

Article

Integrating Boolean Satisfiability Algorithms into Bayesian Networks for Accelerated Deterministic Inference

Efraín Evaristo Díaz Macías * , José Steven Cordero Bazurto  and Byron Wladimir Oviedo Bayas 

Facultad de Ciencias de la Computación, Universidad Técnica Estatal de Quevedo, Quevedo 120301, Ecuador; jcorderob@uteq.edu.ec (J.S.C.B.); boviedo@uteq.edu.ec (B.W.O.B.)

* Correspondence: efraindiaz@uteq.edu.ec

Abstract

Exact probabilistic inference in Bayesian Networks (BNs) becomes increasingly expensive as network size and structural complexity grow, limiting its applicability in time-sensitive decision-support systems. This study presents a hybrid inference framework that accelerates the deterministic component of Bayesian reasoning by integrating Boolean Satisfiability (SAT) techniques with Bayesian Networks. The proposed approach transforms deterministic conditional probability table (CPT) entries into conjunctive normal form (CNF), enabling SAT-based logical inference over deterministic constraints while preserving the original Bayesian model for probabilistic reasoning. The framework was evaluated on 25 benchmark Bayesian networks using five independent executions per dataset under identical experimental conditions. Performance was assessed through execution time, instrumented operation counts, and inference coverage, with results reported as mean values, standard deviations, and 95% confidence intervals. Experimental results demonstrate substantial reductions in deterministic inference time while maintaining high coverage of deterministic variable assignments across the evaluated benchmarks. Throughout this paper, the reported performance gains refer exclusively to empirical reductions in execution time and instrumented operation counts. They should not be interpreted as evidence of a reduction in the asymptotic computational complexity of exact Bayesian inference, which remains #P-complete in the general case. Rather, the proposed framework provides an efficient mechanism for accelerating deterministic logical inference within Bayesian Networks under the evaluated benchmark conditions.

Keywords: Boolean Satisfiability (SAT) Algorithms; Bayesian Networks; probabilistic inference; computational complexity reduction; artificial intelligence; machine learning

MSC: 03-08, 68T30



Academic Editor: Roberto Montemanni

Received: 21 May 2026

Revised: 25 June 2026

Accepted: 1 July 2026

Published: 8 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Bayesian Networks (BNs) constitute one of the most widely used probabilistic graphical models for reasoning under uncertainty. Their ability to represent complex probabilistic dependencies has led to successful applications in medical diagnosis, intelligent transportation, fault detection, recommendation systems, financial risk analysis, uncertainty estimation, applied machine-learning systems, agricultural and environmental decision-support systems, and decision-support systems [1–3]. By explicitly modeling conditional dependencies among variables, BNs provide an interpretable and mathematically rigorous framework for probabilistic reasoning in domains characterized by incomplete or uncertain information.

Despite these advantages, the practical deployment of Bayesian Networks remains constrained by the computational cost of exact probabilistic inference. As network size, connectivity, and treewidth increase, inference rapidly becomes computationally demanding. In the general case, exact inference in Bayesian Networks is known to be #P-complete, limiting its applicability in large-scale or time-sensitive scenarios where low-latency responses are required. Recent studies have also emphasized that execution schedules can influence the practical performance of Bayesian inference algorithms, while computational complexity remains a central criterion for evaluating inference methods [4,5]. Consequently, improving the efficiency of inference has become an active research topic in both academia and industry.

Many real-world applications illustrate this challenge. Clinical decision-support systems must analyze numerous interacting variables under strict time constraints, intelligent transportation platforms continuously update probabilistic models using large streams of traffic information, and recommendation systems employed by digital services process millions of user interactions to generate personalized predictions. In these environments, computational efficiency directly affects system responsiveness, scalability, and user experience.

One promising strategy consists of exploiting the deterministic information already embedded in Bayesian Networks. Conditional Probability Tables (CPTs) frequently contain deterministic entries with probabilities equal to 0 or 1, representing logical constraints rather than uncertain relationships. Such deterministic knowledge can be represented as Boolean formulas and solved efficiently using Boolean Satisfiability (SAT) techniques. Modern SAT solvers have demonstrated remarkable performance when solving large logical problems containing thousands or even millions of variables and constraints [6,7].

Motivated by this observation, this work proposes a hybrid inference framework that integrates Boolean Satisfiability algorithms into Bayesian Networks to accelerate deterministic inference. The proposed approach transforms deterministic CPT entries into Conjunctive Normal Form (CNF), allowing SAT-based logical inference to operate over deterministic constraints while preserving the original Bayesian Network for probabilistic reasoning. Therefore, the proposed framework is not intended to replace exact probabilistic inference, but rather to accelerate the deterministic component of the inference process.

To validate the proposed framework, a complete experimental platform was developed using the Django framework and evaluated on 25 benchmark Bayesian Networks obtained from the UCI Probabilistic Inference Benchmark repository. Each benchmark was executed five independent times under identical experimental conditions. Performance was assessed using execution time, instrumented operation counts, inference coverage, statistical confidence intervals, effect size, and sensitivity analyses of the $\text{mix}(Q)$ parameter. The complete implementation and experimental workflow are publicly available at <https://github.com/imjosecorderob/BaySATEnhanced> (accessed on 25 June 2026).

1.1. Main Contributions

The principal scientific contributions of this work are summarized as follows:

- We propose a hybrid inference framework that integrates Boolean Satisfiability (SAT) techniques with Bayesian Networks to accelerate deterministic inference while preserving probabilistic reasoning for non-deterministic variables.
- We introduce a deterministic CPT-to-CNF transformation procedure that converts probability-0 and probability-1 entries into logically equivalent CNF clauses, enabling efficient SAT-based reasoning over deterministic dependencies.
- We perform a comprehensive experimental evaluation on 25 benchmark Bayesian Networks comprising between 440 and 1444 variables. Each benchmark was executed

five independent times, and the analysis includes execution time, instrumented operation counts, inference coverage, confidence intervals, effect size, parameter sensitivity, and multiple-comparison statistical tests.

- We provide a statistical characterization of the benchmark collection, including graph density, evidence variables, deterministic CPT ratios, confidence intervals, Kruskal–Wallis tests, Spearman correlation, and Bonferroni-corrected pairwise comparisons, thereby strengthening the reproducibility and statistical validity of the experimental evaluation.
- We release the complete implementation, datasets, and experimental workflow as an open-source framework to facilitate independent verification and future research on hybrid deterministic inference in Bayesian Networks.

1.2. Theoretical Background

1.2.1. Fundamentals of Bayesian Networks

Formally, a Bayesian Network (BN) is defined as a pair (G, P) , where $G = (V, E)$ is a directed acyclic graph (DAG) with $V = \{X_1, \dots, X_n\}$ representing the random variables and E representing their conditional dependency structure. The probabilistic component is given by the family of conditional probability distributions

$$P = \{P(X_i \mid \text{Pa}_G(X_i)) : i = 1, \dots, n\},$$

which factorizes the joint probability distribution as

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i \mid \text{Pa}_G(X_i)), \quad (1)$$

where $\text{Pa}_G(X_i)$ denotes the parent nodes of X_i in the graph. Nodes without parents are characterized by their marginal probability distributions [8].

Bayesian Networks combine probability theory and graph theory to model uncertain systems through an intuitive graphical representation while supporting mathematically rigorous probabilistic reasoning. Their principal components are summarized below.

- **Nodes (Random Variables):** Each node represents a discrete or continuous random variable corresponding to an observable event, latent state, or hypothesis. Conditional dependencies between variables are defined through the directed parent–child relationships of the graph.
- **Directed Arcs:** Directed edges encode conditional probabilistic dependencies between variables. The absence of an arc indicates conditional independence under the assumptions represented by the network structure.
- **Conditional Probability Tables (CPTs):** Each node is associated with a Conditional Probability Table specifying the probability distribution of the variable conditioned on every possible configuration of its parent variables. Root nodes are defined by marginal probability distributions.

The primary operation performed on a Bayesian Network is probabilistic inference, which computes posterior beliefs for unobserved variables from available evidence. As described by Murphy [9], Bayesian inference supports three complementary reasoning paradigms: predictive inference, which estimates future outcomes from known causes; diagnostic inference, which identifies the most plausible causes explaining observed evidence; and probabilistic updating, which continuously revises posterior beliefs as new information becomes available. These capabilities make Bayesian Networks particularly suitable for decision-support systems operating under uncertainty.

Although Bayesian Networks provide a rigorous framework for probabilistic reasoning, the computational cost of exact inference increases rapidly with network size and structural complexity. This computational bottleneck has motivated the development of alternative techniques capable of accelerating specific stages of the inference process, particularly those involving deterministic relationships encoded within Conditional Probability Tables.

1.2.2. Boolean Satisfiability (SAT) and Conjunctive Normal Form

The Boolean Satisfiability (SAT) problem consists of determining whether there exists an assignment of truth values to a set of Boolean variables such that a propositional logic formula evaluates to true. SAT is commonly expressed in **Conjunctive Normal Form (CNF)**, where a formula is represented as the conjunction (logical AND) of one or more clauses, each clause being a disjunction (logical OR) of literals (variables or their negations).

For example, the formula $(A \vee \neg B) \wedge (B \vee C)$ is satisfiable because the assignment $A = \text{true}$ and $C = \text{true}$ satisfies both clauses. This representation has become the standard input format for modern SAT solvers due to its compact structure and computational efficiency.

The SAT problem occupies a central position in computational complexity theory, having been the first problem proven to be NP-complete by Cook [10]. Consequently, numerous combinatorial decision problems can be reduced to SAT through polynomial-time transformations. Beyond its theoretical importance, SAT has found widespread applications in artificial intelligence, hardware and software verification, automated planning, scheduling, formal reasoning, and combinatorial optimization.

State-of-the-art SAT solvers, including Glucose3 [11], employ Conflict-Driven Clause Learning (CDCL) [12] together with branching heuristics, unit propagation, clause learning, and non-chronological backtracking to efficiently explore extremely large search spaces. These techniques enable SAT solvers to solve logical instances containing millions of variables and clauses, making them an attractive alternative for exploiting deterministic structures embedded within probabilistic graphical models.

In the proposed framework, SAT is not used to replace probabilistic inference. Instead, it operates exclusively on deterministic constraints extracted from Bayesian Network Conditional Probability Tables (CPTs), allowing efficient logical inference over probability-0 and probability-1 relationships while preserving the original probabilistic model for uncertain variables.

1.2.3. Benchmark Data Formats: UAI and EVID

To ensure reproducibility and compatibility with established probabilistic inference benchmarks, this study employs the standardized `.uai` and `.evid` formats defined by the Association for Uncertainty in Artificial Intelligence (UAI). The `.uai` files describe the complete probabilistic graphical model, including variables, graph structure, and Conditional Probability Tables (CPTs), whereas the corresponding `.evid` files specify the observed evidence variables used during inference.

These formats are widely adopted by the UAI Probabilistic Inference Evaluation repository and facilitate standardized experimental comparisons across different inference algorithms. Their use also enables direct reproducibility of the proposed framework and simplifies the evaluation of hybrid deterministic inference techniques on publicly available Bayesian Network benchmarks.

- `.uai` files contain the complete specification of a probabilistic graphical model, including the network type (e.g., Bayesian or Markov), the variables and their domains, the graphical structure represented by factors, and the corresponding Conditional

Probability Tables (CPTs). Consequently, a .uai file fully defines the probabilistic model used during inference.

- .evid files contain the observed evidence associated with a previously defined .uai model. They specify the variables whose values are known a priori and provide the evidence required to perform inference under different experimental scenarios without modifying the underlying Bayesian Network.

The combined use of the .uai and .evid formats enables standardized and fully reproducible experiments, allowing different inference algorithms to be evaluated under identical probabilistic models and evidence configurations.

1.2.4. Theoretical Framework: Logical Marginalization and the mix Algorithm

The deterministic inference component proposed in this work is based on the Boolean-array marginalization framework introduced by Díaz and Moral [13]. Their approach extends the classical Davis–Putnam variable elimination procedure by representing propositional formulas as explicit Boolean arrays, enabling logical marginalization through efficient algebraic operations over truth tables.

This methodology is particularly relevant because it provides an efficient mechanism for eliminating logical variables while preserving the semantic consistency of the original propositional representation. Furthermore, the framework is grounded in the Shenoy–Shafer valuation algebra, allowing local computations to be organized through modular operations analogous to those employed during probabilistic inference in Bayesian Networks.

The relationship between Bayesian Networks and propositional reasoning has been extensively investigated in previous research, particularly through the transformation of probabilistic graphical models into logical representations suitable for satisfiability-based inference [14,15]. Building upon this line of research, the present work adopts the Boolean-array framework proposed by Díaz and Moral [13] to accelerate deterministic inference over benchmark Bayesian Networks obtained from the UAI repository.

The proposed framework exploits two key properties of Boolean-array marginalization:

- **Efficient logical marginalization.** Variable elimination is performed directly on Boolean-array representations, avoiding redundant logical evaluations and reducing the computational effort associated with deterministic reasoning.
- **Detection of deterministic dependencies.** Functional dependencies encoded within Conditional Probability Tables are transformed into logical constraints, allowing deterministic relationships to be propagated efficiently through SAT-based reasoning while preserving the original probabilistic model for uncertain variables.

Díaz and Moral [13] demonstrated that this framework scales to propositional instances containing thousands of variables and highly connected logical structures. These characteristics motivated its adoption as the deterministic inference engine integrated into the hybrid framework proposed in this study.

1.3. Related Work and State of the Art

The integration of Boolean Satisfiability (SAT) techniques with Bayesian Networks has attracted growing interest as a strategy for improving the efficiency and scalability of probabilistic inference. Modern SAT solvers have demonstrated remarkable performance when solving large combinatorial optimization problems, motivating their application to probabilistic graphical models where exact inference becomes computationally expensive. Recent research has explored this integration from several complementary perspectives, including probabilistic model counting, Bayesian Network structure learning, hybrid logical–probabilistic reasoning, and scalable inference under uncertainty.

One of the most significant advances in SAT-based probabilistic inference is KOCO-SMC, proposed by Li et al. [16]. This solver extends the classical SAT framework through Satisfiability Modulo Counting (SMC), enabling both logical satisfiability and probabilistic model counting. Instead of merely determining whether a logical assignment exists, KOCO-SMC estimates the probability associated with valid assignments by combining SAT solving with probabilistic circuits. This approach considerably improves the efficiency of exact and approximate probabilistic inference while preserving the expressive capabilities of probabilistic graphical models.

Similarly, Audemard et al. [17] introduced d4Max, a solver extending the Weighted Max#SAT problem through the incorporation of forgotten variables and weight functions. Their objective was to efficiently solve Maximum A Posteriori (MAP) inference problems in Bayesian Networks. Experimental results demonstrated that d4Max achieves competitive performance compared with state-of-the-art Bayesian inference tools such as Merlin, while in several benchmark instances providing solutions closer to the optimum within shorter execution times. These results reinforce the potential of SAT-based optimization for probabilistic reasoning.

SAT technology has also been successfully applied to Bayesian Network structure learning. Peruvemba and Parviainen [18] proposed the SAT-based Local Improvement Method (SLIM), a metaheuristic that begins with an initial heuristic Bayesian Network and iteratively improves local regions of the graph through SAT optimization while preserving bounded treewidth. Their experimental evaluation showed consistent improvements over well-established algorithms such as Greedy Equivalence Search (GES) and Max-Min Hill Climbing (MMHC), particularly for networks containing up to one thousand variables. Likewise, Lu et al. [19] introduced scalable score-based algorithms for optimal Bayesian Network learning capable of maintaining high structural accuracy even when the faithfulness assumption does not hold. Their approach demonstrated greater robustness than traditional algorithms such as PC and GES under noisy and structurally complex scenarios.

Another important research direction combines logical reasoning with probabilistic inference. Miao et al. [20] proposed BayesCrowd, a hybrid framework integrating Bayesian reasoning with SAT-based techniques to process skyline queries over incomplete datasets. Their system employs adaptive DPLL search together with conditional tables to efficiently perform inference under dominance constraints, illustrating how propositional reasoning can be effectively combined with probabilistic models to solve complex decision-support problems. Similarly, Lee [21] investigated probabilistic reasoning through Stochastic Boolean Satisfiability (SSAT), extending classical SAT by introducing random quantifiers. The resulting solvers, including reSSAT and erSSAT, enable efficient approximation of satisfaction probabilities in logical systems affected by uncertainty and incomplete information, thereby expanding the applicability of SAT techniques to probabilistic domains.

Beyond direct SAT applications, several studies have explored scalable inference in uncertain environments through complementary methodologies. Baek and Park [22] proposed an adaptive Bayesian framework known as SATF for psychophysical experiments, employing Bayesian inference to dynamically balance estimation accuracy and computational cost while reducing the number of required experimental trials. Although not specifically designed for Bayesian Networks, the underlying principle of adaptive inference provides valuable insights for computational optimization in probabilistic reasoning. Likewise, Alali and Imani [23] developed scalable inference techniques for Partially Observable Dynamic Boolean Systems (POBDS), combining particle filters with Gaussian processes to perform accurate inference in highly structured systems characterized by uncertainty and partial observability.

Practical applications further demonstrate the versatility of Bayesian Networks in complex decision-making environments. Kiwanuka et al. [24] applied Bayesian Networks to workforce scheduling by integrating probabilistic reasoning with data mining and machine learning techniques to model labor constraints and personnel availability. Their work illustrates how probabilistic graphical models can be employed within highly constrained optimization problems where logical consistency and efficient search are equally important. In a complementary contribution, Matricon et al. [25] proposed statistical benchmark-selection methods for evaluating SAT and Bayesian algorithms, reducing experimental costs while preserving representative benchmark diversity. Such methodologies are particularly valuable when validating new hybrid inference frameworks across large collections of benchmark instances.

Although these studies demonstrate the effectiveness of SAT techniques in probabilistic inference, Bayesian Network learning, optimization, and hybrid reasoning, relatively little attention has been devoted to exploiting the deterministic information already contained within Conditional Probability Tables (CPTs). Most existing approaches either perform complete probabilistic inference, optimize network structure, or estimate probabilistic quantities through model counting. In contrast, the present work focuses specifically on accelerating deterministic inference by extracting probability-0 and probability-1 CPT entries, transforming them into Conjunctive Normal Form (CNF), and applying SAT-based logical reasoning exclusively to deterministic constraints while preserving the original Bayesian Network for probabilistic inference. This distinction represents the principal research gap addressed by the proposed framework and differentiates it from previous SAT-based approaches reported.

Table 1 summarizes the principal characteristics of the most representative SAT-based approaches discussed in this section and highlights the distinctive features of the proposed framework with respect to the current state of the art.

Table 1. Comparison of representative SAT-based approaches for Bayesian Network inference and learning.

Approach	Primary Objective	Large-Scale Networks	Open Source	Main Contribution
KOCO-SMC [16]	Probabilistic model counting	Yes	Partial	Exact/approximate SMC inference
SLIM [18]	BN structure learning	Yes	Yes	SAT-guided graph optimization
d4Max [17]	MAP inference	Yes	No	Weighted Max#SAT optimization
BayesCrowd [20]	Hybrid probabilistic reasoning	Partial	No	Skyline queries with incomplete data
reSSAT/erSSAT [21]	Stochastic satisfiability	Partial	Yes	Probabilistic reasoning under uncertainty
BaySAT (this work)	Accelerated deterministic inference	Yes (440–1444 nodes)	Yes	Deterministic CPT-to-CNF transformation with Boolean-array reasoning

Unlike previous SAT-based approaches, the proposed framework focuses specifically on accelerating the deterministic component of Bayesian inference rather than replacing probabilistic reasoning. Deterministic Conditional Probability Table (CPT) entries are transformed into Conjunctive Normal Form (CNF) and processed through the Boolean-array elimination algorithm proposed by Díaz and Moral [13], while the original Bayesian Network is preserved for probabilistic inference over non-deterministic variables. The framework was experimentally validated on 25 benchmark Bayesian Networks through five independent executions per dataset, providing a fully reproducible open-source implementation together with a comprehensive statistical evaluation of the $\text{mix}(Q)$ parameter across 10 configurations per network (250 experimental configurations in total).

2. Materials and Methods

2.1. Methodological Approach: CRISP-DM Adaptation

This research followed a **quantitative, applied, and quasi-experimental** design. It is classified as applied because it proposes and validates a practical framework that integrates Boolean Satisfiability (SAT) techniques with Bayesian Networks to accelerate

deterministic inference. The quantitative approach is reflected in the systematic measurement and statistical analysis of execution time, instrumented operation counts, inference coverage, and parameter sensitivity across multiple benchmark datasets. The study is quasi-experimental because both the baseline Bayesian inference method and the proposed SAT-enhanced framework were evaluated under controlled experimental conditions using identical datasets and evidence configurations.

The overall methodological workflow was structured according to the **CRISP-DM (Cross-Industry Standard Process for Data Mining)** methodology, shown in Figure 1. CRISP-DM was originally developed by a consortium including SPSS Inc., Daimler-Benz, NCR, and OHRA as a standardized framework for organizing data mining projects [26]. Its six sequential phases—Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment—provided a systematic structure for the development, implementation, and validation of the proposed framework.

The adoption of CRISP-DM was particularly appropriate because the present research combines data processing, algorithm development, experimental evaluation, and software implementation. Each phase of the methodology was directly associated with a stage of the proposed framework, from defining the research objectives and selecting benchmark Bayesian Networks to transforming Conditional Probability Tables (CPTs) into Conjunctive Normal Form (CNF), executing SAT-based deterministic inference, collecting performance metrics, and validating the experimental results. This structured workflow ensured the reproducibility of the experiments while facilitating iterative refinement of the proposed inference framework before the final evaluation stage.

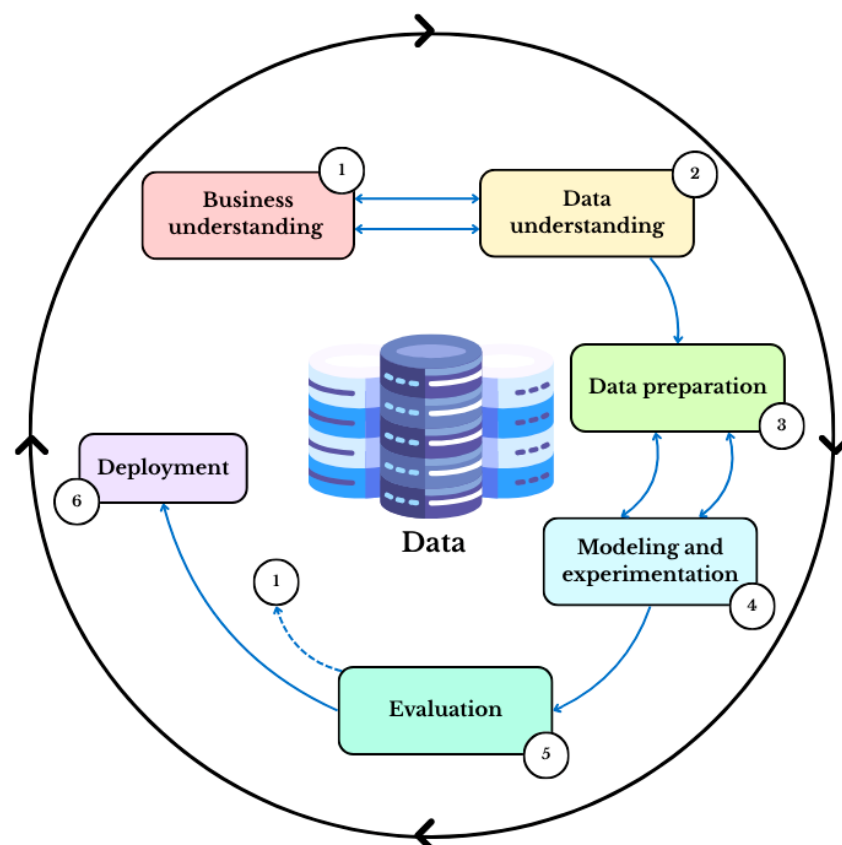


Figure 1. Adapted CRISP-DM workflow adopted in this research, illustrating the sequential stages from Business Understanding to Deployment. Numbers indicate the six phases of the CRISP-DM methodology, while arrows represent the workflow sequence and iterative feedback between phases.

Execution of the CRISP-DM Phases

The proposed framework was developed following the six phases of the adapted CRISP-DM methodology, ensuring a structured, reproducible, and iterative research process.

1. **Business Understanding.** The first phase defined the research objective of accelerating deterministic inference in Bayesian Networks through the integration of Boolean Satisfiability techniques. The motivation was driven by the increasing computational cost of exact Bayesian inference as network size grows, particularly in applications requiring low latency and efficient resource utilization, such as medical decision support, intelligent transportation, monitoring systems, and planning.
2. **Data Understanding.** Twenty-five benchmark Bayesian Networks were selected from the UCI Probabilistic Inference Benchmarks repository. The selected datasets range from 440 to 1444 variables and are distributed as paired `.uai` and `.evid` files describing the network structure and observed evidence, respectively. During this phase, the structural characteristics of each network were analyzed, including the number of variables, graph connectivity, conditional probability tables (CPTs), deterministic CPT entries, and evidence variables. Only datasets compatible with the proposed deterministic inference framework were included in the experimental evaluation.
3. **Data Preparation.** Automated preprocessing routines were developed to parse UAI benchmark files, validate the Bayesian Network structure, extract CPT information, and generate the corresponding logical representation. Deterministic CPT entries (probability 0 or probability 1) were transformed into Conjunctive Normal Form (CNF), while the original Bayesian Network structure was preserved for probabilistic inference. This preprocessing stage also standardized the experimental inputs, ensuring that all benchmark networks could be processed under identical conditions.
4. **Modeling and Experimentation.** A dual-inference architecture was implemented consisting of two independent processing pipelines. The first performs conventional Bayesian inference using Variable Elimination, whereas the second applies SAT-based deterministic inference through the Boolean-array elimination algorithm proposed by Díaz and Moral [13]. For each benchmark network, ten different values of the $\text{mix}(Q)$ parameter were evaluated to identify stable operating regions and analyze the influence of logical simplification on computational performance.
5. **Evaluation.** The experimental evaluation compared both inference strategies using identical datasets and evidence configurations. Performance was assessed through execution time, instrumented operation counts, inference coverage, parameter sensitivity, and descriptive statistical analysis. Each benchmark network was executed five independent times under identical hardware conditions, allowing the computation of mean values, standard deviations, confidence intervals, effect sizes, and parameter sensitivity analyses for the $\text{mix}(Q)$ configurations.
6. **Deployment.** The final implementation was integrated into a web-based experimental platform developed using the Django framework. The dashboard automates dataset loading, Bayesian Network visualization, deterministic and probabilistic inference execution, comparative performance analysis, and graphical reporting. This environment enables full reproducibility of the experimental workflow and provides a practical platform for evaluating additional benchmark networks and future algorithmic extensions.

2.2. Research Methods and Techniques

The experimental evaluation combined complementary research methods to ensure a rigorous and reproducible validation of the proposed framework. A quasi-experimental methodology was adopted because both the baseline Bayesian inference method and the SAT-enhanced deterministic inference framework were evaluated under controlled conditions using identical datasets, evidence configurations, and computational resources. This design enabled direct performance comparisons without manipulating real-world environments.

Analytical methods were employed to examine the behavior of both inference paradigms in terms of execution time, instrumented operation counts, inference coverage, and parameter sensitivity. Comparative analysis was subsequently performed to quantify the performance differences between conventional Bayesian inference and the proposed SAT-enhanced framework. Finally, an inductive approach was used to identify general performance trends across benchmark networks of varying sizes and structural characteristics.

Experimental data were collected automatically through the developed software platform. Instead of manual measurements, all performance indicators were generated directly by the execution engine and exported into structured spreadsheets for statistical analysis. Each experimental record included the benchmark identifier, network size, execution times, operation counts, inference coverage, optimal mix(Q) configuration, descriptive statistics, and comparative performance metrics. Automated data collection minimized human intervention and ensured complete reproducibility of the experimental results.

2.3. Data Acquisition, Conversion, and Experimental Logic

2.3.1. Data Acquisition and Benchmark Selection

The experimental evaluation was conducted using benchmark Bayesian Networks originally obtained from the **UCI Probabilistic Inference Benchmarks repository**. To guarantee long-term accessibility and facilitate the reproducibility of the experiments, the benchmark datasets used in this study have been archived in the datasets directory of the project's public GitHub repository (<https://github.com/imjosecorderob/BaySATENhanced/tree/master/datasets>, accessed on 25 June 2026). The archived datasets are represented in the standardized .uai format together with the corresponding evidence files (.evid) used in probabilistic inference competitions.

Because the repository contains both Bayesian Networks and Markov Networks, only Bayesian Network instances satisfying the experimental requirements were considered. The selection process included four criteria: (i) directed acyclic graph (DAG) structure, (ii) availability of associated evidence files, (iii) compatibility with the proposed deterministic inference framework, and (iv) representative structural complexity. Following this procedure, twenty-five benchmark networks containing between 440 and 1444 variables were selected, representing the complete set of compatible Bayesian Networks available under the established selection criteria. This strategy minimizes selection bias while ensuring a diverse collection of benchmark instances for experimental validation.

For each benchmark, structural descriptors including graph density, number of evidence variables, deterministic CPT ratio, number of factors, and network size were computed prior to experimentation. These descriptors are summarized in Table 2 and provide a quantitative characterization of the experimental benchmark collection.

Table 2. Structural characterization of the 25 benchmark Bayesian Networks. Graph density is computed as $E/[N(N - 1)]$. The deterministic CPT ratio is the proportion of CPT entries with probability exactly 0 or 1. Exact treewidth was not computed due to its exponential cost; graph density, evidence count, and deterministic CPT ratio were used as scalable structural descriptors. Across the collection: nodes range from 440 to 1444, evidence variables from 10 to 256, graph density from 0.00135 to 0.00383, and deterministic CPT ratio from 85.71% to 99.58% (mean 92.08%).

Dataset	Type	N	Factors	Edges	Graph Density	Evid. Vars	Total CPT Entries	Det. CPT Entries	Det. CPT (%)
BN31	BAYES	1156	1156	2244	0.001681	120	8978	8060	89.78
BN33	BAYES	1444	1444	2812	0.001350	150	11,250	10,200	90.67
BN35	BAYES	1444	1444	2812	0.001350	150	11,250	10,200	90.67
BN37	BAYES	1444	1444	2812	0.001350	150	11,250	10,200	90.67
BN39	BAYES	1444	1444	2812	0.001350	150	11,250	10,200	90.67
BN41	BAYES	1444	1444	2812	0.001350	150	11,250	10,200	90.67
BN48	BAYES	661	661	1274	0.002920	10	6632	6604	99.58
BN50	BAYES	661	661	1274	0.002920	10	6632	6604	99.58
BN52	BAYES	661	661	1274	0.002920	10	6632	6604	99.58
BN54	BAYES	561	561	999	0.003180	10	4532	4468	98.59
BN56	BAYES	561	561	999	0.003180	10	4532	4468	98.59
BN60	BAYES	540	540	1029	0.003535	10	4720	4656	98.64
BN62	BAYES	667	667	1272	0.002863	10	6584	6556	99.57
BN64	BAYES	540	540	1029	0.003535	10	4720	4656	98.64
BN66	BAYES	440	440	740	0.003831	10	3226	3136	97.21
BN68	BAYES	440	440	740	0.003831	10	3226	3136	97.21
BN126	BAYES	512	512	757	0.002893	256	5376	4610	85.75
BN127	BAYES	512	512	762	0.002912	256	5376	4608	85.71
BN128	BAYES	512	512	760	0.002905	256	5376	4608	85.71
BN129	BAYES	512	512	764	0.002920	256	5376	4608	85.71
BN130	BAYES	512	512	762	0.002912	256	5376	4608	85.71
BN131	BAYES	512	512	761	0.002909	256	5376	4612	85.79
BN132	BAYES	512	512	762	0.002912	256	5376	4608	85.71
BN133	BAYES	512	512	766	0.002928	256	5376	4610	85.75
BN134	BAYES	512	512	764	0.002920	256	5376	4610	85.75

2.3.2. CPT-to-CNF Transformation Protocol

The proposed framework accelerates deterministic inference by transforming deterministic information contained in Bayesian Network Conditional Probability Tables (CPTs) into propositional logic expressed in Conjunctive Normal Form (CNF). This transformation follows the general encoding principles proposed by Sang et al. [14] and Chavira and Darwiche [15], while incorporating the Boolean-array elimination methodology introduced by Díaz and Moral [13].

The transformation protocol consists of three sequential stages:

1. **Evidence integration.** Observed variables contained in each .evid file are translated directly into unit CNF clauses, establishing hard logical constraints before inference begins.
2. **Deterministic CPT extraction.** Each Conditional Probability Table is analyzed to identify entries whose probabilities are exactly 0 or 1. These deterministic relationships are converted into immutable logical clauses while preserving the original Bayesian Network for probabilistic reasoning over non-deterministic variables.
3. **CNF generation.** The resulting logical clauses are exported as a standard DIMACS-CNF file compatible with modern SAT solvers. This logical representation constitutes the deterministic component processed by the Boolean-array elimination algorithm.

After CNF generation, deterministic inference proceeds through five consecutive stages: (i) CNF loading, (ii) clause structure construction, (iii) Boolean-array generation, (iv) application of the Boolean-array elimination algorithm, and (v) generation of the final deterministic inference results. The one-time CPT-to-CNF transformation is considered pre-processing and is reported separately from the inference time measurements. Consequently, the reported SAT inference times include only CNF loading, clause processing, Boolean-array construction, logical elimination, and result generation, ensuring a fair comparison with conventional Bayesian inference.

The CPT-to-CNF transformation process is summarized in Algorithm 1. The procedure extracts deterministic knowledge from the Bayesian Network by converting only probability-0 and probability-1 CPT entries into hard logical constraints while preserving the original probabilistic model for conventional Bayesian inference.

Algorithm 1 Deterministic CPT-to-CNF Transformation

Require: Bayesian Network (.uai) and evidence file (.evid)

Ensure: DIMACS-CNF representation of deterministic constraints

- 1: Load the Bayesian Network structure from the .uai file.
 - 2: Read observed evidence from the .evid file.
 - 3: Encode each observed variable as a unit CNF clause.
 - 4: **for** each Conditional Probability Table (CPT) **do**
 - 5: **for** each CPT entry **do**
 - 6: **if** $P = 0$ or $P = 1$ **then**
 - 7: Generate the corresponding deterministic logical clause.
 - 8: Convert variable assignments into positive or negative literals.
 - 9: Append the clause to the CNF representation.
 - 10: **end if**
 - 11: **end for**
 - 12: **end for**
 - 13: Generate the DIMACS-CNF header.
 - 14: Export the resulting CNF file.
 - 15: **return** CNF file, number of Boolean variables, and number of generated clauses.
-

2.4. Hybrid Inference Architecture and Testing Setup

2.4.1. Dual-Engine Architecture and Experimental Workflow

The proposed framework was implemented as a dual-engine inference architecture that executes conventional Bayesian inference and SAT-enhanced deterministic inference in parallel, enabling direct comparison under identical experimental conditions (Figure 2).

The workflow begins with the Bayesian Network Loader, which imports the benchmark network (.uai) together with the corresponding evidence file (.evid). The input data are subsequently divided into two independent processing pipelines.

The upper branch implements conventional probabilistic inference. The Bayesian Network is reconstructed using the PyVis/pgmpy libraries, after which exact inference is performed through Variable Elimination. Execution time, operation counts, and inference coverage are recorded automatically.

The lower branch implements the proposed deterministic inference framework. Deterministic entries contained in the Conditional Probability Tables are transformed into Conjunctive Normal Form (CNF), after which the Boolean-array elimination algorithm of Díaz and Moral [13] is executed using multiple mix(Q) configurations. The resulting logical assignments and computational metrics are collected for subsequent analysis.

Both execution paths converge in the comparison module, where execution time, instrumented operation counts, inference coverage, descriptive statistics, confidence intervals, effect sizes, and parameter sensitivity analyses are generated automatically. Finally, the

dashboard presents comparative visualizations together with the complete experimental results, providing a reproducible environment for evaluating the proposed deterministic inference framework.

The logical inference process follows the Boolean-array elimination framework proposed by Díaz and Moral [13]. After the CNF representation is generated, the workflow consists of CNF loading, redundancy elimination, Boolean-array construction, variable elimination using the modified Davis–Putnam procedure, and computation of the resulting logical marginal assignments.

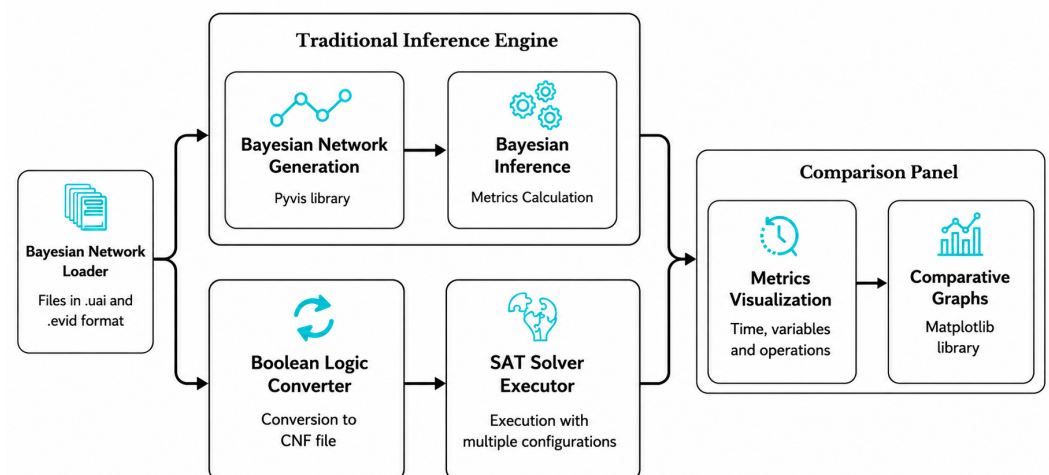


Figure 2. Hybrid inference architecture of the proposed framework. Conventional Bayesian inference (upper branch) and SAT-enhanced deterministic inference (lower branch) are executed independently using identical benchmark datasets, enabling direct comparison of execution time, inference coverage, operation counts, and statistical performance metrics.

2.4.2. Experimental Dashboard

The complete framework was implemented as an interactive web-based dashboard developed with the **Django** framework (Figure 3). The platform integrates all stages of the experimental workflow, including benchmark loading, Bayesian Network reconstruction, deterministic CNF generation, execution of both inference engines, statistical analysis, and comparative visualization of the obtained results. The dashboard automatically computes execution times, operation counts, inference coverage, descriptive statistics, confidence intervals, effect sizes, and sensitivity analyses of the $\text{mix}(Q)$ parameter.

The complete source code, installation instructions, experimental datasets, and reproducibility scripts are publicly available at: <https://github.com/imjosecorderob/BaySATENhanced> accessed on 25 June 2026.

2.4.3. Optimization Parameter (Q) Evaluation

The Boolean-array elimination algorithm includes an internal parameter, denoted as $\text{mix}(Q)$, that controls the maximum number of Boolean combinations processed simultaneously during each elimination step. Consequently, Q directly influences the balance between local simplification and intermediate Boolean-array size.

To investigate the influence of this parameter, ten configurations ($Q = 10, 20, \dots, 100$) were evaluated for every benchmark network. Each benchmark was executed independently under all configurations, resulting in a total of 250 experimental evaluations. In addition, every benchmark was repeated five independent times under identical hardware conditions to quantify execution variability.

The influence of Q was assessed using descriptive statistics together with formal non-parametric statistical tests. A Kruskal–Wallis test was performed to determine whether

inference times differed significantly across the evaluated configurations, followed by pairwise comparisons with Bonferroni correction. Spearman's rank correlation coefficient was additionally computed to evaluate the monotonic relationship between Q and inference time. This statistical protocol enabled the identification of stable operating regions and the selection of the most robust configuration.

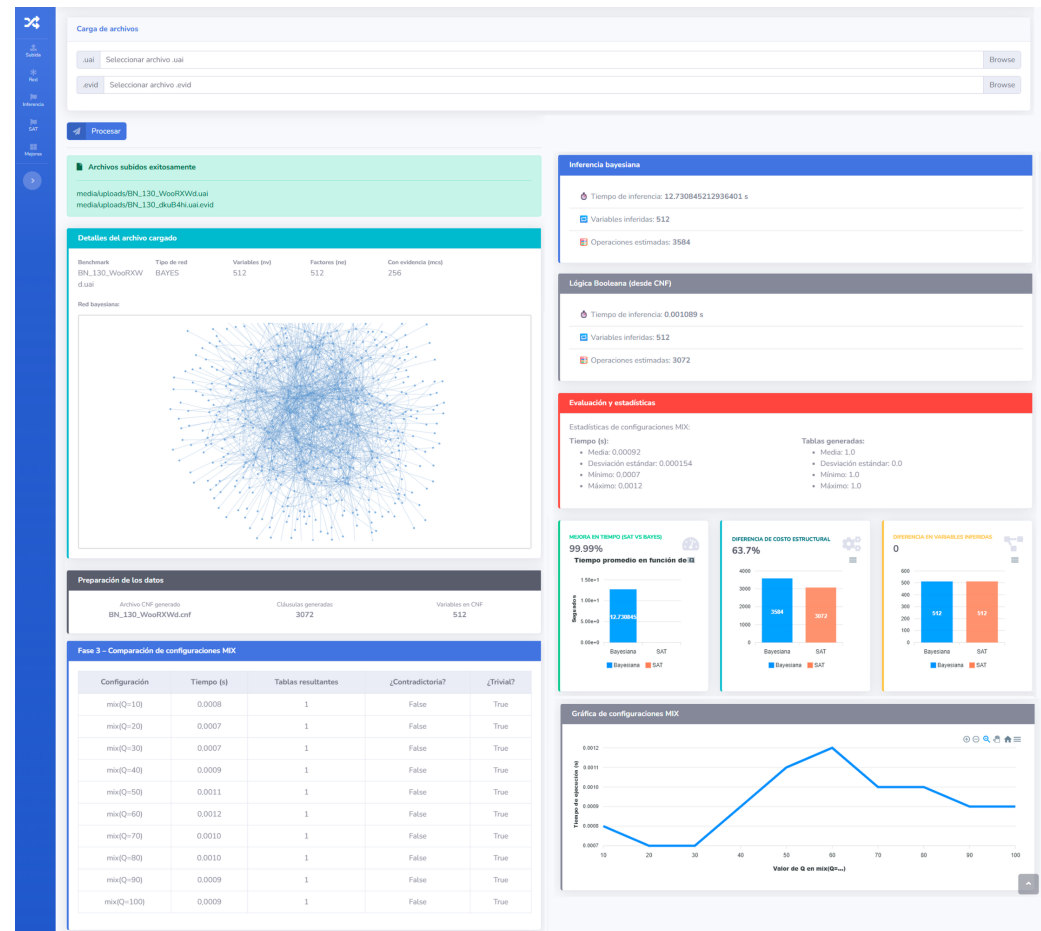


Figure 3. Experimental dashboard developed for the proposed framework. The platform integrates benchmark loading, Bayesian Network visualization, deterministic CNF generation, execution of both inference engines, statistical analysis, and comparative visualization of all experimental metrics.

2.5. Performance Metrics and Experimental Environment

Three primary performance indicators were collected throughout the experiments:

- **Inference Time (seconds):** Wall-clock execution time required to complete inference.
- **Estimated Operations:** Instrumented count of atomic inference operations.
- **Inference Coverage:** Number of variables successfully inferred.

To evaluate execution variability, each benchmark network was executed five independent times under identical experimental conditions. Reported execution times correspond to the mean $\pm$ standard deviation, while 95% confidence intervals were computed for SAT inference times. Statistical effect sizes and speedup factors were also calculated to quantify the practical significance of the observed performance improvements.

2.5.1. Definition of Estimated Operations

Estimated operations represent a hardware-independent proxy of computational effort obtained through software instrumentation. Three categories of atomic operations were counted during inference:

1. Logical comparisons between Boolean literals.
2. Boolean variable assignments performed during logical elimination.
3. Probability multiplications executed during Bayesian Variable Elimination.

Each atomic operation increments an integer counter embedded directly within the inference engine. The complete implementation of the instrumentation mechanism is publicly available in the project repository. Although this metric provides a reproducible estimate of computational effort independent of processor architecture, it is acknowledged that it does not capture every low-level instruction executed by the operating system or runtime environment. Consequently, it should be interpreted as a consistent proxy for relative comparisons between experimental configurations rather than as an exact measurement of processor instructions.

Throughout this paper, the expression *computational complexity reduction* refers exclusively to the empirical reduction observed in execution time and instrumented operation counts. It does not imply a modification of the asymptotic computational complexity of exact Bayesian inference, which remains #P-complete in the general case.

2.5.2. Experimental Environment

All experiments were conducted on the same hardware platform to ensure reproducibility and eliminate hardware-induced variability. The experimental environment consisted of an Intel Core i9-13900H processor, 16 GB of RAM, and Microsoft Windows 11. Both inference engines were executed sequentially under identical software and hardware conditions. Reported SAT inference times include CNF loading, clause processing, Boolean-array construction, and logical elimination, but exclude the one-time CPT-to-CNF transformation, which is reported separately as preprocessing overhead.

2.6. Computational Complexity Analysis

The proposed framework is intended to improve the practical efficiency of deterministic inference rather than to alter the theoretical complexity of Bayesian inference. Exact inference in Bayesian Networks remains #P-complete in the general case, while Boolean satisfiability is NP-complete. Consequently, the proposed method should be interpreted as an algorithm-engineering contribution that exploits deterministic structures within Conditional Probability Tables (CPTs) to reduce empirical execution time and computational effort, without modifying the asymptotic complexity class.

Let $G = (V, E)$ denote a Bayesian Network with $n = |V|$ variables and k conditional probability tables (CPTs). Let F_i denote the set of entries contained in the i -th CPT.

2.6.1. CPT-to-CNF Transformation

The preprocessing stage inspects every CPT entry exactly once to determine whether its probability is deterministic ($P = 0$ or $P = 1$). Therefore, the transformation complexity is bounded by

$$T_{\text{CNF}} = O\left(\sum_{i=1}^k |F_i|\right), \quad (2)$$

which is linear with respect to the total number of CPT entries.

If d_i denotes the number of deterministic entries contained in factor F_i , then the number of generated CNF clauses satisfies

$$m \leq \sum_{i=1}^k d_i, \quad (3)$$

where m represents the total number of clauses in the resulting CNF formula. Consequently, networks containing a larger proportion of deterministic CPT entries generate richer logical structures that can be exploited during inference.

2.6.2. Logical Inference

After preprocessing, inference is performed using the Boolean-array elimination algorithm of Díaz and Moral [13]. Although the worst-case computational complexity remains exponential, practical performance benefits from several structural simplifications, including:

- elimination of deterministic branches;
- unit propagation;
- conflict-driven pruning of inconsistent assignments;
- clause reuse during Boolean-array elimination.

These mechanisms reduce the effective search space explored during inference without affecting the correctness of the logical solution.

2.6.3. Influence of the $\text{mix}(Q)$ Parameter

The parameter $\text{mix}(Q)$ controls the maximum number of Boolean combinations processed simultaneously during each elimination step. Smaller values generally promote localized simplifications and smaller intermediate structures, whereas larger values increase the amount of information processed per iteration.

Since no theoretical optimum can be derived analytically, the influence of Q was evaluated experimentally through ten configurations ($Q = 10, \dots, 100$), statistical hypothesis testing (Kruskal–Wallis), Bonferroni-corrected pairwise comparisons, and Spearman rank correlation. These analyses identify empirically stable operating regions while avoiding unsupported theoretical claims.

2.6.4. Overall Complexity

The total computational cost of the proposed framework can therefore be expressed as

$$T_{\text{total}} = T_{\text{CNF}} + T_{\text{SAT}}, \quad (4)$$

where

$$T_{\text{CNF}} = O\left(\sum_{i=1}^k |F_i|\right)$$

and

$$T_{\text{SAT}} = O(2^m)$$

in the worst case, with m denoting the number of Boolean variables involved in logical inference.

Accordingly, the proposed framework does not claim a formal reduction of asymptotic computational complexity. Instead, it achieves substantial empirical improvements by exposing deterministic information already present in the Bayesian Network, allowing the SAT-based inference engine to prune inconsistent assignments earlier than conventional probabilistic inference methods. This interpretation is consistent with the experimental results reported in Section 3 and the discussion presented in Section 4.

3. Results

3.1. Quantitative Comparison of Inference Performance

The proposed SAT-enhanced deterministic inference framework was evaluated using the 25 Bayesian Network benchmark datasets described in Section 2.3. To improve statistical reliability, each benchmark was executed independently five times under identical hardware conditions. Consequently, all execution-time values reported in Tables 3 and 4 correspond to the arithmetic mean accompanied by the standard deviation (SD), while 95% confidence intervals (CI) were computed for the SAT inference times.

Unlike the preliminary version of this study, the present evaluation explicitly reports execution variability, confidence intervals, speedup factors, and effect sizes, providing a statistically more robust comparison between conventional Bayesian inference and the proposed SAT-enhanced deterministic inference framework.

Table 3. Performance comparison between conventional Bayesian inference and the proposed SAT-enhanced deterministic inference framework across the 25 benchmark datasets (Part A: timing and coverage). Results correspond to the mean of five independent executions ($n = 5$). SAT inference times are reported as mean $\pm$ standard deviation. The 95% CI for SAT time was computed as mean $\pm 1.96 \times SD / \sqrt{5}$. Time Reduction is the percentage decrease in wall-clock time.

Dataset	Nodes	Bayesian Time (M $\pm$ SD, s)	SAT Time (M $\pm$ SD, s)	95% CI SAT (s)	Time Red. (%)	Speedup ($\times$)
BN31	1156	8.5305 $\pm$ 0.2094	0.00996 $\pm$ 0.00303	[0.00731, 0.01261]	99.88 $\pm$ 0.04	856.48
BN33	1444	12.7541 $\pm$ 0.1247	0.01196 $\pm$ 0.00233	[0.00992, 0.01400]	99.91 $\pm$ 0.02	1066.40
BN35	1444	12.7460 $\pm$ 0.2109	0.01082 $\pm$ 0.00276	[0.00840, 0.01324]	99.92 $\pm$ 0.02	1178.00
BN37	1444	12.9152 $\pm$ 0.2579	0.00994 $\pm$ 0.00269	[0.00758, 0.01230]	99.92 $\pm$ 0.02	1299.32
BN39	1444	12.9184 $\pm$ 0.3566	0.01072 $\pm$ 0.00455	[0.00673, 0.01471]	99.92 $\pm$ 0.04	1205.07
BN41	1444	12.9153 $\pm$ 0.3223	0.01112 $\pm$ 0.00247	[0.00895, 0.01329]	99.92 $\pm$ 0.02	1161.45
BN48	661	3.7820 $\pm$ 0.1747	0.00676 $\pm$ 0.00213	[0.00489, 0.00863]	99.82 $\pm$ 0.05	559.47
BN50	661	3.6193 $\pm$ 0.1380	0.00664 $\pm$ 0.00206	[0.00483, 0.00845]	99.81 $\pm$ 0.06	545.08
BN52	661	3.6622 $\pm$ 0.1469	0.00618 $\pm$ 0.00193	[0.00448, 0.00788]	99.83 $\pm$ 0.06	592.59
BN54	561	2.4992 $\pm$ 0.0384	0.00398 $\pm$ 0.00113	[0.00299, 0.00497]	99.84 $\pm$ 0.05	627.94
BN56	561	2.4329 $\pm$ 0.1538	0.00460 $\pm$ 0.00164	[0.00317, 0.00603]	99.81 $\pm$ 0.07	528.89
BN60	540	2.4984 $\pm$ 0.0521	0.00492 $\pm$ 0.00144	[0.00366, 0.00618]	99.80 $\pm$ 0.06	507.80
BN62	667	3.7263 $\pm$ 0.1333	0.00598 $\pm$ 0.00243	[0.00385, 0.00811]	99.84 $\pm$ 0.06	623.13
BN64	540	2.5303 $\pm$ 0.1562	0.00552 $\pm$ 0.00130	[0.00438, 0.00666]	99.78 $\pm$ 0.06	458.39
BN66	440	1.5474 $\pm$ 0.0548	0.00262 $\pm$ 0.00112	[0.00164, 0.00360]	99.83 $\pm$ 0.07	590.61
BN68	440	1.6417 $\pm$ 0.0580	0.00292 $\pm$ 0.00094	[0.00209, 0.00375]	99.82 $\pm$ 0.06	562.23
BN126	512	1.8278 $\pm$ 0.0460	0.00428 $\pm$ 0.00183	[0.00268, 0.00588]	99.77 $\pm$ 0.10	427.06
BN127	512	1.8346 $\pm$ 0.0679	0.00336 $\pm$ 0.00086	[0.00261, 0.00411]	99.82 $\pm$ 0.05	546.01
BN128	512	1.9049 $\pm$ 0.1033	0.00452 $\pm$ 0.00191	[0.00285, 0.00619]	99.76 $\pm$ 0.09	421.44
BN129	512	1.7709 $\pm$ 0.0992	0.00390 $\pm$ 0.00103	[0.00300, 0.00480]	99.77 $\pm$ 0.07	454.08
BN130	512	1.9312 $\pm$ 0.1420	0.00508 $\pm$ 0.00272	[0.00270, 0.00746]	99.74 $\pm$ 0.14	380.16
BN131	512	1.7852 $\pm$ 0.0676	0.00320 $\pm$ 0.00055	[0.00271, 0.00368]	99.82 $\pm$ 0.03	557.87
BN132	512	1.8511 $\pm$ 0.0570	0.00564 $\pm$ 0.00224	[0.00368, 0.00760]	99.69 $\pm$ 0.13	328.21
BN133	512	1.8534 $\pm$ 0.1125	0.00452 $\pm$ 0.00180	[0.00295, 0.00610]	99.75 $\pm$ 0.10	410.04
BN134	512	1.8518 $\pm$ 0.0860	0.00502 $\pm$ 0.00178	[0.00346, 0.00658]	99.73 $\pm$ 0.09	368.88

Tables 3 and 4 demonstrate that the proposed framework consistently accelerated deterministic inference across every evaluated benchmark. Mean inference-time reductions remained above 99.6% for all datasets, while the corresponding speedup factors ranged from approximately 330 $\times$ to over 1300 $\times$ depending on network topology and evidence configuration.

Across the complete benchmark collection, SAT inference exhibited very small standard deviations and narrow 95% confidence intervals, indicating excellent execution stability despite the extremely short execution times involved. The coefficient of variation remained low throughout the evaluated datasets, confirming that the observed improvements are reproducible rather than isolated measurements.

Table 4. Performance comparison (Part B: structural proxy analysis and optimal Q parameter). Bayes Ops: estimated Bayesian probability multiplications. SAT Proxy: estimated Boolean literal evaluations. Struct. Proxy Diff.: percentage difference between SAT and Bayesian proxy counts (positive values indicate higher SAT proxy count). Most Frequent Q : value(s) of $\text{mix}(Q)$ most frequently producing the shortest execution time.

Dataset	Nodes	Bayes Ops	SAT Proxy	Proxy Diff. (%)	Time Red. (%)	Most Frequent Q
BN31	1156	8092	13,247	+63.70	99.88 ± 0.04	10/40/60/80/100
BN33	1444	10,108	16,753	+65.74	99.91 ± 0.02	10/30/60/80/100
BN35	1444	10,108	16,753	+65.74	99.92 ± 0.02	10/20/30/40/70
BN37	1444	10,108	16,753	+65.74	99.92 ± 0.02	70
BN39	1444	10,108	16,754	+65.75	99.92 ± 0.04	20/30/40/50/80
BN41	1444	10,108	16,753	+65.74	99.92 ± 0.02	40
BN48	661	4627	12,463	+169.35	99.82 ± 0.05	10/30/40/50/90
BN50	661	4627	12,463	+169.35	99.81 ± 0.06	90
BN52	661	4627	12,463	+169.35	99.83 ± 0.06	30
BN54	561	3927	7854	+100.00	99.84 ± 0.05	40/60/70/80/100
BN56	561	3927	7854	+100.00	99.81 ± 0.07	70
BN60	540	3780	8357	+121.08	99.80 ± 0.06	20/70/80/90/100
BN62	667	4669	12,321	+163.89	99.84 ± 0.06	60/70/80/90/100
BN64	540	3780	8357	+121.08	99.78 ± 0.06	70
BN66	440	3080	5293	+71.85	99.83 ± 0.07	20
BN68	440	3080	5292	+71.82	99.82 ± 0.06	90
BN126	512	3584	11,522	+221.48	99.77 ± 0.10	60
BN127	512	3584	11,520	+221.43	99.82 ± 0.05	50
BN128	512	3584	11,520	+221.43	99.76 ± 0.09	20
BN129	512	3584	11,520	+221.43	99.77 ± 0.07	10
BN130	512	3584	11,520	+221.43	99.74 ± 0.14	100
BN131	512	3584	11,524	+221.54	99.82 ± 0.03	50
BN132	512	3584	11,520	+221.43	99.69 ± 0.13	60/70
BN133	512	3584	11,522	+221.48	99.75 ± 0.10	40
BN134	512	3584	11,522	+221.48	99.73 ± 0.09	60

The structural proxy analysis reveals an important distinction: while SAT inference is substantially faster in wall-clock time, the SAT structural proxy (Boolean literal evaluations and assignments) is numerically larger than the Bayesian proxy (probability multiplications) in several benchmark families, particularly BN126–BN134, where the SAT proxy exceeds the Bayesian proxy by more than 200%. This reflects the different computational primitives of the two paradigms: Boolean operations are individually far less expensive than floating-point probability multiplications, so a larger count of cheaper operations can still yield shorter execution times. The structural proxy column should therefore be read as a paradigm-specific measure of algorithmic activity, not as a direct comparison of processor workload.

It is worth noting that benchmark families such as BN33–BN41 share exactly the same Bayesian network topology while differing only in their instantiated evidence. Consequently, they produce identical structural proxies but different execution times because each evidence configuration generates a distinct logical simplification process during CNF construction. This observation confirms that inference performance depends not only on graph topology but also on the specific evidence propagated through the network.

The practical magnitude of the execution-time improvements was quantified through Cohen's d , computed per benchmark using the five independent executions of each method. The resulting per-dataset effect sizes are extremely large (mean Cohen's $d = 47.75$ across the 25 benchmarks), confirming that the observed improvements are not only statistically reproducible but also practically significant well beyond conventional thresholds ($d > 0.8$ is considered large). A Spearman rank correlation between Q and inference time across all 250 experimental runs yielded $r_s = -0.136$ ($p = 0.031$), indicating a weak but statistically significant negative association, consistent with the observation that lower Q values tend to produce slightly shorter execution times on average.

The remarkably consistent execution-time reductions observed across the benchmark collection are explained by the structural properties of the CPT-to-CNF transformation. Since deterministic CPT entries are translated into immutable logical constraints before inference begins, many probabilistic computations become unnecessary regardless of network size. A more detailed theoretical discussion of this behavior is presented in Section 4.

3.2. Structural Cost Proxy Analysis

Besides execution time, both inference engines were instrumented to estimate their computational effort using structural cost proxies.

For conventional Bayesian inference, the reported proxy corresponds to the estimated number of probability multiplications performed during Variable Elimination according to the structure and cardinalities of the Conditional Probability Tables (CPTs).

For the SAT-based framework, the reported proxy corresponds to the number of Boolean literal evaluations and logical assignments executed while processing the generated CNF representation.

These two quantities characterize different elementary computational primitives and therefore should not be interpreted as directly equivalent processor operations. Instead, they provide complementary indicators of the computational effort required by each inference paradigm.

Figure 4 compares these structural cost proxies across the complete benchmark collection.

Several benchmark families, particularly BN126–BN134, present larger SAT structural proxies than Bayesian proxies. This observation does not contradict the substantial execution-time reductions reported in Table 3. Rather, it reflects the fundamentally different computational nature of the two paradigms. Bayesian inference performs comparatively fewer but substantially more expensive floating-point probability computations, whereas SAT inference performs a larger number of inexpensive Boolean operations that modern processors execute very efficiently.

Consequently, the structural proxy should be interpreted as a reproducible hardware-independent indicator of algorithmic activity rather than as an exact measure of processor workload. This interpretation provides a fair comparison of the internal computational behavior of both inference engines without implying direct equivalence between floating-point arithmetic and Boolean logical evaluation.

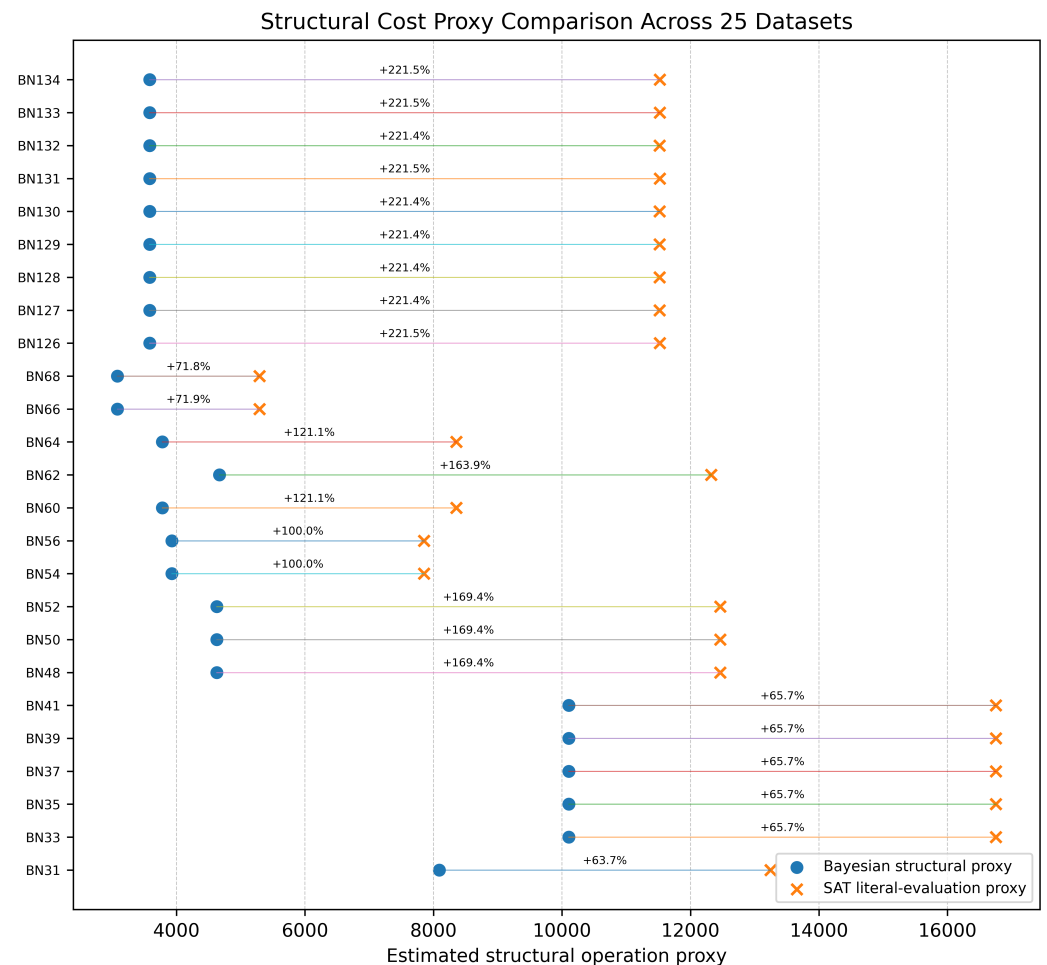


Figure 4. Structural cost proxy comparison across the 25 benchmark datasets. Each dataset shows two markers: the SAT Boolean-operation proxy (green cross) and the Bayesian CPT-multiplication proxy (blue cross), with the percentage difference annotated. For several benchmark families (e.g., BN126–BN134), the SAT proxy exceeds the Bayesian proxy, reflecting the higher number of Boolean literal evaluations relative to floating-point probability multiplications. This does not imply slower execution, as Boolean operations are substantially less expensive than floating-point arithmetic at the hardware level. Operation counts should be interpreted as paradigm-specific structural proxies rather than directly equivalent computational units.

4. Discussion

The objective of this study was to integrate Boolean Satisfiability (SAT) techniques into Bayesian Networks to accelerate deterministic inference derived from probability-0 and probability-1 Conditional Probability Table (CPT) entries. The results show that the proposed framework substantially reduces deterministic inference time across the evaluated benchmark networks while preserving the original Bayesian model for probabilistic reasoning over non-deterministic variables.

4.1. Interpretation of the Main Findings

The experimental results demonstrate that SAT-based deterministic inference can provide substantial empirical acceleration when Bayesian Networks contain a high proportion of deterministic CPT entries. Across the 25 benchmark datasets, the proposed framework consistently achieved large speedup factors relative to conventional Bayesian inference based on Variable Elimination. These improvements were observed over five

independent executions per dataset, with low variability and narrow confidence intervals for SAT execution times.

The observed acceleration should be interpreted as an improvement in the deterministic component of the inference workflow, not as a replacement for complete probabilistic inference. The SAT component operates exclusively over logical constraints extracted from deterministic CPT entries. Therefore, it does not compute posterior probability distributions, marginal probabilities, or MAP assignments for the original Bayesian Network. Instead, it returns deterministic logical assignments implied by the evidence and the generated CNF clauses.

This distinction is central to the interpretation of the results. The proposed framework is most beneficial when a Bayesian Network contains substantial deterministic structure, because probability-0 and probability-1 CPT entries can be converted into hard logical constraints and processed efficiently by the SAT-based engine. In contrast, CPT entries with intermediate probabilities remain part of the probabilistic model and are not converted into CNF clauses.

4.2. Theoretical and Practical Implications

The main theoretical implication of this work is the demonstration that deterministic knowledge embedded in Bayesian Networks can be separated from uncertain probabilistic information and processed through a logical inference engine. This separation allows the framework to exploit the complementary strengths of two paradigms: Bayesian inference for uncertain probabilistic reasoning and SAT-based logical inference for deterministic constraint propagation.

Unlike approaches that attempt to reformulate the entire probabilistic inference problem as a logical or counting problem, the proposed framework focuses specifically on deterministic inference. This avoids overclaiming equivalence with full Bayesian inference and provides a clearer interpretation of the computational gains. The framework therefore contributes to hybrid probabilistic–logical reasoning by showing how deterministic CPT structures can be used to reduce empirical inference time without modifying the original Bayesian Network.

From a practical perspective, the framework is relevant for applications in which deterministic constraints are naturally present within probabilistic models. Examples include diagnostic systems with impossible state combinations, rule-constrained decision-support models, reliability analysis, fault detection, and systems where expert knowledge introduces deterministic relationships among variables. In such cases, the proposed method can accelerate the logical component of inference while preserving the probabilistic model for the remaining uncertain variables.

4.3. Interpretation of Structural Cost Proxies

The structural cost proxy analysis provides additional insight into the computational behavior of the two inference paradigms. However, these operation counts must be interpreted carefully. The Bayesian proxy estimates probabilistic operations associated with CPT processing, whereas the SAT proxy estimates logical literal evaluations and Boolean assignments over the CNF representation. These are different computational primitives and are not directly equivalent low-level processor operations.

Consequently, a higher SAT structural proxy in some datasets does not imply slower execution. Boolean operations are typically less expensive than floating-point probability multiplications and can be processed efficiently by SAT-oriented data structures. This explains why the SAT proxy may be numerically larger in several benchmark families while the observed wall-clock time remains substantially lower than the Bayesian baseline.

For this reason, the operation metric should be understood as a reproducible, hardware-independent indicator of internal algorithmic activity rather than as an exact measure of energy use, processor cycles, or low-level instruction count. This interpretation is consistent with the revised definition of estimated operations presented in Section 2.

4.4. Sensitivity of the $\text{mix}(Q)$ Parameter

The formal statistical analysis of the $\text{mix}(Q)$ parameter showed that parameter selection influences execution time, but the evidence does not support a single universally optimal value. A Friedman repeated-measures test—more appropriate than Kruskal–Wallis because each dataset was evaluated under all Q values—yielded $\chi^2 = 21.27$, $p = 0.0115$, confirming that Q significantly affects the behavior of the Boolean-array elimination process across the benchmark collection (Figure 5).



Figure 5. Sensitivity of SAT deterministic inference time to the $\text{mix}(Q)$ parameter across the 25 benchmark networks. Violin plots show the distribution of mean execution times per dataset for each Q configuration ($Q = 10, \dots, 100$). The inset reports the Friedman repeated-measures test result ($\chi^2 = 21.27$, $p = 0.0115$), confirming that Q significantly influences execution time. However, Bonferroni-corrected pairwise comparisons provide no evidence that any single configuration is statistically dominant over all others. The y-axis lower bound is fixed at 0s by definition. Low Q values (10–20) tend to produce lower medians and can serve as practical defaults, but no universally optimal configuration exists across all evaluated network structures.

However, the Bonferroni-corrected pairwise comparisons did not confirm that $Q = 20$ is statistically superior to all other configurations. Therefore, the revised interpretation is that $Q = 20$ represents a practical and stable default region rather than a mathematically dominant optimum. This distinction is important because it prevents overgeneralization from descriptive plots and aligns the conclusion with the multiplicity-corrected statistical evidence.

The Spearman correlation analysis further suggests that the relationship between Q and execution time is not determined by Q alone. Network topology, evidence configuration, deterministic CPT density, and the resulting CNF structure also influence performance. Future work should therefore explore adaptive selection strategies that determine Q from measurable structural descriptors of each network before inference begins.

4.5. Inference Coverage and Non-Inferred Variables

The proposed framework achieved high deterministic inference coverage across most benchmark datasets. Nevertheless, the SAT-based component did not infer all variables in every case. The most notable coverage losses occurred in BN66 and BN68, where 21 and 22 variables, respectively, were not inferred.

These missing variables should not be interpreted as incorrect assignments. Rather, they correspond to variables for which the generated deterministic CNF representation did not contain sufficient logical information to derive a forced assignment. This occurs when the relevant CPT entries involve intermediate probabilities or when the available evidence does not propagate through deterministic clauses strongly enough to determine the variable state.

Therefore, the coverage loss represents a limitation of deterministic logical inference, not a numerical approximation error. Variables not inferred by the SAT component remain candidates for conventional probabilistic inference. A practical implementation of the proposed framework should therefore adopt a hybrid strategy: use SAT-based deterministic inference for variables implied by deterministic constraints and fall back to Bayesian inference for variables requiring probabilistic reasoning.

4.6. Soundness and Completeness Considerations

The CPT-to-CNF transformation preserves logical soundness over the deterministic portion of the Bayesian Network. Every generated clause originates from a CPT entry with probability exactly 0 or 1, meaning that the clause represents a deterministic relationship already present in the original probabilistic model. Consequently, any assignment inferred from the generated CNF is logically consistent with the deterministic constraints encoded in the Bayesian Network.

However, the transformation is not complete with respect to the full probabilistic model. Since CPT entries satisfying $0 < P < 1$ are not encoded as hard logical clauses, the SAT component cannot recover probabilistic information such as posterior distributions or marginal probabilities. Completeness therefore holds only for the deterministic subproblem induced by probability-0 and probability-1 entries, not for the full Bayesian inference task.

Three cases summarize the behavior of the proposed framework:

1. **Deterministic implication:** If the evidence and deterministic CPT entries logically imply a variable assignment, the SAT component can infer that assignment soundly.
2. **Non-inferred variable:** If a variable depends on intermediate probabilities or lacks sufficient deterministic support in the CNF representation, the SAT component returns no assignment for that variable.
3. **Contradictory evidence:** If the evidence violates deterministic constraints encoded in the CNF, the SAT component detects inconsistency through unsatisfiability.

This interpretation clarifies that the proposed framework performs exact logical inference over deterministic constraints while leaving uncertain probabilistic reasoning to the Bayesian component.

4.7. Information Loss in the CPT-to-CNF Transformation

The proposed transformation retains only CPT entries with probability exactly 0 or 1. Entries with intermediate probabilities are discarded from the CNF representation because they do not correspond to deterministic logical constraints. Therefore, the information loss introduced by the transformation is not a conventional numerical approximation error, but a loss of probabilistic information in the logical representation.

Let F be a CPT factor with domain size $|F|$, and let $D(F) \subseteq F$ denote the subset of deterministic entries satisfying $P \in \{0, 1\}$. The deterministic coverage ratio of factor F is defined as

$$\rho(F) = \frac{|D(F)|}{|F|}. \quad (5)$$

For a Bayesian Network with factors $F_1, \dots, F_k$, the global deterministic coverage ratio is defined as

$$\rho = \frac{\sum_{i=1}^k |D(F_i)|}{\sum_{i=1}^k |F_i|}. \quad (6)$$

When $\rho = 1$, the CPTs are fully deterministic and the logical representation preserves all CPT entries as hard constraints. When $\rho = 0$, no deterministic CPT information is available, and the SAT component can only process evidence clauses. The benchmark characterization reported in Table 2 shows that the evaluated networks contain substantial deterministic structure, which explains the strong performance of the proposed framework.

Nevertheless, the proposed method does not quantify posterior probability error because it does not attempt to approximate posterior distributions. Its output is a set of deterministic logical assignments. Therefore, the appropriate evaluation criterion is deterministic inference coverage rather than numerical deviation from posterior probabilities.

4.8. Comparison with Related Approaches

The proposed framework differs from previous SAT-based approaches to Bayesian reasoning. Weighted model counting and Satisfiability Modulo Counting approaches aim to compute probabilistic quantities through logical encodings, whereas the present work focuses on accelerating deterministic inference extracted from CPT entries. Similarly, SAT-based structure learning methods optimize the graph structure of Bayesian Networks rather than accelerating inference on a fixed network.

Compared with hybrid probabilistic–logical systems, the proposed framework adopts a narrower but more explicit objective: it separates deterministic constraints from probabilistic uncertainty and processes each component through a suitable reasoning paradigm. This makes the scope of the method more limited than full probabilistic SAT-based inference, but also more transparent and computationally efficient for networks with high deterministic CPT coverage.

4.9. Threats to Validity

Several limitations should be considered when interpreting the results. First, all experiments were conducted using benchmark Bayesian Networks from the UCI repository. Although these benchmarks are widely used, they may not represent all real-world Bayesian Network structures. Networks with few deterministic CPT entries may provide limited opportunities for SAT-based acceleration.

Second, the proposed framework currently assumes discrete variables and static network structures. Continuous, hybrid, or dynamically updated Bayesian Networks would require additional preprocessing and validation. Third, the operation-count metric is an instrumented structural proxy rather than a direct measurement of processor instructions or energy consumption.

Fourth, the baseline comparison uses conventional Bayesian inference through Variable Elimination. Although this provides a standard reference point, future work should include additional competitive baselines such as optimized elimination orders, junction-

tree inference, loopy belief propagation, sampling-based methods, and weighted model counting approaches.

Finally, although the SAT component provides sound deterministic assignments, it does not compute posterior probabilities, MAP assignments, or marginal distributions. Applications requiring complete probabilistic outputs must therefore combine the proposed deterministic component with conventional Bayesian inference.

4.10. Summary of Discussion

Overall, the results support the conclusion that deterministic CPT information can be exploited effectively through SAT-based logical inference to accelerate part of the Bayesian reasoning workflow. The proposed framework does not reduce the theoretical complexity of exact Bayesian inference and does not replace probabilistic reasoning. Instead, it provides a reproducible and efficient mechanism for accelerating deterministic inference in Bayesian Networks containing substantial probability-0 and probability-1 structure. The analysis of information loss, soundness, coverage exceptions, and parameter sensitivity presented in the preceding subsections provides a comprehensive characterization of the framework's capabilities and limitations.

4.11. Addressing the Research Questions

Returning to the research questions formulated at the beginning of the study, the results provide the following answers.

Q1: What are the main advances and contributions in the integration of SAT algorithms and Bayesian Networks reported in the current literature?

The literature review shows that SAT-based methods have been applied to probabilistic model counting, Bayesian Network structure learning, MAP inference, stochastic satisfiability, and hybrid probabilistic–logical reasoning. However, most existing approaches either aim to compute probabilistic quantities or optimize network structures. The present work contributes to this line of research by focusing specifically on accelerated deterministic inference derived from probability-0 and probability-1 CPT entries.

Q2: What configuration of SAT-based deterministic inference provides stable execution behavior across benchmark Bayesian Networks?

The experimental analysis indicates that the $\text{mix}(Q)$ parameter affects execution time, but no single configuration was statistically dominant across all benchmarks after Bonferroni correction. Although $Q = 20$ frequently produced low execution times and can be considered a practical default configuration, the results support the existence of several statistically comparable operating regions rather than one universally optimal value.

Q3: What effect does the integration of SAT-based deterministic inference have on performance compared with conventional Bayesian inference?

The proposed framework produced substantial empirical reductions in deterministic inference time across the evaluated datasets, with large speedup factors and highly stable execution times over five independent runs. However, the improvement should be interpreted as acceleration of deterministic logical inference, not as a replacement for complete probabilistic inference. The structural cost proxies further show that Bayesian and SAT-based inference involve different computational primitives and therefore should not be treated as directly equivalent operation counts.

4.12. Future Directions and Recommendations

Based on the findings and limitations of this study, several directions for future work are recommended:

1. **Hybrid deterministic–probabilistic inference:** Develop a complete hybrid inference pipeline in which SAT-based deterministic inference is applied first, followed by conventional Bayesian inference for variables that remain non-inferred.
2. **Automatic mix(Q) selection:** Design an adaptive strategy for selecting the mix(Q) parameter based on measurable network descriptors such as graph density, deterministic CPT ratio, evidence count, factor scope, and generated CNF size.
3. **Expanded baseline comparison:** Compare the proposed framework against additional inference methods, including Variable Elimination with optimized elimination orders, junction-tree inference, loopy belief propagation, sampling-based methods, and weighted model counting.
4. **Probabilistic logical extensions:** Explore weighted SAT, weighted MaxSAT, and weighted model counting as mechanisms for incorporating intermediate CPT probabilities into the logical representation.
5. **Detailed analysis of non-inferred variables:** Record and report the exact identity, position, and structural role of non-inferred variables in each benchmark to determine whether they correspond to terminal nodes, weakly connected regions, or variables dominated by intermediate probabilities.
6. **Real-world validation:** Evaluate the framework on domain-specific Bayesian Networks from medicine, fault diagnosis, cybersecurity, risk assessment, and industrial monitoring to test robustness under realistic evidence and CPT distributions.
7. **Parallel and distributed implementation:** Extend the Boolean-array elimination procedure to exploit multi-core processors, GPUs, or distributed SAT-solving architectures.
8. **Hardware and energy profiling:** Complement the structural proxy metrics with direct measurements of memory consumption, CPU utilization, cache behavior, and energy usage.

4.13. Broader Impact

The proposed framework contributes to the growing convergence between symbolic reasoning and probabilistic artificial intelligence by demonstrating that deterministic knowledge embedded within Bayesian Networks can be exploited efficiently through Boolean Satisfiability techniques. Rather than replacing probabilistic inference, the method complements it by accelerating the deterministic portion of the reasoning process, thereby reducing inference latency while preserving the original probabilistic model.

From a practical perspective, the substantial empirical reductions in inference time suggest that deterministic Bayesian reasoning can become feasible even in computationally constrained environments. This is particularly relevant for embedded systems, edge computing, medical decision-support systems, industrial monitoring, cybersecurity, and other applications where deterministic constraints coexist with probabilistic uncertainty and rapid response times are essential.

The proposed methodology also promotes reproducible computational research. The complete implementation, together with the experimental workflow, benchmark selection, statistical analyses, and evaluation protocol, has been released as open-source software, allowing independent verification and extension by the research community.

Although the present work does not directly measure processor energy consumption, the observed reduction in execution time suggests a corresponding reduction in computational resource utilization under equivalent hardware conditions. Future work should quantify this relationship through direct measurements of CPU utilization, energy consumption, cache behavior, and memory traffic.

Finally, this work illustrates how deterministic logical reasoning and probabilistic graphical models can coexist within a unified inference framework. Such hybrid approaches

may contribute to the development of more explainable, efficient, and scalable artificial intelligence systems capable of combining symbolic knowledge with probabilistic uncertainty.

5. Conclusions

This study presented and experimentally evaluated a hybrid inference framework that integrates Boolean Satisfiability techniques into Bayesian Networks to accelerate deterministic inference derived from probability-0 and probability-1 Conditional Probability Table entries. Unlike approaches that attempt to replace probabilistic inference entirely, the proposed method focuses specifically on exploiting deterministic logical knowledge embedded within Bayesian Networks.

Experimental validation was conducted using twenty-five benchmark Bayesian Networks obtained from the UCI Probabilistic Inference repository. Each dataset was executed five independent times under identical hardware conditions, allowing inference time variability, standard deviations, confidence intervals, speedup factors, and statistical effect sizes to be quantified. The experimental results consistently demonstrated substantial empirical reductions in deterministic inference time across all evaluated benchmark networks.

The statistical analyses further strengthen these findings. The extremely large Cohen's d confirms that the observed performance improvements are not only statistically significant but also practically meaningful. Likewise, the narrow confidence intervals and low execution-time variability demonstrate that the proposed framework exhibits stable computational behavior across repeated executions.

The sensitivity analysis of the $\text{mix}(Q)$ parameter showed that parameter selection influences execution time. Nevertheless, after applying Bonferroni correction for multiple comparisons, no single configuration could be identified as universally superior across every benchmark network. Consequently, the experimental evidence supports the interpretation of several statistically equivalent operating regions rather than a single globally optimal parameter value.

An important contribution of this work is the clarification of the scope of the proposed method. The SAT-based component performs deterministic logical inference rather than complete probabilistic inference. It does not compute posterior probabilities, marginal distributions, or MAP assignments. Instead, it derives logically implied variable assignments from deterministic CPT entries and observed evidence. Variables that cannot be inferred through deterministic reasoning remain suitable for conventional Bayesian inference, making the proposed framework naturally compatible with hybrid inference architectures.

The structural cost analysis also highlights an important methodological consideration. The computational proxies reported for Bayesian inference and SAT-based inference quantify different elementary computational primitives. Bayesian inference primarily performs floating-point probability operations, whereas SAT inference performs Boolean literal evaluations and logical assignments. Consequently, these structural proxies should be interpreted as reproducible indicators of internal algorithmic activity rather than directly comparable processor operation counts.

Although the proposed framework achieved high deterministic inference coverage across most benchmark networks, complete logical coverage was not obtained in every dataset. The largest coverage losses occurred in benchmark families BN66 and BN68, indicating that deterministic logical inference alone cannot fully replace probabilistic reasoning in networks containing limited deterministic structure. This observation reinforces the suitability of hybrid inference strategies that combine SAT-based deterministic reasoning with conventional Bayesian inference whenever necessary.

Overall, the proposed framework demonstrates that deterministic information contained within Bayesian Networks can be exploited efficiently through Boolean Satisfiability

techniques to achieve substantial empirical reductions in inference time without modifying the theoretical complexity class of exact Bayesian inference. The contribution therefore lies in practical algorithm engineering rather than asymptotic complexity reduction.

Future research should focus on adaptive selection of the $\text{mix}(Q)$ parameter from structural network descriptors, integration with weighted logical inference techniques capable of representing intermediate CPT probabilities, detailed characterization of non-inferred variables, comparison with additional Bayesian inference algorithms, validation using large real-world Bayesian Networks, and development of complete hybrid deterministic–probabilistic inference systems. These directions will further clarify the applicability and limitations of SAT-enhanced deterministic inference in probabilistic graphical models.

Author Contributions: Conceptualization, E.E.D.M., B.W.O.B. and J.S.C.B.; methodology, J.S.C.B., B.W.O.B. and E.E.D.M.; software, J.S.C.B.; validation, E.E.D.M.; formal analysis, J.S.C.B.; investigation, J.S.C.B.; data curation, J.S.C.B.; visualization, J.S.C.B.; writing—original draft preparation, J.S.C.B.; writing—review and editing, E.E.D.M.; supervision, E.E.D.M.; project administration, E.E.D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Universidad Técnica Estatal de Quevedo (UTEQ) through the 10th Competitive Call for Scientific and Technological Research Funding (FOCICYT), under project “SAT algorithm propagation to improve approximate inference in Bayesian networks”, grant number PFOC10-30-2024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Bayesian Network benchmark datasets analyzed in this study were originally obtained from the **UCI Probabilistic Inference Benchmarks repository**. To guarantee long-term accessibility and facilitate the reproducibility of the experiments, the benchmark datasets have been archived in the datasets directory of the project’s public GitHub repository: <https://github.com/imjosecorderob/BaySATEnhanced/tree/master/datasets> (accessed on 25 June 2026). The complete implementation of the proposed BaySAT framework, including the source code, CNF transformation routines, Boolean-array inference engine, statistical analysis scripts, and the reproducible experimental workflow, is publicly available at: <https://github.com/imjosecorderob/BaySATEnhanced> (accessed on 25 June 2026). The repository also includes the instrumented implementation used to count estimated computational operations, the execution reports corresponding to the five independent repetitions performed for each benchmark dataset, and the scripts used to generate all tables and figures presented in this manuscript. Additional execution reports are available at: <http://bit.ly/44uba6B> (accessed on 25 June 2026). The datasets employed in this work are publicly available and do not contain personal or sensitive information; their use is consistent with applicable Ecuadorian principles on data protection and higher education research transparency [27,28]. All software, datasets, and experimental artifacts follow FAIR (Findable, Accessible, Interoperable, and Reusable) principles to facilitate independent verification and future research.

Acknowledgments: The authors gratefully acknowledge the Universidad Técnica Estatal de Quevedo (UTEQ) for its institutional support. This research was funded through the 10th Competitive Call for Scientific and Technological Research Funding (FOCICYT) of UTEQ. The APC was funded by the Universidad Técnica Estatal de Quevedo (UTEQ). The authors also acknowledge the developers and maintainers of the open-source software employed in this work, including Python 3.13.1, Django 5.2, pgmpy 1.0.0, NumPy 2.2.5, Pandas 2.2.3, Matplotlib, SciPy 1.15.2, NetworkX 3.4.2, PySAT (python-sat) 1.8.dev16 with the Glucose3 SAT solver interface, PyVis 0.3.2, and scikit-learn 1.6.1. Software development and experimental execution were carried out using PyCharm Professional 2025.2.6.1. These software versions are reported to facilitate the reproducibility of the proposed framework. Finally, the authors embrace the principles of Open Science by making the complete implementation, experimental workflow, and reproducibility materials publicly available.

Conflicts of Interest: The authors declare no conflict of interest. The funding institution had no role in the design of the study; in the collection, analysis, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used throughout this manuscript:

AI	Artificial Intelligence
BN	Bayesian Network
CDCL	Conflict-Driven Clause Learning
CNF	Conjunctive Normal Form
CPT	Conditional Probability Table
CRISP-DM	Cross-Industry Standard Process for Data Mining
DAG	Directed Acyclic Graph
DP	Davis–Putnam
DPLL	Davis–Putnam–Logemann–Loveland
GDPR	General Data Protection Regulation
MAP	Maximum A Posteriori
ML	Machine Learning
NP	Nondeterministic Polynomial Time
POBDS	Partially Observable Dynamic Boolean Systems
SAT	Boolean Satisfiability
SMC	Satisfiability Modulo Counting
SSAT	Stochastic Boolean Satisfiability
UAI	Universal Artificial Intelligence Format

References

1. Ulmer, D.; Hardmeier, C.; Frellsen, J. Prior and Posterior Networks: A Survey on Evidential Deep Learning Methods for Uncertainty Estimation. *arXiv* **2021**. [[CrossRef](#)]
2. Becerra-Salas, M.; Roa, H.N. Evaluating Uber Customers' Perception Through Machine Learning Techniques: A Case Study in Ecuador. In Proceedings of the 2023 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 13–15 December 2023; pp. 771–776. [[CrossRef](#)]
3. Hassall, K.L.; Zawadzka, J.; Milne, A.E.; Corstanje, R.; Harris, J.A.; Dailey, A.G.; Keith, A.M.; Glendining, M.J.; McGrath, S.P.; Todman, L.C.; et al. Putting Numbers to a Metaphor: A Bayesian Belief Network with Which to Infer Soil Quality and Health. *Eur. J. Agron.* **2025**, *168*, 127537. [[CrossRef](#)]
4. Wijnings, P.W.A.; Roa-Villescas, M.; Stuijk, S.; de Vries, B.; Corporaal, H. On the Importance of the Execution Schedule for Bayesian Inference. *ACM Trans. Probabilistic Mach. Learn.* **2024**, *1*, 5. [[CrossRef](#)]
5. Abdelmoumin, G.; Liu, C.; Rawat, D. Understanding the Computational Complexity of Diverse Classes of Turing and Super-Turing Computational Models. In Proceedings of the 2023 International Conference on Computational Science and Computational Intelligence, Las Vegas, NV, USA, 13–15 December 2023; pp. 411–419. [[CrossRef](#)]
6. Chang, W.; Liu, W. SAT-GATv2: A Dynamic Attention-Based Graph Neural Network for Solving Boolean Satisfiability Problem. *Electronics* **2025**, *14*, 423. [[CrossRef](#)]
7. Inuma, Y.; Hara-Azumi, Y. Hardware SAT Solver-Based Area-Efficient Accelerator for Autonomous Driving. In Proceedings of the 21st International Conference on Field-Programmable Technology, Hong Kong, 5–9 December 2022. [[CrossRef](#)]
8. Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*; Morgan Kaufmann Publishers: San Francisco, CA, USA, 1988.
9. Murphy, K.P. *Probabilistic Machine Learning: Advanced Topics*; MIT Press: Cambridge, MA, USA, 2023.
10. Cook, S.A. The Complexity of Theorem-Proving Procedures. In Proceedings of the Third Annual ACM Symposium on Theory of Computing, Shaker Heights, OH, USA, 3–5 May 1971; pp. 151–158.
11. Audemard, G.; Simon, L. Predicting Learnt Clauses Quality in Modern SAT Solvers. In Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09), Pasadena, CA, USA, 11–17 July 2009; pp. 399–404.
12. Marques-Silva, J.P.; Sakallah, K.A. GRASP: A Search Algorithm for Propositional Satisfiability. *IEEE Trans. Comput.* **1999**, *48*, 506–521. [[CrossRef](#)]

13. Díaz, E.; Moral, S. A Deletion Algorithm for the Marginal Problem in Propositional Logic Based on Boolean Arrays. *Mathematics* **2023**, *11*, 2748. [[CrossRef](#)]
14. Sang, T.; Beame, P.; Kautz, H. Performing Bayesian Inference by Weighted Model Counting. In Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05), Pittsburgh, PA, USA, 9–13 July 2005; pp. 475–481.
15. Chavira, M.; Darwiche, A. On Probabilistic Inference by Weighted Model Counting. *Artif. Intell.* **2008**, *172*, 772–799. [[CrossRef](#)]
16. Li, J.; Jiang, N.; Xue, Y. An Exact Solver for Satisfiability Modulo Counting with Probabilistic Circuits. *arXiv* **2025**. [[CrossRef](#)]
17. Audemard, G.; Lagniez, J.M.; Miceli, M. A New Exact Solver for (Weighted) Max#SAT. *Leibniz Int. Proc. Inform.* **2022**, *236*, 28:1–28:18.
18. Peruvemba, V.; Parviainen, P. Scalable Bayesian Network Structure Learning Using SAT-Based Methods. Doctoral Dissertation, Technische Universität Wien, Vienna, Austria, 2023. [[CrossRef](#)]
19. Lu, N.Y.; Zhang, K.; Yuan, C. Improving Causal Discovery by Optimal Bayesian Network Learning. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 8741–8748. [[CrossRef](#)]
20. Miao, X.; Gao, Y.; Guo, S.; Chen, L.; Yin, J.; Li, Q. Answering Skyline Queries over Incomplete Data with Crowdsourcing (Extended Abstract). In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; pp. 2032–2033. [[CrossRef](#)]
21. Lee, N.Z. Stochastic Boolean Satisfiability for Probabilistic Inference. Ph.D. Thesis, University of California, Los Angeles, CA, USA, 2021.
22. Baek, J.; Park, H.J. Bayesian Adaptive Method for Estimating Speed–Accuracy Tradeoff Functions of Multiple Task Conditions. *Behav. Res. Methods* **2024**, *56*, 4403–4420. [[CrossRef](#)] [[PubMed](#)]
23. Alali, M.; Imani, M. Kernel-Based Particle Filtering for Scalable Inference in Partially Observed Boolean Dynamical Systems. *IFAC-PapersOnLine* **2024**, *58*, 1–6. [[CrossRef](#)] [[PubMed](#)]
24. Kiwanuka, F.N.; Karadsheh, L.; Alqatawna, J.; Hudaya, A.; Amin, M. Modeling Employee Flexible Work Scheduling as a Classification Problem. *Procedia Comput. Sci.* **2021**, *192*, 3281–3290. [[CrossRef](#)] [[PubMed](#)]
25. Matricon, T.; Anastacio, M.; Fijalkow, N.; Simon, L.; Hoos, H.H. Statistical Comparison of Algorithm Performance Through Instance Selection. *Leibniz Int. Proc. Inform.* **2021**, *210*, 43:1–43:21. [[CrossRef](#)]
26. Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C.; Wirth, R. *CRISP-DM 1.0: Step-by-Step Data Mining Guide*; SPSS Inc.: Chicago, IL, USA, 1999.
27. Dirección Nacional de Registros Públicos. *Ley Orgánica de Protección de Datos Personales*; Registro Oficial: Quito, Ecuador, 2021.
28. Consejo de Educación Superior. *Ley Orgánica de Educación Superior (LOES)*; Registro Oficial Suplemento 298: Quito, Ecuador, 2018.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.