Causal Discovery in Action: Learning Chain-Reaction Mechanisms from Interventions

Panayiotis Panayiotou PP2024@BATH.AC.UK
Department of Computer Science, University of Bath, UK

Özgür Şimşek O.SIMSEK@BATH.AC.UK
Department of Computer Science, University of Bath, UK

Editors: Bijan Mazaheri and Niels Richard Hansen

Abstract

Causal discovery is challenging in general dynamical systems because, without strong structural assumptions, the underlying causal graph may not be identifiable even from interventional data. However, many real-world systems exhibit directional, cascade-like structure, in which components activate sequentially and upstream failures suppress downstream effects. We study causal discovery in such chain-reaction systems and show that the causal structure is uniquely identifiable from blocking interventions that prevent individual components from activating. We propose a minimal estimator with finite-sample guarantees, achieving exponential error decay and logarithmic sample complexity. Experiments on synthetic models and diverse chain-reaction environments demonstrate reliable recovery from a few interventions, while observational heuristics fail in regimes with delayed or overlapping causal effects.

Keywords: causal discovery, interventions, chain-reaction systems, mechanistic causal models

1. Introduction

Physical systems are composed of interacting components whose effects propagate through structured mechanisms. While causal discovery is challenging in general dynamical systems Peters et al. (2017); Runge et al. (2019), many engineered and natural systems exhibit strongly directional, cascade-like structure: components activate sequentially, and blocking an upstream component reliably suppresses all downstream effects. Such *chain-reaction systems* arise in a wide range of settings, including mechanical safety interlocks and emergency-stop systems (Leveson, 2016), biological signaling and gene-regulatory cascades (Kauffman, 1969; Shmulevich et al., 2002), relay and logic circuits, and software or infrastructure dependency graphs. In these systems, the absence of an upstream activation propagates monotonically downstream: disabling a safety switch prevents all subsequent actuators from engaging, knocking out an upstream protein suppresses downstream gene expression, cutting power to a relay halts all dependent components, and removing a prerequisite service prevents dependent services from executing.

In this work, we study causal discovery in this restricted but practically relevant regime, and show that its strong mechanistic asymmetry enables simple and identifiable recovery of the underlying causal graph from blocking interventions. Specifically, we study *chain-reaction systems* inspired by Rube Goldberg machines (Goldberg, 2013). These systems are intentionally constructed so that objects play *asymmetric functional roles* in a directed cascade: a ball strikes a domino, the domino presses a button, the button releases a gate, and so on. See Figure 1 for an illustrative example. While the underlying physical forces are symmetric at the level of Newtonian dynamics, the system itself is purposefully directional.

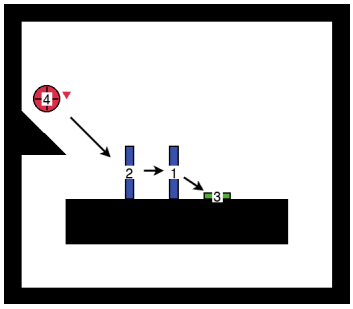


Figure 1: **Causal discovery in a chain-reaction system.** Directed edges represent *causal responsibility* rather than low-level physical forces. Holding an object in place breaks the chain reaction and deterministically suppresses all downstream activations, revealing ancestor–descendant relations.

A natural question is whether such causal structure can be inferred from observation alone. In a simple demonstration, the temporal order of object activations may reveal a clear causal chain, suggesting that tracking object motion or detecting collisions could suffice. However, temporal order does not, in general, identify causation. Distinct causal graphs can induce identical activation time sequences when multiple objects activate simultaneously or when causal effects are delayed, for example when pressing a button releases a platform at a distance after some time. Methods that track collisions have similar issues, since some interactions don't involve direct visible contact (such as a button press releasing a ball). More fundamentally, purely observational heuristics rely on strong assumptions about perception, such as accurately identifying collisions and attributing effects to the correct causes. Even the most sophisticated observational methods become brittle when multiple upstream objects provide equally plausible explanations for a downstream activation (e.g., two buttons pressed at the same time triggering different mechanisms). Such ambiguities cannot be resolved from observation alone and are precisely what causal interventions are designed to address (Eberhardt et al., 2006; Eberhardt and Scheines, 2007; Pearl, 2009).

We adopt a causal abstraction where each object is represented by a binary variable indicating whether it becomes *active* (e.g., moves) during an episode, and a directed edge $i \to j$ indicates that object i is causally responsible for triggering the activation of object j as part of the chain reaction. The goal is to recover the directed chain-reaction graph from repeated interactions with the system. We model interaction with the chain-reaction system as an interventional data collection process. In each execution (episode), the experimenter can apply a *blocking intervention* that prevents a chosen object i from activating while leaving the rest of the system intact, i.e., $\boldsymbol{do}(X_i) = 0$. Intuitively, blocking an object "breaks" the chain reaction: downstream activations that depend on that object fail to activate, revealing which components are causally downstream. Physical executions are imperfect, so we add small stochastic variations in the initial setup (slight displacements of objects when setting up the system), which can prevent the chain reaction from fully propagating, even in the absence of a blocking intervention (e.g., a domino may fall without triggering the next one). To model this stochasticity in a Structural Causal Model (SCM), we allow each variable to fail to activate with some probability even when all of its parents are active. This yields a simple SCM for chain reactions, enabling the recovery of the causal structure through blocking interventions.

We show that under this monotone cascade model, the causal structure of a chain-reaction mechanism is *provably identifiable* from single-object blocking interventions (Theorem 2). In

particular, an object j is a descendant (but not necessarily a child) of an intervened object i if and only if j is never observed to activate under $\boldsymbol{do}(X_i = 0)$. Since we assume that each object has exactly one direct trigger, the causal graph is a directed tree, and it can be obtained via transitive reduction. We propose a simple finite-sample estimator and derive theoretical guarantees, including exponential error decay and logarithmic sample complexity in the number of objects (Theorem 3).

We empirically validate the exponential error decay and logarithmic sample complexity using synthetic SCMs, where cascade failure probabilities can be directly and precisely controlled. We then evaluate our method on a suite of chain-reaction environments exhibiting parallel, delayed, and intertwined interactions. Across all settings, our approach reliably and rapidly recovers the correct causal structure from a small number of interventional samples. In contrast, observational heuristics based on temporal order or collision detection fail precisely in regimes where causal ambiguity arises, such as near-simultaneous events.

Contributions. We summarize our contributions as follows:

1. **Problem formulation.** We formalize causal discovery in chain-reaction systems via a simple causal abstraction: representing physical interactions with binary object-activation variables and a directed tree graph encoding *causal responsibility*.

2. **Structural identifiability from blocking interventions.** Under a monotone cascade structural causal model, we prove that the causal graph is uniquely identifiable from *single-object blocking interventions*, with ancestor–descendant relations revealed directly by interventional activation probabilities.

3. **Finite-sample estimation with theoretical guarantees.** We introduce a minimal estimator for recovering the graph under single-object blocking interventions and prove exponential error decay with logarithmic sample complexity in the number of objects.

4. **Evaluation on synthetic data and chain-reaction environments.** We validate the theory using synthetic causal models and also demonstrate reliable structure recovery in diverse chain-reaction environments, where observational heuristics based on temporal order or contact fail due to delayed or overlapping causal effects.

2. Related Work

Causal discovery with interventions. A central result in causal inference is that purely observational data are, in general, insufficient to completely identify causal structure, and that interventions can reduce or eliminate ambiguities (Pearl, 2009). Classical constraint-based and score-based causal discovery methods, such as PC and GES (Spirtes et al., 2000; Chickering, 2002), are defined for observational data and identify a Markov equivalence class. When interventional data are available, targeted manipulations can shrink these equivalence classes and make additional edge directions identifiable (Eberhardt et al., 2006; Hauser and Bühlmann, 2012). Building on this insight, subsequent work has developed causal discovery methods that explicitly leverage interventional data—either with known or unknown intervention targets—to further orient edges beyond what is identifiable from observations alone (Wang et al., 2017; Mooij et al., 2020; Brouillard et al., 2020). These methods aim for broad applicability across large classes of structural causal models. As a result, they may yield partial identifiability or require substantial data to resolve ambiguity, depending on the

intervention regime and modeling assumptions. In contrast, our work focuses on a restricted but meaningful class of systems in which blocking interventions induce deterministic cascade patterns. This structure enables simple, provably identifiable recovery of the full causal graph with explicit finite-sample guarantees.

Causal discovery under structural and invariance assumptions. Another line of work makes additional structural assumptions to make causal structure identifiable. Invariant causal prediction (Peters et al., 2016) exploits the stability of causal mechanisms across environments to identify causal parents, while other approaches leverage functional restrictions such as linearity with non-Gaussian noise (Shimizu et al., 2006), additive noise models (Hoyer et al., 2008), or monotonicity and ordering assumptions to simplify structure learning. Our work follows this philosophy by imposing a simple and interpretable structural assumption on how failures and activations propagate in chain-reaction systems, enabling exact identifiability and an efficient estimator from interventional data.

Deterministic and cascade-style propagation models. Deterministic and near-deterministic propagation models have been studied in several related domains, including Boolean networks (Kauffman, 1969; Shmulevich et al., 2002), reliability theory and fault-tree analysis (Haasl et al., 1981), and models of failure propagation in engineered systems (Leveson, 2016). These models capture systems in which downstream behavior is tightly constrained by upstream component states, often with stochastic noise at individual nodes. While these models are not typically framed in the language of causal discovery, they share the key structural property that downstream behavior is constrained by upstream states.

3. Problem Setup

3.1. Preliminaries

A structural causal model consists of a set of variables $X = (X_1, \dots, X_N)$, a directed acyclic graph $G^\star$ encoding causal relationships, and a collection of structural equations $X_j := f_j(X_{\mathrm{Pa}(j)}, U_j)$, where $\mathrm{Pa}(j)$ denotes the parents of node j in $G^\star$ and U_j are exogenous noise variables. An *intervention* $\boldsymbol{do}(X_i = x)$ replaces the structural equation for X_i with $X_i \equiv x$, thereby cutting all incoming edges into X_i while leaving the rest of the mechanisms unchanged (Pearl, 2009). In this work, we consider *single-node interventions* and reason about causal effects through interventional distributions of the form $\Pr(X_j \mid \boldsymbol{do}(X_i = x))$. We write $\mathrm{Desc}(i)$ and $\mathrm{Anc}(i)$ for the sets of descendants and ancestors of node i in $G^\star$, respectively.

3.2. Chain-Reaction Causal Model

We model a Rube Goldberg–style machine as a collection of N objects, each associated with a binary random variable on each execution

$$X_j \in \{0, 1\}, \qquad j = 1, \dots, N,$$

where $X_j = 1$ indicates that object j becomes *active* (e.g., moves, topples, or is pressed) at some point during the execution, and $X_j = 0$ indicates that it remains inactive. The causal structure is represented by an unknown directed graph $G^\star = (V, E^\star)$ with $V = \{1, \dots, N\}$. A directed edge $i \to j$ means that object i is *responsible for triggering* the activation of object j as part of the chain reaction.

We focus on systems that implement a directed cascade: each object has at most one direct trigger, but may trigger multiple downstream objects. Accordingly, we assume that $G^\star$ is a directed tree with a single root node. This abstraction captures the functional structure of Rube Goldberg machines, in which effects propagate along a designed sequence of components rather than through symmetric physical interactions.

3.3. Monotone Cascade Structural Causal Model

The activation variables evolve according to a *monotone cascade* SCM. For each node j, the structural equation is

$$X_j = \begin{cases} 0, & \text{if } \exists\, p \in \mathrm{Pa}(j) \text{ with } X_p = 0, \\ Z_j, & \text{otherwise,} \end{cases} \tag{1}$$

where Z_j are exogenous Bernoulli noise variables with non-zero success probability. Intuitively, Z_j captures stochastic failures in physical execution, such as small misalignments when resetting the machine (e.g., imperfectly spaced dominoes or slightly displaced objects). Thus, an object may fail to activate spontaneously even if all required upstream triggers are active. However, if any required upstream trigger fails, activation of all downstream objects is deterministically prevented.

This SCM reflects key properties of the chain-reaction systems we consider: (i) *asymmetry of responsibility*, induced by the tree structure, where each object has a unique upstream trigger and causal influence flows in a well-defined upstream–downstream direction (ii) *monotonicity*, since upstream activations can enable downstream activations, but never inhibit them, and (iii) *cascade-style propagation*, since activations propagate sequentially along directed paths and blocking an upstream object deterministically suppresses all downstream activations.

3.4. Interventions and Data Collection

We observe the system through repeated executions. In each execution e, we perform either (i) an observational run with no intervention, or (ii) a *blocking intervention* on a single object. A blocking intervention on object i is modeled as

$$\boldsymbol{do}(X_i = 0)$$

which prevents object i from activating while leaving the rest of the system unchanged. Operationally, this corresponds to physically holding an object in place and allowing the remainder of the machine to run. Under the monotone cascade model (1), this intervention deterministically forces all descendants of i to remain inactive, while non-descendants behave according to their observational distribution.

Formally, the interventional dataset consists of samples

$$\{(I_e, X^{(e)})\}_{e=1}^{M},$$

where $I_e \in \{1, \ldots, N\}$ denotes the intervened object in episode e (and $I_e = \varnothing$ indicates no intervention), and $X^{(e)} \in \{0,1\}^N$ is the observed activation vector at the end of the episode. For example, for Figure 1, a possible dataset is $\{(I_e = \varnothing,\, X^{(e)} = (1,1,1,1)), (I_e = \varnothing,\, X^{(e)} = (0,1,0,1)), (I_e = 2,\, X^{(e)} = (0,0,0,1)), (I_e = 4,\, X^{(e)} = (0,0,0,0))\}$ where blocking object i deterministically suppresses all of its descendants. A larger example is provided in Appendix D.

3.5. Learning Objective

Our goal is to recover the true causal graph $G^\star$ governing the chain-reaction system from interventional data. Given repeated executions under blocking interventions, we seek to identify the directed structure that encodes causal responsibility between objects. Crucially, we do not assume access to temporal orderings, collision events, or detailed physical state trajectories. All information is extracted from binary activation outcomes under interventions. In the following section, we show that under the monotone cascade model, single-object blocking interventions are sufficient to uniquely identify the full causal structure, and we provide a simple estimator with finite-sample guarantees.

4. Method

We now present our causal discovery method for chain-reaction systems. We first establish identifiability of the causal structure and then describe a finite-sample estimator, the reconstruction algorithm, and provide theoretical guarantees.

4.1. Identifiability from Blocking Interventions

For distinct objects $i \neq j$, define the *interventional activation probability*

$$p_{ij} \coloneqq \Pr\big(X_j = 1 \mid \textbf{\textit{do}}(X_i = 0)\big). \tag{2}$$

Intuitively, p_{ij} measures whether object j can still become active when object i is blocked. The montone cascade model allows the following characterization of descendant relations.

Lemma 1 (Deterministic cascade relation) *For any $i \neq j$,*

$$j \in \mathrm{Desc}(i) \quad \Longleftrightarrow \quad p_{ij} = 0.$$

Proof If j is a descendant of i, then blocking i forces every object on every directed path $i \rightsquigarrow j$ to remain inactive under (1), hence $X_j = 0$ and $p_{ij} = 0$. Conversely, if j is not a descendant of i, then the structural equation for X_j does not depend on X_i. Since Z_j has non-zero success probability, $X_j = 1$ remains possible under $\textbf{\textit{do}}(X_i = 0)$, implying $p_{ij} > 0$. ■

Lemma 1 shows that single-object blocking interventions reveal the full ancestor–descendant relation. We now define the *ancestor matrix*

$$A(i, j) \coloneqq \mathbb{1}\{p_{ij} = 0\}.$$

Because the true causal graph $G^\star$ is a directed tree, each node has a unique parent / closest ancestor. Once all ancestor–descendant relations are identified, the direct triggering structure (the directed tree) can be recovered by *transitive reduction*.

Theorem 2 (Identifiability) *Under the monotone cascade model and single-object blocking interventions, the true causal tree $G^\star$ is uniquely identifiable from the interventional activation probabilities $\{p_{ij}\}_{i,j}$.*

Proof Lemma 1 shows that the interventional probabilities $\{p_{ij}\}$ uniquely determine the ancestor–descendant relation. Since the true causal graph $G^\star$ is a directed tree, this relation admits a unique transitive reduction, which coincides with $G^\star$ (Aho et al., 1972). ■

Algorithm 1 Cascade Tree Reconstruction

Require: Interventional dataset $\{(I_e, X^{(e)})\}_{e=1}^{M}$
1: Compute empirical probabilities $\hat{p}_{ij}$ using (3)
2: Construct ancestor matrix $\hat{A}(i,j) = \mathbb{1}\{\hat{p}_{ij} = 0\}$
3: Enforce acyclicity of $\hat{A}$ by breaking directed cycles
4: Compute the transitive reduction of $\hat{A}$
5: **Return** reconstructed causal graph $\hat{G}$

4.2. Estimation from a Finite Interventional Dataset

Let n_i denote the number of executions in which object i is blocked. For each ordered pair (i,j), we estimate the interventional activation probability p_{ij} by the empirical frequency

$$\hat{p}_{ij} \;=\; \frac{1}{n_i} \sum_{e:I_e=i} \mathbb{1}\{X_j^{(e)} = 1\}. \tag{3}$$

Under the monotone cascade model, descendants of i are deterministically inactive under $\boldsymbol{do}(X_i = 0)$. Therefore, we classify j as a descendant of i whenever $\hat{p}_{ij} = 0$. Conversely, if object j is observed to activate at least once when i is blocked, we conclude that $j \notin \mathrm{Desc}(i)$. This yields an empirical ancestor matrix

$$\hat{A}(i,j) \coloneqq \mathbb{1}\{\hat{p}_{ij} = 0\}.$$

Behavior in the low-sample regime. When the number of interventions n_i is small, it is possible that $\hat{p}_{ij} = 0$ for many non-descendant pairs. In this regime, the estimator will overestimate the ancestor relation. However, note that the resulting errors are one-sided: true descendants are never misclassified as non-descendants under the monotone cascade model, and false positives vanish as n_i grows. As our theoretical guarantees show in Section 4.4, the probability of such spurious ancestor relations decays exponentially in n_i, and the true ancestor matrix is recovered with high probability once n_i exceeds a logarithmic threshold.

4.3. Reconstruction Algorithm

Given the estimated ancestor matrix $\hat{A}$, we reconstruct the causal graph by enforcing acyclicity and computing a transitive reduction. In the low-sample regime, we can have spurious ancestor relations including 2-cycles when both $\hat{p}_{ij}$ and $\hat{p}_{ji}$ equal zero. If observational episodes are available, such cycles can sometimes be pruned using the monotone implication that $j \in \mathrm{Desc}(i)$ would forbid observing $X_j = 1$ while $X_i = 0$, when such events are observed under stochastic failures. Any remaining cycles are broken deterministically (e.g., by index) to obtain a DAG prior to transitive reduction. These heuristics are not required for identifiability and only affect the low-sample regime. As the number of interventions grows, spurious relations vanish exponentially (Theorem 3). Algorithm 1 summarizes the reconstruction.

4.4. Theoretical Guarantees

We now quantify the finite-sample behavior of the estimator. Define the smallest non-descendant activation probability as follows

$$q_{\min} \coloneqq \min_{i \neq j:\, j \notin \mathrm{Desc}(i)} \Pr\big(X_j = 1 \mid \textbf{\textit{do}}(X_i = 0)\big), \qquad q_{\min} > 0,$$

Theorem 3 (Sample Complexity) *If $j \notin \mathrm{Desc}(i)$, then*

$$\Pr\big(\hat{A}(i,j) = 1\big) \;\leq\; \exp(-q_{\min} n_i). \tag{4}$$

That is, the probability of a false positive ancestor relation decays exponentially in the number of interventions n_i. Moreover,

$$\Pr(\hat{A} = A) \;\geq\; 1 - N(N-1)\,\exp(-q_{\min} n_{\min}), \tag{5}$$

where $n_{\min} = \min_i n_i$. Thus, if

$$n_{\min} \;\geq\; \frac{1}{q_{\min}} \Big(\log(N(N-1)) + \log \tfrac{1}{\delta} \Big),$$

then $\hat{A} = A$ with probability at least $1 - \delta$.

We provide the proof in Appendix B.

Corollary 4 (Exact Tree Recovery) *If $\hat{A} = A$, then the transitive reduction of $\hat{A}$ equals $G^\star$. In particular,*

$$\Pr(\hat{G} = G^\star) \;\geq\; 1 - \delta.$$

Inequality (4) quantifies how quickly spurious ancestor relations ($j \notin \mathrm{Desc}(i)$) disappear as more interventions are collected. For a fixed non-descendant pair (i, j), the only way j can be incorrectly classified as downstream of i is if it never activates in any of the n_i trials under $\textbf{\textit{do}}(X_i = 0)$. Since j activates with probability at least $q_{\min}$ in each such trial, this event becomes exponentially unlikely in n_i. Inequality (5) shows that, once each object is blocked a logarithmic number of times in the number of objects, with high probability no spurious ancestor relations remain and the estimated and true ancestor matrices coincide.

5. Experiments

We evaluate our method with two goals: (i) to empirically validate the finite-sample guarantees established in Section 4.4, and (ii) to demonstrate reliable causal structure recovery in chain-reaction environments inspired by Rube Goldberg machines. All experiments[1] were conducted on a shared CPU server (AMD EPYC 9454, 96 cores, 192 threads, 1.5 TB RAM). Complete experimental results are reported in Appendix C; here we present the most informative findings.

1. Code is available at https://github.com/panispani/chain-reaction-causal-discovery

5.1. Environments

We evaluate on six chain-reaction environments (Figure 2), each consisting of interacting objects simulated with the Pymunk physics engine, together with buttons that trigger delayed, non-contact effects by opening platforms (i.e., removing wall segments). To model imperfect setup and physical variability, we introduce a displacement parameter Δ: at the start of each episode, the position of every object is independently perturbed by a uniform displacement in $[-\Delta, \Delta]$. These perturbations induce stochastic execution failures and correspond to the exogenous noise variables Z_j in the monotone cascade SCM.

Each episode consists of a single *blocking intervention*. Interventions are performed in rounds: in each round, every object is intervened on exactly once, in a random order. The intervened object is held fixed by disabling its physics interactions, while the rest of the system evolves unchanged. For each episode, we record only the final binary activation vector indicating which objects became active. From observation alone, the causal structure of these environments is generally ambiguous. For example, in the Slot-machine and Parallel Trigger environments, multiple buttons may be pressed simultaneously, releasing different downstream objects at the same time. Appendix A provides worked-out execution rollouts illustrating these mechanisms.

The six environments are:

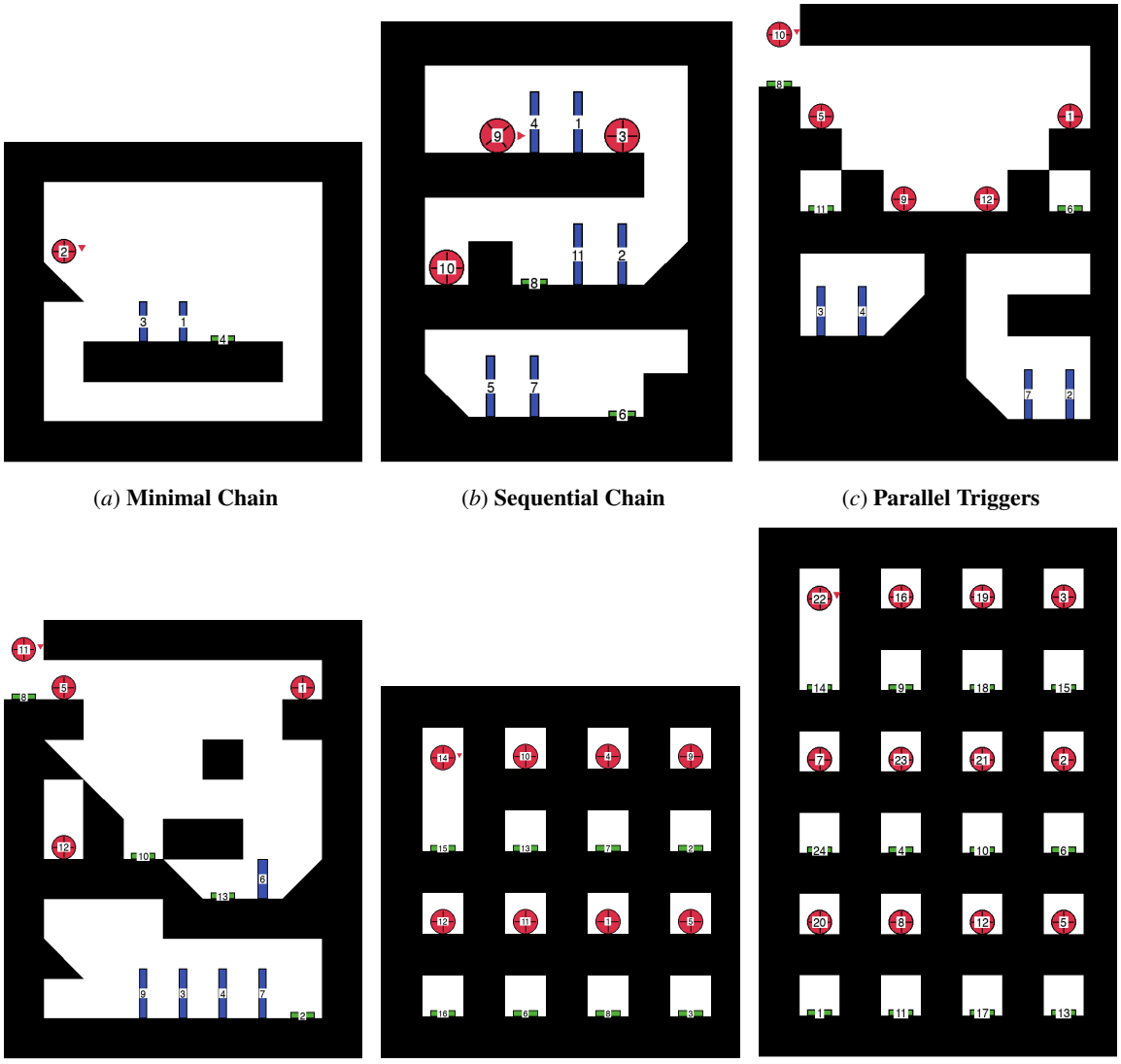
- **Minimal Chain.** A short linear cascade with four objects.
- **Sequential Chain.** A longer linear system with eleven objects.
- **Parallel Triggers.** Parallel activations cause simultaneous downstream effects.
- **Intertwined Mechanisms.** Concurrent interactions occur in different regions of the environment but work jointly for the whole system.
- **Linear Slot-Machine.** Cascades with non-contact effects via button releases.
- **Large Slot-Machine.** A larger Slot-machine variant with multiple concurrent causal branches.

5.2. Baselines

We compare against two observational heuristic baselines, each given privileged access to perfect collision detection and exact activation times. *Collision-as-influence* adds an edge $i \to j$ whenever object i collides with an inactive object j. *Temporal precedence* assigns edges based on activation order, attributing non-contact activations to the most recent collision event. Both baselines operate purely on observational data and are included to show the limits of even oracle-level perception in the absence of interventions. We also include results for the PC algorithm (Spirtes et al., 2000).

5.3. Validating the finite sample guarantees

Theoretical analysis predicts exponential decay of error with the number of blocking interventions per object. In the physical environments, we control execution noise indirectly via the displacement parameter Δ. However, Δ cannot be increased arbitrarily without causing objects to collide or overlap at initialization, which would invalidate the mechanism (e.g., adjacent objects are separated by ≈ 0.6 units in some environments, corresponding to a maximum displacement of 0.3).

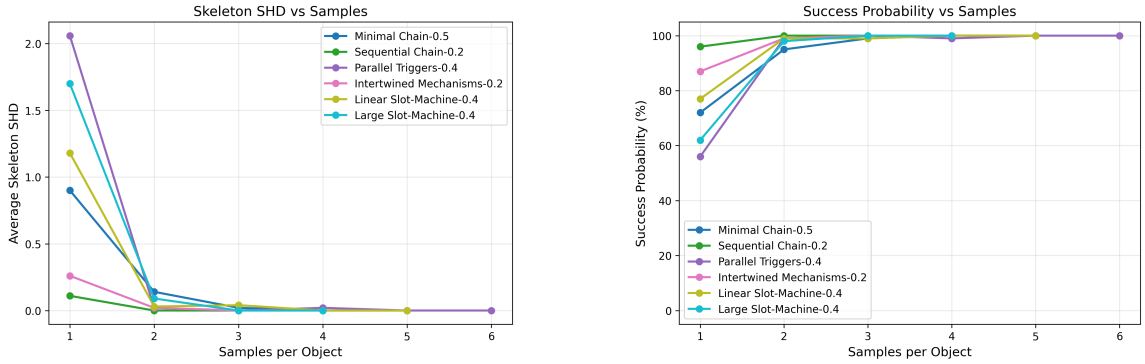


(*a*) **Minimal Chain** (*b*) **Sequential Chain** (*c*) **Parallel Triggers**

(*d*) **Intertwined Mechanisms** (*e*) **Linear Slot-Machine** (*f*) **Large Slot-Machine**

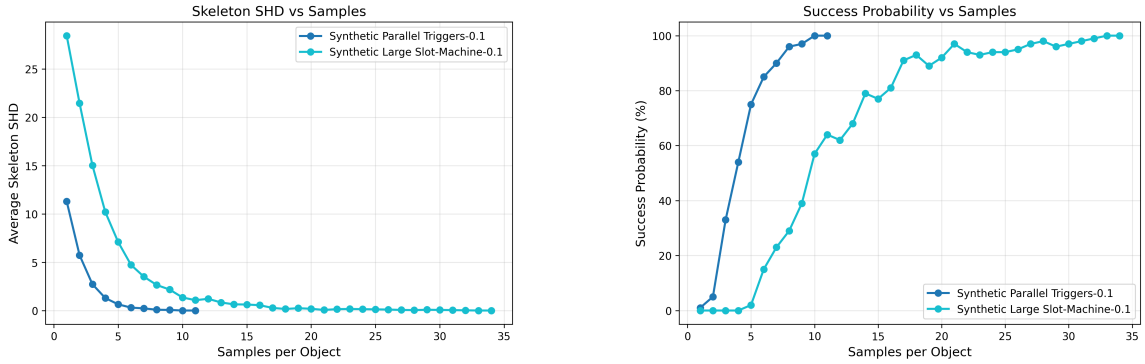
Figure 2: Collection of environments used for evaluation.

Figure 3 plots skeleton SHD and recovery probability as a function of the number of interventions per object across all environments, using the largest feasible displacement for each. Since these environments are typically solved with few samples with our method, we additionally construct synthetic SCMs using the causal graphs of "Parallel Triggers" and "Large Slot-Machine" and the monotone cascade equations (1). This allows us to directly control the failure probability of each variable. Figure 4 shows that in this controlled setting, the skeleton SHD decays exponentially with the number of interventions, and the probability of exact recovery increases exponentially (in agreement with Theorem 3).



(*a*) (*b*)

Figure 3: Scaling with the number of blocking interventions per object. (*Left*) Average skeleton SHD as a function of the number of interventions. (*Right*) Probability of exact recovery. Each curve corresponds to one environment evaluated at its largest feasible displacement Δ. We see strong sample efficiency.



(*a*) (*b*)

Figure 4: Scaling with the number of blocking interventions per object on synthetic SCMs (*Left*) Average skeleton SHD. (*Right*) Probability of exact recovery. In this controlled setting, skeleton SHD decays exponentially, and recovery probability increases exponentially with the number of interventions, in agreement with the finite-sample guarantees of Theorem 3. The number suffix denotes the Bernoulli failure parameter.

5.4. Performance on Rube Goldberg–Inspired Environments

We evaluate performance across displacement levels by progressively increasing the number of blocking interventions per object. For each environment and displacement Δ, we identify the minimum number of interventions per object required for our method to achieve at least 95% exact recovery over 100 random seeds, where exact recovery means $\hat{G} = G^\star$ after transitive reduction. We denote this quantity by $M_{\min}$. Baselines are evaluated using the same number of episodes for a fair comparison. Throughout, our method relies exclusively on interventional data.

Table 1 summarizes results at the largest displacement $\Delta_{\max}$ considered for each environment, corresponding to the noisiest setting in which the mechanism remains meaningful. We report $M_{\min}$

Environment	N	$\Delta_{\max}$	$M_{\min}$	Our (F1 / Skel. SHD)	Best baseline (F1 / Skel. SHD)
Minimal Chain	4	0.5	2	0.963 / 0.14	0.825 / 0.52
Sequential Chain	11	0.2	1	0.995 / 0.07	0.714 / 2.57
Parallel Triggers	12	0.4	2	0.999 / 0.02	0.699 / 5.55
Intertwined Mechanisms	13	0.2	2	0.999 / 0.02	0.702 / 4.95
Linear Slot-Machine	16	0.4	2	0.999 / 0.03	0.726 / 7.00
Large Slot-Machine	24	0.4	2	0.998 / 0.09	0.686 / 11.00

Table 1: **Performance at maximal displacement.** For each environment, we report the noisiest displacement $\Delta_{\max}$ tested, the minimum number of blocking interventions per object $M_{\min}$ for $\geq 95\%$ exact recovery, and performance at that setting. Our method consistently achieves near-perfect recovery using only 1–2 interventions per object, while observational heuristics have large structural errors. Full sweeps across displacements and additional metrics are in Appendix C.

together with directed-edge F1 score and skeleton structural Hamming distance (SSHD, lower is better), both for our method and for the strongest observational baseline at $\Delta_{\max}$.

Key takeaways. Two consistent patterns emerge. First, recovery is highly sample-efficient: across all environments, $M_{\min} \in \{1, 2\}$ interventions per object suffice for reliable exact recovery, even under substantial execution noise. Second, the gap to observational heuristics is qualitative rather than incremental: collision- and time-based methods exhibit large skeleton errors at $\Delta_{\max}$, reflecting fundamental causal ambiguity in the presence of parallel activations, delayed effects, and non-contact interactions. These results support the central claim of the paper: when the causal abstraction aligns with mechanistic responsibility and interventions respect the cascade semantics, causal discovery can become both simple and reliable.

6. Discussion and Limitations

Observational identifiability under stochastic failures. In our monotone cascade model, purely observational data can asymptotically emulate blocking interventions when every component fails with positive probability, since conditioning on $X_i = 0$ can reveal the same downstream suppression pattern as blocking i. But if some parts of the mechanism are deterministic, those informative "failure" events may never occur, so the causal structure can remain ambiguous even with infinite observational data (e.g., a “Large Slot-Machine” where Δ is small and balls always press the buttons).

Structural assumptions. Our method assumes that the underlying causal graph is a directed tree, so that each object has at most one direct trigger. This reflects the functional design of many chain-reaction systems, but excludes settings in which an object requires multiple parents to be activated (e.g., two objects must act together to activate a third). Extensions to such systems require interventions on sets of objects, leading to a combinatorial increase in experimental complexity. Understanding potential trade-offs between feasible intervention design and causal expressivity is an important direction for future work.

Robustness to measurement noise. Our theoretical analysis assumes activation variables are observed without error. In practice, activation must be inferred from noisy measurements, and labels may be corrupted (e.g., objects mistakenly marked as active or inactive). Our method extends to this

setting by replacing the strict criterion $\hat{p}_{ij} = 0$ with a threshold that accounts for finite samples and label noise. Robustness to measurement error is an important extension for real-world applications.

Active discovery. In our experiments, interventions are selected uniformly at random, and the outcomes of these interventions are collected into an offline dataset from which the causal graph is reconstructed. A natural next step is to pose causal discovery as a sequential decision-making problem, with the goal of minimizing the number of interventions needed to recover the graph with high confidence. This would require maintaining uncertainty over the current causal structure and prioritizing interventions that are maximally informative. This direction is particularly relevant in domains where interventions are costly, risky, or otherwise constrained.

Acknowledgements

This research was supported by the UKRI Centre for Doctoral Training in Accountable, Responsible and Transparent AI (ART-AI) [EP/S023437/1].

References

Alfred V. Aho, Michael R Garey, and Jeffrey D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. *Advances in Neural Information Processing Systems*, 33:21865–21877, 2020.

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

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of science*, 74(5):981–995, 2007.

Frederick Eberhardt, Clark Glymour, and Richard Scheines. N-1 experiments suffice to determine the causal relations among n variables. In *Innovations in machine learning: theory and applications*, pages 97–112. Springer, 2006.

Rube Goldberg. *The Art of Rube Goldberg:(A) Inventive (B) Cartoon (C) Genius*. Abrams, 2013.

David F Haasl, Norman H Roberts, William E Vesely, and Francine F Goldberg. Fault tree handbook. Technical report, Nuclear Regulatory Commission, Washington, DC (USA). Office of Nuclear, 1981.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1): 2409–2464, 2012.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21, 2008.

Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

Nancy G Leveson. *Engineering a safer world: Systems thinking applied to safety*. The MIT Press, 2016.

Joris M Mooij, Sara Magliacane, and Tom Claassen. Joint causal inference from multiple contexts. *Journal of machine learning research*, 21(99):1–108, 2020.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: Identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):947–1012, 10 2016. ISSN 1369-7412. doi: 10.1111/rssb.12167. URL https://doi.org/10.1111/rssb.12167.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT press, 2017.

Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):2553, 2019.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

Ilya Shmulevich, Edward R Dougherty, Seungchan Kim, and Wei Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2): 261–274, 2002.

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

Yuhao Wang, Liam Solus, Karren Yang, and Caroline Uhler. Permutation-based causal inference algorithms with interventions. *Advances in Neural Information Processing Systems*, 30, 2017.

Appendix A. Worked-Out Execution Rollouts

This appendix provides qualitative visualizations of the chain-reaction environments used throughout the paper. For each environment, we include frame-by-frame execution rollouts illustrating how activations propagate through the system.

How to read the figures. Each figure is organized as a sequence of frames. Frames should be read from *left to right* and then *top to bottom*, similar to a comic strip. Unless there is an intervention mentioned, it is an observational rollout.

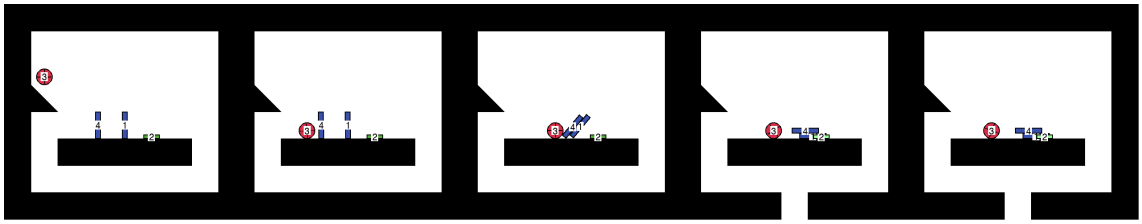


Figure 5: **Observational rollout of Minimal Chain**

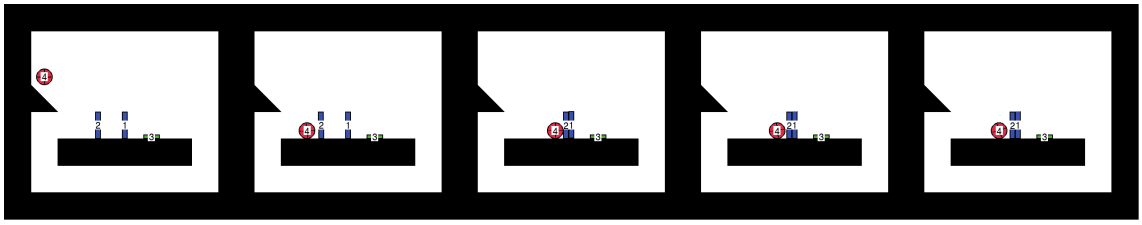


Figure 6: **Interventional rollout of Minimal Chain (intervention on object 1)**

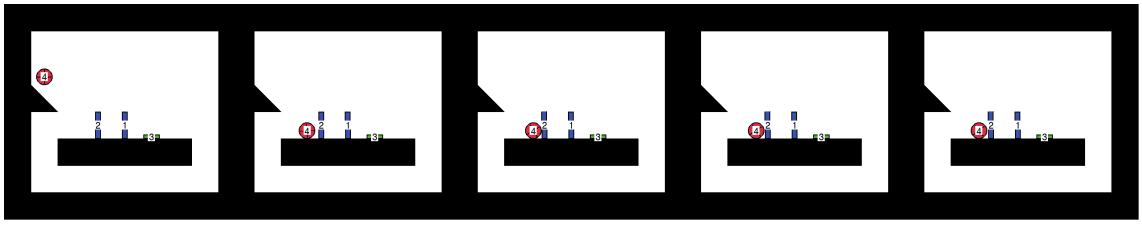


Figure 7: **Interventional rollout of Minimal Chain (intervention on object 2)**

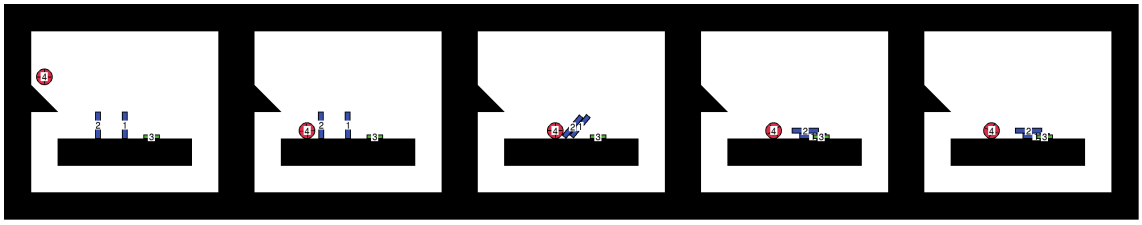


Figure 8: **Interventional rollout of Minimal Chain (intervention on object 3)**

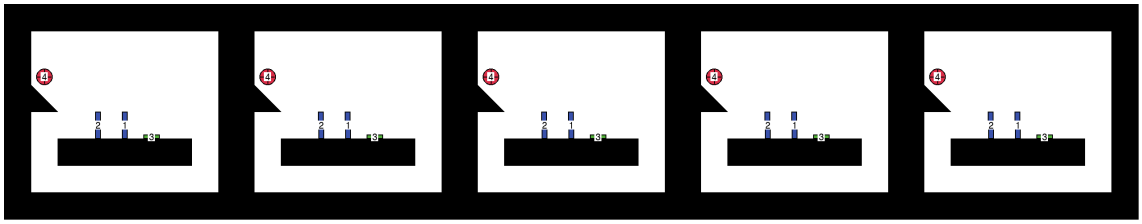


Figure 9: **Interventional rollout of Minimal Chain (intervention on object 4)**

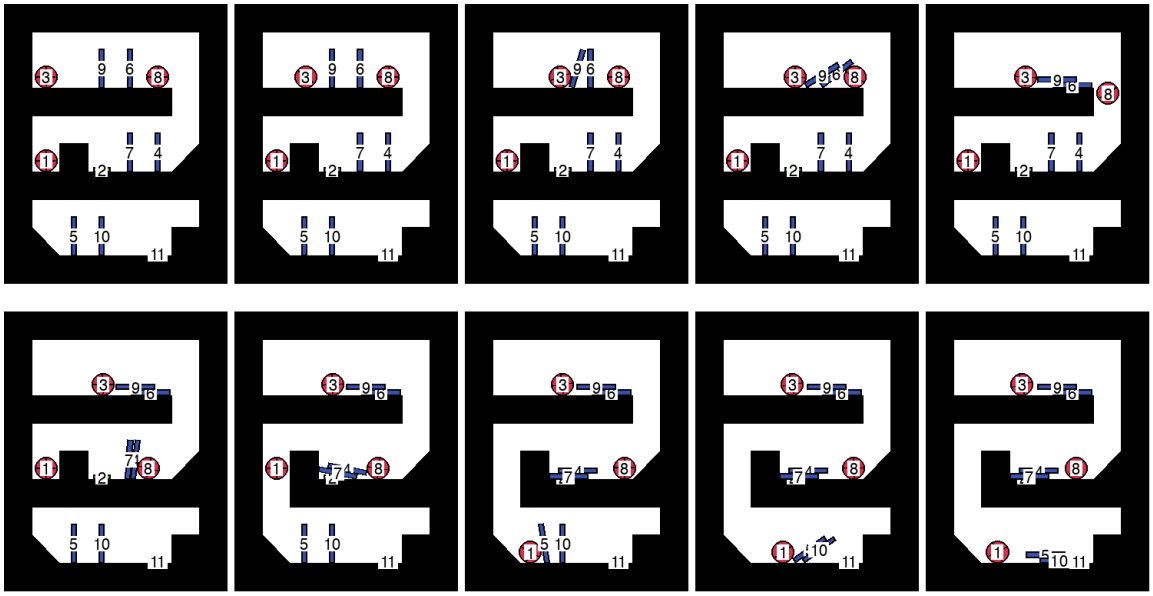


Figure 10: **Observational rollout of Sequential Chain**

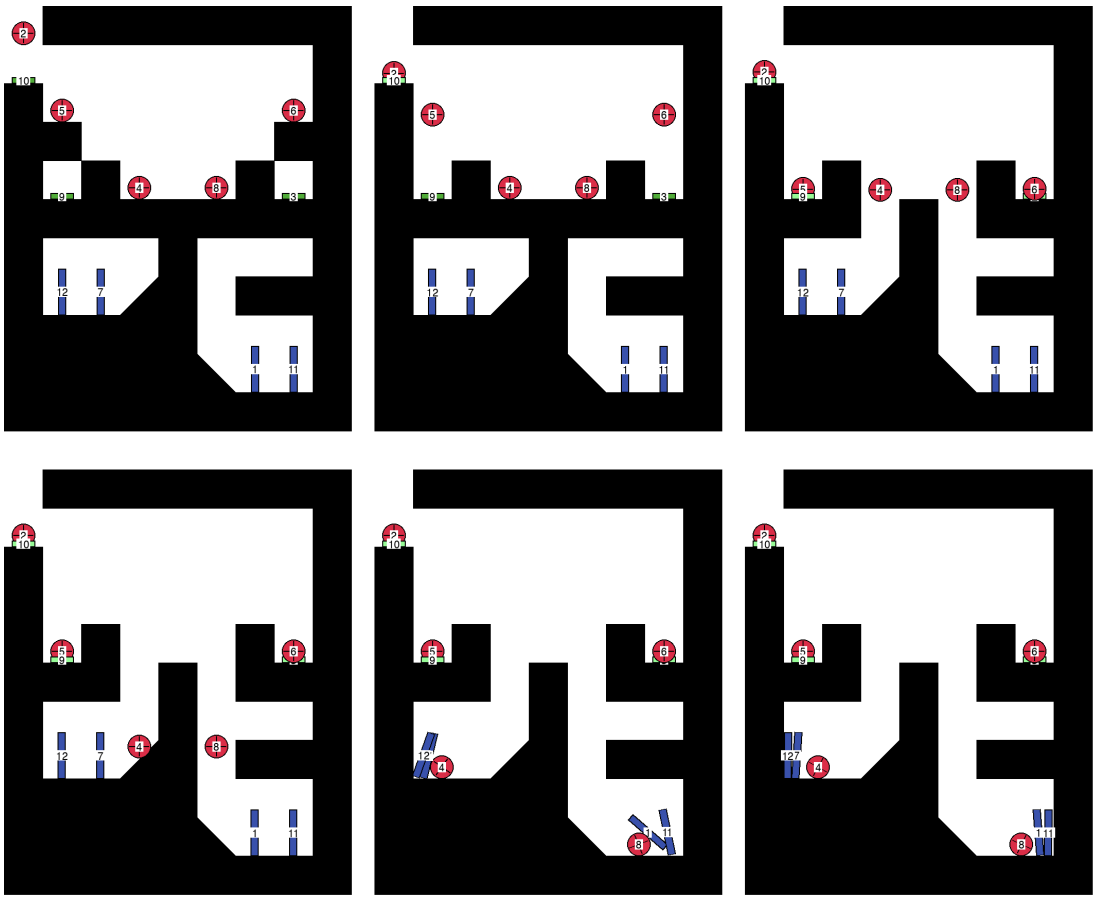


Figure 11: **Observational rollout of Parallel Triggers**

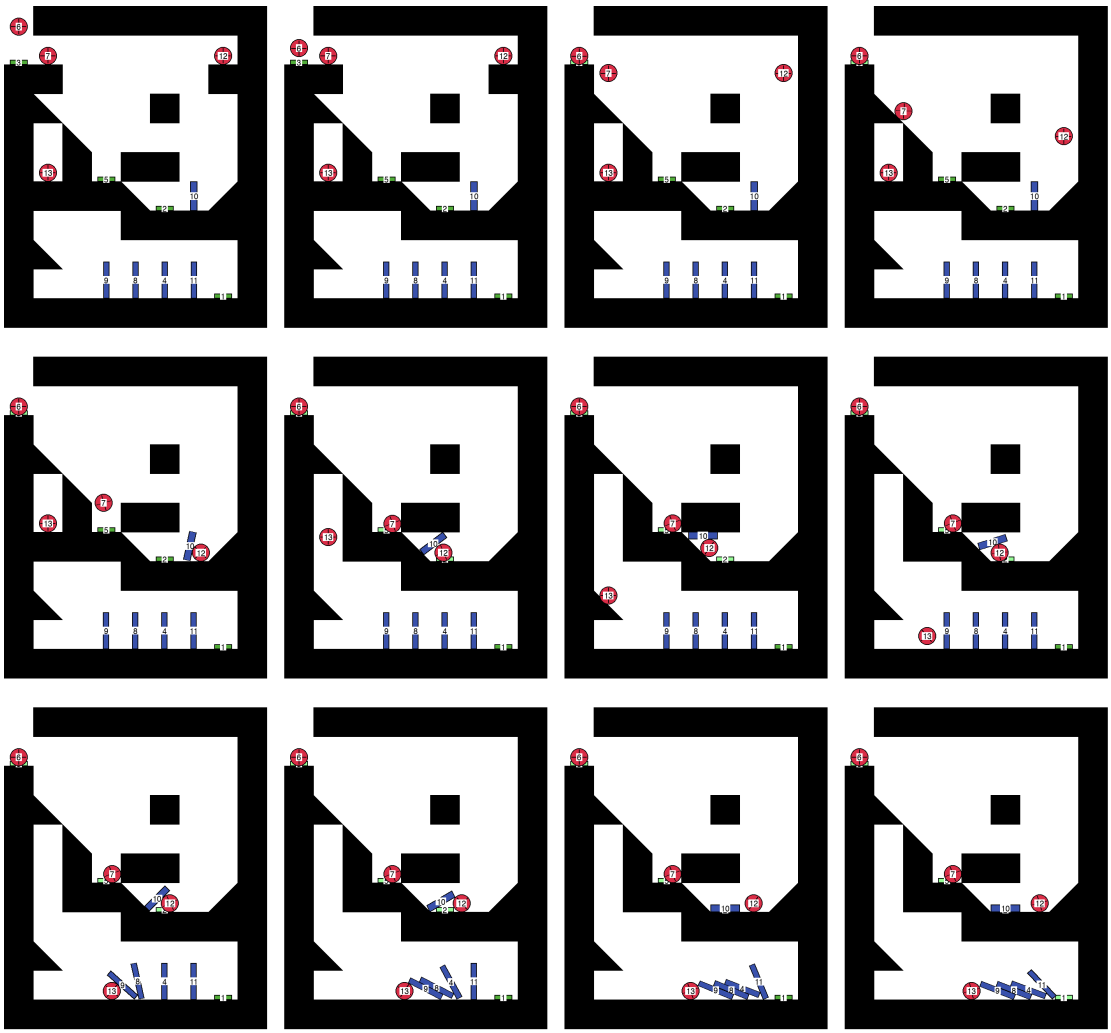


Figure 12: **Observational rollout of Intertwined Mechanisms**

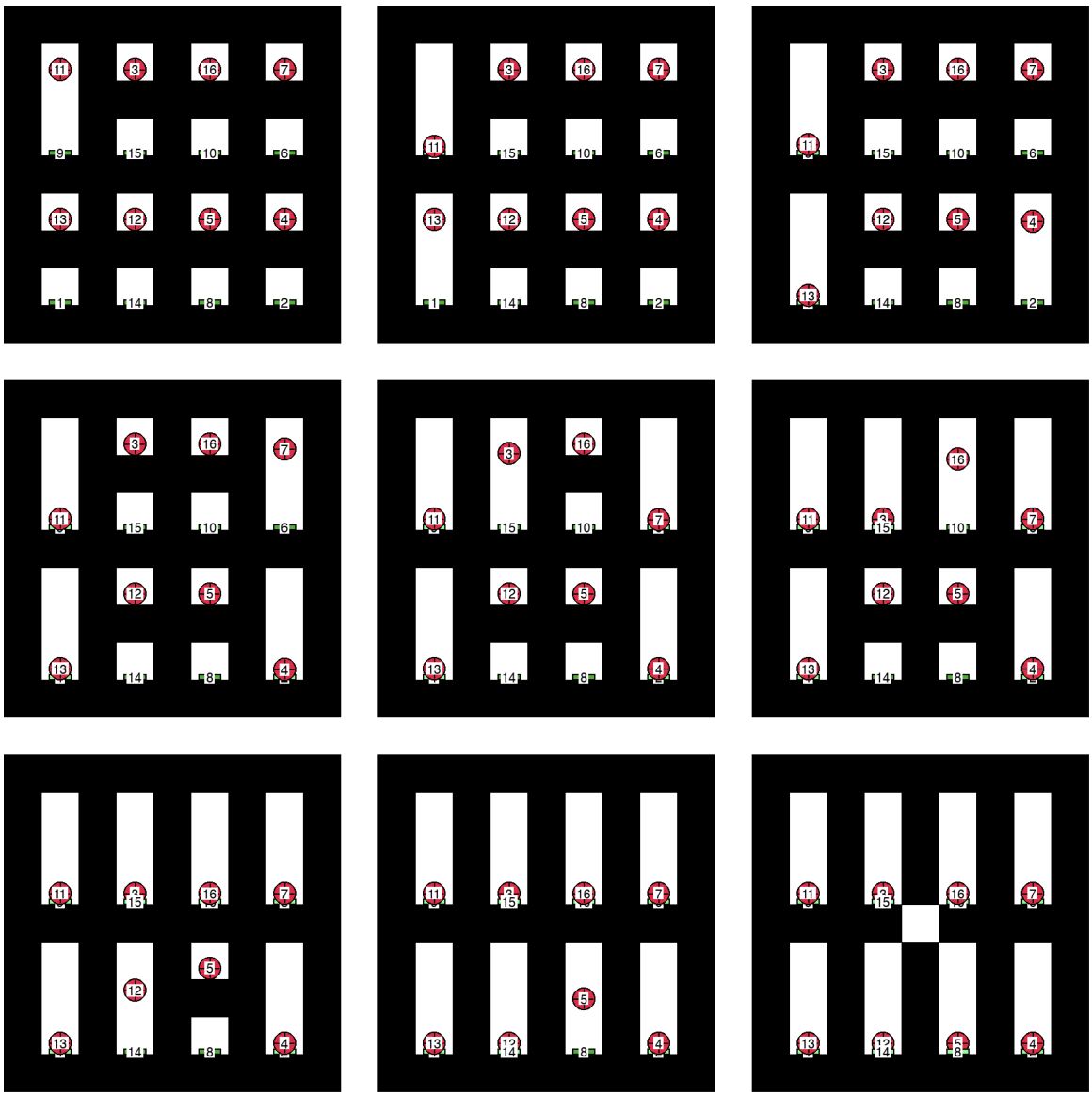


Figure 13: **Observational rollout of Linear Slot-Machine**

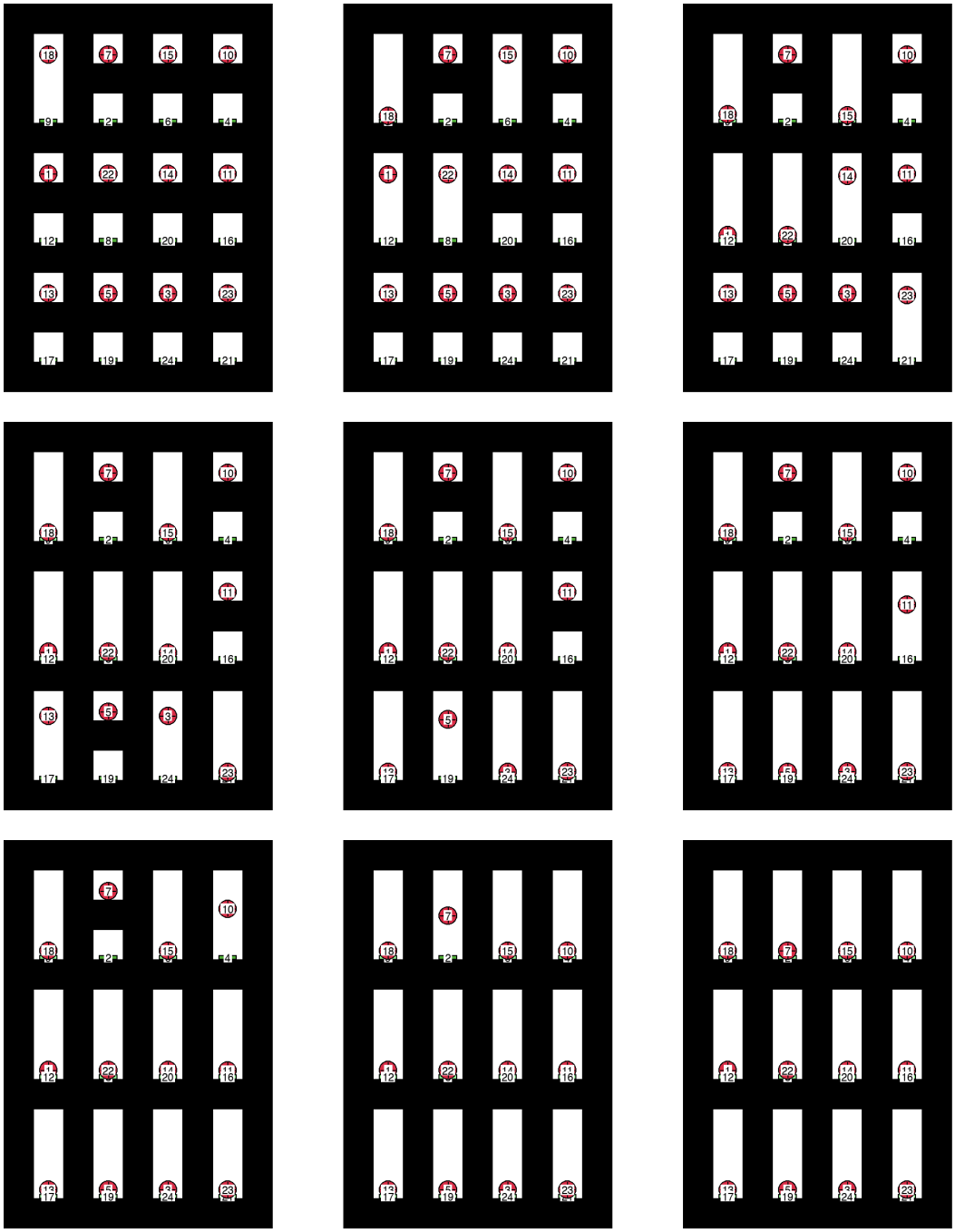


Figure 14: **Observational rollout of Large Slot-Machine**

The solutions for the six environments are:

- **Minimal chain (Figure 5):** $3 \to 4 \to 1 \to 2$.

- **Sequential chain (Figure 10):** $3 \to 9 \to 6 \to 8 \to 4 \to 7 \to 2 \to 1 \to 5 \to 10 \to 11$ (The button 2 releases ball 1 by removing the wall underneath it).
- **Parallel triggers (Figure 11):** $2 \to 10$, $10 \to 5 \to 9 \to 8 \to 1 \to 11$, $10 \to 6 \to 3 \to 4 \to 7 \to 12$
- **Intertwined mechanisms (Figure 12):** $6 \to 3$, $3 \to 7 \to 5$, $3 \to 12 \to 10 \to 2 \to 13 \to 9 \to 8 \to 4 \to 11 \to 1$.
- **Linear Slot-machine (Figure 13):** $11 \to 9 \to 13 \to 1 \to 4 \to 2 \to 7 \to 6 \to 3 \to 15 \to 16 \to 10 \to 12 \to 14 \to 5 \to 8$.
- **Large slot-machine (Figure 14):** $18 \to 9$, $9 \to 1 \to 12 \to 23 \to 21 \to 13 \to 17$, $9 \to 22 \to 8$, $9 \to 15 \to 6 \to 14 \to 20 \to 3 \to 24 \to 5 \to 19 \to 11 \to 16 \to 10 \to 4 \to 7 \to 2$.

Appendix B. Proof of Theorem 3 (Sample complexity)

Proof Fix $i \neq j$. If $j \notin \mathrm{Desc}(i)$, then under the intervention $\boldsymbol{do}(X_i = 0)$ the activation of X_j is possible with probability $p_{ij} = \Pr(X_j = 1 \mid \boldsymbol{do}(X_i = 0)) \geq q_{\min}$. Across the n_i independent executions under $\boldsymbol{do}(X_i = 0)$, the outcomes $X_j^{(e)}$ are independent Bernoulli trials with success probability p_{ij}. Therefore,

$$\Pr(\hat{A}(i,j) = 1) = \Pr(\hat{p}_{ij} = 0) = \Pr\Big(\sum_{e=1}^{n_i} X_j^{(e)} = 0\Big) = (1 - p_{ij})^{n_i} \leq e^{-p_{ij} n_i} \leq e^{-q_{\min} n_i},$$

which proves (4). Now, define the failure event

$$E \coloneqq \{\hat{A} \neq A\} = \big\{\exists\, i \neq j : \hat{A}(i,j) \neq A(i,j)\big\}.$$

If $j \in \mathrm{Desc}(i)$, then whenever the intervention $\boldsymbol{do}(X_i = 0)$ is performed, then $X_j = 0$ (Lemma 1). Since descendant pairs cannot produce errors, the failure event can only arise from non-descendant pairs. Therefore,

$$E \subseteq \bigcup_{i \neq j:\, j \notin \mathrm{Desc}(i)} \{\hat{A}(i,j) = 1\}.$$

Applying the union bound gives

$$\Pr(\hat{A} \neq A) \leq \sum_{i \neq j:\, j \notin \mathrm{Desc}(i)} \Pr\big(\hat{A}(i,j) = 1\big).$$

For any non-descendant pair (i,j), (4) says that $\Pr(\hat{A}(i,j) = 1) \leq \exp(-q_{\min} n_i)$. Using $n_i \geq n_{\min}$ for all i and noting that there are at most $N(N-1)$ ordered pairs with $i \neq j$, we obtain

$$\Pr(\hat{A} \neq A) \leq N(N-1)\, \exp(-q_{\min} n_{\min}),$$

which yields (5). To ensure $\Pr(\hat{A} \neq A) \leq \delta$, it suffices that

$$N(N-1)\, \exp(-q_{\min} n_{\min}) \leq \delta.$$

Solving for $n_{\min}$ gives

$$n_{\min} \;\geq\; \frac{1}{q_{\min}}\Big(\log\big(N(N-1)\big) + \log\tfrac{1}{\delta}\Big).$$

■

Appendix C. Additional experimental results

Section 5 describes the experimental setup and evaluation protocol. In this appendix, we report the complete set of results across all environments and displacement settings we considered in our experiments. PC is omitted from some tables because it failed with a math domain error. SSHD denotes skeleton SHD.

We estimate $\hat{q}_{\min}$ for each environment and displacement by running 1000 blocking interventions per object, computing $\hat{p}_{ij}$ for all non-descendant pairs (i, j) using the ground-truth graph, and taking the smallest $\hat{p}_{ij}$. Note that $\hat{q}_{\min}$ is estimated from a separate interventional dataset and is reported only to characterize the sample-complexity regime induced by displacement Δ, not as an input to the learning algorithm.

For the synthetic environments considered, $q_{\min}$ can instead be computed exactly from the known causal graph and the uniform node failure probability. Because these graphs are branched, the minimum non-descendant activation probability is achieved by the deepest leaf node, yielding $q_{\min} = (1-p)^L$, where L is the length of the longest directed path. For Synthetic Parallel Triggers-0.1 this gives $q_{\min} = 0.9^7 \approx 0.478$, while for Synthetic Large Slot-Machine-0.1 it is $q_{\min} = 0.9^{16} \approx 0.185$.

Method	Δ	$\hat{q}_{\min}$	Precision	Recall	F1	SHD	SSHD	Time (s)
Our Method (M=4)	0.1	0.395	0.980	0.980	**0.980**	**0.09**	**0.06**	0.312
Collision-as-influence	0.1	0.395	0.970	0.647	0.776	1.06	1.00	0.000
Temporal-precedence	0.1	0.395	1.000	0.667	0.800	1.00	1.00	0.000
PC	0.1	0.395	0.476	0.919	0.626	3.00	3.24	0.006
Our Method (M=3)	0.2	0.396	0.973	0.973	**0.973**	**0.13**	**0.10**	0.324
Collision-as-influence	0.2	0.396	0.945	0.630	0.756	1.11	1.00	0.000
Temporal-precedence	0.2	0.396	0.985	0.657	0.788	1.03	1.00	0.000
PC	0.2	0.396	0.459	0.863	0.598	3.00	3.41	0.005
Our Method (M=3)	0.3	0.408	0.977	0.977	**0.977**	**0.12**	**0.10**	0.326
Collision-as-influence	0.3	0.408	0.920	0.670	0.770	0.99	0.80	0.000
Temporal-precedence	0.3	0.408	0.963	0.693	0.801	0.92	0.83	0.000
PC	0.3	0.408	0.466	0.835	0.594	3.00	3.29	0.006
Our Method (M=3)	0.4	0.420	0.987	0.987	**0.987**	**0.06**	**0.04**	0.325
Collision-as-influence	0.4	0.420	0.912	0.737	0.807	0.79	0.56	0.000
Temporal-precedence	0.4	0.420	0.923	0.793	0.845	0.62	0.41	0.000
PC	0.4	0.420	0.453	0.762	0.563	3.00	3.39	0.006
Our Method (M=2)	0.5	0.424	0.963	0.963	**0.963**	**0.18**	**0.14**	0.290
Collision-as-influence	0.5	0.424	0.835	0.663	0.732	1.01	0.60	0.000
Temporal-precedence	0.5	0.424	0.889	0.787	0.825	0.75	0.52	0.000
PC	0.5	0.424	0.430	0.729	0.535	3.00	3.58	0.006

Table 2: Results for Minimal Chain (4 variables, 100 seeds, 16 samples)

Method	Δ	$\hat{q}_{\min}$	Precision	Recall	F1	SHD	SSHD	Time (s)
Our Method (M=1)	0.1	0.457	0.996	0.995	**0.995**	**0.08**	**0.07**	0.325
Collision-as-influence	0.1	0.457	0.666	0.683	0.674	4.44	2.27	0.000
Temporal-precedence	0.1	0.457	0.729	0.744	0.736	3.79	2.23	0.000
Our Method (M=1)	0.2	0.457	0.994	0.995	**0.995**	**0.09**	**0.07**	0.329
Collision-as-influence	0.2	0.457	0.637	0.674	0.655	4.86	2.60	0.000
Temporal-precedence	0.2	0.457	0.696	0.734	0.714	4.23	2.57	0.000

Table 3: Results for Sequential Chain (11 variables, 100 seeds)

Method	Δ	$\hat{q}_{\min}$	Precision	Recall	F1	SHD	SSHD	Time (s)
Our Method (M=1)	0.1	0.647	1.000	1.000	**1.000**	**0.00**	**0.00**	0.293
Collision-as-influence	0.1	0.647	0.870	0.588	0.701	5.00	4.47	0.000
Temporal-precedence	0.1	0.647	0.910	0.613	0.732	4.70	4.44	0.000
Our Method (M=1)	0.2	0.647	1.000	1.000	**1.000**	**0.00**	**0.00**	0.291
Collision-as-influence	0.2	0.647	0.870	0.588	0.701	5.00	4.47	0.000
Temporal-precedence	0.2	0.647	0.908	0.620	0.736	4.72	4.54	0.000
Our Method (M=1)	0.3	0.647	1.000	1.000	**1.000**	**0.00**	**0.00**	0.327
Collision-as-influence	0.3	0.647	0.853	0.594	0.699	5.16	4.69	0.000
Temporal-precedence	0.3	0.647	0.913	0.626	0.742	4.69	4.58	0.000
Our Method (M=2)	0.4	0.622	0.999	0.999	**0.999**	**0.02**	**0.02**	0.295
Collision-as-influence	0.4	0.622	0.807	0.594	0.684	5.59	5.12	0.000
Temporal-precedence	0.4	0.622	0.808	0.620	0.699	5.73	5.55	0.000

Table 4: Results for Parallel Triggers (12 variables, 100 seeds)

Method	Δ	$\hat{q}_{\min}$	Precision	Recall	F1	SHD	SSHD	Time (s)
Our Method (M=2)	0.1	0.590	0.999	0.999	**0.999**	**0.02**	**0.02**	0.299
Collision-as-influence	0.1	0.590	0.694	0.634	0.663	6.35	4.96	0.000
Temporal-precedence	0.1	0.590	0.722	0.661	0.690	6.05	4.98	0.000
PC	0.1	0.590	0.154	1.000	0.267	12.00	66.00	7.099
Our Method (M=2)	0.2	0.588	0.999	0.999	**0.999**	**0.02**	**0.02**	0.296
Collision-as-influence	0.2	0.588	0.691	0.633	0.661	6.40	5.00	0.000
Temporal-precedence	0.2	0.588	0.736	0.671	0.702	5.90	4.95	0.000
PC	0.2	0.588	0.153	0.994	0.265	12.00	66.07	7.158

Table 5: Results for Intertwined Mechanisms (13 variables, 100 seeds)

Method	Δ	$\hat{q}_{\min}$	Precision	Recall	F1	SHD	SSHD	Time (s)
Our Method (M=1)	0.1	0.465	1.000	1.000	**1.000**	**0.00**	**0.00**	0.337
Collision-as-influence	0.1	0.465	1.000	0.533	0.696	7.00	7.00	0.000
Temporal-precedence	0.1	0.465	1.000	0.533	0.696	7.00	7.00	0.000
Our Method (M=1)	0.2	0.465	1.000	1.000	**1.000**	**0.00**	**0.00**	0.300
Collision-as-influence	0.2	0.465	1.000	0.533	0.696	7.00	7.00	0.000
Temporal-precedence	0.2	0.465	1.000	0.533	0.696	7.00	7.00	0.000
Our Method (M=1)	0.3	0.465	1.000	1.000	**1.000**	**0.00**	**0.00**	0.303
Collision-as-influence	0.3	0.465	1.000	0.533	0.696	7.00	7.00	0.000
Temporal-precedence	0.3	0.465	1.000	0.533	0.696	7.00	7.00	0.000
Our Method (M=2)	0.4	0.458	0.999	0.999	**0.999**	**0.03**	**0.03**	0.297
Collision-as-influence	0.4	0.458	1.000	0.533	0.696	7.00	7.00	0.000
Temporal-precedence	0.4	0.458	0.888	0.622	0.726	7.00	7.00	0.000

Table 6: Results for Linear Slot-Machine (16 variables, 100 seeds)

Method	Δ	$\hat{q}_{\min}$	Precision	Recall	F1	SHD	SSHD	Time (s)
Our Method (M=1)	0.1	0.686	1.000	1.000	**1.000**	**0.00**	**0.00**	0.316
Collision-as-influence	0.1	0.686	1.000	0.522	0.686	11.00	11.00	0.000
Temporal-precedence	0.1	0.686	1.000	0.522	0.686	11.00	11.00	0.000
Our Method (M=1)	0.2	0.686	1.000	1.000	**1.000**	**0.00**	**0.00**	0.297
Collision-as-influence	0.2	0.686	1.000	0.522	0.686	11.00	11.00	0.000
Temporal-precedence	0.2	0.686	1.000	0.522	0.686	11.00	11.00	0.000
Our Method (M=1)	0.3	0.686	1.000	1.000	**1.000**	**0.00**	**0.00**	0.297
Collision-as-influence	0.3	0.686	1.000	0.522	0.686	11.00	11.00	0.000
Temporal-precedence	0.3	0.686	1.000	0.522	0.686	11.00	11.00	0.000
Our Method (M=2)	0.4	0.673	0.998	0.998	**0.998**	**0.09**	**0.09**	0.296
Collision-as-influence	0.4	0.673	1.000	0.522	0.686	11.00	11.00	0.000
Temporal-precedence	0.4	0.673	0.724	0.564	0.629	15.46	15.46	0.000

Table 7: Results for Large Slot-Machine (24 variables, 100 seeds)

Appendix D. Additional Dataset Example: Parallel Triggers

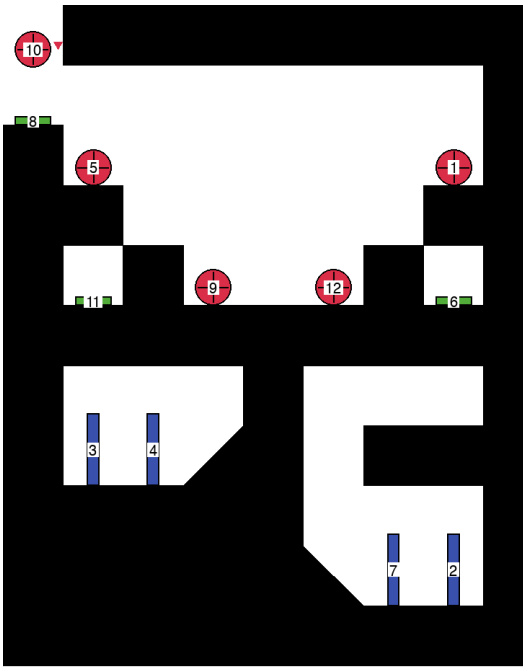


Figure 15: **Parallel Triggers.**

We provide a more detailed example of an interventional dataset (as defined in Section 3.4) for a chain-reaction system with *parallel triggers*, illustrated in Figure 15. In this system, button 8 triggers two branches in parallel: ball 5 presses button 11, releasing ball 12, while ball 1 presses button 6, releasing ball 9. Buttons 11 and 6 are typically pressed at approximately the same time. The causal structure of this system is

$$10 \to 8, \quad 8 \to 5 \to 11 \to 12 \to 7 \to 2, \quad 8 \to 1 \to 6 \to 9 \to 4 \to 3.$$

Dataset structure. Each execution e yields a pair $(I_e, X^{(e)})$, where $I_e \in \{1, \dots, N\} \cup \{\varnothing\}$ denotes the intervened object and $X^{(e)} \in \{0,1\}^N$ records which objects became active during the execution.

Example executions. A typical observational execution may produce an activation vector of the form

$$(\varnothing,\ X^{(e)}) \quad \text{with} \quad X^{(e)}_{11} = X^{(e)}_{6} = 1,$$

followed by activation of their respective downstream cascades (e.g., both balls 9 and 12 move). Under a blocking intervention such as $\boldsymbol{do}(X_{11} = 0)$, the dataset instead contains samples of the form

$$(11,\ X^{(e)}) \quad \text{with} \quad X^{(e)}_{12} = X^{(e)}_{7} = X^{(e)}_{2} = 0,$$

while the parallel branch through button 6 will be active ($X^{(e)}_{9} = X^{(e)}_{4} = X^{(e)}_{3} = 1$). Analogously, $\boldsymbol{do}(X_1 = 0)$ suppresses the branch releasing ball 9 while leaving the other branch intact. This example

illustrates how interventional datasets in chain-reaction systems capture selective suppression of downstream activations under blocking interventions, even in environments with simultaneous or parallel triggering events.

Example dataset. An example interventional dataset for this environment may contain samples such as

$$
\begin{aligned}
(\varnothing,\ X) &= (1,1,1,1,1,1,1,1,1,1,1,1),\\
(\varnothing,\ X) &= (1,0,1,1,1,1,1,1,1,1,1,1),\\
(11,\ X) &= (1,0,1,1,1,1,0,1,1,1,0,0),\\
(6,\ X) &= (1,1,0,0,1,0,1,1,0,1,1,1),\\
(10,\ X) &= (0,0,0,0,0,0,0,0,0,0,0,0),
\end{aligned}
$$

where each vector is ordered according to object indices 1 through 12. Note that in the second observational sample, the absence of activation of domino 2 may arise from stochastic failure (e.g., if domino 2 is placed too far during resetting, domino 7 may fail to reach it).