

Short-time quantum Fourier transform processing

Sreeraj Rajindran Nair

Dolby Laboratories,

University of Technology Sydney

Sydney, Australia

sreeraj.r.nair@student.uts.edu.au

Benjamin Southwell

Dolby Laboratories

Sydney, Australia

benjamin.southwell@dolby.com

Christopher Ferrie

University of Technology Sydney

Sydney, Australia

christopher.ferrie@uts.edu.au

Abstract—Algorithms for processing data in short-time batches are critical for both online and offline processing of streamed and large data respectively due to the quadratic relation between signal length and computational cost of convolution-based processing schemes. Whilst quantum analogs to some digital signal processing algorithms have been discovered, including the quantum Fourier transform (QFT), there has been no development of short-time processing techniques in the quantum domain. In this manuscript, we introduce the short-time QFT (STQFT) processing technique to bridge this gap in research. We develop a novel overlap-add reconstruction technique in the quantum domain using a permutation gate to combine subsequent windows. With this in mind, we discuss convolution under our novel STQFT processing scheme. We demonstrate filtering in the quantum Fourier domain with a filter stored in a quantum register as well as in a block encoded unitary gate. Throughout the paper, we elaborate upon implementation details such as applying DC offsets to input signals, skipping input data frames whenever necessary, the use of overlap-save as a reconstruction technique and mitigating time-varying scaling due to normalization of the windowed input data and filters.

Index Terms—quantum signal processing, convolution, quantum Fourier transform, short-time Fourier transform, sub-band processing, overlap-add

I. INTRODUCTION

Digital signal processing (DSP) using the short-time Fourier transform (STFT) [1] is vital for real-time applications and for offline processing of large data in smaller, more manageable chunks, as it is often infeasible to process the entire signal at once due to time or resource constraints [2] [3]. Short-time processing is well established within the classical domain, where it transforms $O(n^2)$ operations into $O(n\log(n))$ for convolution-based algorithms [4]. The STFT involves windowing a subset of data to be processed, transforming the windowed data into the Fourier domain, processing the signal, applying an inverse Fourier transform to the processed data and then reconstructing the output signal if synthesis is required. Signal reconstruction is typically done using overlap-add (OLA) or overlap-save [5].

The STFT is the basis for more advanced analysis and processing techniques that extend the STFT. This includes the use of a modified discrete Fourier transform (DFT) [6] as well as banding and transforming the values of the STFT

The research in this paper was carried out during a PhD internship at Dolby Australia. The first author was supported by Sydney Quantum Academy (SQA).

output, e.g. mel-frequency cepstrum coefficients [7]. Moreover, processing signals in the sub-bands of the STFT can often lead to more performant algorithms [8] in addition to an overall reduction in compute complexity if processing involves operations that scale quickly as the length increases such as convolution. These benefits are crucial to many applications including machine listening [7], [9], audio & video coding [10]–[12] and echo management [8], [13] for example.

Recent advances in quantum signal processing (QSP) have resulted in quantum counterparts to traditional DSP algorithms. Of particular significance is the quantum Fourier transform (QFT) [14]–[16] which is the counterpart to the discrete Fourier transform (DFT). Thus, it is of interest to investigate the short-time QFT (STQFT) to determine if such a processing paradigm can exist in the quantum domain and what the practical implementation issues are. In this manuscript, we introduce signal processing using the STQFT, present multiple methods for implementing filters in the quantum Fourier domain (QFD), develop a quantum overlap-add (QOLA) algorithm based on a permutation matrix and discuss practical implementation details. We also note how the overlap-save method is trivially applied to STQFT processing.

II. WINDOWING SIGNALS IN THE QUANTUM DOMAIN

The classical algorithm for performing signal reconstruction involves batching of the input signal into smaller manageable window frames. There are a set of standard windowing functions each with unique characteristics suitable for particular instances [17]. A common window function is the boxcar/rectangular window. It generates a subset of the long input signal of the corresponding window length. For the quantum reconstruction algorithm we perform windowing in similar manner. There have been few studies into windowing in the quantum domain [18]–[20]. Quantum windowing delineates from classical windowing as quantum encoding schemes require Euclidean normalization of the windowed signals [21]. After normalization we encode the windowed signal onto the qubits.

Quantum computing offers various forms of data encoding schemes, such as basis encoding and phase encoding [21] [22]. In this manuscript, we employ amplitude encoding [21] where the individual discrete sample values are embedded directly onto the namesake complex amplitudes of the qubit vector data types. This is an efficient scheme, as

N qubits can accommodate 2^N data samples, and it has been used to construct windowed quantum states [18]. We thus, are dealing with 2^N dimensional Hilbert space. While encoding, we propose that the windowed signal is padded out beforehand to accommodate for the expected expansion due to the convolution in the subsequent processing. The number of qubits is the closest power of 2 over the quantity $w_l + f_l - 1$ where w_l is the window length and f_l is the filter length. The quantum state initialised with this is of the form $|\psi_{window}\rangle = \sum_{i \in (0,1)^N} a_i |i\rangle$, where each a_i is a normalized windowed signal sample scalar. In practice, the measurement of this state will reveal the probabilities of each amplitude i.e a_i^2 , and hence for processing negative amplitudes, a DC offset is applied to the input signal before encoding data. Another resolution would be to modify our algorithm to work with other encoding schemes that build upon amplitude encoding such as Flexible Representation of Quantum Audio (FRQA) [23]. Furthermore, an all-zero windowed signal cannot be encoded with this scheme and hence its essential to track such states without quantum processing them and instead account for them in the final reconstructed signal as part of classical post-processing.

III. FILTERING IN THE SHORT-TIME QUANTUM FOURIER DOMAIN

A. Filter as a quantum register

Our first approach for convolution in the quantum domain is with the filter vector of arbitrary length being encoded within a separate quantum register in similar fashion to the amplitude encoding in the case of the windowed signal. The filter is padded to the same length that the windowed input data was. Once both registers are properly initialized we perform a quantum Fourier transform on both registers separately. With this operation the data is transformed to the Fourier domain which is a crucial part for the application of the convolution theorem [5]. To perform elementwise multiplication required for convolution in the quantum Fourier domain we apply repeated controlled pauli X-gate (C-NOT) operations between respective qubits of the two quantum registers [24] [25]. The two registers have the quantum states $|\psi_{window}\rangle = \sum_{i \in (0,1)^N} a_i |i\rangle$ and $|\psi_{filter}\rangle = \sum_{i \in (0,1)^N} b_i |i\rangle$. After the application of successive controlled Pauli X-gates (C-NOT) on respective qubits we would have $|\psi_{product}\rangle = \frac{1}{\sqrt{\sum |a_i b_i|^2}} \sum_{i \in (0,1)^N} a_i b_i |i\rangle$. This is inherently a probabilistic operation with the required elementwise product being observed on the first register when the second register is postselected for $|00\dots0\rangle$ state. With this postselection put in place the required state will be observed with a probability of $P = \sum |a_i b_i|^2$. Subsequently, the first register is subjected to an inverse quantum Fourier transform to obtain the quantum convolution output state. We used the IBM open-source Python framework Qiskit SDK [26] to implement this scheme and found that it matches the classical DSP convolution output within the tolerance of machine precision. The general circuit construction of this scheme is shown in Figure 1.

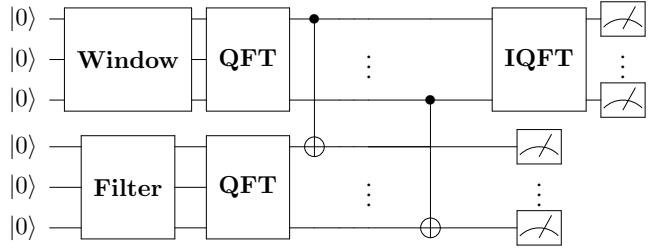


Fig. 1. Quantum convolution circuit with filter as a register circuit

This convolved quantum state can be matched to the classical convolution output by rescaling the classical read out of the quantum circuit with the normalization factors and the probability factors accrued during the operations. Filter as a quantum register approach would require $O(2^N)$ iterations to produce reliable results with further iterations to map out all the amplitudes of the quantum state. Existing work relates quantum amplitude amplification with the filter as a register scheme to produce more efficient run counts [24].

B. Filter as a block encoded matrix

A novel approach to perform same aforementioned elementwise multiplication is to use unitary operations in place of a separate quantum register. To carry out this operation, we need to be able to embed the filter vector as a diagonal matrix. Any other configurations are likely prone to introducing errors into the convolution output. The challenge with this approach is that such a diagonal matrix would not be of unitary nature. A resolution for this is to instead embed the filter vector as the diagonal block matrix within a much larger unitary matrix.

Block encoding [27] is one such scheme for embedding a properly scaled non-unitary matrix $A_{i,j} \in C$ onto a unitary matrix U_{Block} :

$$U_{Block} = \begin{bmatrix} A_{i,j} & * \\ * & * \end{bmatrix} \quad (1)$$

Here * denotes the arbitrary blocks we have no interest in. The action of this block encoded matrix on a general quantum state with ancilla qubits of the form $|\psi\rangle = |0\rangle |\alpha\rangle$ is as follows:

$$\begin{aligned} U_{Block} |\psi\rangle &= \begin{bmatrix} A_{i,j} & * \\ * & * \end{bmatrix} \begin{bmatrix} \alpha \\ 0 \end{bmatrix} = \begin{bmatrix} A_{i,j} \alpha \\ * \end{bmatrix} \\ &= |0\rangle (A_{i,j} |\alpha\rangle) + |1\rangle |* \rangle \end{aligned} \quad (2)$$

We propose the use of a matrix access oracle based block encoding scheme [27] [28]. The general structure of such a scheme as a quantum circuit is shown in Figure 2. U_A and U_B are oracles responsible for encoding the matrix elements of our diagonal matrix $A_{i,j}$ and ensure iteration over all combinations of the indices i, j respectively. A bottleneck in this scheme is that an efficient quantum gate decomposition for U_A and U_B may not always be found.

The oracles' U_A and U_B operation on a quantum state is defined as follows:

$$\begin{aligned} U_A |0\rangle_{anc} |i\rangle |j\rangle &= |A_{i,j}\rangle_{anc} |i\rangle |j\rangle, \\ U_B |i\rangle |j\rangle &= |j\rangle |i\rangle. \end{aligned} \quad (3)$$

where $|A_{i,j}\rangle_{anc} \equiv A_{i,j} |0\rangle_{anc} + \sqrt{1 - |A_{i,j}|^2} |1\rangle_{anc}$.

We use a more efficient scheme built on top of the oracle access approach known as the Fast Approximate BLock Encodings (FABLE) [27]. FABLE uses a combination of single R_y rotation ad controlled pauli X-gates (C-NOT) to construct the circuits for the oracles. The general circuit for these constructions is shown in Figure 2.

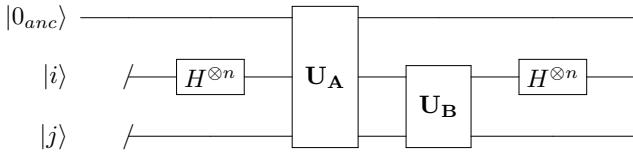


Fig. 2. Oracle access block encoding circuit

Before we can encode our diagonal A matrix we must transform the diagonal elements to the frequency domain. This is performed by finding the QFT coefficients of the filter with a separate circuit before passing these to the matrix A . As before we pad the length of the diagonal with zeros to account for the anticipated expansion to length $n \geq w_l + f_l - 1$ due to the convolution in the subsequent steps. We now have a $n \times n$ filter block within the unitary matrix. This will operate on $\log_2 n$ primary qubits and additional ancilla qubits.

$$U_{Block} = \begin{bmatrix} a_{1,1} & 0 & 0 & \dots & 0 & 0 \\ 0 & \ddots & 0 & \dots & 0 & 0 \\ 0 & 0 & a_{f_l, f_l} & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & \ddots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix} *$$

This block encode gate decomposition is applied to the Fourier transformed windowed signal encoded state. Ancilla qubits are introduced to accommodate the large operational Hilbert space of the U_{Block} unitary operator. Finally, an inverse quantum Fourier transform is performed on this state to obtain the convolution output for the windowed signal. For our experiments we used the open-source Python framework Pennylane SDK [29] to generate the FABLE matrices and their gate decompositions. These gate decompositions were then used within a local quantum simulator to perform windowed quantum convolutions and found it matches the traditional DSP convolution output within the tolerance of machine precision. The general structure of the circuit is shown in Figure 3.

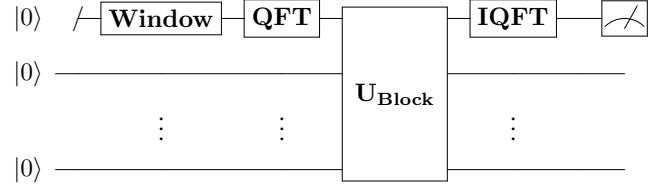


Fig. 3. Quantum convolution circuit with filter as a block matrix

IV. QUANTUM OVERLAP-ADD

Classical reconstruction of a processed windowed signal is widely performed using either overlap-add or overlap-save. To the best of the authors' knowledge, a quantum version of these sub-routines has not been developed. In this section we propose a reconstruction algorithm in the form of a quantum overlap-add (QOLA) circuit.

For QOLA, we collect the classical readouts of windowed quantum convolution outputs. In general, these would be of identical lengths. After rescaling to account for the normalization factors and probability terms, these are zero padded to the closest 2^N length if required. A pair of consecutive window convolution outputs $|\phi_a\rangle$ and $|\phi_b\rangle$ each of length r are amplitude encoded together into a $\log_2(2r)$ qubit register to create the state $|\phi_{QOLA}\rangle = |0\rangle|\phi_a\rangle + |1\rangle|\phi_b\rangle$ with an additional euclidean norm.

As was the case with the windowed quantum state initialization, an all-zero pair of windowed convolution outputs cannot be processed with this scheme and must be relegated to the classical post-processing wherein we insert all-zero frames between reconstructed output frames. A controlled NOT operation is performed between an ancilla qubit and the 1st qubit in $|\phi_{QOLA}\rangle$ to give new decomposition terms $|\phi'_a\rangle$ and $|\phi'_b\rangle$. This gives the state:

$$\begin{aligned} |0\rangle|\phi_{QOLA}\rangle &= |0\rangle|0\rangle|\phi_a\rangle + |1\rangle|1\rangle|\phi_b\rangle \\ &= |0\rangle|\phi'_a\rangle + |1\rangle|\phi'_b\rangle \\ &= |0\rangle \otimes [\phi_1, \phi_2, \dots, \phi_r, 0, 0, \dots, 0] \\ &\quad + |1\rangle \otimes [0, 0, \dots, 0, \phi_{r+1}, \phi_{r+2}, \dots, \phi_{2r}] \end{aligned} \quad (4)$$

An arbitrary ancilla qubit controlled permutation matrix ($C - U_{PERM}$) reorganizes the coefficients in $|\phi_b\rangle$ to the desired overlap ratio configuration. We define a $2r \times 2r$ permutation matrix for l overlapping elements:

$$U_{PERM} = \begin{bmatrix} I_{r-l \times r-l} & \mathbf{0}_{r-l \times l} & \mathbf{0}_{r-l \times r} \\ \mathbf{0}_{r \times r-l} & \mathbf{0}_{r \times l} & I_{r \times r} \\ \mathbf{0}_{l \times r-l} & I_{l \times l} & \mathbf{0}_{l \times r} \end{bmatrix}_{2r \times 2r} \quad (5)$$

U_{PERM} is then converted into the controlled permutation gate ($C - U_{PERM}$).

$$\begin{aligned} \xrightarrow{C - U_{PERM}} & |0\rangle \otimes [\phi_1, \phi_2, \dots, \phi_r, 0, 0, \dots, 0] \\ & + |1\rangle \otimes [0, 0, \dots, 0, \phi_{r+1}, \phi_{r+2}, \dots, \phi_{2r}, 0, 0, \dots, 0] \end{aligned} \quad (6)$$

A Hadamard operation on the ancilla qubit initiates the addition operation. A postselection of $|0\rangle$ on the ancilla qubit implies a quantum overlap-add state on the first register. On the other hand $|1\rangle$ on the ancilla qubit implies a difference state on the first register.

$$\begin{aligned}
 & \xrightarrow{\text{Hadamard}} \frac{1}{\sqrt{2}} |0\rangle \otimes [|\phi_a'\rangle + |\phi_b'\rangle] \\
 & + \frac{1}{\sqrt{2}} |1\rangle \otimes [|\phi_a'\rangle - |\phi_b'\rangle] \\
 & = \frac{1}{\sqrt{2}} |0\rangle \otimes |SUM\rangle \\
 & + \frac{1}{\sqrt{2}} |1\rangle \otimes |DIFFERENCE\rangle
 \end{aligned} \tag{7}$$

Both these measurements occur with probabilities of $\frac{1}{2} ||SUM\rangle^2$ and $\frac{1}{2} ||DIFFERENCE\rangle^2$ respectively. The classical readouts are rescaled with these probability and the normalization factors and then we obtain the required output.

In general, each windowed convolution output undergoes overlap-add twice, one with the preceding windowed convolution output and one with the succeeding one. We replicated these results with Qiskit and found it matches the traditional OLA within the tolerance of machine precision. The (3+1) qubit circuit for this operation is shown in Figure 4.

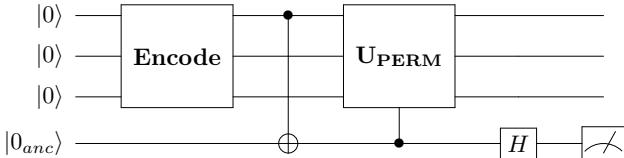


Fig. 4. Quantum overlap-add circuit for (3+1) qubits

V. DISCUSSION

The oracle access based block encoding scheme was found more adept at generating efficient block encodings for the particular filter matrices we are interested in. Investigations into the linear combination of unitaries (LCU) based block encoding schemes didn't yield favorable results for our requirements. It is worth noting that the standard oracle access scheme we demonstrated does not scale efficiently in terms of processing time or circuit depth w.r.t the filter length. We observed that it was better to generate these block encodings for standard filters beforehand. We then call these encodings in, when we perform real-time processing. FABLE does offer more optimal block encodings in the case of certain classes of sparse and/or structured block matrices. However, our required filter matrices did not fit into these classes. In the same vein, it is worth investigating if the structure of the required filter matrices can be exploited to generate efficient block encodings. Furthermore, instead of processing the filter array with a QFT circuit separately, we can directly block encode the filter as such and use techniques from QSP [30] and quantum singular value transform (QSVT) [31] to transform the filter block to the Fourier domain. This is likely to incur penalties in the form of additional ancilla qubits.

The QOLA method accepts inputs as classical data and performs re-encoding of the windowed convolution outputs. It is possible to design QOLA circuits to accept windowed quantum convolution states as input and generate reconstructed frames. However, given that each of the windowed quantum convolution outputs have different norms means that the reconstructed frames would come out scaled unevenly. One workaround is to introduce an arbitrary scalar factor multiplier quantum circuit as a buffer between these two sub-routines, however it is not clear if this can be done without introducing new norm scales along the way. If this can be achieved, either a pair of convolution circuits can be used simultaneously or a single convolution circuit can be used to generate at least two set of output states for each window frame. In later case these states can then be temporarily stored in quantum memory elements such as QRAM [32] before passing it to the QOLA circuit. This is likely to cause latency issues when deployed to a real-time processing environment. It is tantamount that this sub-routine is optimized to handle these anticipated latency issues.

Overlap-save is yet another popular signal reconstruction scheme widely used in DSP. We do not need to design a separate quantum circuit in order to adapt this scheme to the quantum data domain. It uses windowed circular convolution outputs instead of linear convolution outputs and a snipping of the output data. A simple end-to-end stacking of these frames constitutes an overlap-save reconstructed signal. This entails that there is no zero padding to 2^N over $w_l + f_l - 1$ in the case of quantum convolution. We replace the non-overlapping windows from our scheme with overlapping windows, where filter length determines the overlap ratio. The convolved output wraps around in the final windowed quantum convolution state. Classical read-outs are obtained and these are then processed exactly as in traditional DSP.

VI. CONCLUSION

In this manuscript, we introduced a novel algorithm for processing signals within the short-time quantum Fourier domain. Quantum filtering sub-routines were designed with two different approaches: one where the filter was encoded as a quantum register and the other where it was encoded as a block encoded unitary gate. The convolved windowed signal output frames obtained from these quantum convolutions were then reconstructed with the novel quantum overlap-add technique we proposed. We discussed the current bottlenecks in our approach with the critical one being the classical intermediates between the windowed convolution sub-routine and the quantum overlap-add subroutine. These and several other avenues for improvement to our current scheme were discussed. We foresee STQFT processing to have applications in real-time signal processing such as live streaming and in efficient processing of large data in the near future.

ACKNOWLEDGMENT

S.R.N would like to thank Afrah Basheer for valuable discussions.

REFERENCES

- [1] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," *IEEE transactions on acoustics, speech, and signal processing*, vol. 25, no. 3, pp. 235–238, 1977.
- [2] J. Allen and L. Rabiner, "A unified approach to short-time fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.
- [3] P. Duhamel and M. Vetterli, "Fast fourier transforms: A tutorial review and a state of the art," *Signal Processing*, vol. 19, no. 4, pp. 259–299, 1990.
- [4] T. G. Stockham, "High-speed convolution and correlation," in *Proceedings of the April 26–28, 1966, Spring Joint Computer Conference*, ser. AFIPS '66 (Spring). New York, NY, USA: Association for Computing Machinery, 1966, p. 229–233.
- [5] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.
- [6] T. Karp and N. J. Fliege, "Modified dft filter banks with perfect reconstruction," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 11, pp. 1404–1414, 1999.
- [7] G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," *Applied Acoustics*, vol. 158, p. 107020, 2020.
- [8] K.-A. Lee, W.-S. Gan, and S. M. Kuo, *Subband adaptive filtering: theory and implementation*. John Wiley & Sons, 2009.
- [9] A. B. Nassif, I. Shahin, I. Attili, M. Azzeb, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE access*, vol. 7, pp. 19143–19165, 2019.
- [10] M. Vetterli and J. Kovacevic, *Wavelets and subband coding*. Prentice-hall, 1995.
- [11] R. L. Andersen, B. G. Crockett, G. A. Davidson, M. F. Davis, L. D. Fielder, S. C. Turner, M. S. Vinton, and P. A. Williams, "Introduction to dolby digital plus, an enhancement to the dolby digital coding system," in *Audio Engineering Society Convention 117*. Audio Engineering Society, 2004.
- [12] A. Spanias, T. Painter, and V. Atti, *Audio signal processing and coding*. John Wiley & Sons, 2006.
- [13] A. Gilloire and M. Vetterli, "Adaptive filtering in sub-bands with critical sampling: analysis, experiments, and application to acoustic echo cancellation," *IEEE transactions on signal processing*, vol. 40, no. 8, pp. 1862–1875, 1992.
- [14] Y. C. Eldar and A. V. Oppenheim, "Quantum signal processing," *IEEE Signal Processing Magazine*, vol. 19, no. 6, pp. 12–32, November 2002.
- [15] D. Coppersmith, "An approximate fourier transform useful in quantum factoring," 2002, preprint.
- [16] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134.
- [17] K. Prabhu, *Window Functions and Their Applications in Signal Processing*, 10 2013.
- [18] H. Yin, D. Lu, and R. Zhang, "Quantum windowed fourier transform and its application to quantum signal processing," *Int J Theor Phys*, vol. 60, pp. 3896–3918, 2021.
- [19] G. Rendon, T. Izubuchi, and Y. Kikuchi, "Effects of cosine tapering window on quantum phase estimation," *Phys. Rev. D*, vol. 106, p. 034503, Aug 2022.
- [20] S. Greenaway, W. Pol, and S. Sim, "A case study against qsvt: assessment of quantum phase estimation improved by signal processing techniques," 2024, [quant-ph].
- [21] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [22] M. Weigold, J. Barzen, F. Leymann, and M. Salm, "Data encoding patterns for quantum computing," in *Proceedings of the 27th Conference on Pattern Languages of Programs*, ser. PLoP '20. USA: The Hillside Group, 2022.
- [23] Yan *et al.*, "Flexible representation and manipulation of audio signals on quantum computers," *Theoretical Computer Science*, vol. 752, no. 2017, January 2017.
- [24] M. Ramezani, M. Nikaeen, F. Farman, S. M. Ashrafi, and A. Bahrampour, "Quantum multiplication algorithm based on the convolution theorem," *Phys. Rev. A*, vol. 108, p. 052405, Nov 2023.
- [25] A. M. Grigoryan and S. S. Agaian, "3-qubit circular quantum convolution computation using the fourier transform with illustrative examples," *Journal of Quantum Computing*, vol. 6, pp. 1–14, 2024.
- [26] A. Javadi-Abhari *et al.*, "Quantum computing with qiskit," 2024, arxiv.org/abs/2405.08810.
- [27] D. Camps and R. Van Beeumen, "Fable: Fast approximate quantum circuits for block-encodings," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2022, pp. 104–113.
- [28] D. Camps, L. Lin, R. Van Beeumen, and C. Yang, "Explicit quantum circuits for block encodings of certain sparse matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 45, no. 1, pp. 801–827, 2024.
- [29] V. Bergholm *et al.*, "PennyLane," arXiv:1811.04968.
- [30] D. Motlagh and N. Wiebe, "Generalized quantum signal processing," *PRX Quantum*, vol. 5, 2024.
- [31] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetic," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*, 2019, pp. 193–204.
- [32] P. Koustubh, A. Chatterjee, and S. Ghosh, "Quantum random access memory for dummies," *Sensors*, vol. 23, no. 17, p. 7462, 2023.