SAT + NAUTY: Orderly Generation of Small Kochen-Specker Sets Containing the Smallest State-independent Contextuality Set

Zhengyu Li[1] , **Curtis Bright**[2] , **Stefan Trandafir**[3] , **Adán Cabello**[3,4] and **Vijay Ganesh**[1]
[1]Georgia Institute of Technology, Atlanta, USA
[2]University of Waterloo, Canada
[3]Departamento de Física Aplicada II, Universidad de Sevilla, E-41012 Sevilla, Spain
[4]Instituto Carlos I de Física Teórica y Computacional, Universidad de Sevilla, E-41012 Sevilla, Spain
brian.li@gatech.edu, cbright@uwaterloo.ca, strandafir@us.es, adan@us.es vganesh45@gatech.edu,

Abstract

We present a search for small Kochen-Specker (KS) sets in dimension 3, specifically targeting extensions of the 13-ray Yu-Oh set, which has been proven to be the minimal witness to state-independent contextuality. To enable this search, we introduce a novel SAT-based orderly generation framework integrating recursive canonical labeling (RCL) with the graph isomorphism tool NAUTY. We demonstrate that previous SAT approaches relying on lexicographical canonicity suffer from exponential scaling on canonical graphs. This limitation renders them intractable on the large instances (25 to 33 vertices) encountered in our search, whereas our RCL check maintains consistent millisecond-level performance (0.005s), effectively eliminating the bottleneck. Overcoming this bottleneck allows us to perform the first exhaustive enumeration of all KS sets with up to 33 rays containing the complete 25-ray state-independent contextuality (SI-C) set obtained by rigid extensions of the Yu-Oh set in 1,641 CPU hours. We found and verified the 33-ray set discovered by Schütte is the smallest three-dimensional KS set containing the complete 25-ray SI-C set. All non-existence results are backed by independently verifiable proof certificates via an extension of the DRAT proof format.

1 Introduction

A Kochen-Specker (KS) set is a finite set of rays in $\mathbb{C}^d$ for some $d \geq 3$, which does not admit a $\{0,1\}$-assignment of the rays such that (i) each orthogonal basis contains exactly one ray assigned 1, and (ii) in each pair of orthogonal rays, at most one ray is assigned 1.

A state-independent contextuality (SI-C) set is a set of rays $\mathbb{C}^d$ for some $d \geq 3$ for which the corresponding rank-one observables have some non-contextual inequality that is violated by any initial quantum state.

Here we restrict our attention to the smallest possible dimension, 3. It has been proven [Cabello *et al.*, 2016] that the smallest SI-C set contains 13 rays in dimension 3 and consists of all real vectors with coordinates in $\{0, 1, -1\}$ [Yu and Oh, 2012]. In contrast, it is not known what the smallest KS set is in dimension 3: a problem which has been open for just under 60 years since the discovery of the first such set by Kochen and Specker [Kochen and Specker, 1965]. The smallest known is due to Conway and Kochen and has 31 rays. Exhaustive searches have proven that the smallest KS set in dimension 3 must have at least 24 rays (see Kirchweger *et al.* and Li *et al.*).

All KS sets are SI-C sets, but in general KS sets may contain small SI-C sets that are not KS sets. Indeed, each of the small known Kochen-Specker sets in dimension 3 contains a copy of the 13-ray SI-C set. Moreover, the other small 3-dimensional SI-C set in the literature has 21 rays [Bengtsson *et al.*, 2012], and the smallest known KS set containing it has 55 rays [Trandafir and Cabello, 2025].

Recently, using the 13-ray Yu-Oh set as a starting point, a new Kochen-Specker set has been discovered that also has 33 rays, but has only 14 orthogonal bases [Cabello, 2025b]. Given that the history of Kochen-Specker sets is nearly sixty years old [Kochen and Specker, 1967], it is surprising that such a fundamental object remained undiscovered until recently.

It has recently been shown that Kochen-Specker sets correspond exactly to bipartite nonlocal games for which there exists a perfect quantum strategy (BPQS) [Cabello, 2025a]. These are games played between two players, Alice and Bob, who have input sets X and Y (respectively), and output sets A and B (respectively), and for which each choice of inputs and corresponding outputs is either winning or losing. Kochen-Specker sets provide games and accompanying optimal quantum strategies for Alice and Bob, allowing them to win at each round of the game, while no such classical strategy exists [Cinelli *et al.*, 2005; Yang *et al.*, 2005; Aolita *et al.*, 2012; Xu *et al.*, 2022; Kumar *et al.*, 2025]. Small Kochen-Specker sets play a very important role in this setting since they lead to games with low input cardinality $|X||Y|$ (and thus more experimentally feasible scenarios). For example, the aforementioned 33 ray set produces the smallest known input cardinality for dimension 3.

Recently a particular type of SI-C set (or KS set), called a *rigid* SI-C (KS) set has proven to be especially important (in essence the orthogonalities of such a set are unique up to unitary transformations). Xu *et al.* showed that rigid SI-C sets enable certification with any full-rank state (CFR), which simplify preparation of quantum experiments and improve robustness under experimental imperfections. Moreover, such

KS sets provide the only known way to self-test supersinglets of d particles at d levels. Naturally, there has been much interest in identifying small rigid KS sets (see for example [Aravind *et al.*, 2025; Trandafir and Cabello, 2025; Kernaghan, 2026]).

We focus on the construction method where, starting with the Yu-Oh 13-ray SI-C set, we recursively choose pairs of rays and add the unique ray orthogonal to the pair. One may construct the 31-ray and 37-ray [Peres, 1993] sets of Conway and Kochen in this manner, as well as the 33-ray set of Schütte [Bub, 1996; Peres, 1993]. The aplication of this procedure to every pair of the Yu-Oh set yields a rigid set containing 25 rays. This 25-ray core is the natural place to implement a search for the smallest rigid KS set in dimension 3, since such a search will necessarily find any rigid KS set containing the smallest SI-C set.

Our goal in this work is to apply exhaustive search in order to find (rigid) Kochen-Specker sets in dimension 3 obtained from this SI-C set. The main contributions of our work are:

- **A new canonical form that leverages the empirical efficiency of NAUTY [McKay and Piperno, 2014], a tool[1] to compute graph automorphism groups, while preserving the hierarchical property required for SAT-based orderly generation.** Previous SAT-based approaches to orderly generation rely on lexicographical definitions of canonicity, which fail to scale to graphs of large order n. Conversely, while specialized tools like NAUTY are highly efficient at isomorphism checking, they cannot be directly embedded into a SAT solver to perform orderly generation. This is because the standard canonical labeling computed by NAUTY is not *hereditary* (or prefix-preserving): a canonical graph may possess non-canonical subgraphs. Consequently, a solver cannot simply use NAUTY to prune partial assignments, as this would discard valid branches that eventually lead to canonical solutions. We combine the best of both worlds by implementing Recursive Canonical Labeling (RCL), a technique introduced by Afzaly [Afzaly, 2016] that constructs a hierarchical canonical form from any base labeling function. By building RCL on top of NAUTY, we create a **SAT+NAUTY** framework that successfully utilizes NAUTY's empirical efficiency within a valid orderly generation scheme. A SAT + graph isomorphism tool was previously used in the resolution of Lam's problem [Bright *et al.*, 2021] via recording an isomorphism certificate for every graph explored. However, the list of recorded objects can grow large, and to effectively exploit parallelization it needs to be shared across all processors, a limitation that is not present in our approach.
- **Exhaustive enumeration of KS sets extending the complete SI-C set.** We enumerate all Kochen-Specker sets up to 33 rays containing the complete SI-C set of order 25. This set is the unique closure of the original 13-ray Yu-Oh set under the operation of adding orthogonal rays to existing pairs. While general recursive extension from the 13-ray set yields a vast search space, we focus on this dense 25-ray core. Our search confirms that, up to order 33, the 33-ray KS set by Schütte [Bub, 1996; Peres, 1993] is the *only* KS set containing this complete 25-ray structure. The set is illustrated in Figure 1 as a subset of the 37-ray set of Conway and Kochen (which was unpublished, but appears in [Cabello, 1996]). This classification is fundamental to quantum foundations, as the Yu-Oh set constitutes the minimal witness to state-independent contextuality. By exhaustively mapping its rigid extensions, we rigorously define the geometric boundaries of the simplest contextual structures in dimension three.

[1]Open source software available at https://github.com/BrianLi009/SAT-nauty.

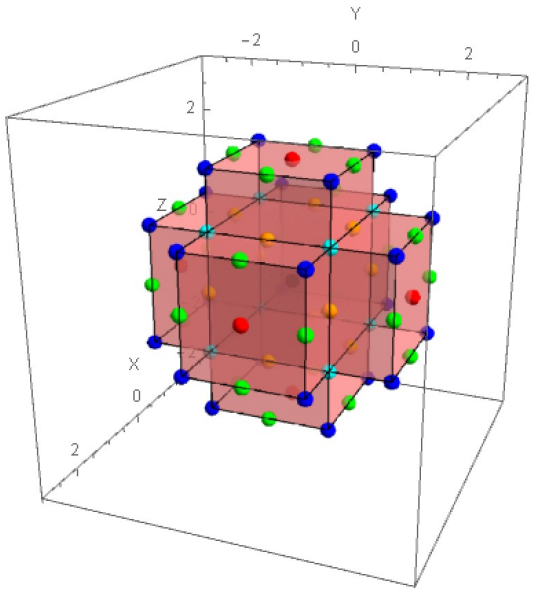

Figure 1: If we connect the center of the figure with all the dots, we get 37 directions (rays) whose components are $(0, 0, 1)$, $(0, 1, \pm 1)$, $(0, 1, \pm 2)$, $(1, 1, \pm 1)$, $(1, \pm 1, \pm 2)$ and their permutations. Schütte-33 is obtained by removing the rays $(0, 1, \pm 2)$ and $(0, 2, \pm 1)$.

2 Related Work

The traditional approach to prevent a SAT solver from repeatedly exploring isomorphic parts of a search space is via the use of *symmetry breaking* techniques. One such symmetry breaking approach is to add "static" constraints prior to the search that reduces the size of the search space [Crawford *et al.*, 1996; Heule, 2019]. Another approach is to "dynamically" break symmetries during the search [Sellmann and Hentenryck, 2005; Metin *et al.*, 2018]. Within the SAT context, two prominent frameworks that use dynamic symmetry breaking in this domain are the SAT+CAS paradigm [Bright *et al.*, 2022] and the SAT Modulo Symmetries (SMS) framework [Kirchweger and Szeider, 2024].

SAT+CAS. Li *et al.* applied the SAT+CAS paradigm to the Kochen-Specker problem, establishing a verified lower bound of 24 for the size of a KS system. This approach combines a SAT solver (MapleSAT [Liang *et al.*, 2016]) with a Computer Algebra System (CAS) to perform isomorph-free

orderly generation. In this framework, canonicity is defined lexicographically based on the concatenation of the above-diagonal entries of the adjacency matrix columns. The solver prunes the search space by blocking partial assignments that are detected to be non-canonical.

SAT Modulo Symmetries (SMS). Kirchweger *et al.* independently established the lower bound of 24 using the SMS framework. The tool integrates symmetry breaking into the CDCL loop of the SAT solver via a custom propagator. Unlike in Li *et al.*'s SAT+CAS approach, SMS defines canonicity based on the lexicographical minimality of the flattened row-wise adjacency matrix. The propagator performs a minimality check on partial assignments to ensure they can be extended to a canonical graph.

Limits of Lex-based Canonicity. Both SAT+CAS and SMS rely on the lexicographical definitions of canonicity. Consequently, neither framework addresses the intrinsic scalability bottleneck associated with these checks. To strictly guarantee that a given graph G is canonical under a lexicographical definition, one must essentially verify that no other permutation of vertices yields a lexicographically smaller adjacency string. In the worst case, this requires testing against the space of all $n!$ permutations, which becomes computationally infeasible as n grows. This shared limitation underscores the advantage of the Recursive Canonical Labeling (RCL) approach, which circumvents the need for such exhaustive permutation checks by leveraging the efficiency of NAUTY while maintaining the hereditary property required for effective pruning.

3 Definitions

A *state-independent contextuality* (SI-C) set in dimension d is a set of rank-one ideal observables (or, equivalently rays) violating a non-contextual inequality for any initial state in dimension d.

A *Kochen-Specker (KS) set* [Kochen and Specker, 1967] is a finite set of rank-one observables $\mathcal{V}$ (or, equivalently, rays) in a Hilbert space $\mathcal{H} = \mathbb{C}^d$ of finite dimension $d \geq 3$, which does not admit an assignment $f\colon \mathcal{V} \to \{0,1\}$ such that $f(u)+f(v) \leq 1$ for $u, v \in \mathcal{V}$ orthogonal, and $\sum_{u\in b} f(u) = 1$ for every orthonormal basis $b \subseteq \mathcal{V}$.

Given a set K of vectors in $\mathbb{C}^d$, the *orthogonality graph* of K is the graph whose vertices are the vectors of K and for which two vertices are adjacent if and only if their corresponding vectors are orthogonal.

A graph G is 010-*colorable* if there is a $\{0,1\}$-coloring of the vertices such that the following two conditions are satisfied simultaneously:

a. Adjacent vertices are not both colored 1. b. For each triangle in G, there is exactly one vertex that is colored 1.

A set of vectors does not form a Kochen-Specker set if its orthogonality graph is 010-colorable. On the other hand, given a graph that is non-010-colorable, it is not immediate that there is an associated Kochen-Specker set. This is only the case if there is a suitable set of vectors whose orthogonalities are captured by that graph (this is called a *faithful orthogonal representation* of the graph G; for brevity, we refer to a faithful orthogonal representation simply as an orthogonal representation).

4 Methodology

In this work, we search for Kochen-Specker (KS) sets in dimension 3 with at most 33 rays containing the Yu-Oh 13-ray SI-C set. We focus on a specific type of subgraph constructions relevant to quantum foundations, followed by a SAT-based encoding to exhaustively enumerate valid KS sets.

4.1 Recursive Extension from the Complete SI-C Set.

The approach builds upon the observation that the 13-ray Yu-Oh set can be naturally extended by adding the unique ray orthogonal to any pair of existing non-parallel rays. The ray orthogonal to two rays is given by the cross product of those two rays. Though the cross product vector itself is scale-dependent, the ray it spans is uniquely defined because we care only about the direction of the vector, not its length. Repeatedly applying this operation to closure yields a unique set of 25 rays, which we term the complete SI-C set. Famous examples of KS sets, such as the 33-ray set of Schütte, are known to contain this specific 25-ray substructure. In this work, we specifically target this class of systems. We fix the 25-ray complete set as a base subgraph and employ our SAT solver to exhaustively enumerate all Kochen-Specker extensions up to order 33. This narrows the search space to the most structured candidates while ensuring we find any variant of the Schütte set that might exist.

4.2 SAT Encoding

To perform the search, we encode the properties of a KS graph into a Boolean Satisfiability (SAT) instance. We employ the encoding detailed in the previous SAT+CAS framework [Li *et al.*, 2024], which formulates the encoding via Boolean variables $e_{i,j}$ (representing vertices i and j are adjacent, i.e., the rays i and j are orthogonal) and $t_{i,j,k}$ (representing rays i, j, k form a mutually orthogonal triple).

We enforce the following constraints to prune the search space effectively:

1. **Structural Constraints:** We leverage properties from graph theory that a minimal KS set must satisfy, specifically:
   - **Squarefree:** The graph must not contain any 4-cycles (C_4).
   - **Minimum Degree:** Every vertex must have a degree of at least 3.
   - **Triangle Property:** Every vertex must be part of at least one triangle (a mutually orthogonal triple).
2. **Non-010-Colorability:** The core KS property requires that the graph admits no 010-coloring. This is encoded by generating blocking clauses for all valid 010-colorings. Following the optimization in [Li *et al.*, 2024], we only generate clauses for colorings where the set of vertices V_1 assigned the value 1 is small (specifically, $|V_1| < \lceil n/2 \rceil$). We prioritize blocking these cases because colorings with a large number of 1s (large $|V_1|$) are unlikely to be 010-colorable, rendering explicit blocking clauses for them unnecessary.

3. **Fixed Subgraphs:** We first compute the canonical labeling of the starting configuration (of order $n = 25$) and fix the corresponding edge variables via unit clauses. This initialization step highlights a critical scalability advantage over lexicographical approaches (like SAT+CAS): merely computing the canonical form of these large base subgraphs is difficult under a pure lexicographical definition, which would prevent such solvers from effectively initializing the search space.

Any satisfying assignment to this formula corresponds to a valid KS candidate graph. The geometric realizability of these candidates is then verified using the orthogonality check described in Section 6.

5 SAT + NAUTY-based Orderly Generation

A key component of our pipeline is the integration of a SAT solver with an isomorph-free orderly generation routine. The orderly generation approach (developed in 1978 in two independent publications [Read, 1978; Faradžev, 1978]) allows for the exhaustive enumeration of non-isomorphic graphs without the need to store previously generated isomorphism classes. This technique is the foundation for efficient combinatorial generators such as GENREG [Meringer, 1999], which prune the search tree by enforcing that every intermediate graph must be the canonical representative of its isomorphism class.

Classical orderly generation relies on a specific definition of canonicity that satisfies the *hereditary property*: if a graph is canonical, then its parent (typically the induced subgraph obtained by removing the last vertex) must also be canonical. This property implies the following:

> *If a partial graph (prefix) is non-canonical, no extension of that graph can ever become canonical.*

This allows the search to prune non-canonical intermediates immediately. However, highly efficient canonical labeling tools like NAUTY [McKay, 2007] produce labelings that are not hereditary, making them unsuitable for direct use in orderly generation.

To resolve this, we implement *Recursive Canonical Labeling* (RCL), a technique introduced by Afzaly [Afzaly, 2016]. RCL acts as a wrapper that transforms *any* base canonical labeling function δ into a hierarchical (hereditary) canonical labeling. In our implementation, we use NAUTY as the base function δ due to its efficiency, but the framework is agnostic to this choice.

5.1 Recursive Canonical Labeling

We first formally define the hierarchical property required for our search, and then describe the Recursive Canonical Labeling (RCL) algorithm used to satisfy it. Let G be a graph on n vertices labeled $\{1, \ldots, n\}$. We denote $G[k]$ as the induced subgraph of G on the vertices $\{1, \ldots, k\}$.

Definition 1 (Hierarchical Canonical Labeling)**.** *A canonical labeling function C is* hierarchical *if for any graph G that is canonical under C (i.e., $C(G) = G$), and for any $1 \leq k < n$, the prefix $G[k]$ is also canonical under C.*

RCL constructs such a labeling recursively. We start with an arbitrary *base canonizer* δ—defined as any function that computes a canonical labeling for a graph (i.e., ensuring that if $G \cong H$, their labeled forms are identical), regardless of whether it satisfies the hierarchical property. In our implementation, we use NAUTY as this base canonizer due to its efficiency.

Given a graph G and the base canonizer δ, the Recursive Canonical Labeling is computed as follows:

1. Compute the base labeling $\pi = \delta(G)$ using the underlying tool.
2. Identify the vertex v mapped to the largest label n by π (i.e., $v = \pi^{-1}(n)$).
3. Permanently assign v the label n in the hierarchical labeling.
4. Remove v and recursively apply the procedure to the subgraph $G \setminus \{v\}$ to determine the labels $\{1, \ldots, n-1\}$.

Algorithm 1: SAT + NAUTY Orderly Generation. The SOLVE routine initializes the solver and registers the symmetry check. CHECKCANONICITY takes in partial assignments to verify that the induced subgraph G_k is canonical. If G_k is not canonical, a blocking clause is added to the solver immediately, forcing it to backtrack. RECURSIVECANONICAL recursively constructs the hereditary canonical form.

```
INPUT  n: number of vertices; Φ: problem constraints (CNF)
OUTPUT All canonical graphs satisfying Φ
FUNCTION Solve(Φ, n):
    Initialize SAT solver with formula Φ
    Register CheckCanonicity as callback
    WHILE SAT solver finds satisfying assignment σ DO
        Output G_σ
        Add blocking clause ¬σ
    END
FUNCTION CheckCanonicity(partial assignment π):
    k ← GetCompletePrefix(π)
    IF k ≥ 2 and k increased since last check THEN
        G_k ← subgraph induced by {1, ..., k} under π
        (G_k^can, γ) ← RecursiveCanonical(G_k)
        IF G_k^can ≠ G_k THEN
            C ← clause blocking assignment of G_k
            AddBlockingClause(C, witness γ)
        END
    END
FUNCTION RecursiveCanonical(G_k):
    IF k = 1 THEN
        RETURN (G_k, identity)
    END
    δ ← Nauty(G_k)
    v ← vertex with δ(v) = k
    G_{k-1} ← G_k[V(G_k) \ {v}]
    (G_{k-1}^can, γ') ← RecursiveCanonical(G_{k-1})
    Extend γ' to γ by setting γ(v) = k
    RETURN (G_k^γ, γ)
```

By recursively anchoring the largest label according to the base canonizer, we create a new labeling that is guaranteed to be hierarchical by construction, since the canonical form of the n-vertex graph is defined by extending the canonical form of its $(n-1)$-vertex subgraph.

5.2 Integration with SAT Solving (CDCL)

We integrate this canonicity check directly into the Conflict-Driven Clause Learning (CDCL) loop of the CaDiCaL [Biere *et al.*, 2024] SAT solver via a custom propagator [Fazekas *et al.*, 2023]. The problem is encoded using Boolean variables $e_{i,j}$ representing the edges.

The Propagator Logic. The SAT solver explores the search space by assigning truth values to edge variables. Although the solver may assign variables in an arbitrary order, our propagator monitors the assignment to detect when a *complete prefix* is formed. A prefix $G[k]$ is considered complete when all $\binom{k}{2}$ edge variables $e_{i,j}$ with $1 \le i < j \le k$ are assigned. Whenever the solver completes a new prefix $G[k]$, the propagator invokes the RCL check:

- If $G[k]$ is canonical, the search continues.
- If $G[k]$ is **non-canonical**, the propagator adds a blocking clause back to the solver.

Blocking Non-Canonical Partials. When a prefix $G[k]$ is found to be non-canonical, we generate a *blocking clause* to prune the search branch. Because RCL is hierarchical, the non-canonicity of $G[k]$ implies that no extension to a larger graph $G[n]$ can be canonical.

The blocking clause C_{block} forbids the specific assignment of edges on the subgraph induced by vertices $\{1, \dots, k\}$. Let π be the current partial assignment. For each pair of vertices $1 \le i < j \le k$, we define the literal $\lambda_{i,j}$ representing the negation of the current value of edge variable $e_{i,j}$:

$$\lambda_{i,j} = \begin{cases} \neg e_{i,j} & \text{if } e_{i,j} \text{ is assigned TRUE in } \pi, \\ e_{i,j} & \text{if } e_{i,j} \text{ is assigned FALSE in } \pi. \end{cases}$$

The blocking clause is then the disjunction of these negated literals:

$$C_{\text{block}} = \bigvee_{1 \le i < j \le k} \lambda_{i,j}.$$

This clause is added to the solver's database, forcing it to backtrack and modify at least one edge decision within the subgraph $G[k]$.

6 Orthogonality Check

Beyond isomorph-rejection, our search incorporates a custom propagator that exploits the specific geometric nature of the Kochen-Specker problem. Our problem domain provides partial geometric information—specifically, the coordinates of the base set—before the search begins. We leverage this domain knowledge to detect and prune combinatorially valid graphs that are geometrically unrealizable in $\mathbb{C}^3$.

6.1 Deriving Vector Coordinates

The search operates by extending a *fixed* subgraph (such as the 25-ray complete SI-C set) to a larger target order (e.g., 33 rays). This creates a distinction between two types of vectors in the system:

1. **Fixed Vectors (Static):** The vectors corresponding to the base subgraph are pre-calculated and fixed. Their coordinates satisfy all orthogonality constraints defined by the subgraph's edges.
2. **Search Vectors (Dynamic):** For the remaining vertices (e.g., rays 26 to 33), the coordinates are unknown at the start of the search. They are determined dynamically by the SAT solver's choices.

As the SAT solver assigns truth values to edge variables connecting a new vertex v to existing vertices u_1 and u_2, the propagator infers the geometric consequence. If v is asserted to be orthogonal to both u_1 and u_2 (where $\vec{u}_1, \vec{u}_2$ are already known), then $\vec{v}$ is uniquely determined (up to a scalar factor) by the cross product: $\vec{v} = \vec{u}_1 \times \vec{u}_2$. This allows the propagator to constructively determine the coordinates of new rays on-the-fly. If a vertex is connected to only one known vector, its coordinates remain undetermined; once a second orthogonal neighbor is assigned, the coordinates become fixed, triggering further checks.

6.2 Detecting and Blocking Geometric Violations

A geometric contradiction arises when dynamically derived vectors fail to satisfy the required orthogonality relations. We identify two conflict types: (1) **Generating Parallel Vectors:** A newly derived vector cannot be parallel to existing vectors. (2) **Violating Orthogonality:** If a derived vector $\vec{v}$ is adjacent to $\vec{w}$ but $\langle \vec{v}, \vec{w} \rangle \neq 0$, the edge constraint is violated.

Isolating the Cause via Dependency Chains. When a violation is detected, a naive approach would be to block the entire partial assignment. However, this is inefficient as it does not isolate the specific set of decisions that caused the geometric failure. Instead, we compute a *dependency chain* for every derived vector to construct a minimal blocking clause, which only contains vectors that are involved in causing such violation.

Let $D(v)$ be the set of edge variables responsible for determining the coordinates of vector $\vec{v}$. For a fixed vertex, $D(v) = \emptyset$. For a derived vertex v computed from parents u_1, u_2, the dependency set is defined recursively as $D(v) = \{e_{v,u_1}, e_{v,u_2}\} \cup D(u_1) \cup D(u_2)$. When a conflict occurs (e.g., $\vec{v} \cdot \vec{w} \neq 0$ despite edge variable $e_{v,w}$ being true), we learn the clause equivalent to the constraint

$$C_{\text{block}} = \neg \left(e_{v,w} \wedge \bigwedge_{e \in D(v) \cup D(w)} e \right).$$

This clause blocks only the specific chain of edge choices that forced the creation of the incompatible vectors, allowing the solver to retain the valid parts of the partial graph.

6.3 Exact Arithmetic

To ensure correctness, we perform all geometric computations using exact arithmetic, avoiding floating-point instability. The

arithmetic field is automatically selected based on the input: we use $\mathbb{Z}^3$ for real-valued sets (verifying orthogonality via $\vec{a} \cdot \vec{b} = 0$) and the appropriate algebraic field for complex-valued candidates (using the Hermitian inner product $\langle \vec{a}, \vec{b} \rangle = 0$). Since derived vectors are generated exclusively via cross products, the field remains closed under the operations of the base set.

7 Verification

To ensure the correctness of our exhaustive search, we produce independently verifiable proofs. Our pipeline extends the standard DRAT proof format to support two types of domain-specific axioms: canonicity clauses (t-clauses) and orthogonality clauses (o-clauses). External clauses are labeled as trusted during the standard RUP (Reverse Unit Propagation) check and are independently validated by a simple specialized checker.

7.1 Proof Format and Trust Boundary

The proof consists of a sequence of clause additions and deletions. Standard learned clauses are verified via RUP, ensuring they are logical consequences of the current formula. If a standard RUP clause blocks the canonical form, then the problem constraints themselves forbid that specific graph.

The correctness of the external clauses relies on one key principle: a t-clause must never block the unique canonical representative of an isomorphism class. The verifier validates this criteria using a trusted core of graph permutation and exact arithmetic routines (implemented in a lightweight C++ verifier), avoiding dependency on the SAT solver or a Computer Algebra System.

7.2 Canonicity Clause Verification

For each t-clause C blocking a graph G, the verifier must ensure that this blocking step is sound — that is, that at least one graph isomorphic to G remains reachable by the search. The witness π provided in the proof exhibits such a survivor: applying π to G yields an isomorphic graph $G' = \pi(G)$, and the verifier confirms that the clause encoding G' and all of its prefixes are not among the blocking clauses. Since permutation preserves the isomorphism class, $G' \cong G$; since G' is unblocked, the iso-class is not eliminated. No lexicographic or ordering relation between G and G' is asserted or checked—in practice π is supplied by nauty's recursive canonical labeling, so G' is the canonical representative, but the verifier requires only that G' be some unblocked isomorphic copy.

The verification logic relies on a global consistency check against a hash table $\mathcal{S}$, which stores all t-clauses asserted so far. To validate a clause C with witness π, the verifier performs the following steps:

1. **Apply Permutation:** Construct the adjacency matrix M' of the isomorphic graph G' using the witness π (i.e., $M' = \pi(A_G)$).
2. **Check Recursive Validity:** Verify that neither the graph defined by M', nor any of its generating ancestors (prefixes), appears in the blocked set $\mathcal{S}$.

This check ensures that while G is pruned, its canonical alternative G' (and the path leading to it) remains valid for exploration. Because the maximum graph order is $N = 33$ and the search starts from a base of size $p \approx 13$, checking the ancestors of M' requires fewer than 20 lookups in $\mathcal{S}$ per clause. This limited recursion depth ensures that the verification overhead remains low relative to the solver's search time.

7.3 Orthogonality Clause Verification

For o-clauses, the verifier must confirm that the blocked subgraph contains a geometric contradiction. The witness includes the minimal set of edges $E_{\text{wit}} \subseteq E$ required to derive the contradiction.

The verification proceeds in two steps:

1. **Re-derivation:** The verifier does not trust the vector coordinates provided in the witness. Instead, starting from the fixed base vectors, it *re-computes* the coordinates for the vertices in the witness using the edges in E_{wit} and the cross-product rules. This ensures the vectors are strict algebraic consequences of the graph structure.
2. **Contradiction Check:** Using the re-derived vectors, the verifier checks for one of two fundamental violations mentioned in Section 6.2.

This pipeline ensures that: (1) t-clauses never block at least one representative of every graph arising during the enumeration process (soundness of isomorph rejection), and (2) o-clauses never block a geometrically valid graph (soundness of pruning). Combined with the standard RUP check, this provides a complete certificate that the search space was exhaustively explored.

8 Results

We present our results in two parts: a performance validation of the **SAT+NAUTY** framework followed by the exhaustive enumeration of Kochen-Specker sets. We begin with a comparison study demonstrating that standard lexicographical symmetry breaking is intractable on the canonical partial assignments dominating this search space. A direct end-to-end comparison is effectively impossible: as noted in Section 4.2 (Fixed Subgraphs), the lexicographical approach cannot practically canonize even the starting base graph required to initialize the search. The quantitative comparison is reported in Table 1 and visualized in Figures 2.

8.1 Performance Analysis: RCL vs. Lex-based

We next isolate the core technical bottleneck addressed by our approach: checking canonicity on the highly structured graphs arising in Kochen-Specker (KS) generation. As discussed in Section 4.2, lexicographical symmetry breaking is attractive because it yields a hereditary (prefix-preserving) notion of canonicity, which enables pruning of partial assignments. However, in KS-style searches the solver spends most of its time near canonical (and near-canonical) partial graphs, precisely where lex-least checks exhibit their worst-case behavior. In contrast, our Recursive Canonical Labeling (RCL) construction provides a hereditary canonical form while inheriting the practical efficiency of NAUTY as its base labeling routine.

Table 1: Runtime comparison on 100 lex-least canonical 4-chromatic K_4-free graphs per order.

Order	N	Lex Avg	RCL Avg	Lex Total	RCL Total	Speedup
15	100	31.9 ms	26.6 ms	3.19 s	2.66 s	1×
16	100	34.6 ms	26.9 ms	3.46 s	2.69 s	1×
17	100	56.5 ms	27.0 ms	5.65 s	2.70 s	2×
18	100	82.4 ms	26.7 ms	8.24 s	2.67 s	3×
19	100	304.2 ms	26.5 ms	30.42 s	2.65 s	11×
20	100	542.7 ms	27.1 ms	54.27 s	2.71 s	20×
21	100	2.51 s	36.7 ms	4.2 min	3.67 s	68×
22	100	6.75 s	33.7 ms	11.3 min	3.37 s	200×
23	100	46.74 s	29.1 ms	1.3 hr	2.91 s	1,606×
24	100	51.22 s	27.9 ms	1.4 hr	2.79 s	1,836×
25	100	3.2 min	27.0 ms	5.3 hr	2.70 s	7,094×
26	100	3.8 min	26.9 ms	6.3 hr	2.69 s	8,499×

Dataset construction: canonical 4-chromatic K_4-free graphs. To obtain an apples-to-apples comparison on graphs that mimic the structure of KS orthogonality graphs, we generated a benchmark set of graphs constrained to satisfy (i) connectedness, (ii) the presence of a triangle, (iii) K_4-freeness (clique number $\omega(G) = 3$), (iv) not 3-colorable ($\chi(G) > 3$), and (v) 4-colorable ($\chi(G) \leq 4$). While these benchmark graphs are not KS orthogonality graphs, they share similar constraints and structural features (e.g., triangle-rich, K_4-free, and highly structured near-canonical instances), making them a useful proxy for stressing canonicity checks in KS-style generation. For each order $n \in \{15, \ldots, 26\}$ we produced 100 such graphs (for a total of 1,200 instances).

We then transformed each instance into a *lex-least canonical* graph using the standard witness-guided procedure: repeatedly run the lex-least canonicity checker; if it returns a witness permutation exhibiting an isomorphic but lexicographically smaller labeling, relabel the graph accordingly and iterate until no improving witness exists. In practice, this canonization process becomes prohibitively expensive beyond this range; we stop at $n = 26$ because many graphs of order 27 required hours (or timed out) to reach the lex-least canonical representative.

Experimental comparison. On this dataset of lex-least canonical graphs, we measured the runtime of (a) the lex-least canonicity check itself and (b) our RCL canonicity check (with NAUTY as the base labeling). Figure 2 summarizes the results. The lex-least check scales exponentially with n: its mean runtime increases from roughly 32ms at $n = 15$ to about 3.8 minutes at $n = 26$ (approximately a 7,000× increase). By contrast, RCL remains essentially flat at around 27ms throughout, i.e., effectively independent of graph size in this range.

The performance gap accelerates rapidly after $n \approx 21$. At $n = 20$ the speedup is a modest $\sim 20\times$, but it jumps to $\sim 200\times$ at $n = 22$, 1,606× at $n = 23$, and reaches 8,499× at $n = 26$. Moreover, RCL's total time across all 100 graphs at a fixed order (about 2.7s) is less than what the lex-least check spends on a *single* canonical graph at $n = 26$ (about 3.8 minutes). These measurements explain why lexicographical symmetry breaking becomes the dominant bottleneck in our target instances, and why replacing the lex-least checker with RCL is necessary to make the exhaustive enumeration in Section 8.2 feasible.

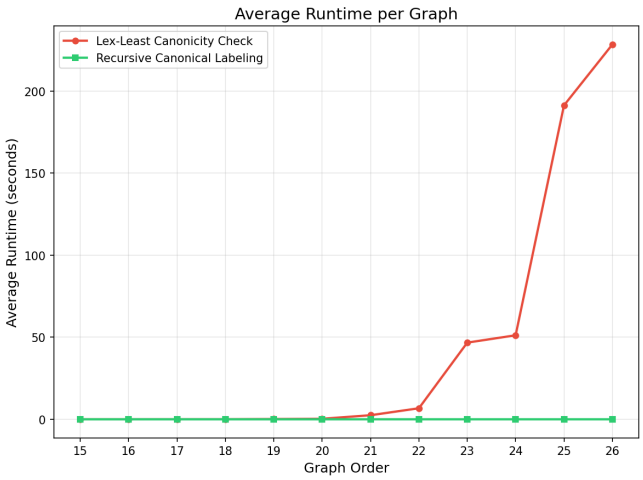

Figure 2: Average canonicity-check runtime on the benchmark dataset (same data as Table 1).

8.2 Exhaustive Enumeration of KS Sets Extending the Complete SI-C Set

Using our RCL-based solver, we performed an exhaustive search for Kochen-Specker sets extending the 25-ray complete SI-C set. The search space was partitioned into 128 independent sub-problems using AlphaMapleSAT [Jha *et al.*, 2024] and executed on a high-performance cluster of Dual AMD EPYC 7713 processors.

The enumeration for order 33 consumed 1,641 hours of CPU time (approximately 68 days), which was completed in roughly 2 days of wall-clock time across 128 cores. The solver identified 44 non-isomorphic candidates that satisfy the combinatorial Kochen-Specker conditions. However, the embeddability check post-processing step confirmed that only **one** of these candidates admits an orthogonal representation in either $\mathbb{R}^3$ or $\mathbb{C}^3$. This unique solution is isomorphic to the 33-ray set discovered by Schütte [Bub, 1996]; thus, we formally confirm that, up to order 33, the Schütte set is the *only* Kochen-Specker set containing the minimal SI-C set of order 25.

To ensure the correctness of this uniqueness result, the entire search process was formally certified. The solver produced a comprehensive proof trace in an extended DRAT format, recording all learned clauses alongside domain-specific axioms for isomorph-rejection (t-clauses) and geometric pruning (o-clauses). For the order-33 enumeration, the total proof size was approximately 13 TiB. This certificate was independently validated by a custom verifier implemented as a standalone Python script. While the verifier code is not formally proven, it is designed to be minimal and auditable, reducing the trusted computing base to a simple set of logical checks that provide a rigorous guarantee that no valid Kochen-Specker sets were overlooked.

9 Conclusion

We presented a SAT-based orderly generation framework that integrates NAUTY to overcome the scalability limitations of lexicographical symmetry breaking. Our analysis confirms that while standard lexicographical checks suffice for random graphs, they incur exponential overhead on the highly structured subgraphs inherent to combinatorial search problems, resulting in a performance gap of nearly four orders of magnitude compared to RCL. Because the search for a canonical solution inevitably requires the verification of canonical and near-canonical partial graphs, standard lexicographical checkers are forced into their worst-case performance. By adopting Recursive Canonical Labeling (RCL), we achieved negligible isomorphism overhead, enabling the first exhaustive enumeration of Kochen-Specker sets extending the 25-ray SI-C closure up to order 33. This framework allowed us to settle the uniqueness of the Schütte set relative to the complete 25-ray core. The SAT+NAUTY framework offers a practical solution for combinatorial generation problems where symmetry handling is the primary bottleneck and can generalize to other domains requiring isomorph-free generation of complex combinatorial structures.

References

[Afzaly, 2016] Seyedeh Narjess Afzaly. *Generation of graph classes with efficient isomorph rejection*. PhD thesis, Australian National University, 2016.

[Aolita *et al.*, 2012] Leandro Aolita, Rodrigo Gallego, Antonio Acín, Andrea Chiuri, Giuseppe Vallone, Paolo Mataloni, and Adán Cabello. Fully nonlocal quantum correlations. *Phys. Rev. A*, 85:032107, 2012.

[Aravind *et al.*, 2025] Padmanabhan K Aravind, Justin YJ Burton, David Richter, and Guillermo Nuñez Ponasso. Triacontagonal proofs of the Bell-Kochen-Specker theorem. *Journal of Physics A: Mathematical and Theoretical*, 2025.

[Bengtsson *et al.*, 2012] I. Bengtsson, K. Blanchfield, and A. Cabello. A Kochen–Specker inequality from a SIC. *Phys. Lett. A*, 376:374–376, 2012.

[Biere *et al.*, 2024] Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt. Cadical 2.0. In *International Conference on Computer Aided Verification*, pages 133–152. Springer, 2024.

[Bright *et al.*, 2021] Curtis Bright, Kevin K. H. Cheung, Brett Stevens, Ilias Kotsireas, and Vijay Ganesh. A SAT-based resolution of Lam's problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):3669–3676, May 2021.

[Bright *et al.*, 2022] Curtis Bright, Ilias Kotsireas, and Vijay Ganesh. When satisfiability solving meets symbolic computation. *Communications of the ACM*, 65(7):64–72, 2022.

[Bub, 1996] Jeffrey Bub. Schütte's tautology and the Kochen-Specker theorem. *Found. Phys.*, 26:787–806, 1996.

[Cabello *et al.*, 2016] A. Cabello, M. Kleinmann, and J. R. Portillo. Quantum state-independent contextuality requires 13 rays. *J. Phys. A: Math. Theor.*, 49:38LT01, 2016.

[Cabello, 1996] Adán Cabello. *Pruebas algebraicas de imposibilidad de variables ocultas en mecánica cuántica*. PhD thesis, Universidad Complutense de Madrid, 1996.

[Cabello, 2025a] Adán Cabello. Simplest bipartite perfect quantum strategies. *Phys. Rev. Lett.*, 134:010201, Jan 2025.

[Cabello, 2025b] Adán Cabello. The simplest Kochen-Specker set. *Phys. Rev. Lett.*, 135:190203, 2025.

[Cinelli *et al.*, 2005] C. Cinelli, M. Barbieri, R. Perris, P. Mataloni, and F. De Martini. All-Versus-Nothing Nonlocality Test of Quantum Mechanics by Two-Photon Hyperentanglement. *Phys. Rev. Lett.*, 95:240405, 2005.

[Crawford *et al.*, 1996] James M. Crawford, Matthew L. Ginsberg, Eugene M. Luks, and Amitabha Roy. Symmetry-breaking predicates for search problems. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning*, KR'96, page 148–159, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[Faradžev, 1978] I A Faradžev. Constructive enumeration of combinatorial objects. In *Problèmes combinatoires et théorie des graphes*, pages 131–135, 1978.

[Fazekas *et al.*, 2023] Katalin Fazekas, Aina Niemetz, Mathias Preiner, Markus Kirchweger, Stefan Szeider, and Armin Biere. IPASIR-UP: User propagators for CDCL. In Meena Mahajan and Friedrich Slivovsky, editors, *26th International Conference on Theory and Applications of Satisfiability Testing (SAT 2023)*, volume 271 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

[Heule, 2019] Marijn J. H. Heule. Optimal symmetry breaking for graph problems. *Mathematics in Computer Science*, 13(4):533–548, May 2019.

[Jha *et al.*, 2024] Piyush Jha, Zhengyu Li, Zhengyang Lu, Curtis Bright, and Vijay Ganesh. AlphaMapleSAT: An MCTS-based cube-and-conquer SAT solver for hard combinatorial problems, 2024.

[Kernaghan, 2026] Michael Kernaghan. The algebraic landscape of Kochen-Specker sets in dimension three. *arXiv preprint arXiv:2603.16988*, 2026.

[Kirchweger and Szeider, 2024] Markus Kirchweger and Stefan Szeider. SAT Modulo Symmetries for graph generation and enumeration. *ACM Trans. Comput. Log.*, 25(3), 2024.

[Kirchweger *et al.*, 2023] Markus Kirchweger, Tomáš Peitl, and Stefan Szeider. Co-certificate learning with SAT modulo symmetries. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 1944–1953. International Joint Conferences on Artificial Intelligence Organization, 2023.

[Kochen and Specker, 1965] S. Kochen and E. P. Specker. Logical structures arising in quantum theory. In J. W. Addison, L. Henkin, and A. Tarski, editors, *Symposium on the Theory of Models: Proceedings of the 1963 International Symposium at Berkeley*, pages 177–189. North-Holland, Amsterdam, 1965.

[Kochen and Specker, 1967] Simon Kochen and Ernst P. Specker. The Problem of Hidden Variables in Quantum Mechanics. *J. Math. Mech.*, 17:59–87, 1967.

[Kumar *et al.*, 2025] Shashwat Kumar, Eliott Rosenberg, et al. Quantum-classical separation in bounded-resource tasks arising from measurement contextuality, 2025. arXiv:2512.02284.

[Li *et al.*, 2024] Zhengyu Li, Curtis Bright, and Vijay Ganesh. A SAT solver + computer algebra attack on the minimum Kochen–Specker problem. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 1898–1906. International Joint Conferences on Artificial Intelligence Organization, 2024.

[Liang *et al.*, 2016] Jia Hui Liang, Vijay Ganesh, Pascal Poupart, and Krzysztof Czarnecki. Learning rate based branching heuristic for SAT solvers. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 123–140. Springer, 2016.

[McKay and Piperno, 2014] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60:94–112, 2014.

[McKay, 2007] Brendan D McKay. Nauty user's guide (version 2.4). *Computer Science Dept., Australian National University*, pages 225–239, 2007.

[Meringer, 1999] Markus Meringer. Fast generation of regular graphs and construction of cages. *Journal of Graph Theory*, 30(2):137–146, 1999.

[Metin *et al.*, 2018] Hakan Metin, Souheib Baarir, Maximilien Colange, and Fabrice Kordon. CDCLSym: Introducing effective symmetry breaking in SAT solving. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 99–114, New York, 2018. Springer International Publishing.

[Peres, 1993] Asher Peres. *Quantum Theory: Concepts and Methods*. Kluwer, Dordrecht, 1993.

[Read, 1978] Ronald C Read. Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations. *Annals of Discrete Mathematics*, 2:107–120, 1978.

[Sellmann and Hentenryck, 2005] Meinolf Sellmann and Pascal Van Hentenryck. Structural symmetry breaking. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05*, pages 298–303, California, 2005.

[Trandafir and Cabello, 2025] Stefan Trandafir and Adán Cabello. Two fundamental solutions to the rigid Kochen-Specker set problem and the solution to the minimal Kochen-Specker set problem under one assumption. *Phys. Rev. A*, 111:052204, May 2025.

[Xu *et al.*, 2022] Jia-Min Xu, Yi-Zheng Zhen, Yu-Xiang Yang, Zi-Mo Cheng, Zhi-Cheng Ren, Kai Chen, Xi-Lin Wang, and Hui-Tian Wang. Experimental Demonstration of Quantum Pseudotelepathy. *Phys. Rev. Lett.*, 129:050402, Jul 2022.

[Xu *et al.*, 2024] Zhen-Peng Xu, Debashis Saha, Kishor Bharti, and Adán Cabello. Certifying sets of quantum observables with any full-rank state. *Phys. Rev. Lett.*, 132:140201, Apr 2024.

[Yang *et al.*, 2005] Tao Yang, Qiang Zhang, Jun Zhang, Juan Yin, Zhi Zhao, Marek Żukowski, Zeng-Bing Chen, and Jian-Wei Pan. All-Versus-Nothing Violation of Local Realism by Two-Photon, Four-Dimensional Entanglement. *Phys. Rev. Lett.*, 95:240406, 2005.

[Yu and Oh, 2012] Sixia Yu and C. H. Oh. State-independent proof of Kochen-Specker theorem with 13 rays. *Phys. Rev. Lett.*, 108:030402, 2012.