

Quantum Worst-Case to Average-Case Reductions for All Linear Problems

Vahid R. Asadi* Alexander Golovnev† Tom Gur‡ Igor Shinkar§
Sathyawageeswar Subramanian¶

Abstract

We study the problem of designing worst-case to average-case reductions for quantum algorithms. For all linear problems, we provide an explicit and efficient transformation of quantum algorithms that are only correct on a small (even sub-constant) fraction of their inputs into ones that are correct on *all* inputs. This stands in contrast to the classical setting, where such results are only known for a small number of specific problems or restricted computational models. En route, we obtain a tight $\Omega(n^2)$ lower bound on the average-case quantum query complexity of the Matrix-Vector Multiplication problem.

Our techniques strengthen and generalise the recently introduced *additive combinatorics* framework for classical worst-case to average-case reductions (STOC 2022) to the quantum setting. We rely on quantum singular value transformations to construct quantum algorithms for linear verification in superposition and learning Bogolyubov subspaces from noisy quantum oracles. We use these tools to prove a quantum local correction lemma, which lies at the heart of our reductions, based on a noise-robust probabilistic generalisation of Bogolyubov's lemma from additive combinatorics.

*University of Waterloo. Email: vasadi@uwaterloo.ca.

†Georgetown University. Email: alexgolovnev@gmail.com.

‡University of Warwick. Email: tom.gur@warwick.ac.uk. Tom Gur is supported by the UKRI Future Leaders Fellowship MR/S031545/1 and an EPSRC New Horizons Grant EP/X018180/1.

§Simon Fraser University. Email: ishinkar@sfu.ca.

¶University of Warwick. Email: Sathya.Subramanian@warwick.ac.uk.

Contents

1	Introduction	1
1.1	Our contributions	1
1.2	Related work	3
1.3	Open problems	3
2	Techniques	4
2.1	An additive combinatorics approach	5
2.2	Quantum local correction via a robust Bogolyubov lemma	7
2.3	A toolkit of quantum algorithms for local correction	9
2.4	Quantum worst-case to average-case reductions	12
3	Preliminaries	14
3.1	Quantum unitary oracles and Fourier transforms	14
3.2	Average-case quantum algorithms	15
4	Quantum worst-case to average-case reductions	16
4.1	Quantum algorithms for all linear problems	17
4.2	Matrix-vector multiplication in the query model	17
5	Quantum toolkit for local correction	18
5.1	Flagging correct matrix-vector products in superposition	19
5.2	Noisy quantum oracles approximating the indicator function 1_X	25
5.3	Quantum sampling from the set of good inputs	26
5.4	Learning Bogolyubov subspaces from noisy quantum oracles	27
6	Robust quantum local correction via additive combinatorics	30
6.1	Robust probabilistic Bogolyubov lemma	30
6.2	Quantum local correction lemma	32
7	Reductions for linear problems	33
7.1	Properties of threshold sets	34
7.2	Proof of Lemma 4.1	34
A	Quantum singular value transformation techniques	42
A.1	Fixed-point amplitude amplification	42
A.2	Singular value threshold projections	42

1 Introduction

Average-case complexity is a central area of research in the theory of computing, which studies algorithms that solve problems on average inputs. This notion provides a paradigm for designing efficient algorithms that work on many relevant inputs, even if the worst-case complexity of the problem is high (cf., the standard textbooks [Gol08; AB09]).

Worst-case to average-case reductions are transformations of algorithms that are correct on a small fraction of their inputs into algorithms that are correct *on all* inputs. That is, given an algorithm ALG for computing a function f that satisfies $\Pr_x[\text{ALG}(x) = f(x)] \geq \alpha$, the goal is to boost the *success rate* α (i.e., the fraction of inputs upon which the algorithm is correct) to 1 without significantly increasing the algorithm’s complexity. We stress that for general problems even if there is an efficient way to verify the output of ALG (e.g., if ALG outputs a flag indicating that it has succeeded), it is still unclear if such a goal is achievable at all.

We can view worst-case to average-case reductions both as a means for deriving average-case hardness results from worst-case lower bounds, and also as a paradigm for designing worst-case algorithms by first constructing algorithms that are only required to succeed on a small fraction of their inputs and then using the reduction to obtain algorithms that are correct on all inputs.

In this work, we study *efficient* worst-case to average-case reductions where the success rate is low (e.g., where the average-case algorithm is only correct on 1% of the inputs¹) in the setting of *quantum computing*. On the one hand, the quantum setting is more complex and poses significant challenges since we need to transform a much larger class of average-case algorithms. On the other hand, the quantum setting also allows us to use powerful quantum procedures in the design of the worst-case algorithms to avoid classical bottlenecks.

This paper deals with the following fundamental question regarding quantum average-case complexity:

Is it possible to transform efficient quantum algorithms that are only correct on 1% of their inputs into efficient quantum algorithms that are correct *on all* inputs?

1.1 Our contributions

We provide a strong, positive answer to the question above. In fact, we show that not only are such transformations possible, but that we can construct explicit and efficient quantum worst-case to average-case reductions *for all linear problems* with only constant blowup in the complexity.

This stands in stark contrast to the case of classical algorithms, where such reductions are only known for a small number of specific problems or restricted models. Furthermore, our reduction not only supports average-case algorithms in the 1% regime but also algorithms where the success rate α tends to zero; that is, algorithms that are only correct on a vanishing fraction of their inputs.

In the following, generalising the definition of **BQP** algorithms to fine-grained search problems in the standard way, we define an average-case quantum algorithm as a uniformly generated set of quantum circuits $\{C_n\}_{n \in \mathbb{N}}$ which, upon measurement, output correctly with probability α , where the probability is taken over *both* the random input and the measurements. (See formal definitions in Section 4.)

¹We stress that the 1% regime is far more challenging. Indeed, in the 99% regime, simple self-correction can be used to obtain fine-grained worst-case to average-case reductions for a number of problems [BLR90].

A linear problem is characterised by a family of matrices $\mathcal{M} := \{M_n \in \mathbb{F}^{n \times n}\}_{n \in \mathbb{N}}$, where on input $v \in \mathbb{F}^n$ the solution to the problem is the vector Mv , omitting the subscript on M for readability. Linear problems constitute one of the most fundamental classes of problems, generalising many important computational tasks such as polynomial evaluation, computing discrete Fourier transformations, homology, and various computational tasks for error correcting codes.

Our main result is a worst-case to average-case reduction which shows that for all linear problems, an efficient quantum algorithm that is only successful on a small fraction of its inputs can be explicitly transformed into a similarly efficient quantum algorithm that is correct with high probability *on all inputs*. Π_x below denotes an orthogonal projection on the output register of ALG that represents measuring the outcome $x \in \mathbb{F}^n$ in the standard basis.

Theorem 1 (Informally stated; see Theorem 4.2). *Let \mathbb{F} be a finite field, $M := \{M_n \in \mathbb{F}^{n \times n}\}_{n \in \mathbb{N}}$ be any linear problem, and ALG be an average-case quantum algorithm of (gate) complexity T for M satisfying*

$$\Pr_{v, \text{ALG}}[\text{ALG}(v) = Mv] = \mathbb{E}_{v \in \mathbb{F}^n} \left[\|\Pi_{Mv} \text{ALG} |v\rangle |0\rangle\|^2 \right] \geq \alpha .$$

Then, for every constant $\delta > 0$, there exists a worst-case quantum algorithm ALG' of (gate) complexity $(T + n^{3/2}) \cdot \text{poly}(1/\alpha)$ that succeeds over all inputs with high probability, i.e.,

$$\forall v \in \mathbb{F}^n \quad \Pr_{\text{ALG}'}[\text{ALG}'(v) = Mv] = \|\Pi_{Mv} \text{ALG}' |v\rangle |0\rangle\|^2 \geq 1 - \delta .$$

Note that every linear problem can be trivially solved in time $O(n^2)$. Our result shows that any non-trivial (subquadratic) average-case quantum algorithm can be transformed into a non-trivial (sub-quadratic) quantum algorithm that works for *all* inputs. We also remark that constructing worst-case to average-case reductions for linear problems becomes significantly harder as we consider smaller fields, as we discuss in the technical overview (Section 2). We stress that our reductions also hold for small fields, including \mathbb{F}_2 .

Our proof of Theorem 1 builds upon a machinery that we develop in the quantum query model. This allows us to obtain the following worst-case to average-case reduction for the fundamental and well-studied problem of Matrix-Vector Multiplication in the quantum query model [BŠ06; Kot14].

Theorem 2 (Informally stated; see Theorem 4.3). *Let ALG be an average-case quantum query algorithm with oracle access to a matrix M and a vector v over a finite field \mathbb{F} . Suppose that ALG makes q queries and satisfies*

$$\Pr_{\substack{M, v, \\ \text{ALG}}}[\text{ALG}^{M, v} = Mv] = \mathbb{E}_{\substack{M \in \mathbb{F}^{n \times n} \\ v \in \mathbb{F}^n}} \left[\|\Pi_{Mv} \text{ALG}^{M, v} |0\rangle\|^2 \right] \geq \alpha .$$

Then, for every constant $\delta > 0$, there exists a worst-case quantum query algorithm ALG' with query complexity $(q + n^{3/2}) \cdot \text{poly}(1/\alpha)$ that succeeds on all inputs with high probability, i.e.,

$$\forall M \in \mathbb{F}^{n \times n}, v \in \mathbb{F}^n \quad \Pr_{\text{ALG}'}[(\text{ALG}')^{M, v} = Mv] = \|\Pi_{Mv} (\text{ALG}')^{M, v} |0\rangle\|^2 \geq 1 - \delta .$$

In the query model, it is known that the worst-case quantum query complexity of Matrix-Vector Multiplication has a tight lower bound of $\Theta(n^2)$ (see, e.g., [Kot14]). Hence as an immediate corollary of Theorem 2, we obtain a tight unconditional average-case lower bound for Matrix-Vector multiplication, showing that the problem remains hard even if the quantum algorithm is only required to succeed on a small fraction of the inputs.

Corollary 3. *For every constant $\alpha > 0$, every average-case quantum query algorithm for Matrix-Vector Multiplication with success rate α must make $\Omega(n^2)$ queries.*

1.2 Related work

The study of the average-case complexity originates in the work of Levin [Lev86], and follow-up works such as [BCG⁺92]. A long line of works established various barriers to designing worst-case to average-case reductions for **NP**-complete problems (see, e.g., [IL90; Imp11] and references therein). We refer the reader to the classical surveys by Impagliazzo [Imp95], Bogdanov and Trevisan [BT06], and Goldreich [Gol11] on this topic.

On the other hand, there are known worst-case to average-case reductions for certain problems [Lip91; FF93; BFN⁺93; Ajt96; STV01]. For example, the problems underlying the classical number-theoretic cryptography (such as the RSA, discrete logarithm, and quadratic residuosity problems) are random self-reducible, and, therefore, admit efficient worst-case to average-case reductions (for fixed parameters). The celebrated work of Shor [Sho94] gave a polynomial time quantum algorithm breaking the number-theoretic cryptosystems, which sparked interest in post-quantum cryptography, i.e., cryptography secure even against (polynomial time) quantum adversaries. A number of quantum and classical worst-case to average-case reductions [Ajt96; MR04; Reg09; LPR13; LS15; Gen10] allowed us to base the security of (post-quantum) lattice-based cryptography on the *worst-case* quantum hardness assumptions for certain computational problems. Another interesting example of a (quantum) worst-case to average-case reduction was recently given in [LdW21] for problems related to phase estimation.

Recently, the study of fine-grained complexity [Vas18] of algorithmic problems sparked interest in designing *efficient* worst-case to average-case reductions, i.e., reductions that do not suffer a polynomial overhead in the running time. Such reductions are often motivated by fine-grained cryptographic applications [BRS⁺17; BRS⁺18; GR18; LLV19; BBB19; DLV20].

In a recent work [AGG⁺22], a new framework for showing efficient worst-case to average-case reductions was introduced. This framework uses the quasi-polynomial Bogolyubov-Ruzsa lemma to show reductions in the classical setting that support the low-agreement regime, where the average-case algorithm is only guaranteed to succeed on 1% of the inputs. This framework was used to obtain reductions for a few specific problems or restricted models of classical computation.

1.3 Open problems

Our work opens up several new directions of investigation. Below, we highlight three open problems of particular interest. Recall that in Theorem 1, we have shown fine-grained *quantum* worst-case to average-case reductions, where the success rate is arbitrarily small, *for all linear problems*. These reductions make crucial use of quantum procedures that speed up classical computational tasks such as linear verification and learning of Bogolyubov subspaces from approximate indicators encoded in noisy quantum oracles.

Interestingly, unlike the general result above for quantum algorithms, in the classical setting, the aforementioned computational tasks constitute a complexity bottleneck, and in turn such worst-case to average-case reductions are only known for a small number of specific problems or restricted computational models [AGG⁺22]. It remains open whether such general results for all linear problem can also be obtained in the classical setting.

Open Problem 1. *Are there efficient transformations of classical algorithms (or circuits) for general linear problems, which are only correct on 1% of their inputs, into similarly efficient worst-case classical algorithms (or circuits)?*

Returning to the quantum setting, it is natural to ask whether the framework we constructed can be extended beyond the class of linear problems.

Open Problem 2. *Can efficient average-case quantum algorithms with success rate 1% for large classes of non-linear problems be transformed into efficient worst-case quantum algorithms?*

The last open problem we would like to raise refers to quantum algorithms that act on data of exponential size, encoded in the amplitudes of a quantum state, as in the celebrated HHL algorithm [HHL09]. It would be highly appealing to extend the framework presented in this paper to this setting, where quantum algorithms are extremely powerful.

Open Problem 3. *Can we obtain fine-grained worst-case to average-case reductions for quantum algorithms such as the HHL algorithm, in which the data (and the output) are encoded in the amplitudes of a quantum state?*

Acknowledgements

We thank Richard Cleve for discussions that inspired this work. We also thank the anonymous referees for their very helpful suggestions.

Organisation

The rest of the paper is organised as follows. In Section 2, we provide an overview of our techniques and a high-level overview of the proof of Theorem 2. In Section 3, we give the necessary background material. In Section 4, we state a general technical lemma (Lemma 4.1) and show how to use it to derive our worst-case to average-case reductions (Theorem 1 and Theorem 2). The subsequent sections construct the components that are necessary to prove the aforementioned main technical lemma, as follows.

In Section 5, we provide a toolkit of quantum algorithms for local correction; our techniques employ the quantum singular value transformation machinery, which we discuss in Appendix A. Then, in Section 6, we prove a robust and probabilistic version of Bogolyubov’s lemma from additive combinatorics, and we use it together with our toolkit of quantum algorithms to obtain a quantum local correction lemma, which will play a key role in our worst-case to average-case reductions. Finally, in Section 7 we use all the tools above to prove the main technical lemma (Lemma 4.1).

2 Techniques

We provide a technical overview, highlighting the main conceptual and technical ideas that we use. Our approach builds on the *additive combinatorics* framework, recently introduced in [AGG⁺22]; however, our setting, which captures all linear problems, is significantly more involved and presents non-trivial challenges that require new additive combinatorics techniques, complexity theoretic ideas, and quantum algorithms that are tailored to this setting.

We start in Section 2.1, where we present our high-level approach for transforming average-case quantum algorithms for linear problems into worst-case quantum algorithms. In Section 2.2, we

discuss the key technical component underlying our reductions, which is a quantum local correction lemma based on a robust, probabilistic version of Bogolyubov’s lemma. Our quantum local correction lemma crucially relies on four quantum algorithms, which we outline in Section 2.3, that allow us to efficiently flag correct solutions in superposition, construct noisy oracles for approximate indicator functions, efficiently sample good inputs, and learn Bogolyubov subspaces from noisy set-indicating quantum oracles. Finally, in Section 2.4, we discuss how to apply the quantum local correction lemma to obtain the desired worst-case to average-case reductions.

2.1 An additive combinatorics approach

In this overview, for simplicity of exposition, we will focus on the proof of Theorem 2 in the quantum query model; we in fact optimise both the query and gate complexities, so that we can then show how to extend the result to (uniform) quantum circuits for all linear problems.

Recall that the Matrix-Vector Multiplication problem (MvM) is defined as follows.

MATRIX-VECTOR MULTIPLICATION (MvM)

Input: Oracle access to a matrix $M \in \mathbb{F}^{n \times n}$ and a vector $v \in \mathbb{F}^n$.

Output: The matrix-vector product Mv .

Let ALG be an *average-case* quantum algorithm that is given oracle access to M and v , defined by

$$U_M |j, k, z\rangle = |j, k, z \oplus M_{jk}\rangle \quad \text{and} \quad U_v |j, z\rangle = |j, z \oplus v_j\rangle,$$

for all indices $j, k \in [n]$ and $z \in \mathbb{F}$. Suppose that using q queries to M and v , the quantum algorithm ALG satisfies

$$\Pr_{\substack{M, v, \\ \text{ALG}}} [\text{ALG}^{M, v} = Mv] = \mathbb{E}_{\substack{M \in \mathbb{F}^{n \times n} \\ v \in \mathbb{F}^n}} \left[\left\| \Pi_{Mv} \text{ALG}^{M, v} |0\rangle \right\|^2 \right] \geq \alpha,$$

where Π_{Mv} is an orthogonal projection on the output register of ALG that indicates whether the algorithm outputs the correct answer.

We would like to explicitly transform ALG into a *worst-case* quantum algorithm ALG' that computes Mv with high probability for *every* matrix M and vector $v \in \mathbb{F}^n$; that is, show that for every constant $\delta > 0$, there exists a worst-case quantum algorithm ALG' satisfying

$$\forall M \in \mathbb{F}^{n \times n}, v \in \mathbb{F}^n \quad \Pr_{\text{ALG}'} [(\text{ALG}')^{M, v} = Mv] = \left\| \Pi_{Mv} (\text{ALG}')^{M, v} |0\rangle \right\|^2 \geq 1 - \delta.$$

To simplify the discussion, unless specified otherwise, in this overview we restrict our attention to the field \mathbb{F}_2 , and to arbitrarily small constant values of the success rate parameter $\alpha > 0$ of average-case algorithms (say $\alpha = 0.01$).

Worst-case to average-case reductions via additive combinatorics. Our starting point is the additive combinatorics framework that was recently introduced in [AGG⁺22]. Namely, we’d like to decompose each input vector v into a sum of vectors on each of which the average-case algorithm ALG is correct. However, since a simple linear decomposition of each $v \in \mathbb{F}^n$ into correctly-computed inputs does not exist,² we shall need more involved machinery from the field of additive combinatorics.

²Indeed, as in the classical setting, consider the simple counterexample where the average-case algorithm $\text{ALG}^{M, v}$ outputs $M \cdot v$ in case that the first coordinate of v is 1 and otherwise outputs 0. Note that in this case the success

To this end, we will start with Bogolyubov’s lemma, a fundamental result in additive combinatorics which shows that the 4-ary sumset of any dense set in \mathbb{F}_2^n contains a large linear subspace. More accurately, recall that the sumset of a set A is defined as $A + A = \{a_1 + a_2 : a_1, a_2 \in A\}$, and similarly $4A = \{a_1 + a_2 + a_3 + a_4 : a_1, a_2, a_3, a_4 \in A\}$. These objects can be thought of as a combinatorial analogue of an approximate subgroup. Considering these sumsets allows us to extract subspace structure out of an unstructured set, as encapsulated in the following lemma.

Lemma 2.1 (Bogolyubov’s lemma). *For any subset $A \subseteq \mathbb{F}_2^n$ of density $|A|/2^n \geq \alpha$, there exists a subspace $V \subseteq 4A$ of dimension at least $n - \alpha^{-2}$.*

To see the initial intuition for the additive combinatorics approach to designing worst-case to average-case reductions, we first make the simplifying assumption that the average-case algorithm ALG receives a “good” matrix $M \in \mathbb{F}^{n \times n}$ for which it is successful with probability α , taken over the measurement and the random input vector $v \in \mathbb{F}^n$; that is,

$$\Pr_v[\text{ALG}^{M,v} = Mv] = \mathbb{E}_{v \in \mathbb{F}^n} \left[\|\Pi_{Mv} \text{ALG}^{M,v} |0\rangle\|^2 \right] \geq \alpha .$$

In Section 2.4, we will show how to extend our approach to work on average-case matrices as well.

First note that by an averaging argument, there exists a set of size $(\alpha \cdot 2^n)/2$ of input vectors $v \in \mathbb{F}^n$ on which ALG correctly computes the output with probability at least $\alpha/2$; denote this set by X . Next, observe that Bogolyubov’s lemma shows that there exists a large subspace V such that every $v \in V$ can be decomposed as

$$v = x_1 + x_2 + x_3 + x_4, \text{ for } x_1, x_2, x_3, x_4 \in X . \tag{1}$$

Recall that each $x_i \in X$ can be computed correctly by the average-case algorithm with probability at least $\alpha/2$. This suggests the natural approach of locally correcting each $v \in V$ using four inputs upon which the average-case algorithm has a non-negligible success probability.

The challenges. While the discussion above outlines a promising approach, already at this point there are substantial difficulties that arise when trying to pursue it. For starters, even on good inputs in X the average-case algorithm only computes correctly with probability $\alpha/2$ (which could tend to zero!), and so it is unclear how to amplify the success probability of the algorithm such that it computes correctly on all four elements of X in the decomposition in Eq. (1) with high probability, which is necessary for the self-correction of inputs in the Bogolyubov subspace V .

We stress that *this is not the case* with the problems that were considered in [AGG⁺22] (where verification was done using Freivald’s algorithm for matrix-matrix multiplication, or using the unbounded preprocessing power in the data structure model). In contrast, in the setting of general linear problems, naive verification of a matrix-vector product costs $O(n^2)$, which completely trivialises the problem; indeed, in the classical setting, $o(n^2)$ -query verification of matrix-vector products is impossible. Further issues include the fact that Bogolyubov’s lemma only guarantees the existence of a decomposition into elements of X , whereas we need to explicitly obtain such a decomposition, as well as the fact that the argument above only holds for correcting inputs that are *inside the Bogolyubov subspace V* , whereas we need to locally correct *all inputs*.

rate is $\alpha \geq 1/2$, yet no linear decomposition could self-correct matrix-vector multiplication where the first coordinate of v is 1. Indeed, any such decomposition $v = \sum_i v_i$ would have a v_i with the first element 1, where $\text{ALG}^{M,v}$ fails.

Unfortunately, the classical framework that was shown in [AGG⁺22] fails to address the aforementioned problems in the setting of general linear problems. Indeed, it is not clear whether the ambitious task of constructing worst-case to average-case reductions for all linear problems is at all possible for classical algorithms.

Instead, our approach for overcoming these challenges is *inherently quantum*, and it would require more involved ideas, both technically and conceptually. We will show a new, noise-robust version of Bogolyubov’s lemma (see Section 2.2), and together with a toolkit of quantum algorithms (building on quantum singular value transformations) that we develop (see Section 2.3), we will prove a new quantum local correction lemma that will play a key role in our reductions, which are tailored to the strengths and limitations of the quantum setting.

2.2 Quantum local correction via a robust Bogolyubov lemma

Our starting point for addressing the difficulties outlined in Section 2.1 is a new *robust* and *probabilistic* analogue of Bogolyubov’s lemma. These additional structural properties will in turn enable us to deal with inputs outside of the Bogolyubov subspace and efficiently obtain explicit decompositions of inputs inside the Bogolyubov subspace into inputs upon which the average-case algorithm is correct.

Then, using the robust probabilistic Bogolyubov lemma, we will show a local correction lemma for quantum algorithms for linear problems. Our quantum local correction lemma will crucially make use of the robustness and probabilistic properties, as well as the four quantum procedures that we present in Section 2.3.

A robust probabilistic Bogolyubov lemma. Recall that our high-level idea for locally correcting faulty inputs in the subspace V guaranteed by Bogolyubov’s lemma proceeds by decomposing vectors $v \in V$ into a linear combination of *good inputs* from the set X . However, to prove our quantum local correction lemma, we need to be able to: (1) deal with vectors *outside of the Bogolyubov subspace* V , and (2) efficiently obtain an explicit decomposition of vectors inside V into good inputs in X . To this end, we shall first need to strengthen Bogolyubov’s lemma to obtain the following structural properties.

- *Robustness.* Our local correction lemma relies on learning the heavy part of the Fourier spectrum of a noisy representation of an approximate indicator of the set X in superposition, using a quantum procedure that we discuss in Section 2.3. In this setting, we can only learn the heavy Fourier coefficients of a function g satisfying

$$\forall S \subset [n] \quad \left| |\widehat{1_X}(S)|^2 - |\widehat{g}(S)|^2 \right| \leq \varepsilon ,$$

for a sufficiently small ε . In fact, each Fourier coefficient we obtain may come from a *different function* g . To deal with this problem, which is *unique to the quantum setting*, we need a version of Bogolyubov’s lemma where the subspace is defined via a robust set of linear constraints that can accommodate for the noisy Fourier spectrum.

- *Probabilistic decomposition.* Bogolyubov’s lemma shows the *existence* of a large subspace $V \subset 4X$, admitting a linear decomposition of each $v \in V$ into four vectors from the set of good inputs X upon which the average-case algorithm is correct. However, we need to efficiently obtain an

explicit decomposition $v = x_1 + x_2 + x_3 + x_4$, where each $x_i \in X$, for each vector $v \in V$. Hence, we show that each vector admits sufficiently many such decompositions such that one can be efficiently sampled using a quantum sampling procedure that we show in Section 2.3.

We thus prove the following *robust probabilistic Bogolyubov lemma*, which will allow us to deal with the challenges above. Given an unstructured set $X \subseteq \mathbb{F}^n$, denote the heavy part, exceeding a threshold γ , of the Fourier spectrum of X by $\text{Spec}_X(\gamma) = \{r \in \mathbb{F}^n \setminus \{0\} : |\widehat{1}_X(r)| \geq \gamma\}$.

Lemma 2.2 (Informally stated, see Lemma 6.1). *For every $X \subseteq \mathbb{F}^n$ of density at least α , let $R \subseteq \mathbb{F}^n$ be a set such that $\text{Spec}_X(\alpha^{3/2}) \subseteq R \subseteq \text{Spec}_X(\frac{\alpha^{3/2}}{2})$. Let $V = \{v \in \mathbb{F}^n : \langle v, r \rangle = 0 \ \forall r \in R\}$. Then $\dim(V) \geq n - O(\alpha^{-2})$, and for all $v \in V$ it holds that*

$$\Pr_{x_1, x_2, x_3 \in X} [v - x_1 - x_2 - x_3 \in X] \geq \alpha^2 .$$

We stress that the probability in Lemma 2.2 is taken over x_1, x_2, x_3 that are uniformly sampled from X (rather than \mathbb{F}^n). To efficiently obtain such a decomposition, we can sample x_1, x_2, x_3 using the quantum sampling procedure shown in Section 2.3.

Locally correcting outside of the Bogolyubov subspace. While the robust probabilistic Bogolyubov lemma allows us to locally correct inputs inside the subspace $V \subseteq 4X$, our goal is to obtain a worst-case quantum algorithm, hence we need to be able to handle any vector $v \in \mathbb{F}^n$, and not just those in V .

Towards this end, we would like to shift each vector $v \in \mathbb{F}^n \setminus V$ into the Bogolyubov subspace. Using the robustness property of Lemma 2.2 and the quantum algorithm for learning Bogolyubov subspaces from a noisy representation of indicator functions in superposition (see Section 2.3), we further decompose each vector in \mathbb{F}^n into a sum of elements in the subspace V and a *sparse* shift-vector s , which can then be corrected efficiently due to its sparsity.

In more detail, let $R \subseteq \mathbb{F}^n \setminus \{0\}$ and $V = \{v \in \mathbb{F}^n : \langle v, r \rangle = 0 \ \forall r \in R\}$. We observe that there exists a collection of $t \leq |R|$ vectors $B = \{b_1, \dots, b_t\}$, $b_i \in \mathbb{F}^n$ and indices $k_1, \dots, k_t \in [n]$ such that $\text{span}(B) = \text{span}(R)$ and every vector $y \in \mathbb{F}^n$ can be written as $y = v + s$, where $v \in V$ and $s = \sum_{j=1}^t c_j \cdot \vec{e}_{k_j}$ for $c_j = \langle y, b_j \rangle$ and \vec{e}_{k_j} is a unit vector. We emphasize that the sparsity of the decomposition is critical as it allows us to shift arbitrary vectors into the Bogolyubov subspace V without unfavourably blowing up the complexity.

To efficiently obtain the basis $b_1, \dots, b_t \in \mathbb{F}^n$ and indices $k_1, \dots, k_t \in [n]$, we rely on the quantum algorithm for learning Bogolyubov subspaces described in Section 2.3, which allows us to obtain the approximate high Fourier coefficients of an approximation of 1_X from a noisy representation in superposition, within the desired complexity. This, in turn, allows us to compute the required basis and set of indices.

The quantum local correction lemma. Putting together all of the components above, we can now state our quantum local correction lemma, which builds upon the robust probabilistic version of Bogolyubov's lemma and the quantum procedures that we will present in Section 2.3.

Loosely speaking, our local correction lemma allows us to efficiently obtain an explicit decomposition of any vector $v \in \mathbb{F}^n$ into a linear combination of the form

$$v = x_1 + x_2 + x_3 + x_4 + s ,$$

where $x_1, x_2, x_3, x_4 \in X$ and $s \in \mathbb{F}^n$ is a *sparse* vector.

Specifically, in Section 2.3, we will construct a noisy quantum oracle for approximating the indicator $1_X(v)$, quantum algorithm for uniformly sampling from X , a quantum verification procedure \tilde{O}_X that for each $v \in \mathbb{F}^n$ computes the indicator $1_X(v)$ correctly with high probability, and a quantum algorithm that learns the Bogolyubov subspace from the noisy oracle that approximates 1_X . By combining these four ingredients we prove the following result.

Lemma 2.3 (informally stated, see Lemma 6.2). *For a field $\mathbb{F} = \mathbb{F}_p$ and α -dense set $X \subseteq \mathbb{F}^n$, there exists $t \leq 4/\alpha^2$ vectors $b_1, \dots, b_t \in \mathbb{F}_2^n$ and indices $k_1, \dots, k_t \in [n]$ satisfying the following. Given a vector $y \in \mathbb{F}^n$, define $s = \sum_{j=1}^t \langle y, b_j \rangle \cdot \vec{e}_{k_j}$, where $(\vec{e}_i)_{i \in [n]}$ is the standard basis. Then,*

$$\Pr_{x_1, x_2, x_3 \in X} [x_4 \in X] \geq \alpha^2 \quad ,$$

where x_4 is the vector such that $y = s + x_1 + x_2 + x_3 + x_4$.

Furthermore, there exists a quantum algorithm that calls the quantum verification procedure \tilde{O}_X for $O(\log(1/\delta) \cdot (1/\alpha^5))$ times, uses $n \cdot \text{poly}(1/\alpha) \cdot \text{poly} \log |\mathbb{F}|$ additional elementary gates, and with probability at least $1 - \delta$ returns the vectors $b_1, \dots, b_t \in \mathbb{F}^n$ and indices $k_1, \dots, k_t \in [n]$.

This quantum local correction lemma forms the cornerstone of our quantum average-case to worst-case reductions, as we explain in Section 2.4. However, before we can turn to the details of the reduction, we need to address three gaps that still remain: How can we efficiently verify Matrix-Vector products, so that we can amplify the success probability of quantum algorithms that are only correct with probability α (which could tend to zero)? Can we efficiently sample from the set of good inputs X ? How can we learn the Bogolyubov subspace from a noisy quantum oracle that encodes an approximation of the indicator 1_X ?

2.3 A toolkit of quantum algorithms for local correction

Next, we address the foregoing questions by presenting the key quantum procedures that are required for our quantum local correction lemma. We stress that these solutions employ the power of quantum algorithms, and indeed, the lack of such procedures in the classical setting is a bottleneck that blocks the path to general results such as Theorem 1 and Theorem 2 for *classical* algorithms.

We achieve quantum speedups for the following four tasks, which are required for our quantum local correction lemma:

1. Flagging correct matrix-vector products in superposition.
2. Constructing noisy quantum oracles approximating the indicator function 1_X .
3. Sampling of vectors from the set X of good vectors.
4. Learning Bogolyubov subspaces from noisy quantum oracles.

Recall that it is essential to perform the above tasks in complexity $o(n^2)$; indeed our quantum algorithms make at most $O(n^{3/2})$ queries and use at most $\tilde{O}(n^{3/2})$ additional one-qubit and two-qubit gates. We now proceed to briefly describe the above quantum procedures, and offer a quick glimpse at our techniques for obtaining them. See Section 5 for details.

Flagging correct matrix-vector products in superposition. The first tool we shall need is a quantum subroutine for verifying whether a vector $b \in \mathbb{F}^n$ output by a quantum algorithm **ALG** is in fact the correct matrix-vector product Mv , a task that classically requires $\Omega(n^2)$ queries. This will play an important role in both amplifying the success probability of our local correction lemma (see Section 2.2) and in the other quantum algorithms in the toolkit we develop.

Since it is crucial for our reduction to use such a verification procedure as a unitary subroutine in other quantum procedures, we unfortunately cannot apply existing verification algorithms from the literature. Furthermore, since the algorithm **ALG** only succeeds *on average* with low probability, different input vectors have significantly different success probabilities. To address this, we give a procedure that flags all of the computational basis states in a superposition as either right or wrong in a way that respects the success distribution of **ALG**, and then boost the amplitude of the solutions.

However, since the success probabilities of different inputs have high variance, choosing a fixed number of iterations of amplitude amplification causes problems of over-shooting and under-shooting. Instead, we apply a delicate argument involving fixed-point amplitude amplification, which converges monotonically towards the flagged state. Towards this end, we identify that the flagging operation above has the form of a *block encoding*.³ Having such an object enables us to use powerful techniques from the repertoire of quantum singular value transformations.

Implementing the strategy above, we construct the following quantum procedure for verifying matrix-vector products, which works *in superposition* and marks an ancillary qubit attached to the output state of **ALG** whenever the state has non-zero overlap with the correct matrix-vector product $|Mv\rangle$. See Section 5.1 for details.

Lemma 2.4 (Informally stated; see Lemma 5.2). *Given access to a unitary oracle for a matrix $M \in \mathbb{F}^{n \times n}$, and a quantum algorithm **ALG** that takes as input a vector $v \in \mathbb{F}^n$ and produces as output a state that consists of a superposition over vectors in \mathbb{F}^n ,*

$$\text{ALG} |v, 0\rangle = \sum_{z \in \mathbb{F}^n} \gamma_z^v |v\rangle |z\rangle |w(z, v)\rangle ,$$

*there is a quantum algorithm $\text{ALG}_{\text{verified}}$ that annotates the output state of **ALG** with a flag marking the vector Mv as correct, and marking all other vectors $z \neq Mv$ as incorrect with high probability, i.e.*

$$\text{ALG}_{\text{verified}} |v, 0\rangle \approx \gamma_*^v |v\rangle |Mv\rangle |w(Mv, v)\rangle |1\rangle^{\text{flag}} + \left(\sum_{z \neq Mv} \gamma_z^v |v\rangle |z\rangle |w(z, v)\rangle \right) |0\rangle^{\text{flag}} .$$

$\text{ALG}_{\text{verified}}$ makes $O(1)$ uses of **ALG**, $O(n^{3/2})$ queries to U_M and U_M^\dagger , $\tilde{O}(n)$ ancillary qubits, and $\tilde{O}(n^{3/2})$ additional one-qubit and two-qubit gates.

Noisy quantum oracles approximating the indicator function 1_X . A key ingredient in our reductions is the indicator function or membership oracle 1_X for the subset of good inputs exceeding probability threshold τ , given by

$$X_\tau := \{v \in \mathbb{F}^n : \Pr[\text{ALG}^M(v) = Mv] \geq \tau\} .$$

³A unitary U with the form $\begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix}$, which encodes another (subnormalised) matrix H in its upper left block.

Here the notation ALG^M emphasises the fact that ALG has query access to the matrix M . In constructing this indicator function we are faced with yet another challenge: while fixed-point amplitude amplification is able to boost the probability of outputting the correct success/failure flag, it is insufficient to perform the type of thresholding operation that is required to implement the indicator function. The key difficulty is the fact that we need to *simultaneously* mark inputs inside set X_τ with the flag one, and those outside X_τ with the flag zero (with high probability).

To overcome this issue, we note that $\text{ALG}_{\text{verified}}$ is also a block encoding (of another matrix), and by combining it with the heavier machinery of *quantum singular value threshold projection*, we are able to obtain a noisy quantum oracle for a polynomial approximation of the indicator function on the set X_τ of good inputs to ALG . In Section 5.2, we show how the technique of quantum singular value transformations can be used to construct a singular value threshold projection built on $\text{ALG}_{\text{verified}}$, which acts as a noisy indicator function on the set of inputs where ALG succeeds with high probability.

Sampling vectors from the set of good inputs. Using the ability to approximately flag correct and incorrect answers in superposition, we can synthesize a uniform superposition over the set X_τ that consists of all inputs on which the average-case algorithm computes correctly with probability at least τ . Hence by preparing this state and measuring it, we obtain the following quantum speedup for sampling from X_τ . This will allow us to boost the performance of our quantum local correction lemma (see Section 2.2), that needs to sample elements from X_τ for a value of $\tau = \Theta(\alpha)$ in order to correct vectors in the Bogolyubov subspace.

Lemma 2.5 (Informally stated; see Section 5.3). *Given an average-case quantum algorithm ALG with success rate α , there is a procedure Q_{samp} which with high probability produces a vector v on which ALG succeeds with probability at least $\alpha/2$. Q_{samp} uses ALG $O(\frac{1}{\alpha})$ times, in addition to $\tilde{O}(n^{3/2})$ additional one-qubit and two-qubit gates.*

Learning Bogolyubov subspaces from noisy quantum oracles. In order to locally correct vectors that are outside of the Bogolyubov subspace V , we shall need to learn an approximation of V , so that we can shift arbitrary vectors into V via sparse vectors (see Section 2.2).

Standard quantum procedures for sampling characters according to the probabilities defined by the Fourier spectrum of a function f use a unitary oracle for f in superposition, and are built on the Bernstein-Vazirani algorithm. However, in our setting we need to list-decode the heavy Fourier coefficients of a probabilistic implementation of an indicator function that we obtain from the linear verification procedure in Lemma 2.4. Hence, we face the challenge of learning the heavy Fourier characters from a noisy implementation of the function. To this end, we generalise the techniques used to prove Adcock and Cleve’s quantum Goldreich-Levin lemma [AC02], and obtain the following quantum procedure.

Lemma 2.6 (Informally stated; see Lemma 5.5). *For a function $f : \mathbb{F}^n \rightarrow \mathbb{F}_2$, given a noisy quantum oracle U_f acting on m qubits such that for every input x , measuring the output qubit of U_f produces $f(x)$ with probability at least $1 - \varepsilon$, there is a quantum procedure C_{GL} which produces outputs $y \in \mathbb{F}^n$ with probability p_y that satisfy $|p_y - |\hat{f}(y)|^2| \leq 4\varepsilon$. C_{GL} uses U_f and U_f^\dagger once, in addition to $O(n \log |\mathbb{F}|)$ additional one-qubit and two-qubit gates.*

We stress that due to the noisy implementation, we do not obtain the heavy Fourier characters of the actual indicator function of the set of good inputs X , but rather an approximation of these

Fourier coefficients. Fortunately, this guarantee suffices for the noise-robust Bogolyubov lemma that we presented in Section 2.2.

In fact, Lemma 2.6 is not the last link in the chain: a further complication arises from the fact that any polynomial approximation of the indicator function $\mathbf{1}_X$ can only work well in a subset of X and a subset of its complement, necessarily oscillating in a “wasteland” region corresponding to vectors on which the success probability of **ALG** lies in some range $(\tau - \delta, \tau + \delta)$.

The polynomial approximation happens at the level of a real-valued function (the success probability of **ALG** on input vectors) and the corresponding wasteland slice is apparently small, namely, an interval of length 2δ . However this hides the fact that the actual number of input vectors falling into this intermediate set can be alarmingly large, with density as high as $\frac{1-\alpha}{1-\alpha/2}$.

We show that in spite of this difficulty, a combination of careful error analysis of our quantum Bogolyubov subspace learning technique and, as we discuss in Section 2.4, a carefully chosen random selection of the threshold τ , allows us to efficiently learn the Bogolyubov subspace.

2.4 Quantum worst-case to average-case reductions

We first present a high-level overview of our reduction for the Matrix-Vector Multiplication problem in a simplified setting, then sketch how to extend the proof to obtain Theorem 1 and Theorem 2.

Recall that we start with an *average-case* quantum algorithm **ALG** that is correct with probability α on a randomly chosen input, i.e.,

$$\Pr_{\substack{M,v \\ \text{ALG}}}[\text{ALG}^{M,v} = Mv] = \mathbb{E}_{\substack{M \in \mathbb{F}^{n \times n} \\ v \in \mathbb{F}^n}} \left[\|\Pi_{Mv} \text{ALG}^{M,v} |0\rangle\|^2 \right] \geq \alpha .$$

Our goal is to boost the success rate of the algorithm such that we obtain a *worst-case* quantum algorithm that succeeds with high probability *on all* inputs.

For the purpose of a clear exposition, we first make the following simplifying assumptions: (1) we work over $\mathbb{F} = \mathbb{F}_2$, (2) we assume $\alpha > 0$ is an absolute constant (and in turn, we will not optimise the dependency on it), (3) we fix the error parameter to an arbitrarily small constant, and (4) we take the average case only on the vectors $v \in \mathbb{F}^n$. In other words, we only consider the case where the given matrix M is a *good* matrix. Indeed, for the case that this does not hold, we will later present matrix self-correction techniques to ensure such a condition is satisfied with high probability.

In order to describe the reduction the following notation will be convenient. For each $v \in \mathbb{F}^n$ let $p_v = \Pr_{\text{ALG}}[\text{ALG}(v) = Mv]$ be the probability that **ALG** correctly computes the output on input v , where the probability is only over the measurement of **ALG**.

First, we define threshold sets as follows.

$$X_\tau = \{v \in \mathbb{F}^n : p_v > \tau\} .$$

Since $\Pr_{v \in \mathbb{F}^n}[\text{ALG}(v) = Mv] \geq \alpha$, it implies that $\mathbb{E}_{v \in \mathbb{F}^n}[p_v] \geq \alpha$, and hence, by Markov’s inequality, for $\tau \leq \alpha/2$ it holds that $\frac{|X_\tau|}{|\mathbb{F}^n|} \geq \alpha/2$. Next, before describing the high-level overview of our proof, we first discuss a simplified warm-up that contains the main idea, while skipping some technical complications that we discuss later.

Warm-up. Ideally, we could construct the worst-case quantum algorithm ALG' as follows. We first learn all of the significant Fourier coefficients of $X_{\alpha/2}$ using the quantum learning procedure in Lemma 2.6. Then, by the quantum local correction lemma (i.e., Lemma 2.3, which in turn, relies on the robust Bogolyubov lemma), these Fourier coefficients can be used to compute a decomposition of any input $v \in \mathbb{F}^n$ as

$$v = s + x_1 + x_2 + x_3 + x_4, \text{ where } x_1, x_2, x_3, x_4 \in X_{\alpha/2},$$

and $s \in \mathbb{F}^n$ is a sparse vector that has only $O(1)$ -non zero entries, which allows us to shift arbitrary inputs into the Bogolyubov subspace.

More specifically, for each input $v \in \mathbb{F}^n$, we can use the Fourier coefficients of $X_{\alpha/2}$ in order to compute the sparse vector s . Then, we sample x_1, x_2, x_3 from $X_{\alpha/2}$, and set $x_4 = v - x_1 - x_2 - x_3 - s$. By the quantum local correction lemma (Lemma 2.3) we have that $\Pr[x_4 \in X_{\alpha/2}] \geq \text{poly}(\alpha)$, and hence set the algorithm ALG' to compute

$$Ms + \text{ALG}^M(x_1) + \text{ALG}^M(x_2) + \text{ALG}^M(x_3) + \text{ALG}^M(x_4).$$

Note that computing Ms can be done in $O(n)$ time, where $O(\cdot)$ hides the sparsity of s , and Mx_i can be computed correctly using ALG for each $i \in \{1, 2, 3, 4\}$ with probability $\text{poly}(\alpha)$. In total, the above procedure outputs the correct answer with probability at least $\text{poly}(\alpha)$. By repeating the procedure $\text{poly}(1/\alpha)$ times and verifying the result at each time, we obtain a reduction that succeeds with high probability.

Random thresholds. The actual reduction is more subtle. The main reason for this is that the lemma we use to learn the Bogolyubov subspace (Lemma 2.6) cannot sample from the Fourier spectrum of the set $X_{\alpha/2}$ exactly, as computing p_v exactly for a particular v requires a large number of samples, and so, we cannot apply Lemma 2.3 directly on the set $X_{\alpha/2}$.

Instead, we choose a parameter $\tau \in [\alpha/4, \alpha/2]$, and consider the sets X_τ and $X_{\tau'}$, where $\tau' = \tau - O(\alpha^{3/2})$. We choose the parameters so that

$$\left| \frac{|X_{\tau'}|}{|\mathbb{F}^n|} - \frac{|X_\tau|}{|\mathbb{F}^n|} \right| = O(\alpha^{3/2}).$$

See Corollary 7.4 for details on how the random threshold τ is chosen. Then, we apply Lemma 2.3 with respect to *some* set X^* such that with high probability $X_\tau \subseteq X^* \subseteq X_{\tau'}$. Note that we do not have any structural guarantee about X^* containing a particular vector in $X_{\tau'} \setminus X_\tau$.

We use Lemma 2.6 in order to obtain all significant Fourier coefficients of X^* . Observing that $\frac{|X^* \Delta X_\tau|}{|\mathbb{F}^n|} = O(\alpha^{3/2})$, it follows that

$$\left| \widehat{X^*}(y) - \widehat{X_\tau}(y) \right| = O(\alpha^{3/2}),$$

for all $y \in \mathbb{F}^n$, and hence we can use the Fourier coefficients of X^* to approximate the Fourier coefficients of X_τ . Here by \widehat{X} we mean the Fourier coefficients of the indicator function of the set X .

By the discussion above, we may assume that we know a good approximation of all significant Fourier coefficients of X_τ . The rest of the reduction follows the same plan as described in the warm-up above.

Extending the average-case over both matrices and vectors. So far, we obtained a worst-case to average-case reduction where the average-case condition only refers to the vectors in the Matrix-Vector Multiplication problem. To extend the average case to both vectors and matrices, we need an additional layer of matrix local correction. To this end, we use the technique of shifting the given matrix M by a random matrix R , which with probability $\Omega(\alpha)$ shifts the input to the set of good matrices, where matrix-vector multiplications are computed correctly for an $\Omega(\alpha)$ -fraction of vectors. See Section 4.2 for details.

Quantum algorithms for linear problems. Throughout the argument that we outlined above, we employed a proof strategy that only invokes gate-efficient quantum algorithms (i.e., with gate complexities that are larger than the query complexity by at most a polylog factor). We do this precisely due to our interest in uniform quantum algorithms for linear problems: by a careful instantiation of families of quantum circuits as quantum query algorithms that match the setting of Theorem 4.3, we obtain in Theorem 4.2 quantum worst-case to average-case reductions for all linear problems. See Section 4.1 for details.

3 Preliminaries

We establish some minimal standard preliminaries and notation regarding quantum algorithms, and refer the reader to standard textbooks such as [NC10] for details.

3.1 Quantum unitary oracles and Fourier transforms

We start by providing basic notation and definitions regarding discrete and quantum Fourier transforms, as well as quantum oracles.

Discrete and quantum Fourier transforms. Let $\mathbb{F} = \mathbb{F}_p$ be the prime field of size p . For a function $f : \mathbb{F}^n \rightarrow \mathbb{R}$, the Fourier coefficient $\hat{f}(y)$ for $y \in \mathbb{F}^n$ representing characters χ_y is given by

$$\hat{f}(y) = \frac{1}{p^n} \sum_{x \in \mathbb{F}^n} \omega^{x \cdot y} f(x) , \quad (2)$$

where $\omega = e^{2\pi i/p}$ is a primitive p^{th} root of unity.

Denote the quantum Fourier transform over \mathbb{F} by QFT_p , having the action

$$\text{QFT}_p |x\rangle = \frac{1}{\sqrt{p}} \sum_{y \in \mathbb{F}_p} \omega^{x \cdot y} |y\rangle . \quad (3)$$

Quantum unitary oracles. We let quantum algorithms access a matrix $M \in \mathbb{F}^{n \times n}$ via a unitary oracle U_M that performs the map

$$U_M |j, k, z\rangle = |j, k, z \oplus M_{jk}\rangle , \quad (4)$$

for all indices $j, k \in [n]$ and $z \in \mathbb{F}$, where \oplus denotes addition over \mathbb{F} . For vectors $v \in \mathbb{F}^n$, oracle access means a unitary U_v that returns components of the vector when queried with an index, analogously to Eq. (4), i.e.,

$$U_v |j, z\rangle = |j, z \oplus v_j\rangle , \quad (5)$$

for all $j \in [n]$ and $z \in \mathbb{F}$. Unless otherwise stated, whenever we assume access to a unitary U as an oracle we also assume access to U^\dagger .

Linear Problems. A linear problem is characterised by a family of matrices $\mathcal{M} := \{M_n \in \mathbb{F}^{n \times n}\}_{n \in \mathbb{N}}$, where on input $v \in \mathbb{F}^n$ the solution to the problem is the vector Mv , omitting the subscript on M for readability.

LINEAR PROBLEM \mathcal{M}

Input: A vector $v \in \mathbb{F}^n$.

Output: The matrix-vector product Mv .

This notion captures a wide variety of problems, including such fundamental ones as computing Discrete Fourier transforms, and polynomial evaluation (Vandermonde matrices). As discussed in the introduction, in this paper we show that for *every* linear problem \mathcal{M} there exists an efficient quantum worst-case to average-case reduction, a result for which no analogue is known in the case of uniform classical algorithms.

Next, we define the central notion that we study in this paper, namely the average-case behaviour of quantum algorithms.

3.2 Average-case quantum algorithms

An average-case quantum algorithm ALG is one that succeeds with probability at least α in expectation over (uniformly) random inputs. That is, if ALG is to compute some function f mapping some known measurable domain V to some co-domain W , then

$$\Pr_{\substack{\text{ALG} \\ v \in V}} [\text{ALG}(v) = f(v)] := \mathbb{E}_{v \in V} [|\Pi_{f(v)} \text{ALG}(v)|^2] \geq \alpha . \quad (6)$$

Here \Pr_{ALG} denotes the probability over the internal (quantum) randomness of the algorithm arising from its unitary nature and final measurements. Note that the probability above is taken over the inputs *as well as* the internal quantum randomness of the algorithm. This is highlighted by the notation we use, which we elaborate more on below.

This notion of average-case quantum algorithms immediately suggests considering the following natural modification of quantum oracles.

Noisy quantum oracles. For a function $f : \mathbb{F}^n \rightarrow \mathbb{F}^m$, we can consider a unitary oracle U_f which on input $x \in \mathbb{F}^n, z \in \mathbb{F}^m$ performs the map

$$U_f |x\rangle |z\rangle = \beta_{\text{succ}}^x |x\rangle |z + f(x)\rangle + \beta_{\text{fail}}^x |x\rangle |\psi(x)\rangle , \quad (7)$$

where $\beta_{\text{succ}}^x, \beta_{\text{fail}}^x \in \mathbb{C}$ such that $|\beta_{\text{succ}}^x|^2 + |\beta_{\text{fail}}^x|^2 = 1$, the normalised state $|\psi(x)\rangle$ could be an arbitrary superposition over $|z + v\rangle$ for vectors $v \in \mathbb{F}^m \setminus \{f(x)\}$, and $+$ denotes component-wise addition for vectors over \mathbb{F} . We can interpret U_f as a quantum analogue of a classical probabilistic algorithm for computing f — for an input $x \in \mathbb{F}^n$, it outputs the correct value of $f(x) \in \mathbb{F}^m$ with probability $|\beta_{\text{succ}}^x|^2$ when the second register is measured in the computational basis.

More generally, we consider oracles that may entangle a workspace register with the output

$$U_f |x\rangle |w\rangle |z\rangle = \beta_{\text{succ}}^x |x\rangle |z + f(x)\rangle |w(x, z, f(x))\rangle + \beta_{\text{fail}}^x |x\rangle |\Psi(x)\rangle , \quad (8)$$

where $|\Psi(x)\rangle$ is now a normalised state of the form

$$|\Psi(x)\rangle = \sum_{\substack{v \in \mathbb{F}^n \\ v \neq f(x)}} \gamma_v^x |z+v\rangle |w(x, z, v)\rangle . \quad (9)$$

Quantum algorithms for linear problems. A quantum algorithm for a linear problem \mathcal{M} outputs the correct answer with some probability arising from measurement. Such an algorithm is represented by a unitary ALG which on an input vector $v \in \mathbb{F}^n$ has the action

$$\text{ALG} |v\rangle |0\rangle = \beta_{\text{succ}}^v |v\rangle |Mv\rangle |w_0(v)\rangle + \beta_{\text{fail}}^v |v\rangle |\Psi(v)\rangle , \quad (10)$$

where $\beta_{\text{succ}}^v, \beta_{\text{fail}}^v \in \mathbb{C}$ are complex amplitudes such that $|\beta_{\text{succ}}^v|^2 + |\beta_{\text{fail}}^v|^2 = 1$, $|w_0(v)\rangle$ is an arbitrary normalised state of a workspace register, and

$$|\beta_{\text{succ}}^v|^2 = \|\mathbb{1} \otimes \Pi_{Mv} \text{ALG} |v\rangle |0\rangle\|^2$$

is the success probability of the algorithm, where $\Pi_{Mv} := |Mv\rangle\langle Mv|$ is an orthogonal projection on to the correct answer subspace of the output register. $|\Psi(v)\rangle$ is a normalised state that is orthogonal to the state $|Mv\rangle$ that encodes the correct matrix vector product, i.e. $\forall v \in \mathbb{F}^n$, $\langle Mv | \Psi(v)\rangle = 0$. In general, it takes the form

$$|\Psi(v)\rangle = \sum_{\substack{z \in \mathbb{F}^n \\ z \neq Mv}} \gamma_z^v |z\rangle |w(v, z)\rangle . \quad (11)$$

4 Quantum worst-case to average-case reductions

In this section, we provide our quantum worst-case to average-case reductions. To obtain our results in a modular way, we start by stating a general technical lemma from which we can easily derive both Theorem 4.2 and Theorem 4.3.

Lemma 4.1. *Let $\mathbb{F} = \mathbb{F}_p$ be a prime field, $n \in \mathbb{N}$, and $\alpha := \alpha(n) \in (0, 1]$. Let ALG^M be a quantum query algorithm that has oracle access to a matrix $M \in \mathbb{F}^{n \times n}$, gets as input a vector $v \in \mathbb{F}^n$, makes $Q(n)$ queries, and attempts to compute Mv . Then, for every constant $\delta > 0$, there exists a quantum algorithm $(\text{ALG}')^M$ that makes $O(\alpha^{-7/2})$ uses of ALG^M , $O((Q(n) + n^{3/2}) \cdot \alpha^{-7/2})$ queries to U_M and U_M^\dagger , uses $O(n^{3/2} \alpha^{-7/2} \cdot \log n \cdot \text{poly}(\log |\mathbb{F}|))$ additional one-qubit and two-qubit gates, and $O(\alpha^{-2} \cdot n \log n)$ ancillary qubits such that the following holds.*

For every matrix $M \in \mathbb{F}^{n \times n}$ such that ALG^M computes Mv correctly with probability α :

$$\Pr_{v, \text{ALG}^M} [\text{ALG}^M(v) = Mv] = \mathbb{E}_{v \in \mathbb{F}^n} \left[\|\Pi_{Mv} \text{ALG}^M |v\rangle |0\rangle\|^2 \right] \geq \alpha ,$$

the algorithm $(\text{ALG}')^M$ computes Mv correctly for every $v \in \mathbb{F}^n$ with probability $1 - \delta$:

$$\forall v \in \mathbb{F}^n \quad \Pr_{\text{ALG}'} [(\text{ALG}')^M(v) = Mv] = \|\Pi_{Mv} (\text{ALG}')^M |v\rangle |0\rangle\|^2 \geq 1 - \delta .$$

Here and in the following, when we say that ALG' makes k calls to ALG , we mean that ALG' on inputs of length n makes k calls to ALG on inputs of length n .

We prove Lemma 4.1 in Section 7, after we develop a toolkit of quantum algorithms in Section 5 and a quantum local correction lemma in Section 6, which will be necessary for our proof.

Equipped with the technical lemma above, in Section 4.1 we formally restate Theorem 1 and show how it follows from Lemma 4.1. Then, in Section 4.2, we formally restate Theorem 2 and show how to prove it using Lemma 4.1.

4.1 Quantum algorithms for all linear problems

In this section we deal with uniform quantum algorithms; that is, uniformly generated families of quantum circuits $\{C_n\}_{n \in \mathbb{N}}$ for each input length $n \in \mathbb{N}$. Following the literature on quantum algorithms for matrix problems we allow C_n to use a unitary gate U_M that computes matrix entries of M as defined in Eq. (4). We obtain the following quantum worst-case to average-case reduction for linear problems.

Theorem 4.2. *Let $\mathbb{F} = \mathbb{F}_p$ be a prime field, $n \in \mathbb{N}$, and $\alpha := \alpha(n) \in (0, 1]$. Let $\mathcal{M} := \{M_n \in \mathbb{F}^{n \times n}\}_{n \in \mathbb{N}}$ be any linear problem, and ALG be an average-case quantum algorithm for \mathcal{M} that takes as input a vector $v \in \mathbb{F}^n$ and satisfies*

$$\Pr_{v, \text{ALG}} [\text{ALG}(v) = Mv] = \mathbb{E}_{v \in \mathbb{F}^n} \left[\|\Pi_{Mv} \text{ALG} |v\rangle |0\rangle\|^2 \right] \geq \alpha .$$

Then, for every constant $\delta > 0$, there exists a worst-case quantum algorithm ALG' satisfying

$$\forall v \in \mathbb{F}^n \quad \Pr_{\text{ALG}'} [\text{ALG}'(v) = Mv] = \|\Pi_{Mv} \text{ALG}' |v\rangle |0\rangle\|^2 \geq 1 - \delta .$$

ALG' makes $O(\alpha^{-7/2})$ uses of ALG , $O(n^{3/2}\alpha^{-7/2})$ uses of U_M and U_M^\dagger , $O(n^{3/2}\alpha^{-7/2} \log n \cdot \text{poly}(\log |\mathbb{F}|))$ additional one-qubit and two-qubit gates, and $O(\alpha^{-2} \cdot n \log n)$ ancillary qubits.

This result follows immediately from Lemma 4.1. Observe that Theorem 4.2 instantiates a quantum query algorithm ALG as a family $\{C_n\}_{n \in \mathbb{N}}$ of quantum circuits, and considers them as having access to explicit circuits implementing U_M . Thus in particular, the guarantees of Lemma 4.1 about the number of queries to U_M and the number of additional gates used by ALG' hold. There are no additional overheads in circuit size from implementing access to sampled vectors, which only cost $\tilde{O}(n)$ gates.

In general, an algorithm that solves a linear problem may exploit the special structure of M , or in the circuit model the matrix entries may be hard-coded. However, note that there is always a trivial circuit of size $O(n^2)$ that solves a linear problem, both in the classical and quantum settings. Our focus in this work is on fine-grained complexity, and our results hold for the stronger case of non-trivial quantum circuits $\{C_n\}_{n \in \mathbb{N}}$ of *sub-quadratic* size.

4.2 Matrix-vector multiplication in the query model

Next, we provide a quantum worst-case to average-case reduction for the Matrix-Vector Multiplication problem in the quantum query model.

Theorem 4.3 (Query complexity of Matrix-Vector Multiplication). *Let $\mathbb{F} = \mathbb{F}_p$ be a prime field, $n \in \mathbb{N}$, and $\alpha := \alpha(n) \in (0, 1]$. Suppose that there exists a quantum query algorithm ALG that has oracle access to a matrix $M \in \mathbb{F}^{n \times n}$ and a vector v , makes $Q(n)$ queries, and satisfies*

$$\Pr_{\substack{M, v, \\ \text{ALG}}} [\text{ALG}^{M, v} = Mv] = \mathbb{E}_{\substack{M \in \mathbb{F}^{n \times n} \\ v \in \mathbb{F}^n}} \left[\|\Pi_{Mv} \text{ALG}^{M, v} |0\rangle\|^2 \right] \geq \alpha .$$

Then, for every constant $\delta > 0$, there exists a worst-case quantum algorithm ALG' that makes $O(\alpha^{-9/2})$ uses of ALG , makes $O((Q(n) + n^{3/2}) \cdot \alpha^{-9/2})$ queries to U_M and U_M^\dagger , and succeeds on all inputs with high probability:

$$\forall M \in \mathbb{F}^{n \times n}, v \in \mathbb{F}^n \quad \Pr_{\text{ALG}'}[(\text{ALG}')^{M,v} = Mv] = \left\| \Pi_{Mv} (\text{ALG}')^{M,v} |0\rangle \right\|^2 \geq 1 - \delta.$$

We will prove this worst-case to average-case reduction for the Matrix-Vector Multiplication problem in the quantum query model using Lemma 4.1 and an efficient quantum verification algorithm for matrix-vector products (see Lemma 2.4 from Section 2.3, or Lemma 5.1 for a formal statement).

Proof of Theorem 4.3. The algorithm $(\text{ALG}')^{M,v}$ repeats the following procedure $O(1/\alpha)$ times. Sample a uniformly random matrix $R \in \mathbb{F}^{n \times n}$, define $M' = M - R$, and generate unitary oracle $U_{M'}$ for M' . Each query to $U_{M'}$ costs only one query to U_M . Now use Lemma 4.1 with $U_{M'}$ to try to compute $b' = M' \cdot v$. Finally, compute $b_R = R \cdot v$ directly without any queries to M , and set $b = b' + b_R$. Verify whether $b = Mv$ using Lemma 5.1 with $\varepsilon = \delta/2$, and output b if the verification test passed.

The complexity of the algorithm is determined by $O(1/\alpha)$ applications of Lemma 4.1 and verifications from Lemma 5.1, and $O(n)$ queries to read the coordinates of v . In particular, the described algorithm $(\text{ALG}')^{M,v}$ performs $O(\alpha^{-9/2})$ calls of $\text{ALG}^{M,v}$, and $O((Q(n) + n^{3/2}) \cdot \alpha^{-7/2})$ queries to U_M and U_M^\dagger .

In order to analyze the correctness of the algorithm $(\text{ALG}')^{M,v}$, we introduce the following notation. For a fixed matrix $M \in \mathbb{F}^{n \times n}$, let $p_M = \Pr_{v, \text{ALG}}[\text{ALG}^{M,v} = Mv]$ denote the probability of computing $\text{ALG}^{M,v} = Mv$ correctly for the fixed value of M and a uniformly random vector $v \in \mathbb{F}^n$. Let us define the set $X \subseteq \mathbb{F}^{n \times n}$ of good matrices where the average-case algorithm ALG succeeds with probability at least $\alpha/2$:

$$X := \left\{ M \in \mathbb{F}^{n \times n} : p_M \geq \frac{\alpha}{2} \right\}.$$

First we observe that $|X| \geq \alpha |\mathbb{F}|^{n^2} / 2$. Indeed, by the assumption of the theorem we have that $\mathbb{E}_{M \in \mathbb{F}^{n \times n}} [p_M] \geq \alpha$. Then,

$$\alpha \leq \mathbb{E}_M [p_M] < 1 \cdot \Pr_M [p_M \geq \alpha/2] + \alpha/2 \cdot \Pr_M [p_M < \alpha/2] \leq \Pr_M [p_M \geq \alpha/2] + (\alpha/2) \cdot 1.$$

Thus, $|X| = |\mathbb{F}|^{n^2} \cdot \Pr_M [p_M \geq \alpha/2] \geq \alpha |\mathbb{F}|^{n^2} / 2$.

Since in every iteration of the algorithm, the matrix $M' = M - R$ is a uniformly random matrix, we have that $\Pr[M' \in X] \geq \alpha/2$. In the case when $M' \in X$, Lemma 4.1 can correctly compute $b' = M'v$ with probability $2/3$. Thus, in every iteration, the described procedure computes $b = Mv$ with probability $\Omega(\alpha)$. By repeating this procedure $O(1/\alpha)$ times (each time verifying whether $b = Mv$ using Lemma 5.1 with $\varepsilon = \delta/2$), we have that for every matrix M and every vector v , we compute the product Mv correctly with probability $1 - \delta$. \square

5 Quantum toolkit for local correction

In this section, we provide a toolkit of quantum algorithms that will allow us to later obtain a quantum local correction lemma (in Section 6), which will underlie our quantum worst-case to

average-case reductions. Specifically, in Section 5.1, we construct a quantum procedure for flagging correct matrix-vector products in superposition; we use this in Section 5.2 to obtain a unitary implementation of an approximation to the indicator function on the set X of good vectors, and in Section 5.3 to provide a quantum procedure for efficiently sampling from X ; finally, in Section 5.4, we construct a quantum procedure for learning Bogolyubov subspaces from noisy quantum oracles.

5.1 Flagging correct matrix-vector products in superposition

As discussed in the technical overview in Section 2, a key bottleneck in the classical setting is the efficient verification of computing a matrix-vector product. In the setting of quantum query complexity, this is the following problem:

MATRIX-VECTOR PRODUCT VERIFICATION (MvPV)

Input: Quantum oracles U_M , U_v , and U_b for a matrix $M \in \mathbb{F}^{n \times n}$ and vectors $v, b \in \mathbb{F}^n$.

Output: 1 if $Mv = b$ and 0 otherwise.

Classical query algorithms would analogously have access to an oracle that returns matrix (vector) entries when queried with a row and column index pair.

The quantum query complexity of MvPV has been studied in detail, starting with the work of [BŠ06] who showed that matrix-matrix products can be verified with $O(n^{5/3})$ quantum queries, a bound that is sublinear in the size of the input (which is n^2). [MNR⁺07] later showed that the classical techniques of Freivalds can be adapted to the quantum setting to make this algorithm time-efficient. The special case of matrix multiplication over the Boolean semiring has close relations to path and triangle finding in graphs, and its quantum query complexity has also been studied in great depth [MSS07; WW10; Le 12; CKK12; JKM]. [Kot14], gives a detailed review of the complexities and relationships between different variants of the matrix multiplication and MvPV problems over arbitrary semirings.

Problem	Inputs/Output	Quantum query complexity
Matrix-vector Multiplication (MvM)	In: $M \in \mathbb{F}^{n \times n}, v \in \mathbb{F}^n$ Out: Mv	$\Theta(n^2)$
Matrix-vector Product Verification (MvPV)	In: $M \in \mathbb{F}^{n \times n}, v, b \in \mathbb{F}^n$ Out: $Mv = b?$	$\Theta(n^{3/2})$

Table 1: Worst-case quantum query complexity of MvM and MvPV [Kot14].

This extensive literature focuses on the more general problem of matrix-matrix multiplication and product verification over semirings, especially the Boolean case; on the other hand, our interest here lies in the special case of matrix-vector products over finite fields \mathbb{F}_p . In particular, all the results about matrix-vector product verification that we have come across in past work deal exclusively with the query complexity, leaving the algorithm achieving the upper bound, and its computational or gate complexity, implicit. Furthermore, techniques such as those used in [BŠ06] do not extend well to our setting — (1) because they use the computation of a logical AND via Grover search, they do not directly lead to a unitary algorithm that can be queried in superposition,

due to the traditional problems of overshooting associated with vanilla quantum search; and (2) because they are phrased in the usual manner of performing amplitude amplification followed by measurements to extract the output with constant success probability.

In this section, we construct an efficient quantum algorithm for addressing MvPV, with query and gate complexities bounded by the optimal $O(n^{3/2})$, which will be conducive to querying in superposition and consequently to composition with our subsequent subroutine for learning Bogolyubov subspaces. As discussed in Section 2.3, the main technical ingredients we use are fixed-point amplitude amplification and quantum singular value threshold projection.

5.1.1 A simple case of MvPV

We first consider the standard setting where the input vectors v and b are both given by the usual exact quantum query oracles. We later use this simple variant to argue about the case where we only have a noisy version of b , accessed via a noisy quantum oracle.

Lemma 5.1 (Quantum MvPV). *Suppose we are given a quantum oracle U_M for a matrix $M \in \mathbb{F}^{n \times n}$*

$$U_M |j, k, z\rangle = |j, k, z \oplus M_{jk}\rangle \quad (12)$$

for all indices $j, k \in [n]$ and $z \in \mathbb{F}$, and quantum oracles U_v, U_b for input vectors $v, b \in \mathbb{F}$

$$U_v |j, z\rangle = |j, z \oplus v_j\rangle \quad (13)$$

$$U_b |j, z\rangle = |j, z \oplus b_j\rangle \quad (14)$$

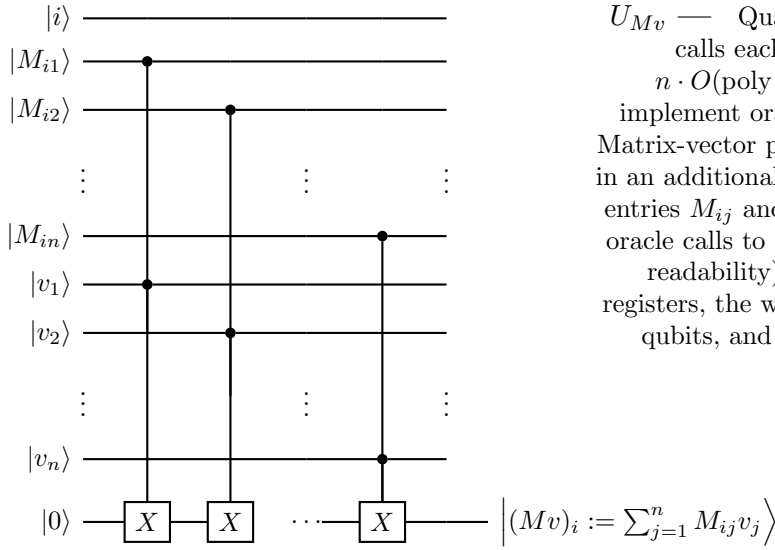
for all $j \in [n]$ and $z \in \mathbb{F}$. Then there is a gate-efficient quantum algorithm $\mathcal{Q}_{\text{verify}}$ that accepts with certainty if $Mv = b$, and rejects with probability $1 - \varepsilon$ if $Mv \neq b$. Furthermore, $\mathcal{Q}_{\text{verify}}$ uses $q = O(n^{3/2} \cdot \log \frac{1}{\varepsilon})$ queries to U_M and U_M^\dagger , $O(n)$ queries to U_v, U_b and their Hermitian conjugates, $O(q \log n \cdot \text{poly log } |\mathbb{F}|)$ additional one-qubit and two-qubit gates, and $O(n \log |\mathbb{F}|)$ ancillary qubits.

Proof. We denote multiplication and controlled addition over a finite field by a CCX gate defined by

$$\text{CCX } |s_1\rangle |s_2\rangle |z\rangle = |s_1\rangle |s_2\rangle |z + s_1 \cdot s_2\rangle, \quad (15)$$

where $s_1, s_2, z \in \mathbb{F}$ and $+$ and \cdot are addition and multiplication in \mathbb{F} . Implementing such an operation requires only $O(\text{poly log } |\mathbb{F}|)$ elementary two-qubit gates [BBF03]. Throughout this paper we assume that all algorithms use quantum registers of dimension $p = |\mathbb{F}|$ and perform arithmetic over \mathbb{F} , since such arithmetic can be simulated using $O(\log |\mathbb{F}|)$ qubits with $O(\text{poly log } |\mathbb{F}|)$ overhead in gate complexity.

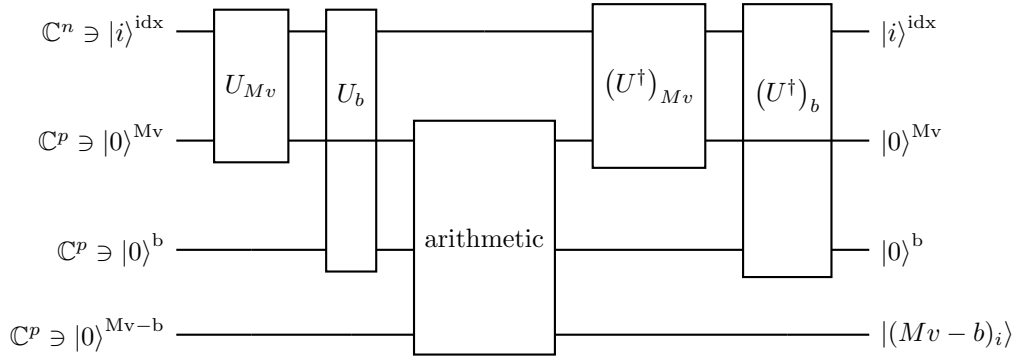
Circuit U_{Mv} : Quantum circuit for entrywise Matrix-vector product



U_{Mv} — Quantum circuit that uses n oracle calls each to $U_M, U_v, U_M^\dagger, U_v^\dagger$, and $n \cdot O(\text{poly log } |\mathbb{F}|)$ elementary gates, to implement oracle access to the entries of the Matrix-vector product Mv . The index i is given in an additional register, using which the matrix entries M_{ij} and vector entries V_j are loaded by oracle calls to U_M and U_v (suppressed here for readability); accounting for all ancillary registers, the width of the circuit is $O(n \log |\mathbb{F}|)$ qubits, and its depth is $O(n \text{ poly log } |\mathbb{F}|)$.

The unitaries U_{Mv}, U_b and their Hermitian conjugates can be used once each, along with quantum arithmetic operations over \mathbb{F} using circuits of size $O(\text{poly log } |\mathbb{F}|)$, to obtain a similar oracle U_{Mv-b} for the vector $Mv - b$. The registers used for computing $(Mv)_i$ and b_i can both be perfectly uncomputed and returned to their initial value in this case. Using quantum adder circuits [Gid18] it is possible to construct a comparator circuit with $O(1)$ ancillas and $O(\log |\mathbb{F}|)$ gates that checks whether an entry $(Mv - b)_i$ is zero, and sets a flag qubit to 1 if not. This becomes an oracle to the component-wise indicator function $1_{(Mv-b)} : [n] \times \mathbb{F} \rightarrow \{0, 1\}$ mapping an index $i \in [n]$ to 0 if $(Mv)_i = b_i$, and 1 otherwise. Denote this oracle by $U_{Mv \stackrel{?}{=} b}$.

Circuit U_{Mv-b} : Quantum circuit for $Mv - b$



U_{Mv-b} — Quantum circuit to implement oracle access to the entries of $Mv - b$ (where we suppress the workspace registers required by U_{Mv} etc.). The width of the circuit is $O(n \log |\mathbb{F}|)$ qubits, and the arithmetics step can be implemented with circuits of size $O(\text{poly log } |\mathbb{F}|)$.

We can then perform a quantum search for nonzero entries in $Mv - b$ using $U_{Mv \stackrel{?}{=} b}$. We start by preparing the uniform superposition over indices, query $U_{Mv \stackrel{?}{=} b}$, and treat nonzero entries as marked.

Define the states $|\psi_0\rangle$, $|\psi_1(v, b)\rangle$, and $|\psi_0(v, b)\rangle$ as the uniform superposition over all indices, and uniform superpositions over indices at which $Mv - b$ has nonzero and zero entries respectively

$$\begin{aligned} |\psi_0\rangle &= \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle \\ |\psi_1(v, b)\rangle &= \frac{1}{\sqrt{m_{vb}}} \sum_{\substack{i \in [n] \\ (Mv)_i \neq b_i}} |i\rangle \\ |\psi_0(v, b)\rangle &= \frac{1}{\sqrt{n - m_{vb}}} \sum_{\substack{i \in [n] \\ (Mv)_i = b_i}} |i\rangle, \end{aligned} \tag{16}$$

and $m_{vb} = |\{i \in [n] : (Mv)_i \neq b_i\}|$. Let $k := \lceil \log n \rceil$. Without loss of generality we can assume n is a power of two since we can pad vectors with zeros if not, while at most doubling the complexity. Consider the operator that creates the uniform superposition over indices and queries $Mv \stackrel{?}{=} b$, i.e.

$$U = \left(H^{\otimes k} \otimes \mathbb{1} \right) U_{Mv \stackrel{?}{=} b}.$$

We omit the workspace registers used by $U_{Mv \stackrel{?}{=} b}$ because in this case the workspace can be uncomputed and returned to the initial all-zeros state.

The action of U on the $|0^{k+1}\rangle$ state is

$$U |0^{k+1}\rangle = \begin{cases} |\psi_0\rangle |0\rangle & Mv = b \\ \sqrt{\frac{n - m_{vb}}{n}} |\psi_0(v, b)\rangle |0\rangle + \sqrt{\frac{m_{vb}}{n}} |\psi_1(v, b)\rangle |1\rangle & Mv \neq b. \end{cases} \tag{17}$$

Since we do not know m_{vb} beforehand, and since we will need a unitary subroutine that is run over a superposition of input vectors in \mathbb{F}^n later on, we use fixed point amplitude amplification [Gro05; YLC14; Gue19; GSL⁺19]. This allows the number of iterations of amplitude amplification to be chosen uniformly for all vectors v without worrying about the problem of under- or overshooting faced in vanilla amplitude amplification. For completeness, we give a brief overview and technical statement of fixed-point amplitude amplification in Appendix A.

The unitary U of Eq. (17) satisfies the conditions required by Theorem A.1 with $\Pi = \mathbb{1}_k \otimes |1\rangle\langle 1|$, preparing a state that has a component flagged with $|1\rangle$ when $Mv \neq b$. In the worst case for $Mv \neq b$, there exists exactly one coordinate $i \in [n]$ at which $(Mv)_i \neq b_i$, so that $m_{vb} = 1$. We hence use fixed point amplification with the worst-case lower bound of $p > \frac{1}{2\sqrt{n}}$ to obtain a unitary U' with the action

$$U' |0^{k+1}\rangle = \begin{cases} |\psi_0\rangle |0\rangle & Mv = b \\ \gamma_{\text{fail}}^{vb} |\psi_0(v, b)\rangle |0\rangle + \gamma_{\text{succ}}^{vb} |\psi_1(v, b)\rangle |1\rangle & Mv \neq b, \end{cases} \tag{18}$$

where $|\gamma_{\text{succ}}^{vb}|^2 \geq 1 - \varepsilon$. This unitary U' makes $q = O(\sqrt{n} \cdot \log \frac{1}{\varepsilon})$ queries to U and U^\dagger , uses $O(q \log n)$ additional elementary gates, and a single ancilla. The only queries in U are those made by $U_{Mv=b}$ to U_M, U_v, U_b and their conjugates; since $U_{Mv=b}$ uses n queries to each of these oracles, the net query complexity of U' is $q' = O(nq) = O(n^{3/2})$. Finally since M is of size n^2 and $q' = O(n^{3/2})$ we can always first query v and b into an ancillary register, since both are vectors of length n .

Effectively, for any desired success probability $\varepsilon \in (0, 1)$ we are able to obtain a unitary $\mathbf{Q}_{\text{verify}} := U'$ that computes $\mathbf{1}_{Mv=b}$ with one-sided bounded error, measuring the output register of which yields the claimed guarantees. \square

5.1.2 Flagging the correct matrix-vector product in superposition

In order to establish our reduction, we need to go beyond verifying whether a certain vector, obtained by running ALG on an input $v \in \mathbb{F}^n$ and measuring its output register, is correct. In particular, we would like to compute the indicator function $\mathbf{1}_X$ which marks the vectors on which ALG succeeds with high probability in superposition over all $v \in \mathbb{F}^n$, in order to learn its Fourier characters of high weight. This is complicated twofold — by the noise or part of the state associated with failure in the output of ALG, and the entanglement with workspace registers. Nevertheless, we can compute an approximate, noisy version of the indicator function on good input vectors. As a first step, we construct an algorithm using ideas from Lemma 5.1 to ensure that whenever ALG outputs the correct answer, it also outputs a flag indicating success.

Lemma 5.2 (Noisy quantum MvPV). *Suppose we are given a quantum oracle U_M for a matrix $M \in \mathbb{F}^{n \times n}$*

$$U_M |j, k, z\rangle = |j, k, z \oplus M_{jk}\rangle \quad (19)$$

for all indices $j, k \in [n]$ and $z \in \mathbb{F}$, and a noisy quantum algorithm ALG as described in Eq. (10), i.e.

$$\text{ALG} |v\rangle |0\rangle = \beta_{\text{succ}}^v |v\rangle |Mv\rangle |w_0(v)\rangle + \beta_{\text{fail}}^v |v\rangle |\Psi(v)\rangle.$$

Then there exists a gate-efficient quantum algorithm $\text{ALG}_{\text{verified}}$ that succeeds and outputs Mv with probability $|\beta_{\text{succ}}^v|^2$ along with a flag indicating success, and similarly outputs a flag indicating failure whenever it outputs a vector $z \neq Mv$, with probability at least $(1 - \varepsilon)|\beta_{\text{fail}}^v|^2$. Furthermore, $\text{ALG}_{\text{verified}}$ uses $q = O(n^{3/2} \cdot \log \frac{1}{\varepsilon})$ queries to U_M and U_M^\dagger , $O(1)$ queries to ALG, $O(q \log n \cdot \text{poly log } |\mathbb{F}|)$ additional one-qubit and two-qubit gates, and $O(n \log |\mathbb{F}|)$ ancillary qubits.

Remark 5.3. *The input vectors v and b are taken to be given as states $|v\rangle$ and the right hand side of Eq. (10)—created using a single query to ALG—respectively. Using a hard-coded circuit of size $O(n \log n)$ we can implement entrywise oracles U_v and U_b that copy out the i^{th} entries of these vectors into a target register controlled on an index register.*

Proof. If ALG had the property that either $\beta_{\text{fail}}^v = 0$ or $\beta_{\text{succ}}^v = 0$ always, then we could have proceeded as in Lemma 5.1. Nevertheless, if we first prepare the state $\text{ALG} |v\rangle |0\rangle$, prepare a superposition over indices $i \in [n]$ in an ancillary register, and compute the indicator function $\mathbf{1}_{(Mv=b)}$ using fixed circuits of size $O(n \log n)$ gates to copy (controlled on an index register) the i^{th}

entry from the output registers of **ALG**, we now have a unitary U that prepares the state

$$\begin{aligned}
& U \left(|v\rangle \otimes \left(\beta_{\text{succ}}^v |Mv\rangle |w_0(v)\rangle + \beta_{\text{fail}}^v \left(\sum_{\substack{z \in \mathbb{F}^n \\ z \neq Mv}} \gamma_z^v |z\rangle |w(v, z)\rangle \right) \right) \otimes |0^{k+1}\rangle \right) \\
&= |v\rangle \otimes \left(\beta_{\text{succ}}^v |Mv\rangle |w_0(v)\rangle |\psi_0\rangle |0\rangle + \right. \\
&\quad \left. \beta_{\text{fail}}^v \left(\sum_{\substack{z \in \mathbb{F}^n \\ z \neq Mv}} \gamma_z^v |z\rangle |w(v, z)\rangle \left[\sqrt{\frac{n - m_{vz}}{n}} |\psi_0(v, z)\rangle |0\rangle + \sqrt{\frac{m_{vz}}{n}} |\psi_1(v, z)\rangle |1\rangle \right] \right) \right), \tag{20}
\end{aligned}$$

where the states $|\psi_0\rangle, |\psi_0(v, z)\rangle, |\psi_1(v, z)\rangle$ are defined analogously to Eq. (16), m_{vz} is the number of indices at which a vector z disagrees with the correct answer Mv , and to avoid clutter we have not explicitly written the registers used for querying the entries M_{ij} , since they can be uncomputed as before.

As in Lemma 5.1, we can apply fixed-point amplitude amplification to the unitary U , with the goal of amplifying the part of the superposition flagged by 1 in the last qubit. Incorrect answers $z \neq Mv$ may in general be wrong only at a single co-ordinate, so that in the worst case there may be a single $\gamma_z^v = 1$ with the corresponding $m_{vz} = 1$. Thus we can use the worst-case lower bound of $\frac{1}{\sqrt{n}}$ on the amplitude of the target state flagged by 1. Since U does not affect the state of the input, output, or workspace registers of **ALG**, the amplified operator $\text{ALG}_{\text{verified}} := U'$ will also preserve the same superposition that is produced by **ALG**, with the component of the state flagged by 1 in the final register amplified:

$$\begin{aligned}
& \text{ALG}_{\text{verified}} \left(|v\rangle \otimes \left(\beta_{\text{succ}}^v |Mv\rangle |w_0(v)\rangle + \beta_{\text{fail}}^v \left(\sum_{\substack{z \in \mathbb{F}^n \\ z \neq Mv}} \gamma_z^v |z\rangle |w(v, z)\rangle \right) \right) \otimes |0^{k+1}\rangle \right) \\
&= |v\rangle \otimes \left(\beta_{\text{succ}}^v |Mv\rangle |w_0(v)\rangle |\psi_0\rangle |0\rangle + \right. \\
&\quad \left. \beta_{\text{fail}}^v \left(\sum_{\substack{z \in \mathbb{F}^n \\ z \neq Mv}} \gamma_z^v |z\rangle |w(v, z)\rangle \left[\gamma_{\text{fail}}^{vz} |\psi_0(v, z)\rangle |0\rangle + \gamma_{\text{succ}}^{vz} |\psi_1(v, z)\rangle |1\rangle \right] \right) \right), \tag{21}
\end{aligned}$$

where $|\gamma_{\text{succ}}^{vz}|^2 > 1 - \varepsilon$, and $\text{ALG}_{\text{verified}}$ uses $q = O(n^{3/2} \cdot \log \frac{1}{\varepsilon})$ queries to U_M and U_M^\dagger , $O(1)$ queries to **ALG**, and $O(q \log n \cdot \text{poly log } |\mathbb{F}|)$ additional elementary gates, and a single ancillary qubit.

Measuring the register containing the output of **ALG**, we see that $\text{ALG}_{\text{verified}}$ outputs Mv with probability at least $|\beta_{\text{succ}}^v|^2$ just as **ALG** does, but now the last register contains a flag qubit set to

zero to indicate success. When $\text{ALG}_{\text{verified}}$ outputs an incorrect vector $z \neq Mv$, with probability at least $(1 - \varepsilon)|\beta_{\text{succ}}^v|^2$ the flag qubit is set to 1 to indicate failure, giving the guarantees claimed in the lemma. \square

5.2 Noisy quantum oracles approximating the indicator function 1_X

To obtain our next two quantum procedures: an efficient sampler for the set X of good inputs to ALG and a learner for the Bogolyubov subspace, we first need to obtain a unitary implementation of an approximate version of the indicator 1_X , which is a Boolean valued function defined on \mathbb{F}^n . We start by observing that the subroutine $\text{ALG}_{\text{verified}}$ that we constructed in the previous section almost has the basic property that a unitary implementing 1_X should have: it attaches flags zero and one to vectors in X and its complement respectively. However, it is *noisy* in two ways: it errs with high probability on vectors outside X , and only succeeds with low probability on vectors in X . What we would like is (a noisy version of) the oracle

$$U_X |v\rangle |0\rangle = |v\rangle |1_X(v)\rangle.$$

To obtain such a unitary, we use the machinery of quantum singular value threshold projection on top of $\text{ALG}_{\text{verified}}$. We now go into the details of our construction below, and give a brief technical statement of the quantum singular value threshold projection technique in Appendix A.

For an average-case algorithm ALG with average success probability α as defined in Eq. (6), recall that we are interested in the associated set of “good” input vectors defined by

$$X = \{v \in \mathbb{F}^n : |\beta_{\text{succ}}^v|^2 \geq \frac{\alpha}{2}\},$$

which we know has density at least $\alpha/2$ in \mathbb{F}^n , by the averaging argument in Claim 7.2. The indicator function of this set takes the value $1_X(v) = 1$ when $v \in X$ and $1_X(v) = 0$ otherwise. Denote by U the algorithm $\text{ALG}_{\text{verified}}$ constructed in Lemma 5.2. Bundling all the workspace registers together for brevity, we note that it has the following property: defining the projectors $\Pi = \mathbb{1}^v \otimes |0\rangle\langle 0|^{\text{work,flag}}$ and $\tilde{\Pi} = \mathbb{1} \otimes |0\rangle\langle 0|^{\text{flag}}$, where the identity term acts on all registers except the flag qubit, we have that

$$\tilde{\Pi} U \Pi = \sum_{v \in \mathbb{F}^n} |\beta_{\text{succ}}^v| |w_v\rangle\langle v, 0, 0|.$$

We interpret the right hand side above as the singular value decomposition of a matrix with right singular vectors $|v, 0, 0\rangle$, and left singular vectors $|w_v\rangle$ given by

$$|w_v\rangle = \frac{1}{|\beta_{\text{succ}}^v|} \left(\beta_{\text{succ}}^v |v, Mv, w_0(v), \psi_0\rangle + \beta_{\text{fail}}^v \sum_{\substack{z \in \mathbb{F}^n \\ z \neq Mv}} \gamma_z^v \gamma_{\text{fail}}^{vz} |v, z, w(v, z), \psi_0(v, z)\rangle \right) \otimes |0\rangle^{\text{flag}}. \quad (22)$$

The singular values $|\beta_{\text{succ}}^v|$ are defined by the relation

$$|\beta_{\text{succ}}^v|^2 = |\beta_{\text{succ}}^v|^2 + |\beta_{\text{fail}}^v|^2 \sum_{\substack{z \in \mathbb{F}^n \\ z \neq Mv}} |\gamma_z^v|^2 |\gamma_{\text{fail}}^{vz}|^2. \quad (23)$$

Fixing a threshold parameter $t \in (0, 1)$, consider the partition of \mathbb{F}^n into the following three sets:

1. a good set $X_t^g = \{v \in \mathbb{F}^n : |\beta_{\text{succ}}^v|^2 \geq t + t^2\}$

2. an intermediate set $W_t = \{v \in \mathbb{F}^n : |\beta_{\text{succ}}^v|^2 \in (t - 2t^2, t + t^2)\}$
3. a bad set $X_t^b = \{v \in \mathbb{F}^n : |\beta_{\text{succ}}^v|^2 \leq t - 2t^2\}$.

Choosing $\varepsilon = t^2$ in Lemma 5.2 we have that when $v \in X_t^g$, $|\beta_{\text{succ}}^v|^2 \geq t + t^2$, and when $v \in X_t^b$, $|\beta_{\text{succ}}^v|^2 \leq t - t^2$.

We see that $U = \text{ALG}_{\text{verified}}$ hence satisfies the conditions required in Theorem A.2, so that we can use it to select the vectors in X_t^g with high probability using the technique of singular value threshold projection. Since $\sqrt{t + t^2} \geq \sqrt{t} \left(1 + \frac{1}{2}t - \frac{1}{8}t^2\right)$, we can choose the thresholds in Theorem A.2 to be t as above, and $\delta = \frac{1}{2}t^{3/2} - \frac{1}{8}t^{5/2}$, and obtain a unitary U_q with the action

$$U_q |v, 0, 0\rangle = \tilde{\beta}_{\text{succ}}^v |w_v\rangle |0\rangle^{\text{flag}} + \tilde{\beta}_{\text{fail}}^v |\Psi_v\rangle |1\rangle^{\text{flag}}, \quad (24)$$

where the amplitudes satisfy the following guarantees ($\Pr(\text{flag} = 0) := |\tilde{\beta}_{\text{succ}}^v|^2$ etc):

1. for inputs $v \in X_t^g$, $\Pr(\text{flag} = 0) \geq 1 - 2\varepsilon$;
2. for inputs $v \in X_t^b$, $\Pr(\text{flag} = 1) \geq 1 - 2\varepsilon$.

Importantly, we note that for inputs $v \in W_t$, we get no useful guarantee other than the consistency condition $\Pr(\text{flag} = 0) \leq 1$. The unitary U_q can be implemented using $q = O\left(\frac{1}{\delta} \log \frac{1}{\varepsilon}\right) = O\left(\frac{1}{t^{3/2}} \log \frac{1}{\varepsilon}\right)$ queries to U , i.e. to $\text{ALG}_{\text{verified}}$, and $O(q)$ additional one-qubit and two-qubit gates.

U_q represents a noisy version of the indicator function on X_t^g — applying a Pauli-X gate to flip the flag qubit, U_q is a noisy quantum oracle for $\mathbf{1}_X$ as defined in Eq. (8), with one additional complication: there is a “wasteland” slice $W \subseteq \mathbb{F}^n$ on which U_q gives no guarantee as to the value of $\mathbf{1}_X$. Note that this can in principle be a very large set — by inspecting the averaging argument in Claim 7.2 we see that it is possible to construct adversarial examples of ALG for which W can have density as large as $\frac{1-\alpha}{1-\alpha/2}$ for $t = \alpha/2$. Nevertheless, we will show how to overcome this difficulty in Section 7.1.

5.3 Quantum sampling from the set of good inputs

Using the subroutine $\text{ALG}_{\text{verified}}$, it is also possible to sample from the set of vectors on which ALG succeeds with at least a desired probability. Suppose we are interested in the set $X_\tau = \{v \in \mathbb{F}^n : |\beta_{\text{succ}}^v|^2 > \tau\}$ for some $\tau \in (0, 1)$. We use the same ideas that we presented in the previous section — using singular value threshold projection from Theorem A.2 with the choice of threshold $t = \sqrt{\tau}$ and $\delta = \eta t$ for some $\eta > 0$ and so $q = O\left(\frac{1}{\eta\sqrt{\tau}} \log \frac{1}{\varepsilon}\right)$, we get a unitary U_q that implements a noisy version of an approximation to $\mathbf{1}_X$. It is sufficient to choose constant η , e.g. $\eta = 0.01$, because unlike in the case of the indicator function in the previous section, we do not require high precision with respect to the wasteland slice W_t : instead of sampling from X_τ , we can easily work with $X_{\tau'} \subseteq X_\tau$ for $\tau' = (1 + \eta)\tau$ without any difficulties.

Suppose we prepare the uniform superposition over all $v \in \mathbb{F}^n$ and run U_q . If the density of X_τ is $\mu(X_\tau)$, we see that with the projector $\tilde{\Pi} = \mathbf{1} \otimes |0\rangle\langle 0|^{\text{flag}}$ we have $\left\| \langle X_\tau, 0^{\text{flag}} | \tilde{\Pi} U_q | 0 \rangle \right\|^2 \geq (1 - \varepsilon)\mu(X_\tau)$, where

$$|X_\tau\rangle := \frac{1}{\sqrt{|X_\tau|}} \sum_{v \in X_\tau} |v\rangle$$

is the uniform superposition over vectors in X_τ . In particular, we can apply fixed-point amplitude amplification by Theorem A.1 to U_q , to obtain a boosted unitary U'_q with $q' = O\left(\frac{1}{\sqrt{\mu(X_\tau)}} \log \frac{1}{\delta}\right)$ uses of U_q and U_q^\dagger , such that $\left\| \tilde{\Pi} U'_q |0\rangle - |X_\tau, 0^{\text{flag}}\rangle \right\| \leq \delta$. Note that this procedure gives us a way to sample approximately from the uniform distribution on X_τ that is quadratically faster than a classical algorithm that draws random samples and verifies whether the ALG computes correctly on the drawn sample. Applying this argument to the case where $\tau = \alpha/2$, we have the following corollary.

Corollary 5.4. *Given an average-case quantum algorithm ALG with oracle access to the matrix $M \in \mathbb{F}^{n \times n}$ via U_M , there is a quantum algorithm Q_{samp} that uses ALG $O\left(\frac{1}{\sqrt{\tau\mu(X_\tau)}}\right)$ times and with high probability outputs a vector $v \in X_\tau$ on which ALG succeeds with probability at least τ . Q_{samp} makes $q = O\left(n^{3/2} \cdot \frac{1}{\sqrt{\tau\mu(X_\tau)}}\right)$ queries to U_M and U_M^\dagger , $O\left(q \log n \cdot \text{poly log } |\mathbb{F}|\right)$ additional one-qubit and two-qubit gates, and $O(n \log |\mathbb{F}|)$ ancillary qubits.*

5.4 Learning Bogolyubov subspaces from noisy quantum oracles

We would next like to design a quantum procedure for efficiently learning the Bogolyubov subspace from the noisy quantum oracle we constructed above in Section 5.2. Since the Bogolyubov subspace is characterised by the Fourier coefficients of the indicator 1_X , our problem reduces to learning the Fourier spectrum of functions that are encoded in noisy quantum oracles of the aforementioned form.

[AC02] studied the problem of using noisy evaluations of a linear Boolean function $f(x) = a \cdot x$ to determine the underlying string $a \in \mathbb{F}_2^n$, given a guarantee that the evaluations have an average probability of at least $\frac{1}{2} + \varepsilon$ of being correct over random $x \in \mathbb{F}_2^n$. Here we show that the circuit that they consider can be used more generally to sample characters from the Fourier spectrum of a function $f : \mathbb{F}^n \rightarrow \mathbb{F}_2$ that is accessed via a noisy quantum oracle of the type defined in Eq. (8). In particular, we have the following result.

Lemma 5.5. *Suppose we are given as input a noisy quantum oracle U_f to $f : \mathbb{F}^n \rightarrow \mathbb{F}_2$, such that*

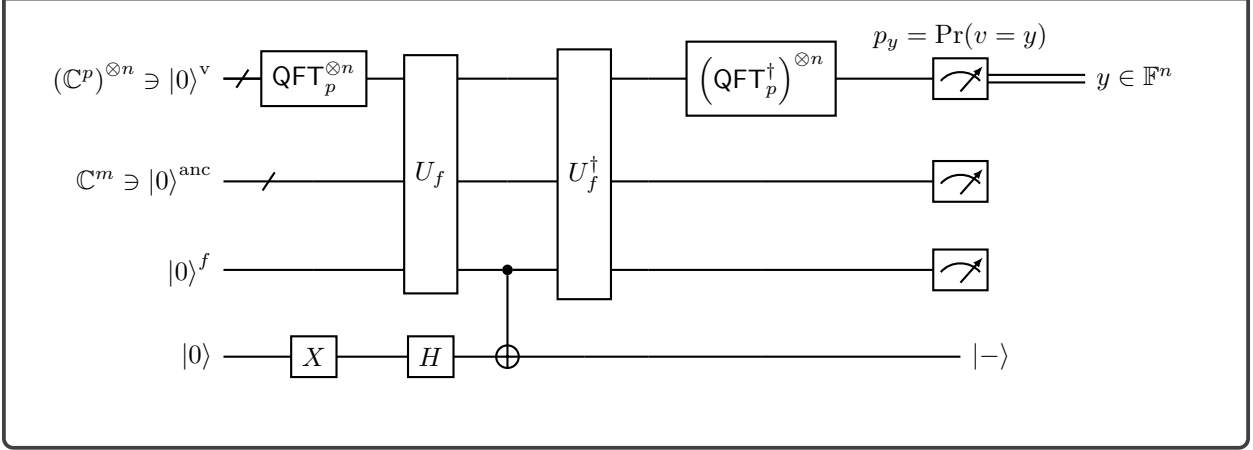
$$U_f |x\rangle |0^{m+1}\rangle = |x\rangle \left(\beta_{\text{succ}}^x |w_0(x)\rangle |f(x)\rangle + \beta_{\text{fail}}^x |w_1(x)\rangle |\overline{f(x)}\rangle \right), \quad (25)$$

where $\forall x \in \mathbb{F}^n$, $|\beta_{\text{succ}}^x|^2 + |\beta_{\text{fail}}^x|^2 = 1$ and $|\beta_{\text{succ}}^x|^2 \geq 1 - \varepsilon$, $\overline{f(x)} = f(x) \oplus 1$ is NOT($f(x)$), and $|w_0(x)\rangle$ and $|w_1(x)\rangle$ are arbitrary m -qubit states of the workspace register. Then measuring the top three registers of the following circuit C_{GL} produces output $y \in \mathbb{F}^n$, 0^{m+1} with probability p_y such that $\forall y \in \mathbb{F}^n$,

$$\left| |\widehat{f}(y)|^2 - p_y \right| \leq 4\varepsilon,$$

where $\widehat{f}(y)$ are the Fourier coefficients of f , i.e. $f(x) = \sum_{y \in \mathbb{F}^n} \widehat{f}(y) \chi_y(x)$ with $\chi_y(x) := \omega^{-x \cdot y}$.

Circuit C_{GL} : Quantum Goldreich-Levin algorithm for Fourier sampling



Note that the number m of ancillary qubits in the workspace register need not have any relation to n .

Remark 5.6. In our application, we actually have a weaker assumption on the input U_f , which in fact is the indicator function of Section 5.2. The guarantee we obtain on the Fourier sampler is then

$$\left| |\widehat{f}(y)|^2 - p_y \right| \leq 4\varepsilon + 4\rho_W,$$

where $\rho_W := |W|/|\mathbb{F}^n|$ is the density of the intermediate set W .

Proof. The input state undergoes the following transformations through the circuit C_{GL} :

$$\begin{aligned} |0\rangle |0\rangle |0\rangle |0\rangle &\xrightarrow{\text{QFT}_p^{\otimes n} \otimes \mathbb{1} \otimes \mathbb{1} \otimes X} \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{F}^n} |x\rangle |0\rangle |0\rangle |1\rangle \\ &\xrightarrow{U_f \otimes H} \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{F}^n} |x\rangle \left(\beta_{\text{succ}}^x |w_0(x)\rangle |f(x)\rangle + \beta_{\text{fail}}^x |w_1(x)\rangle |\overline{f(x)}\rangle \right) |-\rangle \\ &\xrightarrow{\mathbb{1} \otimes \mathbb{1} \otimes \text{CNOT}} \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{F}^n} |x\rangle \left((-1)^{f(x)} \beta_{\text{succ}}^x |w_0(x)\rangle |f(x)\rangle + (-1)^{\overline{f(x)}} \beta_{\text{fail}}^x |w_1(x)\rangle |\overline{f(x)}\rangle \right) |-\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{F}^n} (-1)^{f(x)} |x\rangle \left(\beta_{\text{succ}}^x |w_0(x)\rangle |f(x)\rangle - \beta_{\text{fail}}^x |w_1(x)\rangle |\overline{f(x)}\rangle \right) |-\rangle, \end{aligned} \quad (26)$$

where $N = p^n$. In order to establish that the output is $y \in \mathbb{F}^n$ with probability p_y such that $|p_y - \widehat{f}(y)| < 4\varepsilon$ on measuring the first register after executing the entire circuit, we will compute the inner product

$$\beta_y := \left\langle y, 0, 0, - \left| C \right| 0, 0, 0, 0 \right\rangle, \quad (27)$$

since $p_y = |\beta_y|^2$. Notice that we also have

$$\begin{aligned} |y, 0, 0, -\rangle &\xrightarrow{\text{QFT}_p^{\otimes n \otimes 1 \otimes 1 \otimes 1}} \frac{1}{\sqrt{N}} \sum_{v \in \mathbb{F}^n} \omega^{y \cdot v} |v, 0, 0, -\rangle \\ &\xrightarrow{U_f \otimes 1} \frac{1}{\sqrt{N}} \sum_{v \in \mathbb{F}^n} \omega^{y \cdot v} |v\rangle \left(\beta_{\text{succ}}^v |w_0(v)\rangle |f(v)\rangle + \beta_{\text{fail}}^v |w_1(v)\rangle |\overline{f(v)}\rangle \right) |-\rangle, \end{aligned} \quad (28)$$

and we can compute the amplitude β_y in Eq. (27) by taking the inner product between the states on the last lines of Eqs. (26) and (28). Since $f(x) \in \mathbb{F}_2$ we have that $\langle f(x) | \overline{f(x)} \rangle = 0$. Similarly, $\langle v | x \rangle = \delta_{vx}$, and while the states of the workspace register may not be orthogonal, they are normalised. Hence we have

$$\beta_y = \frac{1}{N} \sum_{v \in \mathbb{F}^n} \omega^{y \cdot v} (-1)^{f(v)} \left(|\beta_{\text{succ}}^v|^2 - |\beta_{\text{fail}}^v|^2 \right). \quad (29)$$

Since for every $v \in \mathbb{F}^n$, $|\beta_{\text{succ}}^v|^2 \geq 1 - \varepsilon$ and $|\beta_{\text{succ}}^v|^2 + |\beta_{\text{fail}}^v|^2 = 1$, it always holds that $|\beta_{\text{succ}}^v|^2 - |\beta_{\text{fail}}^v|^2 \geq 1 - 2\varepsilon$. Unlike the Boolean case (i.e. $\mathbb{F} = \mathbb{F}_2$), the Fourier coefficients of f are no longer real numbers, and so need some additional care. Nevertheless, using their definition from Eq. (2), and the Cauchy-Schwarz inequality, we have

$$\begin{aligned} \left| \widehat{f}(y) - \beta_y \right| &= \frac{1}{N} \left| \sum_{v \in \mathbb{F}^n} \omega^{y \cdot v} (-1)^{f(v)} \left(1 - \left(|\beta_{\text{succ}}^v|^2 - |\beta_{\text{fail}}^v|^2 \right) \right) \right| \\ &\leq \frac{1}{N} \left| \sum_{v \in \mathbb{F}^n} \omega^{2y \cdot v} (-1)^{2f(v)} \right|^{1/2} \left| \sum_{v \in \mathbb{F}^n} \left(1 - \left(|\beta_{\text{succ}}^v|^2 - |\beta_{\text{fail}}^v|^2 \right) \right)^2 \right|^{1/2} \\ &\leq \frac{1}{N} \cdot \sqrt{N} \cdot 2\varepsilon \sqrt{N} \\ &\leq 2\varepsilon. \end{aligned} \quad (30)$$

As $|\widehat{f}(y)|, |\beta_y| \leq 1$, and since $\left| |x|^2 - |y|^2 \right| \leq |x^* + y^*| |x - y| \leq 2|x - y|$ when $0 \leq |x|, |y| \leq 1$ and x^* denotes complex conjugation, we finally have that

$$\left| |\widehat{f}(y)|^2 - p_y \right| \leq 4\varepsilon, \quad (31)$$

showing that the circuit C_{GL} can sample approximately from the Fourier spectrum of f . \square

The quantum Fourier transform QFT_p over finite fields \mathbb{F}_p can be implemented efficiently with $O(\log^2 |\mathbb{F}|)$ elementary gates and in depth $O(\log |\mathbb{F}|)$ [Bea97; Hoy97; MRR06].

Learning subspace coefficients of 1_X : Our interest is in the indicator function $1_X : \mathbb{F}^n \rightarrow \{0, 1\}$, which takes value 1 on the set of vectors on which ALG succeeds with appreciable probability $|\beta_{\text{succ}}^v|^2 \geq \alpha$. Using the unitary U_q of Section 5.2 almost satisfies the condition required for Lemma 5.5: $|\beta_{\text{succ}}^v|^2 \geq 1 - \varepsilon$ and $|\beta_{\text{fail}}^v|^2 \leq \varepsilon$, where ε can be chosen to be $o(1)$. To see the bound

noted in Remark 5.6, suppose the densities of the sets X_t^g, X_t^b and W_t are ρ_g, ρ_b and ρ_W respectively and let $\pi_v := \left(1 - \left(|\beta_{\text{succ}}^v|^2 - |\beta_{\text{fail}}^v|^2\right)\right)^2$. Then in Eq. (30) we have

$$\begin{aligned}
|\widehat{f}(y) - \beta_y| &\leq \frac{1}{N} \left| \sum_{v \in \mathbb{F}^n} \omega^{2y \cdot v} (-1)^{2f(v)} \right|^{1/2} \left| \sum_{v \in \mathbb{F}^n} \left(1 - \left(|\beta_{\text{succ}}^v|^2 - |\beta_{\text{fail}}^v|^2\right)\right)^2 \right|^{1/2} \\
&\leq \frac{1}{N} \cdot \sqrt{N} \cdot \left| \sum_{v \in X_t^g} \pi_v + \sum_{v \in X_t^b} \pi_v + \sum_{v \in W_t} \pi_v \right|^{1/2} \\
&\leq \frac{1}{\sqrt{N}} \cdot |2\varepsilon\rho_g N + 2\varepsilon\rho_b N + 2\rho_W N|^{1/2} \\
&\leq 2\varepsilon + 2\rho_W,
\end{aligned} \tag{32}$$

where on the third line we used the definition of U_q to see that the quantity $\pi_v \leq 2\varepsilon$ on both X_t^g and X_t^b , while we only have the guarantee that $\pi_v \leq 2$ on the set W_t , and that $\rho_g + \rho_b + \rho_W = 1$. Hence we have the following corollary.

Corollary 5.7. *Given an average-case quantum algorithm ALG^M as described in Eq. (10) and the oracle U_M for a matrix $M \in \mathbb{F}^{n \times n}$, we can learn heavy Fourier characters χ_y with $\widehat{f}(y) \geq c$ of the indicator function on the set X of inputs on which ALG succeeds with high probability, using ALG $O(1/c)$ times, $q = O\left(\frac{1}{c} \cdot n^{3/2}\right)$ queries to U_M and U_M^\dagger , $O(q \cdot \log n \cdot \text{poly log } |\mathbb{F}|)$ additional one-qubit and two-qubit gates, and $O(n \log |\mathbb{F}|)$ ancillary qubits.*

The procedure is to simply measure the state output by C_{GL} to obtain a character $y \in \mathbb{F}^n$, and then run standard quantum amplitude estimation to estimate the value of the corresponding Fourier coefficient to additive precision $c/100$, which will use C_{GL} a total of $O(1/c)$ times.

6 Robust quantum local correction via additive combinatorics

In this section, we prove our main technical tool: a robust quantum local correction lemma for linear problems. Towards this end, we first prove a noise-robust generalisation of Bogolyubov's lemma from additive combinatorics.

6.1 Robust probabilistic Bogolyubov lemma

Recall that Bogolyubov's lemma states that for any subset $A \subseteq \mathbb{F}_2^n$ of density $|A|/2^n \geq \alpha$, there exists a subspace $V \subseteq 4A$ of dimension at least $n - \alpha^{-2}$.

We would like to use Bogolyubov's lemma to locally correcting faulty inputs by *explicitly* computing a decomposition into a linear combination of *good inputs*, shifted by a sparse vector (see Section 2.2). Computing the sparse shift vector requires learning the (significant) Fourier spectrum of the subspace implied by Bogolyubov's lemma. The caveat is that our quantum algorithm for learning the subspace encoded in a noisy quantum state (see Section 5.4) can only obtain an approximation of the spectrum.

Hence, we need to strengthen Bogolyubov's lemma to obtain the following structural properties:

1. *Robustness*, in the sense that the linear constraints that define the Bogolyubov subspace can lie within a range of Fourier thresholds; and
2. *Density*, in the sense that each element of the Bogolyubov subspace admits many decompositions into valid inputs, and in turn can be sampled probabilistically using our quantum sampling procedure (see Section 5.3).

Next, we prove a generalisation of Bogolyubov's lemma, which achieves the aforementioned structural properties. In the following, given a set $X \subseteq \mathbb{F}^n$ define $\text{Spec}_X(\gamma) = \{r \in \mathbb{F}^n \setminus \{0\} : |\widehat{1}_X(r)| \geq \gamma\}$.

Lemma 6.1 (Robust Bogolyubov lemma). *Let $\mathbb{F} = \mathbb{F}_p$ be a prime field, and let $X \subseteq \mathbb{F}^n$ be a set of size $|X| = \alpha \cdot |\mathbb{F}|^n$ for some $\alpha \in (0, 1]$. Let $R \subseteq \mathbb{F}^n$ be a set such that $\text{Spec}_X(\alpha^{3/2}) \subseteq R \subseteq \text{Spec}_X(\frac{\alpha^{3/2}}{2})$.*

Let $V = \{v \in \mathbb{F}^n : \langle v, r \rangle = 0 \forall r \in R\}$. Then $\dim(V) \geq n - 4/\alpha^2$, and for all $v \in V$ it holds that

$$\Pr_{x_1, x_2, x_3 \in \mathbb{F}^n} [x_1, x_2, x_3, v - x_1 - x_2 - x_3 \in X] \geq \alpha^5,$$

or equivalently

$$\Pr_{x_1, x_2, x_3 \in X} [v - x_1 - x_2 - x_3 \in X] \geq \alpha^2.$$

Proof. Note first that by Parseval's identity we have

$$\alpha = \langle 1_X, 1_X \rangle = \|1_X\|_2^2 = \sum_r |\widehat{1}_X(r)|^2.$$

In particular,

$$\frac{\alpha^3}{4} \cdot |R| \leq \sum_{r \in R} |\widehat{1}_X(r)|^2 \leq \sum_r |\widehat{1}_X(r)|^2 = \alpha;$$

and hence $|R| \leq \frac{4}{\alpha^2}$. In particular, $\dim(V) \geq n - |R| \geq n - 4/\alpha^2$.

Furthermore, we have

$$\sum_{r \in \mathbb{F}^n \setminus (R \cup \{0\})} |\widehat{1}_X(r)|^4 \leq \alpha^3 \cdot \sum_{r \in \mathbb{F}^n \setminus (R \cup \{0\})} |\widehat{1}_X(r)|^2 \leq \alpha^3(\alpha - \alpha^2) \leq \alpha^4 - \alpha^5,$$

where the second inequality uses that $\sum_r |\widehat{1}_X(r)|^2 = \alpha$, and $|\widehat{1}_X(0)|^2 = \alpha^2$. Noting that for every $v \in V$ we have $\chi_r(v) = \omega^{\langle v, r \rangle} = \omega^0 = 1$ for all $r \in R$, it follows that

$$\begin{aligned} \Pr_{x_1, x_2, x_3 \in \mathbb{F}^n} [x_1, x_2, x_3, v - x_1 - x_2 - x_3 \in V] &= (1_X * 1_X * 1_X * 1_X)(v) \\ &= \sum_{r \in \mathbb{F}^n} (\widehat{1}_X(r))^4 \chi_r(v) \\ &= |\widehat{1}_X(0)|^4 \chi_0(v) + \sum_{r \in R} |\widehat{1}_X(r)|^4 \chi_r(v) \\ &\quad + \sum_{r \in \mathbb{F}^n \setminus (R \cup \{0\})} |\widehat{1}_X(r)|^4 \chi_r(v) \\ &\geq \alpha^4 + (\alpha^{3/2}/2)^4 \cdot |R| - (\alpha^4 - \alpha^5) \\ &\geq \alpha^5, \end{aligned}$$

as required. \square

6.2 Quantum local correction lemma

Equipped with the noise-robust Bogolyubov lemma, we can now employ the four quantum procedures we showed in Section 5 and prove the following quantum local correction lemma for linear problems, which lies at the heart of our worst-case to average-case reductions.

Lemma 6.2 (Quantum local correction). *Let $\mathbb{F} = \mathbb{F}_p$ be a prime field, and let $X \subseteq \mathbb{F}^n$ be a set of size $|X| = \alpha \cdot |\mathbb{F}|^n$, for some $\alpha \in (0, 1]$. Then, there exists a non-negative integer $t \leq 4/\alpha^2$, a collection of t vectors $B = \{b_1, \dots, b_t \in \mathbb{F}^n\}$, and t indices $k_1, \dots, k_t \in [n]$ satisfying the following:*

Given a vector $y \in \mathbb{F}^n$, define $s = \sum_{j=1}^t \langle y, b_j \rangle \cdot \vec{e}_{k_j}$ where $(\vec{e}_i)_{i \in [n]}$ is the standard basis. Then

$$\Pr_{x_1, x_2, x_3 \in \mathbb{F}^n} [x_1, x_2, x_3, x_4 \in X] \geq \alpha^5 ,$$

where $x_4 = y - x_1 - x_2 - x_3 - s$. Equivalently, we have

$$\Pr_{x_1, x_2, x_3 \in X} [x_4 \in X] \geq \alpha^2 .$$

Furthermore, suppose we have a quantum membership oracle \tilde{O}_{X^*} (that we can query in superposition) for a set X^* satisfying the following conditions

- $|\hat{1}_{X^*}(r) - \hat{1}_X(r)| \leq \frac{\alpha^{3/2}}{8}$ for all $r \in \mathbb{F}^n$;
- for every input $x \in \mathbb{F}^n$, \tilde{O}_{X^*} computes the indicator $1_{X^*}(x)$ correctly with probability at least $1 - \alpha^{3/2}/10$.

Then, there exists a quantum algorithm that with probability at least $1 - \delta$ returns vectors $b_1, \dots, b_t \in \mathbb{F}^n$ and indices $k_1, \dots, k_t \in [n]$ as described above. This algorithm makes $O(\alpha^{-7/2})$ blackbox queries to \tilde{O}_{X^*} , uses an additional number

$$O\left(n\alpha^{-7/2} \cdot \log n \cdot \text{poly log } |\mathbb{F}|\right)$$

of one-qubit and two-qubit gates, and $O(n\alpha^{-2} \log |\mathbb{F}|)$ ancillary qubits.

Proof. Fix a set $X \subseteq \mathbb{F}^n$ of size $|X| = \alpha \cdot |\mathbb{F}|^n$ for some $\alpha \in (0, 1]$. By applying Lemma 6.1, we obtain a subspace $V \subseteq \mathbb{F}^n$ of dimension $\dim(V) = n - t$ for $t = 4/\alpha^2$. Let $R \subseteq \mathbb{F}_2^n \setminus \{0\}$ be a set of vectors in \mathbb{F}^n of size t such that $V = \{v \in \mathbb{F}_2^n : \langle v, r \rangle = 0 \forall r \in R\}$. Indeed, we can let R be a set of t linearly independent vectors in V^\perp .

By writing the vectors of R in a matrix and diagonalizing the matrix, we obtain: (1) a set of vectors $B = \{b_1, \dots, b_t \in \mathbb{F}_2^n\}$ such that $\text{span}(B) = \text{span}(R)$, and (2) the corresponding pivot indices $k_1, \dots, k_t \in [n]$ such that $b_j[k_j] = 1$ and $b_j[k_{j'}] = 0$ for all $j \neq j'$.

Given a vector $y \in \mathbb{F}^n$, define $s = \sum_{j=1}^t \langle y, b_j \rangle \cdot \vec{e}_{k_j}$, where $(\vec{e}_i)_{i \in [n]}$ is the standard basis, and let $v = y - s$. It is straightforward to verify that $v \in V$. Then for any $j \in [t]$ we have

$$\langle v, b_j \rangle = \langle y, b_j \rangle - \sum_{j=1}^t c_j \cdot \langle \vec{e}_{k_j}, b_j \rangle \stackrel{(*)}{=} \langle y, b_j \rangle - c_j \cdot \langle \vec{e}_{k_j}, b_j \rangle \stackrel{(**)}{=} \langle y, b_j \rangle - \langle y, b_j \rangle = 0 ,$$

where (*) is because $\langle \vec{e}_{k_j}, b_j \rangle = b_j[i_{j'}] = 0$ for $j \neq j'$, and (**) is because $\langle \vec{e}_{k_j}, b_j \rangle = b_j[i_j] = 1$.

Now, since $v \in V$, by the guarantees of Lemma 6.1 it follows that

$$\Pr_{x_1, x_2, x_3 \in \mathbb{F}^n} [x_1 \in X, x_2 \in X, x_3 \in X, v - x_1 - x_2 - x_3 \in X] \geq \alpha^5,$$

which is equivalent to

$$\Pr_{x_1, x_2, x_3 \in X} [v - x_1 - x_2 - x_3 \in X] \geq \alpha^2.$$

For the furthermore part, consider the oracle $\tilde{O}_{X^*}(x)$. By the closeness between the Fourier coefficients of 1_X and 1_{X^*} , we may apply Lemma 6.1. Indeed, letting $R = \text{Spec}_{X^*}(\frac{3}{4}\alpha^{3/2})$ we have a set of Fourier coefficients satisfying the requirements of Lemma 6.1, namely that $\text{Spec}_{X^*}(\alpha^{3/2}) \subseteq R \subseteq \text{Spec}_{X^*}(\alpha^{3/2}/2)$.

By definition, for every input $x \in \mathbb{F}^n$, $\tilde{O}_{X^*}(x)$ computes the indicator $1_{X^*}(x)$ correctly with probability at least $1 - \alpha^{3/2}/10$. Now, we can apply Corollary 5.7 to find a $y \in \mathbb{F}^n$ such that $\hat{1}_{X^*}(y) \geq 3\alpha^{3/2}/4$ with $O(\alpha^{-3/2})$ blackbox queries to \tilde{O}_{X^*} , and $O(n\alpha^{-3/2} \cdot \text{poly log } |\mathbb{F}|)$ additional one-qubit and two-qubit gates. By the closeness between the Fourier coefficients of 1_X and 1_{X^*} , these y 's also satisfy $\hat{1}_X(y) \geq \alpha^{3/2}/2$. Since there are at most $t \leq 4/\alpha^2$ such coefficients, to find all of them with probability $1 - \delta$, we need to repeat the sampling procedure above $O(\alpha^{-2})$ times. Hence, the total query complexity is $O(\alpha^{-7/2})$ blackbox queries to \tilde{O}_{X^*} , the total number of additional gates is

$$O\left(n\alpha^{-7/2} \cdot \log n \cdot \text{poly log } |\mathbb{F}|\right),$$

and the number of ancillary qubits is $O(n \cdot \alpha^{-2} \log |\mathbb{F}|)$. \square

7 Reductions for linear problems

In this section, we prove Lemma 4.1. Namely, we will present an algorithm ALG' whose complexity is as required by the lemma statement. In order to prove the correctness of the algorithm, we will consider $M \in \mathbb{F}^{n \times n}$ such that $\Pr_{v, \text{ALG}}[\text{ALG}^M(v) = Mv] \geq \alpha$, and we will prove for this M and every vector $v \in \mathbb{F}^n$, that $\Pr_{\text{ALG}'}[(\text{ALG}')^M(v) = Mv] \geq 1 - \delta$. To this end we fix a matrix M satisfying the premise:

$$\Pr_{v, \text{ALG}}[\text{ALG}^M(v) = Mv] \geq \alpha. \quad (33)$$

Before describing the proof of Lemma 4.1 we introduce the following notation. For each $v \in \mathbb{F}^n$ let $p_v = \Pr_{\text{ALG}}[\text{ALG}(v) = Mv]$ be the probability that ALG computes correctly the output on input v , where the probability is taken only over the quantum randomness of ALG .

Before proceeding with the proof, we need the following definition of threshold sets.

Definition 7.1. *For an algorithm ALG^M and a matrix M satisfying Eq. (33), we define the set of its good inputs, i.e., the inputs $v \in \mathbb{F}^n$ such that $\text{ALG}^M(v) = M \cdot v$ with a non-negligible probability as follows.*

$$X_\kappa := \{v \in \mathbb{F}^n : p_v \geq \kappa\};$$

7.1 Properties of threshold sets

Below we prove several claims regarding the threshold sets X_κ .

Claim 7.2. *For $\kappa \leq \alpha/2$, the density of X_κ is at least $\alpha/2$, i.e., $|X_\kappa| \geq \frac{\alpha}{2}|\mathbb{F}^n$.*

Proof. By the assumption of the lemma we have $\mathbb{E}_{v \in \mathbb{F}^n} [p_v] \geq \alpha$, and hence

$$\alpha \leq \mathbb{E}_v [p_v] \leq 1 \cdot \Pr_v [p_v \geq \alpha/2] + \alpha/2 \cdot \Pr_v [p_v < \alpha/2] \leq \Pr_v [p_v \geq \kappa] + \alpha/2.$$

Therefore, for all $\kappa \leq \alpha/2$, we have that $\Pr_v [p_v \geq \kappa] \geq \alpha/2$, as required. \square

Next, we prove that if we choose a random $\tau \in [\alpha/4, \alpha/2]$ and $\tau' = \tau - 1/K$ for a sufficiently large K , then, with high probability over the random choices of τ the sets X_τ and $X_{\tau'}$ will have almost the same density.

Claim 7.3. *For a parameter K let $r \in \{1, \dots, K\}$ be chosen uniformly at random. Let $\tau = (1 + r/K) \frac{\alpha}{4} \in [\frac{\alpha}{4} + \frac{\alpha}{4K}, \frac{\alpha}{2}]$, and $\tau' = \tau - \frac{\alpha}{4K}$. Then, $\Pr \left[\frac{|X_{\tau'}|}{|\mathbb{F}^n|} - \frac{|X_\tau|}{|\mathbb{F}^n|} \leq \frac{2}{K} \right] > 1/2$.*

Proof. Note that the interval $[\tau, \tau']$ is sampled by dividing $[\alpha/4, \alpha/2]$ into K intervals, and taking one of them uniformly at random, it follows that $\mathbb{E} \left[\frac{|X_{\tau'}|}{|\mathbb{F}^n|} - \frac{|X_\tau|}{|\mathbb{F}^n|} \right] \leq 1/K$. The claim follows by Markov's inequality. \square

In particular Claim 7.3 implies the following corollary.

Corollary 7.4. *For a parameter K let $r \in \{1, \dots, K\}$ be chosen uniformly at random. Let $\tau = (1 + r/K) \frac{\alpha}{4} \in [\frac{\alpha}{4} + \frac{\alpha}{4K}, \frac{\alpha}{2}]$, and $\tau' = \tau - \frac{\alpha}{4K}$.*

Suppose that $X^ \subseteq \mathbb{F}^n$ is an arbitrary set such that $X_\tau \subseteq X^* \subseteq X_{\tau'}$. Then with probability at least $1/2$ (over the choice of τ) it holds that*

$$\left| \widehat{1}_{X^*}(r) - \widehat{1}_{X_\tau}(r) \right| \leq \frac{2}{K}$$

for all $r \in \mathbb{F}^n$.

Proof. By Claim 7.3 we have $\Pr \left[\frac{|X_{\tau'}|}{|\mathbb{F}^n|} - \frac{|X_\tau|}{|\mathbb{F}^n|} \leq \frac{2}{K} \right] > 1/2$. Suppose this event happens and $\frac{|X_{\tau'}|}{|\mathbb{F}^n|} - \frac{|X_\tau|}{|\mathbb{F}^n|} \leq \frac{2}{K}$. Since $X_\tau \subseteq X^* \subseteq X_{\tau'}$, it follows that $\frac{|X^*|}{|\mathbb{F}^n|} - \frac{|X_\tau|}{|\mathbb{F}^n|} \leq \frac{2}{K}$. We observe that

$$\left| \widehat{1}_{X^*}(r) - \widehat{1}_{X_\tau}(r) \right| \leq \left| \frac{|X_\tau|}{|\mathbb{F}^n|} - \frac{|X^*|}{|\mathbb{F}^n|} \right| \leq \frac{2}{K},$$

which finishes the proof of the corollary. \square

7.2 Proof of Lemma 4.1

In this section, we prove Lemma 4.1, which we restate below for convenience.

Lemma 4.1. *Let $\mathbb{F} = \mathbb{F}_p$ be a prime field, $n \in \mathbb{N}$, and $\alpha := \alpha(n) \in (0, 1]$. Let ALG^M be a quantum query algorithm that has oracle access to a matrix $M \in \mathbb{F}^{n \times n}$, gets as input a vector $v \in \mathbb{F}^n$, makes $Q(n)$ queries, and attempts to compute Mv . Then, for every constant $\delta > 0$, there exists a quantum algorithm $(\text{ALG}')^M$ that makes $O(\alpha^{-7/2})$ uses of ALG^M , $O((Q(n) + n^{3/2}) \cdot \alpha^{-7/2})$ queries to U_M and U_M^\dagger , uses $O(n^{3/2}\alpha^{-7/2} \cdot \log n \cdot \text{poly}(\log |\mathbb{F}|))$ additional one-qubit and two-qubit gates, and $O(\alpha^{-2} \cdot n \log n)$ ancillary qubits such that the following holds.*

For every matrix $M \in \mathbb{F}^{n \times n}$ such that ALG^M computes Mv correctly with probability α :

$$\Pr_{v, \text{ALG}^M} [\text{ALG}^M(v) = Mv] = \mathbb{E}_{v \in \mathbb{F}^n} \left[\|\Pi_{Mv} \text{ALG}^M |v\rangle |0\rangle\|^2 \right] \geq \alpha,$$

the algorithm $(\text{ALG}')^M$ computes Mv correctly for every $v \in \mathbb{F}^n$ with probability $1 - \delta$:

$$\forall v \in \mathbb{F}^n \quad \Pr_{\text{ALG}'} [(\text{ALG}')^M(v) = Mv] = \|\Pi_{Mv} (\text{ALG}')^M |v\rangle |0\rangle\|^2 \geq 1 - \delta.$$

Let ALG^M be the average-case quantum algorithm from the lemma statement that has query access to the matrix M . Note that given the algorithm ALG^M we can define an algorithm $\text{ALG}_{\text{boost}}^M$ that given an input v makes $O(1/\alpha)$ calls to $\text{ALG}^M(v)$ independently, and each time verifies the result using Lemma 5.2. Therefore, for every constant δ , we may assume that there is at least an $\alpha/2$ fraction of inputs on which $\text{ALG}_{\text{boost}}^M$ outputs the correct answer with probability at least $1 - \delta/8$.

We define $(\text{ALG}')^M$ as follows.

Algorithm 1: Reduction for linear problems

Input: ALG^M , $v \in \mathbb{F}^n$, $\alpha \in (0, 1)$

Output: $M \cdot v$

1. Let $K = \frac{4}{\alpha^{3/2}}$, and let $r \in \{1, \dots, K\}$ be chosen uniformly at random.
2. Setting a random threshold: Let $\tau = (1 + r/K) \frac{\alpha}{4} \in [\frac{\alpha}{4}, \frac{\alpha}{2} - \frac{1}{K}]$, and $\tau' = \tau - \frac{1}{K}$.
3. Membership Oracle for X^* : Construct a quantum noisy membership oracle \tilde{O}_{X^*} using Section 5.2 such that with high probability $X_\tau \subseteq X^* \subseteq X_{\tau'}$.
4. Learning X_τ : Having \tilde{O}_{X^*} defined above, apply the quantum local correction lemma (Lemma 6.2) with error parameter $\delta = 1/6$ to compute a collection of $t = O(1/\alpha^2)$ vectors $B = \{b_1, \dots, b_t \in \mathbb{F}^n\}$ and indices k_1, k_2, \dots, k_t . Indeed, by Corollary 7.4, we may apply Lemma 6.2.
5. Efficient sampling from X_τ : Sample x_1, x_2, x_3 from X_τ using Corollary 5.4 with $\kappa = \tau$.
6. Self-correcting v : Using the set B and indices k_i computed in Step 3, let s be the t -sparse vector defined as in Lemma 6.2:

$$s = \sum_{j=1}^t \langle v, b_j \rangle \cdot \vec{e}_{k_j}, \text{ and}$$

$$x_4 = v - x_1 - x_2 - x_3 - s.$$

7. Computing $M \cdot v$: Let $b = \text{ALG}_{\text{boost}}^M(x_1) + \text{ALG}_{\text{boost}}^M(x_2) + \text{ALG}_{\text{boost}}^M(x_3) + \text{ALG}_{\text{boost}}^M(x_4) + M \cdot s$, where $M \cdot s$ is computed by multiplying M by all t non-zero entries of s .
8. Verification: Using Lemma 5.1 with $\varepsilon = O(\alpha^2)$ verify whether $b = M \cdot v$. If the verification accepts, return b .
9. Probability amplification: Repeat the steps 5–8 $O(1/\alpha^2)$ times.

Correctness: By our choice of the parameters and the justifications in the algorithm, we see that with constant probability we simultaneously have that (i) the quantum oracle \tilde{O}_{X^*} from Lemma 5.2 for every $x \in \mathbb{F}^n$, outputs $\tilde{O}_{X^*}(x) = 1_{X^*}(x)$ in Step 3; (ii) The set B and indices k_i are computed correctly by Lemma 6.2 in Step 4; (iii) Step 5 produces uniformly random samples $x_1, x_2, x_3 \in X_\tau$.

Assuming successful executions of the previous steps, by Lemma 6.2 for x_4 computed in Step 6 we have that

$$\Pr_{x_1, x_2, x_3 \in X_\tau} [x_4 \in X_\tau] \geq \alpha^2.$$

Observing that for $x_1, x_2, x_3, x_4 \in X^*$ it holds that $\Pr[\text{ALG}_{\text{boost}}^M(x_i) = Mx_i \ \forall i = 1, 2, 3, 4] \geq 1 - \delta/2$. Therefore, with probability $\Omega(\alpha^2)$, the result computed in Step 7 is the correct output $b = M \cdot v$.

Given the representation of X_τ from Step 4 we repeat steps 5–8 $O(1/\alpha^2)$ times to amplify the probability of success to $2/3$ for every input vector v .

Query complexity: First, we bound the number of queries made by our algorithm in each iteration (Steps 3–8). This amounts to bounding the number of queries required to learn the heavy Fourier coefficients of 1_{X^*} in Step 4 (using the oracle from Step 3), queries in Step 5 to sample from X_M , queries in the four executions of $\text{ALG}_{\text{boost}}^M$ in Step 7 (each corresponding to $O(1/\alpha)$ calls to ALG^M and $O(1/\alpha)$ times verifying the computation), queries required to compute $M \cdot s$ in Step 7, and queries required for verification in Step 8.

By Lemma 6.2, in Step 4 we need at most $O(\alpha^{-7/2})$ queries to ALG^M , and $O(n^{3/2} \cdot \alpha^{-7/2})$ queries to U_M and U_M^\dagger to find the set B and indices k_1, k_2, \dots, k_t . Note that the algorithm does not repeat Step 4 $O(\alpha^{-2})$ times.

In step 5, by Corollary 5.4, we make $O(1)$ uses of the noisy unitary implementation of the approximate indicator function in Section 5.2, which translates to $O(n^{3/2} \cdot \alpha^{-3/2})$ queries to U_M and U_M^\dagger and $O(\alpha^{-3/2})$ queries to ALG^M . In Step 7, each application of $\text{ALG}_{\text{boost}}^M$ uses $O(1/\alpha)$ calls to ALG^M and $O(1/\alpha)$ verification steps. In addition s is t -sparse for $t = 4/\alpha^2$, we compute $M \cdot s$ by making at most $O(n/\alpha^2)$ queries. Step 8 makes $O(n^{3/2})$ queries to U_M and U_M^\dagger . Repeating steps 5–8 for $O(1/\alpha^2)$ times leads to the bound of $O(n^{3/2} \cdot \alpha^{-7/2})$ on the query complexity of the constructed algorithm.

Efficiency: Now we count the additional gates used by our algorithm. The membership oracle from Lemma 5.2 in Step 3, Fourier sampling circuit from Lemma 6.2 in Step 4, and efficient sampling circuit from Corollary 5.4 in Step 5 all use $O(n^{3/2} \log n \cdot \alpha^{-3/2} \cdot \text{poly}(\log |\mathbb{F}|))$ additional gates. Additionally, we need $O(n)$ gates to represent vectors s and v_i for $i \in \{1, 2, 3, 4\}$. Thus, in total our algorithm uses $O(n^{3/2} \log n \cdot \alpha^{-7/2} \cdot \text{poly}(\log |\mathbb{F}|))$ additional one-qubit and two-qubit gates, and use $O(\alpha^{-2} \cdot n \log n)$ ancillary qubits.

Large fields: For the case where the field size is large (say, $|\mathbb{F}| \geq 10/\alpha$), there is a simpler worst case to average case reduction. Note that since we can efficiently verify a candidate solution using Lemma 5.1, we can sample a line ℓ passing through the input vector v and a uniformly random point $x \in \mathbb{F}^n$. It is easy to show that with probability at least $\Omega(\alpha)$, at least an $\Omega(\alpha)$ -fraction of the points on ℓ are in the good set X , i.e., the set of points on which ALG^M outputs the correct answer with probability at least $\Omega(\alpha)$. Thus, with probability $\Omega(\alpha^3)$, we sample two random points a, b on ℓ that both belong to X , and then we can interpolate the value of $M \cdot v$ from $\text{ALG}^M(a)$ and $\text{ALG}^M(b)$. We note that this interpolation technique inherently requires large fields, while the proof presented above works over all finite fields.

Making ALG' a unitary: Finally, we note that it is straightforward to turn the algorithm ALG' , which is produced by our reduction, into a unitary quantum algorithm. Yet, it is important to do this carefully so as to keep the overheads in gate complexity of the resulting unitary within the desired bound of $O(n^{3/2})$. We sketch how to do this below.

Suppose we have performed Step 4 and obtained a classical description of the set of vectors B and the corresponding indices k_1, \dots, k_t . We now need to perform Steps 5–9 in a unitary manner. In lieu of Step 5, we maintain three copies of the output state of the unitary Q_{samp} of Corollary 5.4 in three quantum registers, one each for the vectors x_1, x_2 and x_3 . Step 6 only involves arithmetic operations over \mathbb{F} , and we perform the required computation by a unitary circuit that computes the sparse shift vector in the form of a state $|s\rangle$, using B and k_1, \dots, k_t . In an ancillary register,

we then compute a superposition of vectors corresponding to $v - x_1 - x_2 - x_3 - s$, using the states representing the uniform samples x_1, x_2 and x_3 .

By using amplitude amplification, we can construct a unitary implementation corresponding to ALG_{boost}^M with $O(\frac{1}{\sqrt{\alpha}} \log \frac{1}{\delta})$ uses of $\text{ALG}_{verified}$. Next, we apply this unitary to each of the four registers corresponding to x_1, \dots, x_4 , and also compute $M \cdot s$ using the circuit described in Section 5.1. Finally, we perform more arithmetic to compute a state corresponding to the vector b of Step 7 in an ancillary register. Applying Q_{verify} from Lemma 5.1 to the registers containing v and b , we obtain a version of ALG' which attaches a flag indicating success to its output register. This can then be boosted using $O(1/\alpha)$ rounds of fixed-point amplitude amplification to obtain the desired unitary quantum algorithm.

References

- [AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009 (p. 1).
- [AC02] Mark Adcock and Richard Cleve. A quantum Goldreich-Levin theorem with cryptographic applications. In *STACS 2002*, pages 323–334. Springer, 2002 (pp. 11, 27).
- [AGG⁺22] Vahid R. Asadi, Alexander Golovnev, Tom Gur, and Igor Shinkar. Worst-case to average-case reductions via additive combinatorics. In *STOC 2022*, pages 1566–1574. ACM, 2022 (pp. 3–7).
- [Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *STOC 1996*, pages 99–108. ACM, 1996 (p. 3).
- [BBB19] Enric Boix-Adserà, Matthew Brennan, and Guy Bresler. The average-case complexity of counting cliques in Erdős-Rényi hypergraphs. In *FOCS 2019*, pages 1256–1280. IEEE, 2019 (p. 3).
- [BBF03] Stephane Beauregard, Gilles Brassard, and Jose M. Fernandez. Quantum arithmetic on Galois fields, 2003 (p. 20).
- [BCG⁺92] Shai Ben-David, Benny Chor, Oded Goldreich, and Michel Luby. On the theory of average case complexity. *Journal of Computer and System Sciences*, 44(2):193–219, 1992 (p. 3).
- [Bea97] Robert Beals. Quantum computation of fourier transforms over symmetric groups. In *STOC 1997*, pages 48–53. ACM, 1997 (p. 29).
- [BFN⁺93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4):307–318, 1993 (p. 3).
- [BLR90] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *STOC 1990*, pages 73–83. ACM, 1990 (p. 1).
- [BRS⁺17] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Average-case fine-grained hardness. In *STOC 2017*, pages 483–496. ACM, 2017 (p. 3).
- [BRS⁺18] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Proofs of work from worst-case assumptions. In *CRYPTO 2018*, pages 789–819. Springer, 2018 (p. 3).

- [BŠ06] Harry Buhrman and Robert Špalek. Quantum verification of matrix products. In *SODA 2006*, pages 880–889. SIAM, 2006 (pp. 2, 19).
- [BT06] Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Foundations and Trends in Theoretical Computer Science*, 2(1):1–106, 2006 (p. 3).
- [CKK12] Andrew M. Childs, Shelby Kimmel, and Robin Kothari. The quantum query complexity of read-many formulas. In *ESA 2012*, pages 337–348. Springer, 2012 (p. 19).
- [DLV20] Mina Dalirrooyfard, Andrea Lincoln, and Virginia Vassilevska Williams. New techniques for proving fine-grained average-case hardness. In *FOCS 2020*, pages 774–785. IEEE, 2020 (p. 3).
- [FF93] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, 1993 (p. 3).
- [Gen10] Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In *CRYPTO 2010*, pages 116–137. Springer, 2010 (p. 3).
- [Gid18] Craig Gidney. Halving the cost of quantum addition. *Quantum*, 2:74, June 2018 (p. 21).
- [Gol08] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008 (p. 1).
- [Gol11] Oded Goldreich. Notes on Levin’s theory of average-case complexity. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 233–247. Springer, 2011 (p. 3).
- [GR18] Oded Goldreich and Guy Rothblum. Counting t -cliques: worst-case to average-case reductions and direct interactive proof systems. In *FOCS 2018*, pages 77–88. IEEE, 2018 (p. 3).
- [Gro05] Lov K. Grover. Fixed-point quantum search. *Physical Review Letters*, 95(15):1–4, 2005 (pp. 22, 42).
- [GSL⁺19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *STOC 2019*, pages 193–204. ACM, 2019 (pp. 22, 42, 43).
- [Gue19] Gian Giacomo Guerreschi. Repeat-until-success circuits with fixed-point oblivious amplitude amplification. *Physical Review A*, 99:022306 1–022306 10, 2, February 2019 (p. 22).
- [Haa19] Jeongwan Haah. Product decomposition of periodic functions in quantum signal processing. *Quantum*, 3:190, October 2019 (p. 43).
- [HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv: 0811.3171 (p. 4).
- [Hoy97] Peter Hoyer. Efficient quantum transforms, 1997 (p. 29).
- [IL90] Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *FOCS 1990*. IEEE, 1990 (p. 3).
- [Imp11] Russell Impagliazzo. Relativized separations of worst-case and average-case complexities for NP. In *CCC 2011*, pages 104–114. IEEE, 2011 (p. 3).

- [Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *CCC 1995*, pages 134–147. IEEE, 1995 (p. 3).
- [JKM] Stacey Jeffery, Robin Kothari, and Frederic Magniez. *Nested quantum walks with quantum data structures*. In *SODA 2013*. SIAM, pages 1474–1485 (p. 19).
- [Kot14] Robin Kothari. *Efficient algorithms in quantum query complexity*. PhD thesis, 2014, pages 1–137 (pp. 2, 19).
- [LdW21] Noah Linden and Ronald de Wolf. Average-Case Verification of the Quantum Fourier Transform Enables Worst-Case Phase Estimation. *arXiv:2109.10215*, 2021 (p. 3).
- [Le 12] François Le Gall. Improved output-sensitive quantum algorithms for boolean matrix multiplication. In *SODA 2012*, pages 1464–1476. SIAM, 2012 (p. 19).
- [Lev86] Leonid A. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986 (p. 3).
- [Lip91] Richard Lipton. New directions in testing. *Distributed computing and cryptography*, 2:191–202, 1991 (p. 3).
- [LLV19] Rio LaVigne, Andrea Lincoln, and Virginia Vassilevska Williams. Public-key cryptography in the fine-grained setting. In *CRYPTO 2019*, pages 605–635. Springer, 2019 (p. 3).
- [LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM*, 60(6):1–35, 2013 (p. 3).
- [LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015 (p. 3).
- [MNR⁺07] Frederic Magniez, Ashwin Nayak, Jeremie Roland, and Miklos Santha. Search via quantum walk. In *STOC 2007*, pages 575–584. ACM, 2007 (p. 19).
- [MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *FOCS 2004*, pages 372–381. IEEE, 2004 (p. 3).
- [MRR06] Christopher Moore, Daniel Rockmore, and Alexander Russell. Generic quantum fourier transforms. *ACM Transactions on Algorithms*, 2(4):707–723, October 2006 (p. 29).
- [MSS07] Frédéric Magniez, Miklos Santha, and Mario Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007 (p. 19).
- [NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2010, page 676 (p. 14).
- [Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):1–40, 2009 (p. 3).
- [Sho94] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *FOCS 1994*, pages 124–134. IEEE, 1994 (p. 3).
- [STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001 (p. 3).
- [Vas18] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *ICM 2018*. World Scientific, 2018 (p. 3).

- [WW10] Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *FOCS 2010*, pages 645–654. IEEE, 2010 (p. 19).
- [YLC14] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. Fixed-point quantum search with an optimal number of queries. *Physical Review Letters*, 113:210501, 21, November 2014 (p. 22).

A Quantum singular value transformation techniques

In this appendix, we provide self-contained statements of techniques that make use of the quantum singular value transformations literature.

A.1 Fixed-point amplitude amplification

An issue that we frequently face while using standard amplitude amplification is the inability to *a priori* estimate the number of iterations to perform without overshooting the target state. When the goal is to output a classical solution to a search problem, one can sidestep this issue by iterating over the number of rounds of amplification with a multiplicative step size (often called exponential search). But if the goal of amplification is to obtain a unitary subroutine that can later be composed with another unitary, this problem can be addressed by the technique of fixed-point amplitude amplification, which generalises the notion of fixed-point search [Gro05] and converges monotonically towards the marked state as the number of iterations is increased. We use this machinery primarily in the proof of Lemma 5.1.

The precise version of fixed-point amplitude amplification that we need is expressed in the framework of quantum singular value transformations [GSL⁺19, Theorem 27, arXiv version], as follows.

Theorem A.1 (Fixed-point amplitude amplification). *Let U be a unitary acting on $k+1$ qubits and Π an orthogonal projector. If for an input state $|\phi_{\text{in}}\rangle$ and $p > \delta > 0$ we have $\Pi U |\phi_{\text{in}}\rangle = p |\phi_{\text{tar}}\rangle$, then for every $\varepsilon > 0$ there is a unitary U_q such that $\| |\phi_{\text{tar}}\rangle - U_q |\phi_{\text{in}}\rangle \| \leq \varepsilon$. This U_q requires $q = O(\frac{1}{\delta} \log \frac{1}{\varepsilon})$ uses of U and U^\dagger , one ancillary qubit, and $O(q)$ additional one- and two-qubit gates.*

A.2 Singular value threshold projections

To obtain our unitary implementation of the (noisy) indicator function in Section 5.2, we need two conditions to be simultaneously satisfied: (1) the singular values that are larger than a threshold t are boosted close to unity, *and* (2) the singular values that are below the threshold are suppressed close to zero. In particular, fixed-point amplitude amplification only guarantees the former condition and is not the right tool for the task. We hence invoke a more sophisticated tool, namely, quantum singular value threshold projection. We quote the following result that we use [GSL⁺19, Theorem 31, arXiv version].

Theorem A.2 (Singular value threshold projections). *Let U be a unitary acting on $k+1$ qubits and $\Pi, \tilde{\Pi}$ be orthogonal projectors. Suppose*

$$\tilde{\Pi} U \Pi = \sum_{j=1}^m \zeta_j |w_j\rangle\langle v_j|$$

for $\zeta_j \in (0, 1)$ and $\{w_j\}$ and $\{v_j\}$ two sets of orthonormal vectors. Then for any threshold $t \in (0, 1)$ and $\varepsilon, \delta > 0$, there exists a unitary U_q that makes $q = O(\frac{1}{\delta} \log \frac{1}{\varepsilon})$ queries to U and U^\dagger , and uses

$O(q)$ additional one- and two-qubit gates, such that

$$\tilde{\Pi}U_q\Pi = \sum_{j=1}^m \zeta'_j |w_j\rangle\langle v_j|, \quad \text{s.t. } \forall j \in [m], \quad \begin{cases} \zeta'_j \geq 1 - \varepsilon & \zeta_j \in [t + \delta/2, 1] \\ |\zeta'_j| \leq 1 & \text{if } \zeta_j \in (t - \delta/2, t + \delta/2) \\ \zeta'_j \leq \varepsilon & \zeta_j \in [0, t - \delta/2] \end{cases} \quad (34)$$

Computing the circuits for U_q . The two theorems stated above are in fact constructive and provide explicit circuits for the unitaries U_q . In the interest of space, we simply note here that there is a classical runtime overhead of $O(q^3 \text{poly} \log(q/\varepsilon))$, and refer the interested reader to [GSL⁺19; Haa19] for details. For our applications, this overhead is of order $\tilde{O}(n^{3/2})$.