Gravitational-wave matched filtering on a quantum computer

Doğa Veske, Cenk Tüysüz, Mirko Amico, Nicholas T Bronn, Olivia T Lanes, Imre Bartos, Zsuzsa Márka, Sebastian Will and Szabolcs Márka

1 Department of Physics, Columbia University in the City of New York, New York, NY 10027, United States of America
2 Institute for Theoretical Physics, Heidelberg University, Heidelberg 69120, Germany
3 Deutsches Elektronen-Synchrotron DESY, Platanenallee 6, 15738, Zeuthen, Germany
4 Institut für Physik, Humboldt-Universität zu Berlin, Newtonstr. 15, 12489, Berlin, Germany
5 IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights, NY, United States of America
6 Department of Physics, University of Florida, PO Box 118440, Gainesville, FL 32611-8440, United States of America
7 Columbia Astrophysics Laboratory, Columbia University in the City of New York, New York, NY 10027, United States of America

E-mail: veske@thphys.uni-heidelberg.de

Keywords: quantum computing, matched filtering, gravitational wave detection

Abstract
State-of-the-art quantum computers have very limited applicability for accurate calculations. Here, we report the first experimental demonstration of qubit-based matched filtering for a detection of the gravitational-wave signal from a binary black hole merger. With our implementation on noisy superconducting qubits, we obtained a similar signal-to-noise ratio for the binary black hole merger as achievable with classical computation, providing evidence for the utility of qubits for practically relevant tasks. The algorithm we invented for this application is a Monte Carlo algorithm which uses quantum and classical computation together. It provides a quasi-quadratic speed-up for time-domain convolution, similar to achievable with fast Fourier transform.

1. Introduction
Recognizing patterns in large, noisy data sets is a key computational challenge with broad applications such as in signal detection, image analysis and machine learning. With the increasing amount of collected data, improvements on the execution of such analyses will improve our capability to process it. Such improvements may come in many forms from faster calculation algorithms to employing more energy efficient hardware.

Knowing the precise shape of the signal one searches for in a dataset increases the statistical power of the search compared to a blind search for an anomaly. The optimal search technique for finding a known signal buried in Gaussian noise is called matched filtering [1]. Originally developed for detecting radar echoes in the 1950’s [2], its approximate optimality in many real world applications made it the method of choice in many different areas from gravitational-wave (GW) detection [3, 4] to quantum tomography via correlation of noisy measurements [5].

For a known digitized template signal $x_i$ with $N$ points and a data stream $y_i$ with $L$ points ($N \leq L$) in the presence of white noise, in time domain, the signal-to-noise ratio (SNR) of a matched filter can be computed with the convolution

$$\rho[f] = \sum_{i=1}^{N} y_{i+j} x_i$$

(1)

In the presence of colored noise, $x$ and $y$ additionally need to be normalized using the noise’s power spectral density.

Calculations of the form of equation (1) have time-complexity of $O(NL)$ if done in time domain. It can be calculated more efficiently in frequency domain using fast Fourier transform (FFT) with $O(L \log N)$ time complexity. Here we do this calculation with a similar efficiency without using FFT and with a non-conventional hardware, a quantum computer. We developed a hybrid algorithm that uses classical and quantum computers together to calculate the sums of the form equation (1) quasi-quadratically faster than a classical computer for
real valued functions; its time complexity scales as $O(L \log N)^2$. This speed-up in our algorithm is not related
to any mathematical symmetry of complex exponentials, which are the basis for FFT. The improvement in our
algorithm comes from the utilization of quantum operations. With this algorithm, we performed the first qubit-
encoded matched filtering with a quantum computer. Currently there is no clear advantage of using this hybrid
algorithm over using FFT as they both have similar time complexities. However, it is hard to predict the future
advantages that may come from the use of a conceptually different hardware and algorithm. As a speculation,
sharing the computational load to classical and quantum computers may be more energy efficient in the future.

The algorithm presented here is a Monte Carlo approach to solving the problem. We leverage the divide-
and-conquer algorithm [6] to encode a real valued vector on the quantum computer. The only quantum
operations we perform in our method are encoding of the problem into a string of qubits and performing
quantum measurements on them. These operations, without additional quantum computation can, by
themselves, provide a quantum advantage. Quantum encoding and measurement can generate random
bitstrings according to a probability distribution exponentially faster than classical generation by measuring the
end states of amplitude encoding. We basically exploit this attribute. The divide-and-conquer algorithm [6]
requires poly-log depth circuits and since we don’t perform any other operation, this use of low-depth quantum
circuits makes our method more robust against the noise of the quantum hardware without any error
 correction. This is the main reason why we were able to demonstrate our method on real quantum hardware
with high accuracy.

Our experimental implementation we present in this letter, provide another evidence for the use of quantum
computers on practically useful tasks, which are not specifically crafted for quantum computers. We executed
our experiment on a GW signal from a binary black hole merger. Template based GW search algorithms are
searches employing matched filtering using millions of templates [7, 8], where known waveforms of merging
binary systems of black holes or neutron stars, which are computed via Einstein’s field equations, are used [3, 4].
By the end of the third observing run of the LIGO [9] and Virgo [10] detectors, about 90 astrophysical
gravitational-wave (GW) discoveries have been reported [11–14]. Besides the detectors’ unprecedented
sensitivity, observations succeeded thanks to the deployed state-of-the-art search techniques. This
computational load in GW astronomy, has attracted help from quantum computation too; search and analysis
algorithms have been developed which use quantum computation [15–17].

Below, we first briefly explain the basic principles of our method. Then, we explain our implementation and
show our results.

2. Method

First, the signal template and data are encoded to independent sets of qubits via amplitude encoding which can
encode $n$ complex numbers to the amplitudes of the $2n$ different computational basis states spanned by $N_q$
qubits. Despite the logarithmic gain in the number of qubits, the computational complexity of amplitude
encoding scales with $O(n)$ which at first sight seems to negate any gain. However, it has been shown that with the
use of extra qubits (typically called ancilla or auxiliary qubits) during the encoding stage, which are thrown after
the encoding is done, the time complexity of amplitude encoding can be reduced [18]. Specifically [6] discusses a
divide and conquer algorithm which uses $O(n)$ qubits to encode $\log_2 n$ qubits with a circuit depth $O(\log n^2)$.
This improvement enables advantageous uses of qubits for computation without losing the advantage in the
beginning. We use this encoding algorithm in our method. In appendix A we briefly explain how this algorithm
works.

We encode the data and signal such that the coefficients of the computational basis states of the qubit states
are proportional to the square root of the corresponding values of the signal and data, which have been offset by
the absolute value of their minimum negative values to make them all positive. The overall encoded state
$|\psi\rangle = |x\rangle |y\rangle$ is formed by

\[
\text{Data} = |y\rangle = N_y^{-1} \sum_{i=1}^{L} \sqrt{\Delta y + \Delta y^i} |i\rangle_y, \quad (2a)
\]

\[
\text{Signal} = |x\rangle = N_x^{-1} \sum_{i=1}^{N} \sqrt{\Delta x + \Delta x^i} |i\rangle_x \quad (2b)
\]

where $|i\rangle$ are the $2^n$ possible basis states obtainable with $n$ qubits, i.e., $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ for 2 qubits;
$\Delta x = -\min(x_i)$ and $\Delta y = -\min(y_i)$, with normalizations $N_x = \sum_{i=1}^{N} (x_i + \Delta x)$ and $N_y = \sum_{i=1}^{L} (y_i + \Delta y)$.

In the next step, joint measurements of the encoded independent sets of data and signal qubits generate
bitstrings whose appearance probabilities are proportional to the product of the offset signal and data values for
the times (indices) each bitstring uniquely represent. After the measurement, a classical logic circuit relocates
each bitstring to their correct place (index) in the output, according to equation (1), and stacks every occurrence.
Averaging over series of operations of the circuit samples possible occurrences and constructs the desired discrete output signal. Due to the offset of data and signal template, and the normalizations a correction is done at the very end. These operations correspond to calculating the SNR as

\[
\rho[j] = N\sum_{i=1}^{N} |\langle i | j + 1 | i \rangle|^2 - \sum_{i=1}^{N} \Delta_y x_i + \Delta_y x_{i+1} + \Delta_y \Delta x
\]

Among the time complexities of each of these classical logical, mathematical and quantum operations; the dominant one is of the amplitude encoding which is \(O((\log L)^2)\) with the divide and conquer algorithm [6]. Moreover, in order to have a fixed precision of each SNR value, one needs to have certain number of shots per calculated SNR. This necessitates the total shot count to scale with the number of SNRs to be calculated, i.e. \(O(L)\). Therefore, the overall time complexity of this algorithm is \(O(L \log L)^2\). Due to worse scaling than linear scaling, an additional optimization can be done by dividing the data into smaller segments which can reduce the overall time complexity to \(O(L \log N)^2\). The precision of each SNR estimate is

\[
\delta \rho[j] = \frac{\sum_{i=1}^{N} |\rho[i]|}{\rho[j] \times s \times L} \times \left(1 + \frac{\sum_{i=1}^{N} \Delta y x_i + \Delta y x_{i+1} + \Delta y \Delta x}{\rho[j]}\right)
\]

where \(s\) is the total shot count divided by \(L\). More details on precision, time complexity of each operation, and data length manipulation for extra optimization are explained in appendix B. The whole algorithm can be implemented on quantum and classical hardware and can be seen as a convolution gate. The complete block diagram of the circuit is shown in figure 1. A similar hybrid construction of the SWAP test [19, 20], which computes the inner product of two vectors, with the purpose of simplifying the original quantum circuit with classical additions exists in [21], which we were inspired from. We also note that if shallower amplitude encoding algorithms are invented in the future, which were theoretically proven to exist [18], our method’s total time complexity can be further reduced down to \(O(L \log N)^2\).

3. Implementation

We experimentally demonstrated the operation of our method by accessing IBM Quantum systems through the cloud via Qiskit [22]. We computed the SNR time series of the astrophysical gravitational-wave event GW190521 [23] for a binary black hole merger gravitational waveform and demonstrated the accuracy of the method by comparing this result with the output of classical computation (see also appendix C for a preliminary study on artificial random data). IBM Quantum backends consist of fixed-frequency superconducting transmons [24] coupled by transmission line ‘bus’ resonators arranged in a heavy-hexagonal lattice [25]. Single-qubit gates are achieved by on-resonant microwave drives at the ground-to-excited transition frequencies of the

---

**Figure 1.** The convolution gate describing the hybrid method and showing the asymptotic computation times of its parts, without optimizing by dividing the data into smaller segments. The shaded region points out the part with quantum operations. The total time complexity which is \(O(L \log L)^2\) in the figure reduces to \(O(L \log N)^2\) after that optimization (see Appendix B).
transmons, which vary due to the amorphous oxide of the single Josephson junction that forms the inductive part of the transmon (large superconducting pads form the geometric shunting capacitor of the analogous anharmonic oscillator). The cross-resonance entangling gate \([26, 27]\) uses these variations in frequency to create a microwave-activated drive that entangles coupled transmons and forms the basis of the CNOT gate. Measurement is done in the dispersive regime of circuit quantum electrodynamics \([28, 29]\), in which a readout resonator is populated by photons far detuned from the qubit frequency, which dephase each qubit and imparts a state-dependent phase shift on the measurement pulse. The available native basis gates on these backends are highly-calibrated \(\pi/2\) rotations \(sx\) gates, virtual-\(Z\) \([30]\) \(rz\) gates for single qubit operations; and CNOT \(cx\) gates for multi-qubit operations. The other multi-qubit operations are executed with a combination of these gates.

For an extended discussion on the architecture of the systems with superconducting qubits we refer to the \([31, 32]\). Sources of error in these operations are available \([33]\) and show the main sources of error to be the CNOT error (average of 1\%) and readout assignment error (1\%–4\%). As deeper quantum circuits tend to have more CNOTs, these tend to be the biggest source of error. Due to limited physical connectivity, SWAP gates (consisting of 3 CNOTs) are particularly costly when mapping algorithms to current noisy quantum hardware.

Due to these limitations, we non-optimally divided the GW data \((y)\) into segments of length \(k_s = 4\) and the template \((x)\) into segments of length \(k_t = 2\). The main obstacle for not using more data points at once is the significantly greater estimated error in the encoding when data is encoded to more than 2 qubits with the divide and conquer algorithm \([6]\), due to the need of controlled SWAP operations between non-connected qubits. The template lengths \((=2)\) were chosen less than the data lengths \((=4)\) in order to have more than one SNR value calculated with the described method. In this case there are 3 SNR values calculated per one data set. They correspond to the measurement probabilities of the template \((x)\) and data \((y)\) qubit probabilities

\[
P(0|0)P(y_0) + P(1|0)P(y_1) + P(0|1)P(y_0) + P(1|1)P(y_1) + P(1|0)P(y_1) + P(1|1)P(y_1).
\]

For our demonstration with the real gravitational-wave data, we have specifically chosen the event GW190521 \([23]\). It consists of relatively low gravitational-wave frequencies compared to the observed duration of the signal as it is one of the heaviest stellar-mass binary black hole systems ever detected by LIGO Scientific Collaboration and Virgo Collaboration \([11-14]\). This allows us to down-sample the detected data series to reduce the data points to be processed with the quantum hardware. We used the 32 s long aLIGO Livingston \([34]\) detector’s data sampled at 4 kHz.\(^8\) We down-sampled it to 200 Hz, after digitally applying an ideal low-pass filter with the cutoff frequency 99.98 Hz which is above the maximum frequency in the reconstructed astrophysical gravitational-wave. The noise power spectral density was found from the original data with Welch’s average periodogram method \([35]\) with segment lengths of 512, using the \texttt{matplotlib} \([36]\) python package’s \texttt{matplotlib.pyplot.psd} function. According to the estimation of the source properties of the gravitational-wave event at the detector frame \([37]\); the template waveform was generated at 200Hz sampling with the \texttt{gwsurrogate} \([38]\) package with the NRSur7dq4 waveforms \([39]\). The used inputs were 154.7 solar masses and 120.1 solar masses for the detector frame black hole masses, \((0.69, 0, 0)\) and \((0, 0.73, 0)\) for the dimensionless spin vectors, and 12 Hz for the start frequency. With these configurations the SNRs were calculated for the data points starting at the UTC time 1 242 442 967.15 for the consequent 0.45 s. The experiments were ran on the backend \texttt{ibmq\_guadalupe} \([33]\). In each run, from the 16 qubits on the backend, up to 10 qubits were used for independently encoding 3 partial data sets with 3 qubits each and a template part with 1 qubit in parallel. In this configuration, total of 308 circuits were ran for covering the full signal, each with \(10^4\) shots. The photo of the Falcon chip, which is the model of \texttt{ibmq\_guadalupe}, and the layout of \texttt{ibmq\_guadalupe} with the used qubits are shown in figure 2. At the time of execution, the CNOT error probabilities of each qubit connection and readout error probabilities of the qubits were all about 1\%. The SNR for a particular time was found by adding the SNRs of the 2-point partial templates, which overlap with the previous and next partial template at 1 point.

### 4. Results

The 308 quantum circuits each with \(10^4\) shots took 14.5 minutes in total to run. The SNR series obtained with the quantum processor along with the pure classical computation are shown in figure 3. For an easier relative comparison of the quantum processor’s performance, they were scaled in the figure such that the classically computed maximum SNR reaches the value of 1; since the overall factors are calculated classically in the hybrid calculation and do not change the relative accuracy. The hybrid calculation shows high accuracy. The main deviation between the two methods is the slight reduction of highest absolute SNR values, which shows the effect of the non-idealities of the current hardware (see appendix C for our similar findings with artificial data). This

\[^8\] https://www.gw-openscience.org/eventapi/html/GWTC-2/GW190521/
shows that current quantum hardware can be used for performing calculations with comparable accuracy to a classical computer for practically relevant problems.

With technological improvements, these calculations can be performed faster with quantum computers. The reasons why quantum advantage could not be achieved at this point are the CNOT error rates and the limited connectivity between the qubits which increases the required number of CNOT gates for encoding. These prohibit the construction of more intricate configurations for encoding larger number of qubits, which would have advantage over classical computation. Executing circuits with large number of qubits remains to be a hardware challenge in the noisy intermediate-scale quantum era [40]. With future hardware improvements, a useful application of this method can be executed.

5. Outlook

We point out that the described method for matched filtering is straightforward to generalize for any calculations involving multi-dimensional functions; e.g., in image processing, while obeying the same scaling,
where \( N \) and \( L \) represent the total number of points in the higher dimensional spaces, since any higher dimensional finite number of samples can be represented in one-dimensional series. A corresponding one dimensional filter can be found according to the rasterizing scheme to one dimension. Moreover, our method can be generalized to time independent system responses or integral transforms having more than one input or more than one instances of the same input as in non-linear systems, Volterra series or coherent matched filtering of multi-detector data. These constructions with more than one input are simply special cases of multi-dimensional functions.

Our method uses quantum and classical computation complementarily instead of relying completely on quantum computation. This makes it more resilient to the typical gate errors due to having shallow quantum circuits and hence makes it a method that can be used reliably in the nearest future compared to the methods that require deeper quantum circuits. The presented demonstration proved the method’s experimental viability on IBM Quantum’s hardware and which can have an advantage with improving technology, especially with fully connected qubit networks. Although performing similarly to the classical computation with FFT, in the future distributing the computing load to quantum and classical computers may have other practical benefits such as energy saving or hardware scalability.

**Acknowledgments**

This document was reviewed by the LIGO Scientific Collaboration under the document number P2200255. Authors are grateful to Mesut Çalışkan for this review, detailed inspection and helpful comments. Authors thank Columbia University in the City of New York for its unwavering and generous support. D.V. thanks Baran Bodur for suggesting the name convolution gate. C.T. is supported in part by the Helmholtz Association - ‘Innopol Project Variational Quantum Computer Simulations (VQCS)’. I.B. acknowledges the support of the Alfred P. Sloan Foundation. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. This research has made use of data or software obtained from the Gravitational Wave Open Science Center (gwosc.org) [41, 42], a service of the LIGO Scientific Collaboration, the Virgo Collaboration, and KAGRA. This material is based upon work supported by NSF’s LIGO Laboratory which is a major facility fully funded by the National Science Foundation, as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain. KAGRA is supported by Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan Society for the Promotion of Science (JSPS) in Japan; National Research Foundation (NRF) and Ministry of Science and ICT (MSIT) in Korea; Academia Sinica (AS) and National Science and Technology Council (NSTC) in Taiwan.

**Data availability statement**

The data that support the findings of this study are available upon reasonable request from the authors. https://github.com/dveske/quantum-computing-matched-filtering.

**Appendix A. The divide and conquer algorithm**

Here, we describe the divide and conquer data loading algorithm proposed by Araujo et al [6]. The algorithm encodes an \( n \)-dimensional input vector using a tree architecture. The algorithm requires \( \mathcal{O}(n) \) classical resources and \( \mathcal{O}(\log^2(n)) \) quantum resources to amplitude encode an \( n \)-dimensional real vector.

The algorithm is based on a bottom-up approach, where qubits are combined in groups at each step. Therefore, the first step of the algorithm requires single qubit states. These states are generated by single qubit \( R_Y \) rotations. The computation of the angles of these gates are the source of \( \mathcal{O}(n) \) classical resources requirement. Then, a subroutine is required to combine these states. The first level of this step will create two qubit pairs, the second level will combine these to create groups of three qubit states. Therefore, this step requires \( \log_2(n) \) levels. The subroutine to combine these states will dictate the total depth of the algorithm. Araujo et al [6] uses the following circuit (figure A1) to combine two \( m \) qubit states.
This subroutine is used to combine the states sequentially, such that first level combines one qubit states, \( m \)th level combines \( m \) qubit states and the last level combines \( n \log_2 qubit states to encode the \( n \)-dimensional real vector. Thus, the depth of the algorithm is \( O(n \log_2 n) \) and it uses \( O(n) \) qubits in total including the auxiliary qubits. Please refer to Araujo et al. [6] for more details.

**Appendix B. Time complexity and resource analysis**

Our method is based on encoding data and signal templates, measuring the qubits, doing classical logic operations on the measurements and doing mathematical corrections at the end. For obtaining a fixed precision, excluding the pre and post processing parts, the quantum and classical logic circuits need to run multiple times. This repeat factor is proportional to the length of the data \( \Omega(L) \) since every quantum measurement outcome is distributed to one of the \( L \) possible times. Time complexity of each stage is:

- **Pre-encoding calculations:** Encoding requires calculating the state amplitudes to be encoded. This corresponds to calculating the offsets \( \Delta x \) and \( \Delta y \), taking the square root and calculating the normalizations. The time complexity of these operations are proportional to the length of the data, i.e. \( O(L) \). This needs to be done once.

  Second part of the pre-encoding stage is the design of the encoding circuit. This can be done with time complexity \( O(L) \) for the divide and conquer algorithm we use for encoding [6]. This needs to be done once.

- **Encoding (quantum):** With the divide and conquer algorithm this is done in \( O((\log L)^2) \). This is done \( O(L) \) times.

- **Constructing and adding products (classical):** There are going to be total of \( \log_2 NL \) qubits to be measured and \( NL \) possible different outcomes. SNR of each data point \( j \) is proportional to the number of measurements of mutually exclusive \( N \) of these possible different outcomes. Therefore, the depth of the logic circuits constructed with OR gates with 2 inputs is \( O(\log_2 L) \).

  The expansion of \( \log_2 NL \) qubits to \( NL \) different logic functions can be done with 2 input AND gates and inverters with depth \( O(\log\log NL) \). These two operations in series will be done \( O(L) \) times.

- **Post-measurement calculations:** Among the correction terms in equation (3) only the \( \sum_{j=1}^{N} \Delta x_{j+1} \) term is not a constant term. Although it seems that \( N \) additions need to be made for calculating it for every \( L \) values of \( j \), since the difference between the consecutive values of the correction is just 2 numbers (first number of the previous step and the last number of the new step), asymptotically \( O(N + L) \) operations are needed for calculating every \( L \) value of it.

Overall time complexity is dominated by the encoding part, which is \( O(L (\log L)^2) \)

**Extra optimization:** Since the overall scaling \( O(L (\log L)^2) \) grows faster than linear scaling \( O(L) \), one might suggest to divide the data into smaller segments with length \( k \) and perform several smaller computations in series. In order to have completeness of SNRs, each of these segments must overlap with the previous one at the

---

**Figure A1.** Circuit to combine two \( m \)-qubit states. \( a \) and \( b \) do not depend on the input states \( |\psi\rangle \) and \( |\phi\rangle \). The depth of the subroutine is \( O(m) \) and \( m \) qubits are discarded at each step.
neighboring N data points. Then the computation time becomes $O\left(\frac{k}{k-N}L \log k^2 \right)$. The optimal length $k$ depends on the template length which is given by the equation $N = \frac{2L}{\ln k + 2}$. This optimization scheme decreases the total computation time generally to $O(L \log N^2)$. The precise optimization does not affect the asymptotic scaling. For example, with the choices of $k = 2N$ or $k = 3N$, it becomes $O(2L \log (2N)^2)$ or $O(1.5L \log (3N)^2)$ respectively, both being equivalent to $O(L \log N^2)$.

**Hardware requirements:** Total necessary qubit count is $O(L)$. Total number of needed AND gates and inverters, without any logical simplifications, e.g., with a Karnaugh map [43]; will be $O(NL \log NL)$. Similarly, total needed OR gates will be $O(NL)$.

**Precision of the computation:** Without the corrections in equation (3), according to variance and mean of the binomial distribution, the relative precision of the SNR ($\rho[j]$) estimates would be (standard deviation/mean)

$$\frac{\delta \rho[j]}{\rho[j]} = \sqrt{\frac{\sum_{i=1}^{N} \rho[i]}{\rho[j] \times s \times L}} \approx \sqrt{\frac{\rho}{\rho[j] \times s}} \quad (B1)$$

where $s$ is the total shot count divided by $L$, which is taken as constant in the above time complexity analysis; and $\rho$ is the average SNR. With the corrections, this relative precision becomes the precision of $\text{SNR} + \sum_{i=1}^{N} \Delta y x_i + \Delta x y_{i+1} + \Delta y \Delta x$. Therefore, the relative precision of the SNRs become

$$\frac{\delta \rho[j]}{\rho[j]} = \sqrt{\frac{\sum_{i=1}^{N} \rho[i]}{\rho[j] \times s \times L}} \times \left(1 + \frac{\sum_{i=1}^{N} \Delta y x_i + \Delta x y_{i+1} + \Delta y \Delta x}{\rho[j]}\right) \quad (B2)$$

The offsets worsen the precision, and each data segment with different offsets can have different precision with the same shot count.

**Appendix C. Verification with artificial random data**

Prior to analyzing real data, we tested the accuracy of our method on artificially produced random data. The experiments we show here were ran on the backend _ibmq_lima_. The estimated total CNOT error probability and the total measurement error probabilities were both about 7% which are the main estimated sources of error. We have chosen the 2-point signal template arbitrarily as [2, -1]. Each data set of 4 numbers were chosen randomly from a normal distribution of zero mean and unit variance. In order to have all of the possible 8 measurement outcomes, the signal and data were arbitrarily shifted 0.1 more than their minimum negative values, i.e. $\Delta x = -\min(x_i) + 0.1$, $\Delta y = -\min(y_j) + 0.1$. Here we show results from 100 different data sets for each $2 \times 10^4$ measurements (shots) were made. Figure C1 shows an example circuit from our experiment for the encoding of 3 qubits and their measurements. Qubits q1, q2, and q3 are used for encoding the data into the qubits q4 and q5. The signal template is encoded to q4.

Figure C2 shows two scatter plots, one for the calculated $100 \times 3 = 300$ SNRs with quantum measurements versus the true SNR values and one for the SNR errors between them versus the true SNRs. The correlation coefficient for the points in the left figure was found to be 0.99, which should have been 1 ideally, and in the right figure as −0.57. The anti-correlation between the errors and the true SNR is a clear indication that the errors have arisen due to the noise in the circuit. This is due to the fact that in order to have an SNR with a high magnitude, either the amplitude of the data should be high or more relevantly the relative amplitudes of the consequent data points need to have specific values. Any noise in the system can corrupt such delicate data segments. These corruptions increase the marginal entropies of encoded signal and data incoherently resulting in decrease in their mutual information. Low SNR points on the contrary do not get affected by such corruptions as their already low mutual information cannot decrease more. Another observation that can be made is the affinity to having positive SNR errors. This can be explained by the asymmetry in the encoding due to shifting to the positive values. Without the correction in equation (3), only the positive SNR values can be obtained. Therefore, there is a fundamental lower limit on the SNR errors during the encoding. The effect of this lower limit is seen as having mostly positive SNR errors, especially for data segments which have true negative SNR since their SNR values without the correction are the closest to zero. In figure C3 we show the result of a noise-free ideal quantum simulation with the same data and signal. The error in this case is only due to the Poisson errors due to finite number of measurements. The correlation coefficient between the SNRs is 1.0 and between the errors and the true SNRs is −0.1.
Figure C1. One of the circuits ran in the experiments for encoding.

Figure C2. Results of the experiments on the quantum backend ibmq_lima: Left figure shows a scatter of the 300 SNRs computed with the use of qubits vs the true SNRs. The orange line is \( x = y \) just for reference. Right figure shows the dependency on the errors in the computation on the true SNRs.

Figure C3. Results from an ideal noise free quantum simulation. The orange line is again \( x = y \) just for reference. The only source of error is the Poisson uncertainty due to finite number of shots \( = 2 \times 10^4 \).

ORCID iDs

Doğa Veske  https://orcid.org/0000-0003-4225-0895
Cenk Tüysüz  https://orcid.org/0000-0003-0257-9784
Nicholas T Bronn  https://orcid.org/0000-0002-6731-6814

References

[1] Couch L W 2012 Digital and Analog Communication Systems 8th edn (Pearson) ch 6-8
[2] Woodward P 1953 — Simple theory of radar reception Probability and Information Theory with Applications to Radar (Second Edition) International Series of Monographs on Electronics and Instrumentation 2nd edn, ed P Woodward (Pergamon) 81—99
[3] Allen B, Anderson W G, Brady P R, Brown D A and Creighton J D E 2012 Phys. Rev. D 85 122206
[4] Abbott B P et al (LIGO Scientific Collaboration and Virgo Collaboration) 2016 Phys. Rev. Lett. 116 061102
[5] Ryan C A, Johnson B R, Gambetta J M, Chow J M, da Silva M P, Dial O E and Ohki T A 2015 Phys. Rev. A 91 022118
[6] Araújo J F, Park D K, Petruccione F and da Silva A J 2021 Sci. Rep. 11 6329
[7] Cannon K et al 2020 Getlil: A Software Framework for Gravitational Wave Discovery arXiv:2010.05082
[8] Nitz A H, Capano C D, Kumar S, Wang Y F, Kastha S, Schafer M, Dhurkunde R and Cabero M 2021 Astrophys. J. 922 76
[9] Aasi J et al 2015 Classical Quantum Gravity 32 074001
[10] Acernese F et al 2014 Classical Quantum Gravity 32 024001
[11] Abbott B et al 2019 Phys. Rev. X 9 031040
[12] Abbott D et al 2021 Phys. Rev. X 11 021053
[13] Abbott R (The LIGO Scientific Collaboration, The Virgo Collaboration) et al 2024 Gwtt-2.1: Deep Extended Catalog of Compact Binary Coalescences Observed by Ligo and Virgo During the First Half of the Third Observing Run Phys. Rev. D 109 022001
[14] Abbott R (The LIGO Scientific Collaboration, The Virgo Collaboration, The KAGRA Collaboration) et al 2023 Gwtt-3: Compact Binary Coalescences Observed by Ligo and Virgo During the Second Part of the Third Observing Run Phys. Rev. X 13 041039
[15] Miyamoto K, Morrás G, Yamamoto T S, Kuroyanagi S and Nesseris S 2022 Phys. Rev. Res. 4 033150
[16] Gao S, Hayes F, Croke S, Messenger C and Veitch J 2022 Phys. Rev. Research 4 023206
[17] Escrig G, Campos R, Casares P A M and Martin-Delgado M A 2023 Classical Quantum Gravity 40 045001
[18] Zhang X M, Li T and Yuan X 2022 Quantum State Preparation with Optimal Circuit Depth: Implementations and Applications Phys. Rev. Lett. 129 230504
[19] Buhrman H, Cleve R, Watrous J and de Wolf R 2001 Phys. Rev. Lett. 87 167902
[20] Kang M S, Heo J, Choi S G, Moon S and Han S W 2019 Sci. Rep. 9 6167
[21] Cinco L, Subay Y, Sornborger A T and Coles P J 2018 New J. Phys. 20 113022
[22] Treinish M et al (2022) Qiskit: An Open-source Framework for Quantum Computing https://zenodo.org/records/6403335
[23] Abbott R et al (LIGO Scientific Collaboration and Virgo Collaboration) 2020 Phys. Rev. Lett. 125 101102
[24] Koch J, Yu T M, Gambetta J M, Houck A A, Schuster D I, Major J, Blais A, Devoret M H, Girvin S M and Schoelkopf R J 2007 Phys. Rev. D 76 042319
[25] Chamberland C, Zhu G, Yoder T J, Hertzberg J B and Cross A W 2019 Phys. Rev. X 10 (1) 011022
[26] Chow J M et al 2011 Phys. Rev. Lett. 107 080502
[27] Sheldon S, Magesan E, Chow J M and Gambetta J M 2016 Phys. Rev. D 93 060302
[28] Blais A, Huang R S, Wallraff A, Girvin S M and Schoelkopf R J 2004 Phys. Rev. A 69 062320
[29] Wallraff A, Stibnik D I, Blais A, Frunzio L, Major J, Kumar S, Girvin S M, Schoelkopf R J and Huang R S R S 2004 Nature 431 162–7
[30] McKay D C, Wood C J, Sheldon S, Chow J M and Gambetta J M 2017 Phys. Rev. A 96 (2) 022330
[31] Gambetta J M, Chow J M and Steffen M 2017 npj Quantum Information 3 2
[32] Bravyi S, Diao O, Gambetta J M, Gil D and Nazario Z 2022 J. Appl. Phys. 132 160902
[33] IBM Quantum Compute Resources https://quantum-computing.ibm.com/services/resources?tab=systems
[34] Aasi J et al 2015 Classical Quantum Gravity 32 074001
[35] Welch P 1967 IEEE Trans. Audio Electroacoust. 15 70–3
[36] Hunter J D 2007 Comput. Sci. Eng. 9 90–5
[37] Abbott R et al 2020 Astrophys. J. 900 L13
[38] Field S E, Galley C R, Hesthaven J S, Kaye J and Tiglio M 2014 Phys. Rev. X 4 031006
[39] Varma V, Field S E, Scheel M A, Blackman J, Gerosa D, Stein L C, Kidder L E and Pfeiffer H P 2019 Physical Review Research 1 033015
[40] Kechedzhi K, Isakov S V, Mandra S, Villalonga B, Mi X, Boixo S and Smelyanskiy V 2023 Effective Quantum Volume, Fidelity and Computational Cost of Noisy Quantum Processing Experiments Future Generation Computer Systems 153 431–41
[41] Abbott R et al 2021 SoftwareX 13 100658
[42] Abbott R et al 2023 Astrophys. J. Suppl. Ser. 267 29
[43] Karnaugh M 1953 Transactions of the American institute of electrical engineers Part I: Communication and Electronics 72 593–9