Sample complexity of hidden subgroup problem

Zekun Ye
Institute of Quantum Computing and Computer Science Theory, School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510060, China
yezekun@mail2.sysu.edu.cn

Lvzhou Li
Institute of Quantum Computing and Computer Science Theory, School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
Ministry of Education Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Guangzhou 510006, China
lilvzh@mail.sysu.edu.cn

Abstract

The hidden subgroup problem (HSP) has been attracting much attention in quantum computing, since several well-known quantum algorithms including Shor algorithm can be described in a uniform framework as quantum methods to address different instances of it. One of the central issues about HSP is to characterize its quantum/classical complexity. For example, from the viewpoint of learning theory, sample complexity is a crucial concept. However, while the quantum sample complexity of the problem has been studied, a full characterization of the classical sample complexity of HSP seems to be absent, which will thus be the topic in this paper. HSP over a finite group is defined as follows: For a finite group $G$ and a finite set $V$, given a function $f: G \rightarrow V$ and the promise that for any $x, y \in G$, $f(x) = f(xy)$ iff $y \in H$ for a subgroup $H \in \mathcal{H}$, where $\mathcal{H}$ is a set of candidate subgroups of $G$, the goal is to identify $H$. Our contributions are as follows:

i) For HSP, we show that the number of uniform examples necessary to learn the hidden subgroup with bounded error is at least $\Omega\left(\max \min_{H \in \mathcal{H}} \left\{ \log |H|, \frac{|G|}{|H|} \right\} \right)$, and on the other hand, $O\left(\max_{H \in \mathcal{H}} \left\{ r(H), \sqrt{\frac{|G|}{|H|} \log |H|}\right\} \right)$ uniform examples are sufficient, where $r(H) = \max_{H \in \mathcal{H}} r(H)$ and $r(H)$ is the rank of $H$.

ii) By concretizing the parameters of HSP, we consider a class of restricted Abelian hidden subgroup problem (rAHSP) and obtain the upper and lower bounds for the sample complexity of rAHSP.

iii) We continue to discuss a special case of rAHSP, generalized Simon’s problem (GSP), and show that the sample complexity of GSP is $\Theta\left(\max \left\{ k, \sqrt{k \cdot p^{n-k}}\right\} \right)$. Thus we obtain a complete characterization of the sample complexity of GSP.

2012 ACM Subject Classification Theory of computation

Keywords and phrases hidden subgroup problem, sample complexity, finite group, quantum computing

Digital Object Identifier 10.4230/LIPIcs...

Category Track A: Algorithms, Complexity and Games

1 Introduction

1.1 Background

Hidden subgroup problem. The hidden subgroup problem plays an important role in the history of quantum computing. Several important quantum algorithms such as Deutsch-Jozsa algorithm [17], Simon algorithm [43], and Shor algorithm [42] have a uniform description in the framework of the hidden subgroup problem [30]. Moreover, many quantum algorithms were proposed for the instances of the hidden subgroup problem, e.g., [5, 12, 19, 21, 24, 31, 34].
Sample complexity of hidden subgroup problem

The hidden subgroup problem consists of the Abelian hidden subgroup problem and the non-Abelian hidden subgroup problem. Many problems are special cases of the Abelian hidden subgroup problem, such as Simon’s problem [43], generalized Simon’s problem [47], and some important number-theoretic problems [23, 22, 41]. The non-Abelian hidden subgroup problem also received much attention [21, 22, 34, 20, 24, 35, 40]. While there exist efficient quantum algorithms to solve the Abelian hidden subgroup problem [15, 11, 32, 8, 37, 18], many instances of the non-Abelian hidden subgroup problem are not known to have efficient quantum algorithms, such as the dihedral hidden subgroup problem and the symmetric hidden subgroup problem [34, 16].

There exist two versions of the hidden subgroup problem: the identification and decision versions. The task of the identification version is to identify the hidden subgroup, whereas the task of the decision version is to decide whether the hidden subgroup is the trivial group or not. In this paper, all the hidden subgroup problems mentioned belong to the identification version without special instructions.

Sample and query complexity. In learning theory, there are two types of learning models: passive learning and active learning [25, 6]. In passive learning, the algorithm can only receive random labeled examples in a passive way; in active learning, the algorithm can interactively ask for the labels of examples of its own choosing. Thus, active learning may enable us to design more powerful algorithms compared to passive learning for the same problem.

Specifically, we will focus on the task to learn the property about some function $f$ with high success probability in the following. In passive learning, an algorithm can obtain i.i.d. random labeled examples $(x, f(x))$, where $x$ is distributed according to a certain probability distribution. Such an algorithm is called a sample algorithm. The sample complexity of a sample algorithm is the maximum number of i.i.d. random examples needed for learning the property about $f$ in the worst case. On the other hand, in active learning, an algorithm is allowed to make queries. The algorithm can choose some $x$ to learn $f(x)$ in each query. Such an algorithm is called a query algorithm. The query complexity of a query algorithm is the maximum number of queries needed for learning the property about $f$ in the worst case. The sample (query) complexity of a learning problem is the sample (query) complexity of the optimal sample (query) algorithm.

A query algorithm may make one query at a time, using the information from previous queries to decide which example to query next. Thus, a query algorithm may cost less than a sample algorithm. That is, an upper bound on the sample complexity is always an upper bound on the query complexity for the same problem, but a lower bound on the sample complexity is not necessarily a lower bound on the query complexity.

Much work analyzed the quantum advantage via the sample complexity [2]. For example, the concept class of DNF-formulas and $(\log n)$-juntas can be learned in polynomial time from quantum examples under the uniform distribution [9, 1], whereas the best known classical algorithm of these two problems both runs in quasi-polynomial time under the uniform examples [45, 38]. Arunachalam and de Wolf [8] proved that quantum and classical sample complexity are equal up to constant factors in both the PAC and agnostic models. Arunachalam et al. [1] showed a $k$-Fourier-sparse $n$-bit Boolean function can be learned from $O(k^{1.5}(\log k)^2)$ uniform quantum examples for that function, whereas $\Omega(nk)$ uniform examples are necessary in the classical case [20].
Sample complexity of the hidden subgroup problem. The quantum sample complexity of the hidden subgroup problem has been attracting great attention. Bacon et al. [5] presented the quantum sample complexity of the hidden subgroup problem over semidirect product group $A \times \mathbb{Z}_p$ is $\Theta(\log |A|/\log p)$, where $A$ is any Abelian group and $p$ is a prime. Ettinger et al. [19] proved that the quantum sample complexity of the hidden subgroup problem over any finite group $G$ is $O(\log^2 |G|)$. In terms of the candidate set $\mathcal{H}$ of hidden subgroups, Moore and Russell [36] showed the quantum sample complexity of a wide class of the hidden subgroup problem is $O(\log |\mathcal{H}|)$. Furthermore, Hayashi et al. [27] proved that the quantum sample complexity of the hidden subgroup problem is at most $O(\log |\mathcal{H}| / \log \min_{H \neq H' \in \mathcal{H}} (|H|/|H\cap H'|))$ and at least $\Omega(\log |\mathcal{H}| / \log \max_{H \in \mathcal{H}} |H|)$; if all the candidate subgroups in $\mathcal{H}$ have the same prime order $p$, then the quantum sample complexity is $\Theta(\log |\mathcal{H}| / \log p)$.

On the other hand, to our knowledge, almost no related direct result has been obtained in terms of the classical sample complexity of the hidden subgroup problem. However, there exists only some discussion about the classical query complexity for some instances of the hidden subgroup problem. For example, the classical query complexity of Simon’s problem was proven to be $\Theta(\sqrt{2^n})$ [43, 19, 15]. Ye et al. [47] proved that a nearly optimal bound for the classical query complexity of generalized Simon’s problem (GSP). For the order-finding problem over $\mathbb{Z}_2^m \times \mathbb{Z}_2^n$, Cleve [13] proved that the deterministic query complexity is at least $\Omega\left(\sqrt{\frac{2^n}{m}}\right)$, and the bounded-error query complexity is at least $\Omega\left(\frac{2^n/2}{m}\right)$. Kuperberg [34] proved the classical query complexity of the dihedral hidden subgroup problem over the dihedral group $D_n$ is $\Omega(\sqrt{N})$. Childs [11] showed that a classical algorithm must make $\Omega(\sqrt{N})$ queries if there are $N$ candidate subgroups whose only common element is the identity element. Recently, Nayak [39] proposed the deterministic query algorithms for solving the hidden subgroup problem.

It is worth noting that a lower bound on the classical query complexity is also a lower bound on the classical sample complexity, but not necessarily a tight lower bound. For example, for GSP, its classical sample complexity will be shown to be $\Theta\left(\max\left\{k, \sqrt{k \cdot p^{n-1}}\right\}\right)$ in this paper, whereas the known best lower bound on the classical query complexity was given as $\Omega\left(\max\left\{k, \sqrt{p^{n-k}}\right\}\right)$ in [47].

Motivation. The motivation for studying the classical sample complexity of the hidden subgroup problem is as follows: (i) Note that there exists a great understanding of the quantum sample complexity for the hidden subgroup problem. However, as far as we know, how well do classical sample algorithms perform on this problem still needs to be explored. (ii) For some instances of the hidden subgroup problem, such as GSP, the classical query complexity is not tight [47]. Due to the difficulty in exploring query complexity, we hope to obtain a better lower bound on the classical sample complexity of GSP, since the sample model is weaker than the query model.

1.2 Problem statement and our results

In this paper, we consider the classical sample complexity of the hidden subgroup problem (HSP) over any finite group. The definition of HSP is as follows:

Definition 1 (HSP).
Sample complexity of hidden subgroup problem

**Given:** A finite group $G$; a set $\mathcal{H}$ of candidate subgroups of $G$; an (unknown) function $f : G \rightarrow V$, where $V$ is a finite set.

**Promise:** There exists a subgroup $H \in \mathcal{H}$ such that for any $x, y \in G$, $f(x) = f(xy)$ iff $y \in H$.

**Problem:** Identify $H \in \mathcal{H}$.

Unlike the general definition, we explicitly give the candidate subgroups set $\mathcal{H}$, which is a critical component in the hidden subgroup problem. Actually, the hidden subgroup problem only depends on $G$ and $\mathcal{H}$ essentially.

Moreover, we consider some interesting instances of HSP further by giving more concrete parameters. First, we discuss an instance of the hidden subgroup problem over a class of Abelian groups, call the restricted Abelian hidden subgroup problem ($r\text{AHSP}$). The definition of $r\text{AHSP}$ is as follows:

**Definition 2 ($r\text{AHSP}$).**

**Given:** An (unknown) function $f : G \rightarrow V$, where $G = \mathbb{Z}_{p_1}^{n_1} \times \mathbb{Z}_{p_2}^{n_2} \times \cdots \times \mathbb{Z}_{p_m}^{n_m}$ and $p_i$’s are primes; positive integers $k_1, \ldots, k_m$ satisfying that $k_i < n_i$ for any $i \in [m]$.

**Promise:** There exists a subgroup $H$ such that (i) $H = H_1 \times H_2 \times \cdots \times H_m$; (ii) $\text{rank}(H_i) = k_i$ for any $i \in [m]$; (iii) for any $x, y \in G$, $f(x) = f(x + y)$ iff $y \in H$.

**Problem:** Identify $H$.

It is easy to see $r\text{AHSP}$ is a subproblem of HSP, since $\mathcal{H}$ is the set of all the subgroups satisfying the above promise. Note that any candidate subgroup in $\mathcal{H}$ has the same order in $r\text{AHSP}$.

Furthermore, we continue to consider a simplified version of $r\text{AHSP}$, generalized Simon’s problem (GSP), defined as follows:

**Definition 3 (GSP)**

**Given:** An (unknown) function $f : \mathbb{Z}_p^n \rightarrow V$, where $p$ is a prime, $V$ is a finite set; a positive integer $k < n$.

**Promise:** There exists a subgroup $H \leq \mathbb{Z}_p^n$ of rank $k$ such that for any $x, y \in \mathbb{Z}_p^n$, $f(x) = f(y)$ iff $y - x \in H$.

**Problem:** Identify $H$.

In GSP, $G = \mathbb{Z}_p^n$, $\mathcal{H}$ is the set of all the subgroups of rank $k$ in $G$. Additionally, GSP is an extension version of Simon’s problem, which is a well-known problem in the history of quantum computing. Simon’s problem is a special case of GSP with $k = 1$ and $p = 2$, as shown in Definition 4. Similarly, we can express Simon’s problem as an instance of HSP by making $G = \mathbb{Z}_2^n$, $\mathcal{H} = \{\{0, s\} | s \in \{0, 1\}^n / \{0\}\}$.
Definition 4 (Simon’s problem).

**Given:** An (unknown) function \( f : \mathbb{Z}_2^n \rightarrow V \), where \( V \) is a finite set.

**Promise:** There exists a non-zero string \( s \in \{0,1\}^n \) such that for any \( x, y \in \mathbb{Z}_2^n \),
\[ f(x) = f(y) \iff x = y \text{ or } x + y = s. \]

**Problem:** Identify \( s \).

The above problems have the following relationship:

\[
\begin{align*}
\text{HSP} & \quad \rightarrow \quad \text{rAHSP} \quad \rightarrow \quad \text{GSP} \quad \rightarrow \quad \text{Simon’s problem}
\end{align*}
\]

**Figure 1** The relations between the above problems. A rightward arrow from \( A \) to \( B \) means \( B \) is a subproblem of \( A \).

For the above problems, given enough i.i.d. examples, a learning algorithm may give correct answers with bounded error \( \delta \) (\( 0 \leq \delta < 1/2 \)). In this case, we define the sample complexity of the problem as the number of examples needed by the optimal learning algorithm in the worst case. In the classical case, an example has the form \( (x, f(x)) \), where \( x \) is distributed according to a given distribution over \( G \). If \( x \) is assumed to follow the uniform distribution, then \( (x, f(x)) \) is called a uniform example. In the quantum case, a uniform quantum example is such a quantum state
\[
\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |f(x)\rangle.
\]
In this paper, we focus on the classical case, and our problem is: what number of uniform examples is sufficient and necessary to learn the goal with bounded error in the above problems?

We first obtain some characterizations for the sample complexity of HSP. Let \( sr(H) = \max_{H \in \mathcal{H}} \min\{|S| : S \subseteq H, \langle S \rangle = H\} \). Our main result is as follows:

**Theorem 5.** For HSP, the number of uniform examples necessary to learn the hidden subgroup with bounded error is at least \( \Omega \left( \max\left\{ \min_{H \in \mathcal{H}} \frac{\log |H|}{\log \frac{|G|}{|H|}}, \min_{H \in \mathcal{H}} \frac{|G| \log |H|}{|H|} \right\} \right) \). On the other hand, the number of uniform examples sufficient to learn the hidden subgroup with bounded error is at most \( O \left( \max_{H \in \mathcal{H}} \left\{ sr(H), \sqrt{\frac{|G|}{|H|}} sr(H) \right\} \right) \).

We also analyze the sample complexity of rAHSP and GSP. By Theorem 5, we obtain the following corollaries:

**Corollary 6.** For rAHSP, the number of uniform examples necessary to learn the hidden subgroup with bounded error is at least \( \Omega \left( \max_{i \in [m]} \left\{ k_i, \min_{i \in [m]} k_i \prod_{j=1}^{m} p_j^{n_j-k_j} \right\} \right) \). Moreover, the number of uniform examples sufficient to learn the hidden subgroup with bounded error is at most \( O \left( \max_{i \in [m]} \left\{ k_i, \sqrt{k_i \prod_{j=1}^{m} p_j^{n_j-k_j}} \right\} \right) \).

**Corollary 7.** The sample complexity of GSP is \( \Theta \left( \max \left\{ k, \sqrt{k \cdot p^{n-k}} \right\} \right) \).

Additionally, the sample complexity of Simon problem is a well-known result as Claim 8 (e.g. [43, 15]). Corollary 7 matches with Claim 8 when \( k = 1 \) and \( p = 2 \).

**Claim 8.** The sample complexity of Simon’s problem is \( \Theta(\sqrt{2^n}) \).
**Sample complexity of hidden subgroup problem**

We list lower bounds on the sample complexity of Simon’s problem, GSP, HSP in Table 1. The methods in this paper may be helpful to other problems.

| Table 1 | Known results about the sample complexity of Simon’s problem, GSP and HSP. The results of GSP and HSP are obtained in this paper. |
|---|---|
| Simon’s problem | GSP | HSP |
| classical | \(\Theta(\sqrt{p})\) | \(\Theta(\max(k, \sqrt{p^{1-\epsilon}})}\) | \(O\left(\max\left\{\sqrt{p}, \sqrt{\frac{\log|H|}{p}}\right\}\right)\) |
| quantum | \(\Theta(1)\) | \(\Theta(n-k)\) | \(O\left(\frac{\log|H|}{p} \log\log|H|\right)\) |

1.3 Organization

The remainder of the paper is organized as follows. In Section 2, we review some notations used in this paper. In Section 3, we present the lower and upper bounds of sample complexity of HSP. In Section 4, we apply the result in Section 3 to obtain the sample complexity of some special problems, including rAHSP and GSP. Finally, a conclusion is made in Section 5.

2 Preliminary

In this section, we present some notations used in this paper. Let \([m]=\{1, 2, \ldots, m\}\) and \(\mathbb{Z}_p\) denote the additive group of elements \(\{0, 1, \ldots, p-1\}\) with addition modulo \(p\) denoted by \(+\). For two groups \(G_1, G_2\), let \(G_1 \times G_2\) denote the direct product of \(G_1\) and \(G_2\). For a finite group \(G\), a subset \(S\) is said to be a generating set for \(G\) if all elements in \(G\) can be expressed as the finite product of elements in \(S\) and their inverses, i.e., \(G = \langle S \rangle = \{a_1^{l_1}a_2^{l_2} \cdots a_k^{l_k}| a_i \in S, l_i = \pm 1, k \in \mathbb{N}\}\). The rank \(r(G)\) of \(G\) is the cardinality of a minimal generating set of \(G\). For a subset \(H\) of \(G\), then \(H \leq G\); if \(H\) is a proper subgroup of \(G\), then \(H < G\). Note that if \(H\) is a subgroup of \(\mathbb{Z}_p^n\), then \(r(H) = k\) if and only if \(|H| = p^k\). For a set \(H\) consisting of subgroups of \(G\), the **subgroup rank** \(r(H)\) of \(H\) is defined as \(sr(H) = \max_{H \in H} r(H)\). The group with only one element, the identity element, is called the **trivial group**.

Suppose \(X, Y, Z\) are discrete random variables. If \(X \sim p(x)\), then the Shannon entropy associated with \(X\) is defined as \(I(X) = -\sum_x p(x) \log p(x)\). If \((X, Y) \sim p(x, y)\), then the **joint entropy** of \(X\) and \(Y\) is defined as \(I(X, Y) = -\sum_{x,y} p(x, y) \log p(x, y)\); the **entropy of \(X\)** conditional on knowing \(Y\) is defined as \(I(X|Y) = I(X, Y) - I(Y)\); the **mutual information** of \(X\) and \(Y\) is defined as \(I(X : Y) = I(X) + I(Y) - I(X, Y)\); the **conditional mutual information** of \(X\) and \(Y\) conditional on knowing \(Z\) is defined by \(I(X : Y|Z) = I(X|Z) - I(X|Y, Z)\). The **binary entropy** of a bit with distribution \((p, 1-p)\) is defined as \(I(p) = -p \log p - (1-p) \log(1-p)\). Some basic properties of Shannon entropy [14] are useful in this paper:

\[
I(X_1, X_2, \ldots, X_n) \leq \sum_{i=1}^n I(X_i) \quad \text{with equality if and only if } X_i \text{'s are independent random variables.}
\]

1. First, Simon algorithm [43] is a sample algorithm with \(O(n)\) uniform examples. Second, Koiran et al. [33] presented the lower bound on the quantum query complexity of Simon’s problem is \(\Omega(n)\), so this is also a lower bound on the quantum sample complexity of Simon’s problem.

2. GSP can be solved with a generalized Simon’s algorithm with \(O(n-k)\) uniform examples [28]. Additionally, by substituting \(|H| = \prod_{i=0}^{l-1} \frac{\log p}{p} \text{ and } |H| = p^k \text{ (} \forall H \in \mathcal{H}\) into \(\Omega\left(\frac{\log|H|}{p} \log\log|H|\right)\) [27], we can see the lower bound on the quantum sample complexity of GSP is \(\Omega(n-k)\).
First, we prove the left part of the lower bound using the following three-step analysis: Combining these three steps implies

\[ \max_{H \in \mathcal{H}} \log \left( \frac{|G| \log |H|}{|H|^2} \right) \]

For \( \mathcal{H} \), let \( \mathcal{H} \) be the random variable of the hidden subgroup that is uniformly distributed over \( \mathcal{H} \). For a sample algorithm of \( \mathcal{H} \), suppose the number of i.i.d. uniform examples is \( T \). Let \( B_i = (X_i, f(X_i)) \) be the random variable of the \( i \)-th uniform examples for \( i \in [T] \). Let \( B = B_1 \cdots B_T \), \( X = X_1 \cdots X_T \) and \( f(X) = f(X_1) \cdots f(X_T) \). Suppose the range of \( B \) is \( B \). For \( i, j \in [T] \), we define random variable \( Y_{X_i, X_j} \) as follows: if \( f(X_i) = f(X_j) \), let \( Y_{X_i, X_j} = 1 \); if \( f(X_i) \neq f(X_j) \), let \( Y_{X_i, X_j} = 0 \). Let \( Y \) be the sequence of \( Y_{X_i, X_j} \) for any \( i < j \). Since \( Y \) is a function of \( B \), without loss of generality, we assume \( g(B) = Y \).

3 General Bounds for \( \mathcal{H} \)

In this section, we present the general bounds for the sample complexity of \( \mathcal{H} \) in Theorem 5. We show lower and upper bounds in Section 3.1 and 3.2, respectively.

3.1 Lower bound

In this section, we present a lower bound by proving Theorem 9. We use an information-theoretic method.

**Theorem 9** (Lower bound). The number of uniform examples necessary to solve \( \mathcal{H} \) with bounded error is at least \( \Omega \left( \max \left\{ \min_{H \in \mathcal{H}} \log \frac{|H|}{|H|^2}, \min_{H \in \mathcal{H}} \left( \frac{|G| \log |H|}{|H|^2} \right) \right\} \right) \).

**Proof.** First, we prove the left part of the lower bound using the following three-step analysis:

1. \( I(f(X)) \geq (1 - \delta) \log |\mathcal{H}| - H(\delta) \).

   **Proof of item 1.** Since
   \[
   I(\mathcal{H} : B) \leq I(\mathcal{H} : X) + I(\mathcal{H} : f(X)) = 0 + I(\mathcal{H} : f(X)) \leq I(f(X)),
   \]
   we have \( I(f(X)) \geq I(\mathcal{H} : B) > (1 - \delta) \log |\mathcal{H}| - I(\delta) \) by Lemma 10.

2. \( I(f(X)) \leq \sum_{i \in I} I(f(X_i)) \).

3. \( I(f(X_i)) \leq \max_{H \in \mathcal{H}} \log \left( \frac{|G|}{|H|^2} \right) \) for any \( i \).

**Proof of item 3.** The function value of \( f \) is constant on cosets of \( H \) and distinct among different cosets of \( H \), thus \( f(X_i) \) is a uniform variable over \( \frac{|G|}{|H|^2} \) different values. In the worst case, \( H \) is the largest subgroup of \( G \). Thus, \( I(f(X_i)) \leq \max_{H \in \mathcal{H}} \log \left( \frac{|G|}{|H|^2} \right) \).

Combining these three steps implies
\[
T > \frac{(1 - \delta) \log |\mathcal{H}| - I(\delta)}{\max_{H \in \mathcal{H}} \log \left( \frac{|G|}{|H|^2} \right)} = \min_{H \in \mathcal{H}} \left( \frac{(1 - \delta) \log |\mathcal{H}| - I(\delta)}{\log \left( \frac{|G|}{|H|^2} \right)} \right) = \Omega \left( \min_{H \in \mathcal{H}} \log \left( \frac{|G|}{|H|^2} \right) \right).
\]
Second, we continue to prove the right part using a similar method.

1. \( I(Y) \geq (1 - \delta)k \cdot \log |\mathcal{H}| - H(\delta) \).
   
   **Proof of item 1.** By the definition of \( B \) and \( Y \), we have
   \[
   \Pr\{B = b | Y = y\} = \begin{cases} \frac{1}{|\{b \in B : g(b) = y\}|}, & \text{if } g(b) = y \\ 0, & \text{if } g(b) \neq y \end{cases}.
   \]
   Thus, given \( Y \), \( B \) is independent of \( \mathcal{H} \), i.e., \( Y \) is a sufficient statistic of \( \mathcal{H} \), which means \( I(\mathcal{H} : B | Y) = 0 \). As a result, \( I(\mathcal{H} : Y) \geq I(\mathcal{H} : B) \). Hence, \( I(Y) \geq I(\mathcal{H} : Y) \geq I(\mathcal{H} : B) > (1 - \delta)k \cdot \log |\mathcal{H}| - I(\delta) \) by Lemma 10.

2. \( I(Y) \leq \sum_{i<j} I(Y_{i},X_{j}) \).

3. \( I(Y_{i},X_{j}) \leq \max_{H \in \mathcal{H}} 2\frac{|H|}{|G|} \log \frac{|G|}{|H|} \).
   
   **Proof of item 3.** By definition of HSP, \( Y_{i},X_{j} = 1 \) iff \( X_{i}^{-1}X_{j} \in H \). Since \( X_{i},X_{j} \) are independent uniform variables over \( G \), \( (X_{i})^{-1}X_{j} \) is also a uniform variable over \( G \). Hence,
   \[
   \Pr\{Y_{i},X_{j} = 1\} = \Pr\{(X_{i})^{-1}X_{j} \in H\} = \frac{|H|}{|G|},
   \]
   so
   \[
   I(Y_{i},X_{j}) \leq \max_{H \in \mathcal{H}} I\left(\frac{|H|}{|G|}\right) \leq \max_{H \in \mathcal{H}} 2\frac{|H|}{|G|} \log \frac{|G|}{|H|},
   \]
   where the last inequality follows by Claim 11.

Combining these three steps implies
\[
\binom{T}{2} \geq \frac{(1 - \delta) \cdot \log |\mathcal{H}| - H(\delta)}{\max_{H \in \mathcal{H}} 2\frac{|H|}{|G|} \log \frac{|G|}{|H|}},
\]
which means
\[
T = \Omega\left(\min_{H \in \mathcal{H}} \frac{|G| \log |\mathcal{H}|}{|H| \log \frac{|G|}{|H|}}\right).
\]
Finally, we have \( \Omega\left(\max\left\{ \min_{H \in \mathcal{H}} \frac{\log |\mathcal{H}|}{\log \frac{|G|}{|H|}}, \min_{H \in \mathcal{H}} \sqrt{\frac{|G| \log |\mathcal{H}|}{|H| \log \frac{|G|}{|H|}}} \right\}\right) \).

**Lemma 10.** \( I(\mathcal{H} : B) > (1 - \delta) \log |\mathcal{H}| - H(\delta) \).

**Proof.** Let random variable \( \mathcal{H}_{B} \) be the hypothesis that the learner produces (given \( B \)). According to the setting of learning algorithms, it is required that \( \Pr\{\mathcal{H} \neq \mathcal{H}_{B}\} \leq \delta \). By Fano inequality [14], we have \( I(\mathcal{H} | B) \leq I(\delta) + \delta \log(|\mathcal{H}| - 1) \). Thus,
\[
I(\mathcal{H} : B) = I(\mathcal{H}) - I(\mathcal{H} | B) \\
\geq \log |\mathcal{H}| - (I(\delta) + \delta \log(|\mathcal{H}| - 1)) \\\n> (1 - \delta) \log |\mathcal{H}| - I(\delta).
\]

**Claim 11.** For \( 0 < p \leq \frac{1}{2} \), \(- (1 - p) \log(1 - p) \leq -p \log p\).
Proof. Let $f(p) = p \ln p - (1 - p) \ln(1 - p)$, then $f'(p) = \ln p + \ln(1 - p) + 2$, so $f''(p) = \frac{2p - 1}{p(1 - p)^2}$. When $0 < p < \frac{1}{2}$, $f''(p) \geq 0$ and thus $f'(p)$ is an increasing function. Because $\lim_{x \to 0} f'(x) < 0$ and $f'(\frac{1}{2}) > 0$, there exists a point $p_0$ ($0 < p_0 < \frac{1}{2}$) such that $f'(p_0) = 0$. Thus, $f(p)$ is a decreasing function when $0 < p \leq p_0$ and $f(p)$ is an increasing function when $p_0 \leq p < \frac{1}{2}$. Since $\lim_{x \to 0} f(x) = 0$ and $f(\frac{1}{2}) = 0$, we have $f(p) \leq 0$ for any $0 < p \leq \frac{1}{2}$, i.e., $(1 - p) \ln(1 - p) \geq p \ln p$. Thus, $-(1 - p) \log(1 - p) \leq -p \log p$.

3.2 Upper bound

In this section, we give an upper bound on the sample complexity of HSP by proving Theorem 12. Specifically, we propose Algorithm 1 to solve HSP. Then analyze the correctness and sample complexity of Algorithm 1 in Section 3.2.1 and 3.2.2 respectively. The number of examples used in Algorithm 1 is an upper bound on the sample complexity of HSP.

**Theorem 12 (Upper bound).** The number of uniform examples sufficient to solve HSP with bounded error is $O\left(\max\left\{sr(H), \sqrt{\frac{\max |G|}{|H|}sr(H)}\right\}\right)$.

Algorithm 1 is shown as follows. In Algorithm 1 if $\max_{H \in H} \frac{|G|}{|H|} > sr(H)$, let $A = \left[9 \sqrt{\frac{\max |G|}{|H|}sr(H)}\right]$. and $B = \left\lceil\max_{H \in H} \frac{|G|}{|H|}\right\rceil$. if $\max_{H \in H} \frac{|G|}{|H|} \leq sr(H)$, let $A = 9 \max_{H \in H} \frac{|G|}{|H|}$ and $B = 1$. In this way, we always have $AB \geq 9 \max_{H \in H} \frac{|G|}{|H|}$. If two examples $(a, f(a))$ and $(b, f(b))$ satisfies $f(a) = f(b)$, then we say these two examples collide and call $(a, b)$ a collision pair. Furthermore, if there exists at least a collision pair between two example sets $P$ and $Q$, then we also say that $P$ and $Q$ collide.

**Algorithm 1** The sample algorithm of HSP

1. Let $W = \emptyset$.
2. Sample $A$ times to obtain an example set $P$.
3. For $1 \leq i \leq 9 \cdot sr(H)$, sample $B$ times to obtain an example set $Q_i$. If $P$ and $Q_i$ collide, then we randomly select a collision pair $(a_i, b_i)$ such that $(a_i, f(a_i)) \in P$, $(b_i, f(b_i)) \in Q_i$, and insert $a_i^{-1}b_i$ into set $W$.
4. Repeat Step 2-3 $\left\lceil\frac{\ln \frac{1}{\delta}}{\ln 9}\right\rceil$ times, return $\langle W \rangle$.

3.2.1 Correctness Analysis

In this section, Our goal is to prove that the probability of Algorithm 1 failing is no more than $\delta$. By the definition of HSP, we have $f(a_i) = f(b_i)$ if and only if $a_i^{-1}b_i \in H$ for any $i$. Thus, any element added into $W$ in Step 3 is an element in $H$. In the following, it suffices to prove that the probability that $W$ is a generating set of $H$ is not less than $1 - \delta$.

Let $N = \frac{|G|}{|H|}$. We call each execution of Step 2-3 an iteration. In Step 2, let $\zeta_{i,j,l}$ be the indicator random variable for the event that $P_l$ and $Q_{i,j}$ collide, where $P_l$ is the $l$-th sample in $P$ and $Q_{i,j}$ is the $j$-th sample in $Q_i$. Let $\zeta_i = \sum_{j,l} \zeta_{i,j,l}$. Then $E(\zeta_{i,j,l}) = Pr\{\zeta_{i,j,l} = 1\} = \frac{1}{N}$ and $D(\zeta_{i,j,l}) = \frac{1}{N}(1 - \frac{1}{N})$ for any $i,j,l$. Thus, $E(\zeta_i) = \frac{AB}{N}$. Since $\zeta_{i,j_1,l_1}$ and $\zeta_{i,j_2,l_2}$ are
Sample complexity of hidden subgroup problem

independent for any \((j_1, h_1) \neq (j_2, l_2)\), by Chebyshev’s Inequality, we have
\[
Pr \left\{ \left| \frac{\zeta_i}{AB} - \frac{1}{N} \right| \geq \frac{2}{3} \frac{1}{N} \right\} \leq \frac{D(\frac{1}{AB} \zeta_i)}{(\frac{2}{3} \frac{1}{N})^2} = \frac{D(\zeta_i)}{(9N)^2(\frac{2}{3} \frac{1}{N})^2}
\]
\[
= \frac{\sum_{i,j} D(\zeta_{i,j})}{(9N)^2(\frac{2}{3} \frac{1}{N})^2} = \frac{D(\zeta_{1,1})}{9N(\frac{2}{3} \frac{1}{N})^2} = \frac{1}{9}(1 - \frac{1}{N}) < 1/4.
\]

Since \(AB \geq 9N\),
\[
Pr\{\zeta_i \geq 3 \} \geq Pr\{\zeta_i \geq \frac{AB}{3N}\}
\]
\[
= Pr\{\frac{\zeta_i}{AB} \geq \frac{1}{3} \}
\]
\[
> 1 - \frac{1}{4}
\]
\[
= \frac{3}{4},
\]
so the probability that \(P\) and \(Q_i\) collide is larger than \(\frac{3}{4}\).

Let \(\beta_i\) be the indicator random variable for the event that \(P\) and \(Q_i\) collide. Then
\(E(\beta_i) = \frac{3}{4}\) and \(D(\beta_i) = E(\beta_i)(1 - E(\beta_i))\) for any \(i\). Let \(\beta = \sum_{i=1}^{9sr(H)} \beta_i\). Since \(\beta_i\) and \(\beta_j\) are independent for any \(i \neq j\), by Chebyshev’s Inequality, we have
\[
Pr\{\left| \frac{\beta}{9sr(H)} - E(\beta) \right| \geq \frac{2}{3} E(\beta)\} \leq \frac{D(\frac{9sr(H)}{3} \beta)}{(\frac{2}{3} E(\beta))^2} = \frac{D(\beta)}{(9sr(H))2(\frac{2}{3} E(\beta))^2}
\]
\[
= \frac{\sum_{i=1}^{9sr(H)} D(\beta_i)}{(9sr(H))^2(\frac{2}{3} E(\beta))^2} = \frac{D(\beta)}{(9sr(H))(\frac{2}{3} E(\beta))^2}
\]
\[
= \frac{(1 - E(\beta))}{(9sr(H))(\frac{2}{3} E(\beta))} < \frac{1}{3sr(H)}.
\]

Thus,
\[
Pr\{\beta \geq r(H)\} \geq Pr\{\beta \geq \frac{9sr(H)}{4}\}
\]
\[
\geq Pr\{\beta \geq \frac{9sr(H)}{4} \cdot E(\beta)\}
\]
\[
= Pr\{\beta \geq \frac{9sr(H)}{3} \}
\]
\[
\geq Pr\{\frac{\beta}{9sr(H)} \geq \frac{1}{3} E(\beta)\}
\]
\[
> 1 - \frac{1}{3sr(H)},
\]
which means the probability that Algorithm 1 finds at least \(r(H)\) elements in \(H\) is at least \(1 - \frac{1}{3sr(H)}\) in each iteration.

Let \(E\) be the event that \(r(H)\) independent elements in \(H\) construct a generating set of \(H\).

Let \(H_i\) be a subgroup of \(H\) generated by the first \(i\) elements \((1 \leq i \leq r(H))\) and \(H_0\) be the trivial subgroup. If \(E\) happens, then \(H_{i-1} \subset H_i\) for any \(i\). By Lagrange’s Theorem, we have
\[
\frac{|H_i|}{|H_{i-1}|} \geq 2.\text{ Thus, } \frac{|H_i|}{|H|} \geq 2^{r(H)-i}.
\]
In this way, the probability that \(E\) happens is
\[
(1 - \frac{|H_0|}{|H|})(1 - \frac{|H_1|}{|H|}) \cdots (1 - \frac{|H_{r(H)-1}|}{|H|}) \geq (1 - \frac{1}{2r(H)}) \cdots (1 - \frac{1}{2r(H)-1}) > \frac{1}{4},
\]
where the last inequality comes from [46]. Therefore, the probability that the elements added into \( W \) in each iterator construct a generating set of \( H \) is at least \((1 - \frac{\log \delta}{\sqrt{n}}) \geq \frac{3}{4} \cdot \frac{1}{9} = \frac{1}{3}\). As a result, after repeating \( \lceil \frac{\ln 2}{\sqrt{\ln n}} \rceil \geq \log_2 \delta \) times, the probability that \( W \) is not a generating set of \( H \) is no more than \((1 - \frac{1}{3})^\log_2 \delta = \delta \), i.e., Algorithm 1 succeeds with probability at least \( 1 - \delta \).

### 3.2.2 Complexity Analysis

The sample complexity of Algorithm 1 is \( A + 9Bsr(\mathcal{H}) \lceil \frac{\ln 2}{\sqrt{\ln n}} \rceil \). If \( \max_{H \in \mathcal{H}} \frac{|G|}{|H|} > sr(\mathcal{H}) \), then

\[
A = \left[ \frac{9}{\sqrt{\max_{H \in \mathcal{H}} \frac{|G|}{|H|} sr(\mathcal{H})}} \right], B = \left[ \frac{1}{\sqrt{\max_{H \in \mathcal{H}} \frac{|G|}{|H|} sr(\mathcal{H})}} \right],
\]

and thus the sample complexity is \( O\left( \sqrt{\max_{H \in \mathcal{H}} \frac{|G|}{|H|} sr(\mathcal{H})} \right) \); if \( \max_{H \in \mathcal{H}} \frac{|G|}{|H|} < sr(\mathcal{H}) \), then \( A = \max_{H \in \mathcal{H}} \frac{|G|}{|H|} \) and \( B = 1 \), and thus the sample complexity is \( O\left( sr(\mathcal{H}) \right) \). Since \( sr(\mathcal{H}) \geq \max_{H \in \mathcal{H}} \frac{|G|}{|H|} \) is equivalent to \( sr(\mathcal{H}) \geq \max_{H \in \mathcal{H}} \frac{|G|}{|H|} \), the sample complexity of Algorithm 1 can be expressed as \( O\left( \max \left\{ sr(\mathcal{H}), \sqrt{\max_{H \in \mathcal{H}} \frac{|G|}{|H|} sr(\mathcal{H})} \right\} \right) \), equivalently, i.e., Theorem 12 is proved.

### 4 Application

The results in Section 3 can be applied to some more specific classes of HSP, including rAHSP and GSP defined in Section 1.2. We show the sample complexity of these problems by proving Corollary 6 and 7.

#### 4.1 Abelian hidden subgroup problem

**Proof of Corollary 6** In rAHSP, \( G = \mathbb{Z}_{p_1}^{n_1} \times \mathbb{Z}_{p_2}^{n_2} \times \cdots \times \mathbb{Z}_{p_m}^{n_m} \), and thus \( |G| = \prod_{i=1}^{m} p_i^{n_i} \). Since \( H = H_1 \times H_2 \times \cdots \times H_m \), where \( H_i \leq \mathbb{Z}_{p_i}^{n_i} \) and \( r(H_i) = k_i \) for any \( i \in [m] \), we have \( |H| = \prod_{i=1}^{m} p_i^{k_i} \). Hence,

\[
\frac{|G|}{|H|} = \prod_{i=1}^{m} \frac{p_i^{n_i}}{p_i^{k_i}} = \prod_{i=1}^{m} p_i^{n_i - k_i}, \tag{1}
\]

so

\[
\log \frac{|G|}{|H|} = \sum_{i=1}^{m} (n_i - k_i) \log p_i.
\]

By a counting method [44], the number of subgroup of rank \( k \) in \( \mathbb{Z}_p^n \) is

\[
\prod_{j=0}^{k-1} \frac{p^n - p^j}{p^k - p^j} > p^{(n-k)k}.
\]

As a result,

\[
|\mathcal{H}| = \prod_{i=1}^{m} \prod_{j=0}^{k_i-1} \frac{p_i^{n_i} - p_i^j}{p_i^{k_i} - p_i^j} > \prod_{i=1}^{m} p_i^{(n_i-k_i)k_i}.
\]
Sample complexity of hidden subgroup problem

Thus

\[ \log |H| > \sum_{i=1}^{m} (n_i - k_i)k_i \log p_i. \]

Therefore,

\[ \frac{\log |H|}{\log |G|} > \frac{\sum_{i=1}^{m} (n_i - k_i)k_i \log p_i}{\sum_{i=1}^{m} (n_i - k_i) \log p_i} \geq \min_{i \in [m]} k_i, \]

(2)

Moreover, by Claim 13, \( r(H) \leq \max_{i \in [m]} k_i \) for any \( H \in \mathcal{H} \), so

\[ sr(H) \leq \max_{i \in [m]} k_i. \]

(3)

By substituting Equations (1)-(3) into Theorem 5 we obtain that the number of uniform examples required for learning the hidden subgroup with bounded error is at least

\[ \Omega \left( \max_{i \in [m]} \left[ k_i \min_{i \in [m]} \sqrt{k_i \prod_{j=1}^{m} n_j^{n_j - k_j}} \right] \right) \]

and at most \( O \left( \max_{i \in [m]} \left[ k_i \sqrt{k_i \prod_{j=1}^{m} n_j^{n_j - k_j}} \right] \right) \) for rHSP.

\[ \triangleright \text{Claim 13.} \text{ For a finite group } G = G_1 \times G_2 \times \cdots \times G_m, r(G) \leq \max_{i \in [m]} r(G_i). \]

Proof. Suppose \( r(G_i) = r_i \) and \( T_i = \{T_{i1}, \ldots, T_{ir_i}\} \) is a generating set of \( G_i \) for \( i \in [m] \). In the following, we try to construct a generating set of \( G \). For \( 1 \leq i \leq m \), let

\[ T'_{ij} = \begin{cases} T'_{ij}, & j \leq r_i \\ e, & r_i < j \leq \max_{i \in [m]} r_i \end{cases}, \]

where \( e \) is the identity element of \( G_i \). For \( 1 \leq j \leq \max_{i \in [m]} r_i \), let \( s_j = (T'_{1j}, \ldots, T'_{mj}) \) and \( S = \{s_1, \ldots, s_{\max_{i \in [m]} r_i}\} \). Let \( T'_i \) denote the set of the \( i \)-th component of the elements in \( S \), i.e., \( T'_i = \{T'_{i1}, \ldots, T'_{i\max_{i \in [m]} r_i}\} \). Since \( T'_i = T_i \cup \{e\} \), \( T'_i \) is also a generating set of \( G_i \). Thus, \( S \) is a generating set of \( G \), which means \( r(G) \leq \max_{i \in [m]} r_i = \max_{i \in [m]} r(G_i) \).

\[ \triangleright \]

4.2 Generalized Simon’s Problem

Proof of Corollary 7. By substituting \( i = 1 \), \( p_1 = p \), \( n_1 = n \), \( k_1 = k \) into Corollary 6 we find that the sample complexity of GSP is at least \( \Omega \left( \max \left\{ k, \sqrt{k \cdot p^n - k} \right\} \right) \) and at most \( O \left( \max \left\{ k, \sqrt{k \cdot p^n - k} \right\} \right) \), i.e., the sample complexity of GSP is \( \Theta \left( \max \left\{ k, \sqrt{k \cdot p^n - k} \right\} \right) \).

\[ \triangleright \]

5 Conclusion

In this paper, we have discussed the classical sample complexity of the hidden subgroup problem (HSP) over finite groups. We have shown the classical sample complexity of HSP is at least \( \Omega \left( \max \left\{ \min_{H \in \mathcal{H}} \log |H| / \log |G|, \min_{H \in \mathcal{H}} |G| \log |H| / \log |H| \right\} \right) \) and at most \( O \left( \max_{H \in \mathcal{H}} \left( sr(H), \sqrt{|G| / |H|} \right) \right) \). Our result may be helpful to clarify the gap between quantum computing and classical computing on this problem. Furthermore, we have applied the result to obtain the sample complexity of some concrete instances of hidden subgroup problem. Particularly, we have obtained a tight bound \( \Theta \left( \max \left\{ k, \sqrt{k \cdot p^n - k} \right\} \right) \) for the sample complexity of GSP. In the future, we will generalize our results to more instances of the hidden subgroup problem, especially for the non-Abelian case. We also believe the information-theoretic approach to obtain the lower bound in this paper will have further application in other learning problems.
References

1. Srinivasan Arunachalam, Sourav Chakraborty, Troy Lee, Manaswi Paraashar, and Ronald de Wolf. Two new results about quantum exact learning. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*, pages 16:1–16:15, 2019. doi:10.4230/LIPIcs.ICALP.2019.16

2. Srinivasan Arunachalam and Ronald de Wolf. Guest column: A survey of quantum learning theory. *SIGACT News*, 48(2):41–67, 2017. doi:10.1145/3106700.3106710

3. Srinivasan Arunachalam and Ronald de Wolf. Optimal quantum sample complexity of learning algorithms. In *Proceedings of the 32nd Computational Complexity Conference*, pages 25:1–25:31, 2017. doi:10.4230/LIPIcs.CCC.2017.25

4. Alp Atici and Rocco A. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2007. doi:10.1007/s11128-007-0061-6

5. Dave Bacon, Andrew M. Childs, and Wim van Dam. From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups. In *Proceedings of the 46th Annual Symposium on Foundations of Computer Science*, pages 469–478, 2005. doi:10.1109/SFCS.2005.38

6. Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2-3):111–139, 2010. doi:10.1007/s10994-010-5174-y

7. Guangya Cai and Daowen Qiu. Optimal separation in exact query complexities for Simon’s problem. *Journal of Computer and System Sciences*, 97:83–93, 2018. doi:10.1016/j.jcss.2018.05.001

8. Andrew M. Childs. Lecture notes on quantum algorithms. https://www.cs.umd.edu/~amchilds/qa/qa.pdf, 2021.

9. Andrew M. Childs and Wim van Dam. Quantum algorithm for a generalized hidden shift problem. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1225–1232, 2007. doi:10.1137/S009753970139450X

10. Richard Cleve. The query complexity of order-finding. *Information and Computation*, 192(2):162–171, 2004. doi:10.1016/j.ic.2004.04.001

11. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2001.

12. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2001.

13. Ronald de Wolf. Quantum Computing: Lecture Notes. *arXiv preprint*, 2019. arXiv:1907.09415

14. Thomas Decker, Gábor Ivanyos, Miklos Santha, and Pawel Wocjan. Hidden symmetry subgroup problems. *SIAM Journal of Computing*, 42(5):1987–2007, 2013. doi:10.1137/120864416

15. David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.

16. Kirsten Eisenträger, Sean Hallgren, Alexei Y. Kitaev, and Fang Song. A quantum algorithm for computing the unit group of an arbitrary degree number field. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 293–302, 2014. doi:10.1145/2591796.2591860

17. Mark Ettinger, Peter Hoyer, and Emanuel Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Information Processing Letters*, 91(1):43–48, 2004. doi:10.1016/j.ipl.2004.01.024
Sample complexity of hidden subgroup problem

20. Katalin Friedl, Gábor Ivanyos, Frédéric Magniez, Miklos Santha, and Pranab Sen. Hidden translation and orbit coset in quantum computing. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 1–9, 2003. \DOI{10.1145/780542.780544}

21. Michelangelo Grigni, Leonard J. Schulman, Monica Vazirani, and Umesh V. Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. *Combinatorica*, 24(1):137–154, 2004. \DOI{10.1007/s00493-004-0009-8}

22. Sean Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 468–474, 2005. \DOI{10.1145/1060590.1060660}

23. Sean Hallgren. Polynomial-time quantum algorithms for Pell’s equation and the principal ideal problem. *Journal of ACM*, 54(1):4:1–4:19, 2007. \DOI{10.1145/1206035.1206039}

24. Sean Hallgren, Alexander Russell, and Amnon Tashma. The hidden subgroup problem and quantum computation using group representations. *SIAM Journal on Computing*, 32(4):916–934, 2003.

25. Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7(2-3):131–309, 2014. \DOI{10.1561/2200000037}

26. Ishay Haviv and Oded Regev. The list-decoding size of Fourier-sparse Boolean functions. *ACM Transactions on Computation Theory*, 8(3):10:1–10:14, 2016. \DOI{10.1145/2898439}

27. Masahito Hayashi, Akinori Kawachi, and Hirotada Kobayashi. Quantum measurements for hidden subgroup problems with optimal sample complexity. *Quantum Information and Computation*, 8(3):345–358, 2008.

28. Mika Hirvensalo. *Quantum Computing*. Springer-Verlag, 2001.

29. Gábor Ivanyos, Frédéric Magniez, and Miklos Santha. Efficient quantum algorithms for some instances of the non-Abelian hidden subgroup problem. In *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 263–270, 2001. \DOI{10.1145/378580.378679}

30. Richard Jozsa. Quantum algorithms and the Fourier transform. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):323–337, 1998.

31. Julia Kempe and Aner Shalev. The hidden subgroup problem and permutation group theory. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1118–1125, 2005.

32. Alexei Y. Kitaev. Quantum measurements and the Abelian stabilizer problem. *arXiv preprint*, 1995. quant-ph/9511026.

33. Cristopher Moore, Daniel N. Rockmore, Alexander Russell, and Leonard J. Schulman. The power of strong Fourier sampling: Quantum algorithms for affine groups and hidden shifts. *SIAM Journal of Computing*, 37(3):938–958, 2007. \DOI{10.1137/S0097539705447177}

34. Cristopher Moore and Alexander Russell. For distinguishing conjugate hidden subgroups, the pretty good measurement is as good as it gets. *Quantum Information and Computation*, 7(8):752–765, 2007.

35. Michele Mosca and Artur Ekert. The hidden subgroup problem and eigenvalue estimation on a quantum computer. In *Proceedings of the 1st NASA International Conference on Quantum Computing and Quantum Communications*, pages 174–188, 1998. \DOI{10.1007/3-540-49208-9_15}
Elchanan Mossel, Ryan O’Donnell, and Rocco A. Servedio. Learning functions of $k$ relevant variables. *Journal of Computer and System Sciences*, 69(3):421–434, 2004. [doi:10.1016/j.jcss.2004.04.002]

Ashwin Nayak. Deterministic algorithms for the hidden subgroup problem. *arXiv preprint*, 2021. [arXiv:2104.14436]

Oded Regev. Quantum computation and lattice problems. *SIAM Journal of Computing*, 33(3):738–760, 2004. [doi:10.1137/S0097539703440678]

Arthur Schmidt and Ulrich Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 475–480, 2005. [doi:10.1145/1060590.1060661]

Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

Daniel R. Simon. On the power of quantum computation. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 116–123, 1994. [doi:10.1109/SFCS.1994.365701]

Richard P. Stanley. *Enumerative Combinatorics, Volume 1*. Cambridge University Press, 2000.

Karsten A. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, pages 314–326, 1990.

John Watrous. Lecture 6: Simon’s algorithm. [https://cs.uwaterloo.ca/~watrous/QC-notes/QC-notes.06.pdf](https://cs.uwaterloo.ca/~watrous/QC-notes/QC-notes.06.pdf), 2006.

Zekun Ye, Yunqi Huang, Lvzhou Li, and Yuyi Wang. Query complexity of generalized Simon’s problem. *arXiv preprint*, 2019. [arXiv:1907.07367]