WORKING NOTE – PROJECTIVE COVER SATISFIABILITY

THOMAS SINCLAIR

Abstract. We introduce a clause-level, projection-valued semantics for Boolean satisfiability. A Boolean variable x is assigned a projection P_x, negation is interpreted by $P_{\neg x} = I - P_x$, and a clause C is satisfied when its literal projections cover the identity in the Loewner order:

$$\sum_{\ell \in C} P_\ell \geq I.$$

This replaces Boolean OR by an operator lower-cover condition, rather than by a partition-of-unity relation. Commuting projective assignments are exactly classical satisfying assignments, so every genuinely new solution is forced by noncommutativity. We record the first pieces of the resulting calculus: projective 2-SAT collapses to classical 2-SAT; resolution is sound; binary padding is sound; contraction and the usual clause-splitting reduction can fail; and the reduction from longer clauses to ternary clauses becomes a projection interpolation problem. In the opposite direction, the ordinary Boolean gadgets for meet and join force the relevant projections to commute, and locally commuting BCS contexts can be compiled into PSAT by introducing spectral selector variables. Thus local commutativity can be imposed syntactically rather than built into the semantics. The standard CNF encoding of graph q-coloring lifts exactly to projective graph q-coloring, while complete k-CNF contradictions are projectively satisfiable for every $k \geq 4$ by Clifford observables. Finally, we include a reproducible exact rational certificate showing that two-dimensional projective 3-SAT already differs from classical 3-SAT.

Note to the Reader

This is a short working note introducing a projection-valued cover semantics for CNF satisfiability. Boolean variables are assigned projections, negation is interpreted by complementation, and clauses are satisfied by a Loewner lower-cover condition. The note records some elementary structure: projective 2-SAT collapses to ordinary 2-SAT, resolution remains sound, standard clause-splitting can fail, local commutativity can be forced syntactically, graph coloring embeds naturally, and projective 3-SAT already differs from classical 3-SAT via an exact qubit certificate.

Comments are welcome.

1. Introduction

Boolean satisfiability is the canonical NP-complete search problem [1, 2]; see [3] for a modern account of its theory, proof systems, and solving technology. At the syntactic level, SAT is governed by a very simple local rule: a clause is true when at least one of its literals is true. This note studies the operator-valued analogue obtained by replacing that local OR rule with a projection cover. A variable x is assigned a projection P_x, its negation is assigned $I - P_x$, and a clause

$$C = (\ell_1 \vee \cdots \vee \ell_k)$$

Date: July 8, 2026.

is declared satisfied when

$$P_{\ell_1} + \cdots + P_{\ell_k} \geq I$$

in the Loewner order.

The definition is deliberately clause-theoretic. It is not merely a hypergraph partition-of-unity relation. In particular, monotone formulas remain trivially satisfiable by setting every $P_x = I$, just as in ordinary SAT. The tension is created by negation:

$$P_x + P_{\neg x} = I.$$

Complementary literals compete for the same projective resource, while each clause asks the resources appearing in it to cover the identity.

Several nearby operator-valued theories are useful points of comparison. Fritz's free hypergraph C^*-algebras impose partition-of-unity relations

$$\sum_{v \in e} p_v = 1$$

on hyperedges [13]. Quantum graph coloring and projective-rank parameters impose projection relations encoding color partitions and edge exclusivity [14]. Atserias–Kolaitis–Severini study generalized satisfiability problems via operator assignments for Boolean CSPs [8], and the broader distinction between commutative and noncommutative CSPs is studied, for example, in [11]. Boolean constraint system (BCS) algebras provide an especially close comparison: variables are represented by self-adjoint unitaries, variables in a context commute, and forbidden joint spectral projections are set to zero [9]. Thus, for an OR clause, BCS/AKS semantics agrees with the locally commuting version of the cover condition below. Ciardo's quantum-homomorphism/minion framework gives another nearby CSP perspective, but again its quantum objects are built from orthogonal subspace decompositions and local compatibility conditions [10]. Finally, the present model should not be confused with quantum k-SAT: our clauses are lower-cover inequalities for a single global assignment of literal projections, not local Hamiltonian constraints asking for a common zero-energy state [16, 17].

The move isolated here is smaller and more syntactic: keep the literal-complement rule and the Loewner lower-cover inequality, but impose no commutativity even within a clause. In this sense PSAT is neither a new presentation of BCS algebras nor a reformulation of quantum graph coloring. It is the non-locally-commuting CNF semantics obtained by interpreting Boolean OR directly as an operator cover.

At the same time, the local commutativity present in BCS/AKS semantics can be recovered internally. The usual Boolean gadgets for $z = x \wedge y$ and $z = x \vee y$ force the corresponding projections to commute. More generally, a locally commuting BCS context can be compiled into PSAT by introducing selector variables for its allowed joint spectral outcomes. Thus PSAT can recover the locally commuting semantics when desired, but does not build it into the ambient definition.

The elementary facts already give a coherent picture. If all projections commute, projective satisfiability is exactly classical satisfiability. Binary clauses become implications in the projection lattice, and projective 2-SAT therefore collapses to ordinary 2-SAT. Resolution remains sound, but some of the standard equisatisfiable transformations underlying SAT solving and reductions—Davis–Putnam elimination [4], DPLL-style splitting [5], and clause learning/resolution-based methods [6, 7]— require care, because the projection order lacks the Boolean interpolation properties these transformations often hide. The usual CNF encoding of graph q-coloring lifts exactly to projective graph q-coloring, so the framework contains a standard NP-complete problem as its classical/commuting shadow. For clauses of size at least four, anticommuting Clifford observables give explicit projective solutions to classically contradictory formulas. The ternary case is more delicate; the exact rational certificate in Appendix A shows that nonclassical qubit solutions already occur for 3-SAT.

2. Projective Cover Satisfiability

We allow formulas with repeated literals in clauses. This convention is harmless classically, but it is not harmless projectively in general. Accordingly, clauses below should be read as multisets of literals unless otherwise stated.

Definition 2.1. Let Φ be a CNF formula in variables $x_1, \dots, x_n$. A *projective cover assignment* of Φ on a Hilbert space $\mathcal{H}$ is a choice of projections

$$P_{x_i} \in B(\mathcal{H}), \qquad i = 1, \dots, n,$$

with literal projections defined by

$$P_{\neg x_i} = I - P_{x_i},$$

such that for every clause C of Φ,

$$\sum_{\ell \in C} P_\ell \geq I.$$

We say that Φ is *projectively satisfiable* if it has such an assignment on some nonzero Hilbert space. It is *d-dimensionally projectively satisfiable* if it has such an assignment in $M_d(\mathbb{C})$.

Remark 2.2. The condition may be written vectorially as

$$\sum_{\ell \in C} \|P_\ell \xi\|^2 \geq \|\xi\|^2, \qquad \xi \in \mathcal{H}.$$

Thus each clause is a lower frame condition for the selected literal subspaces.

Remark 2.3 (Locally commuting clauses)**.** Suppose the literal projections in a clause C commute. Then they may be simultaneously diagonalized, and the lower-cover condition

$$\sum_{\ell \in C} P_\ell \geq I$$

is equivalent to

$$\prod_{\ell \in C} (I - P_\ell) = 0.$$

Indeed, both conditions say that there is no joint spectral point at which all literals in C have value 0. Thus, after imposing local commutativity inside each clause, projective cover SAT agrees with the usual operator-assignment semantics for the Boolean OR relation in the sense of Atserias–Kolaitis–Severini [8]. It also agrees with the corresponding BCS-algebra constraint: in a BCS context, the relevant variables commute and the nonsatisfying joint spectral projection, in this case the all-false projection, is killed [9]. The point of the present definition is to drop this local commutativity requirement while retaining the order-theoretic cover condition.

The same distinction is familiar for XOR or linear-equation predicates. If one passes from projections to observables $A_x = 2P_x - I$, then a locally commuting XOR constraint requires the observables in the constraint to commute and have a prescribed product. This is the standard operator-solution regime behind linear-system games and magic-square-type examples [8, 12].

Remark 2.4 (Contexts versus covers)**.** The contrast with BCS algebras is useful to keep in mind. A BCS with contexts chooses, for each context, a finite Boolean algebra generated by commuting observables, and then imposes a Boolean relation spectrally inside that context. Projective cover SAT has no separate context relation. The same projection P_x appears in all clauses containing x, and a clause contributes only the single inequality $\sum_{\ell \in C} P_\ell \geq I$. Consequently, classical pp-definability and gadget reductions that are well behaved for BCS algebras need not be conservative here; the usual clause-splitting reduction fails precisely because it asks for a projection interpolant in a Loewner interval.

2.1. **Three ambient models.** It is useful to distinguish three versions of the decision problem. For a fixed clause size k, write k-PSAT for the projective cover satisfiability problem restricted to formulas whose clauses have size at most k; one may also consider the exact-k version when every clause has length exactly k.

Definition 2.5. Let Φ be a CNF formula.

(i) Φ is *matrix satisfiable* if it has a projective cover assignment in $M_d(\mathbb{C})$ for some finite $d \geq 1$.
(ii) Φ is *finite-von-Neumann satisfiable* if it has a projective cover assignment by projections in a finite von Neumann algebra.
(iii) Φ is *Hilbert-space satisfiable* if it has a projective cover assignment by projections in $B(\mathcal{H})$ for some nonzero Hilbert space $\mathcal{H}$.

These models are related by the evident implications

$$\text{matrix satisfiable} \Longrightarrow \text{finite-von-Neumann satisfiable} \Longrightarrow \text{Hilbert-space satisfiable.}$$

The converses are not formal. In particular, a Hilbert-space assignment need not visibly contain any finite-dimensional invariant subspace. Compressing infinite-dimensional projections to a finite-dimensional subspace produces positive contractions, not projections, and converting those contractions back into projections while preserving the clause inequalities is a genuine rounding/dilation issue.

Definition 2.6. A projective cover assignment is *strict* if there is an $\varepsilon > 0$ such that every clause C satisfies

$$\sum_{\ell \in C} P_\ell \geq (1+\varepsilon)I.$$

Strict satisfiability is a useful stability variant, but it is not automatic. For example, the tautological clause $x \vee \neg x$ always gives equality $P_x + P_{\neg x} = I$, so it can never be strict. Still, a strict cover would be much more amenable to finite-dimensional approximation: small perturbations of the inequalities would not destroy satisfiability. This leads to a basic finite-dimensional approximation question.

Question 2.7. *If a finite CNF formula is Hilbert-space satisfiable, must it be matrix satisfiable? More modestly, does every strict Hilbert-space cover admit a matrix cover, perhaps after a controlled dilation or rounding procedure?*

2.2. **Defining local commutativity.** Although PSAT does not impose local commutativity as part of the semantics, the usual Boolean definitional gadgets can force it. This is a useful way to compare PSAT with BCS and AKS-style locally commuting models: local Booleanity need not be built into the ambient semantics, but it can be imposed syntactically when needed.

Proposition 2.8 (Meet gadgets force commutation)**.** *Let P, Q, R be projections on a Hilbert space. Suppose the three PSAT clauses encoding*

$$z = x \wedge y$$

are satisfied:

$$\neg z \vee x, \qquad \neg z \vee y, \qquad \neg x \vee \neg y \vee z.$$

Equivalently,

$$R \le P, \qquad R \le Q, \qquad P + Q \le I + R.$$

Then P and Q commute, and

$$R = PQ = P \wedge Q.$$

Proof. The inequalities $R \le P, Q$ imply that R commutes with both P and Q, and that $P - R$ and $Q - R$ are projections on $(I - R)\mathcal{H}$. The third inequality gives

$$(P - R) + (Q - R) \le I - R.$$

If A and B are projections with $A + B \le I$, then $AB = BA = 0$: for $v \in \mathrm{Ran}(A)$, positivity of $I - A - B$ gives $0 \le \langle (I - A - B)v, v\rangle = -\langle Bv, v\rangle$, hence $Bv = 0$, and similarly with A and B interchanged. Applying this to $A = P - R$ and $B = Q - R$ gives

$$(P - R)(Q - R) = (Q - R)(P - R) = 0.$$

Thus P and Q commute and

$$PQ = (R + (P - R))(R + (Q - R)) = R.$$

□

Proposition 2.9 (Join gadgets force commutation)**.** *Let P, Q, R be projections. Suppose the three clauses encoding*

$$z = x \vee y$$

are satisfied:

$$\neg x \vee z, \qquad \neg y \vee z, \qquad \neg z \vee x \vee y.$$

Equivalently,

$$P \le R, \qquad Q \le R, \qquad R \le P + Q.$$

Then P and Q commute, and

$$R = P \vee Q.$$

Proof. Since $P, Q \le R$, both P and Q commute with R. On the Hilbert space $R\mathcal{H}$, the complements $R - P$ and $R - Q$ are projections. The inequality $R \le P + Q$ is equivalent, inside $R\mathcal{H}$, to

$$(R - P) + (R - Q) \le R.$$

By the same argument as above, $(R - P)(R - Q) = (R - Q)(R - P) = 0$. Hence P and Q commute. The common complement of P and Q inside $R\mathcal{H}$ is zero, so their join inside $R\mathcal{H}$ is R. Therefore $R = P \vee Q$. □

Corollary 2.10 (Local commutativity is PSAT-definable)**.** *Given finitely many literals $\ell_1, \ldots, \ell_m$, one can add auxiliary variables and clauses of size at most three so that every projective cover assignment forces the projections P_{ℓ_i} to commute pairwise. For instance, it is enough to impose a meet gadget*

$$z_{ij} = \ell_i \wedge \ell_j$$

for every pair $i < j$.

Remark 2.11. This sharpens the comparison with BCS algebras. BCS semantics builds local commutativity into the definition of a context and then evaluates Boolean relations spectrally. PSAT drops local commutativity by default, but ordinary SAT clauses can enforce it when one wants to recover the locally commuting context. Thus the difference is not that PSAT cannot express local Booleanity; rather, local Booleanity becomes an optional constraint instead of part of the ambient semantics.

Proposition 2.12 (Classical and commuting solutions)**.** *Every classical satisfying assignment gives a projective cover assignment in dimension one. Conversely, if Φ has a projective cover assignment by mutually commuting projections, then Φ is classically satisfiable.*

Proof. A Boolean assignment gives the one-dimensional projections $P_x = I$ if x is true and $P_x = 0$ if x is false.

Conversely, finitely many commuting projections generate a finite-dimensional commutative C^*-algebra. Decompose the Hilbert space into its nonzero joint spectral subspaces. On each such subspace, every literal projection is either 0 or I. The clause inequality then says that in each joint spectral atom, every clause contains at least one literal with value I. Choosing any nonzero joint spectral atom gives a classical satisfying assignment. □

Thus noncommutativity is the only source of projective satisfiability beyond ordinary SAT.

2.3. **Spectral selector compilation.** The previous corollary forces commutativity pair by pair. There is also a more intrinsic way to compile an entire locally commuting context into ordinary PSAT clauses. This is the direct analogue of replacing a commuting context by its joint spectral projections.

Let U be a finite set of Boolean variables and let $S \subseteq \{0,1\}^U$ be a Boolean relation, regarded as the set of allowed assignments on the context U. Introduce a new selector variable y_α for each $\alpha \in S$. Add the following clauses:

(i) the selector cover clause
$$\bigvee_{\alpha \in S} y_\alpha;$$
(ii) the selector-disjointness clauses
$$\neg y_\alpha \vee \neg y_\beta, \qquad \alpha \neq \beta;$$
(iii) the consistency clauses, for every $\alpha \in S$ and $x \in U$,
$$\neg y_\alpha \vee x \quad \text{if } \alpha(x) = 1,$$
and
$$\neg y_\alpha \vee \neg x \quad \text{if } \alpha(x) = 0.$$

Proposition 2.13 (Spectral selector compilation)**.** *Let U be a finite context and $S \subseteq \{0,1\}^U$ a Boolean relation. The PSAT clauses above are satisfiable in a given Hilbert-space model if and only if there is a representation of the context by commuting projections $\{P_x : x \in U\}$ whose joint spectrum is contained in S.*

Proof. Suppose first that the PSAT selector clauses are satisfied. Write $Q_\alpha = P_{y_\alpha}$. The clauses $\neg y_\alpha \vee \neg y_\beta$ give
$$(I - Q_\alpha) + (I - Q_\beta) \geq I,$$
hence $Q_\alpha + Q_\beta \leq I$. For projections this is equivalent to $Q_\alpha Q_\beta = 0$. Thus the Q_α are pairwise orthogonal. The selector cover clause gives
$$\sum_{\alpha \in S} Q_\alpha \geq I.$$
Since the sum of pairwise orthogonal projections is itself a projection bounded above by I, it follows that
$$\sum_{\alpha \in S} Q_\alpha = I.$$

The consistency clauses say that

$$Q_\alpha \le P_x \quad \text{if } \alpha(x) = 1, \qquad Q_\alpha \le I - P_x \quad \text{if } \alpha(x) = 0.$$

Therefore, for each $x \in U$,

$$A_x := \sum_{\alpha:\alpha(x)=1} Q_\alpha \le P_x, \qquad B_x := \sum_{\alpha:\alpha(x)=0} Q_\alpha \le I - P_x.$$

But $A_x + B_x = I$, so the second inequality gives $P_x \le A_x$. Hence

$$P_x = A_x = \sum_{\alpha:\alpha(x)=1} Q_\alpha.$$

Thus all P_x, $x \in U$, are diagonal with respect to the common orthogonal partition $\{Q_\alpha\}_{\alpha\in S}$. In particular they commute, and their joint spectral outcomes lie in S.

Conversely, suppose $\{P_x : x \in U\}$ are commuting projections with joint spectrum contained in S. For $\alpha \in S$, let Q_α be the joint spectral projection for the assignment α; if α does not occur, then $Q_\alpha = 0$. Since the forbidden joint spectral projections vanish, the allowed projections Q_α, $\alpha \in S$, form an orthogonal partition of identity. Setting $P_{y_\alpha} = Q_\alpha$ satisfies the selector cover, selector-disjointness, and consistency clauses. □

Remark 2.14. Thus locally commuting BCS constraints are not external to PSAT: they are a definable fragment. BCS algebras take contexts and their commuting joint spectral projections as primitive structure. PSAT can introduce those spectral projections as ordinary variables and force the corresponding decomposition by cover clauses. The compilation may be exponential in the arity of a context if one lists all allowed assignments, but for fixed constraint languages this is a constant-size gadget.

2.4. **Binary clauses and projective 2-SAT.** Binary clauses are especially rigid.

Lemma 2.15 (Binary clauses are implications). *For projections P_ℓ and P_m, the projective clause*

$$\ell \vee m$$

is satisfied, i.e.

$$P_\ell + P_m \ge I,$$

if and only if

$$P_{\neg\ell} \le P_m.$$

Equivalently, $\neg\ell \Rightarrow m$ holds in the projection order.

Proof. Since $P_{\neg\ell} = I - P_\ell$, the inequality $P_\ell + P_m \ge I$ is exactly $P_m \ge I - P_\ell = P_{\neg\ell}$. □

Theorem 2.16 (Projective 2-SAT collapses). *A 2-CNF formula is projectively satisfiable if and only if it is classically satisfiable.*

Proof. Only the converse is nontrivial. Unit clauses may be treated as duplicated binary clauses, since $2P \ge I$ forces a projection P to be I. Suppose, then, that a 2-CNF formula has a projective cover assignment. Each clause gives the usual implication in the projection order. If the formula were classically unsatisfiable, the standard implication graph criterion for 2-SAT would give a variable x such that there are implication paths

$$x \Rightarrow \neg x, \qquad \neg x \Rightarrow x.$$

The projection-order implications give

$$P_x \le P_{\neg x} = I - P_x, \qquad I - P_x \le P_x.$$

Thus $P_x = I - P_x$, impossible for a projection on a nonzero Hilbert space. Therefore the formula is classically satisfiable. □

Remark 2.17. This proof also shows that satisfiable 2-CNF formulas impose exactly the same partial-order constraints on projections as their classical implication graphs.

3. Elementary proof calculus

Resolution remains sound for projective covers.

Proposition 3.1 (Resolution is sound)**.** *Suppose the clauses*

$$A \vee x, \qquad B \vee \neg x$$

are projectively satisfied. Then the resolvent

$$A \vee B$$

is projectively satisfied.

Proof. Let

$$S_A = \sum_{\ell \in A} P_\ell, \qquad S_B = \sum_{\ell \in B} P_\ell.$$

The two hypotheses are

$$S_A + P_x \geq I, \qquad S_B + I - P_x \geq I.$$

The second inequality gives $S_B \geq P_x$. Hence

$$S_A + S_B \geq S_A + P_x \geq I.$$

This is precisely the projective satisfaction of $A \vee B$. □

However, classical syntactic manipulations involving repeated literals require care. A particularly useful special case is the following.

Lemma 3.2 (Padding binary clauses)**.** *Let P and Q be projections and let $r \geq 1$ be an integer. Then*

$$Q + rP \geq I \iff Q + P \geq I.$$

Consequently a binary clause $\ell \vee m$ may be encoded as an exactly k-literal clause

$$\ell \vee m \vee m \vee \cdots \vee m$$

for any $k \geq 2$, without changing projective satisfiability.

Proof. The implication $Q + P \geq I \Rightarrow Q + rP \geq I$ is immediate. Conversely, assume $Q + rP \geq I$. If $\xi \in \ker P$, then

$$\langle Q\xi, \xi \rangle \geq \|\xi\|^2.$$

Since Q is a projection, this forces $Q\xi = \xi$. Thus $\ker P \subseteq \mathrm{Ran}(Q)$, equivalently $I - P \leq Q$. Hence $Q + P \geq I$. □

Example 3.3 (Contraction can fail)**.** The preceding lemma is special to binary padding. Let P be the projection onto $e_1 \in \mathbb{C}^2$, and let A and B be the rank-one projections onto

$$u = \left(\frac{3}{5}, \frac{4}{5}\right), \qquad v = \left(\frac{5}{13}, \frac{12}{13}\right).$$

Then

$$A + B = \begin{pmatrix} \frac{2146}{4225} & \frac{3528}{4225} \\ \frac{3528}{4225} & \frac{6304}{4225} \end{pmatrix}.$$

A direct calculation gives

$$A + B + 2P - I \geq 0, \qquad \det(A + B + P - I) = -\frac{378}{845} < 0.$$

Thus $A + B + 2P \geq I$, but $A + B + P \not\geq I$. Equivalently, a clause containing one literal twice may be projectively satisfied even though the contracted clause is not.

Remark 3.4. Thus projective cover semantics is monotone under weakening, but not under arbitrary contraction of repeated literals. This is one source of separation between classical proof transformations and projective cover semantics.

3.1. **Clause splitting and interpolation.** The ordinary reduction from k-SAT to 3-SAT is based on splitting longer clauses by auxiliary variables. In the operator-cover semantics, this step is not formal: it asks for a projection interpolant in a Loewner-order interval.

Consider the classical split of a 4-clause

$$a \vee b \vee c \vee d$$

into

$$(a \vee b \vee z) \wedge (\neg z \vee c \vee d).$$

For projections A, B, C, D, the original clause says

$$A + B + C + D \geq I.$$

The split asks for a projection Z such that

$$A + B + Z \geq I, \qquad I - Z + C + D \geq I.$$

Equivalently,

$$I - A - B \leq Z \leq C + D.$$

Thus clause splitting is exactly a projection interpolation problem.

Proposition 3.5 (The usual split is not projectively sound)**.** *There exist projections $A, B, C, D \in M_2(\mathbb{C})$ such that*

$$A + B + C + D \geq I,$$

but there is no projection $Z \in M_2(\mathbb{C})$ satisfying

$$A + B + Z \geq I, \qquad I - Z + C + D \geq I.$$

Proof. Fix $0 < c < 1$. There are rank-one projections $A, B \in M_2(\mathbb{C})$ with

$$A + B = \begin{pmatrix} 1 - c & 0 \\ 0 & 1 + c \end{pmatrix},$$

and rank-one projections $C, D \in M_2(\mathbb{C})$ with

$$C + D = \begin{pmatrix} c & 0 \\ 0 & 2 - c \end{pmatrix}.$$

Indeed, the sum of two rank-one projections in $M_2(\mathbb{C})$ can have arbitrary spectrum $\{1 - t, 1 + t\}$ with $0 \leq t \leq 1$.

Then

$$A + B + C + D = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix} \geq I,$$

so the original 4-clause is projectively satisfied.

Suppose a projection Z satisfied the split inequalities. Then

$$I - A - B \leq Z \leq C + D,$$

so

$$\begin{pmatrix} c & 0 \\ 0 & -c \end{pmatrix} \le Z \le \begin{pmatrix} c & 0 \\ 0 & 2-c \end{pmatrix}.$$

Let $e_1 = (1, 0)$. The two inequalities imply

$$\langle Ze_1, e_1 \rangle \ge c, \qquad \langle Ze_1, e_1 \rangle \le c.$$

Hence equality holds. Since both $Z - (I - A - B)$ and $(C + D) - Z$ are positive, zero quadratic form on e_1 forces

$$Ze_1 = ce_1.$$

But a projection has only eigenvalues 0 and 1, contradicting $0 < c < 1$. Thus no such interpolating projection exists. ☐

Remark 3.6. The commutative case has the needed interpolation property: after simultaneous diagonalization, the inequalities reduce pointwise to Boolean inequalities, and one can choose the auxiliary truth value z coordinate by coordinate. The noncommutative projection order lacks this classical Riesz-interpolation behavior. Consequently, arity may be a genuine feature of projective cover satisfiability: projective k-SAT need not reduce to projective 3-SAT by the usual classical gadgets.

4. Separation results

4.1. **A Clifford separation for $k \ge 4$.** Let Φ_k be the complete k-CNF contradiction on variables $x_1, \dots, x_k$: it contains all 2^k possible signed k-clauses

$$\ell_1^{\varepsilon_1} \vee \cdots \vee \ell_k^{\varepsilon_k}, \qquad \varepsilon \in \{\pm 1\}^k,$$

where $\ell_i^+ = x_i$ and $\ell_i^- = \neg x_i$. Classically, Φ_k is unsatisfiable, since every Boolean assignment falsifies exactly one of these clauses.

Theorem 4.1 (Clifford projective covers)**.** *For every $k \ge 4$, the formula Φ_k is projectively satisfiable.*

Proof. Let $A_1, \dots, A_k$ be self-adjoint anticommuting unitaries on a finite- dimensional Hilbert space:

$$A_i^2 = I, \qquad A_i A_j + A_j A_i = 0 \quad (i \ne j).$$

Define

$$P_{x_i} = \frac{I + A_i}{2}, \qquad P_{\neg x_i} = \frac{I - A_i}{2}.$$

For a signed clause with signs $\varepsilon_i \in \{\pm 1\}$, the clause sum is

$$\sum_{i=1}^{k} P_{\ell_i^{\varepsilon_i}} = \frac{k}{2} I + \frac{1}{2} \sum_{i=1}^{k} \varepsilon_i A_i.$$

The anticommutation relations give

$$\left(\sum_{i=1}^{k} \varepsilon_i A_i \right)^2 = kI,$$

so the spectrum of $\sum_i \varepsilon_i A_i$ is contained in $\{\pm\sqrt{k}\}$. Therefore

$$\sum_{i=1}^{k} P_{\ell_i^{\varepsilon_i}} \ge \frac{k - \sqrt{k}}{2} I.$$

For $k \ge 4$ this lower bound is at least I. Hence every signed clause is projectively satisfied. ☐

Corollary 4.2. *For every $k \geq 4$, projective k-SAT differs from classical k-SAT.*

Remark 4.3. The construction is sharp for this symmetric example: for $k = 3$ the same Clifford estimate gives $(3 - \sqrt{3})/2 < 1$, so it does not satisfy the complete 3-CNF contradiction. This makes projective 3-SAT a natural borderline case.

4.2. **An exact qubit 3-SAT separation.** For $M_2(\mathbb{C})$, rank-one projections have a Bloch-sphere parametrization

$$P_x = \frac{1}{2}(I + n_x \cdot \sigma), \qquad n_x \in S^2,$$

and negation sends n_x to $-n_x$. Thus a ternary clause $\ell_1 \vee \ell_2 \vee \ell_3$ is satisfied precisely when

$$\|n_{\ell_1} + n_{\ell_2} + n_{\ell_3}\| \leq 1.$$

Appendix A gives a classically unsatisfiable 3-CNF formula with a qubit projective cover assignment. The instance has 10 variables and 42 ternary clauses. The certificate lists rational Bloch vectors

$$n_i \in \mathbb{Q}^3, \qquad \|n_i\| = 1,$$

one for each variable. The accompanying verifier checks, using exact rational arithmetic, that for every listed clause

$$\|\varepsilon_1 n_{i_1} + \varepsilon_2 n_{i_2} + \varepsilon_3 n_{i_3}\|^2 \leq 1.$$

The largest squared clause norm is

$$\frac{339}{361} < 1,$$

so the certificate has a substantial exact margin. The same verifier exhausts all 2^{10} Boolean assignments and confirms that no classical satisfying assignment exists.

Thus two-dimensional projective 3-SAT is strictly weaker than classical 3-SAT.

Remark 4.4. The exact certificate is not meant to be a minimal example. It is included to show that the ternary case already has a genuine qubit gap. Projective 2-SAT collapses to classical 2-SAT, while projective 3-SAT admits noncommuting solutions of classically unsatisfiable formulas.

5. Graph coloring and an NP-complete classical shadow

The operator-cover semantics contains projective graph coloring as a special case for every number of colors. Let $G = (V, E)$ be a graph and let $q \geq 2$. Introduce variables

$$x_{v,i}, \qquad v \in V, \quad i = 1, \dots, q,$$

where $x_{v,i}$ is meant to say that vertex v receives color i.

The usual CNF encoding of graph q-coloring consists of the clauses

$$x_{v,1} \vee x_{v,2} \vee \cdots \vee x_{v,q}$$

for each vertex v, together with the binary clauses

$$\neg x_{v,i} \vee \neg x_{v,j} \quad (i < j)$$

and

$$\neg x_{v,i} \vee \neg x_{w,i} \quad (vw \in E).$$

If one insists on clauses of exactly length q, the binary clauses may be padded by repeating one literal, using the preceding lemma.

Definition 5.1. A *projective q-coloring* of G on a Hilbert space $\mathcal{H}$ is a family of projections $P_{v,i} \in B(\mathcal{H})$ such that

$$\sum_{i=1}^{q} P_{v,i} = I \qquad (v \in V),$$

and

$$P_{v,i}P_{w,i} = 0 \qquad (vw \in E,\ i = 1, \dots, q).$$

Proposition 5.2 (Coloring encoding)**.** *The projective cover assignments of the usual graph q-coloring CNF are exactly the projective q-colorings of G.*

Proof. Write $P_{v,i}$ for the projection assigned to $x_{v,i}$. The positive q-clause at v gives

$$P_{v,1} + \cdots + P_{v,q} \geq I.$$

The at-most-one-color clauses give, for $i < j$,

$$(I - P_{v,i}) + (I - P_{v,j}) \geq I,$$

equivalently

$$P_{v,i} + P_{v,j} \leq I.$$

For projections this is the same as $P_{v,i}P_{v,j} = 0$. Thus the projections $P_{v,1}, \dots, P_{v,q}$ are pairwise orthogonal, and the positive clause inequality upgrades to

$$P_{v,1} + \cdots + P_{v,q} = I.$$

Similarly, the edge clauses give

$$(I - P_{v,i}) + (I - P_{w,i}) \geq I,$$

which is equivalent to $P_{v,i} + P_{w,i} \leq I$, hence to $P_{v,i}P_{w,i} = 0$. This is precisely a projective q-coloring. The converse is immediate by reversing the argument. □

Corollary 5.3. *For every fixed $q \geq 3$, exact-q projective cover SAT contains the usual projective graph q-coloring problem as a syntactic subproblem.*

Remark 5.4 (Complexity calibration)**.** The commuting part of this subproblem is ordinary graph q-coloring, which is NP-complete for every fixed $q \geq 3$ [15]. Thus the operator-cover syntax faithfully contains a standard NP-complete problem as its classical shadow. This observation should not be confused with an NP-hardness theorem for unrestricted projective cover satisfiability: allowing noncommuting projections may turn classically uncolorable instances into projectively colorable ones. The point is more modest but useful: the operator-cover semantics extends a familiar NP-complete CNF encoding, and its noncommutative solutions are genuine relaxations rather than artifacts of a bare hypergraph formalism.

6. Further questions

Question 6.1. *Find a smaller or more conceptual classically unsatisfiable 3-CNF formula with a qubit projective cover assignment. Can one obtain an infinite family?*

Question 6.2. *Is every Hilbert-space solution of projective 3-SAT approximable, or replaceable, by a finite-dimensional matrix solution?*

Question 6.3. *What is the least dimension of a projective cover assignment for the complete 4-CNF contradiction Φ_4? The Clifford construction gives a solution in dimension 4.*

Question 6.4. *Which classical reductions between CNF satisfiability problems preserve projective cover satisfiability? Resolution is sound and binary padding by literal duplication is sound, but the usual clause-splitting reductions are obstructed by the failure of projection interpolation.*

Appendix A. Exact qubit certificate

This appendix records the exact data for the qubit projective-cover certificate used in Section 4.2. For a variable x_i, let

$$P_{x_i} = \frac{1}{2}(I + n_i \cdot \sigma), \qquad P_{\neg x_i} = I - P_{x_i} = \frac{1}{2}(I - n_i \cdot \sigma),$$

where σ denotes the Pauli vector. Thus a clause $\ell_1 \vee \ell_2 \vee \ell_3$ is projectively satisfied exactly when

$$\|n_{\ell_1} + n_{\ell_2} + n_{\ell_3}\|^2 \leq 1,$$

where negation changes the sign of the corresponding Bloch vector.

The Bloch vectors are the following rational unit vectors.

variable	Bloch vector
x_0	$(-\frac{2}{7}, -\frac{3}{7}, -\frac{6}{7})$
x_1	$(0, -\frac{15}{17}, \frac{8}{17})$
x_2	$(\frac{16}{25}, \frac{3}{5}, -\frac{12}{25})$
x_3	$(-\frac{14}{23}, \frac{3}{23}, \frac{18}{23})$
x_4	$(\frac{4}{13}, -\frac{12}{13}, -\frac{3}{13})$
x_5	$(\frac{6}{19}, \frac{17}{19}, \frac{6}{19})$
x_6	$(\frac{6}{19}, -\frac{6}{19}, \frac{17}{19})$
x_7	$(-\frac{17}{19}, -\frac{6}{19}, -\frac{6}{19})$
x_8	$(\frac{9}{11}, -\frac{6}{11}, \frac{2}{11})$
x_9	$(\frac{2}{3}, -\frac{1}{3}, \frac{2}{3})$

The formula has the following 42 clauses.

No.	clause	No.	clause
1	$x_0 \vee x_3 \vee x_8$	2	$\neg x_0 \vee \neg x_3 \vee \neg x_8$
3	$x_0 \vee \neg x_7 \vee \neg x_8$	4	$\neg x_0 \vee x_7 \vee x_8$
5	$x_1 \vee x_2 \vee \neg x_4$	6	$\neg x_1 \vee \neg x_2 \vee x_4$
7	$x_1 \vee \neg x_3 \vee x_5$	8	$\neg x_1 \vee x_3 \vee \neg x_5$
9	$x_1 \vee \neg x_3 \vee \neg x_8$	10	$\neg x_1 \vee x_3 \vee x_8$
11	$x_1 \vee \neg x_4 \vee \neg x_6$	12	$\neg x_1 \vee x_4 \vee x_6$
13	$x_2 \vee x_3 \vee x_4$	14	$\neg x_2 \vee \neg x_3 \vee \neg x_4$
15	$x_2 \vee x_3 \vee x_7$	16	$\neg x_2 \vee \neg x_3 \vee \neg x_7$
17	$x_2 \vee \neg x_4 \vee \neg x_5$	18	$\neg x_2 \vee x_4 \vee x_5$
19	$x_2 \vee x_4 \vee \neg x_8$	20	$\neg x_2 \vee \neg x_4 \vee x_8$
21	$x_2 \vee \neg x_5 \vee x_6$	22	$\neg x_2 \vee x_5 \vee \neg x_6$
23	$x_2 \vee x_7 \vee x_8$	24	$\neg x_2 \vee \neg x_7 \vee \neg x_8$
25	$x_2 \vee \neg x_8 \vee x_9$	26	$\neg x_2 \vee x_8 \vee \neg x_9$
27	$x_3 \vee x_4 \vee \neg x_6$	28	$\neg x_3 \vee \neg x_4 \vee x_6$
29	$x_3 \vee \neg x_5 \vee \neg x_7$	30	$\neg x_3 \vee x_5 \vee x_7$
31	$x_3 \vee \neg x_6 \vee \neg x_7$	32	$\neg x_3 \vee x_6 \vee x_7$
33	$x_4 \vee x_5 \vee \neg x_6$	34	$\neg x_4 \vee \neg x_5 \vee x_6$
35	$x_4 \vee x_6 \vee \neg x_9$	36	$\neg x_4 \vee \neg x_6 \vee x_9$
37	$x_5 \vee x_6 \vee x_7$	38	$\neg x_5 \vee \neg x_6 \vee \neg x_7$
39	$x_5 \vee x_8 \vee \neg x_9$	40	$\neg x_5 \vee \neg x_8 \vee x_9$
41	$x_6 \vee \neg x_7 \vee \neg x_9$	42	$\neg x_6 \vee x_7 \vee x_9$

The exact verification is as follows. Each displayed vector has rational norm one. For every clause C, if s_C is the signed sum of its three Bloch vectors, then direct rational arithmetic gives

$$\|s_C\|^2 \leq \frac{339}{361} < 1.$$

Hence every clause is strictly projectively covered in $M_2(\mathbb{C})$. On the other hand, exhaustive enumeration of the 2^{10} Boolean assignments shows that the formula is classically unsatisfiable.

Appendix B. Exact rational verifier

For reproducibility, the following is the complete verifier used for the certificate in Appendix A. It uses only exact rational arithmetic.

```
"""
Exact qubit projective-cover certificate for a classically UNSAT 3-CNF.
Answers: a classically unsatisfiable 3-CNF formula that admits an exact
qubit (M_2(C)) projective cover, with rational Bloch vectors.

Semantics (qubit, rank-one projections P = 1/2 (I + n.sigma)):
  clause (l1 v l2 v l3) is projectively satisfied  <=>  || n_l1 + n_l2 + n_l3 || <= 1,
  where a negated literal flips its Bloch vector (n -> -n).
All checks below are in exact rational arithmetic (fractions.Fraction).
"""
from fractions import Fraction as F
import itertools

# --- 10 rational unit Bloch vectors, one per variable x0..x9 ---
BLOCH = [
 (F(-2,7), F(-3,7), F(-6,7)),
 (F(0),    F(-15,17),F(8,17)),
 (F(16,25),F(3,5),  F(-12,25)),
 (F(-14,23),F(3,23), F(18,23)),
 (F(4,13), F(-12,13),F(-3,13)),
 (F(6,19), F(17,19), F(6,19)),
 (F(6,19), F(-6,19), F(17,19)),
 (F(-17,19),F(-6,19),F(-6,19)),
 (F(9,11), F(-6,11), F(2,11)),
 (F(2,3),  F(-1,3),  F(2,3)),
]

# --- 42 clauses; each literal is (variable, sign), sign=+1 plain, -1 negated ---
CLAUSES = [
 [(0,1),(3,1),(8,1)],   [(0,-1),(3,-1),(8,-1)],
 [(0,1),(7,-1),(8,-1)], [(0,-1),(7,1),(8,1)],
 [(1,1),(2,1),(4,-1)],  [(1,-1),(2,-1),(4,1)],
 [(1,1),(3,-1),(5,1)],  [(1,-1),(3,1),(5,-1)],
 [(1,1),(3,-1),(8,-1)], [(1,-1),(3,1),(8,1)],
 [(1,1),(4,-1),(6,-1)], [(1,-1),(4,1),(6,1)],
 [(2,1),(3,1),(4,1)],   [(2,-1),(3,-1),(4,-1)],
 [(2,1),(3,1),(7,1)],   [(2,-1),(3,-1),(7,-1)],
 [(2,1),(4,-1),(5,-1)], [(2,-1),(4,1),(5,1)],
 [(2,1),(4,1),(8,-1)],  [(2,-1),(4,-1),(8,1)],
 [(2,1),(5,-1),(6,1)],  [(2,-1),(5,1),(6,-1)],
 [(2,1),(7,1),(8,1)],   [(2,-1),(7,-1),(8,-1)],
 [(2,1),(8,-1),(9,1)],  [(2,-1),(8,1),(9,-1)],
 [(3,1),(4,1),(6,-1)],  [(3,-1),(4,-1),(6,1)],
 [(3,1),(5,-1),(7,-1)], [(3,-1),(5,1),(7,1)],
 [(3,1),(6,-1),(7,-1)], [(3,-1),(6,1),(7,1)],
 [(4,1),(5,1),(6,-1)],  [(4,-1),(5,-1),(6,1)],
 [(4,1),(6,1),(9,-1)],  [(4,-1),(6,-1),(9,1)],
 [(5,1),(6,1),(7,1)],   [(5,-1),(6,-1),(7,-1)],
 [(5,1),(8,1),(9,-1)],  [(5,-1),(8,-1),(9,1)],
 [(6,1),(7,-1),(9,-1)], [(6,-1),(7,1),(9,1)],
]

def check_units():
    for i,(a,b,c) in enumerate(BLOCH):
        assert a*a+b*b+c*c == 1, f"vector {i} not unit"
    return True

def worst_clause():
```

```
    w = F(0); arg=None
    for cl in CLAUSES:
        sx=sy=sz=F(0)
        for (v,s) in cl:
            sx+=s*BLOCH[v][0]; sy+=s*BLOCH[v][1]; sz+=s*BLOCH[v][2]
        nn = sx*sx+sy*sy+sz*sz
        if nn>w: w=nn; arg=cl
    return w, arg

def classically_satisfiable(n=10):
    for bits in itertools.product([0,1],repeat=n):
        if all(any((bits[v] if s==1 else 1-bits[v]) for (v,s) in cl) for cl in CLAUSES):
            return True
    return False

if __name__=="__main__":
    assert check_units()
    w,arg = worst_clause()
    print("variables:", 10, " clauses:", len(CLAUSES))
    print("all Bloch vectors exactly unit:           True")
    print("max clause ||sum||^2 (exact):             ", w, "=", float(w))
    print("every clause satisfied (||sum||^2 <= 1):   ", w<=1)
    print("strict projective margin 1 - max:          ", float(1-w))
    print("classically satisfiable (2^10 enumeration):", classically_satisfiable())
    print()
    print("=> classically UNSAT, but exactly qubit-projectively coverable.")
```

Appendix C. Classical SAT moves in the projective setting

We record, in compressed form, how some standard SAT manipulations behave under projective cover semantics. The list is not meant as a survey of SAT solving; it is meant to identify which familiar Boolean moves survive the passage to operator covers. The recurring theme is that order-theoretic consequences are usually sound, while classically equisatisfiable gadget transformations may require an interpolation or distributivity property which is absent from the projection lattice.

Classical move	Projective behavior
Tautology deletion	Safe. If a clause contains both x and $\neg x$, then $P_x + P_{\neg x} = I$, so the clause is automatically satisfied.
Weakening and subsumption	Safe. Adding literals only increases the clause sum. Hence if $C \subseteq D$, then C projectively implies D, and the larger clause is redundant in a formula containing both.
Pure literal elimination	Safe. If x appears only positively, set $P_x = I$; if it appears only negatively, set $P_x = 0$.
Unit propagation	Safe. A unit clause ℓ forces $P_\ell = I$ and hence $P_{\neg\ell} = 0$. In exact ternary syntax, $\ell \vee \ell \vee \ell$ has the same effect.
Binary implication	Safe and exact. The clause $\neg a \vee b$ is precisely $P_a \leq P_b$.
Equality and complement constraints	Safe when encoded by binary clauses. The pair $(\neg x \vee y) \wedge (x \vee \neg y)$ forces $P_x = P_y$; the analogous pair encoding $x \leftrightarrow \neg y$ forces $P_x = I - P_y$.
Meet and join definitions	Stronger than merely safe. The usual clauses for $z = x \wedge y$ or $z = x \vee y$ force P_x and P_y to commute, with $P_z = P_x \wedge P_y$ or $P_z = P_x \vee P_y$, respectively.
Resolution	Sound. From $A \vee x$ and $B \vee \neg x$ one obtains $A \vee B$ by the projection-order argument in the resolution proposition.
Contraction	Not generally sound. The inequality $S + 2P \geq I$ need not imply $S + P \geq I$. Thus clauses are best treated as multisets of literals.
Davis–Putnam elimination	Not sound in general. Eliminating x requires a projection P_x satisfying simultaneous lower and upper bounds $I - S_{A_i} \leq P_x \leq S_{B_j}$. The resolvents give only the inequalities $I - S_{A_i} \leq S_{B_j}$, not a projection interpolant.
Clause splitting	Not sound in general, for the same reason. Splitting a longer clause by an auxiliary variable asks for a projection in a Loewner interval, as in Section 3.1.
Tseitin transformations	Not automatically conservative. A Tseitin variable for a subformula would have to represent a join, meet, or other Boolean combination of possibly noncommuting projections. Such projections need not exist without additional hypotheses.
Horn least-model methods	Subtle. Binary Horn clauses are projection-order implications, but longer Horn clauses do not become meets of projections unless there is commutativity. The usual least-model/distributive-lattice proof does not transfer formally.

Thus ordinary equisatisfiability is not automatically preserved by projectivization. Projective cover satisfiability respects many monotone logical consequences, but reductions that require choosing an intermediate Boolean value become projection-interpolation problems.

Acknowledgments

The examples in this note were found and checked with the assistance of OpenAI's ChatGPT, which was also used for exploratory discussion of formulations and for editorial assistance. The mathematical claims and final text remain the author's responsibility.

References

[1] S. A. Cook, *The complexity of theorem-proving procedures*, Proc. Third Annual ACM Symposium on Theory of Computing (STOC), 1971, 151–158.
[2] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, Plenum, New York, 1972, 85–103.
[3] A. Biere, M. Heule, H. van Maaren, and T. Walsh (eds.), *Handbook of Satisfiability*, 2nd ed., Frontiers in Artificial Intelligence and Applications, vol. 336, IOS Press, 2021.
[4] M. Davis and H. Putnam, *A computing procedure for quantification theory*, J. ACM **7** (1960), 201–215.
[5] M. Davis, G. Logemann, and D. Loveland, *A machine program for theorem-proving*, Commun. ACM **5** (1962), 394–397.
[6] J. P. Marques-Silva and K. A. Sakallah, *GRASP: A search algorithm for propositional satisfiability*, IEEE Trans. Computers **48** (1999), 506–521.
[7] N. Eén and N. Sörensson, *An extensible SAT-solver*, in Theory and Applications of Satisfiability Testing, SAT 2003, Lecture Notes in Computer Science, vol. 2919, Springer, 2004, 502–518.
[8] A. Atserias, P. G. Kolaitis, and S. Severini, *Generalized satisfiability problems via operator assignments*, J. Comput. System Sci. **105** (2019), 171–198; arXiv:1704.01736.
[9] C. Paddock and W. Slofstra, *Satisfiability problems and algebras of Boolean constraint system games*, Illinois J. Math. **69** (2025), no. 1, 81–107.
[10] L. Ciardo, *Quantum advantage and CSP complexity*, J. Combin. Theory Ser. B **176** (2026), 404–439.
[11] A. A. Bulatov and L. Huang, *Satisfiability of commutative vs. non-commutative CSPs*, arXiv:2404.11709.
[12] R. Cleve, L. Liu, and W. Slofstra, *Perfect commuting-operator strategies for linear system games*, J. Math. Phys. **58** (2017), 012202; arXiv:1606.02278.
[13] T. Fritz, *Curious properties of free hypergraph C^*-algebras*, arXiv:1808.09220.
[14] V. I. Paulsen, S. Severini, D. Stahlke, I. G. Todorov, and A. Winter, *Estimating quantum chromatic numbers*, J. Funct. Anal. **270** (2016), 2188–2222; arXiv:1407.6918.
[15] M. R. Garey, D. S. Johnson, and L. Stockmeyer, *Some simplified NP-complete graph problems*, Theoret. Comput. Sci. **1** (1976), 237–267.
[16] S. Bravyi, *Efficient algorithm for a quantum analogue of 2-SAT*, Contemp. Math. **536** (2011), 33–48; arXiv:quant-ph/0602108.
[17] D. Gosset and D. Nagaj, *Quantum 3-SAT is QMA_1-complete*, SIAM J. Comput. **45** (2016), 1080–1128; arXiv:1302.0290.