Verifiable quantum advantage based on polynomials with planted structures

Markus Bläser[1], Michael Gullans[2,3], Dominik Hangleiter[4,5], Yuxuan Liu[6], and Youming Qiao[7]

[1]Center for Quantum Technologies (QuTe) and Department of Computer Science, Saarland University
[2]QuEra Computing Inc.
[3]Joint Center for Quantum Information and Computer Science, University of Maryland
[4]Institute for Theoretical Physics, ETH Zürich
[5]Simons Institute for the Theory of Computing, University of California at Berkeley
[6]Wuhan University
[7]Centre for Quantum Software and Information, University of Technology Sydney

A central question in the theory of quantum advantage is whether there are quantum advantage protocols with similar resource requirements as random circuit sampling that are also verifiable just from the classical outputs of the quantum computation. Here, we develop the idea of simulation secrets for verifiable advantage. A verifier can use a simulation secret to evaluate a cross-entropy test faster than it would take a classical adversary to pass the test. We instantiate this idea using IQP circuits described by cubic polynomials with *planted independent spaces*. These correspond to the largest independent set in the orbit of a polynomial under the general linear group and yield a low-rank stabilizer decomposition of the corresponding state. We conjecture that large independent spaces are invisible to a computationally bounded adversary, and therefore they cannot exploit them to pass the protocol. A second conjecture regards the fine-grained complexity of producing samples that pass the cross-entropy test for uniformly random polynomials. Under these conjectures, our scheme results in a polynomial gap between the verification time and the time a classical adversary would need to pass the protocol—both are exponential. It has a potential application to generating classically certifiable randomness, since the output distributions have high min-entropy. We estimate that the planted polynomial scheme is implementable using 100 logical qubits at logical error rates around 10^{-6}.

Contents

1 Introduction **2**
1.1 Motivation . . . 2
1.2 Quantum circuits from polynomials with planted independent spaces . . . 4
1.3 Verifiable advantage from planted independent spaces . . . 6
1.4 The planted independent space problem for cubic polynomials . . . 7
1.5 The cubic XEB problem . . . 10
1.6 Discussion and outlook . . . 10
1.7 Overview . . . 12

2 The polynomial planted independent space problem **13**

2.1 Boolean polynomials and tensors . . . 13
2.2 Planted independent space . . . 16
2.3 Statement of independent space problems . . . 17

3 Verification of IQP circuit sampling from planted independent spaces 18
3.1 IQP circuits and their statistical properties . . . 19
3.2 Verification via XEB . . . 22
3.3 Simulating IQP circuits with and without planted independent spaces . . . 23
3.4 Towards certified randomness from planted IQP circuits . . . 26

4 Worst-case hardness of the independent space problem 28
4.1 Slice rank (trilinear case) . . . 28
4.2 Symmetric slice rank (polynomial case) . . . 30

5 Cryptanalysis of the planted independent space problem 33
5.1 The $h/n > 1/2$ regime . . . 33
5.2 Analysis of Bell-sampling attacks . . . 37

A Attack on block-diagonal structure 44

1 Introduction

1.1 Motivation

The experimental demonstration of quantum computational advantage has been a major milestone for the field. Since the first demonstration in 2019 [AAB+19], experiments demonstrating quantum advantage have achieved higher quality and larger circuit volumes [WBC+21; MVM+24; DHL+25; GFZ+25]. A big drawback of those experiments, however, has been that they are not classically verifiable from the experimental outcomes themselves. This means that the validation of the advantage demonstrations has relied on separate data [Han26]. This is both practically and fundamentally significant; it is practically significant, because verification has required immense classical computational efforts to simulate computations just below the advantage threshold. It is fundamentally significant, since the lack of verification of the computational outputs themselves gives room for skeptics to doubt the claims, similar to how the first violations of Bell inequalities [ADR82] left room for loopholes, which were only eventually closed much later [GVW+15; HBD+15; SMC+15].

The reason why the current demonstrations of quantum advantage are not verifiable is because they rely on the computational task of random circuit sampling [HE23], which refers to the task of sampling from the output distribution of a quantum circuit chosen uniformly from a large family of circuits that is natural on a given architecture. The output distributions of those circuits have the so-called anticoncentration property which prevents sample-efficient verification of the correct distribution [HKEG19]. The best available way to benchmark random circuit sampling is to use the so-called *cross-entropy benchmark (XEB)* [BIS+18; AAB+19], which is the average of the ideal probabilities corresponding to the observed samples. This benchmark is sample-efficiently computable and distinguishes ideal samples from random ones. More broadly, generating samples that achieve a sufficiently high XEB score appears to be a classically hard task [AC17; GKC+24; Han26]. However, computing the XEB requires computational effort on a par with simulating an ideal experiment, rendering verification at least as hard as simulating a potentially noisy experiment.

Theoretically, the best known approaches to classically verifiable quantum advantage are interactive protocols based on post-quantum-secure cryptography [BCM+18; BKVV20] and solving the discrete log problem [Sho94] to show quantum advantage [KRVK24] or even break existing cryptosystems such as elliptic curve cryptography and RSA [Gid25; BZG+26]. However, the required circuit sizes for these problems are out of reach for next-generation quantum devices [TCY+26] as they require on the order of 10^8 to 10^9 Toffoli gates and on the order of 1000 logical qubits [KCVY22; BZG+26; Gid25].

This raises the question if random circuit sampling can be made classically verifiable by the use of cryptography while retaining small circuit sizes that might be achievable on first-generation logical processors with roughly 100 logical qubits and error rates on the order of 10^{-6} [Han26; TCY+26]. One approach, popularized by Aaronson [Aar23], is to devise circuits which are computationally indistinguishable from random circuits sampled from a hard-to-simulate family, but have hidden structure in their output distribution that is efficiently detectable. In the simplest case, this structure could be a single *secret peak* in the output distribution, corresponding to a single bitstring that is observed with high probability if the prover is honest, but is hard to find for an adversary [AZ24]. The verifier, on the other hand, would devise the circuits in such a way that she would be able to *plant* a peak of her choosing when sampling a circuit. Using her private knowledge of the peak location she will be able to distinguish an honest prover from an adversarial one.

A compelling candidate construction for this was proposed already by Shepherd and Bremner [SB09], using a family of commuting quantum circuits known as IQP circuits. These circuits have simple classical descriptions in terms of a classical linear code, and it turns out that certain properties of the code are reflected in the output distribution—a property exploited by the Shepherd/Bremner protocol. Unfortunately, cryptanalysis of the protocol revealed a classical solution [Kah23], and even generalizations of the scheme [BCJ25] can be broken [GH25]. Recent work has built off this idea to devise circuit families with peaks in certain conditional distributions [DFG+25], and also investigated the broader question of whether peaked circuits that are indistinguishable from random circuits even exist [AZ24; Zha25]. It remains open whether peaks can be planted to devise a verifiable quantum advantage scheme.

In this work, we consider a different paradigm for classically verifiable advantage based on *simulation secrets*. The idea is to plant secret structure in an otherwise random IQP circuit, knowledge of which simplifies classical simulation of the circuit. At the same time, this structure is invisible to a computationally bounded adversary and does not ostensibly alter the properties of its output distribution. In particular, the output distribution of the planted circuits has high entropy, as for the random case. In this way, the XEB can be used to verify samples since the verifier's private knowledge of the planted structure allows her to compute probabilities significantly faster than what an adversary without knowledge of the secret would be able to do.

We instantiate this idea with IQP circuits defined by random degree-3 Boolean polynomials [BMS16]. The structure we plant is a generalization of planted cliques. The gap between simulation time and verification time we obtain in this way is polynomial: Under a set of fine-grained complexity conjectures, passing the verification test without knowledge of the secret requires time $\Omega(2^{\kappa n})$ for some constant $\kappa > 1/2$ using polynomial space, while classical verification using the secret only requires time $\tilde{O}(2^{n(1/2+\epsilon)})$ and polynomial space for arbitrarily small $\epsilon > 0$. This may seem like a modest gap. However, with the right choice of parameters it allows for the verification of samples that are not simulatable classically directly from the output of the computation itself. What is more, we believe the gap for practical parameter sizes to be significantly larger than the asymptotic conjectures suggest. We estimate that a classically hard computation using at most 100 000 transversal gates on around 100 logical qubits or less could be verified using this scheme. We further conjecture that planted independent spaces remain

computationally hidden from quantum algorithms.

We anticipate that our protocol can be used to generate classically certified randomness. This is because the output distribution of our scheme has high min-entropy, and we conjecture that the planted structure remains invisible even to computationally bounded quantum adversaries. As a first step, we obtain a single-round entropy bound for quantum provers that pass the classical verification test, based on ideas of Aaronson and Hung [AH23]. Thus, our protocol marks a first step towards a practically useful cryptographic application of next-generation logical quantum computers with around 100 logical qubits [TCY+26].

1.2 Quantum circuits from polynomials with planted independent spaces

Our main conceptual contribution towards this end is to identify and study the *planted independent space* (PlantedIndSp) problem for Boolean polynomials. We work with Boolean polynomials of degree at most three, which we call cubic Boolean polynomials throughout this paper. An independent space of a cubic Boolean polynomial is a linear algebraic analogue of an independent set of a 3-uniform hypergraph, as will be defined and explained later. Crucially, given a random cubic polynomial with a planted independent space of dimension h, one can simulate the corresponding Boolean IQP circuits in time 2^{n-h}.

More specifically, we consider IQP circuits $C(g)$ which are defined by a cubic Boolean polynomial g on n-bit strings $x \in \mathbb{F}_2^n$ (all arithmetic is modulo 2)

$$g(x) = \sum_{i<j<k} G_{ijk} x_i x_j x_k + \sum_{i<j} G_{ij} x_i x_j + \sum_{i\in[n]} G_i x_i \,, \tag{1}$$

with $G_{ijk}, G_{ij}, G_i \in \mathbb{F}_2$. Throughout this article, we work with cubic polynomials with constant term 0 unless otherwise stated. Since $x^2 = x$ on $\mathbb{F}_2$, we identify functions $g : \mathbb{F}_2^n \to \mathbb{F}_2$ with their unique multilinear representatives in $\mathbb{F}_2[x_1, \ldots, x_n]$ (see Section 2.1). The corresponding IQP circuit acts on a computational basis state as

$$C(g)\,|x\rangle = (-1)^{g(x)}\,|x\rangle\,. \tag{2}$$

To define the sampling task, we start from a state preparation in the $|+^n\rangle$ state and measure in the X basis, giving rise to outcome amplitudes

$$a_g(x) = \frac{1}{2^n} \sum_{y\in\mathbb{F}_2^n} (-1)^{g(y)+x\cdot y} =: \mathrm{ngap}(g + \langle x, \cdot\rangle), \tag{3}$$

where $\langle x, y\rangle = x\cdot y$ denotes the inner product. As we explain below, in Section 1.5, we conjecture that there is a $\kappa > 1/2$ such that there is no algorithm that uses polynomial space and has runtime $o(2^{\kappa n})$.

To motivate independent spaces, let us begin by observing that a cubic polynomial $g(x) \in \mathbb{F}_2[x_1, \ldots, x_n]$ as in Eq. (1) can be associated naturally to a 3-uniform hypergraph $H(g)$, where each nonzero coefficient G_{ijk} corresponds to a hyperedge (i, j, k). Note that this hypergraph only depends on the degree-3 monomial coefficients of g. Recall that in a 3-uniform hypergraph H, a vertex subset is an independent set, if no hyperedge in H is contained in it. Given an independent set of size h in $H(g)$, we can simulate the corresponding IQP circuit in time $\tilde{O}(2^{n-h})$ and polynomial space. This is because the complement of an independent set is a vertex cover V and we can write the outcome amplitudes as

$$a_g(x) = \frac{1}{2^{|V|}} \sum_{z\in\mathbb{F}_2^{|V|}} (-1)^{x_V\cdot z}\,\mathrm{ngap}(g_z + \langle x_{V^c}, \cdot\rangle), \tag{4}$$

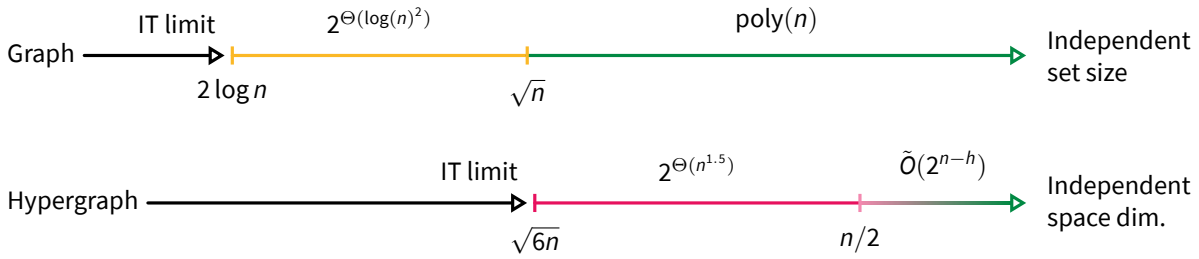


Figure 1: Distinguishing regimes for independent sets of random graphs and independent spaces of random hypergraphs. The information-theoretic (IT) limit denotes the typical size of an independent set/subspace. The second transition is the computational boundary where a better algorithm than the trivial one starts to exist.

where g_z is a quadratic polynomial in $|V^c|$ variables whose gap is efficiently computable and x_S is the restriction of x to $S \subset [n]$ [MBT+24; HKB+25]. Here, an independent set of a hypergraph is a vertex subset that contains no 3-ary hyperedges in the hypergraph. We can use this decomposition for verification: plant a large independent set of size $h = (1-c)n$ in an otherwise uniformly random hypergraph $H(g)$. The existence and knowledge of such an independent set reduce the simulation time to 2^{cn}, giving a κ/c-fold polynomial gap compared to the best unstructured algorithm.

But what is the complexity of finding a planted independent set? Independent sets are in one-to-one correspondence to cliques of the complement graph. Therefore, finding the planted independent set then amounts to solving the planted clique problem, which has been extensively studied for graphs. While the maximum clique problem is NP-complete in the worst case setting, planting a clique in the Erdős–Rényi random graph model $\mathcal{G}(n, 1/2)$ is conjectured to require $n^{\Theta(\log n)}$ time if the planted clique is of size between $(2+\epsilon)\log n$ and $O(\sqrt{n})$ [Jer92; Kuč95; AKS98; HS24]. It is of interest as a cryptographic assumption and it can be used for example as a basis of public-key cryptography schemes [ABW10], as well as to plant secret backdoors in machine-learning models [GKVZ22]. However, $n^{\Theta(\log n)}$ is not sufficiently hard for our purposes and moreover the problem is efficient for linear-size cliques. This is why we require a more sophisticated approach.

To achieve this, we observe that this approach still works even when there is a large independent set somewhere in the *orbit* of g under $\mathrm{GL}(n, \mathbb{F}_2)$ transformations, that is, the set of hypergraphs induced by the polynomials $g \circ A$ defined as $g \circ A(x) = g(Ax)$ for $A \in \mathrm{GL}(n, \mathbb{F}_2)$ and a given cubic polynomial g over $\mathbb{F}_2$. This leads to the following definition.

Definition 1 (Independent space)**.** *Let $g : \mathbb{F}_2^n \to \mathbb{F}_2$ be a cubic polynomial. We say that $V \leq \mathbb{F}_2^n$ is an* independent space *of g, if there exists $A \in \mathrm{GL}(n, \mathbb{F}_2)$ whose first $\dim(V)$ columns generate V and $g \circ A$ has an independent set on the first $\dim(V)$ coordinates.*

One may think of the problem to decide whether a given cubic polynomial g has a large independent space as an exponentially lifted equivalent of the planted independent set problem wherein the n graph vertices are replaced by the 2^n vectors over $\mathbb{F}_2^n$. In other words, while finding an independent set of size h can be thought of as finding an element $P \in S_n$ of the symmetric group such that $g \circ P$ has an independent set on the first h coordinates, finding an independent space asks for an element of the larger, general linear group $\mathrm{GL}(n, \mathbb{F}_2)$.

Analogously to planted independent sets in random graphs, we now consider planted independent spaces in random 3-uniform hypergraphs/random cubic polynomials. We find that a uniformly random cubic polynomial has an independent space of dimension $\leq \sqrt{6n}$. We give evidence that distinguishing a random polynomial from one with a planted (random) h-dimensional

independent space takes time $2^{\Theta(n^{1.5})}$ for $h \leq (1/2-\epsilon)n$ with $\epsilon > 0$. Just like for planted independent sets, this algorithm checks if there is an independent space of dimension $d = 2\sqrt{6n}$ by enumerating all these subspaces in time $2^{O(n^{1.5})}$ in order to distinguish a given polynomial from a random one. For any fixed constant $\epsilon > 0$ and all sufficiently large n, when $h \geq (1/2+\epsilon)n$, there exists a nontrivial algorithm with significantly better runtime $\tilde{O}(2^{n-h})$. This algorithm exploits the rank profile of Hessian slices of the degree-3 component, which may be the analogue of the degree distribution of a graph. The analogy between (graph) planted independent sets and (hypergraph) planted independent space is shown in Fig. 1.

Given these tight correspondences between independent sets and independent spaces and the fact that planted independent sets have been extensively studied, we conjecture these limits to be optimal. For the purposes of our main result, we make the following conjecture.

Conjecture 2 (Exponential planted independent space (ExpPlantedIndSp) conjecture)**.** *Consider cubic polynomials over $\mathbb{F}_2$ in n variables. There is no $o(2^n)$-time algorithm that distinguishes a random cubic polynomial from a random cubic polynomial with a planted h-dimensional independent space, where $h \leq (1/2-\epsilon)n$ for any constant $\epsilon > 0$.*

By planting an independent space of dimension $\lesssim n/2$ in random cubic polynomials we can obtain a polynomial gap between verification and simulation of samples from the corresponding IQP circuits.

1.3 Verifiable advantage from planted independent spaces

We consider a verifier who chooses a polynomial g uniformly at random with a planted uniformly random independent space of dimension $(1-c)n$ for $1/2 < c \leq 1$. She now reveals g to an efficient quantum prover, but keeps the planted independent space secret. The prover returns samples from running the IQP circuit $C(g)$, and the verifier exploits her secret knowledge of the independent space to verify those samples. She then repeats this for polynomially many random choices of g and averages the test outcomes.

For the verification test, the verifier uses the linear XEB, defined as

$$\chi(q,p) \coloneqq 2^n \sum_{x \in \mathbb{F}_2^n} q(x)p(x) - 1, \tag{5}$$

for sampled and target probability distributions over $\mathbb{F}_2^n$, q and p, respectively. The XEB gives a bona-fide benchmark for *anticoncentrating* probability distributions p whose normalized second moments $2^n \sum_x p(x)^2 = O(1)$ are constant. Since the planted independent space keeps the quadratic part of g uniformly random, this is the case and moreover the variance of the XEB is constant so that it can be sample-efficiently evaluated.

To evaluate the XEB for a single circuit the verifier therefore computes the ideal probabilities $p(x)$ corresponding to the observed samples x in time $\tilde{O}(2^{cn})$ using the vertex-cover decomposition algorithm [MBT+24] and averages the results to obtain an estimate $\hat{\chi}$ of the XEB. She accepts if $\hat{\chi} > 1$ (the ideal value is 2) and rejects otherwise. An honest prover running a near-ideal version of $C(g)$ will pass this test with high probability. On the other hand, we conjecture that there is no algorithm using polynomial space that achieves a high XEB score and has runtime $\tilde{O}(2^{n/2})$.

Conjecture 3 (Exponential XEB conjecture)**.** *Sampling from a distribution q that achieves $\chi(q,p) \geq 1$ against the ideal output distribution p of a random degree-3 IQP circuit with inverse polynomial failure probability requires time $\Omega(2^{\kappa n})$ for some constant $\kappa > 1/2$ or superpolynomial space.*

We discuss the evidence for Conjecture 3 below. Conjectures 2 and 3 imply our main result, as we show in Section 3.

Theorem 4 (Verifiable advantage from planted independent spaces)**.** *Let g be a random cubic polynomial with a planted $(1-c)n$-dimensional independent space for $1/2 < c < \kappa$ and assume Conjectures 2 and 3. Then,*

(i) the XEB of the output distribution of the IQP circuit $C(g)$ can be estimated in time $\tilde{O}(2^{cn})$ and polynomial space,

(ii) an honest, polynomial-time quantum prover passes the XEB test with high probability,

(iii) there is no classical polynomial-space algorithm with runtime $o(2^{\kappa n})$ that passes the XEB test with high probability.

In other words, we obtain a polynomial gap of at least $S \sim V^{\kappa/c}$ between verification and simulation times for cubic IQP circuits with $h = (1-c)n$-dimensional planted independent spaces. We note that we expect the actual cost differences to be significant in the parameter regime of $n \lesssim 100$ and $c > 1/2$. In this case, verifying samples is feasible using simulation of an at most 50-qubit IQP circuit, while simulation is beyond the scope of the best classical algorithms on the available hardware.

To the best of our knowledge this constitutes the first example where a significant gap is possible for a random-circuit-like scheme under a plausible cryptographic assumption. It is thus the first such example where a quantum computational task that cannot be solved on the best available classical computers can be verified using those same computers.

The key question for the efficacy of our verifiable advantage protocol is the security of Conjectures 2 and 3.

1.4 The planted independent space problem for cubic polynomials

The core technical contribution of our work is to define and analyze the complexity of the planted independent space problem. We do this in terms of both worst-case and average-case complexity for different parameter regimes.

Problem 5 (Independent space problem, decision (decision-IndSp))**.** *Given a cubic polynomial $g \in \mathbb{F}_2[x_1, \ldots, x_n]$ and $h \in [n]$, decide if g admits an independent space of dimension h.*

Problem 6 (Independent space problem, search (search-IndSp))**.** *Given a cubic polynomial $g \in \mathbb{F}_2[x_1, \ldots, x_n]$ with the promise that g contains a dimension-h independent space, return a dimension-h independent space.*

For the definition of the average-case version, we consider the following distributions.

- Let $\mathcal{P}_3(n)$ be the uniform distribution over cubic polynomials with constant term 0 in $\mathbb{F}_2[x_1, \ldots, x_n]$. That is, the coefficients of $x_i x_j x_k$, $x_i x_j$, and x_k, are sampled independently and uniformly from $\mathbb{F}_2$.
- Let $\mathcal{P}_3(n, h)$ be the distribution over cubic polynomials obtained by first drawing a uniformly random $g \leftarrow \mathcal{P}_3(n)$ and then planting a uniformly random h-dimensional independent space V. This can be done by first setting the coefficients of $x_i x_j x_k$, $1 \le i < j < k \le h$, to be 0, and then sampling a random $A \in \mathrm{GL}(n, \mathbb{F}_2)$ and outputting the multilinear representation of $g \circ A$, obtained using $x_i^2 = x_i$.

Problem 7 (Planted independent space problem, decision (decision-h-PlantedIndSp))**.** *Given a cubic polynomial g decide whether $g \leftarrow \mathcal{P}_3(n)$ or $g \leftarrow \mathcal{P}_3(n,h)$.*

Problem 8 (Planted independent space problem, search (search-h-PlantedIndSp))**.** *Given a cubic polynomial $g \leftarrow \mathcal{P}_3(n,h)$, find the h-dimensional independent space V.*

In order to exploit the vertex-cover algorithm for computing outcome probabilities of planted circuits, we need to solve the search version of the PlantedIndSp problem, and this also solves the decision version. To give the strongest guarantees, our analysis therefore focuses on the (easier) decision version of the worst- and average-case problems.

Worst-case hardness and search-to-decision reductions We study the complexity of both problems in terms of worst-case hardness as well as in terms of the runtime of concrete algorithms solving either problem. As our first result, we determine the worst-case hardness of the decision version of the IndSp problem.

Theorem 9 (Worst-case hardness)**.** decision-IndSp *is* NP*-complete.*

To prove this result, we show tight equivalences between the maximum independent space of a cubic polynomial, the minimum *vertex cover* of its support, its *strength*, and its *symmetric slice rank*. Using these equivalences, we can reduce to the problem of deciding whether the slice rank of a given trilinear form is less than a given r, which was shown to be NP-hard in [BIL+21].

We remark that the independent set problem admits a natural search-to-decision reduction in the worst-case setting. In the average-case setting, the search-to-decision reduction was recently resolved satisfactorily in [HS24].

The independent space problem also admits a search-to-decision reduction in the worst-case setting, though such a reduction is not as natural as in the independent set case. That is, by Theorem 9, decision-IndSp is NP-hard. As it admits natural polynomial-size witnesses (the transformation matrix in $\mathrm{GL}(n,\mathbb{F}_2)$), it is NP-complete. This allows us to guess the NP-witness (such as a basis of the independent space) bit by bit, and for a partially fixed basis, reduce to the decision-IndSp thanks to its NP-completeness. However, this reduction is not natural as the resulting instance of decision-IndSp may not relate to the original input instance clearly.

The average-case search-to-decision reduction for the independent space problem looks rather non-trivial, and we leave this to future work.

Average-case setting. Second, we constrain the region in which we can hope for a polynomial verification-simulation gap by analyzing the complexity of Problems 7 and 8 in the regime where $h \geq n(1/2+\epsilon)$ for any constant $\epsilon > 0$. In this regime, we give non-trivial algorithms that solve both problems using the rank profile of the Hessian slice matrix evaluated at random vectors.

Theorem 10 (The $h \geq n(1/2+\epsilon)$ regime)**.** *Whenever the planted independent space has dimension $h \geq (1/2+\epsilon)n$, where $\epsilon > 0$ is a constant,* search-h-PlantedIndSp *can be solved in time $\tilde{O}(2^{n-h})$.*

The key observation of the algorithm achieving this runtime is that the Hessian matrix of a cubic form with a large planted independent space typically has a rank deficiency compared to a uniformly random cubic polynomial at any point in the independent space whenever $h \geq (1/2+\epsilon)n$. To distinguish against the uniform case, it is therefore sufficient to find a single element of the independent space. To actually find the independent space, a constant number of such points is sufficient since those will pin down the subspace completely, analogously to how large planted cliques of size $s \gtrsim \sqrt{n}$ can be found quickly once a few of its elements have been found [FR10].

On the other hand, we show that the largest independent space of a random cubic polynomial has dimension $h \leq \sqrt{6n}$ with high probability. This implies that $\mathcal{P}_3(n)$ and $\mathcal{P}_3(n,h)$ are statistically distinguishable for $h > \sqrt{6n}$ as outlined above.

Theorem 11. *For $h > \sqrt{6n}$, there is an algorithm solving* decision-h-PlantedIndSp *in time* $2^{O(n^{1.5})}$.

Based on this, we conjecture that $n/2$ is indeed the computational threshold, yielding Conjecture 2.

Heuristic algorithms and intuition for Conjecture 2. The PlantedIndSp problem is yet another addition to the family of planted structure problems. As mentioned, a classical problem in this family is the planted clique problem, which we reviewed above.

Another related problem is a problem underlying the so-called Unbalanced Oil and Vinegar (UOV) digital signature scheme [Pat96; KPG99]; see [BCD+25] for more recent updates. This problem concerns the following planted structure. Let $\mathsf{M}(n, \mathbb{F}_q)$ denote the set of $n \times n$ matrices over the field $\mathbb{F}_q$. Given symmetric matrices $A_1, \ldots, A_m \in \mathsf{M}(n, \mathbb{F}_q)$, plant a random totally-isotropic space $U \leq \mathbb{F}_q^n$ of dimension d; here, U is totally-isotropic if every A_i, viewed as a symmetric bilinear form on $\mathbb{F}_q^n$, evaluates to 0 on U.

One main approach to attack UOV is to recover this planted totally-isotropic space, and current best algorithms still run in exponential time for $d = m \leq n/3$. This can be done via the Gröbner-basis approach by formulating a system of polynomial equations [BCD+25]. To analyze the performance of Gröbner-basis attacks, one usually resorts to regularity or semiregularity analyses [BFS15]. For h-PlantedIndSp in the regime $h \leq (1/2 - \epsilon)n$ this prediction is $2^{O(n^{3/2})}$ as suggested by the naïve distinguishing algorithm outlined above.

Recently, the planted totally-isotropic space problem was also introduced in another setting in [LQTZ26]. The work [LQTZ26] was motivated by transferring the private simultaneous messages and secret sharing schemes based on planted cliques [ABI+23] to a linear algebraic setting. For a better analogy with planted cliques (or equivalently, independent sets), the planted totally-isotropic problem was introduced for alternating matrices. Recall that $A \in \mathsf{M}(n, \mathbb{F}_q)$ is alternating, if for any $u \in \mathbb{F}_q^n$, $u^t A u = 0$. For $(A_1, \ldots, A_m)$, $A_i \in \mathsf{M}(n, \mathbb{F}_q)$ alternating, the main parameter range considered in [LQTZ26] was when $m = \lceil n/\log n \rceil$, and $d \ll n/2$.

Note that $(A_1, \ldots, A_m)$ naturally defines an alternating bilinear map $\phi : U \times U \to W$, where $U = \mathbb{F}_q^n$ and $W = \mathbb{F}_q^m$. We can also compare the complexity of solving PlantedIndSp for polynomials with PlantedIndSp for alternating bilinear maps $\phi : U \times U \to W$ satisfying $\phi(V, V) = 0$ for a planted subspace $V \leq U$ [LQTZ26]. An alternating bilinear map naturally admits an interpretation as a subspace of alternating matrices $\langle M_k \rangle_{k \in [\dim(W)]}$, where $M_k \in V \times V$ are alternating matrices, or, equivalently, a cubic tensor on $U \times U \times W$. Whereas an h-dimensional independent space of a polynomial corresponds to a basis in which there is an $h \times h \times h$ zero-subcube in the symmetric coefficient tensor, an independent space of a bilinear form corresponds to an $h \times h \times \dim(W)$ *tube* of zeros in the corresponding tensor. This interpretation allows for a relatively simple proof of NP-hardness of finding the planted subspace by embedding a graph into a tensor using linear independence [BCG+19]. Correspondingly, in the bilinear-form case, the PlantedIndSp problem can be solved by asking that $v^T M_k w = 0$ for all $v, w \in V$ and $k \in [\dim(W)]$. In contrast, in the polynomial/symmetric setting we are in, the condition is the stronger

$$u^T M_v w = 0 \quad \text{for all } u, v, w \in V, \tag{6}$$

where $M_v = \sum_k v_k M_k$ is a certain linear combination of the tensor slices. This increase in the degree of the problem makes the naïve solution strategies harder. Indeed, even in the $h \geq (1/2+$

$\epsilon)n$ setting, the best we can do is an $\tilde{O}(2^{n-h})$-time algorithm for solving PlantedIndSp, while in the same scaling scenario, the alternating bilinear PlantedIndSp problem admits a polynomial-time solution [LQTZ26].

1.5 The cubic XEB problem

Let us first consider algorithms for the gap of cubic Boolean polynomials. This corresponds to an amplitude estimation algorithm, which can be used as a subroutine to generate samples. In fact, all known ways to near-exactly sample from IQP circuits involve amplitude estimation as a subroutine [HE23].

Our central piece of evidence for Conjecture 3 is the fact that there is no known algorithm for computing the gaps of Boolean polynomials that uses runtime $o(2^{n(1-\epsilon)})$ for any $\epsilon > 0$ and polynomial space. The naïve and best known polynomial-space algorithm for the Boolean gap just computes the sum of size 2^n by going through all n-bitstrings in Gray-code order using time $O(2^n n^2)$.

If we drop the polynomial space requirement, surprisingly, one can improve the naïve asymptotic runtime $\Theta(2^n)$ to $\tilde{O}(2^{0.9965n})$ [LPT+17; DHKP20]. The key idea of Lokshtanov et al. [LPT+17] is to find polynomial representations of the partial sums

$$F_b(y) \coloneqq \sum_{a\in\{0,1\}^b} (-1)^{f(y,a)}, \tag{7}$$

for $b = (1-c)n$ computing and storing which requires space scaling strictly better than 2^n. Then, in a second step, one can evaluate F_b on all 2^{cn} inputs using the compressed representation of F_b. In upcoming work, we show that this route can be improved significantly, yielding an algorithm with runtime $\tilde{O}(2^{0.552n+O(\sqrt{n})})$ [BHL+26]. However, there remain barriers to reaching runtime $\tilde{O}(2^{n/2})$ for arbitrary or random cubic polynomials using this approach: we would need to find a size-$2^{n/2}$ representation that lets us compute $F_{n/2}$ for every input. It also seems as though this approach to computing the gap intrinsically requires a representation of the functions F_b, and therefore it would not lead to a polynomial-space algorithm. From the other side, the best known MA algorithm for computing the gap has runtime $\tilde{O}(2^{n/2})$ [Wil16]—and uses a significantly stronger model of computation. As we elaborate in the main text, we believe that achieving this runtime in the standard model would be highly surprising. This lends further justification to our polynomial-space conjecture.

An alternative route to disprove Conjecture 3 would be to show that one can pass the XEB using samples from a different distribution that can be simulated more efficiently than the ideal distribution. This is the approach that XEB spoofers for noisy circuits take [BCG21; GKC+24]. However, all known spoofers for the XEB exploit noise in the circuit, and there is no known sub-2^n time classical algorithm that achieves an XEB score close to ideal as required in Conjecture 3. Again, finding such an algorithm would constitute a breakthrough in the theory of XEB.

1.6 Discussion and outlook

Our work opens up several interesting directions of research, both in terms of potential applications and potential improvements.

Application to classically certified randomness An important advantage of our approach to verifiable advantage compared to peaked distributions is that the sample distribution has high entropy. This means that they can potentially be used to generate classically certified randomness, an idea that is due to Aaronson and Hung [AH23]. This yields a useful potential application of our verifiable advantage protocol.

For the generation of cryptographically certified randomness it is important to take into account that an adversarial prover has access to a quantum computer which they can use to perform efficient quantum computations. Using a quantum computer, an adversary can pass XEB in time $O(2^{n/2})$ using amplitude amplification while sampling from a distribution with lower min entropy. Therefore, the security gap collapses for quantum computers with exponential runtime. However, we conjecture that our scheme remains secure against polynomial-time quantum computers, and $O(2^{n/2})$-time and polynomial-space classical computers. We take this to be a reasonable adversary model: The best a polynomial-time quantum adversary can do is to run the true circuits and return the measurement samples to the classical verifier. The best a polynomial-space classical adversary can do is to spend time $\tilde{O}(2^{\kappa n})$ in order to simulate the correct distribution. The verifier will then use her knowledge of the planted subspace to verify the samples (in time $\tilde{O}(2^{cn})$). This is in contrast with the original proposal using random circuits, where generating certifiable randomness required specific timing assumptions and fast response times of the prover [LSN+25; LND+25]. We sketch this application in Section 3.4, leaving a full security analysis to future work.

Towards efficient verification One of the central open questions raised by our work is whether our approach can be extended to achieve polynomial-time verification of quantum advantage, that is, an exponential gap between verification and simulation runtimes. As we find that polynomials with large independent spaces of dimension $h > n/2$ are distinguishable from random ones in time $\tilde{O}(2^{n-h})$ and the secret independent space can be recovered in time $\tilde{O}(2^{n-h})$, our scheme hits a barrier at $h = n/2$. This implies that it can at most yield a polynomial gap between verification and simulation.

Nonetheless, one can think more generally about our approach to plant secret structures in degree-3 circuits that simplify simulation of the circuit. Several directions remain to be explored. First, observe that our approach relied on planting a secret low-rank stabilizer decomposition of the output state. This makes simulation amenable to stabilizer-rank simulators given knowledge of the secret structure. Another natural structure that renders classical simulation more efficient than in the worst case is low entanglement. To see how low entanglement manifests in the polynomial coefficients, consider a cubic polynomial g whose coefficients $G_{ijk} = 0$ whenever $|k-i| > \log n$ and random otherwise. The coefficient tensor G then has a *tube* of nonzero entries along the diagonal, and the corresponding circuit has low depth on a 1D line. We can again obfuscate the corresponding polynomial as $g = f \circ A$ by a linear map $A \in \mathrm{GL}(n, \mathbb{F}_2)$. We leave it to future work to fully assess the feasibility and security of this approach.

As a first step, we show in Section A that block-diagonal structure can always be efficiently recovered. This is a special instance of the low-entanglement structure, where the circuit is unentangled between blocks. This has applications to a related proposal by Bravyi, Gheorghiu, and Temme [BGT25] for the forrelation problem. The forrelation problem is to decide whether

$$F(f, g) \coloneqq \langle f, \hat{g} \rangle \geq 1/\mathsf{poly}(n), \tag{8}$$

given Boolean functions f, g and the Fourier transform

$$\hat{g}(s) = \frac{1}{2^n} \sum_x (-1)^{xs} g(x). \tag{9}$$

This problem can be solved by a quantum computer since

$$F(f, g) = \langle +^n | \, C(g) H^{\otimes n} C(f) \, | +^n \rangle \,, \tag{10}$$

i.e., there is a simple quantum circuit solving the problem. Bravyi, Gheorghiu, and Temme [BGT25] consider the forrelation problem for cubic polynomials $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ with a hidden

product/block structure. They observe that the forrelator F can be efficiently computed classically if the product structure is known, but is classically hard if detecting that structure is hard. Our recovery algorithm rules out this approach to verifiable quantum advantage.

Further directions Another interesting problem with potential practical applications to verifiable advantage is to consider planted structures under different orbits. Here, we extended the independent set problem corresponding to orbits over $S(n)$, to independent spaces which are orbits over $\mathrm{GL}(n, \mathbb{F}_2)$. Further interesting orbits to consider are Clifford circuits that map cubic polynomial phase states to cubic polynomial phase states, as well as families of efficiently computable and invertible Boolean functions.

We restricted ourselves to Boolean polynomials over $\mathbb{F}_2$, but the planted independent space problem is also well defined over arbitrary finite fields $\mathbb{F}_q$. The ideas of several results in this article should carry over to fields of higher order, especially for fields of characteristic > 3 due to the correspondence between symmetric trilinear forms and cubic forms. Still, it would be interesting to examine more carefully how things behave over larger fields.

1.7 Overview

The technical part of this paper is structured as follows. In Section 2 we formally introduce polynomials and symmetric tensors, state some basic properties and introduce the planted independent space problem and state our conjecture regarding it. In Section 3 we then detail the headline application of the exponential hardness of PlantedIndSp, namely, how it can be used for verifiable quantum advantage under an additional conjecture on the exponential difficulty of spoofing the XEB. Then, in Sections 4 and 5 we analyse in detail the worst- and average-case hardness of the PlantedIndSp problem, respectively.

AI Statement

All of the original concepts and proof strategies of this paper are our own. We used generative AI of various generations for assistance in proving lemmas, checking technical correctness, and for feedback on earlier versions of the draft.

Acknowledgements

We are grateful for discussions with Michael Walter and Jeroen Zuiddam. Part of this work was supported by the Simons Institute for the Theory of Computing, and conducted when MB and YQ were visiting the program "Complexity and Linear Algebra". DH was supported by a Simons postdoctoral fellowship through DOE QSA and NSF QLCI Grant No. 2016245, and by the Swiss National Science Foundation through Ambizione Grant No. 223764. MJG acknowledges support from NSF QLCI grant OMA-2120757, NSF QLCI grant OMA-2553574, and NSF NQVL:Design:ORAQL grant 2533041.

2 The polynomial planted independent space problem

2.1 Boolean polynomials and tensors

We will work over the field $\mathbb{F} = \mathbb{F}_2$ and be concerned with cubic Boolean polynomials $f : \mathbb{F}_2^n \to \mathbb{F}_2$ with constant term being 0 as follows:

$$f(x) = \sum_{1 \le i < j < k \le n} F_{ijk} x_i x_j x_k + \sum_{1 \le i < j \le n} F_{ij} x_i x_j + \sum_{1 \le i \le n} F_i x_i, \tag{11}$$

where we understand all addition modulo 2. We decompose the polynomial into unique multilinear degree-3, 2 and 1 forms, that is, $f(x) = f^{(3)}(x) + f^{(2)}(x) + f^{(1)}(x)$. This is natural, as for any $x \in \mathbb{F}_2$, we have $x = x^k$ for any $k \in \mathbb{N}$ and therefore a cubic term of the form $x_i^2 x_j = x_i x_j$ can be reduced to a quadratic term. Throughout, we identify every polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ with its unique multilinear representative in $\mathbb{F}_2[x_1, \ldots, x_n]$, i.e., we work in $\mathbb{F}_2[x_1, \ldots, x_n]/(x_i^2 - x_i)$. When referring to degrees, coefficients and (formal) derivatives we refer to this representative, and we reduce compositions such as $f \circ A$ back to this representative. A homogeneous degree-3 polynomial f is also called a *cubic form*.

In the following, we record some structural results about cubic polynomials over $\mathbb{F}_2$. While these should be known in the literature, we prove them for completeness.

Hessian matrices of cubic polynomials. Let $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a cubic polynomial as in Eq. (11).

$$\partial_i \partial_j f = f(x + e_i + e_j) + f(x + e_i) + f(x + e_j) + f(x). \tag{12}$$

Fact 12. *1. For $i, j \in [n]$, $\partial_i \partial_j f = \partial_j \partial_i f$.*

2. When $i = j$, $\partial_i \partial_j f = 0$.

3. When $i < j$, $\partial_i \partial_j f = \sum_{1 \le k < i} F_{k,i,j} x_k + \sum_{i < k < j} F_{i,k,j} x_k + \sum_{j < k \le n} F_{i,j,k} x_k + F_{i,j}$.

Proof. Items (1) and (2) are straightforward. For (3), it is enough to check those monomials in f. For degree-3 monomials $x_a x_b x_c$ where $1 \le a < b < c \le n$, we distinguish between

- $\{i, j\} \subseteq \{a, b, c\}$. For example, for $M = x_i x_j x_c$, then $M(x + e_i + e_j) + M(x + e_i) + M(x + e_j) + M(x) = (x_i + 1)(x_j + 1)x_c + (x_i + 1)x_j x_c + x_i(x_j + 1)x_c + x_i x_j x_c = x_c$.
- i or j is in $\{a, b, c\}$. For example, for $M = x_i x_b x_c$, then $M(x + e_i + e_j) + M(x + e_i) + M(x + e_j) + M(x) = 0$ as e_j is irrelevant in M.
- Neither i nor j is in $\{a, b, c\}$. In this case the evaluation is zero as above.

Then among quadratic terms, only $x_i x_j$ would contribute 1, and linear terms all contribute to 0. The result then follows. □

Definition 13. *Let $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a cubic polynomial.*

The Hessian of f, H^f, is an $n \times n$ matrix whose (i, j)th entry $\mathrm{H}^f(i, j) = \partial_j \partial_i f(x)$.

Let H_v^f be the alternating matrix whose (i, j)th entry $\mathrm{H}_v^f(i, j) = \partial_j \partial_i f(v)$. More specifically,

$$\mathrm{H}_v^f(i, j) = \partial_j \partial_i f(v) = f(v + e_i + e_j) + f(v + e_i) + f(v + e_j) + f(v). \tag{13}$$

The matrix associated with the quadratic part of f is the $n \times n$ alternating matrix Q^f defined entrywise as $\mathrm{Q}^f(i, j) = \partial_i f^{(2)}(e_j)$, which evaluates to $\mathrm{Q}^f(i, j) = F_{i,j}$ for $1 \le i < j \le n$, $\mathrm{Q}^f(i, j) = F_{j,i}$ for $1 \le j < i \le n$, and $\mathrm{Q}^f(i, i) = 0$ for $i \in [n]$.

Fact 14. *Let f, H^f, and Q^f be from Definition 13. Then $\mathrm{H}^f_{u+v} + \mathrm{Q}^f = (\mathrm{H}^f_u + \mathrm{Q}^f) + (\mathrm{H}^f_v + \mathrm{Q}^f)$. That is, the matrices $\{\mathrm{H}^f_u + \mathrm{Q}^f \mid u \in \mathbb{F}_2^n\}$ form a linear space.*

Proof. By Fact 12, the entries in H^f are linear polynomials whose constant terms are $F_{i,j}$, the entries of the matrix Q^f. As a result, $\mathrm{H}^f_{u+v} - \mathrm{H}^f_u - \mathrm{H}^f_v = \mathrm{Q}^f$. □

From cubic polynomials to trilinear forms Let $T : \mathbb{F}_2^n \times \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2$ be a trilinear form. We say that T is alternating, if whenever two arguments are the same, T evaluates to 0; that is, for any $u, v \in \mathbb{F}_2^n$, $T(u, u, v) = T(u, v, u) = T(v, u, u) = 0$.

Definition 15. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a cubic polynomial as in Eq. (11). The alternating trilinear form $\mathrm{T}^f : \mathbb{F}_2^n \times \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2$ associated with the cubic part of f is obtained by specifying its evaluations on the standard basis vectors $\{e_i\}_{i=1}^n$ and extending by linearity as follows:*

1. *For distinct indices $i < j < k$ and any permutation $\pi \in S_3$, we set $\mathrm{T}^f(e_{\pi(i)}, e_{\pi(j)}, e_{\pi(k)}) = F_{ijk}$.*
2. *If any indices coincide, T^f evaluates to 0.*

With this formalization, we can establish the basic relationship between the cubic polynomial f and its associated trilinear form T^f via the following identity.

Lemma 16. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a cubic polynomial and T^f be its associated alternating trilinear form as in Definition 15. Then for any vectors $u, v, w \in \mathbb{F}_2^n$,*

$$\mathrm{T}^f(u, v, w) = f(u + v + w) + f(u + v) + f(v + w) + f(u + w) + f(u) + f(v) + f(w). \tag{14}$$

Proof. We first consider the cubic monomials. In Equation 14,

$$\begin{aligned} RHS = \sum_{1 \le a < b < c \le n} F_{abc}\Big(& (u_a + v_a + w_a)(u_b + v_b + w_b)(u_c + v_c + w_c) + \\ & (u_a + v_a)(u_b + v_b)(u_c + v_c) + \\ & (v_a + w_a)(v_b + w_b)(v_c + w_c) + \\ & (u_a + w_a)(u_b + w_b)(u_c + w_c) + \\ & u_a u_b u_c + v_a v_b v_c + w_a w_b w_c \Big). \end{aligned}$$

For a fixed unordered set $\{a, b, c\}$, we expand the expression and count the terms based on the number of components they contain from the vector u:

- The term $u_a u_b u_c$ appears in the expansion of $(u_a + v_a + w_a)(u_b + v_b + w_b)(u_c + v_c + w_c)$ once; appears in the expansion of $(u_a + v_a)(u_b + v_b)(u_c + v_c)$ once; appears in the expansion of $(u_a + w_a)(u_b + w_b)(u_c + w_c)$ once; appears in the expansion of $u_a u_b u_c$ once. It occurs exactly $4 \equiv 0 \pmod 2$ times and cancels out completely.
- The term $u_a u_b v_c$ (which contains two components of u) appears in the expansion of $(u_a + v_a + w_a)(u_b + v_b + w_b)(u_c + v_c + w_c)$ once; appears in the expansion of $(u_a + v_a)(u_b + v_b)(u_c + v_c)$ once. It occurs exactly $2 \equiv 0 \pmod 2$ times and cancels out completely.
- The term $u_a v_b w_c$ (containing exactly one u, one v, and one w component) only appears in $(u_a + v_a + w_a)(u_b + v_b + w_b)(u_c + v_c + w_c)$.

Therefore, after cancellation, we have

$$RHS = \sum_{a<b<c} F_{abc}(u_a v_b w_c + u_a w_b v_c + v_a u_b w_c + v_a w_b u_c + w_a u_b v_c + w_a v_b u_c).$$

On the other hand,

$$LHS = \mathrm{T}^f(u, v, w) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} u_i v_j w_k \, \mathrm{T}^f(e_i, e_j, e_k).$$

Since $\mathrm{T}^f(e_i, e_j, e_k)$ is non-zero only if the indices are mutually distinct, the triple summation essentially iterates over all triplets of distinct indices, summing the values over their $3! = 6$ permutations. By rewriting the left-hand side using the subset condition $a < b < c$, we obtain:

$$\mathrm{T}^f(u, v, w) = \sum_{a<b<c} F_{abc} \left(\sum_{\pi \in S_3} u_{\pi(a)} v_{\pi(b)} w_{\pi(c)} \right).$$

We then deal with quadratic and linear terms on the right-hand side of Equation 14, and it can be verified that these contribute 0, which concludes the proof. □

Theorem 17. *Let $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a cubic polynomial. For $u \in \mathbb{F}_2^n$, let $\mathrm{T}_u^f : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2$ be defined as $\mathrm{T}_u^f(v, w) = \mathrm{T}^f(u, v, w)$. Then $\mathrm{T}_u^f = \mathrm{H}_u^f + \mathrm{Q}^f$.*

Proof. By substituting $v = e_i$ and $w = e_j$ into Equation 14, we obtain

$$T^f(u, e_i, e_j) = f(u + e_i + e_j) + f(u + e_i) + f(e_i + e_j) + f(u + e_j) + f(u) + f(e_i) + f(e_j).$$

We first note that in either the above equation or the definitions of H_u^f and Q^f, the linear term vanishes by linearity over $\mathbb{F}_2$. This leaves us to consider only cubic and quadratic terms, which in turn indicate that vectors of weights ≤ 1 evaluate to zero. Therefore, the (i, j)th entry of T_u^f is

$$\mathrm{T}^f(u, e_i, e_j) = f(u + e_i + e_j) + f(u + e_i) + f(e_i + e_j) + f(u + e_j) + f(u).$$

Now note that $f(u + e_i + e_j) + f(u + e_i) + f(u + e_j) + f(u)$ is equal to the (i, j)th entry of H_u^f, and $f(e_i + e_j)$ is equal to the (i, j)th entry of Q^f. This concludes the proof. □

Proposition 18. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a cubic polynomial. For $A \in \mathrm{GL}(n, \mathbb{F}_2)$, let $g = f \circ A$. Then we have the following.*

1. $(f \circ A)^{(3)} = (f^{(3)} \circ A)^{(3)} = g^{(3)}$.
2. *For all $u, v, w \in \mathbb{F}_2^n$, $\mathrm{T}^g(u, v, w) = \mathrm{T}^f(Au, Av, Aw)$.*
3. *As matrices of bilinear forms, $\mathrm{T}_u^g = A^t \, \mathrm{T}_{Au}^f \, A$. Consequently,*

$$\mathrm{H}_u^g + \mathrm{Q}^g = A^t(\mathrm{H}_{Au}^f + \mathrm{Q}^f)A.$$

Proof. For (1), this is because the degree-3 part of $f \circ A$ only depends on $f^{(3)}$. For (2), this is due to Lemma 16, which expresses T^f as evaluations by f. (3) is a consequence of (2) and Theorem 17. □

Remark 19 (Tensor Slices Perspective)**.** *The reason for introducing* T^f *is that a trilinear form allows for specializing one argument to obtain bilinear forms, where ranks and normal forms are usually well-studied. This strategy is in line with associating quadratic forms with bilinear forms in the case of field characteristic not* 2*. It can be implemented relatively easily for trilinear forms over fields of characteristic not* 2 *or* 3*: for a cubic form* f*, the associated symmetric trilinear form is defined as*

$$\mathrm{T}^f(u,v,w) = \frac{1}{6}\big(f(u+v+w) - f(u+v) - f(v+w) - f(u+w) + f(u) + f(v) + f(w)\big).$$

What we observed in the above is that over $\mathbb{F}_2$*, a cubic polynomial can be associated with an alternating trilinear form defined in Definition 15 or alternatively by Equation 14 which essentially captures the cubic coefficients of* f *and is unaffected by its quadratic and linear terms.*

2.2 Planted independent space

In Section 1.2, independent spaces were defined using independent sets after a linear change of variables, where independence concerns only the cubic terms. The following proposition gives an equivalent characterization in terms of the associated alternating trilinear form.

Proposition 20. *Let* $g : \mathbb{F}_2^n \to \mathbb{F}_2$ *be a cubic polynomial and let* $V \leq \mathbb{F}_2^n$*. Then* V *is an independent space of* g*, as defined in Definition 1, if and only if* $\mathrm{T}^g|_{V^3} \equiv 0$*, i.e.,* $\mathrm{T}^g(u,v,w) = 0$ *for all* $u,v,w \in V$*, or equivalently,* $\deg(g|_V) \leq 2$*.*

Proof. Let $d = \dim V$. By Definition 1, choose $A \in \mathrm{GL}(n,\mathbb{F}_2)$ whose first d columns form a basis of V. In the multilinear representation of $g \circ A$, the coefficient of $x_i x_j x_k$, for $1 \leq i < j < k \leq d$, is

$$\mathrm{T}^{g\circ A}(e_i,e_j,e_k) = \mathrm{T}^g(Ae_i, Ae_j, Ae_k).$$

The first d coordinates form an independent set of $g \circ A$ if and only if all these coefficients vanish. Since $Ae_1,\ldots,Ae_d$ form a basis of V, and T^g is alternating and trilinear, this is equivalent to $\mathrm{T}^g|_{V^3} \equiv 0$. □

We further say that V is a *maximum independent space* if

$$V \in \arg\max_{W \leq \mathbb{F}_2^n : \mathrm{T}^g|_{W^3}=0} \dim(W) \tag{15}$$

We use $\alpha(g)$ to denote the maximum dimension over independent spaces of g.

Independent spaces have a natural connection to independent sets in the hypergraph associated with a polynomial.

Definition 21 (Polynomial hypergraph)**.** *Given a cubic polynomial* $f \in \mathbb{F}_2[x_1,\ldots,x_n]$*, the hypergraph* $H(f) = (V,E)$ *is defined by the trilinear coefficients* F_{ijk} *as follows: Let* $V = [n]$ *be the vertex set, and let the set of hyperedges be* $E = \{\{i,j,k\} : F_{ijk} = 1, i < j < k\}$*.*

Remark 22 (Independent set)**.** *Consider a cubic polynomial* $f \in \mathbb{F}_2[x_1,\ldots,x_n]$*. For* $I \subseteq [n]$*,* I *is a (strong) independent set of* $H(f)$ *if and only if* $\mathrm{span}\{e_i : i \in I\}$ *is an independent space of* f*.*

Moreover, notice that finding a maximum independent set is equivalent to finding a maximum clique in the complement graph and therefore computationally equivalent.

Random cubic polynomials. A random cubic polynomial in $\mathbb{F}_2[x_1, \ldots, x_n]$ is sampled by sampling the coefficient of every term $x_i x_j x_k$ $(1 \le i < j < k \le n)$, $x_i x_j$ $(1 \le i < j \le n)$, x_i $(i \in [n])$ independently uniformly from $\mathbb{F}_2$.

Lemma 23. *Let $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a random cubic polynomial. Then $\alpha(f) \le \lfloor \sqrt{6n} \rfloor$ with high probability.*

Proof. This is a standard probabilistic method. For a dimension-d space $U \le \mathbb{F}_2^n$, the probability of U being an independent space for a random f is $1/2^{\binom{d}{3}}$. The number of d-dimensional spaces of $\mathbb{F}_2^n$ is upper bounded by $2^{d(n-d)+2}$. By a union bound, over random choices of f,

$$\Pr\left[\exists U \le \mathbb{F}_2^n, \dim(U) = d, T^f|_{U^3} = 0\right] \le \frac{2^{d(n-d)+2}}{2^{\binom{d}{3}}} = \frac{1}{2^{\frac{d(d-1)(d-2)}{6} - d(n-d) - 2}}.$$

Note that when $d \ge \lceil \sqrt{6n} \rceil$, $d(d-1)(d-2)/6 - d(n-d) - 2 \ge d^2/2 \ge 3n$, so

$$\Pr\left[\exists U \le \mathbb{F}_2^n, \dim(U) = d, T^f|_{U^3} = 0\right] \ll 1/2^{3n},$$

showing the existence of f without dimension-d independent spaces. □

2.3 Statement of independent space problems

Recall the distributions $\mathcal{P}_3(n)$ and $\mathcal{P}_3(n, h)$, as well as the decision and search problems introduced in Section 1.4.

We now describe the planted sampling procedure for $\mathcal{P}_3(n, h)$ in detail. Then in this section, we formally specify the planted independent space problems and discuss the relations between these problems.

Fix an integer $1 \le h \le n$ and let $U = \mathrm{span}\{e_1, \ldots, e_h\}$. We first sample a cubic Boolean polynomial $f \leftarrow \mathcal{P}_3(n)$. We then plant an independent space in its cubic component by setting the coefficient of $x_i x_j x_k$ to zero whenever $1 \le i < j < k \le h$, leaving all other coefficients unchanged. Write the resulting polynomial as $\widetilde{f} = c_0 + r_0$, where c_0 is its homogeneous cubic component and r_0 is its independently sampled quadratic and linear part. Thus, $\mathrm{T}^{\widetilde{f}}|_{U^3} = 0$.

To hide this coordinate subspace, we independently sample $A \leftarrow \mathrm{GL}(n, \mathbb{F}_2)$ uniformly and set

$$g(x) = \widetilde{f}(Ax), \qquad V = A^{-1}U,$$

and then reduce g to its unique multilinear representation using $x_i^2 = x_i$. Then V is a uniformly random h-dimensional subspace satisfying $\mathrm{T}^g|_{V^3} = 0$, or equivalently, $\deg(g|_V) \le 2$. We denote the resulting distribution of g by $\mathcal{P}_3(n, h)$.

The linear substitution may produce quadratic and linear terms from c_0. These terms are added to $r_0 \circ A$, and all resulting coefficients are retained in the public polynomial g. Since composition with A is a bijection on the space of Boolean polynomials of degree at most two with zero constant term, $r_0 \circ A$ is uniformly distributed even after fixing c_0 and A. Adding the lower-degree contribution from $c_0 \circ A$ preserves this uniformity. Consequently, the final lower-degree part $g^{(2)} + g^{(1)}$ is uniformly random and independent of the pair $(g^{(3)}, V)$.

The public instance consists of n, h, and the full coefficient description of g, while V and A are kept secret. The following computational problems associated with finding maximum planted independent spaces are natural generalizations of finding maximum or planted independent sets or cliques.

Problem 24 (Independent space problem (decision-IndSp))**.** *Given a cubic Boolean polynomial g and an integer h, decide whether g has an independent space of dimension at least h.*

Next, we introduce the distributional version of this problem, where one is asked to distinguish between a random polynomial and one with a planted independent space.

Problem 25 (Planted independent space problem (decision-h-PlantedIndSp))**.** *Given a cubic Boolean polynomial g decide whether $g \leftarrow \mathcal{P}_3(n)$ or $g \leftarrow \mathcal{P}_3(n,h)$, promised one is the case.*

Finally, we define the associated search problems, which ask to *find* an independent space of a given dimension h.

Problem 26 (Independent subspace problem (search-IndSp))**.** *Given a cubic Boolean polynomial g with an independent space V of dimension h, find V.*

Problem 27 (Planted independent space problem (search-h-PlantedIndSp))**.** *Given a cubic Boolean polynomial $g \leftarrow \mathcal{P}_3(n,h)$, find the h-dimensional independent space V.*

Our central problem is the search-h-PlantedIndSp problem, and there are some obvious relations between the hardness of these problems

$$\text{decision-}h\text{-PlantedIndSp} \leq_p \text{search-}h\text{-PlantedIndSp} \leq_p \text{search-IndSp} \tag{16}$$

$$\text{decision-}h\text{-PlantedIndSp} \leq_p \text{decision-IndSp} \leq_p \text{search-IndSp}, \tag{17}$$

where $\leq_p$ represents polynomial-time reductions. However, it remains an interesting open problem whether there are search-to-decision reductions for the independent space problems, analogous to those for planted clique [HS24].

Hardness conjecture The basis of our cryptographic applications of the PlantedIndSp problem will be the following conjecture about the computational complexity of finding a large planted independent space.

Conjecture 28 (Exponential planted independent space (ExpPlantedIndSp) conjecture)**.** *For any $\epsilon > 0$ and $h \leq (1/2 - \epsilon)n$, every classical algorithm $\mathcal{A}$ that outputs 1 for planted instances and 0 for uniform instances and has runtime $o(2^n)$ satisfies*

$$\left| \Pr_{g \leftarrow \mathcal{P}_3(n,h)}[\mathcal{A}(g,h) = 1] - \Pr_{g \leftarrow \mathcal{P}_3(n)}[\mathcal{A}(g,h) = 1] \right| = \text{negl}(n), \tag{18}$$

where the probabilities include the internal randomness of $\mathcal{A}$.

We provide evidence for Conjecture 28 in Sections 4 and 5 below by analysing its worst- and average-case complexity in different parameter regimes. We note that in fact we believe the PlantedIndSp problem to be exponentially hard even for quantum computers. This is relevant to our envisioned application of planted independent spaces for classically certifiable randomness. We leave a more detailed analysis to future work, however.

3 Verification of IQP circuit sampling from planted independent spaces

In this section, we introduce the family of IQP circuits, and show how the planted independent space conjecture allows us to obtain a verifiable quantum advantage scheme. We first introduce IQP circuits and their statistical properties in Section 3.1 and then show how the XEB of outcome samples of IQP circuits is used to verify quantum advantage in Section 3.2. In Section 3.3 we show how planted independent spaces are *simulation secrets* that simplify verification. Finally, in Section 3.4, we sketch an application of planted IQP circuit sampling to classically certifiable randomness.

3.1 IQP circuits and their statistical properties

Let $f(x) \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a cubic Boolean polynomial as in Eq. (11). The associated degree-3 *IQP circuit* $C(f)$ is defined by its action on basis states as

$$C(f) \left|x\right\rangle = (-1)^{f(x)} \left|x\right\rangle . \tag{19}$$

Degree-3 IQP circuits are therefore diagonal quantum circuits composed of Z, CZ, and CCZ gates acting on the qubits specified in the coefficients F_i, F_{ij}, and F_{ijk}, respectively. Applying an IQP circuit C_f to an initial $\left|+^n\right\rangle$ state yields the *polynomial phase state*

$$\left|f\right\rangle \coloneqq C(f) \left|+^n\right\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} \left|x\right\rangle . \tag{20}$$

The amplitudes of $\left|f\right\rangle$ in the Hadamard basis

$$a_f(x) \coloneqq \left\langle x\right| H^{\otimes n} \left|f\right\rangle , \tag{21}$$

yield the corresponding probabilities $p_f(x) = a_f(x)^2$ of observing x when measuring in this basis.

IQP circuits have been extensively studied in the context of quantum advantage demonstrations. This is because they are a subuniversal family of quantum circuits whose output distributions are yet hard to simulate classically assuming plausible complexity-theoretic conjectures [SB09; BJS10; BMS16; BMS17; HKB+25]. The key observations leading to this result are that the output amplitudes $a_f(x) = \mathrm{ngap}(f_x)$ are given by the gap of the Boolean function $f_x(y) = f(y) + \langle x, y\rangle$, defined as

$$\mathrm{ngap}(f) \coloneqq \frac{1}{2^n} \left(\left|\{x : f(x) = 0\}\right| - \left|\{x : f(x) = 1\}\right|\right) . \tag{22}$$

Computing $a_f(x)$ up to a constant relative error is therefore a GapP-hard problem, where GapP is the class of counting problems that computes the difference of two #P functions [GG17].

Theorem 29 ([BMS16])**.** *Let f be a uniformly random degree-3 polynomial. Then there exists $\epsilon = O(1)$ such that producing samples from any distribution q that is ϵ-close in total-variation distance to p_f is classically intractable assuming that*

i. the polynomial hierarchy does not collapse to its third level, and

ii. the outcome probabilities $p_f(x)$ are GapP-hard to approximate up to constant relative error for a constant fraction of the f-instances.

While condition i. is widely believed in complexity theory [AB09], condition ii.—while commonly regarded as highly plausible—is more specific to the literature on quantum advantage. The arguments for its truth are reviewed in detail in Ref. [HE23]. In short, they boil down to: (a) there is no exploitable structure in a random degree-3 polynomial that would simplify the computation compared to the worst case, and (b) a large fraction of the outcome probabilities are comparable in size, a property known as anticoncentration,

$$\mathop{\mathbb{E}}_f \left[\sum_x p_f(x)^2\right] \leq \frac{\alpha}{2^n}, \tag{23}$$

so that roughly the same absolute error must be achieved on a large fraction of the probabilities.

Since anticoncentration also plays an important role in verification, an important fact about IQP circuits is going to be the first few moments of their output distributions.

- Let $\mathcal{P}_d(n)$ be the uniform distribution over polynomials with degree at most d in $\mathbb{F}_2[x_1, \ldots, x_n]$ with constant term 0, i.e., the coefficient of each multilinear monomial of degree between 1 and d is sampled independently and uniformly from $\mathbb{F}_2$.
- Let $\mathcal{M}(n)$ be the distribution over cubic Boolean polynomials in $\mathbb{F}_2[x_1, \ldots, x_n]$ obtained by defining $f = g + f^{(3)}$, where $g \leftarrow \mathcal{P}_2(n)$ and $f^{(3)}$ is drawn from an arbitrary distribution $\mathcal{D}$ over cubic polynomials.

Lemma 30 (Moment bounds for IQP circuits)**.** *For $k = 1, 2, 3$, the normalized moments of $\mathcal{M}(n)$ are bounded as*

$$m_k \coloneqq 2^{nk} \mathop{\mathbb{E}}_{f \leftarrow \mathcal{M}(n)} p_f^k(0) = (2k-1)!! + o(1), \tag{24}$$

where $(2k-1)!! = 1 \cdot 3 \cdot 5 \cdots (2k-1)$.

Proof. Consider the special case that $\mathcal{D}$ is the point measure on a fixed cubic form $f^{(3)}$, i.e., a cubic polynomial with no linear or quadratic terms. We follow Eq. (42) of Dalzell et al. [DHKP20] and find that

$$\begin{aligned}
2^{2nk} \mathop{\mathbb{E}}_{f \leftarrow \mathcal{M}(n)} \left[a_f(x)^{2k} \right] &= \sum_{x^1, \ldots, x^{2k}} \mathop{\mathbb{E}}_{g \leftarrow \mathcal{P}_2(n)} (-1)^{(g+f^{(3)})(x^1) + \ldots + (g+f^{(3)})(x^{2k})} && (25)\\
&= \sum_{x^1, \ldots, x^{2k}} (-1)^{f^{(3)}(x^1) + \ldots + f^{(3)}(x^{2k})} \mathop{\mathbb{E}}_{g \leftarrow \mathcal{P}_2(n)} (-1)^{g(x^1) + \ldots + g(x^{2k})} && (26)\\
&= \sum_{x^1, \ldots, x^{2k}} (-1)^{f^{(3)}(x^1) + \ldots + f^{(3)}(x^{2k})} \prod_{i,j=1}^{n} \mathbb{1}(x_i^1 x_j^1 + \ldots + x_i^{2k} x_j^{2k} = 0) && (27)\\
&= \sum_{x^1, \ldots, x^{2k}} \prod_{i,j=1}^{n} \mathbb{1}(x_i^1 x_j^1 + \ldots + x_i^{2k} x_j^{2k} = 0) && (28)\\
&= 2^{2nk} \mathop{\mathbb{E}}_{g \leftarrow \mathcal{P}_2(n)} \left[a_g(x)^{2k} \right], && (29)
\end{aligned}$$

i.e., the first three moments of $\mathcal{M}(n)$ are given by the moments of $\mathcal{P}_2(n)$.

To see why Eq. (28) holds, write the matrix $X = (x_i^l)_{l,i}$, and observe that the product of indicators enforces that $X(i)X(j) = 0$ for all $i, j \in [n]$, that is, all columns of X are orthogonal and have even Hamming weight. We can then write

$$\sum_{l=1}^{2k} f^{(3)}(x^l) = \sum_{l=1}^{2k} \sum_{i<j<k} a_{ijk} x_i^l x_j^l x_k^l =: \sum_{i<j<k} a_{ijk} \langle X(i), X(j), X(k) \rangle, \tag{30}$$

where we have defined $\langle u, v, w \rangle = \sum_{l=1}^{m} u_l v_l w_l \mod 2$ for $u, v, w \in \mathbb{F}_2^m$ and notice that $\langle u, v, w \rangle = |\operatorname{supp}(u) \cap \operatorname{supp}(v) \cap \operatorname{supp}(w)| \mod 2$. We now observe that any self-orthogonal subspace $C \leq \mathbb{F}_2^m$ such that $C \subset C^\perp$ with $m \leq 6$ satisfies

$$\langle u, v, w \rangle \equiv 0. \tag{31}$$

To prove this, we partition the set $[m]$ into the seven disjoint regions $t = \operatorname{supp}(u) \cap \operatorname{supp}(v) \cap \operatorname{supp}(w)$, $p_{uv} = \operatorname{supp}(u) \cap \operatorname{supp}(v) \setminus t$, and $s_u = \operatorname{supp}(u) \setminus t \setminus p_{uv} \setminus p_{vw} \setminus p_{uw}$. Self-orthogonality implies

$$\begin{aligned}
\langle u, v \rangle = |t| + |p_{uv}| = 0 \quad &\Rightarrow \quad |p_{uv}| = |t| && (32)\\
\langle u, u \rangle = |t| + |p_{uv}| + |p_{uw}| + |s_u| = 0 \quad &\Rightarrow \quad |s_u| = 3t, && (33)
\end{aligned}$$

and therefore all regions have parity $|t|$. But for $|t| = \langle u, v, w\rangle$ to be odd, $m \geq 7$ is necessary.

We can now apply the bounds from Theorem 8 by Dalzell et al. [DHKP20], using their subsequent observation for degree-2 polynomials, to obtain the claimed result.

The generalization to arbitrary distributions $\mathcal{D}$ over cubic polynomials is immediate: for arbitrary cubic polynomials, just decompose them into the cubic and the quadratic plus linear terms. The uniform measure $\mathcal{P}_2(n)$ is invariant under addition of an arbitrary quadratic polynomial. For arbitrary distributions over cubic polynomials, just apply the argument above pointwise. □

In particular, Lemma 30 implies that

$$m_2 = 3 + o(1), \quad \text{and} \quad m_3 = 15 + o(1). \tag{34}$$

Finally, we note that for uniformly random cubic polynomials, Dalzell et al. [DHKP20] proved that all moments match those of random functions. We state a slightly more precise version of the theorem.

Lemma 31 ([DHKP20])**.** *For all $n, k \in \mathbb{N}$*

$$\left| 2^{nk} \mathop{\mathbb{E}}_{f\sim\mathcal{P}_3(n)} \mathrm{ngap}(f)^{2k} - (2k-1)!! \right| \leq C_k 2^{-n}, \tag{35}$$

where $C_k = (2k-1)!!\, 2^k + k\, 2^{k^2}$.

Proof sketch. The proof is verbatim that of [DHKP20, Theorem 8], and only needs to make the k-dependence explicit. The only implicit constant is c'_k appearing in the last paragraph of the proof, the number of subspaces $H \leq \mathbb{F}_2^{2k}$ of dimension $d < k$ satisfying the closure condition $H^\times \subseteq H^\perp$ (see [DHKP20] for the definitions of $H^\times, H^\perp$), which the original proof only asserts to "depend on k." In particular, every H lies in the even-weight subspace of $\mathbb{F}_2^{2k}$, which has dimension $2k-1$. We bound c'_k crudely by the total number of such even-weight subspaces of $\mathbb{F}_2^{2k}$ of dimension $d < k$, via the Gaussian binomial estimate

$$\binom{2k-1}{d}_2 \leq 2^{d(2k-d)} \leq 2^{k^2-1} \quad \text{for } d < k, \qquad \text{so} \qquad c'_k \leq \sum_{d=0}^{k-1} \binom{2k-1}{d}_2 \leq k \cdot 2^{k^2-1}. \tag{36}$$

Inserting this into the original error accounting with the $d = k$ correction at most $(2k-1)!! \cdot 2^k 2^{(k-1)n}$ and the $d < k$ terms contributing at most $c'_k 2^{(k-1)n}$ shows the claim. □

Lemma 32 (Min-entropy bounds for random IQP circuits)**.** *We have the min-entropy bounds*

$$\Pr_{f\sim\mathcal{M}(n)} \left[H_\infty(p_f) \geq 2n/3 - O(\log(1/\delta))\right] \geq 1 - \delta \tag{37}$$

$$\Pr_{f\sim\mathcal{P}_3(n)} \left[H_\infty(p_f) \geq n - O(\sqrt{n})\right] \geq 1 - \exp(-n). \tag{38}$$

Proof. We show the lemma using a general min-entropy bound based on moments. We follow the proof of Lemma 5 in Ref. [HKEG19] with the second moment replaced by the k-th. We then use that $H_\infty(p) \geq \frac{k-1}{k} H_k(p)$, where

$$H_\alpha(p) \coloneqq -\frac{1}{\alpha - 1} \log\left(\sum_{x\in\mathbb{F}_2^n} p(x)^\alpha \right) \tag{39}$$

is the α-Rényi entropy. This gives us that for any $k, n \in \mathbb{N}$, given any distribution $\mathcal{D}(n)$ over n-bit Boolean polynomials, with probability at least $1-\delta$ over the choice of $f \sim \mathcal{D}(n)$,

$$H_\infty(p_f) \geq \frac{1}{k}\left(\log\delta - \log \mathop{\mathbb{E}}_{f\sim\mathcal{D}(n)}\left[\sum_{x\in\mathbb{F}_2^n} p_f(x)^k\right]\right).$$

For the first bound, we choose $k=3$ and use Lemma 30, for the second, we let $k=\lfloor\sqrt{n}\rfloor$, $\delta=\exp(-n)$ and use Lemma 31. □

3.2 Verification via XEB

Our verifiable quantum advantage scheme is based on sampling from the output distribution of IQP circuits randomly drawn from some distribution $\mathcal{M}(n)$ over cubic polynomials as in Lemma 30. Our quantum advantage test is based on the so-called *cross-entropy benchmark (XEB)* defined for two distributions $q, p : \mathbb{F}_2^n \to [0,1]$ as follows:

$$\chi(q,p) \coloneqq 2^n \sum_x q(x)p(x) - 1, \tag{40}$$

where we think of q as the sampled distribution and p as the target distribution. The XEB can be estimated empirically, given samples $x_1,\ldots,x_\ell \leftarrow q$ as

$$\hat{\chi}(x_1,\ldots,x_\ell) = 2^n \cdot \frac{1}{\ell}\sum_{i=1}^{\ell} p(x_i) - 1. \tag{41}$$

We can further define the average XEB with respect to the measure $\mathcal{M}(n)$ as

$$\overline{\chi} \coloneqq \mathop{\mathbb{E}}_{f\leftarrow\mathcal{M}(n)} \chi(q_f, p_f), \tag{42}$$

where for every target distribution p_f, we consider an actually sampled distribution q_f. Again, the average XEB can be estimated given samples $x_1^f,\ldots,x_\ell^f \leftarrow q_f$ for $f = f_1,\ldots,f_L$ sampled from $\mathcal{M}(n)$ as

$$\hat{\overline{\chi}}(\{x_i^{f_j}\}_{ij}) = 2^n \cdot \frac{1}{\ell L}\sum_{j=1}^{L}\sum_{i=1}^{\ell} p_{f_j}(x_i^{f_j}) - 1. \tag{43}$$

It is easy to see that

$$\overline{\chi} = \begin{cases} 0 & \text{if } \forall f : q_f = \mathbb{1}/2^n \\ m_2 - 1 & \text{if } \forall f : q_f = p_f, \end{cases} \tag{44}$$

and therefore the average XEB serves as a bona-fide test that distinguishes uniform samples from ideal samples with ideal score $\overline{\chi} = 2 + o(1)$ by Eq. (34). In fact, it has been conjectured that this is already sufficient for a quantum advantage test in that no classical polynomial-time algorithm will be able to achieve a high average XEB value for random quantum circuits [AC17; AG20]. We formalize this task as follows.

Task 33 ($\mathsf{XEB}_{b,\ell}$)**.** *Given polynomials $f_1,\ldots,f_L \in \mathbb{F}_2[x_1,\ldots,x_n]$, output samples $x_1^{f_1},\ldots,x_\ell^{f_L}$ satisfying*

$$\hat{\overline{\chi}}(\{x_i^{f_j}\}_{i\in[\ell],j\in[L]}) \geq b. \tag{45}$$

We state here the hardness of achieving high XEB for random cubic polynomials as a conjecture analogous to the arguments of Aaronson and Gunn [AG20][1].

Conjecture 34 (Computational soundness of XEB)**.** *Let $f_1, \ldots, f_L$ be drawn independently from $\mathcal{P}_3(n)$, and let $L, \ell = \mathsf{poly}(n)$. Then there is no $\mathsf{poly}(n, L, \ell)$-time classical algorithm that solves $\mathsf{XEB}_{1,\ell}(f_1, \ldots, f_L)$ with probability at least $1 - 1/\mathsf{poly}(L)$ over the choice of polynomials and the algorithm's randomness.*

Moreover, for completeness of the XEB test, we would like to ensure that a sampler from the ideal distribution p_f indeed passes this test with high probability for sufficiently large L, ℓ. To see why this is true, consider the estimator obtained for $\ell = 1$.

Lemma 35 (Completeness for the ideal sampler)**.** *For $\ell = 1$ and $q_f = p_f$, $\hat{\overline{\chi}} \geq 1$ with probability at least $1 - 10/L$.*

Proof. Let $Z = 2^n p_f(X) - 1$ with $f \sim \mathcal{M}(n)$ and $X \sim p_f$. Then $\mathbb{E}[Z] = \overline{\chi}$. Using the fact that $\mathcal{M}(n)$ is invariant under additions of arbitrary linear terms, which correspond to the different outcomes (the hiding property), the single-shot variance is given by

$$\mathrm{Var}[Z] = 2^{3n} \mathop{\mathbb{E}}_{f \leftarrow \mathcal{M}(n)} p_f(x)^3 - \left(2^{2n} \mathop{\mathbb{E}}_{f \leftarrow \mathcal{M}(n)} p_f(x)^2 \right)^2 = m_3 - m_2^2 = 6 + o(1), \tag{46}$$

and therefore $\mathrm{Var}[\hat{\overline{\chi}}] = (m_3 - m_2^2)/L$. By Chebyshev's inequality (choosing $\epsilon = m_2 - 2$) we therefore find

$$\Pr\left[\hat{\overline{\chi}} < 1\right] \leq \frac{\mathrm{Var}[\hat{\overline{\chi}}]}{(m_2 - 2)^2}, \tag{47}$$

which completes the proof. □

3.3 Simulating IQP circuits with and without planted independent spaces

In this section, we will establish that knowledge of a large (planted) independent space in a polynomial f as in Eq. (11) reduces the complexity of (strongly) simulating the corresponding IQP circuit in the sense of computing the outcome amplitudes a_f up to exponentially small error. This is the meaningful task to consider, since all known sampling algorithms for random IQP circuits involve computing the outcome amplitudes as a subroutine [HE23].

Simulating random polynomials Let us first consider the complexity of simulating IQP circuits corresponding to random cubic polynomials. The best known algorithm has been shown by Dalzell et al. [DHKP20] building on the breakthrough results of Lokshtanov et al. [LPT+17]. Surprisingly, this algorithm achieves a runtime that is asymptotically better than 2^n. In other words, to compute the gap of a Boolean polynomial, one need not naïvely evaluate the polynomial on all inputs.

Theorem 36 (Asymptotic complexity [DHKP20])**.** *There is an algorithm that computes* $\mathrm{gap}(f)$ *for a cubic polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ in time $\tilde{O}(2^{0.9965n})$.*

[1]Strictly speaking, Aaronson and Gunn [AG20] reduce the hardness of XEB to a task they dub XQUATH, which shall not concern us here, since its hardness is a stronger conjecture and has been proven false for certain low-depth settings, where achieving high XEB is conceivably still hard [GKC+24; AGL+23].

The central idea underlying this algorithm is the observation that the partial sum

$$F_b(y) = \sum_{x \in \mathbb{F}_2^b} (-1)^{f(y,x)}, \tag{48}$$

can be expressed as a low-degree polynomial modulo a certain power of 2. The running time depends on the cost of constructing this modular representation and recovering the partial sums from it. The polynomial for $b = cn$ for some $c > 0$ can then be evaluated in time strictly less than $2^{(1-c)n}$ and therefore the overall sum $\sum_y F_b(y)$ can be evaluated in time less than 2^n. The fundamental limit of this variable-splitting approach appears to be runtime $\tilde{O}(2^{n/2})$ since the best one can hope for is to choose $b = n/2$, i.e., equipartition the variables, and find an algorithm that computes $F_{n/2}(y)$ for each y in time at most $\tilde{O}(2^{n/2})$ [DHKP20]. This limit is achieved by an MA algorithm for computing the gap due to [Wil16].

In upcoming work, we build on this idea and develop an algorithm with a significantly improved exponent, approaching the limit of this approach at $n/2$.

Theorem 37 (Improved gap algorithms [BHL+26])**.** *Given a cubic polynomial $f(x) \in \mathbb{F}_2[x_1, \ldots, x_n]$,* $\mathrm{gap}(f)$ *can be computed exactly in expected time $2^{0.552n+O(\sqrt{n})}$.*

The counting version of the strong exponential time hypothesis (#SETH) implies that for any $\epsilon > 0$ there is a constant $d > 2$ such that the gap of degree-d Boolean polynomials cannot be computed in time $(2-\epsilon)^n$ [Wil18] and the same statement holds true for any field $\mathbb{F}_q$ [DHKW24]. While these results do not rule out a specific exponent for $d = 3$, they rule out algorithms that are not sensitive to the polynomial degree. For these reasons, we believe that even achieving runtime $\tilde{O}(2^{n/2})$ without a space constraint would constitute a breakthrough.

Contrasting with the standard approach of evaluating the sum over 2^n phases, this asymptotically improved approach has the drawback that it needs to store explicit coefficient tables for F_b in the middle of the computation. These will be exponentially large, and in fact for the algorithm of Theorem 37, they have size $2^{0.552n+O(\sqrt{n})}$, matching the time bound.

However, in practice, space is a more limited resource than time, in particular so for parallelizable algorithms such as the ones used for computing the gap. This motivates considering only algorithms that use polynomial space. To the best of our knowledge, the best such algorithm has runtime $O(n^2 2^n)$ compared to the naïve $O(n^3 2^n)$. This algorithm is based on using Gray codes to efficiently evaluate the gap. Moreover, there is no known algorithm that achieves high XEB values in the absence of noise that circumvents full-circuit simulation. We therefore conjecture that in fact achieving a high constant XEB score requires genuinely exponential time for some constant $\kappa > 1/2$.

Conjecture 38 (Exponential XEB (ExpXEB) conjecture)**.** *There exists a $\kappa > 1/2$ such that any classical polynomial-space algorithm which, given uniformly random $f_1, \ldots, f_L \leftarrow \mathcal{P}_3(n)$ solves $\mathsf{XEB}_{1,\ell}(f_1, \ldots, f_L)$ for $\ell, L = \mathsf{poly}(n)$ with at most inverse polynomial failure probability, requires time $\Omega(2^{\kappa n} \cdot \mathsf{poly}(n))$.*

Simulating planted polynomials The runtime reduction for planted polynomials is possible due to an observation of Maslov et al. [MBT+24].

Definition 39 (Vertex cover)**.** *Let $H = (V, E)$ be a hypergraph. Then a* vertex cover *is a set $C \subset V$ satisfying $\forall e \in E\, \exists v \in e : v \in C$, i.e., every hyperedge overlaps with C.*

The key observation is that given f, a vertex cover of the hypergraph $H(f)$ defined in Definition 21 yields a low-rank stabilizer decomposition of a_f. For notation, given a set $S \subset [n]$ and a bit string $x \in \mathbb{F}_2^n$ write $x_S = (x_i)_{i \in S} \in \mathbb{F}_2^{|S|}$ for the substring corresponding to the indices in S, and let $S^c = [n] \setminus S$.

Lemma 40 (Vertex cover decomposition of amplitudes)**.** *Let $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a cubic polynomial. Given a vertex cover $C \subset [n]$ of $H(f)$, we can write*

$$a_f(x) = \frac{1}{2^{|C|}} \sum_{z \in \mathbb{F}_2^{|C|}} \mathrm{ngap}(f_z^C + \langle x_{C^c}, \cdot \rangle)(-1)^{\langle x_C, z \rangle}, \tag{49}$$

for quadratic polynomials $f_z^C \in \mathbb{F}_2[x_1, \ldots, x_{|C^c|}]$ specified in the proof.

Proof. For simplicity, let us assume that f is homogeneous. Then we can write

$$f(x) = \sum_{j<k \in C^c} \left(\sum_{i \in C} (F_{ijk} + F_{jik} + F_{jki}) x_i \right) x_j x_k + \sum_{k \in C^c} \left(\sum_{i<j \in C} (F_{ijk} + F_{ikj} + F_{kij}) x_i x_j \right) x_k + \left(\sum_{i<j<k \in C} F_{ijk} x_i x_j x_k \right) =: f_{x_C}^C(x_{C^c}), \tag{50}$$

which defines an (inhomogeneous) quadratic polynomial over $\mathbb{F}_2$, given an assignment of the cover variables $x_i, i \in C$. This means that we can write the outcome amplitudes as

$$2^n a_f(x) = \sum_{y_C} (-1)^{\langle x_C, y_C \rangle} \left(\sum_{y_{C^c}} (-1)^{f_{y_C}^C(y_{C^c}) + \langle x_{C^c}, y_{C^c} \rangle} \right) \tag{51}$$

$$= \sum_{z \in \mathbb{F}_2^{|C|}} \mathrm{gap}(f_z^C + \langle x_{C^c}, \cdot \rangle)(-1)^{\langle x_C, z \rangle}. \tag{52}$$

The inhomogeneous case follows from the fact that the remaining terms are quadratic to begin with. □

The vertex-cover decomposition directly implies an algorithm for strongly simulating—i.e., computing the amplitudes of—cubic IQP circuits.

Theorem 41. *Let f be a cubic polynomial with h-dimensional independent space V. There is an algorithm that exactly computes $a_f(x)$ in time $O(2^{n-h}n^3)$ given f and a basis of V.*

Proof. To start, let $V = \langle e_1, \ldots, e_h \rangle$ be the subspace spanned by the first h coordinates. Then $C = [h+1, \ldots n]$ is a vertex cover. The proof follows directly from applying the polynomial-time algorithm for computing the gap of quadratic polynomials [EK90] to the decomposition of Lemma 40.

For an arbitrary subspace, let $A \in \mathrm{GL}(n, \mathbb{F}_2)$ be the change of basis that maps the first h coordinates to V, i.e., $V = A(\langle e_1, \ldots, e_h \rangle)$. Then $a_f(x) = a_{f \circ A}(A^T x)$ and $f \circ A$ has vertex cover $[h+1, \ldots, n]$, so we can apply the algorithm above. □

Remark 42 (Quantum estimation of gap)**.** *The gap of cubic polynomials can be estimated on a quantum computer in time $O(2^{n/2})$ for typical instances. To see this, let $|\psi_f\rangle = H^{\otimes n} C(f) H^{\otimes n} |0^n\rangle$ be the output state of the IQP circuit. Now, observe that applying phase estimation on the Grover operator $(\mathbb{1} - 2|0\rangle\langle 0|)(\mathbb{1} - 2|\psi_f\rangle\langle\psi_f|)$ can estimate the angle $\theta = 2\arcsin(\mathrm{ngap}(f))$ up to precision ϵ in time $O(1/\epsilon)$ [BHMT02]. But $\mathrm{ngap}(f) \sim 2^{-n/2}$ for typical f which shows the claim.*

Proof of main result We are now ready to prove our main result (Theorem 4), which we here state formally.

Theorem 43 (Verifiable advantage from planted independent spaces)**.** *Assume the* ExpPlantedIndSp *and* ExpXEB *conjectures (28 and 38). Let* $h = (1-c)n$ *with* $1/2 < c < \kappa \leq 1$ *and* $\ell, L = \mathsf{poly}(n)$*. The verifier chooses* $f_1, \ldots, f_L \leftarrow \mathcal{P}_3(n,h)$ *independently and asks the prover to solve* $\mathsf{XEB}_{1,\ell}(f_1, \ldots, f_L)$*. Then,*

(i) given samples $\{x_i^{f_j}\}_{i\in[\ell],j\in[L]}$ *from the prover, using knowledge of the planted independent spaces, the verifier can compute* $\hat{\tilde{\chi}}(\{x_i^{f_j}\}_{i\in[\ell],j\in[L]})$ *in time* $O(L\ell n^3 2^{cn})$ *and polynomial space,*

(ii) an honest, polynomial-time quantum prover passes $\mathsf{XEB}_{1,\ell}(f_1, \ldots, f_L)$ *test with probability at least* $1 - 10/L$*,*

(iii) there is no classical polynomial-space algorithm with runtime $o(2^{\kappa n})$ *that passes the XEB test with failure probability* $1/\mathsf{poly}(n)$*.*

Proof. (i) follows from Theorem 41 and (ii) follows from Lemma 35. To show (iii), consider the following hybrid conjecture.

Conjecture 44. *There exists a* $1/2 < \kappa \leq 1$ *such that any classical polynomial-space algorithm which, given random planted polynomials* $f_1, \ldots, f_L \leftarrow \mathcal{P}_3(n,h)$ *solves* $\mathsf{XEB}_{1,\ell}(f_1, \ldots, f_L)$ *for* $\ell, L = \mathsf{poly}(n)$ *with at most inverse polynomial failure probability, requires time* $\Omega(2^{\kappa n} \cdot \mathsf{poly}(n))$*.*

Conjecture 44 directly implies (iii). We now argue that it is implied by Conjecture 38 and 28. To this end, suppose Conjecture 44 was false. Then there exists an algorithm $\mathcal{A}$ with runtime $o(2^{\kappa n})$ using polynomial space which passes $\mathsf{XEB}_{1,\ell}$ with inverse polynomial failure probability for planted polynomials. We can use $\mathcal{A}$ as a distinguisher between $\mathcal{P}_3(n)$ and $\mathcal{P}_3(n,h)$ by running $\mathcal{A}$ on random $f_1, \ldots, f_L \leftarrow \mathcal{P}_3(n,h)$ and outputting `planted` if it terminates in time $o(2^{\kappa n})$ and the returned samples pass $\mathsf{XEB}_{1,\ell}$. On the other hand, Conjecture 38 implies that for uniformly random polynomials, any polynomial-space algorithm passing $\mathsf{XEB}_{1,\ell}$ requires runtime $\Omega(2^{\kappa n})$. In either case, the XEB can be computed in time $O(2^{0.9965n})$ by Theorem 36. Therefore $\mathcal{A}$ yields a distinguisher with overall runtime $o(2^n)$, contradicting Conjecture 28. □

3.4 Towards certified randomness from planted IQP circuits

In this section, we sketch the application of IQP circuit sampling with planted independent spaces to generate classically verifiable randomness. The idea is to apply the framework of Aaronson and Hung [AH23] for verifiable randomness from random circuit sampling to our setting of planted IQP circuits. This is possible since as shown in Lemma 32, the output distributions of random IQP circuits—like random circuits—have high min-entropy with high probability. But in contrast to random circuits, there are instance sizes of planted IQP circuits that are verifiable but not classically simulatable. This overcomes the key obstacles present in the protocol by Aaronson and Hung [AH23] and Liu et al. [LSN+25]: Their protocol requires that a quantum server quickly respond to queries in order to prevent it from spending classical resources on spoofing XEB, while the classical client is allowed to spend a long time verifying the obtained samples post-hoc. In contrast, when basing the certifiable randomness protocol on planted IQP circuits, we showed that ExpXEB and ExpPlantedIndSp imply a gap between verification and simulation time, allowing us to certify samples more efficiently. This means that the timing requirement can at least be relaxed, if not removed from the protocol since spoofing the protocol using a classical computer will be out of reach for any available resources.

Crucially, however, certifiable randomness requires stronger guarantees on the hardness of the solved task than quantum advantage, since we must ensure that no *efficient quantum adversary* can pass the protocol using samples with low entropy. We conjecture that this is true. While a naïve application of the [AH23] protocol concerns polynomial-time quantum provers, we discuss stronger quantum hardness conjectures and attacks with a characteristic exponent 1/2. We defer a detailed analysis of the implications of $\tilde{O}(2^{n-h})$-time classical and $O(2^{n/2})$-time quantum spoofing algorithms for the security of the protocol to future work.

3.4.1 The certified randomness protocol

We consider the protocol of Aaronson and Hung [AH23] for random cubic IQP circuits with planted independent spaces. We showed in Lemma 32 that both planted cubic IQP circuits and uniformly random cubic IQP circuits generate $2n/3 - O(\log 1/\delta)$ and $n - O(\sqrt{n})$ bits of min entropy with high and overwhelming probability, respectively. The idea of Aaronson and Hung is that—under their hardness assumption—any efficient quantum adversary that passes XEB (for random circuits, in their case) with high probability must have generated samples with a linear amount of average entropy.

For the purpose of this work, we restrict ourselves to their single-round entropy analysis, without adversarial side information. Their reduction turns a sufficiently successful low-entropy prover into a protocol for the following task.

Task 45 (Long List Quantum Supremacy Verification $\mathsf{LLQSV}(\mathcal{U})$ [AH23])**.** *We are given oracle access to $M = 2^{3n}$ quantum circuits $C_1, \ldots, C_M$, each on n qubits, which are promised to be drawn independently from the distribution $\mathcal{U}$. We're also given oracle access to M strings $s_1, \ldots, s_M \in \{0,1\}^n$. Then the task is to distinguish the following two cases:*

(1) ***No-Case****: Each s_i is sampled independently, uniformly from $\{0,1\}^n$.*

(2) ***Yes-Case****: Each s_i is sampled independently from p_{C_i}, the output distribution of C_i.*

They make the following conjecture.

Conjecture 46 ($\mathsf{LLHA}_B(\mathcal{U})$)**.** *For some parameter $B < n$:*

$$\mathsf{LLQSV}(\mathcal{U}) \notin \mathsf{QCAMTIME}(2^B n^{O(1)})/q(2^B n^{O(1)}), \tag{53}$$

where QCAM *is the class of problems that admit an* AM *protocol with classical communication and a quantum verifier, and* $\mathsf{QCAMTIME}(T)$ *is a generalization where the verifier can use running time T but communication is still polynomial in n. Finally, in* $\mathsf{QCAMTIME}(T)/q(A)$ *we additionally give the verifier A qubits of advice that depend only on n.*

Based on this conjecture, they show the following theorem.

Theorem 47 ([AH23, Theorem 5.10])**.** *Let $\mathcal{U}$ be an ensemble of n-qubit circuits satisfying $\mathsf{LLHA}_B(\mathcal{U})$, and let $b > 0$ be fixed. Let $\mathcal{A}$ be a quantum polynomial-time algorithm that solves $\mathsf{XEB}_{b,\ell}(C)$ for $C \leftarrow \mathcal{U}$ with probability at least q over the choice of circuit and $\mathcal{A}$'s internal randomness. Then $\mathcal{A}$'s output $s_1, \ldots, s_\ell$ satisfies*

$$H(s_1, \ldots, s_\ell \mid C) \geq \frac{B}{2}\left(\frac{(b+1)q - 1}{b} - o(1)\right),$$

where H denotes the average Shannon entropy $H(\cdot|C) = \mathbb{E}_{c\leftarrow\mathcal{U}}[H(\cdot|C=c)]$.

The key idea of the reduction is that there is an AM protocol which uses an adversary who passes XEB with too little average entropy in order to solve LLQSV. Intuitively, the assumption (Conjecture 46) is plausible because the output distributions of the considered circuit ensemble have very high min entropy.

3.4.2 Application to cubic IQP circuits

We can straightforwardly apply the single-round result of Aaronson and Hung [AH23] to the uniformly random cubic IQP ensemble under $\mathsf{LLHA}_B(\mathcal{P}_3(n))$. For that ensemble, the verifier has no planted-space simulation secret and the protocol is completely analogous to the random-circuit one, where verification can be performed after receiving a fast response. We want to apply the certified randomness argument to instances with planted independent spaces that ease verification. To this end, we assume $\mathsf{LLHA}_B(\mathcal{P}_3(n,h))$ with $B = \Theta(n)$ and apply Aaronson and Hung's reduction to this planted ensemble.[2] This will rule out quantum polynomial-time algorithms that pass $\mathsf{XEB}_{b,\ell}$ without producing high-entropy samples. A simple result we can directly infer is an entropy bound obtained against polynomial-time quantum provers.

Corollary 48 (Entropy bound for planted IQP circuits)**.** *Assume* $\mathsf{LLHA}_B(\mathcal{P}_3(n,h))$ *with* $B = \Theta(n)$*. Let* $\mathcal{Q}$ *be a polynomial-time quantum prover that, given* $f \leftarrow \mathcal{P}_3(n,h)$*, passes the single-circuit test* $\mathsf{XEB}_{1,\ell}(f)$ *with probability at least* $1-\varepsilon$*. Then* $\mathcal{Q}$*'s output* $s_1,\ldots,s_\ell$ *satisfies*

$$H(s_1,\ldots,s_\ell \mid f) \geq \frac{B}{2}\left(1-2\varepsilon-o(1)\right).$$

Proof. Apply Theorem 47 to the planted IQP circuit ensemble with $b=1$ and $q \geq 1-\varepsilon$. □

For a full security analysis of our protocol, we also want to rule out classical polynomial-space provers with a time budget comparable to the verifier's $2^{n-h}\mathsf{poly}(n)$. This requires a separate classical hardness assumption for the same planted ensemble, since the polynomial-time restriction in Theorem 47 does not cover exponential-time classical provers. Specifically, for $h = \lfloor \alpha n \rfloor$ with fixed $0 < \alpha < 1/2$, assume there exist constants $\kappa > 1-\alpha$ and $\varepsilon_{\mathrm{cl}} \in (0, 1/4)$ such that every randomized classical polynomial-space algorithm of runtime $o(2^{\kappa n})$ passes this test with probability at most $1-\varepsilon_{\mathrm{cl}}$ for all sufficiently large n. Such algorithms cannot attain the acceptance probability in Corollary 48 for fixed $\varepsilon < \varepsilon_{\mathrm{cl}}$. Deriving this assumption from ExpXEB and ExpPlantedIndSp requires a reduction that preserves the test's success parameters and fits the assumed time and space bounds. We leave these details of a complete protocol with the appropriate security guarantees for future work.

4 Worst-case hardness of the independent space problem

For the moment, we are working over arbitrary fields $\mathbb{F}$ in this section. In this section, we will prove that it is NP-hard in the worst-case to decide whether a given cubic form has an independent subspace of a given size. This is done by showing that a cubic form has an independent subspace of large size iff its symmetric slice rank is low. We then show that it is NP-hard to decide whether the symmetric slice rank of a cubic form is bounded. Brand, Kaski, and Wang [BKW26] have recently shown the same hardness result for symmetric slice rank with a different proof, independently of our work. We still keep our proof here, since our main result is that for a trilinear form whose support is an antichain, slice rank as a trilinear form and symmetric slice rank as a cubic form coincide (Corollary 62). This is a structural insight of independent interest.

4.1 Slice rank (trilinear case)

Let x, y, and z be three sets of variables.

[2] Here, in some abuse of notation, we denote the distribution over circuits $H^{\otimes n}C(f)H^{\otimes n}$ for $f \leftarrow \mathcal{P}_3(n,h)$ by $\mathcal{P}_3(n,h)$ as well.

Definition 49 (Sawin and Tao [ST16])**.** *Let $P \in \mathbb{F}[x,y,z]$ be a trilinear form.*

1. *A slice decomposition is a collection of linear forms $\ell_{x,i}(x)$ and bilinear forms $B_{x,i}(y,z)$, $1 \le i \le r_x$, linear forms $\ell_{y,j}(y)$ and bilinear forms $B_{y,j}(x,z)$, $1 \le j \le r_y$, as well as linear forms $\ell_{z,k}(z)$ and bilinear forms $B_{z,k}(x,y)$, $1 \le k \le r_z$, such that*

$$P(x,y,z) = \sum_{i=1}^{r_x} \ell_{x,i}(x) B_{x,i}(y,z) + \sum_{j=1}^{r_y} \ell_{y,j}(y) B_{y,j}(x,z) + \sum_{k=1}^{r_z} \ell_{z,k}(z) B_{z,k}(x,y).$$

2. *The slice rank $\mathrm{SR}(P)$ is the minimum r such that there is a slice decomposition as above with $r_x + r_y + r_z \le r$.*

Theorem 50 (Bläser et al. [BIL+21])**.** *Deciding whether the slice rank of a given trilinear form is smaller than a given r is* NP*-hard.*

Definition 51. *A trilinear form $P(x,y,z) = \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} p_{i,j,k} x_i y_j z_k$ has an independent triple of spaces of dimension h if there are spaces $V_x \le \mathbb{F}^\ell$, $V_y \le \mathbb{F}^m$, and $V_z \le \mathbb{F}^n$ of dimensions h_x, h_y, and h_z with $h_x + h_y + h_z = h$ such that $P(V_x, V_y, V_z) = \{0\}$.*

If there is an independent triple as above, then there are $A \in \mathrm{GL}(\ell,\mathbb{F})$, $B \in \mathrm{GL}(m,\mathbb{F})$, and $C \in \mathrm{GL}(n,\mathbb{F})$ such that in $P(Ax,By,Cz) = \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} p'_{i,j,k} x_i y_j z_k$, $p'_{i,j,k} = 0$ for all $1 \le i \le h_x$, $1 \le j \le h_y$, and $1 \le k \le h_z$.

Remark 52. *The notion of an independent triple is less intuitive for trilinear forms than independent spaces for cubic forms, as there is always an independent triple of dimension $m+n$ by setting $h_x = 0$ or any of the other two parameters. But this is in accordance with the fact that the slice rank is at most ℓ and this trivial triple corresponds to the slice decomposition in which all linear forms depend only on x.*

There is a one-to-one correspondence between the slice rank of a trilinear form and the dimension of an independent triple of spaces, which immediately follows from [ST16, Lemma 1(iv)].

Theorem 53. *For any trilinear form P, $\mathrm{SR}(P) \le r$ iff there is an independent triple of spaces of dimension $\ge \ell + m + n - r$.*

For a trilinear form $P(x,y,z) = \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} p_{i,j,k} x_i y_j z_k$, let $\mathrm{supp}(P) = \{(x_i, y_j, z_k) \mid p_{i,j,k} \neq 0\}$. $(x \times y \times z, \mathrm{supp}(P))$ is a 3-uniform, 3-partite hypergraph, which, by a slight abuse of notation, we also denote by $\mathrm{supp}(P)$. For a hypergraph H, let $\tau(H)$ denote the vertex cover number, that is, the minimum size of a vertex cover.

Lemma 54 ([ST16])**.** $\mathrm{SR}(P) \le \tau(\mathrm{supp}(P))$.

Since the slice rank is invariant under invertible linear transformations of the variables, that is, under $\mathrm{GL}(\ell,\mathbb{F}) \times \mathrm{GL}(m,\mathbb{F}) \times \mathrm{GL}(n,\mathbb{F})$, we can take the minimum over all supports under this action on the right hand side.

Assume that there are three total orderings on the variable sets x, y, and z, respectively. Consider the induced (partial) product order on $x \times y \times z$ defined by

$$(x,y,z) \ge (u,v,w) \quad \Longleftrightarrow \quad x \ge u, \quad y \ge v, \quad z \ge w.$$

We moreover say $(x,y,z) > (u,v,w)$ if $(x,y,z) \ge (u,v,w)$ and $(x,y,z) \neq (u,v,w)$. Let $\max \mathrm{supp}(P)$ denote all triples in the support that are maximal in the product order.

Lemma 55 ([ST16])**.** $\mathrm{SR}(P) \ge \tau(\max \mathrm{supp}(P))$.

If $\operatorname{supp} P$ is an antichain, that is, all elements are maximal, then the slice rank of P equals $\tau(\operatorname{supp} P)$.

Definition 56. *A trilinear form $P(x, y, z)$ is called* oblique *if there are linear transforms A, B, C such that $\operatorname{supp} P(Ax, By, Cz)$ is an antichain.*

The hardness proof by Bläser et al. [BIL+21] only constructs instances of oblique trilinear forms. Even stronger, all instances are presented in such a way that the support of the constructed trilinear form is already an antichain.

4.2 Symmetric slice rank (polynomial case)

Let $P(x)$ be a cubic form and let us write P in the standard form

$$P(x) = \sum_{i \le j \le k} p_{i,j,k} x_i x_j x_k. \tag{54}$$

Over the binary field, the $=$ cases capture lower-degree terms since $x_i^2 = x_i$.

Definition 57. *Let $P \in \mathbb{F}[x]$ be a cubic form.*

- *A symmetric slice decomposition is a collection of linear forms $\ell_i(x)$ and quadratic forms $Q_i(x)$, $1 \le i \le r$ such that*
$$P(x) = \sum_{i=1}^{r} \ell_i(x) Q_i(x)$$
- *P has symmetric slice rank $\le r$, denoted by $\operatorname{symSR}(P) \le r$, if there is a symmetric slice decomposition with $\le r$ terms.*

Remark 58. *We have $\operatorname{symSR}(P) = \operatorname{symSR}(P \circ A)$ for any $A \in \mathrm{GL}(n, \mathbb{F})$ as formal polynomials. Functionally, however, this might not be true over $\mathbb{F}_2$, the field in which we are most interested. Consider the elementary symmetric polynomial*

$$E = abc + abd + acd + bcd.$$

It is easy to check that $\operatorname{symSR}(E) = 2$. However, if we transform $d \to a + b + c + d$ and keep all other variables fixed and reduce all resulting squares, then we get

$$E' = d(ab + ac + bc),$$

which has slice rank 1. However, our NP*-hardness proof also works over $\mathbb{F}_2$, since we consider trilinear forms whose support forms an antichain and we can work with vertex cover numbers, see Lemma 61.*

The relation of symmetric slice rank to regular slice rank seems to be rather unexplored. Oneto and Ventura [OV25] show: The monomial $x_1 x_2^2$ obviously has symmetric slice rank 1, but the corresponding symmetric trilinear form $\frac{1}{3}(x_1 y_2 z_2 + x_2 y_1 z_2 + x_2 y_2 z_1)$ has slice rank 2.

We will here do the opposite. We start with a trilinear form $T(x, y, z)$ and view it as a cubic form $P_T(x \cup y \cup z)$ in the variable set $x \cup y \cup z$. In the former case, the natural action on T is $\mathrm{GL}(n_x, \mathbb{F}) \times \mathrm{GL}(n_y, \mathbb{F}) \times \mathrm{GL}(n_z, \mathbb{F})$, while on P_T, it is $\mathrm{GL}(n_x + n_y + n_z, \mathbb{F})$. If we order the variables in $x \cup y \cup z$ as $x_1, \ldots, x_{n_x}, y_1, \ldots, y_{n_y}, z_1, \ldots, z_{n_z}$, then the support $\operatorname{supp}(T)$ of T and the hypergraph $H(P)$ of P are the same (up to renaming of vertices). As a main result we show that for certain trilinear forms $T(x, y, z)$, the slice rank of T and the symmetric slice rank of $P_T(x \cup y \cup z)$ coincide.

Ananyan and Hochster [AH20] define a related concept to symmetric slice rank called the *strength*. The strength $\mathrm{Str}(P)$ of a form P of degree d is the minimum r such that there exist F_i, G_i, $1 \leq i \leq r$, with $1 \leq \deg F_i, \deg G_i < d$ such that

$$P(x) = \sum_{i=1}^{r} F_i(x) G_i(x).$$

Remark 59. *For cubic forms P,* $\mathrm{symSR}(P) = \mathrm{Str}(P)$.

Theorem 60. *For any cubic form P,* $\mathrm{symSR}(P) \leq r$ *iff P has a formally independent space of dimension $\geq n - r$.*

Here formally independent means that P is the zero polynomial after plugging the equations of the subspace into P. Since formally independent implies functionally independent as defined in Definition 1, the "⇒"-direction is also true functionally. However, we do not need the above theorem for the hardness proof, but will work with Lemma 61 instead.

Proof. Let

$$P(x) = \sum_{i=1}^{r} \ell_i(x) Q_i(x)$$

be a symmetric slice decomposition. We can assume that the ℓ_i are linearly independent. The kernel of $\ell_1, \ldots, \ell_r$ is an independent space and has dimension $n - r$.

For the other direction, assume that P has an independent space V of dimension $n - r$. Let $A \in \mathrm{GL}(n, \mathbb{F})$ be an invertible matrix whose first $n - r$ columns form a basis of V. Then all monomials $x_i x_j x_k$ in $P(Ax)$ have an index $> n - r$. Thus we can write

$$P(Ax) = \sum_{i=1}^{r} x_{n-i+1} Q_i(Ax).$$

for appropriate quadratic forms. Therefore, $\mathrm{symSR}(P) \leq r$. □

We now prove the main lemma of this section, which, given a trilinear form T, connects the symmetric rank of the corresponding cubic form P_T to the vertex cover number of $H(P_T)$.

Lemma 61. *Let $T(x, y, z)$ be a trilinear form. Assume that there is a total ordering on the variables such that every x-variable is larger than every y-variable and every y-variable is larger than every z-variable. Furthermore, assume that* $\mathrm{supp}\, T$ *is an antichain with respect to the product ordering induced by the total ordering restricted to the three variable sets. Then* $\mathrm{symSR}(P_T) \geq \tau(H(P_T))$.

Proof. Note that $\mathrm{supp}\, T$ and $H(P_T)$ are isomorphic as hypergraphs. For simplicity, we will simply write P instead of P_T during the rest of the proof.

Let the variables be $x_1, \ldots, x_{n_x}$, $y_1, \ldots, y_{n_y}$ and $z_1, \ldots, z_{n_z}$ and assume that this is the ordering, that is, x_1 is the largest element, etc. Let $N = n_x + n_y + n_z$. Let

$$P(x, y, z) = \sum_{i=1}^{r} \ell_i(x, y, z) Q_i(x, y, z)$$

be a symmetric slice decomposition with linear forms $\ell_1, \ldots, \ell_r$. Let $u_1, \ldots u_{N-r}$ be a basis of the kernel of the span of $\ell_1, \ldots, \ell_r$. Think of u as row vectors and order the entries of u with respect to the total ordering from left to right. That is, the entries $1, \ldots, n_x$ correspond to the x-variables, the entries $n_x+1, \ldots, n_x+n_y$ to the y-variables, and the entries $n_x+n_y+1, \ldots, n_x+$

$n_y + n_z$ to the z-variables. Write $u_1, \dots, u_{N-r}$ as a matrix and bring it to row echelon form, that is, each u_i looks like $(0, \dots, 0, 1, *, \dots, *)$. This can be achieved purely by invertible row operations which just means that we choose a different basis. Assume that u_i has the first 1 in column j_i. Let $D = \{j_1, \dots, j_{N-r}\}$. By the properties of row echelon form, $|D| = N - r$. Let $C = \{1, \dots, N\} \setminus D$. We claim that C is a vertex cover of $H(P)$.

Let $e \in H(P)$ be a hyperedge. Assume that e corresponds to the monomial $c_e \cdot x_a y_b z_c$. Assume for a contradiction that e is not covered, that is, the three indices $a, n_x + b, n_x + n_y + c \in D$, that is, they correspond to a 1 in the row echelon form. Let v_a, v_b, v_c be the corresponding vectors (each of them is one of the $u_1, \dots, u_{N-r}$). Consider the three-dimensional space V generated by $d_a v_a + d_b v_b + d_c v_c$, where d_a, d_b, d_c are variables. Since $u_1, \dots, u_{N-r}$ is a basis of the kernel of $\langle \ell_1, \dots, \ell_r \rangle$, P vanishes on V.

We now study the behavior of the monomials of P on V. The entry of an element $v \in V$ at the position corresponding to x_a in V is $v_{x_a} = d_a$, at the position y_b it is $\lambda d_a + d_b$ for some fixed $\lambda \in \mathbb{F}$ independent of v and at the position z_c it is $\lambda' d_a + \lambda'' d_b + d_c$ for $\lambda', \lambda'' \in \mathbb{F}$. This is due to the fact that the indices $a, n_x + b, n_x + n_y + c$ correspond to 1's in the row echelon form. Thus, the monomial corresponding to e has the value

$$c_e \cdot d_a(\lambda d_a + d_b)(\lambda' d_a + \lambda'' d_b + d_c) = c_e \cdot d_a d_b d_c + O(d_a^3 + d_a^2 d_b + d_a^2 d_c + d_a d_b^2)$$

on V.

We will now derive a contradiction to P vanishing on V by showing that the term $d_a d_b d_c$ in the polynomial P on V cannot be generated by any other monomial of P and therefore the term $d_a d_b d_c$ cannot be canceled by any other monomial. To this end, let $e' = (a', b', c') \in \operatorname{supp} P$ be an arbitrary different element corresponding to the monomial $x_{a'} y_{b'} z_{c'}$ and consider its behavior on V. We will show that the resulting polynomial does not contain the monomial $d_a d_b d_c$.

We do this by using that $\operatorname{supp} P$ forms an antichain, and that $e' \neq e$, that is, at least one of $x_{a'} > x_a$, $y_{b'} > y_b$, or $z_{c'} > z_c$ must be true.

Case 1: $x_{a'} > x_a$. Then the value at the position corresponding to $x_{a'}$ of any vector in V is 0. Thus, the monomial corresponding to e' evaluates to 0 on V.

Case 2: $y_{b'} > y_b$. Then the value at the position corresponding to $x_{a'}$ is μd_a and the value at the position corresponding to $y_{b'}$ is $\mu' d_a$ for some $\mu, \mu' \in \mathbb{F}$. Thus the monomial on V takes the form $O(d_a^2(d_a + d_b + d_c))$

Case 3: $z_{c'} > z_c$. Then the value at the position corresponding to $x_{a'}$ is μd_a, to $y_{b'}$ is $\mu' d_a + \mu'' d_b$, to $z_{c'}$ is $\mu''' d_a + \mu'''' d_b$ for $\mu, \mu', \mu'', \mu''', \mu'''' \in \mathbb{F}$. Thus the monomial $x_{a'} y_{b'} z_{c'}$ on V takes the form $O(d_a^3 + d_a^2 d_b + d_a d_b^2)$.

Altogether, we have

$$0 = P(V) = c_e \cdot d_a d_b d_c + \text{other terms}$$

which implies $c_e = 0$ contradicting the fact that $e \in H(P)$. Therefore, C is a vertex cover of size r. □

Corollary 62. *If $T(x, y, z)$ is a trilinear form such that* $\operatorname{supp} T$ *is an antichain, then* $\mathrm{SR}(T) = \mathrm{symSR}(P_T)$.

Proof. This follows from the fact that $\mathrm{SR}(T) = \tau(\operatorname{supp} T)$ when $\operatorname{supp} T$ is an antichain [ST16, Prop. 4] and the fact that $\tau(H(P_T)) \leq \mathrm{symSR}(P_T)$ (Lemma 61). We have $\mathrm{symSR}(P_T) \leq \mathrm{SR}(T)$, since every slice decomposition of T is also a symmetric slice decomposition of P_T. Finally, $H(P_T)$ and $\operatorname{supp} P$ are isomorphic and their cover numbers are the same. □

Corollary 63. *If T is a trilinear form such that* $\operatorname{supp} T$ *is an antichain, then* $\mathrm{Str}(P_T) = \mathrm{SR}(T)$.

Corollary 64. *Computing* $\mathrm{symSR}(P)$ *is* NP*-hard, that is, deciding whether a given cubic form* P *has an independent space of dimension* h *is* NP*-hard.*

Proof. This follows from the fact that the NP-hardness proof for slice rank [BIL+21] uses only instances whose support is an antichain. For these instances, symmetric slice rank and slice rank coincide by Corollary 62. In particular, by the same pivot argument as in Lemma 61, the maximum dimensions of formally independent and functionally independent spaces coincide, and the hardness proof also works for $\mathbb{F}_2$. □

Corollary 65. *Computing* $\mathrm{Str}(P)$ *is* NP*-hard.*

Remark 66. *A related result is the following. That is, for a bilinear map* $\phi : U \times U \to V$ *where* $U \cong \mathbb{F}^n$ *and* $V \cong \mathbb{F}^m$*, we say that* $W \leq U$ *is a totally-isotropic space of* ϕ *if for any* $w, w' \in W$*,* $\phi(w, w') = 0$*. In [BCG+19], it is shown that the maximum totally-isotropic space dimension of alternating bilinear maps is* NP*-hard, by a reduction from the independent set problem on graphs. Here, our proof of* NP*-hardness for the maximum independent space dimension for symmetric trilinear forms is more involved due to the lack of corresponding graph or hypergraph problems to start with.*

5 Cryptanalysis of the planted independent space problem

5.1 The $h/n > 1/2$ regime

5.1.1 Search algorithm for $h \geq (1/2 + \epsilon)n$ through rank profiles

Let f be a cubic Boolean polynomial with constant term 0. Recall that T^f is the alternating trilinear form associated with f, as defined in Definition 15.

In the generation phase of the planted distribution $\mathcal{P}_3(n, h)$, an h-dimensional planted independent space is constructed by forcing an $h \times h \times h$ corner block of the tensor to be entirely zero, followed by a secret random change of basis to hide this structure. In other words, there exists a secret h-dimensional subspace $V \leq \mathbb{F}_2^n$ such that $\mathrm{T}^f|_{V^3} = 0$.

By Definition 15 we know that T^f captures the homogeneous degree-3 part of f: $\mathrm{T}^f = \mathrm{T}^{f^{(3)}}$. For the purpose of our algorithm, we only deal with this homogeneous cubic component, which is a cubic form. In the remainder of this subsection, we assume that f is homogeneous, that is, a cubic form. Since f has no quadratic part, Theorem 17 gives $\mathrm{T}_v^f = \mathrm{H}_v^f$ for every $v \in \mathbb{F}_2^n$. Our algorithm exploits the rank profile of these tensor slices, which is invariant under invertible linear changes of variables. We show that collecting sufficiently many linearly independent vectors whose slices have low rank allows us to recover the planted independent space.

Algorithm 1 Rank-Profile Search Algorithm for Cubic Forms with Planted Independent Spaces

Input: A homogeneous degree-3 Boolean polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$, and $h \geq (1/2 + \epsilon)n$.
Output: A basis of the planted independent space, or `Failure`.

1. Let $N = 4h \cdot 2^{n-h}$ and initialize an empty list $\mathcal{B}$.
2. For $t = 1$ to N:
   (a) Sample a vector $v^{(t)} \in \mathbb{F}_2^n$ uniformly at random, independently of all previous samples.
   (b) Compute the Hessian slice matrix $M_{v^{(t)}} \in \mathbb{F}_2^{n \times n}$, where $(M_{v^{(t)}})_{ij} = \partial_j \partial_i f(v^{(t)})$.
   (c) Compute the rank of $M_{v^{(t)}}$ over $\mathbb{F}_2$ using Gaussian elimination.
   (d) If $\operatorname{rank}(M_{v^{(t)}}) \leq 2(n-h)$ and $v^{(t)} \notin \operatorname{span}(\mathcal{B})$, append $v^{(t)}$ to $\mathcal{B}$.
   (e) If $|\mathcal{B}| = h$, output $\mathcal{B}$ and halt.
3. If the loop terminates without halting, output `Failure`.

Theorem 67. *For any fixed constant $0 < \epsilon < 1/2$ and sufficiently large n, let h be an integer satisfying $(1/2 + \epsilon)n \leq h \leq n$, and let f be the cubic component of a polynomial sampled from $\mathcal{P}_3(n, h)$. Algorithm 1 recovers the planted independent space in time $O(hn^3 2^{n-h})$ and polynomial space with probability $1 - e^{-\Omega(n)}$.*

Proof. Since f is a homogeneous multilinear cubic form, Theorem 17 gives $M_v = \mathrm{H}_v^f = \mathrm{T}_v^f$ for every $v \in \mathbb{F}_2^n$. For the analysis, we regard all N vectors as sampled independently in advance, even if the algorithm halts before examining all of them.

Suppose f contains an h-dimensional independent space V, so that $\mathrm{T}^f|_{V^3} = 0$. Whenever $v \in V$, we have $\mathrm{T}^f(v, x, y) = 0$ for all $x, y \in V$. In a basis whose first h vectors span V, the matrix M_v therefore has the block form

$$M_{v^{(t)}} = \begin{pmatrix} 0_{h \times h} & B \\ B^T & D \end{pmatrix}.$$

Since B has dimensions $h \times (n-h)$, the rank of the first h columns is at most $n-h$, and the rank of other $n-h$ columns is at most $n-h$. Therefore we obtain:

$$\operatorname{rank}(M_{v^{(t)}}) \leq 2(n-h) < n.$$

Thus, every sampled vector in V satisfies the rank condition in Step 2(d).

Next, we bound the probability that a sampled vector outside V satisfies the same rank condition. By the planted sampling procedure, T^f is uniformly distributed over the space of alternating trilinear forms satisfying $\mathrm{T}^f|_{V^3} = 0$.

For any fixed $v \notin V$, choose a basis $v, w_1, \ldots, w_{n-1}$ with $V = \operatorname{span}\{w_1, \ldots, w_h\}$. The planted condition only constrains the coefficients $\mathrm{T}^f(w_i, w_j, w_k)$ with $1 \leq i < j < k \leq h$. Thus, the coefficients $\mathrm{T}^f(v, w_j, w_k)$, $1 \leq j < k \leq n-1$, remain independent uniform bits. Since $\mathrm{T}^f(v, v, w) = 0$ for every w, the matrix M_v in this basis has the form $0 \oplus A$, where A is a uniformly random alternating $(n-1) \times (n-1)$ matrix.

For a uniformly random alternating $m \times m$ matrix A, requiring a fixed d-dimensional subspace to lie in $\ker A$ imposes $d(m-d) + \binom{d}{2}$ independent linear constraints. There are at most $4 \cdot 2^{d(m-d)}$ such subspaces, so a union bound gives

$$\Pr[\dim \ker A \geq d] \leq 4 \cdot 2^{-\binom{d}{2}}, \qquad 0 \leq d \leq m.$$

Set $p = 2^{h-n}$ and $r = 2h - n - 1$, which is positive for sufficiently large n. Applying this bound with $m = n - 1$ and $d = r$, and taking a union bound over the N samples, we obtain

$$\Pr\big[\exists t \in [N] : v^{(t)} \notin V, \mathrm{rank}(M_{v^{(t)}}) \le 2(n-h)\big] \le 4N(1-p)\,2^{-\binom{r}{2}}.$$

It remains to show that the sampled vectors in V span V. If they do not, they are all contained in some hyperplane $H < V$. There are $2^h - 1$ such hyperplanes, and for each one, $|V \setminus H| = 2^{h-1}$. Each independent sample therefore avoids $V \setminus H$ with probability $1 - p/2$. A union bound gives

$$\Pr\big[\mathrm{span}\{v^{(t)} : t \in [N],\ v^{(t)} \in V\} \neq V\big] \le (2^h - 1)\left(1 - \frac{p}{2}\right)^N \le 2^h e^{-\frac{Np}{2}} = e^{-(2-\ln 2)h},$$

where we used $N = 4h/p$.

Unless one of these two events occurs, every vector appended to $\mathcal{B}$ lies in V, and the sampled vectors in V span V. Hence the algorithm returns a basis of V, with failure probability at most

$$4N(1-p)\,2^{-\binom{r}{2}} + e^{-(2-\ln 2)h} \le 16h\,2^{n-h-\binom{r}{2}} + e^{-(2-\ln 2)h}.$$

Since $r \ge 2\epsilon n - 1$, the first term is $2^{-\Omega(n^2)}$ and the second is $e^{-\Omega(n)}$. This proves the claimed success probability. The bound holds for every fixed V and therefore also after averaging over the choice of the planted space.

Finally, each iteration takes $O(n^3)$ operations, including constructing the slice, computing its rank, and updating a basis for $\mathrm{span}(\mathcal{B})$. Thus, the total running time is $O(hn^3 2^{n-h})$. □

Remark 68. *The same technique also gives a distinguisher: using only $N = c\,2^{n-h}$ samples, where c is a sufficiently large constant, output* `Planted` *as soon as one sampled vector satisfies the rank condition, and* `Random` *otherwise.*

Remark 69. *Our algorithm works when $h \ge (1/2+\epsilon)n$. A similar idea was used in [LQTZ26] in the context of planting a totally-isotropic space in an alternating bilinear map (see Remark 66). There, a similar distinguisher runs in polynomial time for $h > n/2$. For $h = n/2$ and even $h = n/2 - o(n)$, the algorithm is also in sub-exponential time. The main idea is to view the tensor as m matrix slices $A_1, \ldots, A_m$ and analyse their rank profiles. Any linear combination of A_i always has an $h \times h$ zero block on the top-left so it is easy to distinguish by the probability of certain rank deficiency.*

While the distinguishers in [LQTZ26] and the search algorithm in Theorem 67 follow the same strategy, the results are quantitatively different. Recall that in the process of sampling from $\mathcal{P}_3(n,h)$, we set the upper-left-frontal $h \times h \times h$ corner block to be 0. The Hessian slice matrix M_v exhibits a top-left zero block whenever v belongs to the independent space, whose probability is 2^{h-n}. So in step 2 of Algorithm 1, we need to sample $N = 2^{n-h}$ vectors to obtain one in the isotropic space with high probability, so that already demonstrates a major difference between our setting and the setting in [LQTZ26]. Finally, we note that when $h > (1/2+\epsilon)n$, the rank helps to distinguish between whether $v \in H$ or not. When $h \le n/2$, we need to compute the probability of certain rank deficiency to distinguish between a random matrix and a matrix with zero block.

5.1.2 Search algorithm for $h \ge (1/2+\epsilon)n$ through computing shrunk subspaces

We now devise another $2^{n-h} \cdot \mathrm{poly}(n)$-time algorithm to find a planted independent space of dimension $h \ge (1/2+\epsilon)n$ with high probability. While this algorithm may not be more efficient than the algorithm in Theorem 67, it uses fewer samples from $\mathbb{F}_2^n$ and introduces another technique (shrunk subspace computation), so we decide to record it here.

We basically follow the setting in Section 5.1.1. Let $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a cubic form. Recall the alternating trilinear form T^f associated with f defined in Definition 15. Let $V \leq \mathbb{F}_2^n$ be the planted independent space, satisfying that $h = \dim(V) = \lceil (1/2 + \epsilon)n \rceil$.

The algorithm follows the same structure as Algorithm 1, but it requires the following preparation. In the following, C is a constant that will be determined later.

Definition 70. *Let $\mathbf{A} = (A_1, \ldots, A_C) \in \mathsf{M}(n, \mathbb{F})^C$ be a matrix tuple. For $U \leq \mathbb{F}^n$, the image of U under $\mathbf{A}$ is $\mathbf{A}(U) = \mathrm{span}\{\cup_{i \in [C]} A_i U\}$.*

For $c \in \mathbb{N}$, we say that U is a c-shrunk subspace of $\mathbf{A}$, if $\dim(U) - \dim(\mathbf{A}(U)) \geq c$. Equivalently, U is a c-shrunk subspace of $\mathbf{A}$, if there exists $W \leq \mathbb{F}^n$ of dimension $n - \dim(U) + c$, such that for any $u \in U$ and $w \in W$, and any $i \in [C]$, $w^t A_i u = 0$.

Let T be a random $n \times n \times n$ alternating tensor over $\mathbb{F}_2$, with the top-left-front $h \times h \times h$ subcube set to 0. Let $M_1, \ldots, M_C$ be generic combinations of the first h frontal slices of T. Suppose $n \gg C$. Note that $M_i = \begin{bmatrix} 0 & M_i' \\ -M_i'^t & M_i'' \end{bmatrix}$, where 0 indicates the zero matrix of size $h \times h$. Furthermore, the entries in M_i' and the upper triangular part of M_i'', for $i \in [C]$, are independent. That is, we may view M_i's as alternating matrices whose upper-triangular entries are independent.

Let H be the subspace spanned by the first h standard basis vectors. Note that H is a c-shrunk subspace where $c = \lceil (1/2+\epsilon)n \rceil - (n - \lceil (1/2+\epsilon)n \rceil) \approx 2\epsilon n$. Indeed, viewing $M_1, \ldots, M_C$ as bilinear forms on $\mathbb{F}_2^n$, $H \leq \mathbb{F}_2^n$ satisfies that for any $i \in [C]$, $M_i(H, H) = 0$.

Proposition 71. *Let $C \geq 4$, and let $M_1, \ldots, M_C$ be random $n \times n$ alternating matrices over $\mathbb{F}_2$ with the top-left $h \times h$ submatrix set to 0, with $h \geq (1/2 + \epsilon)n$. Then with high probability, $(V, W) = (H, H)$ is the only pair of subspaces of dimension h satisfying that $\forall i \in [C]$, $M_i(V, W) = 0$. Furthermore, H is the $(2h - n)$-shrunk subspace of the smallest dimension.*

Proof. Let (V, W) be a pair of subspaces of dimension h.

First, consider the case when $V = H$. Suppose $W \cap H$ is of codimension k in W, $k \geq 1$. Choose $w_1, \ldots, w_k \in W$ such that $(W \cap H) \cup \{w_1, \ldots, w_k\}$ span W. Then $M_i(H, W) = 0$ for every $i \in [C]$ implies that $M_i(H, w_j) = 0$ for every $i \in [C]$ and $j \in [k]$, which has probability $\leq 1/2^{k\left(Ch - \binom{C+1}{2}\right)}$. The number of W such that $W \cap H$ is of codimension k in W is upper bounded by $2^{k(h-k)+kn}$. As $h \geq (1/2+\epsilon)n$, when $C \geq 6$ is a constant, we have $k(Ch - \binom{C+1}{2}) \geq k(Ch - C^2) = kC(h - C) \geq 3nk - kC^2 \geq hk + nk + (nk - kC^2) \geq k(h - k) + kn$; note that we assume $n \geq C^2$. Therefore, by the union bound, when $C \geq 4$, the probability of having (H, W), $W \neq H$ such that $M_i(H, W) = 0$ for $i \in [C]$ is negligible.

Second, the case of $V \neq H$ can be dealt with in the same approach, considering their intersections with H. We omit the routine calculations here.

For the furthermore statement, note that the existence of a $(2h - n)$-shrunk subspace of dimension larger than h is equivalent to the existence of (V, W) where $\dim(V) = h+d$, $\dim(W) = h - d$, where $d \geq 1$, such that $M_i(V, W) = 0$ for $i \in [C]$. Again, it can be shown that the probability of random M_i's allowing such (V, W) is negligible by specifying bases of V and W with respect to H. □

Proposition 71 ensures that H is the c-shrunk subspace where $c = 2h - n$ of the smallest dimension. Such a shrunk subspace is called the canonical shrunk subspace in [IMQ22], which can be computed by the algorithm in [IQS18].

We can now describe the algorithm. As it is similar to Algorithm 1, we present a sketch instead of a formal description.

1. Let $N = 4C \cdot 2^{n-h}$. Initialize an empty list $\mathcal{B}$.
2. For $t = 1 \to N$:
   (a) Sample a random $v^{(t)} \in \mathbb{F}_2^n$.
   (b) Let M_t be the Hessian slice matrix $\mathrm{H}^f_{v^{(t)}}$.
   (c) Compute the rank of M_t over $\mathbb{F}_2$.
   (d) If $\mathrm{rank}(M_t) \leq 2(n-h)$ and $v^{(t)} \notin \mathrm{span}(\mathcal{B})$, append $v^{(t)}$ to $\mathcal{B}$.
   (e) If $|\mathcal{B}| = C$, continue to the next step.
3. Compute the canonical shrunk subspace V' of matrices in $\mathcal{B}$ using [IQS18].
4. Return V' if $\dim(V') = \lceil (1/2 + \epsilon)n \rceil$. Report "Failure" otherwise.

Following the analysis of Theorem 67, we see that the algorithm succeeds with high probability in sampling C matrices in the planted space H after step 2. Then by Proposition 71, the shrunk subspace V' of matrices in $\mathcal{B}$ coincides with the planted subspace H.

Clearly, the algorithm runs in time $2^{n-h} \cdot \mathsf{poly}(n)$, and uses $O(C \cdot 2^{n-h})$ many samples from $\mathbb{F}_2^n$, instead of $O(h \cdot 2^{n-h})$-many in Algorithm 1.

5.2 Analysis of Bell-sampling attacks

In this section, we consider a potential attack on the security of decision-h-PlantedIndSp based on Bell sampling. The motivation is the following: it has been shown that a certain measure of magic for quantum states—the *stabilizer nullity* [BCHK20]—can be efficiently estimated using Bell sampling [HG24; GIKL24]. On the other hand, Bell sampling can be efficiently (strongly) simulated for degree-3 circuits, since every outcome amplitude is described by a Clifford circuit [HKB+25]. But the entire point of degree-3 circuits with a planted independent space is that they have low stabilizer rank compared to random states (2^{cn} compared to 2^n). Therefore, one may be worried that Bell sampling can be used to efficiently solve decision-h-PlantedIndSp for arbitrary h.

The resolution of why this attack does not work is that while the stabilizer nullity gives an upper bound to the stabilizer rank this upper bound is "maximally" loose for vertex-cover decompositions of random circuits.

Definition 72 (Stabilizer nullity)**.** *Given a quantum state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$, the stabilizer nullity is defined as*

$$\nu(|\psi\rangle) = n - \dim(\mathcal{S}). \tag{55}$$

Here, $\mathcal{S} \leq \mathcal{P}_n$ is the (Pauli) stabilizer group of $|\psi\rangle$, i.e., $S \in \mathcal{S} \Rightarrow S|\psi\rangle = |\psi\rangle$.

Lemma 73. *Let $h \leq n-2$ and g be a random cubic polynomial with planted h-dimensional independent space. Then*

$$\Pr_{g \sim \mathcal{P}_3(n,h)}[\nu(|g\rangle) < n] \leq 2^{-n+4}. \tag{56}$$

Proof. Consider the stabilizer group $\mathcal{S}(|+^n\rangle) = \langle X_i \rangle_{i \in [n]}$. Then a stabilizer subgroup of $|g\rangle$ is generated by the Clifford operators $S_i = C(g) X_i C(g)^\dagger = X_i \cdot C(g_i)$ where

$$g_i(x) = g(x + e_i) + g(x) = \sum_{j<k} G_{ijk} x_j x_k + \ell_i(x), \tag{57}$$

where $G_{ijk} = \mathrm{T}^g(e_i, e_j, e_k)$ and ℓ_i is affine linear.

The stabilizer nullity is determined by the number of linearly independent *Pauli stabilizers*. We therefore need to determine how many Pauli stabilizers the group $\langle S_i \rangle_i$ contains. To this end, we observe that

$$S_i S_j = X_i X_j C(g_i(\,\cdot + e_j) + g_j), \tag{58}$$

and therefore an element $S(v) \in \langle S_i \rangle_i$ labelled by $v \in \mathbb{F}_2^n$ is given by

$$S(v) = X(v)C(g_v), \qquad g_v(x) = g(x+v) + g(x). \tag{59}$$

Every Pauli stabilizer belongs to this group: the diagonal part $C(g_v)$ is uniquely determined by its X-part and the nonzero computational-basis amplitudes of $|g\rangle$. $S(v) \in \mathcal{P}_n$ iff g_v is affine linear, that is, whenever $G(v) = \sum_i v_i G_{ijk} \in \mathbb{F}_2^{n\times n}$ is diagonal. Letting $G'_{i,jk} = G_{ijk}$ for $j < k$, $i \notin \{j,k\}$ and 0 otherwise, this is the case iff $v^T G' = 0$, and therefore the stabilizer nullity of $|g\rangle$ is given by

$$\nu(|g\rangle) = \mathrm{rank}(G'). \tag{60}$$

Next, we show that G' has full rank with high probability. Wlog. let the independent space of g be spanned by $e_1, \ldots, e_h$. Given $v \neq 0$, we bound $\Pr\left[v^T G' = 0\right]$ by exhibiting many entries of the row vector $v^T G'$ that are independent uniform bits. Pick any i_0 with $v_{i_0} = 1$ and consider only the entries of $v^T G'$ indexed by pairs $\{j,k\}$ with $i_0 \notin \{j,k\}$ and $\{i_0, j, k\} \not\subseteq [h]$. Each such entry has the form

$$(v^T G')_{jk} = G_{i_0 jk} + \sum_{\substack{i \in \mathrm{supp}(v) \setminus \{i_0\} \\ i \notin \{j,k\}}} G_{ijk}, \tag{61}$$

and the coefficient $G_{i_0 jk}$ appears in no other entry under consideration.[3] Conditioned on all remaining coefficients, these entries are therefore independent uniformly random bits so that

$$\Pr\left[v^T G' = 0\right] \leq 2^{-N(v)}, \qquad N(v) = \left|\left\{\{j,k\} \subseteq [n] \setminus \{i_0\} : \{i_0, j, k\} \not\subseteq [h]\right\}\right|. \tag{62}$$

- If $\mathrm{supp}(v) \not\subseteq [h]$, choose $i_0 > h$ and $\{i_0, j, k\} \not\subseteq [h]$ holds automatically, giving $N(v) = \binom{n-1}{2}$.
- If $\mathrm{supp}(v) \subseteq [h]$, then $i_0 \leq h$ and $N(v) = \binom{n-1}{2} - \binom{h-1}{2} = \frac{(n-h)(n+h-3)}{2}$.

The union bound over the at most 2^n (respectively at most 2^h) vectors of each type gives

$$\Pr\left[\mathrm{rank}(G') < n\right] \leq 2^{n - \binom{n-1}{2}} + 2^{h - \frac{(n-h)(n+h-3)}{2}}. \tag{63}$$

The exponent of the second term is increasing in h and hence maximized at $h = n-2$, where it equals $-(n-3)$; the first term is at most $2^{-(n-3)}$ for $n \geq 6$. This gives the claimed 2^{-n+4}. □

Note that for $h \leq (1-\epsilon)n$ the same argument gives the much stronger bound $2^{-\Omega_\epsilon(n^2)}$.

[3] Its only other occurrences in $v^T G'$ are in the entries $i_0 j$ and $i_0 k$, which we have excluded.

References

[AAB+19] Frank Arute et al. “Quantum Supremacy Using a Programmable Superconducting Processor”. In: *Nature* 574 (2019), pp. 505–510.

[Aar23] Scott Aaronson. “Verifiable Quantum Supremacy: What I Hope Will Be Done”. Talk at the Quantum Summer Cluster Workshop, Simons Institute for the Theory of Computing. 2023.

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity A Modern Approach*. Cambridge: Cambridge University Press, 2009.

[ABI+23] Damiano Abram, Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Varun Narayanan. “Cryptography from Planted Graphs: Security with Logarithmic-Size Messages”. In: *Theory Cryptogr.* Ed. by Guy Rothblum and Hoeteck Wee. Cham: Springer Nature Switzerland, 2023, pp. 286–315.

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. “Public-Key Cryptography from Different Assumptions”. In: *Proc. Forty-Second ACM Symp. Theory Comput.* STOC ’10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 171–180.

[AC17] Scott Aaronson and Lijie Chen. “Complexity-Theoretic Foundations of Quantum Supremacy Experiments”. In: *32nd Comput. Complex. Conf. CCC 2017*. Ed. by Ryan O’Donnell. Vol. 79. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 22:1–22:67. arXiv: 1612.05903.

[ADR82] Alain Aspect, Jean Dalibard, and Gérard Roger. “Experimental Test of Bell’s Inequalities Using Time-Varying Analyzers”. In: *Phys. Rev. Lett.* 49.25 (1982), pp. 1804–1807.

[AG20] Scott Aaronson and Sam Gunn. “On the Classical Hardness of Spoofing Linear Cross-Entropy Benchmarking”. 2020. arXiv: 1910.12085.

[AGL+23] Dorit Aharonov, Xun Gao, Zeph Landau, Yunchao Liu, and Umesh Vazirani. “A Polynomial-Time Classical Algorithm for Noisy Random Circuit Sampling”. In: *Proc. 55th Annu. ACM Symp. Theory Comput.* 2023, pp. 945–957. arXiv: 2211.03999 [quant-ph].

[AH20] Tigran Ananyan and Melvin Hochster. “Small subalgebras of polynomial rings and Stillman’s conjecture”. In: *Journal of the American Mathematical Society* 33.1 (2020), pp. 291–309.

[AH23] Scott Aaronson and Shih-Han Hung. “Certified Randomness from Quantum Supremacy”. In: *Proc. 55th Annu. ACM Symp. Theory Comput.* STOC 2023. New York, NY, USA: Association for Computing Machinery, 2023, pp. 933–944.

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. “Finding a large hidden clique in a random graph”. In: *Random Structures & Algorithms* 13.3-4 (1998), pp. 457–466.

[AZ24] Scott Aaronson and Yuxuan Zhang. *On Verifiable Quantum Advantage with Peaked Circuit Sampling*. 2024. arXiv: 2404.14493 [quant-ph]. URL: http://arxiv.org/abs/2404.14493 (visited on 04/24/2024). Pre-published.

[BCD+25] Ward Beullens et al. *UOV: Unbalanced Oil and Vinegar*. Algorithm Specifications and Supporting Documentation, Version 2.0, Round 2 Submission to the NIST Post-Quantum Cryptography Additional Digital Signature Standardization Process. February 5, 2025. 2025.

[BCG+19] Xiaohui Bei, Shiteng Chen, Ji Guan, Youming Qiao, and Xiaoming Sun. *From Independent Sets and Vertex Colorings to Isotropic Spaces and Isotropic Decompositions*. 2019. arXiv: 1904.03950 [cs]. URL: http://arxiv.org/abs/1904.03950 (visited on 02/09/2026). Pre-published.

[BCG21] Boaz Barak, Chi-Ning Chou, and Xun Gao. "Spoofing Linear Cross-Entropy Benchmarking in Shallow Quantum Circuits". In: *12th Innov. Theor. Comput. Sci. Conf. ITCS 2021*. Ed. by James R. Lee. Vol. 185. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, 30:1–30:20. arXiv: 2005.02421.

[BCHK20] Michael Beverland, Earl Campbell, Mark Howard, and Vadym Kliuchnikov. "Lower Bounds on the Non-Clifford Resources for Quantum Computations". In: *Quantum Sci. Technol.* 5.3 (2020), p. 035009.

[BCJ25] Michael J. Bremner, Bin Cheng, and Zhengfeng Ji. "Instantaneous Quantum Polynomial-Time Sampling and Verifiable Quantum Advantage: Stabilizer Scheme and Classical Security". In: *PRX Quantum* 6.2 (2025), p. 020315.

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. "A Cryptographic Test of Quantumness and Certifiable Randomness from a Single Quantum Device". In: *2018 IEEE 59th Annu. Symp. Found. Comput. Sci. FOCS*. 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS). 2018, pp. 320–331. arXiv: 1804.00640.

[BFS15] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. "On the complexity of the F5 Gröbner basis algorithm". In: *J. Symb. Comput.* 70 (2015), pp. 49–70.

[BGT25] Sergey Bravyi, Alexandru Gheorghiu, and Kristan Temme. "Verifiable quantum advantage: old and new ideas". 2025.

[BHL+26] Markus Blaeser, Dominik Hangleiter, Baitian Li, Yuxuan Liu, and Youming Qiao. "Improved algorithms for the gap of cubic polynomials". work in progress. 2026.

[BHMT02] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. "Quantum Amplitude Amplification and Estimation". In: vol. 305. 2002, pp. 53–74. arXiv: quant-ph/0005055.

[BIL+21] Markus Bläser, Christian Ikenmeyer, Vladimir Lysikov, Anurag Pandey, and Frank-Olaf Schreyer. "On the Orbit Closure Containment Problem and Slice Rank of Tensors". In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*. Ed. by Dániel Marx. SIAM, 2021, pp. 2565–2584.

[BIS+18] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. "Characterizing Quantum Supremacy in Near-Term Devices". In: *Nat. Phys.* 14.6 (2018), pp. 595–600. arXiv: 1608.00263.

[BJS10] M. J. Bremner, R. Jozsa, and D. J. Shepherd. "Classical Simulation of Commuting Quantum Computations Implies Collapse of the Polynomial Hierarchy". In: *Proc. R. Soc. Math. Phys. Eng. Sci.* 467.2126 (2010), pp. 459–472. arXiv: 1005.1407.

[BKVV20] Zvika Brakerski, Venkata Koppula, Umesh Vazirani, and Thomas Vidick. "Simpler Proofs of Quantumness". 2020. arXiv: 2005.04826.

[BKW26] Cornelius Brand, Petteri Kaski, and Jiaheng Wang. *Partition Rank and Algebraic Circuit Lower Bounds*. 2026. arXiv: 2607.02241 [cs.CC].

[BMS16] Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. "Average-Case Complexity Versus Approximate Simulation of Commuting Quantum Computations". In: *Phys. Rev. Lett.* 117.8 (2016), p. 080501.

[BMS17] Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. "Achieving Quantum Supremacy with Sparse and Noisy Commuting Quantum Computations". In: *Quantum* 1 (2017), p. 8.

[BZG+26] Ryan Babbush, Adam Zalcman, Craig Gidney, Michael Broughton, Tanuj Khattar, Hartmut Neven, Thiago Bergamaschi, Justin Drake, and Dan Boneh. *Securing Elliptic Curve Cryptocurrencies against Quantum Vulnerabilities: Resource Estimates and Mitigations*. 2026. arXiv: 2603.28846 [quant-ph]. Pre-published.

[DFG+25] Abhinav Deshpande, Bill Fefferman, Soumik Ghosh, Michael Gullans, and Dominik Hangleiter. "Peaked Quantum Advantage Using Error Correction". 2025. arXiv: 2510.05262.

[DHKP20] Alexander M. Dalzell, Aram W. Harrow, Dax Enshan Koh, and Rolando L. La Placa. "How Many Qubits Are Needed for Quantum Computational Supremacy?" In: *Quantum* 4 (2020), p. 264.

[DHKW24] Holger Dell, Anselm Haak, Melvin Kallmayer, and Leo Wennmann. "Solving Polynomial Equations Over Finite Fields". 2024. arXiv: 2410.20162 [cs].

[DHL+25] M. DeCross et al. "Computational Power of Random Quantum Circuits in Arbitrary Geometries". In: *Phys. Rev. X* 15.2 (2025), p. 021052.

[EK90] Andrzej Ehrenfeucht and Marek Karpinski. *The Computational Complexity of (XOR, AND)-Counting Problems*. Technical Report. University of Bonn, 1990.

[FR10] Uriel Feige and Dorit Ron. "Finding Hidden Cliques in Linear Time". In: *Discrete Math. Theor. Comput. Sci.* DMTCS Proceedings vol. AM, 21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms (AofA'10) (Proceedings 2010).

[GFZ+25] Dongxin Gao et al. "Establishing a New Benchmark in Quantum Computational Advantage with 105-Qubit Zuchongzhi 3.0 Processor". In: *Phys. Rev. Lett.* 134.9 (2025), p. 090601.

[GG17] Leslie Ann Goldberg and Heng Guo. "The Complexity of Approximating Complex-Valued Ising and Tutte Partition Functions". In: *Comput. Complex.* 26.4 (2017), pp. 765–833.

[GH25] David Gross and Dominik Hangleiter. "Secret-Extraction Attacks against Obfuscated Instantaneous Quantum Polynomial-Time Circuits". In: *PRX Quantum* 6.2 (2025), p. 020314.

[Gid25] Craig Gidney. "How to Factor 2048 Bit RSA Integers with Less than a Million Noisy Qubits". 2025. arXiv: 2505.15917 [quant-ph].

[GIKL24] Sabee Grewal, Vishnu Iyer, William Kretschmer, and Daniel Liang. *Improved Stabilizer Estimation via Bell Difference Sampling*. 2024. arXiv: 2304.13915 [quant-ph]. Pre-published.

[GKC+24] Xun Gao, Marcin Kalinowski, Chi-Ning Chou, Mikhail D. Lukin, Boaz Barak, and Soonwon Choi. "Limitations of Linear Cross-Entropy as a Measure for Quantum Advantage". In: *PRX Quantum* 5.1 (2024), p. 010334. arXiv: 2112.01657.

[GKVZ22] Shafi Goldwasser, Michael P. Kim, Vinod Vaikuntanathan, and Or Zamir. *Planting Undetectable Backdoors in Machine Learning Models*. arXiv.org. 2022. URL: https://arxiv.org/abs/2204.06974v2 (visited on 05/13/2026).

[GVW+15] Marissa Giustina et al. “Significant-Loophole-Free Test of Bell’s Theorem with Entangled Photons”. In: *Phys. Rev. Lett.* 115.25 (2015), p. 250401.

[Han26] Dominik Hangleiter. “Has Quantum Advantage Been Achieved?” 2026. arXiv: 2603.09901.

[HBD+15] B. Hensen et al. “Loophole-Free Bell Inequality Violation Using Electron Spins Separated by 1.3 Kilometres”. In: *Nature* 526.7575 (2015), pp. 682–686.

[HE23] Dominik Hangleiter and Jens Eisert. “Computational Advantage of Quantum Random Sampling”. In: *Rev. Mod. Phys.* 95.3 (2023), p. 035001. arXiv: 2206.04079.

[HG24] Dominik Hangleiter and Michael J. Gullans. “Bell Sampling from Quantum Circuits”. In: *Phys. Rev. Lett.* 133.2 (2024), p. 020601. arXiv: 2306.00083 [cond-mat, physics:quant-ph].

[HKB+25] Dominik Hangleiter, Marcin Kalinowski, Dolev Bluvstein, Madelyn Cain, Nishad Maskara, Xun Gao, Aleksander Kubica, Mikhail D. Lukin, and Michael J. Gullans. “Fault-Tolerant Compiling of Classically Hard Instantaneous Quantum Polynomial Circuits on Hypercubes”. In: *PRX Quantum* 6.2 (2025), p. 020338.

[HKEG19] Dominik Hangleiter, Martin Kliesch, Jens Eisert, and Christian Gogolin. “Sample Complexity of Device-Independently Certified “Quantum Supremacy””. In: *Phys. Rev. Lett.* 122.21 (2019), p. 210502. arXiv: 1812.01023.

[HS24] Shuichi Hirahara and Nobutaka Shimizu. “Planted Clique Conjectures Are Equivalent”. In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O’Donnell. ACM, 2024, pp. 358–366.

[IMQ22] Gábor Ivanyos, Tushant Mittal, and Youming Qiao. “Symbolic Determinant Identity Testing and Non-Commutative Ranks of Matrix Lie Algebras”. In: *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2022, pp. 87–1.

[IQS18] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. “Constructive Non-Commutative Rank Computation Is in Deterministic Polynomial Time”. In: *Comput. Complex.* 27.4 (2018), pp. 561–593.

[Jer92] Mark Jerrum. “Large Cliques Elude the Metropolis Process”. In: *Random Struct. Algorithms* 3.4 (1992), pp. 347–359.

[Kah23] Gregory D. Kahanamoku-Meyer. “Forging Quantum Data: Classically Defeating an IQP-based Quantum Test”. In: *Quantum* 7 (2023), p. 1107. arXiv: 1912.05547 [quant-ph].

[KCVY22] Gregory D. Kahanamoku-Meyer, Soonwon Choi, Umesh V. Vazirani, and Norman Y. Yao. “Classically Verifiable Quantum Advantage from a Computational Bell Test”. In: *Nat. Phys.* 18.8 (8 2022), pp. 918–924.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. “Unbalanced Oil and Vinegar Signature Schemes”. In: *Advances in Cryptology — EUROCRYPT ’99*. Vol. 1592. Lecture Notes in Computer Science. Springer, 1999, pp. 206–222.

[KRVK24] Gregory D. Kahanamoku-Meyer, Seyoon Ragavan, Vinod Vaikuntanathan, and Katherine Van Kirk. “The Jacobi Factoring Circuit: Quantum Factoring with Near-Linear Gates and Sublinear Space and Depth”. 2024. arXiv: 2412.12558 [quant-ph].

[Kuč95] Luděk Kučera. “Expected Complexity of Graph Partitioning Problems”. In: *Discrete Appl. Math.* Combinatorial Optimization 1992 57.2 (1995), pp. 193–212.

[LND+25] Minzhao Liu et al. *Certified Randomness Amplification by Dynamically Probing Remote Random Quantum States*. 2025. arXiv: 2511.03686 [quant-ph]. Pre-published.

[LPT+17] Daniel Lokshtanov, Ramamohan Paturi, Suguru Tamaki, Ryan Williams, and Huacheng Yu. "Beating Brute Force for Systems of Polynomial Equations over Finite Fields". In: *Proceedings of the 2017 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Proceedings. Society for Industrial and Applied Mathematics, 2017, pp. 2190–2202.

[LQTZ26] Yuxuan Liu, Youming Qiao, Gang Tang, and Chuanqi Zhang. *The Planted Tensor Problem over Finite Fields: Algorithms and Cryptography*. 2026. arXiv: 2609.31256 [cs.DS]. Pre-published.

[LSN+25] Minzhao Liu et al. "Certified Randomness Using a Trapped-Ion Quantum Processor". In: *Nature* (2025), pp. 1–6.

[MBT+24] Dmitri Maslov, Sergey Bravyi, Felix Tripier, Andrii Maksymov, and Joe Latone. *Fast Classical Simulation of Harvard/QuEra IQP Circuits*. 2024. arXiv: 2402.03211 [quant-ph]. Pre-published.

[MVM+24] A. Morvan et al. "Phase Transitions in Random Circuit Sampling". In: *Nature* 634.8033 (2024), pp. 328–333.

[OV25] Alessandro Oneto and Emanuele Ventura. "Ranks of tensors: geometry and applications". In: *Bollettino dell'Unione Matematica Italiana* 18 (3 2025), pp. 751–778.

[Pat96] Jacques Patarin. "Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms". In: *Adv. Cryptol. — EUROCRYPT '96*. Ed. by Ueli Maurer. Red. by Gerhard Goos, Juris Hartmanis, and Jan Van Leeuwen. Vol. 1070. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 33–48.

[SB09] Dan Shepherd and Michael J. Bremner. "Temporally Unstructured Quantum Computation". In: *Proc. R. Soc. Lond. Math. Phys. Eng. Sci.* 465.2105 (2009), pp. 1413–1439. arXiv: 0809.0847.

[Sho94] P.W. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring". In: *Proc. 35th Annu. Symp. Found. Comput. Sci.* Proceedings 35th Annual Symposium on Foundations of Computer Science. 1994, pp. 124–134.

[SMC+15] Lynden K. Shalm et al. "Strong Loophole-Free Test of Local Realism". In: *Phys. Rev. Lett.* 115.25 (2015), p. 250402.

[ST16] William Sawin and Terence Tao. *Notes on the "slice rank" of tensors*. https://terrytao.wordpress.com/2016/08/24/notes-on-the-slice-rank-of-tensors/. Blog post, accessed 2026-03-26. 2016.

[TCY+26] Felix Tripier et al. *Fault-Tolerant Quantum Computing with Trapped Ions: The Walking Cat Architecture*. 2026. arXiv: 2604.19481. Pre-published.

[WBC+21] Yulin Wu et al. "Strong Quantum Computational Advantage Using a Superconducting Quantum Processor". In: *Phys. Rev. Lett.* 127.18 (2021), p. 180501.

[Wil16] Ryan Williams. "Strong ETH Breaks With Merlin and Arthur: Short Non-Interactive Proofs of Batch Evaluation". 2016. arXiv: 1601.04743 [cs.CC].

[Wil18] R. Ryan Williams. "Counting Solutions to Polynomial Systems via Reductions". In: *1st Symp. Simplicity Algorithms SOSA 2018*. Ed. by Raimund Seidel. Vol. 61. Open Access Series in Informatics (OASIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 6:1–6:15.

[Zha25] Yuxuan Zhang. “Complexity and Hardness of Random Peaked Circuits”. 2025. arXiv: 2510.00132 [quant-ph].

A Attack on block-diagonal structure

Suppose we have tensors $T_i \in U_i \otimes V_i \otimes W_i$, $i \in [s]$. Let $U = \oplus_i U_i$, $V = \oplus_i V_i$, and $W = \oplus_i W_i$. We can then form the direct sum of T_i as $\oplus_i T_i \in U \otimes V \otimes W = (\oplus_i U_i) \otimes (\oplus_i V_i) \otimes (\oplus_i W_i)$. Such a block-diagonal tensor $\oplus_i T_i$ can then be transformed to T via $\mathrm{GL}(U) \times \mathrm{GL}(V) \times \mathrm{GL}(W)$. Therefore, T is isomorphic to a block-diagonal tensor $\oplus_i T_i$, and our task is to compute the transformation that reveals this block-diagonal structure.

This can be done via some known procedures in computer algebra. In the following, by slight abuse of notation, we assume that T is a trilinear form $T : U \times V \times W \to \mathbb{F}$. For $u \in U$, let $T_u : V \times W \to \mathbb{F}$ be the bilinear form by specifying the first argument of T to u. Similarly, for $v \in V$ and $w \in W$, we can define T_v and T_w analogously.

First, we may assume that T (and therefore $\oplus_i T_i$) is non-degenerate (or concise by Strassen). Recall that T is degenerate, if there exists $x = u \in U$, or $x = v \in V$, or $x = w \in W$, such that T_x is the zero form.

Second, for a non-degenerate T, its centroid algebra $\mathrm{Cent}(T)$ consists of $(A, B, C) \in \mathrm{End}(U) \oplus \mathrm{End}(V) \oplus \mathrm{End}(W)$, such that for any $u \in U$, $v \in V$, and $w \in W$, $T(Au, v, w) = T(u, Bv, w) = T(u, v, Cw)$. Here, $\mathrm{End}(U)$ denotes the endomorphism algebra of U. Note that a linear basis of $\mathrm{Cent}(T)$ can be computed efficiently. It can be shown that $\mathrm{Cent}(T)$ is a commutative algebra, and T is isomorphic to a block-diagonal tensor if and only if $\mathrm{Cent}(T)$ contains a non-trivial idempotent, that is, $(P, Q, R) \in \mathrm{Cent}(T)$ such that $P^2 = P$, $Q^2 = Q$, and $R^2 = R$.

Third, we are reduced to the question of deciding whether $\mathrm{Cent}(T)$ contains a non-trivial idempotent. This can be solved by first computing the algebra structure of $\mathrm{Cent}(T)$, namely its Jacobson radical and the semisimple quotient decomposition. The non-trivial idempotent task is then solved by first working this problem out in the simple components, and if every simple component admits a non-trivial idempotent, gluing the solutions and lifting through the radical.