(3).

Peter Spirtes and Clark Glymour. 1991. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72.

Peter Spirtes, Clark N Glymour, and Richard Scheines. 2000. *Causation, prediction, and search*. MIT press.

Ryosuke Takata, Atsushi Masumori, and Takashi Ikegami. 2024. Spontaneous emergence of agent individuality through social interactions in llm-based communities. *arXiv preprint arXiv:2411.03252*.

Masayuki Takayama, Tadahisa Okuda, Thong Pham, Tatsuyoshi Ikenoue, Shingo Fukuma, Shohei Shimizu, and Akiyoshi Sannai. 2024. Integrating large language models in causal discovery: A statistical causal approach. *arXiv preprint arXiv:2402.01454*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. 2006. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78.

Hai Wang, Lei Zhao, Kaile Yan, Xiaoyi Wang, Yile Liang, and Jinghua Hao. 2025a. Meituan: Operational-level on-demand food delivery data for the 2025 informs tsl data-driven research challenge. *Available at SSRN 5696423*.

Qian Wang, Jiaying Wu, Zhenheng Tang, Bingqiao Luo, Nuo Chen, Wei Chen, and Bingsheng He. 2025b. What limits llm-based human simulation: Llms or our design? *arXiv preprint arXiv:2501.08579*.

Anpeng Wu, Kun Kuang, Minqin Zhu, Yingrong Wang, Yujia Zheng, Kairong Han, Baohong Li, Guangyi Chen, Fei Wu, and Kun Zhang. 2024. Causality for large language models. *arXiv preprint arXiv:2410.15319*.

Yuchen Xia, Daniel Dittler, Nasser Jazdi, Haonan Chen, and Michael Weyrich. 2024. Llm experiments with simulation: Large language model multi-agent system for simulation model parametrization in digital twins. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Kui Yu, Lin Liu, Jiuyong Li, and Huanhuan Chen. 2018. Mining markov blankets without causal sufficiency. *IEEE transactions on neural networks and learning systems*, 29(12):6333–6347.

Xiangning Yu, Zhuohan Wang, Linyi Yang, Haoxuan Li, Anjie Liu, Xiao Xue, Jun Wang, and Mengyue Yang. 2025a. Causal sufficiency and necessity improves chain-of-thought reasoning. *arXiv preprint arXiv:2506.09853*.

Xiangning Yu, Xiao Xue, Deyu Zhou, Gang Wang, and Zhiyong Feng. 2025b. Unlocking complexity: Harnessing value entropy for advanced multidimensional utility evaluation in service ecosystems. *IEEE Transactions on Services Computing*.

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

Jiji Zhang. 2008. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9(7).

Xinnong Zhang, Jiayu Lin, Xinyi Mou, Shiyue Yang, Xiawei Liu, Libo Sun, Hanjia Lyu, Yihang Yang, Weihong Qi, Yue Chen, Guanying Li, Ling Yan, Yao Hu, Siming Chen, Yu Wang, Xuanjing Huang, Jiebo

Luo, Shiping Tang, Libo Wu, and 2 others. 2025. Socioverse: A world model for social simulation powered by llm agents and a pool of 10 million real-world users. *Preprint*, arXiv:2504.10157.

Lecheng Zheng, Zhengzhang Chen, Dongjie Wang, Chengyuan Deng, Reon Matsuoka, and Haifeng Chen. 2024. Lemma-rca: A large multi-modal multi-domain dataset for root cause analysis. *arXiv preprint arXiv:2406.05375*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Theory Proofs

This appendix provides proofs for the theoretical results in § 4. Throughout, we focus on properties that are directly implied by the refinement procedure and do not assume global identifiability or set-wise convergence unless explicitly stated.

Proof of Theorem 4.1

At refinement step $t+1$, the data-realized representation is first augmented by admitting a (possibly empty) set of new computable factors $\mathcal{Z}_{t+1}$:

$$\tilde{V}^{(t+1)} = V^{(t)} \cup \mathcal{Z}_{t+1}, \tag{8}$$

and then pruned by removing variables that are conditionally redundant for predicting the target Y given the remaining variables. Define the pruned set

$$R_t := \left\{ W \in \tilde{V}^{(t+1)} \ : \ Y \perp\!\!\!\perp W \mid \tilde{V}^{(t+1)} \backslash \{W\} \right\}, \tag{9}$$

and the updated representation

$$V^{(t+1)} = \tilde{V}^{(t+1)} \setminus R_t. \tag{10}$$

Let B_t denote the Markov boundary of Y in $V^{(t)}$, and let $m_t := |\mathcal{Z}_{t+1}|$. Consider the candidate set

$$C_{t+1} := (B_t \cup \mathcal{Z}_{t+1}) \setminus R_t. \tag{11}$$

For any $W \in R_t$, by definition (9),

$$Y \perp\!\!\!\perp W \mid \tilde{V}^{(t+1)} \setminus \{W\}, \tag{12}$$

so removing W does not reduce the conditional predictive sufficiency of the remaining variables for Y. Hence, there exists a sufficient conditioning set $B'_{t+1} \subseteq C_{t+1}$ such that

$$Y \perp\!\!\!\perp V^{(t+1)} \setminus B'_{t+1} \mid B'_{t+1}. \tag{13}$$

By construction,

$$|B'_{t+1}| \le |B_t| + m_t - |B_t \cap R_t|. \tag{14}$$

Since B_{t+1} is the *minimal* sufficient set for Y in $V^{(t+1)}$, it follows that

$$|B_{t+1}| \le |B'_{t+1}| \le |B_t| + m_t, \tag{15}$$

which proves (4). Moreover, if $B_t \cap R_t \neq \varnothing$, then at least one previously boundary variable is pruned and the bound tightens accordingly. □

Proof of Theorem 4.2

Let

$$F_t := I(Y; X_{\mathrm{obs}} \mid V^{(t)}) \tag{16}$$

denote the residual conditional dependence between the target Y and the full observation space at refinement step t. Let E_t denote the event that the refinement step achieves the contraction condition in (6). By assumption,

$$\Pr(E_t) \ge p. \tag{17}$$

By definition of the refinement capability,

$$F_{t+1} \le \begin{cases} (1-C)F_t, & \text{if } E_t \text{ occurs,} \\ F_t, & \text{otherwise.} \end{cases} \tag{18}$$

Taking conditional expectation and using $\Pr(E_t) \ge p$, we obtain

$$\mathbb{E}[F_{t+1} \mid F_t] \le (1 - pC)F_t. \tag{19}$$

Iterating this inequality yields

$$\mathbb{E}[F_t] \le (1 - pC)^t F_0, \tag{20}$$

which establishes geometric decay of the residual dependence. This result implies convergence in the sense that the representation stabilizes with respect to predictive and interventional sufficiency, without requiring set-wise convergence or uniqueness of the limiting Markov boundary. □

Proof of Proposition 1

Consider an edge between variables X and Z whose orientation is ambiguous under the observationally identified CPDAG or PAG. Define the simulator-implied interventional effect

$$\delta(X \to Z) := \sup_{x,x'} D_{\mathrm{TV}}\Big(P(Z \mid \mathrm{do}(X{=}x)), P(Z \mid \mathrm{do}(X{=}x'))\Big). \tag{21}$$

If $\delta(X \to Z) > 0$, then intervening on X induces a detectable change in the distribution of Z. Under the locality/modularity assumption (i.e., the intervention is approximately surgical in the simulator), any candidate orientation in which X is not a (possibly indirect) cause of Z would imply invariance of Z under $\mathrm{do}(X)$, contradicting the observed interventional effect.

By the law of large numbers, empirical estimates of the interventional distributions obtained from repeated paired simulator rollouts converge almost surely to their simulator-implied limits. Consequently, candidate orientations incompatible with the observed counterfactual contrast are eliminated with probability one as the number of rollouts $n \to \infty$. □

Controlling for invariance in probing. Importantly, our counterfactual probing is an in-simulator operation and the resulting orientations should be interpreted as simulator-specific causal evidence. To make simulator counterfactual probing as close as possible to a localized (approximately surgical) intervention, we generate paired rollouts that (i) start from the same simulator state/trajectory prefix, (ii) use aligned randomness via shared seeds (common random numbers) for the environment and scheduling, and (iii) keep all non-target simulator components fixed (environment dynamics, scheduler, and agent policy/system prompt), while only clamping the target variable/mechanism X.

B Supplementary Experimental Results

B.1 A2–A3 Iterative Refinement (Add/Prune)

Figure 5 illustrates how CAMO iteratively refines the candidate factor set via the A2–A3 loop. Overall, the add/prune mechanism rapidly compresses an initially over-complete proposal into a compact, stable set, while progressively recovering the ground-truth local neighborhood (Markov boundary size $= 2$) across rounds.

(1) Candidate-set dynamics. At each round, A2 proposes a (potentially over-complete) set of candidate factors, which is then sharply reduced by A3's prune step. After a few rounds, the candidate set size stabilizes at a much smaller scale, indicating effective regularization against uncontrolled growth.

(2) Role separation: prune vs. add. Across LLM backbones, a consistent pattern emerges: the prune step accounts for the dominant reduction (removing irrelevant factors), whereas the add step introduces only modest corrections by re-inserting a small number of missed-but-necessary factors. This division of labor prevents early over-pruning and improves recall without sacrificing compactness.

(3) Markov-boundary recovery. We additionally track the estimated Markov boundary size, which remains substantially smaller than the full candidate set throughout. More importantly, it gradually approaches the ground-truth Markov boundary (size 2 in this setting) and converges toward it across rounds, demonstrating that the refinement loop increasingly identifies the true local causal neighborhood.

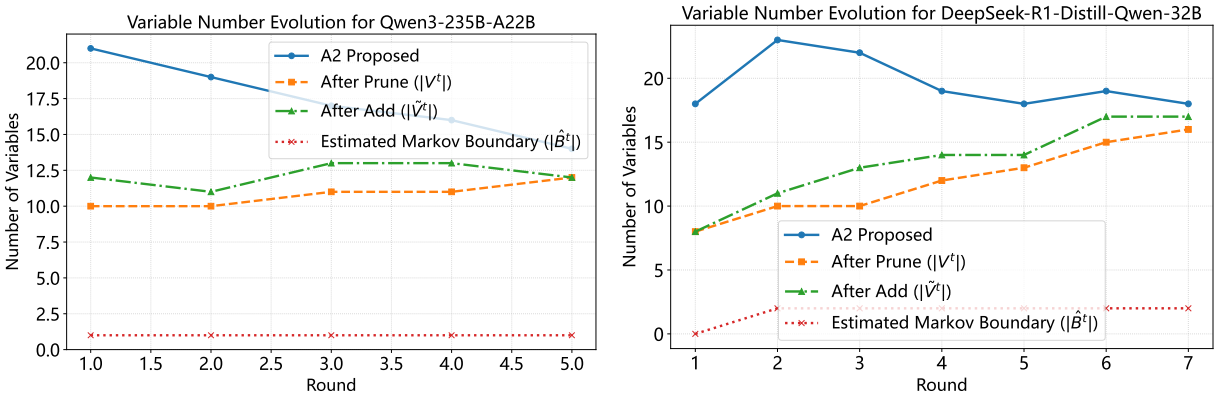
Together, these results suggest that the A2–A3 loop serves as a principled "compress-and-correct" procedure: it regularizes the search space by pruning spurious variables, while preserving recall through selective additions, thereby progressively isolating the true local causal structure.

B.2 Effect of LLM backbones in the full factor space (no projection).

Table 5 compares CAMO under different LLM backbones when evaluating directly in the full induced factor space (without projection); qualitative graphs are shown in Figure 7.

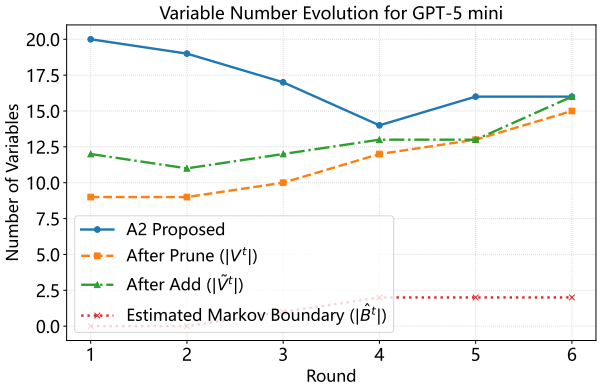
Sufficiency. Across backbones, CAMO maintains strong upstream coverage, attaining high recall for all but Gemma3-27B. This suggests that once factors are elicited, the framework largely preserves the target-relevant causal pathways.

Minimality. Stronger backbones (DeepSeek-V3.2, GPT-5 mini) produce markedly more compact and faithful graphs, achieving higher Prc/F1 and lower FPR/SHD, i.e., fewer redundant relations in the induced factor space. Even with smaller backbones, CAMO still yields usable graphs and retains competitive structure recovery, though with a looser sparsity–accuracy trade-off.

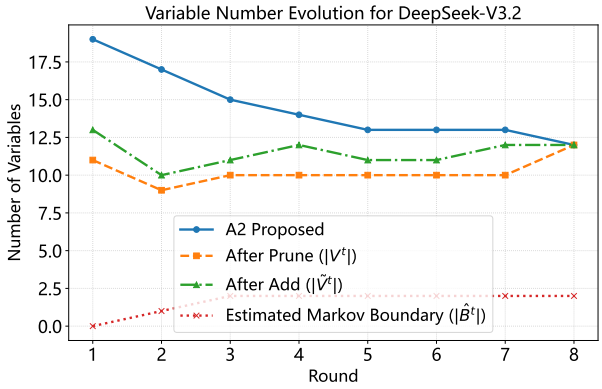


(a) Qwen3-235B-A22B

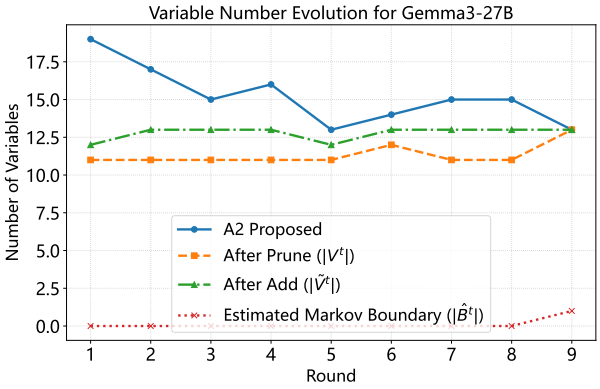
(b) DeepSeek-R1-Distill-Qwen-32B



(c) GPT-5 mini



(d) DeepSeek-V3.2



(e) Gemma3-27B

Figure 5: Add/prune dynamics of CAMO (A2–A3) under different LLM backbones. We plot, per round, the A2 proposal size, sizes after prune/add, and the estimated Markov boundary (MB) size. Pruning sharply compresses the proposal set, while add makes small corrections; meanwhile, the estimated MB progressively recovers the ground-truth MB size (2 in this setting).

B.3 Case Study

We present qualitative case studies of the causal graphs learned by CAMO across multiple scenarios and emergent causal patterns. In particular, we visualize the learned causal structures for three representative emergent phenomena: (i) agent coordination (Figure 10), (ii) opinion polarization (Figure 11), and (iii) the spread of inflammatory messages (Figure 12).

These examples illustrate how CAMO abstracts over low-level agent interactions and recovers compact, high-level causal structures that capture the key mechanisms underlying each emergent phenomenon. Importantly, all graphs shown here are learned without access to any ground-truth causal structure. They are therefore intended to support qualitative inspection and interpretability, rather than serving as a quantitative evaluation benchmark.

For each setting, the learned graph highlights a small set of upstream factors that are most actionable for intervention, while the remaining nodes

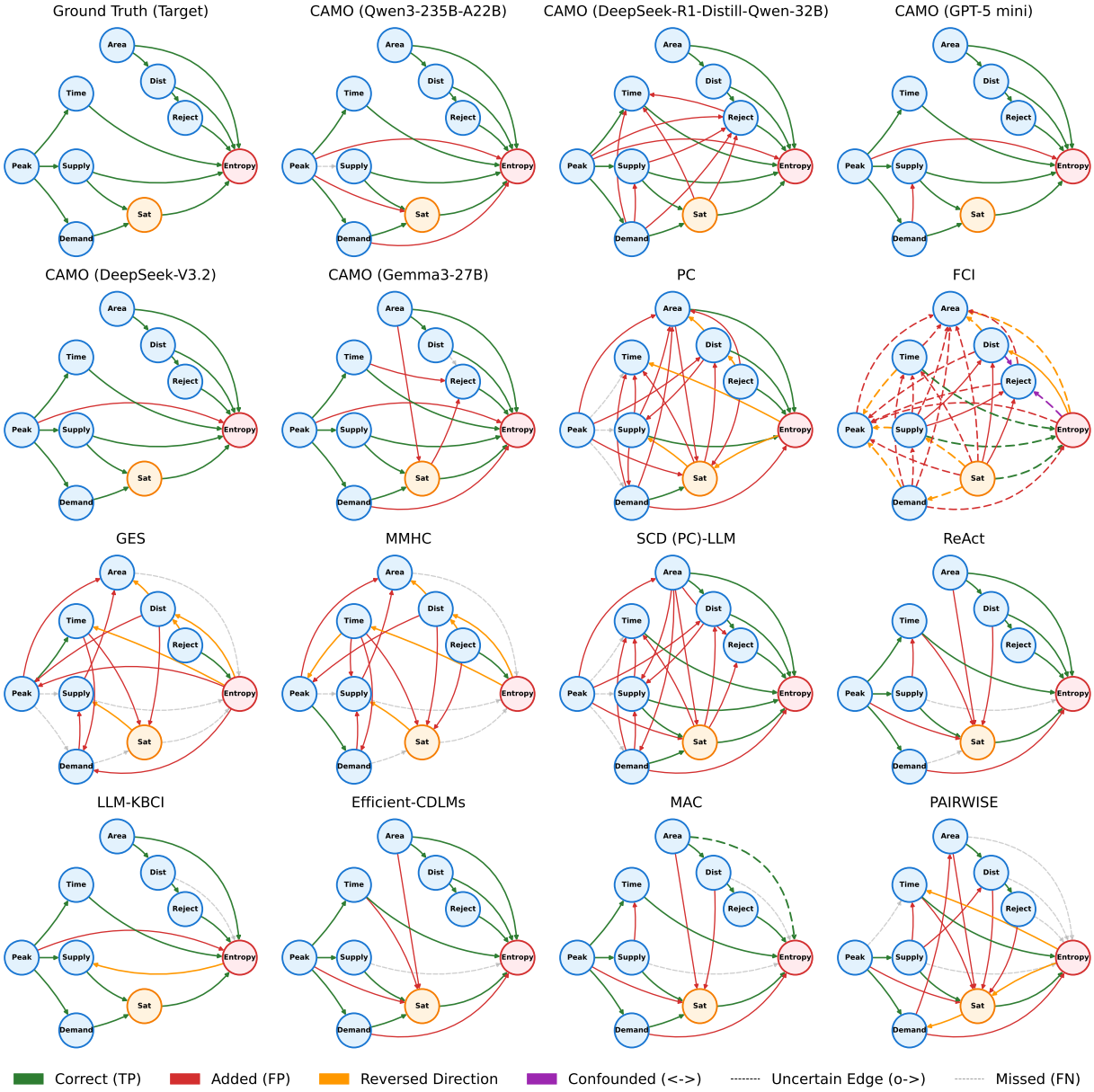


Figure 6: Recovered causal structures for *all* methods on the O2O delivery simulation after applying the observed-variable projection. The figure provides a complete qualitative comparison complementing Figure 3.

represent environmental conditions, intermediate system states, or emergent outcome indicators.

In addition, for one intervention trial in the agent coordination setting, we visualize the temporal evolution of the emergent coordination behavior based on the learned causal graph (see Figure 8). This timeline links the onset of the coordination phenomenon to the key causal factors identified by CAMO, providing an intuitive view of how interventions propagate through intermediate states and ultimately shape the emergent outcome.

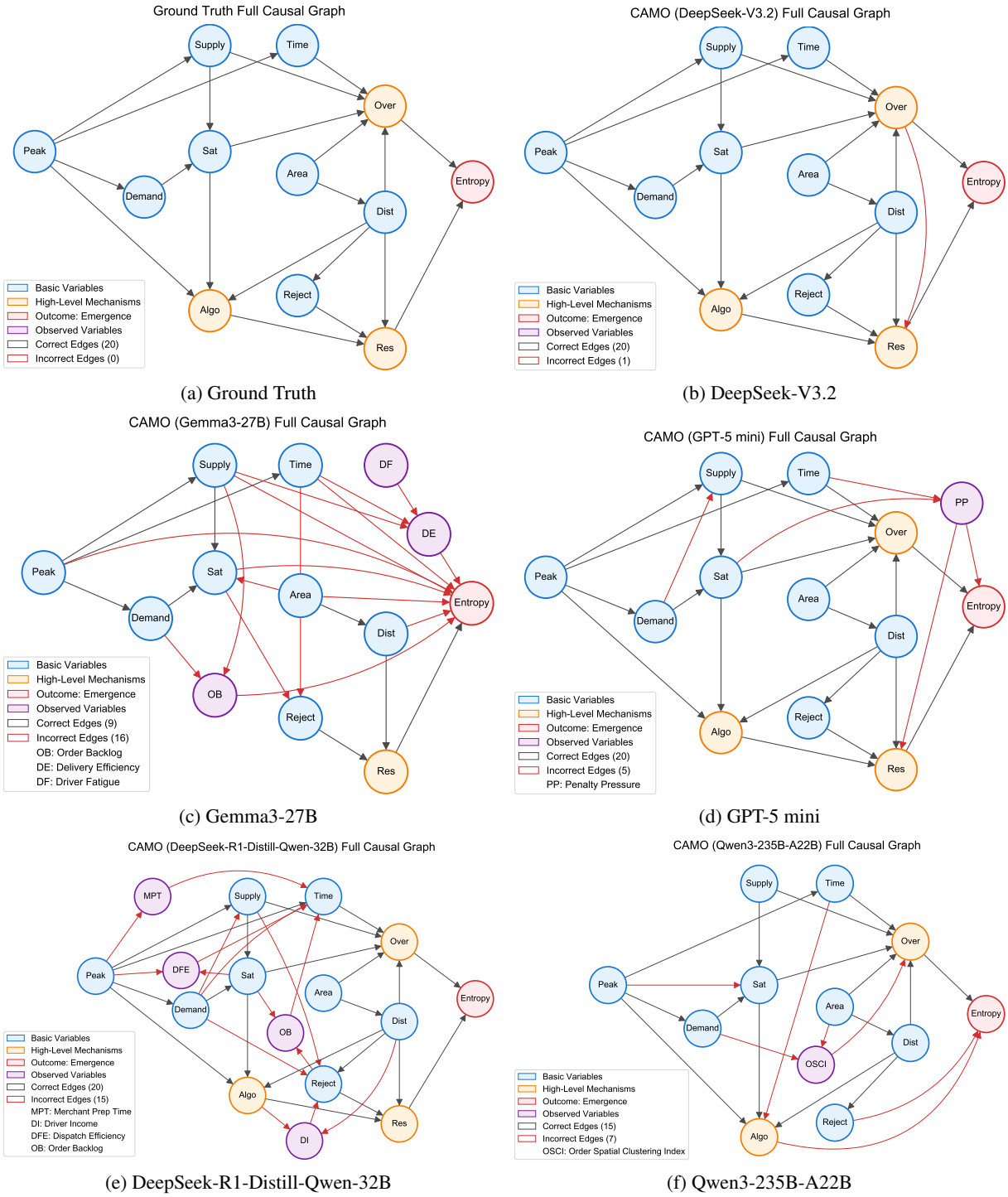
B.4 Analysis of Estimated Refinement Effectiveness (p, C)

Figure 4 summarizes the estimated refinement-effectiveness parameters (p, C) across LLM backbones. Recall that p captures the probability that an iteration yields an informative refinement step, while C measures the typical (relative) contraction strength conditioned on being informative; together, they govern the expected geometric decay rate through the product pC (Theorem 4.2).

We observe clear backbone-dependent differences that are consistent with the refinement dynamics in Figure 5. DeepSeek-V3.2 exhibits the

Table 5: Causal structure recovery of CAMO on the O2O delivery simulation under different LLM backbones (results *without graph projection*).

LLM Backbone	Prc↑	Rec↑	F1↑	Acc↑	Anc-F1↑	FPR↓	SHD↓	Add.↓	Mis.↓	Rev.↓	Unor.↓
Qwen3-235B-A22B	0.68	0.75	0.71	0.92	0.79	0.05	12	7	5	**0**	**0**
DeepSeek-R1-Distill-Qwen-32B	0.57	**1.00**	0.73	0.94	0.62	0.07	15	15	**0**	**0**	**0**
GPT-5 mini	0.80	**1.00**	0.89	0.97	0.90	0.04	5	5	**0**	**0**	**0**
DeepSeek-V3.2	**0.95**	**1.00**	**0.98**	**0.99**	**0.98**	**0.01**	**1**	**1**	**0**	**0**	**0**
Gemma3-27B	0.36	0.45	0.40	0.87	0.59	0.08	27	16	11	**0**	**0**



(a) Ground Truth

(b) DeepSeek-V3.2

(c) Gemma3-27B

(d) GPT-5 mini

(e) DeepSeek-R1-Distill-Qwen-32B

(f) Qwen3-235B-A22B

Figure 7: Qualitative comparison of recovered causal structures (O2O delivery simulation; without projection).

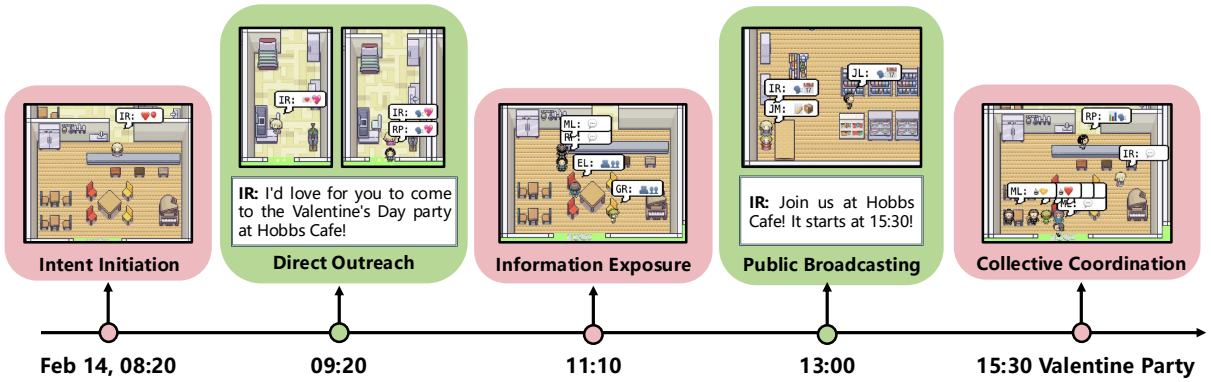
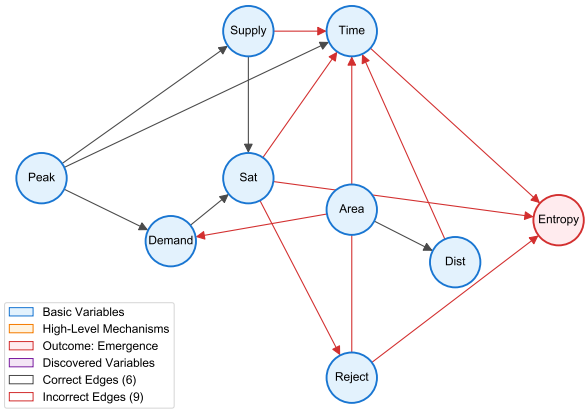
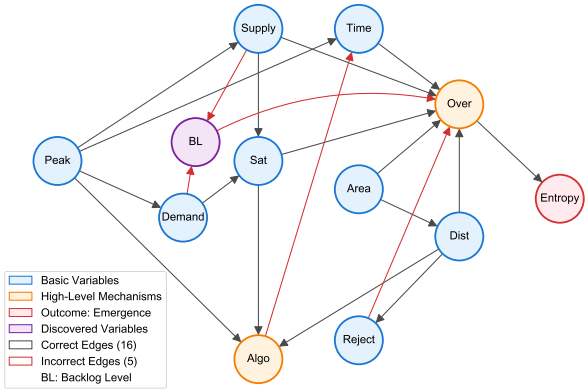


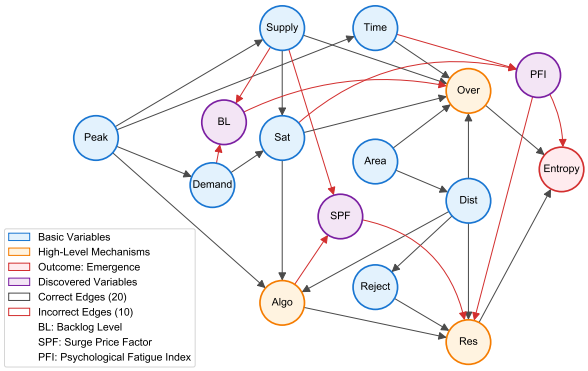
Figure 8: Timeline of an intervention trial in the agent coordination setting. The figure illustrates the emergence of collective coordination over time, progressing from intent initiation and direct outreach to information exposure and public broadcasting, culminating in collective coordination at the event time (15:30).



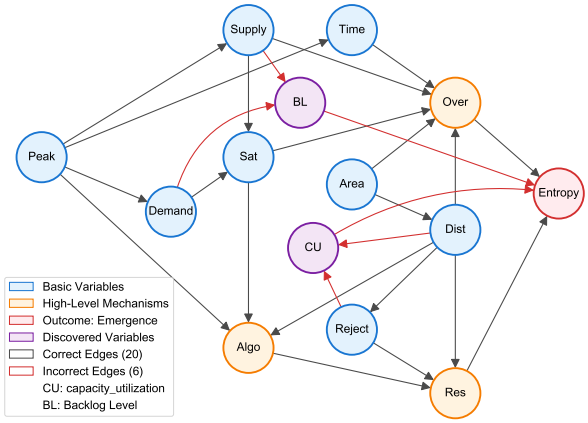
(a) w/o A1&A2



(b) w/o A3-add



(c) w/o A5 (no counterfactual)



(d) w/o competing hypotheses

Figure 9: Ablation study on the O2O delivery simulation.

strongest overall effectiveness, with both a relatively large p and a strong conditional contraction, resulting in the largest estimated pC and thus the fastest expected decay of residual dependence. GPT-5 mini follows with a similarly balanced profile, suggesting that informative steps occur frequently and yield reliable contraction.

Qwen3-235B-A22B shows a comparatively larger C but a smaller p, indicating more intermittent progress: when refinement succeeds it tends to contract strongly, but successful steps occur less often. This is consistent with the add/prune curves where the candidate set is pruned effectively yet improvements are not uniformly sustained across rounds. DeepSeek-R1-Distill-Qwen-32B exhibits a smaller p (with moderate C), implying that infor-

mative refinement steps are rarer, aligning with its slower stabilization behavior over rounds.

Finally, Gemma3-27B yields a smaller pC, indicating less consistent contraction overall. Notably, this does not preclude early stabilization of the estimated Markov-boundary size in the add/prune curves; rather, it suggests that subsequent refinement is less reliably contractive (e.g., weaker regularization of the broader candidate set), which can delay stabilization in terms of residual dependence.

The concrete estimation procedure for (p, C) from the empirical $\{F_t\}$ trajectories is provided in Appendix C.1.

B.5 Ablation Design and Analysis

Ablation design. We ablate four core components of CAMO while keeping the rest of the pipeline unchanged. **w/o A1&A2** removes the worldview parsing and integration stages, so the system runs without structured prior hypotheses about variables and relations. **w/o A3-add** disables the add step in the A2–A3 refinement loop, i.e., factors are only pruned/refined without explicitly re-introducing missed candidates. **w/o A5 (no counterfactual)** removes the counterfactual adjudication module, so candidate edges/hypotheses are not filtered by intervention-based checks. **w/o competing hypotheses** disables hypothesis competition, replacing set-based competing hypotheses with a single-pass selection to test whether explicit competition is necessary. All variants are evaluated under the same simulator setting and budget as the full model.

Analysis. Table 4 shows that each module supports either *coverage* of the true upstream ancestry or *compactness* of the learned structure. Removing A1&A2 causes factor discovery to fail, indicating that worldview parsing and integration are necessary to surface causally relevant factors. Disabling A3-add weakens both local (MB) and upstream recovery, suggesting the add step is important for bringing back missed-but-necessary causes. Removing A5 (no counterfactual) largely preserves the local interface but introduces more spurious structure, showing that counterfactual adjudication primarily acts as a redundancy filter.

C Implementation Details

C.1 Estimating Refinement-Capability Parameters

This appendix describes how we compute practical diagnostics $(\hat{p}, \hat{C})$ from the A3 refinement trajectory. These quantities are meant to be simple, interpretable summaries of refinement effectiveness, not statistically exact estimators of the theoretical (p, C) in Theorem 4.2.

C.1.1 A computable uncertainty score H_b

For any variable set U, we use $H_b(Y \mid U)$ to denote a *computable* score of how uncertain Y remains after conditioning on U. In practice, $H_b(Y \mid U)$ is estimated by a held-out predictive loss of a classifier/regressor that predicts Y from U (e.g., cross-entropy for discrete Y). Smaller $H_b(Y \mid U)$ means U explains/predicts Y better.

C.1.2 Information gain and factor admission

At refinement step t, let $V^{(t)}$ be the current admitted factor set. For a candidate factor Z, we define its *computable information gain* as

$$\Delta^c I(Z) \;:=\; H_b(Y \mid V^{(t)}) - H_b(Y \mid V^{(t)}, Z). \tag{22}$$

This measures how much adding Z reduces the residual uncertainty of Y beyond what $V^{(t)}$ already explains. We admit Z if

$$\Delta^c I(Z) > \tau_I, \tag{23}$$

where $\tau_I \geq 0$ is a small margin (default $\tau_I = 0$).

Targeted evaluation. When estimating H_b and $\Delta^c I(\cdot)$, we follow Liu et al. (2024) and evaluate them on the subset of records that is hardest to predict under the current $V^{(t)}$ (the "worst-explained" group). This prevents improvements on easy cases from masking failures on hard cases.

C.1.3 Proxy residual difficulty $\widehat{F}_t$

Theorem 4.2 analyzes a residual dependence quantity F_t that is not directly computable. Along the refinement trajectory, we track a computable surrogate,

$$\widehat{F}_t \;:=\; H_b(Y \mid V^{(t)}), \tag{24}$$

where H_b is evaluated using the same protocol as above (i.e., on the worst-explained group at step t). Smaller $\widehat{F}_t$ indicates that the current admitted set explains Y better in the hardest region.

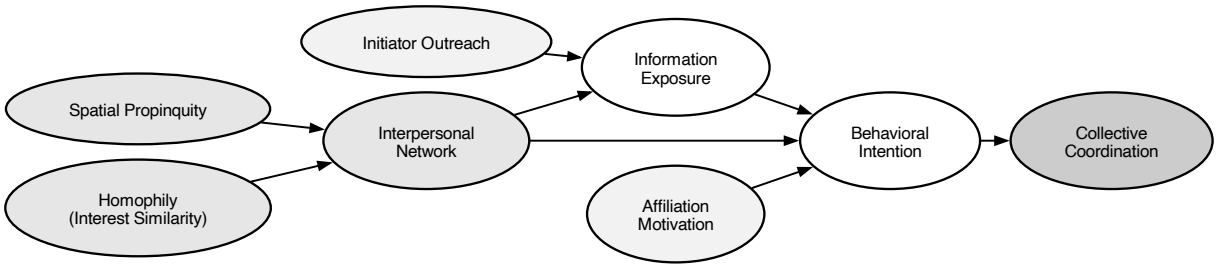


Figure 10: **Learned causal graph for agent coordination.** Visualization of a representative causal graph learned by CAMO under the *agent coordination* emergent phenomenon.

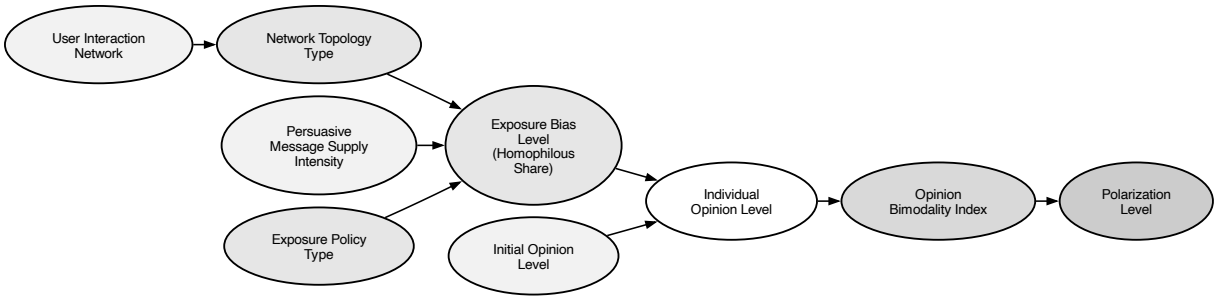


Figure 11: **Learned causal graph for opinion polarization.** Visualization of a representative causal graph learned by CAMO under the *opinion polarization* emergent phenomenon.

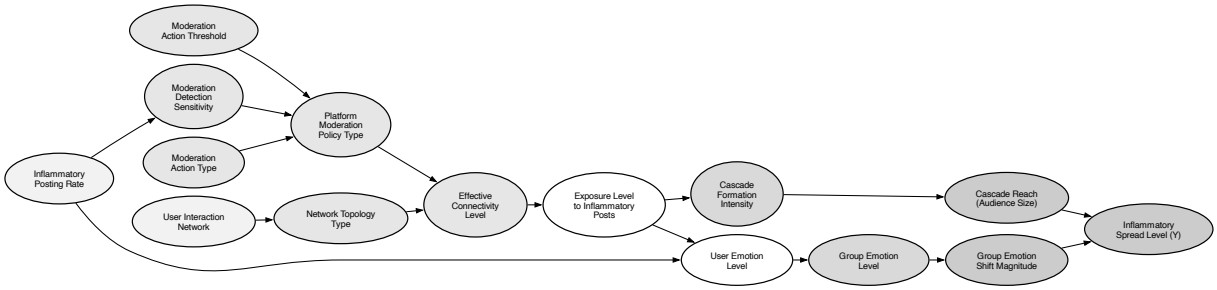


Figure 12: **Learned causal graph for inflammatory message spread.** Visualization of a representative causal graph learned by CAMO under the *spread of inflammatory messages* emergent phenomenon.

C.1.4 Successful steps and estimating $\hat{p}$ and $\hat{C}$

Because $\widehat{F}_t$ is measured with finite samples, we introduce a small tolerance $\epsilon > 0$ and call step t *successful* if the proxy difficulty decreases by at least ϵ:

$$\widehat{F}_{t+1} \leq \widehat{F}_t - \epsilon. \tag{25}$$

Success frequency $\hat{p}$. We estimate $\hat{p}$ as the fraction of successful refinement steps:

$$\hat{p} = \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{1}\left\{ \widehat{F}_{t+1} \leq \widehat{F}_t - \epsilon \right\}. \tag{26}$$

Typical contraction $\hat{C}$. On successful steps, we measure the relative decrease

$$r_t = \frac{\widehat{F}_t - \widehat{F}_{t+1}}{\widehat{F}_t}, \tag{27}$$

and estimate the typical contraction as

$$\hat{C} = \text{median}\{\, r_t \mid \widehat{F}_{t+1} \leq \widehat{F}_t - \epsilon \,\}. \tag{28}$$

C.1.5 Interpretation

$\hat{p}$ answers *how often* refinement produces a meaningful reduction in proxy residual difficulty, while $\hat{C}$ summarizes *how large* that reduction typically is when it happens. Their product $\hat{p}\hat{C}$ provides a simple summary of empirical refinement effectiveness along the A3 trajectory.

C.2 Decision Worldview Scoring and Selection

To score and select candidate decision worldviews at iteration k, A2 uses a lightweight two-stage LLM-judge procedure adapted from Zheng et al. (2023).

Stage I: rubric scoring. Given n candidates $\{W_i\}$, the judge assigns each W_i a 1–10 score under a fixed rubric (e.g., feasibility, novelty, rigor, and target alignment; with optional weights). A2 keeps only the top m candidates (typically $m \in \{3, 4\}$) to avoid exhaustive $\binom{n}{2}$ comparisons.

Stage II: pairwise comparison. On the shortlist, the judge performs pairwise comparisons and outputs a winner (or tie) with a brief justification. A2 aggregates outcomes into a ranking and selects the working set; if preference cycles occur, A2 runs one additional multi-way adjudication on the tied set to obtain a stable order. The final decision worldview $W^{(k)}$ is chosen from the working set.

C.3 Edge Prioritization for A4

This appendix specifies a simple, fully computable heuristic used by A4 to prioritize unresolved edges for simulator-based interventions.

Inputs from A3. Let $\mathcal{E}_u$ denote the set of unresolved edges after observational discovery (e.g., ambiguous orientations in a CPDAG/PAG). For each $e \in \mathcal{E}_u$, A3 provides an importance score $w_e \in [0, 1]$ and an uncertainty score $u_e \in [0, 1]$. Intuitively, w_e measures how consequential resolving e is for downstream causal accuracy, while u_e measures how weakly e is identified from observational constraints.

Importance and uncertainty signals. The importance score w_e is derived from A3's stabilized representation, for example using (i) association with the target outcome, (ii) frequency of e across repeated discovery runs. The uncertainty score u_e is derived from the learned CPDAG/PAG; in a minimal implementation, $u_e = 1$ if e is not fully oriented (or contains ambiguous PAG marks) and $u_e = 0$ otherwise.

Optionally, A3 may provide a stability signal $\hat{\sigma}_e \in [0, 1]$ by re-running observational discovery on subsampled observations and quantifying orientation variability. One simple choice is $\hat{\sigma}_e = 1 - \mathrm{agree}(e)$, where $\mathrm{agree}(e)$ is the fraction of subsamples that return the same orientation for e.

Optimism-weighted scoring. Given (w_e, u_e) and, when available, $\hat{\sigma}_e$, A4 ranks edges by the UCB-inspired heuristic

$$\mathrm{score}(e) \;=\; w_e\big(u_e + \beta\,\hat{\sigma}_e\big), \tag{29}$$

where $\beta > 0$ controls the weight on stability-based exploration. If no stability estimate is available, we set $\hat{\sigma}_e = 0$, yielding $\mathrm{score}(e) = w_e u_e$.

A4 selects the top-K edges by $\mathrm{score}(e)$ and generates intervention scripts accordingly.

D Experimental Details and Metrics

D.1 Factor Discovery Metrics

We define factor discovery metrics based on the ground-truth causal graph of the simulator. Let Y denote the target variable, $B(Y)$ its Markov boundary, and $\mathrm{An}(Y)$ the set of all causal ancestors of Y (excluding Y itself). Given a set of proposed high-level factors S, we categorize factors as:

- **MB**: $S \cap B(Y)$, factors belonging to the Markov boundary;
- **AN**: $S \cap (\mathrm{An}(Y) \setminus B(Y))$, factors that are causal ancestors but not in the boundary;
- **OT**: $S \setminus \mathrm{An}(Y)$, off-target or spurious factors.

MB recovery. We report the counts of MB, AN, and OT factors. In addition, we compute Precision, Recall, and F1 with respect to the Markov boundary $B(Y)$:

$$\begin{aligned}
\mathrm{Precision}_{\mathrm{MB}} &= \frac{|S \cap B(Y)|}{|S|}, \\
\mathrm{Recall}_{\mathrm{MB}} &= \frac{|S \cap B(Y)|}{|B(Y)|}, \\
\mathrm{F1}_{\mathrm{MB}} &= \frac{2\,\mathrm{Precision}_{\mathrm{MB}}\,\mathrm{Recall}_{\mathrm{MB}}}{\mathrm{Precision}_{\mathrm{MB}} + \mathrm{Recall}_{\mathrm{MB}}}.
\end{aligned} \tag{30}$$

Ancestor F1. To assess recovery of ancestral relations with respect to the target variable Y, we evaluate *reachability* (transitive) relations rather than ancestor node-set overlap. Let $\mathrm{TC}(G)$ denote the transitive closure of G. We define the target-specific ancestor-relation set

$$\begin{aligned}
R_Y(G) = \{(u, Y) \in V^{\mathrm{obs}} \times V^{\mathrm{obs}} :\; & u \neq Y, \\
& (u, Y) \in \mathrm{TC}(G)\}.
\end{aligned} \tag{31}$$

Let $R^*_Y = R_Y(G^*)$ and $\widehat{R}_Y = R_Y(\hat{G})$. We compute precision/recall/F1 over these ordered pairs:

$$\begin{aligned}
\mathrm{Precision}_{\mathrm{anc}} &= \frac{|\widehat{R}_Y \cap R^*_Y|}{|\widehat{R}_Y|}, \\
\mathrm{Recall}_{\mathrm{anc}} &= \frac{|\widehat{R}_Y \cap R^*_Y|}{|R^*_Y|}, \\
\text{Anc-F1} &= \frac{2\,\mathrm{Precision}_{\mathrm{anc}}\,\mathrm{Recall}_{\mathrm{anc}}}{\mathrm{Precision}_{\mathrm{anc}} + \mathrm{Recall}_{\mathrm{anc}}}.
\end{aligned} \tag{32}$$

This metric credits a method if it recovers the correct *transitive* upstream influence on Y (e.g., $A \to B \to Y$ implies $(A, Y) \in \mathrm{TC}$), regardless of whether the influence is realized by a direct edge or a multi-hop path in the predicted graph.

D.2 Projection to Observed-Variable Space

CAMO may introduce computable factor nodes (constructed variables) that do not exist in the raw log space. To fairly compare against baselines that operate only on observed variables, we evaluate graph-structure metrics on the observed-variable space by projecting any learned graph G onto $\mathcal{V}^{\mathrm{obs}}$.

Observed variables and factor supports. Let $\mathcal{V}^{\mathrm{obs}}$ be the set of logged (raw) variables. Each constructed factor node Z is associated with a *support set* $\mathrm{supp}(Z) \subseteq \mathcal{V}^{\mathrm{obs}}$, defined as the set of raw variables used by the deterministic construction rule of Z (e.g., counts, ratios, summary statistics, graph metrics). For raw variables $X \in \mathcal{V}^{\mathrm{obs}}$, we define $\mathrm{supp}(X) = \{X\}$.

Projection operator. Given a learned graph $G = (V, E)$ that may include constructed factors, we define the projected graph $\Pi(G) = (\mathcal{V}^{\mathrm{obs}}, E_\Pi)$ by removing all non-observed nodes and *lifting* each edge to observed supports. For every directed or partially oriented edge $(U \to V) \in E$, we add edges from $\mathrm{supp}(U)$ to $\mathrm{supp}(V)$:

$$E_\Pi := \bigcup_{(U \to V) \in E} \Big\{ x \to z \ : \ x \in \mathrm{supp}(U), \ z \in \mathrm{supp}(V) \Big\}. \tag{33}$$

Note that when both U and V are observed variables, the projection reduces to the original edge between them. If the learned structure is a CPDAG/-PAG with an ambiguous endpoint on $(U \circ\!\!-\!\!\circ V)$, we project it similarly while preserving ambiguity:

$$\Pi\big(U \circ\!\!-\!\!\circ V\big) = \Big\{ x \circ\!\!-\!\!\circ z \ : \ x \in \mathrm{supp}(U), \ z \in \mathrm{supp}(V) \Big\}. \tag{34}$$

Finally, we simplify $\Pi(G)$ by removing duplicate edges. Self-loops $(x \to x)$ are discarded.

Rationale. This projection attributes dependencies mediated by constructed factors back to the raw variables from which they are computed, enabling direct comparison with baselines that cannot introduce factor nodes. All graph-level structure metrics (e.g., SHD, NHD, Added/Missed/Reversed edges) are computed on $\Pi(G)$.

D.3 Causal Structure Recovery Metrics

We evaluate causal structure recovery by comparing the projected learned graph $\hat{G} = \Pi(G)$ with the ground-truth graph G^* in the observed-variable space (Appendix D.2). Let $G^* = (V^{\mathrm{obs}}, E^*)$ and $\hat{G} = (V^{\mathrm{obs}}, \hat{E})$.

Directed edge metrics. Following prior work, we evaluate edge-wise performance on the set of *directed edges*. Let $E^* \subset V^{\mathrm{obs}} \times V^{\mathrm{obs}}$ and $\hat{E} \subset V^{\mathrm{obs}} \times V^{\mathrm{obs}}$ denote the sets of ground-truth and predicted directed edges, respectively. We define

$$\begin{aligned} \mathrm{TP} &= |\hat{E} \cap E^*|, \\ \mathrm{FP} &= |\hat{E} \setminus E^*|, \\ \mathrm{FN} &= |E^* \setminus \hat{E}|, \\ \mathrm{TN} &= |V^{\mathrm{obs}}|(|V^{\mathrm{obs}}| - 1) - \mathrm{TP} - \mathrm{FP} - \mathrm{FN}. \end{aligned} \tag{35}$$

Based on these counts, we compute

$$\begin{aligned} \mathrm{Precision} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \\ \mathrm{Recall} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \\ \mathrm{F1} &= \frac{2\,\mathrm{Precision}\,\mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \end{aligned} \tag{36}$$

Accuracy. We additionally report the accuracy (Acc) over directed edges, following the edge-set evaluation in our implementation. Let $n = |V^{\mathrm{obs}}|$ and consider all ordered pairs (u, v) with $u \neq v$ as candidate directed edges. Using $\mathrm{TP}, \mathrm{FP}, \mathrm{FN}$ as defined above, we set

$$\mathrm{TN} = n(n-1) - \mathrm{TP} - \mathrm{FP} - \mathrm{FN}, \tag{37}$$

and compute

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} = \frac{\mathrm{TP} + \mathrm{TN}}{n(n-1)}. \tag{38}$$

False positive rate. We additionally report the false positive rate (FPR) over directed edges:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{39}$$

SHD and error decomposition. Let $S(G)$ denote the (undirected) skeleton of a graph G, i.e., $S(G) = \{\{u, v\} : (u \to v) \in E(G) \text{ or } (v \to u) \in E(G)\}$. We decompose the structural Hamming distance (SHD) into four components:

Table 6: AgentSociety controllable mechanisms as causal nodes (one row per node).

Node v	Controllable mechanism (simulator knob)
ExposurePolicyType	Recommendation/exposure policy family.
ExposureBiasLevel	Strength of homophilous exposure bias.
NetworkTopologyType	Clustered vs. mixing interaction topology.
PersuasiveMsgSupplyIntensity	Supply intensity of persuasive messages.
InitialOpinionShift	Shift of initial opinion distribution.
InflammatoryPostingRate	Rate of inflammatory content injection.
ModerationPolicyType	Moderation on/off policy family.
ModerationAggressiveness	Moderation strictness / action threshold.

Table 7: Smallville controllable mechanisms as causal nodes (one row per node).

Node v	Controllable mechanism (direct input edit)
SeedInfoCoverage	fraction of agents receiving the coordination seed.
SeedInfoSalience	salience/strength of injected seed memory.
PublicAnnouncement	presence of a shared public bulletin/notice.
EncounterBudget	encounter/opportunity budget for interactions.
KeyAgentSeeding	whether hub/central agents are seeded first.

- **Added**: $|S(\hat{G}) \setminus S(G^*)|$, extra adjacencies;
- **Missed**: $|S(G^*) \setminus S(\hat{G})|$, missing adjacencies;
- **Reversed**: common adjacencies whose orientation in $\hat{G}$ is the opposite of G^*;
- **Unoriented**: common ground-truth adjacencies whose direction is not resolved in $\hat{G}$ (e.g., `o-o` or `-` in CPDAG/PAG outputs).

We define

$$\text{SHD} = \text{Added}+\text{Missed}+\text{Reversed}+\text{Unoriented}. \tag{40}$$

Ancestor metrics. The computation of Anc-F1 is described in Appendix D.1.

E Interventional Ranking without Ground-Truth Graphs

This appendix specifies the evaluation protocol when no ground-truth causal graph is available. The goal is to test whether a learned graph provides *actionable guidance* by ranking effective simulator-supported interventions ahead of ineffective (placebo) ones.

E.1 Pre-registered Intervention Pool and Success Labels

For each environment–target pair, we pre-register a fixed pool of simulator-supported intervention scripts

$$S = \{s_1, \ldots, s_{|S|}\}, \quad |S| = 12, \tag{41}$$

where each script modifies exactly one controllable mechanism while keeping other settings unchanged. To increase discriminability, S intentionally contains many *placebo/weak-effect* scripts (small-magnitude changes, counteracting changes, or mechanism-irrelevant toggles), so that the set of effective scripts is sparse.

Each script is executed under multiple random seeds. A script is labeled *successful* if it induces a statistically significant change in the emergent outcome Y relative to a no-intervention baseline:

$$\text{success}(s) \in \{0, 1\}. \tag{42}$$

We denote the set of successful scripts by

$$S^+ = \{s \in S : \text{success}(s) = 1\}. \tag{43}$$

Crucially, S^+ is obtained from rollouts and is *method-independent*: no method is given S^+; all methods only output a learned graph.

E.2 From a Causal Graph to an Intervention Ranking (RWR)

Methods output a directed causal graph $G = (V, E)$ but do not directly output a ranking over scripts. We derive a comparable ranking using Random Walk with Restart (RWR), propagating influence *from the target back to upstream mechanisms*.

Script-to-node mapping. Each script s targets a mechanism node $v(s)$ defined in the intervention tables below. If $v(s) \notin V$ for a learned graph G, we assign score 0 to s (ties are broken deterministically).

Upstream propagation. Let A be the row-normalized adjacency matrix of G restricted to nodes connected to Y. We run RWR on the reversed flow:

$$\pi = \alpha e_Y + (1-\alpha)A^\top \pi, \tag{44}$$

where e_Y is one-hot on Y and $\alpha \in (0,1)$ is the restart probability. We rank scripts by $\pi(v(s))$ in descending order.

E.3 Ranking Metrics

We report ranking quality at $K = 5$ (a short actionable shortlist). Let $s_{(i)}$ be the i-th ranked script.

Precision@K.

$$\mathrm{P@K} = \frac{1}{K}\sum_{i=1}^{K} \mathrm{success}(s_{(i)}). \tag{45}$$

MAP@K.

$$\mathrm{MAP@K} = \frac{1}{\min(K, |S^+|)} \sum_{i=1}^{K} \Big(\mathrm{P@i}\Big) \cdot \mathbf{1}\big[\mathrm{success}(s_{(i)}) = 1\big]. \tag{46}$$

MRR.

$$\mathrm{MRR} = \frac{1}{\min\{i : \mathrm{success}(s_{(i)}) = 1\}}. \tag{47}$$

E.3.1 AgentSociety: Polarization and Inflammatory-Message Spread

We study two emergent targets in AgentSociety: *opinion polarization* and the *spread of inflammatory messages*. For reference, the manually annotated root causes are provided in Table 6.

E.3.2 Smallville: Agent Coordination

We evaluate on Smallville with the emergent target *agent coordination*. We use only simulator-realizable interventions implemented as direct edits to memories/public information/encounter opportunities (no hand-crafted behavior rules). For reference, the manually annotated root causes are provided in Table 7.

F Baselines

F.1 Factor Discovery Baselines

This appendix lists prompt templates for factor-discovery baselines used in § 5. All baselines use the same simulator description and target outcome Y, and output a set of candidate factors S in the same format as CAMO. Unless stated otherwise, baselines use no external causal feedback (e.g., CI tests, graph feedback, or interventions) and no post hoc pruning beyond the model's own output.

LLM-Text (single-round). Given only the simulator description and task context, the LLM proposes a list of high-level candidate factors for Y in one shot. The returned list is used directly as S.

LLM-Data (single-round). Same as LLM-Text, but additionally provides a small observational data. The one-shot output list is used directly as S.

LLM-CoT (single-round, text-only). Same inputs as LLM-Text (no observational data), but the LLM is instructed to reason before listing factors. The final list is used as S.

LLM-MultiRound (multi-round, text-only). Same inputs as LLM-Text (no observational data). The LLM iteratively revises the factor list across multiple rounds using only the dialogue history. No external verification signal is provided, and the final round's list is taken as S.

F.2 Factor Discovery Baseline Prompts

F.2.1 Text-only Prompt

> **Prompt.** You are given: (1) a description of a multi-agent simulation environment and (2) an outcome of interest Y.
> Propose a compact set of high-level factors that could causally influence Y. Each factor should be a nonlinear, computable abstraction over collective agent behavior (e.g., aggregation effects, feedback loops, distributional patterns), not a single low-level action or implementation detail.
> Output a list of factors. For each factor, provide a short name and one sentence describing how it could affect Y.

F.2.2 Data-grounded Prompt

Prompt. You are given: (1) a description of a multi-agent simulation environment, (2) observational summaries from simulation runs (e.g., samples and/or summary statistics), and (3) an outcome of interest Y.
Using the observed data, propose a compact set of high-level, computable factors that may causally influence Y. Each factor should be supported by or consistent with the observed patterns.
Output a list of factors. For each factor, provide a short name and one sentence describing how it could affect Y.

F.2.3 Chain-of-Thought Prompt (text-only)

Prompt. You are given: (1) a description of a multi-agent simulation environment and (2) an outcome of interest Y.
First, briefly reason step by step about which collective properties of agent behavior could plausibly influence Y. Then propose a compact set of high-level, computable factors.
Output only the final factor list. For each factor, provide a short name and one sentence describing how it could affect Y.

F.2.4 Multi-round Prompt (text-only, no causal feedback)

Round 1 Prompt. You are given: (1) a description of a multi-agent simulation environment and (2) an outcome of interest Y.
Propose an initial compact set of high-level, nonlinear, computable factors that may causally influence Y. For each factor, provide a short name and one sentence description.

Round $t > 1$ Prompt. Here is your current factor list. Revise it using only this conversation: merge near-duplicates, clarify ambiguous items, and optionally add missing high-level factors. Keep the final list compact and interpretable.
Output only the updated factor list (name + one sentence per factor).

F.3 Causal Structure Recovery Baselines

We compare against three classes of methods.

Statistical causal discovery (SCD) methods include PC (Spirtes and Glymour, 1991), FCI (Spirtes et al., 2000), GES (Chickering, 2002) and MMHC (Tsamardinos et al., 2006), all applied directly to observed variables. PC is a constraint-based procedure that uses conditional-independence tests to learn an equivalence class of DAGs (typically returned as a CPDAG) under causal sufficiency; FCI extends PC to allow latent confounders and selection bias, returning a PAG with possibly undirected/partially oriented edges. GES is a score-based greedy search over Markov equivalence classes that adds and then deletes edges to optimize a decomposable score (e.g., BIC), producing a CPDAG; MMHC is a hybrid approach that first performs constraint-based neighbor selection (max–min parents-and-children) to prune candidates, followed by a score-based hill-climbing phase to orient/refine edges.

Pure LLM methods include Efficient-CDLMs (Jiralerspong et al., 2024), MAC (Le et al., 2024), and PAIRWISE (Kiciman et al., 2023). Efficient-CDLMs employs a BFS-style prompting strategy that leverages the DAG structure to construct causal graphs with fewer LLM queries than exhaustive pairwise prompting. MAC formulates causal discovery as a multi-agent debate among LLMs, where agents argue for/against candidate relations and the final graph is obtained by aggregating debated judgments. PAIRWISE queries an LLM for pairwise causal direction decisions and aggregates the resulting judgments into a global graph without relying on explicit statistical tests, potentially leaving relations uncertain when evidence is weak.

Finally, *hybrid SCD+LLM methods* refine an initial SCD-produced graph using LLM reasoning, including SCD-LLM, ReAct (Yao et al., 2022), and LLM-KBCI (Takayama et al., 2024). SCD-LLM applies a single LLM pass conditioned on the SCD output (e.g., adjacency list) and variable meta-data to revise edge existence and/or directions. ReAct interleaves natural-language reasoning with tool-based actions (e.g., graph edits and external checks) to iteratively refine the SCD graph. LLM-KBCI adopts a two-stage ZS-CoT prompting framework: it first elicits explanations for each (non-)edge and then makes a final discrete decision, yielding an LLM-refined causal graph.

G Instantiation of $\mathrm{Conn}_{\min}$ by Path-Cover Closure

This appendix describes a practical instantiation of $\mathrm{Conn}_{\min}(\mathcal{R} \Rightarrow \mathrm{MB}^{\mathcal{H}}(Y))$ when we aim to preserve *all* upstream emergence pathways from $\mathcal{R}$ to each boundary variable. Here, "$\min$" denotes the *minimal closure* that preserves *all* directed paths from $\mathcal{R}$ to every $B \in \mathrm{MB}^{\mathcal{H}}(Y)$, i.e., it keeps the union of these paths and removes any nodes/edges not on any such path.

Key idea. We keep exactly the nodes and directed edges that lie on at least one directed path from any root $r \in \mathcal{R}$ to any boundary variable $B \in \mathrm{MB}^{\mathcal{H}}(Y)$ in the (constrained) retained mechanism graph. Equivalently, the connecting subgraph is the union of *all* directed $\mathcal{R} \to B$ paths, which preserves alternative micro/meso mechanisms leading to the same boundary factor.

Constrained graph. Start from the retained mechanism graph $G_W^{(k)}$ and enforce all simulation-confirmed edges as hard constraints (e.g., fixed orientations and/or edge inclusion). Let the resulting directed graph be $\widetilde{G}_W^{(k)}$.

Reachability-based path cover. Let $\mathrm{Reach}^{+}(U)$ be the set of nodes reachable from a node set U via directed edges in $\widetilde{G}_W^{(k)}$, and let $\mathrm{Reach}^{-}(V)$ be the set of nodes that can reach a node set V (i.e., reachability in the reverse graph). For each boundary variable B, define

$$\mathcal{P}(B) \;:=\; \mathrm{Reach}^{+}(\mathcal{R}) \,\cap\, \mathrm{Reach}^{-}(\{B\}). \quad (48)$$

A node is in $\mathcal{P}(B)$ iff it lies on at least one directed path from $\mathcal{R}$ to B. We then take the union over all boundary variables:

$$\mathcal{P} \;:=\; \bigcup_{B \in \mathrm{MB}^{\mathcal{H}}(Y)} \mathcal{P}(B). \quad (49)$$

We instantiate $\mathrm{Conn}_{\min}(\mathcal{R} \Rightarrow \mathrm{MB}^{\mathcal{H}}(Y))$ as the subgraph of $\widetilde{G}_W^{(k)}$ induced by $\mathcal{P}$ (keeping only directed edges). By construction, this subgraph contains *all* directed $\mathcal{R} \to B$ paths that are consistent with the enforced constraints.

Pruning. After inducing the subgraph and enforcing hard constraints, we drop any non-terminal node (excluding $\mathcal{R}$ and $\mathrm{MB}^{\mathcal{H}}(Y)$) that is not on a directed path from $\mathcal{R}$ to any boundary variable. This removes dangling components irrelevant to the emergence explanation.

Result. The resulting subgraph serves as $\mathrm{Conn}_{\min}(\mathcal{R} \Rightarrow \mathrm{MB}^{\mathcal{H}}(Y))$. Combined with $\{Y\} \cup \mathrm{MB}^{\mathcal{H}}(Y)$, it yields the explanatory subgraph E_Y used to trace micro–meso mechanisms that give rise to Y.

H O2O Delivery Platform Simulation Details

To comprehensively validate the effectiveness of our method, we construct a multi-agent simulation environment for on-demand food delivery. The simulator models a heterogeneous collaborative ecosystem composed of *Merchants*, *Riders*, and *Customers*. *Importantly, the simulator's micro-to-macro emergence patterns are explicitly instantiated from ground-truth mechanisms calibrated on real-world data, rather than being assumed implicitly or left to uncontrolled dynamics.*

H.1 Ecosystem Overview and Interaction Loop

At each discrete time step $t = 1, 2, \ldots$, the environment evolves through an event-driven pipeline:

1. **Order generation:** Customer agents generate new orders over time following **empirical trends extracted from real-world data** (e.g., time-of-day demand curves and spatial demand patterns).

2. **Merchant processing:** Merchant agents accept and prepare orders, potentially with queueing and preparation delays.

3. **Dispatching:** The Dispatch Center assigns available orders to rider agents according to a global policy.

4. **Rider response and execution:** Riders respond to dispatch instructions and execute pickup-and-delivery actions (e.g., accept/reject, reroute, reprioritize).

5. **Logging:** The simulator records states, actions, outcomes, and derived variables for subsequent causal analysis.

This setup provides a controllable testbed while preserving realistic complexities such as dynamic supply–demand imbalance, congestion, and delayed effects.

H.2 LLM-Based Cognition-Driven Decision-Making

Unlike traditional rule-based simulators, core entities in our environment are embedded with a Large Language Model (LLM) core. As a result, an agent's behavior is not a deterministic function mapping from state to action, but a stochastic process driven by semantic understanding and contextual reasoning.

A representative example is the **Rider agent**. Given (i) its role/persona, (ii) the current *explicit observable state*, and (iii) an *implicit psychological state*, the rider may nonlinearly weigh and respond to dispatch instructions (e.g., accept, reject, delay, or reorder tasks). This design intentionally introduces the real-world *intention–behavior gap*, injecting valuable noise and uncertainty that makes identifying causal chains substantially more challenging and realistic.

H.3 Multi-Level Dynamic Games and Cascading Effects

The system reproduces a complex supply–demand game with multi-level decision-making:

- The **Dispatch Center** attempts to allocate orders under a global objective (e.g., latency, throughput, service stability).
- Individual **Riders** respond according to local optimality (e.g., personal revenue, route preferences, workload considerations).

The conflict between centralized dispatching and decentralized execution, together with queueing-theoretic stochastic order arrivals, yields a highly coupled dynamical system. Small perturbations at a single node (e.g., a rider rejecting an order or a merchant delaying preparation) can propagate through the system and be nonlinearly amplified via cascading effects, making causal tracing and attribution particularly difficult.

H.4 Micro-Interactions, Macro-Emergence, and Dynamic Latent State Space

Macroscopic system performance (e.g., average delivery time, overtime rate, backlog, stability) is not determined by any single variable; instead, it emerges from nonlinear interactions among a large number of micro-level entities.

To operationalize and quantitatively detect such emergence, following the value-entropy-based indicator proposed by Yu et al. (2025b), we define the emergence score Y as the degree to which the system deviates from its optimal operating regime due to excessive disorder, measured by an edge value-entropy loss:

$$Y = 1 - \exp\left(-\frac{H_T - H_{\text{best}}}{H_{\text{best}}}\right). \tag{50}$$

Here, H_T denotes the system entropy at time step t, computed from the empirical probability distribution of *order contract deviations*, i.e., the difference between the actual delivery time and the promised delivery time, which reflects whether an order is delivered early, on time, or late. H_{best} is the optimal reference entropy corresponding to the best operating efficiency of the platform (in our calibration, $H_{\text{best}} \approx 1.1609$). When $Y = 0$, the system stays within an ordered (or mildly fluctuating) regime where the vast majority of orders can be delivered as expected, and the overall macro-state is controllable and predictable. As Y increases and approaches 1, the system exhibits severe disorder characterized by large-scale delays or capacity collapse, indicating an emergence regime where normal scheduling becomes ineffective and service quality turns highly unstable and difficult to predict.

H.5 Experimental Configuration and Scalability

In our experiments, we use a default configuration with **12 merchants**, **41 customers**, and **20 riders**. All population sizes are configurable, allowing us to scale the number of merchants/users/riders and to modify load conditions (e.g., demand intensity and service capacity) for robustness evaluation.

I Prompt Templates for CAMO Agents

I.1 A1: Worldview Parser

A1: System Prompt

```
You construct a structured, computable worldview from "fragmented facts".
You must follow the Global Contract. When {fragmented_facts} are insufficient to support recurring patterns
or latent mechanisms, you must retrieve supporting literature using available tools before answering.
```

A1: Task 1 — Multi-Stakeholders and Resources/Constraints/Goals

```
Extract multi-stakeholders relevant to {req} from {fragmented_facts}.
For each stakeholder, list resources, constraints, and goals. Keep items concise and actionable.

JSON:
{
  "stakeholders": [
    {
      "name": "string",
      "resources": ["string"],
      "constraints": ["string"],
      "goals": ["string"]
    }
  ]
}
```

A1: Task 2.1 — Behavior Patterns (with automatic evidence retrieval)

```
Identify recurring behavior patterns that appear across multiple independent sources or multiple cases.
Express each pattern as an If...Then... rule (or equivalent).
If multi-source/multi-case support is not clear from {fragmented_facts}, retrieve empirical/survey evidence via LIT_SEARCH.
Attach evidence ids to each pattern (from retrieved items or directly from {fragmented_facts} if traceable).

JSON:
{
  "patterns": [
    {
      "if_clause": "string",
      "then_clause": "string",
      "drivers": ["string"],
      "tag": "multi_sources | multi_cases",
      "evidence_ids": ["string"]
    }
  ]
}
```

A1: Task 2.2 — Latent Factors (ground mechanisms with tools when needed)

```
Derive latent motivations/mechanisms behind the extracted behavior patterns.
If mechanisms are not directly supported by {fragmented_facts}, retrieve literature evidence (LIT_SEARCH) and,
when necessary, verify details via PAPER_FETCH.
Label each factor as "directly observed", "inferred", or "severely underspecified".
Attach evidence ids for each factor.

JSON:
{
  "latent_factors": [
    {
      "name": "string",
      "description": "string",
      "label": "directly observed | inferred | severely undersspecified",
      "evidence_ids": ["string"]
    }
  ]
}
```

A1: Task 3 — Multi-Scale Structural Relationships (Micro/Meso/Macro)

```
Construct multi-scale structures (micro/meso/macro) from {fragmented_facts} and {latent_variables}.
For each scale, provide one structural assumption and map latent variables (by name) to exactly one scale.


JSON:
{
  "micro": {"structural_assumption": "string", "mapping": ["string"]},
  "meso":  {"structural_assumption": "string", "mapping": ["string"]},
  "macro": {"structural_assumption": "string", "mapping": ["string"]}
}
```

A1: Task 4 — Consistency and Stability Measurement

```
Compute the intersection and union sizes between two sets.
Treat semantically equivalent elements as identical when counting.

JSON:
{
  "intersection_size": "int",
  "union_size": "int"
}
```

I.2 A2: Worldview Integrator

A2: System Prompt

You unify and resolve conflicting worldviews arising from multiple perspectives, sources, and explanations.
You must follow the Global Contract.

A2: Task 1 — Language Unification of Indicators

Deduplicate latent variables with the same semantics but different names.
Align statistical calibers and provide a unique definition for each unified indicator.

JSON:
```
{
  "unified_indicators": [
    {
      "name": "string",
      "description": "string",
      "calculation_formula": "string",
      "data_source": "string",
      "time_window": "string"
    }
  ]
}
```

A2: Task 2 — Determine Variable Links

Determine association relationships among variables in {unified_latent_variables} using {behavior_patterns}.
All variables must be selected from {unified_latent_variables}.
If multiple relationships exist for the same pair, include all with distinct descriptions.

JSON:
```
{
  "variable_links": [
    {
      "name": "string",
      "targets": [{"name": "string", "description": "string"}]
    }
  ]
}
```

A2: Task 3 — Maintain Competing Explanations

For each (source,target) link, provide competing explanations that differ materially.
Include prerequisites, evidence, and a support estimate in [0,1].
Output must include every link, even if only one explanation exists.

JSON:
```
{
  "competition_relationship": [
    {
      "source": "string",
      "target": "string",
      "explanations": [
        {"text": "string", "prerequisites": "string", "evidence": "string", "support_estimation": "float"}
      ]
    }
  ]
}
```

A2: Task 4 — Rating of Candidate Causal Graph

Rate the causal graph from three aspects in [0,100]: fit, simplicity, explainability.

JSON:
```
{"fit":"int","simplicity":"int","explainability":"int"}
```

A2: Task 5 — Reassess Uncertain Variables (Data Feedback)

Rejudge whether uncertain variables should remain, using {gt_feedback} and A3's reasons.
Set is_important=false for at most 5 least reliable variables; set true for all others.

JSON:
```
{
  "uncertain_variable_reassess": [
    {"name":"string","is_important":"bool","reason":"string"}
  ]
}
```

A2: Task 6 — Latent Variable Regeneration

Redesign variables flagged for improvement (definition/formula/source/window).
Optionally add up to 2 new variables (must not duplicate {certainty_latent_variables}).

JSON:
```
{
  "regenerated_uncertain_indicators": [
    {"name":"string","description":"string","
      calculation_formula":"string","data_source":"string
      ","time_window":"string"}
  ]
}
```

I.3 A3: Causal Cartographer

A3: System Prompt

You assemble candidate causal structures by aligning worldview variables with dataset constraints,
then annotate edges with identifiability and produce feedback.
You must follow the Global Contract. When causal discovery methods, identifiability criteria,
or orientation rules require external evidence, you must retrieve literature (LIT_SEARCH / WEB_SEARCH)
and verify key details (PAPER_FETCH) before finalizing outputs.

A3: Task 1 — Node Adaptation

For each latent variable, decide whether it is observable, constructible, or uncertain given {data_columns}.
If constructible, provide a formal formula using only names from {data_columns}.
Variables in {certain_variables_names} must be retained as observable or constructible.

JSON:
```
{
  "variable_examination": [
    {"name":"string","flag":"observable | constructible |
      uncertain","explanation":"string","formula":"string
      "}
  ]
}
```

A3: Task 1.5 — Feedback to A2

Provide concise guidance (<=200 words) on accepted/rejected variables and how to improve variable generation.

JSON:
```
{
  "feedback_to_A2":"string",
  "accepted_variables":["string"],
  "rejected_variables":[{"name":"string","reason":"string
      "}]
}
```

A3: Task 2 — Causal Structure Generation (Merge + Tool-Triggered Method Lookup)

Merge worldview−driven and data−driven structures into a candidate structure.
Use node adaptation formulas: if a variable is constructible, add implied edges consistent with its formula.
If you need to choose or justify a causal discovery strategy (e.g., PC/GES/LiNGAM/NOTEARS/FGES/FCI, etc.),
or need rules about identifiability under assumptions, you MUST retrieve evidence via LIT_SEARCH/WEB_SEARCH first,
and optionally PAPER_FETCH for verification. Record evidence ids in each edge if such justification is used.

JSON:
```
{
  "candidate_structure": [
    {"source":"string","target":"string","description":"
      string","support_estimation":"float","flag":"world |
      data | both",
     "evidence_ids":["string"]}
  ]
}
```

A3: Task 3 — Edge Identifiability Analysis (tool-using when criteria are external)

Add an identifiability label to each edge: identifiable, assumption−dependent, or non−identifiable.
If your identifiability decision relies on formal criteria beyond the given structure/assumptions, retrieve references
via LIT_SEARCH/WEB_SEARCH and verify with PAPER_FETCH when needed. Attach evidence ids when used.

JSON:
```
{
  "edge_identifiability": [
    {"source":"string","target":"string","description":"
      string","support_estimation":"float","flag":"world |
      data | both",
     "identifiability_flag":"identifiable | assumption−
      dependent | non−identifiable",
     "evidence_ids":["string"]}
  ]
}
```

A3: Task 3.5 — Counterfactual Edge Orientation

Determine the edge direction based on counterfactual experimental results.
If you invoke any external orientation principle (e.g., invariance assumptions, SCM criteria), retrieve evidence first and attach evidence ids.

JSON:
```
{
  "oriented":"bool",
  "direction":"source_to_target | target_to_source | bidirectional | undetermined",
  "confidence":"float",
  "reasoning":"string",
  "evidence_source_to_target":"string",
  "evidence_target_to_source":"string",
  "evidence_ids":["string"]
}
```

A3: Task 4 — Feedback Iteration (Blind Spots)

Identify which parts of outcome_variable remain poorly explained by the current graph.
Return feedback items as hypotheses for missing variables/structures.

JSON:
```
{"feedback_items":["string"]}
```

I.4 A4: Simulation Scriptwright.

A4: System Prompt

You design structured simulation experiment packages to test important and uncertain causal edges.
You must follow the Global Contract.

A4: Task 1 — Edge Importance Evaluation

Add an importance score in [0,1] for each edge, reflecting sensitivity to the objective in {req}, and provide a brief explanation.

JSON:
```
{
  "edge_importance_res": [
    {"source":"string","target":"string","description":"
       string","support_estimation":"float","flag":"string",
     "identifiability_flag":"string","importance":"float","
       importance_explanation":"string"}
  ]
}
```

I.5 A5: Counterfactual Adjudicator

A5: System Prompt

You verify causal graphs using counterfactual simulation results: evaluate credibility, design interventions, verify edges, and refine graphs. You must follow the Global Contract.

A5: Task 1 — Model Credibility Evaluation

Compare simulation output with real−world observations and return a credibility score in [0,1].
If credibility is low, treat downstream findings as in−model evidence only.

JSON:
```
{"credibility_score":"float","explanation":"string"}
```

A5: Task 2 — Experiment Design

Design default settings and multiple intervention settings by varying target_variable while keeping others fixed.

JSON:
```
{
  "default_values":{"var_name":"value"},
  "experiments":[{"experiment_id":"string","description
      ":"string","settings":{"var_name":"value"}}]
}
```

A5: Task 3 — Causal Edge Verification

For each edge, assign one label:
"In−model strong support" | "In−model partial support" | "In−model refutation" | "Insufficient evidence".

JSON:
```
[
  {"source":"string","target":"string","verification_label
      ":"string","reasoning":"string"}
]
```

A5: Task 4 — Refine Causal Graph

Remove refuted edges, keep supported edges, and decide how to treat insufficient−evidence edges.

JSON:
```
[
  {"source":"string","target":"string","relation":"string"}
]
```

J Licenses and Terms

This work uses third-party artifacts (models, software libraries, and public aggregate statistics). We summarize the corresponding licenses/terms, usage, and redistribution status in Table 8. We do not redistribute any third-party proprietary artifacts

(e.g., API-served model weights); all third-party resources are accessed and used under their original licenses/terms.

Our released code and any derived artifacts (e.g., prompts and configurations) are intended solely for research and reproducibility in controlled simulation settings. We do not claim or recommend direct real-world deployment without domain-specific validation and appropriate compliance and ethical review.

Privacy note. Real-world data are used only in non-identifying, aggregate form for simulator calibration (Table 8); we do not collect, store, or release any personally identifiable information. All analyses are performed on synthetic simulation rollouts.

K Computational Budget and Infrastructure

We report the computing infrastructure and approximate resource usage for the experiments. Hardware details are summarized below, and the average LLM API consumption per trial is reported in Table 9.

K.1 Computing Infrastructure

All local computations (multi-agent coordination, environment state updates, and data processing) were executed on a dedicated server:

- **CPU:** Dual AMD EPYC 9654 (96 cores, 2.4 GHz per CPU), totaling 192 physical cores.
- **GPU:** 4× NVIDIA RTX A6000 (48 GB per card; 192 GB total).
- **RAM:** 512 GB DDR5 ECC (16 × 32 GB; 4800 MHz).
- **Software:** Ubuntu 22.04 LTS and Python-based simulation frameworks.

K.2 GPU Hours and Parallelization

GPUs were used for lightweight model inference, while CPUs handled discrete-event simulation logic and agent state management. The total GPU usage was approximately 2.5 GPU-hours.

K.3 LLM API Consumption

Table 9 reports the average number of API calls and tokens per trial, where N denotes the number of major iterative loops in the simulation.

L Parameters for Packages

We report third-party packages used for causal discovery, preprocessing, and (partial) LLM inference, together with the key parameter settings in Table 10.

Table 8: Licenses and terms for artifacts used in this work.

Artifact	Provider / Source	License / Terms	How used	Re-dist.
LLM backbones				
Qwen3-235B-A22B	Qwen (official release)	Apache License 2.0	LLM backbone	No
DeepSeek-R1-Distill-Qwen-32B	DeepSeek (official release)	MIT License	LLM backbone	No
DeepSeek-V3.2	DeepSeek (official release)	MIT License	LLM backbone	No
GPT-5 mini	OpenAI API	OpenAI Terms of Use + Service Terms + Usage Policies + Data controls	LLM backbone (API)	No
Gemma3-27B	Google DeepMind	Gemma Terms of Use	LLM backbone	No
Data / statistics				
Meituan Research (Wang et al., 2025a) aggregate statistics	Meituan-INFORMS-TSL Research Challenge repo	CC BY-NC 4.0	Simulator calibration (aggregate only)	No
Software dependencies				
SCD library (PC/FCI/GES/MMHC implementation)	causal-learn/pgmpy	MIT License	SCD baselines	No

Table 9: Average LLM API consumption per experimental trial.

Task Type	API Calls / Run	Tokens / Run
Method (Core Logic)	50–80	350,000–1,000,000
Simulation (Environment)	70–150	$(200{,}000\text{–}400{,}000) \times N$

Note: N denotes the number of major iterative loops in the simulation.

Table 10: Packages and key parameter settings.

Package / Algorithm	Implementation	Key parameters
pgmpy / PC	pgmpy.estimators.PC	significance_level=0.05
pgmpy / GES	pgmpy.estimators.HillClimbSearch	scoring_method=BicScore(df), max_iter=1e4
pgmpy / MMHC	pgmpy.estimators.MmhcEstimator	default (estimate())
causal-learn / FCI	causallearn.search.ConstraintBased.FCI.fci	chisq test, alpha=0.05
scikit-learn / Discretization	sklearn.preprocessing.KBinsDiscretizer	n_bins=5, encode=ordinal, strategy=quantile
vLLM / LLM inference (partial)	vLLM serving/decoding	temperature=0.7, quantization=AWQ, context=131072, max tokens=32768