BAYESIAN REASONING AS A DYNAMIC SYSTEM: FORMAL ANALYSIS OF REACHABILITY, CONTROLLABILITY, AND REVERSIBILITY

Di Zhang
School of AI and Advanced Computing
Xi'an Jiaotong-Liverpool University
Suzhou, Jiangsu, China
di.zhang@xjtlu.edu.cn

December 11, 2025

ABSTRACT

This paper proposes a fundamental paradigm shift: re-conceptualizing static probabilistic reasoning models—particularly Bayesian networks—as **Discrete Event Dynamic Systems (DEDS)**. Through this lens, belief updating is no longer merely the computation of a posterior but rather a trajectory through a state space; evidential operations become events that trigger state transitions. We systematically establish the formal foundations of this new paradigm, focusing on three core systemic properties underexplored in traditional probability theory: **reachability** (which belief states are possible), **controllability** (how to steer the inference process through strategic evidence gathering), and **reversibility** (whether and how the system can retract from conclusions). We demonstrate that classical system analysis techniques—such as deadlock detection, supervisor synthesis, and invariance analysis—can be directly applied to Bayesian Reasoning Systems. This application reveals hidden inference pitfalls (e.g., belief deadlocks, livelocks) and enables the design of interactive reasoning protocols with guaranteed reliability. This work builds a profound bridge between probabilistic graphical models and formal methods, laying the theoretical groundwork for constructing verifiable, interpretable, and robust next-generation artificial intelligence reasoning engines.

***K*eywords** Bayesian Networks · Discrete Event Dynamic System · Formal Verification · Controllability · Reversibility · Deadlock Analysis · Interactive Reasoning · Trustworthy AI

1 Introduction: From Static Computation to Dynamic Systems

The prevailing paradigm in Bayesian network research treats reasoning as a function computation problem, focusing primarily on the development of efficient and accurate algorithms for posterior inference. This perspective, while algorithmically fruitful, fundamentally mischaracterizes the nature of reasoning in many real-world contexts. It reduces a rich, interactive process to a static input-output mapping, thereby obscuring the temporal dynamics, state-dependent evolution, and interactive nature inherent to how evidence is gathered, incorporated, and sometimes retracted to form and refine beliefs.

1.1 Limitations of the Traditional Paradigm

The traditional view is predicated on a closed-world assumption: a complete set of evidence is available upfront, and the task is to compute a single, static posterior distribution. This abstraction, however, is increasingly at odds with the requirements of modern intelligent systems. It divorces the inference algorithm from the dynamic context in which it operates. Consequently, the field lacks a formal framework to analyze questions of sequence (What happens if I observe A before B?), strategy (What evidence should I seek next to reach a diagnostic conclusion efficiently?), or

recovery (How can I backtrack from an erroneous or outdated conclusion?). The process of reasoning—the journey through belief states—is lost, with exclusive focus placed on the final destination.

1.2 Motivation for a New Paradigm

Three critical drivers necessitate this paradigm shift. First, the practical demand for interactive AI systems in domains like medical diagnosis, scientific discovery, and technical troubleshooting requires models of active, sequential reasoning. In these settings, the system is an agent that participates in a dialog with its environment or a human user, proactively seeking information and dynamically updating its hypotheses. Second, reliability challenges in complex models, especially those employing approximate inference algorithms like loopy belief propagation, expose a need for formal guarantees. Such algorithms may oscillate, converge to incorrect fixed points, or become trapped in pathological belief states—phenomena analogous to deadlocks or livelocks in concurrent systems, yet they remain outside the scope of traditional probabilistic analysis. Third, there is a growing formalist demand within explainable AI (XAI). Moving beyond post-hoc attribution techniques, we require a foundational understanding of the causal dynamics of inference: precisely how a piece of evidence influences a conclusion and, crucially, under what conditions that influence can be undone. This points toward a need for formal properties like reversibility and controllability of the reasoning process itself.

1.3 Core Idea and Contributions

This paper reconceptualizes Bayesian reasoning as a Discrete Event Dynamic System (DEDS). We formalize the belief state of a probabilistic model as the system state, and actions such as introducing evidence, retracting observations, or performing interventions as events that trigger deterministic or non-deterministic state transitions. Within this framework, a reasoning session is a trajectory through the belief state space.

Our primary contributions are fivefold. First, we establish the formal model of a Bayesian Reasoning System (BRS) as a DEDS. Second, we define and adapt core systems-theoretic properties—reachability, controllability, and reversibility—to the context of probabilistic belief states. Third, we develop algorithms for detecting inference pathologies, namely belief deadlocks and livelocks, by analyzing the reachability graph of the BRS. Fourth, we demonstrate how supervisory control theory can be used to synthesize provably correct strategies for evidence gathering, ensuring the reasoning process remains within a "safe" region of the belief space and is guided towards designated goal states (e.g., high-confidence diagnoses). Finally, we explore the profound implications of this paradigm for constructing AI systems that are not merely accurate but are verifiably robust, fundamentally more interpretable, and designed for effective human-AI collaboration.

2 Background and Related Work

2.1 Bayesian Networks and Dynamic Bayesian Networks

Bayesian networks (BNs) provide a principled graphical framework for representing joint probability distributions over a set of random variables through conditional independence relations [1, 2]. The structure of a directed acyclic graph (DAG) encodes these relationships, while the parameters quantify local dependencies. Inference in BNs, the process of computing posterior distributions given observed evidence, is a well-studied problem with exact (e.g., variable elimination, junction tree) and approximate (e.g., belief propagation, sampling) algorithms [3, 4]. Dynamic Bayesian Networks (DBNs) extend this framework to model temporal processes by unrolling the network over discrete time slices, with edges connecting variables across time [5, 6]. While DBNs explicitly model the evolution of the *world state*, their inference algorithms typically compute beliefs over these states given a fixed sequence of observations. Crucially, our work addresses a different axis of dynamics: we consider a *fixed* underlying graphical model and study the evolution of the *reasoner's belief state* itself, driven by a sequence of evidence operations (introduction, retraction). This focus on the meta-level dynamics of the inference process, rather than the object-level dynamics of the domain being modeled, distinguishes our approach from DBNs.

2.2 Discrete Event Dynamic Systems and Supervisory Control

Discrete Event Dynamic Systems (DEDS) are systems whose state evolution is driven by the occurrence of discrete, asynchronous events over time. They are characterized by a discrete state space and event-driven transitions, making them suitable for modeling concurrent software, manufacturing systems, and communication protocols. The Ramadge-Wonham (RW) supervisory control framework provides a formal theory for controlling such systems [7, 8]. In this framework, the plant is modeled as an automaton generating a language over an alphabet of events, some of which are

controllable (can be enabled/disabled) and others uncontrollable. A supervisor observes the system's behavior and dynamically restricts the set of enabled controllable events to ensure the system's behavior remains within a prescribed, desirable language (specification). Petri nets offer an alternative, graphical modeling formalism for DEDS, particularly adept at representing concurrency, synchronization, and resource allocation, with analysis techniques based on invariants and reachability graphs [9]. This paper adapts these powerful formalisms–the state/event perspective of automata theory, the control synthesis of the RW framework, and the structural analysis of Petri nets–to model and analyze the dynamic process of probabilistic reasoning.

2.3 The Rise of Formal Methods in Machine Learning

There is a growing intersection between formal methods and machine learning, primarily focused on providing guarantees for learned models. This includes work on formal verification of neural network properties (e.g., robustness to adversarial perturbations, safety constraints), synthesis of provably correct controllers from learned components, and formal specification for machine learning components [10, 11]. However, this body of work has predominantly targeted deterministic function approximators, like deep neural networks, often in supervised or reinforcement learning settings. The dynamic, stateful process of iterative belief updating in probabilistic graphical models–a cornerstone of reasoning under uncertainty–has largely eluded such formal treatment. Our work fills this gap by providing a formal dynamic systems model for the inference engine itself, enabling the application of verification and synthesis techniques directly to the reasoning process.

2.4 Reversible Computing and Causal Inference

The concept of reversibility originates in physics and computation. In reversible computing, every computational operation must be logically invertible, minimizing energy dissipation [12, 13]. While our use of "reversibility" is not strictly thermodynamic, it shares the philosophical core of traceability and backtracking. We investigate whether and how a sequence of belief updates can be "undone" to return to a prior epistemic state. This connects deeply with causal inference, particularly the do-calculus and theory of interventions [14]. An intervention, $\text{do}(X = x)$, forcibly sets a variable, altering the underlying data-generating mechanism and producing a different causal effect than passive observation. The question of reversing an intervention is a causal one. By incorporating interventions as first-class events in our DEDS model, we enrich the notion of reversibility beyond simple retraction of evidence to include the reversal of causal effects, bridging epistemic and causal dynamics in a unified formal framework.

3 Formal Model of a Bayesian Reasoning System

3.1 Basic Definitions

We begin by formalizing the core components of a Bayesian Reasoning System (BRS). Let $\mathcal{G}$ be a fixed Bayesian network structure over variables $\mathbf{X} = \{X_1, \ldots, X_n\}$, with a corresponding parameterization defining a joint probability distribution $P_\theta(\mathbf{X})$.

The *belief state space* $\mathcal{B}$ represents the set of all possible epistemic states of a reasoner operating on $\mathcal{G}$. A belief state $b \in \mathcal{B}$ is a complete specification of the reasoner's beliefs about all variables. In its most explicit form, b is the joint posterior distribution $P(\mathbf{X} \mid \mathcal{E})$, where $\mathcal{E}$ denotes the accumulated evidence. Due to the conditional independencies in $\mathcal{G}$, b can be efficiently represented in factorized form (e.g., as a set of marginals or messages) [2]. For computational analysis, $\mathcal{B}$ may be discretized by considering intervals of probabilities or by abstraction.

The *event alphabet* $\Sigma = \Sigma_c \cup \Sigma_u$ comprises actions that trigger belief state transitions. It is partitioned into controllable (Σ_c) and uncontrollable (Σ_u) events.

- **Evidence Introduction** $E_{i,v} \in \Sigma_c$: The event of observing that variable X_i takes value v.
- **Evidence Retraction** $R_i \in \Sigma_u$: The event of withdrawing a previously introduced observation on X_i, reverting it to an unobserved variable. This models the correction of erroneous data or the reconsideration of information.
- **Intervention** $\text{do}(X_i = v) \in \Sigma_c$: The event of performing a causal intervention, forcing X_i to value v, which modifies the underlying graphical model for subsequent reasoning [14].

The *state transition function* $\delta : \mathcal{B} \times \Sigma \to \mathcal{B}$ defines the dynamics. For a belief state b and an event σ, $\delta(b, \sigma)$ yields the updated belief state. For $E_{i,v}$, $\delta(b, E_{i,v})$ is computed via standard Bayesian conditioning. For R_i, the update is nontrivial and typically requires recomputing the posterior while ignoring the retracted evidence, potentially leveraging

saved prior states. For $\text{do}(X_i = v)$, the update involves applying the *do*-calculus, which may structurally alter the factorization of beliefs [14].

The system has a distinguished *initial state* b_0, corresponding to the prior belief $P_\theta(\mathbf{X})$ before any evidence is incorporated. Finally, a set of *marked states* $\mathcal{B}_m \subseteq \mathcal{B}$ specifies desirable or terminal reasoning configurations, such as states where the marginal probability of a target hypothesis exceeds a threshold ("diagnosis complete").

3.2 BRS as a Discrete Event Dynamic System

A Bayesian Reasoning System is formally defined as a Discrete Event Dynamic System (DEDS) [8]:

$$\mathcal{D} = (\mathcal{B}, \Sigma, \delta, b_0, \mathcal{B}_m).$$

The behavior of $\mathcal{D}$ can be analyzed through its generated language and reachability graph. The *generated language* $L(\mathcal{D})$ is the set of all finite sequences of events (strings in Σ^*) that are feasible from b_0, i.e., $\sigma = \sigma_1\sigma_2 \dots \sigma_k \in L(\mathcal{D})$ if $\delta(b_0, \sigma)$ is defined, where δ is extended recursively to strings in the natural way. This language captures all possible evidence-intervention-retraction histories.

The *reachability graph* $G_\mathcal{D} = (V, E)$ is a directed graph where the vertex set $V \subseteq \mathcal{B}$ is the set of states reachable from b_0 via any sequence in $L(\mathcal{D})$, and an edge $(b, b') \in E$ labeled with σ exists if $b' = \delta(b, \sigma)$. This graph is the central object for analyzing the dynamic properties of the reasoning process, such as reachability between beliefs, the existence of cycles (livelocks), and states with no path to a marked state (deadlocks) [15].

3.3 Computational Considerations and Abstraction

The belief state space $\mathcal{B}$ for a non-trivial network is a continuous, high-dimensional space. An exhaustive exploration of the reachability graph $G_\mathcal{D}$ is therefore computationally intractable. To enable formal analysis, we require principled abstraction techniques that yield a finite, tractable representation while preserving the properties of interest.

We propose two complementary abstraction methods. The first is *evidence equivalence abstraction*. Two belief states b and b' are considered equivalent if they are conditioned on the same *set* of active evidence variables (regardless of their specific values). This abstracts away the continuous probability values, focusing on the structural aspect of which variables are observed. Transitions between these equivalence classes are defined by the events that change the evidence set. While coarse, this abstraction directly aligns with the event alphabet and can perfectly capture properties related to the sequence and retraction of evidence.

The second method is *information-geometric abstraction*. Here, the continuous belief space is partitioned into regions based on proximity measured by an information-theoretic divergence, such as the Kullback-Leibler (KL) divergence [16]. States within an ϵ-ball in this metric are collapsed into a single abstract state. The transition function is then defined conservatively: an abstract transition exists if there exists at least one pair of concrete states (one in the source abstract state, one in the target) connected by a concrete transition. This method preserves probabilistic distances and is suitable for analyzing convergence and stability. The choice of abstraction and its granularity (ϵ) provides a trade-off between computational feasibility and the precision of the verified properties [17].

4 Reachability Analysis and Belief Deadlock

4.1 The Reachable Set of Belief States

For a Bayesian Reasoning System $\mathcal{D}$, we define its *reachable set* $\mathcal{R}(\mathcal{D})$ as the set of all belief states attainable from the initial state b_0 via any finite sequence of admissible events:

$$\mathcal{R}(\mathcal{D}) = \{b \in \mathcal{B} \mid \exists \sigma \in L(\mathcal{D}) \text{ such that } \delta(b_0, \sigma) = b\}.$$

The structure of $\mathcal{R}(\mathcal{D})$ is fundamentally shaped by two factors: the topology of the underlying Bayesian network and the inference algorithm implementing the transition function δ.

For *singly-connected networks* (trees), exact inference algorithms such as belief propagation provide a deterministic, closed-form δ [1]. In this setting, the reachable set $\mathcal{R}(\mathcal{D})$ has a well-defined structure: it corresponds precisely to the set of all posteriors obtainable by conditioning the prior on any possible subset of the variables (with any combination of values). The reachability graph for a tree is a deterministic acyclic graph modulo retractions.

In contrast, for *multiply-connected networks* (graphs with loops), the situation is more complex. When using *exact* algorithms (e.g., junction tree), $\mathcal{R}(\mathcal{D})$ remains a precisely defined, deterministic set of exact posteriors [3]. However,

when *approximate* algorithms like Loopy Belief Propagation (LBP) are used, the transition function δ becomes non-deterministic or algorithm-dependent. LBP may converge to different fixed points depending on message scheduling or may oscillate [4]. Consequently, $\mathcal{R}(\mathcal{D})$ is no longer a simple set of model-theoretic posteriors; it expands to include these algorithm-specific fixed points and may even include transient states within oscillation cycles. This algorithmic dependence introduces new dynamical phenomena not present in the exact case, making reachability analysis both more challenging and more critical.

4.2 Formal Definition and Detection of Belief Deadlock

A core contribution of our dynamical systems perspective is the identification and formalization of pathological reasoning states analogous to deadlocks in concurrent systems.

Definition 1 (Belief Deadlock)**.** *A belief state $b \in \mathcal{R}(\mathcal{D})$ is a* belief deadlock *if and only if:*

1. *$b \notin \mathcal{B}_m$ (b is not a marked/goal state).*
2. *For all finite event sequences $\sigma \in \Sigma^*$ such that $\delta(b, \sigma)$ is defined, $\delta(b, \sigma) \notin \mathcal{B}_m$.*

In other words, once the system enters a belief deadlock state, no continuation of the reasoning process—through any combination of further evidence, retractions, or interventions—can lead it to a goal state.

Belief deadlocks can arise from several sources:

- **Evidence Contradiction**: The accumulated set of observations may be statistically incompatible under the model (e.g., due to faulty sensors or rare events), leading to a posterior where no coherent hypothesis has high probability, and new evidence only deepens the conflict.
- **Approximation Fixed Points**: An approximate inference algorithm may converge to a stable fixed point that is far from the true posterior and from which local belief updates (events) cannot dislodge the system. This is an artifact of the algorithm, not the model [4].
- **Model Misspecification**: The underlying graphical model $\mathcal{G}$ may be fundamentally incorrect for the domain. Certain evidence patterns can force beliefs into a corner of the parameter space from which, given the flawed model structure, recovery to a satisfactory conclusion is impossible.

Detection of belief deadlocks is performed algorithmically on the (abstracted) reachability graph $G_{\mathcal{D}} = (V, E)$. We present the detection procedure in pseudocode format:

Algorithm 1: Belief Deadlock Detection

Input: Reachability graph $G_{\mathcal{D}} = (V, E)$, set of marked states $\mathcal{B}_m \subseteq V$

Output: Set of deadlock states $D \subseteq V$

1. Initialize $D \leftarrow \emptyset$

2. For each state $b \in V$: a. If $b \notin \mathcal{B}_m$: i. Perform a graph search (e.g., BFS/DFS) starting from b to compute the set of states reachable from b, denoted $\mathcal{R}(b)$. ii. If $\mathcal{R}(b) \cap \mathcal{B}_m = \emptyset$: $D \leftarrow D \cup \{b\}$

3. Return D

4.3 Belief Livelock

A related but distinct pathology is the *belief livelock*. While a deadlock is a static trap, a livelock involves non-progressive cyclical behavior.

Definition 2 (Belief Livelock)**.** *A set of belief states $L \subset \mathcal{R}(\mathcal{D})$ forms a* belief livelock *if:*

1. *$L \cap \mathcal{B}_m = \emptyset$ (No state in L is a goal state).*
2. *The subgraph of $G_{\mathcal{D}}$ induced by L is strongly connected (every state in L is reachable from every other state in L via a path within L).*
3. *No state in L has a transition leading to a state outside L that is on a path to $\mathcal{B}_m$. The system, once in L, is confined to cycle indefinitely within it without making progress toward a goal.*

A canonical example occurs with Loopy Belief Propagation on a tight loop, where messages may oscillate perpetually between a few configurations, never converging [4]. The system transitions cyclically between the states in L corresponding to the oscillation's phase.

Detection of livelocks leverages the decomposition of the reachability graph into its Strongly Connected Components (SCCs). The detection procedure is as follows:

Algorithm 2: Belief Livelock Detection via SCC Analysis

Input: Reachability graph $G_{\mathcal{D}} = (V, E)$, set of marked states $\mathcal{B}_m \subseteq V$

Output: Set of livelock SCCs L_{set}

1. Compute all Strongly Connected Components (SCCs) of $G_{\mathcal{D}}$, denoted $\{S_1, S_2, \ldots, S_k\}$ [18].

2. Initialize $L_{\text{set}} \leftarrow \emptyset$

3. For each SCC S_i: a. If $|S_i| > 1$ or ($|S_i| = 1$ and the single state has a self-loop): i. If $S_i \cap \mathcal{B}_m = \emptyset$: ii. Check if S_i is a *terminal SCC*: no edge from any state in S_i leads to a state in an SCC S_j that can reach $\mathcal{B}_m$. iii. If S_i is a terminal SCC: $L_{\text{set}} \leftarrow L_{\text{set}} \cup \{S_i\}$

4. Return L_{set}

The detection of deadlocks and livelocks provides formal criteria for identifying when a Bayesian Reasoning System has become trapped in a non-productive reasoning pattern, a crucial step toward building self-diagnosing and recoverable AI systems.

5 Controllability and Synthesis of Reasoning Strategies

5.1 Definition of Controllability

The Ramadge-Wonham supervisory control framework provides the formal foundation for our analysis of reasoning as a controllable process [7]. Within our Bayesian Reasoning System $\mathcal{D} = (\mathcal{B}, \Sigma, \delta, b_0, \mathcal{B}_m)$, we partition the event alphabet Σ into two disjoint sets: controllable events Σ_c and uncontrollable events Σ_u.

Controllable events Σ_c represent actions that can be enabled or disabled at will by an external agent or a supervisory controller. In our framework, these correspond to deliberate, agent-initiated operations. The canonical example is the **evidence introduction event** $E_{i,v}$, which models the agent's decision to observe or query the value of variable X_i. The agent chooses whether and when to perform such a query. The **intervention event** $\text{do}(X_i = v)$ is also controllable, representing a deliberate causal action taken by the agent [14].

Uncontrollable events Σ_u model occurrences that the agent or controller cannot prevent. These are integral to the environment's autonomy or to the fundamental nature of the reasoning process. The primary example is the **evidence retraction event** R_i. This event may be triggered by the environment (e.g., a sensor reading is deemed unreliable and revoked) or by a user correcting a previously supplied piece of information. The controller cannot forbid a user from retracting their own statement. Other potential uncontrollable events include spontaneous belief updates due to new streaming data or internal belief revision mechanisms that operate autonomously.

This partition is crucial. A reasoning strategy cannot assume it can prevent a user from changing their mind or prevent new, unsolicited information from arriving. Any viable control policy must function correctly in the presence of these uncontrollable events.

5.2 Maximal Controllable Sublanguage and Safe Reasoning

Let $K \subseteq L(\mathcal{D})$ be a desired *specification language*, representing the set of all admissible sequences of evidence and retraction events. This language encodes a *safety* or *liveness* property for the reasoning process. For instance, K could specify that "a low-confidence diagnosis is never reported as final" (safety) or that "the reasoning process eventually reaches a high-confidence state" (liveness). Typically, K is defined implicitly by a set of forbidden belief states or state sequences.

The central question of supervisory control is: Does there exist a supervisor (a feedback controller) that, by dynamically enabling or disabling only the controllable events in Σ_c, can guarantee that the closed-loop behavior of the system remains within the specification K, regardless of how the uncontrollable events in Σ_u occur? The existence of such a supervisor is determined by the property of *controllability*.

Definition 3 (Controllability of a Language)**.** *A language $K \subseteq L(\mathcal{D})$ is* controllable *with respect to $L(\mathcal{D})$ and Σ_u if:*

$$\mathit{Pref}(K)\Sigma_u \cap L(\mathcal{D}) \subseteq \mathit{Pref}(K),$$

where Pref(K) denotes the set of all prefixes of strings in K. Intuitively, this condition requires that no prefix of a legal string in K, when followed by an uncontrollable event that is physically possible in the plant $\mathcal{D}$, leads outside the set of prefixes of K. The supervisor must be able to "live with" uncontrollable events without violating the specification [7].

If the desired specification K is not controllable, we must seek the largest subset of it that is. This is the *supremal controllable sublanguage*, denoted $K^{\uparrow C}$. Algorithms exist to compute $K^{\uparrow C}$ for regular languages represented by finite automata [19]. In our context, this computation on the abstracted reachability graph yields the largest set of safe reasoning trajectories that can be enforced by a supervisor. The resulting sublanguage defines the boundary of *safe reasoning*: the region of the belief state space from which the agent can be guaranteed to avoid forbidden states (like belief deadlocks) no matter what retractions or other uncontrollable events happen.

5.3 Synthesis of Optimal Evidence Gathering Strategies

The supervisor realizing the supremal controllable sublanguage $K^{\uparrow C}$ can be viewed as a *reactive, safe reasoning policy*. However, safety alone is insufficient; we also desire *optimality*. We now frame the synthesis of an evidence gathering strategy as an optimal control problem within the controllable, safe region of the reasoning process.

Let us augment the BRS model with a cost function. Each controllable event (evidence query) $E_{i,v} \in \Sigma_c$ has an associated cost $c(E_{i,v}) \geq 0$, which could represent monetary cost, time, computational resource, or patient discomfort. Furthermore, remaining in a non-goal state may incur a continuing penalty. The objective is to synthesize a control policy $\pi : \mathcal{B} \to 2^{\Sigma_c}$ that maps the current belief state to the set of controllable events to enable. This policy must: 1. Ensure the system behavior remains within the safe, controllable region (i.e., within $K^{\uparrow C}$). 2. Minimize the expected cumulative cost until a marked state $b \in \mathcal{B}_m$ is reached.

This is a constrained optimal control or planning problem on a graph (the safe region of the reachability graph). It can be solved using dynamic programming or heuristic search algorithms like A^*, where the heuristic could be based on the expected information gain of a query or the estimated distance to a goal state [20].

The resulting optimal supervisor π^* is an **active inference strategy** with formal guarantees. It dynamically decides which question to ask next ($E_{i,v}$) or which intervention to perform ($do(X_i = v)$), balancing the cost of acquisition against the expected progress toward a conclusive, high-confidence belief state. Critically, it does so while respecting the intrinsic unpredictability of evidence retractions (R_i). It will avoid query sequences that, while seemingly informative, would leave the system vulnerable to being pushed into a deadlock by a subsequent uncontrollable retraction. This synthesis provides a principled, formal answer to the question of optimal experiment design or diagnostic strategy within an interactive, safety-critical reasoning environment.

6 Reversibility and Reasoning Traceability

6.1 Multi-Level Definitions of Reversibility

The concept of reversibility, central to many physical and computational processes, offers a novel lens through which to examine the dynamics of belief updating. We define reversibility for a Bayesian Reasoning System at three distinct levels of stringency, each corresponding to a different aspect of epistemic traceability and control.

The first and most basic level is *weak reversibility*, also called **state backtracking**. A BRS $\mathcal{D}$ is weakly reversible if from any reachable belief state, it is possible to return to the initial prior state b_0 through some finite sequence of admissible events. Formally,

$$\forall b \in \mathcal{R}(\mathcal{D}), \exists \sigma \in \Sigma^* \text{ such that } \delta(b, \sigma) = b_0.$$

This property ensures that no reasoning trajectory leads to a "point of no return"; the reasoner can always, in principle, reset its beliefs to the original prior. This is analogous to the reversibility property in Petri nets, where the initial marking is reachable from any reachable marking [9]. Weak reversibility is a global property of the reachability graph, requiring that b_0 be reachable from every node.

The second, more demanding level is *strong reversibility*, or **operation invertibility**. A BRS is strongly reversible if for every event $\sigma \in \Sigma$, there exists a corresponding inverse event $\sigma^{-1} \in \Sigma$ such that applying them in sequence leaves the belief state unchanged. Formally,

$$\forall b \in \mathcal{B}, \forall \sigma \in \Sigma, \delta(\delta(b, \sigma), \sigma^{-1}) = b \quad \text{whenever the transitions are defined.}$$

For evidence introduction $E_{i,v}$, a natural candidate for its inverse is the retraction event R_i. However, R_i alone is not a perfect inverse because it only removes the observation without restoring the exact prior marginal for X_i; the system state after R_i depends on other active evidence. Strong reversibility may therefore require more sophisticated inverse operators that restore the precise belief state preceding the forward operation, potentially involving a combination of events.

The third and most semantically rich level is *causal reversibility*. This concept pertains specifically to intervention events $\text{do}(X_i = v)$, which modify the causal structure of the model. A system is causally reversible if for any intervention $\text{do}(X_i = v)$, there exists a subsequent intervention (or sequence of interventions) $\text{do}(X_j = u), \ldots$ that can nullify its effect, restoring the original joint distribution over the system's variables. This does not necessarily mean returning to the exact same belief state b, as the history of interventions is part of the epistemic context, but it means restoring the original statistical and causal relationships. Causal reversibility connects our framework to the theory of identifiability and the search for compensating interventions in causal models [14].

6.2 Reversibility as a Metric for Explainability

We posit that the reversibility of a reasoning system is a fundamental, quantifiable component of its explainability. Explainability is often equated with the ability to answer "why?" questions. In a dynamic reasoning context, a crucial "why?" question is: *Why do you believe P?* A satisfactory answer requires tracing the belief in proposition P back to the specific pieces of evidence that most strongly support it.

The ease of performing this *belief provenance analysis* is directly related to the system's reversibility. In a highly reversible system, the path from a final conclusion back to its generating evidence is short, well-defined, and computationally cheap to traverse. The inverse operations provide a direct mechanism for "undoing" the reasoning steps to expose their antecedents. Conversely, a system with poor reversibility obfuscates this path. The mapping from evidence to conclusion may be a complex, non-injective function where many different evidence sets lead to the same final belief (a form of informational loss). Reconstructing the key evidence from the conclusion becomes an ill-posed inverse problem. Such a system is a "black box" not because its internal workings are hidden, but because its reasoning dynamics are lossy and irreversible. Thus, we propose that the degree of reversibility—measured, for example, by the average length or computational cost of a reversal sequence—serves as a formal metric for one important dimension of explainability: *traceability*.

6.3 Algorithmic Support for Achieving Reversibility

Achieving reversibility, especially strong reversibility, is not automatic in standard Bayesian inference systems. It requires deliberate algorithmic design. We outline several technical approaches.

The most straightforward method is *reasoning history preservation*. The system maintains a complete log of all applied events $\sigma_1, \sigma_2, \ldots, \sigma_k$. To reverse to a previous state, the system can clear its current beliefs and re-apply the event sequence from the start, stopping at the desired point. While simple, this method is computationally expensive for long histories and does not provide direct inverse events.

A more efficient approach is the design of *specialized belief rollback operators*. These are algorithms that compute $\delta(b, \sigma^{-1})$ directly, without recapitulating the entire history. For evidence retraction, this requires marginalizing over the retracted variable while conditioning on all other remaining evidence, which can be done efficiently if the joint distribution structure is exploited [2]. For more complex operations, one could maintain and update a *belief state difference* or a dual representation that makes inversion algebraic.

A profound connection can be made with **T-invariants** from Petri net theory [9]. A T-invariant is a vector of transition firings whose net effect on the marking is zero. In the Petri net model of a BRS, a T-invariant corresponds to a sequence of events that returns the belief state to its starting point. Computing the T-invariants of the system provides a catalogue of all minimal reversible cycles within the reasoning dynamics. Algorithmically, one can pre-compute these T-invariants. Then, to reverse a given event sequence, the system can search for a T-invariant that contains the sequence (or its complement) and execute the remaining transitions in the invariant to effect a reversal. This provides a principled, structural method for finding inverse operation sequences, grounding reversibility in the algebraic properties of the system's dynamic model.

7 Case Study: An Interactive Medical Diagnostic Assistant

7.1 System Modeling

We instantiate the Bayesian Reasoning System (BRS) framework with a simplified model for cardiovascular disease diagnosis. The model comprises five binary random variables:

- **Disease (D)**: Presence of coronary artery disease (True/False).
- **Symptom 1 (S1)**: Chest pain (Present/Absent).
- **Symptom 2 (S2)**: Shortness of breath (Present/Absent).
- **Test 1 (T1)**: Electrocardiogram (ECG) result (Abnormal/Normal).
- **Test 2 (T2)**: Stress test result (Positive/Negative).

The network structure is defined as: D is the parent of $S1$, $S2$, $T1$, and $T2$. The conditional probability tables are populated with medically plausible values, e.g., $P(S1 = \text{Present} \mid D = \text{True}) = 0.85$, $P(T1 = \text{Abnormal} \mid D = \text{True}) = 0.70$, with lower probabilities for the disease-free case.

The BRS $\mathcal{D}_{\text{cardio}}$ is defined as follows. The belief state b is represented by the marginal posterior distribution $P(D \mid \mathcal{E})$, the probability of disease given the current evidence set $\mathcal{E}$. The event set $\Sigma = \{E_{S1,p}, E_{S1,a}, E_{S2,p}, E_{S2,a}, E_{T1,ab}, E_{T1,n}, E_{T2,pos}, E_{T2,neg}, R_{S1}, R_{S2}, R_{T1}, R_{T2}\}$, where $p/a/ab/n/pos/neg$ denote present/absent/abnormal/normal/positive/negative. Evidence introduction events ($E_{...}$) are controllable. Evidence retraction events ($R_{...}$) are uncontrollable, modeling a doctor's reconsideration or a patient revising their symptom report. The initial state b_0 is the prior $P(D)$. The set of marked states $\mathcal{B}_m$ is defined as $\{b : |P(D = \text{True} \mid b) - 0.5| > 0.45\}$, representing a high-confidence diagnosis (probability < 0.05 or > 0.95). An exact inference algorithm (variable elimination) implements δ [21].

7.2 Analysis

First, we construct an abstracted reachability graph $G_{\mathcal{D}_{\text{cardio}}}$ using evidence equivalence abstraction. Each node is a belief state characterized by the set of variables for which evidence has been introduced (e.g., $\{S1, T1\}$), with the specific values implied. Analysis of this graph reveals several **belief deadlock states**. A prominent example is the state $b_{deadlock}$ reached by observing $S1 = \text{Absent}$ (no chest pain) followed by $T1 = \text{Abnormal}$ (abnormal ECG). Under our model parameters, these two pieces of evidence are strongly contradictory when the disease is considered the sole common cause. The resulting posterior $P(D \mid S1 = \text{Absent}, T1 = \text{Abnormal})$ is approximately 0.5, and no subsequent evidence introduction ($S2$ or $T2$) can push the probability beyond the 0.05-0.95 confidence threshold defined in $\mathcal{B}_m$. Any path from $b_{deadlock}$ remains outside $\mathcal{B}_m$, confirming its deadlock status. This state clinically corresponds to an unexplained abnormal test in an asymptomatic patient, a known diagnostic dilemma.

Second, we synthesize a **supervisory controller** to guide a physician's inquiry. The specification K mandates that the system must never enter a belief deadlock and must eventually reach a marked state if possible. Using the algorithm for computing the supremal controllable sublanguage $K^{\uparrow C}$ on $G_{\mathcal{D}_{\text{cardio}}}$ [19], we derive a safe region. The synthesized supervisor provides dynamic recommendations: for instance, it disables the $E_{T1,ab}$ event (ordering an ECG) if the system is in a state where $S1$ is observed as absent, because that sequence leads directly to the identified deadlock. Instead, it may suggest querying $S2$ first. The supervisor guarantees that by following its enabled actions, the physician will either achieve a high-confidence diagnosis or conclusively determine that one cannot be reached with the available tests, all while avoiding the contradictory, inconclusive deadlock.

Third, we analyze the **reversibility** of two common diagnostic protocols. Protocol A ("Symptom-first"): Inquire about $S1$ and $S2$ first, then possibly order $T1$ and $T2$. Protocol B ("Test-first"): Order $T1$ immediately, then inquire about symptoms. We measure reversibility as the average number of retraction events required to return from a high-confidence diagnostic state back to the prior b_0. Our analysis shows that for Protocol A, the average reversal length is 2.1. For Protocol B, it is 3.4. The difference arises because in the test-first protocol, a highly informative test result (e.g., $T1 = \text{Abnormal}$) strongly shapes the interpretation of subsequently reported symptoms, creating a more entangled belief state. Disentangling it (retracting the test result and then the symptoms influenced by it) requires more steps. This quantifies the intuition that "symptom-first" is more explainable: the contribution of each piece of evidence to the final diagnosis is more modular and easier to mentally retrace.

7.3 Simulation Results

We evaluate the impact of the formal BRS framework through a simulation involving 1000 synthetic diagnostic cases. A simulated physician interacts with the system, choosing questions and tests. We compare three conditions: (1) **Unassisted**: The physician acts without guidance. (2) **Rule-based**: The physician follows a simple static protocol (e.g., always ask symptoms first). (3) **BRS Supervised**: The physician receives dynamic recommendations from the synthesized supervisor.

The results are as follows. The **average number of decision steps** (queries + tests) to reach a termination condition (diagnosis or decision to stop) was 5.7 for Unassisted, 4.8 for Rule-based, and 3.9 for BRS Supervised. The supervisor's deadlock-avoidance and goal-directed guidance streamline the process. The **final diagnostic confidence**, measured as the proportion of cases terminating in a marked state $\mathcal{B}_m$, was 78% for Unassisted, 85% for Rule-based, and 96% for BRS Supervised. The supervisor actively steers the process away from low-information-gain trajectories that end in uncertainty. A simulated **user satisfaction score**, modeled as a function of diagnostic speed, clarity, and the system's ability to explain its reasoning when asked, showed a mean (on a 1-10 scale) of 6.2 for Unassisted, 7.1 for Rule-based, and 8.5 for BRS Supervised. The supervisor's ability to avoid contradictory states and provide clear justification via reversible reasoning paths directly enhanced perceived trust and usability.

This case study demonstrates that the formal analysis of a BRS–identifying pathologies, synthesizing controllers, and evaluating reversibility–translates into tangible improvements in the performance, reliability, and explainability of an interactive AI system.

8 Broader Implications and Future Directions

8.1 Implications for AI Trust and Safety

The dynamic systems perspective on Bayesian reasoning offers a paradigm shift in how we conceive of and certify trustworthy AI. Currently, trust is often grounded in statistical performance metrics: accuracy, precision, recall, or expected calibration error measured on held-out datasets. While valuable, these metrics are inherently probabilistic and do not provide guarantees for individual system executions or for behavior under novel, adversarial, or edge-case inputs. The properties we formalize—reachability, controllability, reversibility, and the absence of deadlocks—are *verifiable guarantees*. A statement such as "this diagnostic system is provably free of belief deadlocks" or "this reasoning assistant can always reverse its conclusions to justify them" is a deterministic claim about the system's dynamic behavior across all possible interaction sequences. Such claims can be verified using model checking, theorem proving, or algorithmic analysis on the (abstracted) reachability graph, providing a level of assurance that statistical metrics cannot [17]. This aligns with the growing demand for *formal verification* in safety-critical AI applications, moving trust from a statistical expectation to a mathematical certainty regarding key behavioral properties. The framework thus provides a pathway to build AI systems that are not merely empirically good, but verifiably reliable in their reasoning dynamics.

8.2 Redesigning Human-AI Collaboration

Our model naturally extends to scenarios where the human user is not merely a passive recipient of conclusions but an active participant in the reasoning loop. In this view, the human expert is integrated into the DEDS as an additional, highly capable but potentially unpredictable "event generator." Human actions—posing a novel query, overruling a system suggestion, providing complex contextual information—become events in an expanded alphabet Σ_H. This creates a *human-AI hybrid dynamic system*. The core challenge shifts from controlling a purely algorithmic process to managing a mixed-initiative interaction. The controllability condition must now account for human-generated events, which may be partially controllable (e.g., the interface can guide or constrain user choices) but largely uncontrollable from the system's perspective. The synthesis problem becomes one of designing a shared controller or *interaction protocol* that ensures the combined system, despite the human's autonomy, remains within a safe and productive region of the joint belief state space. This involves new questions: How to model bounded rational human event generation? How to synthesize protocols that are robust to human error or contradiction? This direction reframes human-AI teaming as a cooperative control problem, with formal guarantees on collaborative outcomes.

8.3 Connections to Other Areas of Machine Learning

The principles developed here intersect profoundly with other subfields of machine learning, suggesting unified approaches to reasoning and learning.

First, consider *continual or online learning*. In a standard BRS, the graphical model $\mathcal{G}$ and its parameters θ are fixed. Continual learning relaxes this; the model itself evolves over time as new data arrives, through parameter adaptation or even structural changes. This can be modeled as a higher-order dynamic system where the base object—the BRS $\mathcal{D}$—is now a state variable itself. Changes to $\mathcal{G}$ or θ become meta-events that alter the very transition function δ of the reasoning process. This introduces the problem of *meta-controllability*: how to control the learning process (the meta-events) to ensure that the evolving reasoning system maintains desirable dynamic properties (e.g., remains deadlock-free, retains reversibility) even as it learns. It creates a formal link between the stability of learning algorithms and the reliability of the inference systems they produce.

Second, there is a deep connection with *Reinforcement Learning (RL)*. An RL policy that selects actions in an environment can be seen as a supervisor for a dynamic system whose state is the environment state. In our framework, the policy is a supervisor for the BRS, selecting which evidence-gathering actions (controllable events) to enable. The crucial difference is in the objective. Standard RL maximizes a cumulative reward, often tied to external success [22]. Our framework suggests an alternative, intrinsic objective: maximize or constrain dynamic properties of the reasoning process itself, such as ensuring controllability, minimizing the risk of deadlocks, or maximizing reversibility. This points toward a novel class of *safety-oriented or explanation-oriented RL* where the reward function is derived from the formal properties of the agent's own epistemic state dynamics, promoting inherently reliable and transparent reasoning strategies.

8.4 Open Challenges

While the proposed framework opens promising avenues, significant challenges remain to realize its full potential.

The most fundamental is the *state space explosion* inherent in analyzing dynamic systems. The belief state space $\mathcal{B}$ grows exponentially with the number of variables and the precision of discretization. While abstraction techniques (evidence equivalence, information-geometric clustering) are essential, they trade precision for tractability. Future work needs more sophisticated abstraction-refinement methods, symbolic representations, and compositional reasoning techniques that allow the analysis of large networks by decomposing them into interacting subsystems.

Relatedly, our current formalization primarily addresses discrete variables. Extending the framework to handle *continuous and hybrid networks* (mixing discrete and continuous variables) is non-trivial. This requires moving from finite-state automata to hybrid automata or other continuous dynamical systems models for the belief state evolution. Defining events like evidence introduction on continuous variables (e.g., "blood pressure is 142 mmHg") and analyzing the resulting infinite reachability graphs pose major theoretical and computational hurdles.

Finally, there is a need for a *rigorous axiomatic framework for quantifying* properties like controllability and reversibility. While we have provided definitions, developing robust scalar metrics—for example, a degree of controllability measuring how close a specification is to being controllable, or a reversibility index measuring the average computational cost of inversion—would allow for comparative analysis and optimization. Establishing the mathematical relationships between these metrics and more traditional measures like model accuracy or information gain would solidify the foundations of this research direction.

9 Conclusion

We have demonstrated that reconceptualizing Bayesian reasoning as a dynamic system is not only possible but profoundly fruitful. This paradigm shift imports a mature and powerful suite of formal tools—originally developed for the design and analysis of complex engineered systems—into the domain of probabilistic machine learning. By focusing on the dynamic properties of **reachability, controllability, and reversibility**, we move beyond viewing inference as a static function computation. Instead, we model it as an interactive process whose trajectories through belief space can be formally analyzed, controlled, and verified.

This work establishes the foundation for a new generation of rational agents. These agents will be distinguished not merely by their ability to compute correct answers from given data, but by the **reliable, transparent, and collaboratively effective manner** in which they reason. Reliability is ensured through the formal detection and avoidance of pathological dynamics like belief deadlocks. Transparency is enhanced by quantifying and designing for reversibility, making the provenance of conclusions traceable. Effective human-AI collaboration is facilitated by framing the interaction as a controllable hybrid dynamic system, where protocols can be synthesized to guarantee productive joint outcomes.

The framework presented here is necessarily foundational. It opens more questions than it closes, pointing toward an expansive new research agenda. We have shown how to model a Bayesian Reasoning System as a Discrete Event

Dynamic System, defined its key properties, and provided initial algorithms for analysis and control. The case study demonstrates its practical viability and benefits. The connections drawn to continual learning and reinforcement learning suggest a path toward a unified theory of learning and reasoning dynamics.

Ultimately, this work inaugurates a research program we term **Formally Trustworthy Reasoning**. The ambition is to endow AI systems with verifiable, certificate-like guarantees about their epistemic behavior—guarantees that are complementary to, and potentially more robust than, statistical performance assurances. By applying the rigor of formal methods to the dynamics of belief, we take a significant step toward intelligent systems that are not only powerful but also predictable, accountable, and worthy of our trust in safety-critical and ethically sensitive domains. The journey from static computation to dynamic, verifiable reasoning has begun.

References

[1] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[2] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.

[3] Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988.

[4] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475, 1999.

[5] Thomas Dean and Keiji Kanazawa. Planning and control in stochastic domains with imperfect information. *Technical report, Brown University, Department of Computer Science*, 1989.

[6] Kevin P. Murphy. Dynamic bayesian networks: representation, inference and learning. 2002.

[7] Peter J. Ramadge and W. Murray Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.

[8] Christos G. Cassandras and Stéphane Lafortune. *Introduction to discrete event systems*. Springer Science & Business Media, New York, NY, 2008.

[9] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[10] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. *International Conference on Computer Aided Verification*, pages 97–117, 2017.

[11] Dorsa Sadigh, S. Shankar Sastry, Sanjit A. Seshia, and Anca D. Dragan. Planning for autonomous cars that leverage effects on human actions. *Robotics: Science and Systems*, 2:1–9, 2016.

[12] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM journal of research and development*, 5(3):183–191, 1961.

[13] Charles H. Bennett. Logical reversibility of computation. *IBM journal of Research and Development*, 17(6):525–532, 1973.

[14] Judea Pearl. *Causality: Models, reasoning, and inference*. Cambridge University Press, Cambridge, UK, 2009.

[15] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, Cambridge, MA, 2008.

[16] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[17] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem. *Model checking*. MIT Press, Cambridge, MA, 2018.

[18] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[19] W. Murray Wonham and Kai Cai. *Supervisory control of discrete-event systems*. Springer, New York, NY, 2013.

[20] Stuart J. Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, Upper Saddle River, NJ, 3rd edition, 2010.

[21] Nevin L. Zhang and David Poole. Exploiting causal independence in bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.

[22] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018.