

Full Length Article



# Quantum wave packet transforms with compact frequency support: Implementations for wavelets and Gabor atoms

Hongkang Ni \*, Lexing Ying <sup>1</sup>

Stanford University, 94305, Stanford, CA, USA

## ARTICLE INFO

Communicated by: Radu Balan

2020 MSC:

81P68

68Q12

42C40

65T60

Keywords:

Quantum algorithm

Wavelet transform

Gabor atom

Wave packet transform

## ABSTRACT

Various wave packet transforms are widely used to extract multiscale structures in signal processing. This paper introduces the quantum circuit implementation of a broad class of wave packets, including Gabor atoms and wavelets, with compact frequency support. Our approach operates in the frequency space, involving reallocation and reshuffling of signals tailored for manipulation on quantum computers. The resulting implementation differs from existing quantum algorithms for spatially compactly supported wavelets and can be readily extended to quantum transforms of other wave packets with compact frequency support.

## 1. Introduction

Various types of wave packets, such as wavelets [1], Gabor atoms, curvelets [2], wavelet packets [3], and wave atoms [4], have been extensively explored and applied to many applications. In contrast to the Fourier and the Dirac bases, these wave packets exhibit locality in both the spatial and frequency domains, along with a controlled level of smoothness, making them advantageous for image or other signal processing tasks. Additionally, research has demonstrated that the structure of these wave packet bases offers sparse representations for some differential operators, enabling their use as effective preconditioners for partial differential equations and facilitating the development of rapid algorithms [5,6].

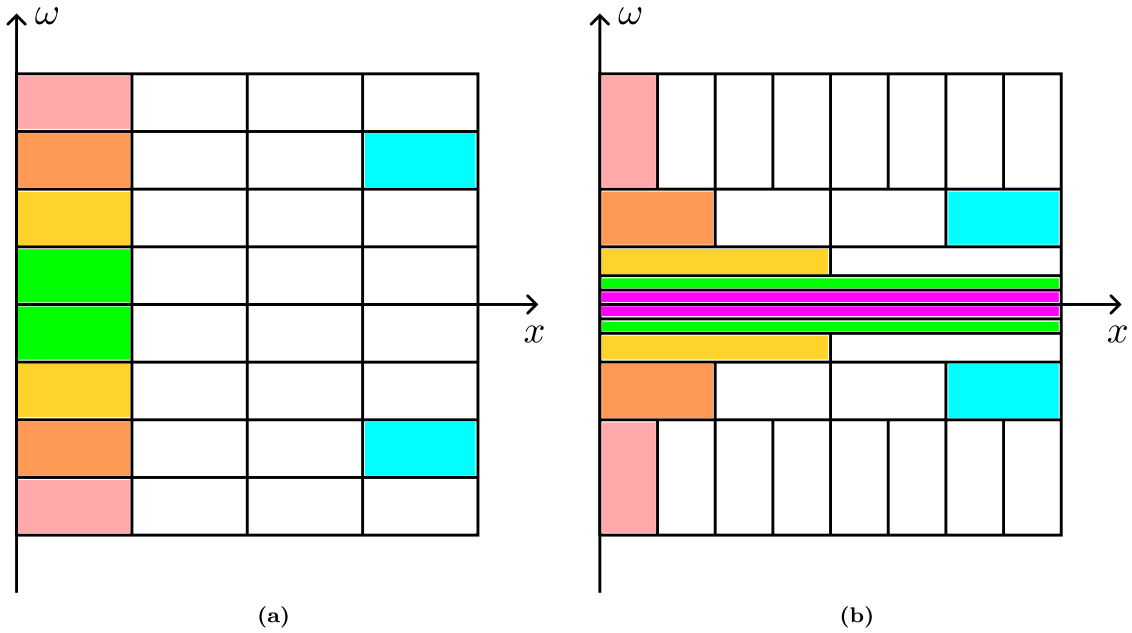
There has been growing interest in quantum algorithms utilizing wavelet or other wave packet transformations. Research has demonstrated the applicability of quantum wavelet transforms across various areas of image processing [7,8], including image compression and encryption [9], image denoising [10], and image watermarking [11–14]. Furthermore, quantum wavelet transform has found applications in diverse fields such as quantum steganography [15], quantum-to-classical data decoding [16], and signal reconstruction [17]. Notably, quantum wavelet and curvelet transforms have also been found valuable in encoding differential operators [18,19] and in the domain of quantum sampling [20].

Similar to the Fourier transform, quantum computing offers the potential for exponential speedup in wave-packet transforms. Various quantum wavelet transforms have been implemented. Earlier studies included well-known wavelets such as the Haar wavelet

\* Corresponding author.

E-mail addresses: [hongkang@stanford.edu](mailto:hongkang@stanford.edu) (H. Ni), [lexing@stanford.edu](mailto:lexing@stanford.edu) (L. Ying).

<sup>1</sup> The work of L.Y. is partially supported by the National Science Foundation under awards DMS-2011699 and DMS-2208163.



**Fig. 1.** (a) Illustration for Gabor atom tiling. Each pair of tiles symmetric with respect to the  $x$  axis represents the essential support of a basis function  $\psi_{j,p}$ . The red, orange, yellow, and green tiles represent  $\psi_{4,0}$ ,  $\psi_{3,0}$ ,  $\psi_{2,0}$ , and  $\psi_{1,0}$  respectively. All tiles with the same  $\omega$  are at the same level  $j$ . For instance, the blue tile represents  $\psi_{3,3}$ . (b) Illustration for wavelet tiling. Each pair of tiles symmetric with respect to the  $x$  axis represents the essential support of a basis function  $\psi_{j,p}$ . This diagram shows the truncation up to  $n = 4$ . The red, orange, yellow, green, and blue tiles represent  $\psi_{1,0}$ ,  $\psi_{2,0}$ ,  $\psi_{3,0}$ ,  $\psi_{4,0}$ , and  $\psi_{2,3}$  respectively. The purple tiles represent the scaling function. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and Daubechies D4 wavelet [21,22]. Research has been extended to higher-order wavelets [23] and a more general setting [24]. The exploration of multi-dimensional quantum wavelet transform has also been undertaken [25–27]. Wavelet packet transforms have been addressed in studies like [28,29]. However, there is a noticeable gap in research concerning other types of wave packet transforms. Liu [20] discussed the multidimensional curvelet transform and its potential applications, but the discussion remains conceptual, lacking a detailed exploration of its implementation.

### 1.1. Main contributions

Previous studies on quantum wavelet transforms have mostly used finite-size spatial-domain filters, which are applicable only to spatially compactly supported wavelets. On the other hand, wave packet constructions with compact frequency support, such as the Meyer wavelet, exhibit favorable characteristics in the frequency domain. As far as we know, no existing algorithm has explored the actual implementations on quantum computers. Our contribution lies in operating in the frequency domain and implementing such wave packet constructions.

We first consider the Gabor atoms of uniform frequency partitioning, with both sharp and blended frequency windows. The implementation given here is the first one on quantum computers and also serves as a preparation for more advanced wave packets.

Next, we consider the wavelets of multiscale frequency partitioning with both sharp and blended frequency windows. The two examples studied in detail are Shannon wavelets and Meyer wavelets. We show that these wavelets may be realized with no more than three ancilla qubits, thanks to their favorable forms in the frequency domain. In contrast, for the wavelets with compact spatial support, the number of ancilla qubits typically grows with the order of the wavelet [23].

### 1.2. Background on Gabor atoms and wavelets

Different tilings of phase-space diagrams result in different basis sets, which can be collectively called wave packets. This work focuses on two such wave packets: Gabor atoms and wavelets.

#### 1.2.1. Gabor atoms

Let us first consider a uniform phase-space tiling, as illustrated in Fig. 1(a). The basis functions given by this uniform tiling are called Gabor atoms. A basis function can be indexed by its frequency level  $j \in \mathbb{Z}_{>0}$  and its space position  $p \in \mathbb{Z}$ . As the simplest example, the Gabor atoms with *sharp* frequency windows are given by

$$\widehat{\psi}_{j,p}(\omega) = \sqrt{\frac{A}{2\pi}} e^{iA\omega p} \chi_{[j\frac{\pi}{A}, (j+1)\frac{\pi}{A}]}(|\omega|), \tag{1.1}$$

where  $\chi$  stands for the indicator function. Here, we adopt the convention

$$\hat{\psi}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \psi(x)e^{i\omega x} dx,$$

for continuous Fourier transform, and  $A \in \mathbb{Z}_{>0}$  is a scaling parameter that can be arbitrarily chosen. Though with perfect frequency localization, these atoms have slow spatial decay.

A smooth frequency windowing is needed for Gabor atoms with better spatial localization. For this purpose, we can define a bump function

$$g(s) = \begin{cases} \cos(\frac{\pi}{2}\beta(s/\pi)) & \text{if } -\pi < s < \pi, \\ 0 & \text{otherwise,} \end{cases} \tag{1.2}$$

where the function  $\beta(s)$  should satisfy

$$\beta(s) + \beta(1 - s) = 1 \text{ and } \beta(-s) = \beta(s). \tag{1.3}$$

The bump function  $g(s)$  can be viewed as a smoothed version of the characteristic function  $\chi_{[-\frac{\pi}{2}, \frac{\pi}{2}]}$ , and the order of smoothness is determined by  $\beta$ . The Gabor atoms with *blended* frequency windows are given by

$$\widehat{\psi}_{j,p}(\omega) = \sqrt{\frac{A}{2\pi}} e^{iA\omega p} \left( e^{-i\frac{1}{2}(A\omega - (j + \frac{1}{2})\pi)} g\left(A\omega - (j + \frac{1}{2})\pi\right) + e^{-i\frac{1}{2}(A\omega + (j + \frac{1}{2})\pi)} g\left(A\omega + (j + \frac{1}{2})\pi\right) \right). \tag{1.4}$$

### 1.2.2. Wavelets

For wavelets, the phase-space tiling is given as in Fig. 1(b). More specifically, a wavelet basis is given by a function  $\psi : \mathbb{R} \rightarrow \mathbb{C}$  and its multiscale counterparts:

$$\psi_{j,p}(x) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{x - 2^j p}{2^j}\right),$$

such that  $\{\psi_{j,p}\}_{(j,p) \in \mathbb{Z}^2}$  form an orthonormal basis of  $L^2(\mathbb{R})$ . As a result, their Fourier transforms

$$\widehat{\psi}_{j,p}(\omega) = \sqrt{\frac{2^j}{2\pi}} e^{i2^j \omega p} \hat{\psi}(2^j \omega) \tag{1.5}$$

also form an orthonormal basis of  $L^2(\mathbb{R})$ . When the frequency window  $\hat{\psi}(\omega)$  is a *sharp* indicator function

$$\hat{\psi}(\omega) = \chi_{[-2\pi, -\pi] \cup [\pi, 2\pi]}(\omega) = \chi_{[\pi, 2\pi]}(|\omega|), \tag{1.6}$$

the resulting basis functions are the Shannon wavelets.

We may also consider smooth *blended* frequency windows, for example,

$$\hat{\psi}(\omega) = \begin{cases} e^{-i\frac{\omega}{2}} g\left(\frac{3\omega}{2} - 2\pi\right) & \text{if } \frac{2\pi}{3} \leq \omega \leq \frac{4\pi}{3}, \\ e^{-i\frac{\omega}{2}} g\left(\frac{3\omega}{4} - \pi\right) & \text{if } \frac{4\pi}{3} \leq \omega \leq \frac{8\pi}{3}, \\ 0 & \text{if } 0 \leq \omega \leq \frac{2\pi}{3} \text{ or } \omega \geq \frac{8\pi}{3}, \\ \hat{\psi}(-\omega)^* & \text{if } \omega < 0, \end{cases} \tag{1.7}$$

where  $\hat{\psi}(-\omega)^*$  means the conjugate of complex number  $\hat{\psi}(-\omega)$ . This results in the famous Meyer wavelets [1, Chapter 7.2.2].

In this article, the matrix subscripts ‘‘G’’ and ‘‘W’’ indicate that they are related to Gabor atoms and wavelets, respectively, and the subscripts ‘‘S’’ and ‘‘B’’ stand for sharp and blended frequency windows, respectively. In particular, the four main transformations in the paper are denoted as  $U_{GS}$ ,  $U_{GB}$ ,  $U_{WS}$ , and  $U_{WB}$ , i.e, sharp Gabor atoms, blended Gabor atoms, Shannon (sharp) wavelets, and Meyer (blended) wavelets. For the matrices that we want to specify the dimension, we write the dimension as a subscript. The italicized subscripts representing dimensions, such as  $N$ ,  $M$ , and  $B$ , can be readily distinguished from the Roman subscripts denoting names, such as the aforementioned ‘‘G’’, ‘‘W’’, ‘‘S’’, and ‘‘B’’.

### 1.3. Notations and background on quantum circuits

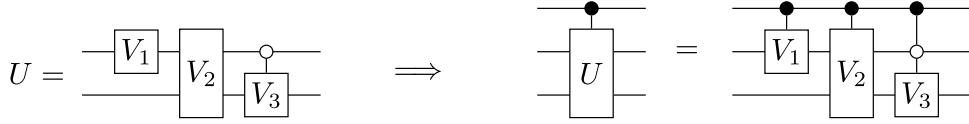
Let us first summarize the notations and conventions used in this paper, which are compatible with the standard notations in quantum computing literature. We use  $\otimes$  to denote the tensor product and  $\oplus$  for the direct product of matrices. Occasionally,  $\oplus$  is also used as the notation of modulo 2 plus in bit-wise operations, but the meaning should be clear based on the context. The notation  $[M]$  stands for the set  $\{0, 1, \dots, M - 1\}$ . The matrix  $I_M$  means the identity matrix of dimension  $M$ . We use  $z^*$  to represent the conjugate of a complex number  $z$ , and  $A^\dagger$  the conjugate transpose of matrix  $A$ .

We adopt Dirac’s notation  $|\cdot\rangle$  to represent a quantum state as a column vector and  $\langle \cdot |$  as its conjugate transpose. A qubit lives in the space  $\mathbb{C}^2 = \text{span}\{|0\rangle, |1\rangle\}$ , where  $|0\rangle$  and  $|1\rangle$  represent  $[1, 0]^T$  and  $[0, 1]^T$ , respectively. For a non-negative integer  $x$  with binary representation  $x = (x_{m-1}x_{m-2} \dots x_0)_2$ , we denote the  $m$ -qubit state

$$|x\rangle := |x_{m-1}x_{m-2} \dots x_0\rangle = |x_{m-1}\rangle |x_{m-2}\rangle \dots |x_0\rangle = |x_{m-1}\rangle \otimes |x_{m-2}\rangle \otimes \dots \otimes |x_0\rangle. \tag{1.8}$$



**Fig. 2.** (a) CNOT gate. (b)  $|0\rangle$ -controlled-NOT gate. (c) SWAP gate. (d) An example of a multi-qubit control gate  $|11\rangle\langle 11| \otimes U + (I_4 - |11\rangle\langle 11|) \otimes I$ . The two filled nodes indicate that  $U$  is applied to the third qubit if and only if both the first two qubits are at state  $|1\rangle$ .



**Fig. 3.** An example of controlled- $U$  circuit. The rightmost gate represents the unitary  $|10\rangle\langle 10| \otimes V_3 + (I_4 - |10\rangle\langle 10|) \otimes I$ , which applies  $V_3$  to the third qubit only when the first qubit is in state  $|1\rangle$  and the second qubit is in state  $|0\rangle$ .

In this convention, we can identify any unit vector  $f \in \mathbb{C}^M$  as an  $m$ -qubit state  $\sum_{x=0}^{M-1} f(x) |x\rangle$ . Due to the basic assumptions of quantum mechanics, we always assume the vectors have norm 1, and all operators we consider are unitary matrices correspondingly.

The unitary manipulations are often drawn as boxes in the quantum circuits, where we use horizontal wires to represent qubits. The ordering convention we adopt here is that the higher digit of the binary representation (1.8) will correspond to the upper qubit when drawing the quantum circuit. More examples of quantum circuits can be found in quantum computing textbooks [30,31]. In the present paper, we denote  $n = \log_2 N$  as the number of qubits utilized for implementing the wave packet transform in the space  $\mathbb{C}^N$ . Additionally,  $M = 2^m$  is often used for the dimension of matrices acting as intermediate steps, with the value of  $M$  left temporarily unspecified.

$X, Y, Z$ , and  $H$  represent the Pauli- $X$ , Pauli- $Y$ , Pauli- $Z$ , and Hadamard matrices respectively, defined as

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

We also extensively use some basic two-qubit gates. The  $|1\rangle$ -controlled-NOT gate is  $|0\rangle\langle 0| \otimes I_2 + |1\rangle\langle 1| \otimes X$ , and is usually just called the CNOT gate. In a quantum circuit, this gate is drawn as in Fig. 2(a), where the filled node means only applying the lower gate when the upper qubit is  $|1\rangle$ . We usually use “ $\oplus$ ” to represent “ $X$ ” in quantum circuits. The related  $|0\rangle$ -controlled-NOT gate,  $|0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes I_2$ , is given in Fig. 2(b), where the unfilled node has just the opposite meaning as the filled one. The swap gate  $\text{SWAP} = |00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|$ , which swaps the value of two qubits, is shown as Fig. 2(c). We sometimes also make use of the multi-qubit controlled gate, which can be built by single-qubit gates and CNOT gates [32]. For instance,  $|11\rangle\langle 11| \otimes U + (I_4 - |11\rangle\langle 11|) \otimes I$  is shown in Fig. 2(d).

In general, we can also build a controlled version of any unitary  $U$  with a potentially complicated implementation. Once we express  $U$  as a sequence of elementary gates, we can form controlled- $U$  by controlling each gate in that sequence. In particular, the controlled version of a controlled- $V$  gate becomes a multi-qubit controlled- $V$  gate. An example is shown in Fig. 3.

An  $(m+n)$ -qubit unitary operator  $U$  is called a  $(\gamma, m, \epsilon)$ -block-encoding of an  $N \times N$  matrix  $A$ , if

$$\|A - \gamma(\langle 0^m| \otimes I_N)U(|0^m\rangle \otimes I_N)\| \leq \epsilon.$$

In the matrix form, a  $(\gamma, m, \epsilon)$ -block-encoding is a  $2^{m+n}$  dimensional unitary matrix

$$U = \begin{pmatrix} \tilde{A}/\gamma & * \\ * & * \end{pmatrix}$$

where  $*$  can be any block matrices of the correct size, and  $\|\tilde{A} - A\| \leq \epsilon$ .

The inner product of two complex vectors  $f_1$  and  $f_2$  is denoted by

$$\langle f_1 | f_2 \rangle := \langle f_1, f_2 \rangle = \sum_{t=0}^{N-1} f_1(t) f_2(t)^*. \tag{1.9}$$

We denote  $f(m_1 : m_2)$  as the restriction of vector  $f$  from the  $m_1$ th element to the  $m_2$ th element.

For any  $j$  in the computational basis, the (discrete) forward Fourier transform is defined as

$$U_{\text{FT}(N)}|j\rangle = \frac{1}{\sqrt{N}} \sum_{k \in [N]} e^{i2\pi \frac{kj}{N}} |k\rangle,$$

which is widely used in the quantum computing literature, but has a sign difference compared to the classical signal processing convention Figs. 2, 7.

The indices of vectors in this paper are always considered in the modulo  $N$  sense, i.e.,  $f(-k) = f(N - k)$ . In particular, we adopt the convention  $|-k\rangle = |N - k\rangle$  in the bracket notation.

## 2. Gabor atoms

This section delves into the implementation of the quantum version of the Gabor atom transform. Specifically, we will discuss the discretization and present the quantum circuits for both the sharp Gabor atoms (1.1) and the blended Gabor atoms (1.4). Implementing the sharp Gabor atoms can be viewed as first reshuffling the Fourier transform  $\hat{f}$  of the input and then performing an inverse Fourier transform for each sharp frequency window. The blended Gabor atoms are more intricate due to the overlapping of neighboring packets. Our approach entails first recombining parts of  $\hat{f}$  to the proper locations, which is a unitary process, and then applying the circuit of the sharp Gabor atoms. The recombining process involves the implementation of certain diagonal matrices, which will also be discussed in detail.

### 2.1. Sharp Gabor atoms

Given the basis functions (1.1), let us introduce the discrete sharp Gabor atoms in the frequency domain. We restrict  $\omega \in [-\pi, \pi)$  to grid points  $\{\frac{2\pi}{N}k\}$ , where  $k = -N/2, \dots, N/2 - 1$ . Denote  $B = \frac{N}{2A}$  and  $b = \log_2 B$ . The discrete sharp Gabor atoms are defined in the (discrete) frequency space as

$$\widehat{\psi}_{j,p}(k) := \frac{1}{\sqrt{2B}} e^{2\pi i \frac{pk}{2B}} [\chi_{[jB, (j+1)B)}(k) + \chi_{[-(j+1)B, -jB)}(k)],$$

where  $\psi_{j,p}$  is for the discrete basis function in order to distinguish from its continuous counterpart  $\Psi_{j,p}$ . We also point out that the discrete and continuous basis corresponds in the sense that  $\widehat{\psi}_{j,p}(k) := \sqrt{\frac{2\pi}{N}} \widehat{\Psi}_{j,p}(\frac{2\pi}{N}k)$ . Only the basis functions with indices  $j \in [A]$  and  $p \in [2B]$  are used in the discrete setting, which is in total  $N$  basis functions. Typically, we may choose  $A = 2^{\lfloor n/2 \rfloor}$  and  $B = 2^{\lfloor (n-1)/2 \rfloor}$  to balance the resolution of space and frequency, though other choices of  $A$  and  $B$  are also acceptable. To simplify indexing, we further denote  $\psi_{j,p}$  as  $\psi_{2Bj+p}$ , thereby aligning the basis functions with indices ranging from 0 to  $N - 1$ . More explicitly, the Fourier coefficients of the sharp Gabor atom basis are

$$\widehat{\psi_{2Bj+p}}(k) := \widehat{\psi_{j,p}}(k), \tag{2.1}$$

for  $j \in [\frac{N}{2B}]$ ,  $p \in [2B]$ , and  $k \in \{-\frac{N}{2}, -\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\}$ . Here,  $\chi_I$  means the characteristic function on the interval  $I$ . In some other literature, there may be an additional phase on the characteristic functions in the sharp Gabor atoms, which is not adopted here and can be viewed as a special version of the blended Gabor atoms discussed in Section 2.2.

For a signal  $f \in \mathbb{C}^N$ , one can expand  $f$  under this basis

$$f = \sum_{j \in [\frac{N}{2B}]} \sum_{p \in [2B]} a_{2Bj+p} \psi_{2Bj+p} = \sum_{n=0}^{N-1} a_n \psi_n, \tag{2.2}$$

where  $a_{2Bj+p}$ 's are the Gabor atom coefficients. Notice that the  $\psi_{2Bj+p}$  here is the discrete Fourier inverse of (2.1), which is not the direct restriction of the continuous  $\Psi_{j,p}$  in the space domain. The sharp Gabor atom transform is defined as

$$U_{GS} : \mathbb{C}^N \rightarrow \mathbb{C}^N : f \mapsto a = (a_0, a_1, \dots, a_{N-1})^T. \tag{2.3}$$

One can check that the basis given by (2.1) is orthonormal. Therefore, the coefficients of the signal  $f$  are given by the inner product

$$a_{2Bj+p} = \langle f, \psi_{2Bj+p} \rangle = \langle \hat{f}, \widehat{\psi_{2Bj+p}} \rangle = \left( \sum_{k=jB}^{(j+1)B-1} + \sum_{k=-(j+1)B}^{-jB-1} \right) \frac{1}{\sqrt{2B}} e^{-2\pi i \frac{pk}{2B}} \hat{f}(k), \tag{2.4}$$

where the second step is the Plancherel's identity. When  $j$  is even, this gives

$$\begin{aligned} a_{2Bj+p} &= \frac{1}{\sqrt{2B}} \left( \sum_{k=0}^{B-1} e^{-2\pi i \frac{p(k+jB)}{2B}} \hat{f}(k+jB) + \sum_{k=B}^{2B-1} e^{-2\pi i \frac{p(k-(j+2)B)}{2B}} \hat{f}(k-(j+2)B) \right) \\ &= \frac{1}{\sqrt{2B}} \left( \sum_{k=0}^{B-1} e^{-2\pi i \frac{pk}{2B}} \hat{f}(k+jB) + \sum_{k=B}^{2B-1} e^{-2\pi i \frac{pk}{2B}} \hat{f}(k-(j+2)B) \right). \end{aligned} \tag{2.5}$$

Similarly, when  $j$  is odd,

$$a_{2Bj+p} = \frac{1}{\sqrt{2B}} \left( \sum_{k=0}^{B-1} e^{-2\pi i \frac{pk}{2B}} \hat{f}(k-(j+1)B) + \sum_{k=B}^{2B-1} e^{-2\pi i \frac{pk}{2B}} \hat{f}(k+(j-1)B) \right). \tag{2.6}$$

This form implies that if we permute the elements of  $\hat{f}$ , then  $a_{2Bj+p}$  can be obtained from an inverse Fourier transform of size  $2B$ . Specifically, let  $S_{G(N,B)}$  be the permutation matrix such that for  $j \in [\frac{N}{2B}]$ ,  $k \in [B]$ , we have

$$\begin{aligned} S_{G(N,B)} |Bj+k\rangle &= \begin{cases} |2Bj+k\rangle & \text{if } j \text{ is even,} \\ |2Bj+B+k\rangle & \text{if } j \text{ is odd,} \end{cases} \\ S_{G(N,B)} |B(\frac{N}{B}-1-j)+k\rangle &= \begin{cases} |2Bj+k\rangle & \text{if } j \text{ is odd,} \\ |2Bj+B+k\rangle & \text{if } j \text{ is even.} \end{cases} \end{aligned} \tag{2.7}$$

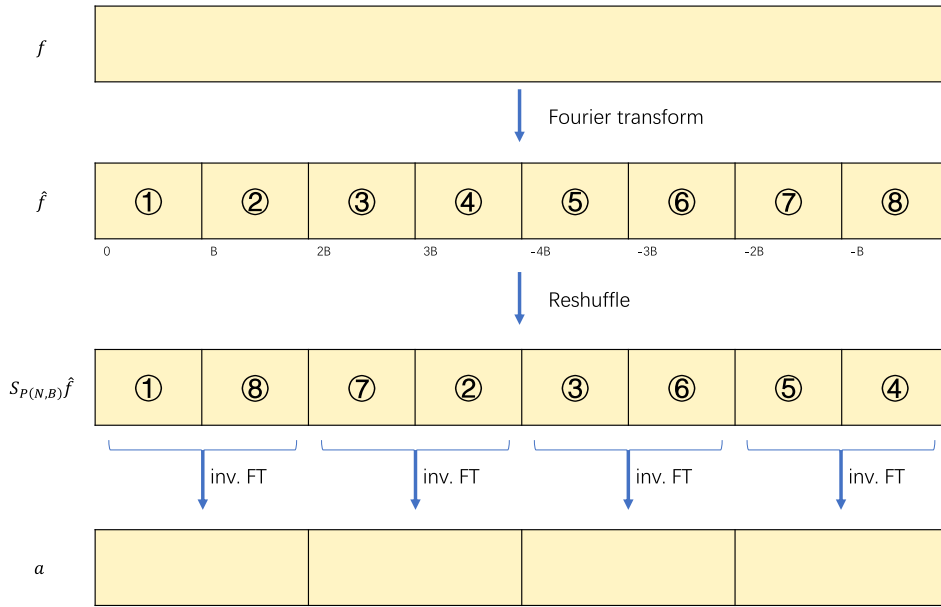


Fig. 4. The process of the sharp Gabor atoms transform. Reshuffle the indices in the frequency domain, and then perform an inverse Fourier transform on each of the  $2B$ -size blocks.

After acting  $S_{G(N,B)}$  on  $\hat{f}$ , we have

$$a_{2Bj+p} = \frac{1}{\sqrt{2B}} \left( \sum_{k=0}^{2B-1} e^{-2\pi i \frac{pk}{2B}} (S_{G(N,B)}\hat{f})(k + 2Bj) \right), \tag{2.8}$$

which is just an inverse Fourier transform of size  $2B$  as desired. This process is shown in Fig. 4.

As an intermediate step of the implementation of  $S_{G(N,B)}$ , we first introduce another permutation matrix. For an integer  $m$  and  $M = 2^m$ , let  $G_{G(M)}$  be the  $M \times M$  permutation matrix such that

$$\begin{aligned} G_{G(M)} |j\rangle &= |2j\rangle, \\ G_{G(M)} |M-1-j\rangle &= |2j+1\rangle, \end{aligned} \tag{2.9}$$

for  $j \in [M/2]$ . Notice that  $G_{G(M)}$  can be represented by the bit manipulation

$$\begin{aligned} |x_{m-1}x_{m-2} \cdots x_0\rangle &\mapsto |x_{m-1}(x_{m-2} \oplus x_{m-1}) \cdots (x_0 \oplus x_{m-1})\rangle \\ &\mapsto |(x_{m-2} \oplus x_{m-1}) \cdots (x_0 \oplus x_{m-1})x_{m-1}\rangle. \end{aligned} \tag{2.10}$$

Therefore, it can be implemented on the quantum computer by  $m-1$  CNOT-gates followed by  $m-1$  SWAP-gates.

Now let us come back to the implementation of matrix  $S_{G(N,B)}$ . One can see that  $S_{G(N,B)}$  only operates on the first  $n-b$  qubits, and can be factorized as

$$S_{G(N,B)} = S_{G(\frac{N}{B},1)} \otimes I_{2^b} = (I_{2^{n-b-2}} \otimes \text{CNOT} \otimes I_{2^b})(G_{G(N/B)} \otimes I_{2^b}).$$

Here, the extra CNOT-gate is used for deciding whether the  $j$  in (2.7) is even or odd. If  $j$  is odd, we need an extra switch between  $|2Bj+k\rangle$  and  $|2Bj+B+k\rangle$  after applying  $G_{G(N/B)}$  on the first  $n-b$  qubits.

Finally, the complete circuit from  $f$  to  $a$  is given by

$$U_{GS} = (I_{2^{n-b-1}} \otimes U_{\text{FT}(2B)}^\dagger)(I_{2^{n-b-2}} \otimes \text{CNOT} \otimes I_{2^b})(G_{G(N/B)} \otimes I_{2^b})U_{\text{FT}(N)}, \tag{2.11}$$

which is demonstrated in Fig. 5.

### 2.2. Blended Gabor atoms

As discussed in Section 1.2, the aim of introducing the blended Gabor atoms is to incorporate blended frequency windows, resulting in rapid decay in the space domain. Typically, a blended Gabor atom basis function consists of two frequency bumps symmetrically positioned about the origin, such as the second and seventh bump in Fig. 6.

To define the discrete blended Gabor atoms, we restrict  $\omega$  onto a grid and represent the index  $(j,p)$  as  $2Bj+p$  for convenience in ordering. The Fourier transform of a discrete blended Gabor atom is defined by

$$\widehat{\psi}_{2Bj+p}(k) = \frac{1}{\sqrt{2B}} e^{2\pi i \frac{pk}{2B}} \left[ e^{\frac{1}{2}\pi i \left( \frac{1}{2} - \frac{k-Bj}{B} \right)} g_{\text{per}} \left( \pi \left( \frac{k-Bj}{B} - \frac{1}{2} \right) \right) + e^{\frac{1}{2}\pi i \left( -\frac{1}{2} - \frac{k+Bj}{B} \right)} g_{\text{per}} \left( \pi \left( \frac{k+Bj}{B} + \frac{1}{2} \right) \right) \right] \tag{2.12}$$

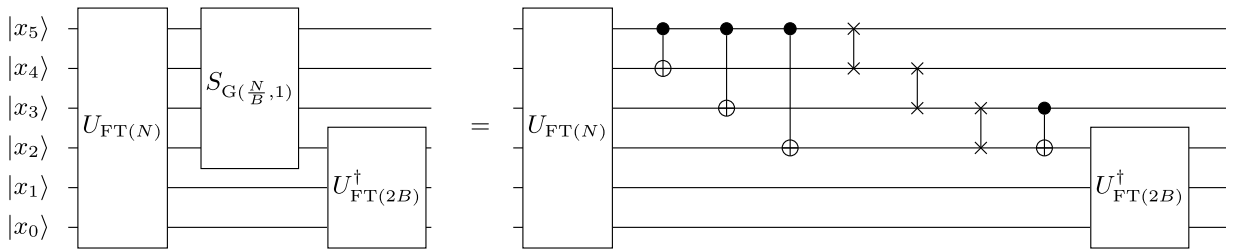


Fig. 5. The circuit of  $U_{GS}$  when  $N = 64$  and  $B = 4$ .

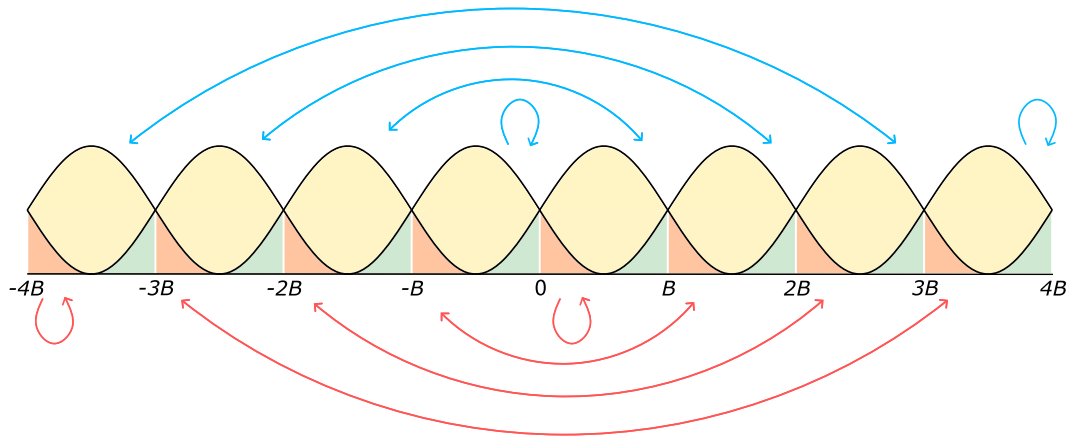


Fig. 6. Illustration of reallocation when  $N = 8B$ . Each basis function  $\widehat{\psi_{2Bj+p}}$  contains two bumps supported at  $[(j - \frac{1}{2})B, (j + \frac{3}{2})B]$  and  $[-(j + \frac{3}{2})B, -(j - \frac{1}{2})B]$ . The arrows indicate how the small tails are transposed such that all the relevant  $\hat{f}(k)$ 's are placed in  $[jB, (j + 1)B] \cup [-(j + 1)B, -jB]$  with correct proportion. All these reallocations are unitary manipulations. This figure exclusively illustrates magnitude transpositions, omitting the change of phase. In particular, the self-transpositions are, in fact, phase rotations.

for  $j \in [\frac{N}{2B}]$ ,  $p \in [2B]$ , where

$$g_{\text{per}}(x) = \sum_{q \in \mathbb{Z}} g(x + q\pi \frac{N}{B}) \tag{2.13}$$

is the periodic version of  $g$  defined in (1.2). Here,  $g_{\text{per}}$  is used instead of  $g$  due to the considerations of boundary conditions. This periodization also automatically makes  $\widehat{\psi_{2Bj+p}}(k) = \widehat{\psi_{2Bj+p}}(N + k)$  hold for any integer  $k$ , which is compatible with our convention.

We also point out that, akin to the sharp Gabor atom, this discrete basis function corresponds to its continuous counterpart through  $\widehat{\psi_{j,p}}(\frac{2\pi}{N}k) = \sqrt{\frac{2\pi}{N}}\widehat{\psi_{j,p}}(k)$ , except for those  $k$  values near the boundary.

One can make straightforward calculations to show that  $\psi_0, \dots, \psi_{N-1}$  form an orthonormal basis of  $\mathbb{C}^N$ . Similar to the sharp case, the coefficients of a vector  $f$  are  $a_{2Bj+p} = \langle \hat{f}, \widehat{\psi_{2Bj+p}} \rangle$ , and the blended Gabor atom transform is defined as

$$U_{GB} : \mathbb{C}^N \rightarrow \mathbb{C}^N : f \mapsto a = (a_0, a_1, \dots, a_{N-1})^T. \tag{2.14}$$

In this scenario, the support of different basis functions may intersect. Therefore, we first reallocate  $\hat{f}$  to the correct position. Subsequently, we perform the rearrangement of indices, identical to the procedure for sharp Gabor atoms. This reallocation process is shown in Fig. 6. To make it mathematically concrete, we introduce the intermediate variable  $h(k)$  defined as

$$h(jB + q) = \hat{f}(jB + q)g\left(-\frac{\pi}{2} + q\frac{\pi}{B}\right)e^{\frac{i}{2}\left(-\frac{\pi}{2} + q\frac{\pi}{B}\right)} + \hat{f}(-jB + q)g\left(\frac{\pi}{2} + q\frac{\pi}{B}\right)e^{\frac{i}{2}\left(\frac{\pi}{2} + q\frac{\pi}{B}\right)}, \tag{2.15}$$

$$h\left(\left(j + \frac{1}{2}\right)B + q\right) = \hat{f}\left(\left(j + \frac{1}{2}\right)B + q\right)g\left(q\frac{\pi}{B}\right)e^{\frac{i}{2}\left(q\frac{\pi}{B}\right)} + \hat{f}\left(-\left(j + \frac{3}{2}\right)B + q\right)g\left(-\pi + q\frac{\pi}{B}\right)e^{\frac{i}{2}\left(-\pi + q\frac{\pi}{B}\right)} \tag{2.16}$$

for  $j = -\frac{N}{2B}, -\frac{N}{2B} + 1, \dots, \frac{N}{2B} - 1$ ,  $q \in [\frac{B}{2}]$ . This reallocation process guarantees that

$$\left( \sum_{k=jB}^{(j+1)B-1} + \sum_{k=-(j+1)B}^{-jB-1} \right) \frac{1}{\sqrt{2B}} e^{-2\pi i \frac{pk}{2B}} h(k) = \langle \hat{f}, \widehat{\psi_{2Bj+p}} \rangle = a_{2Bj+p}. \tag{2.17}$$

The proof is provided in Section 4.1. Despite its complex appearance, the formulas for  $h(k)$  are derived in reverse to satisfy (2.17), so the proof is straightforward calculations.

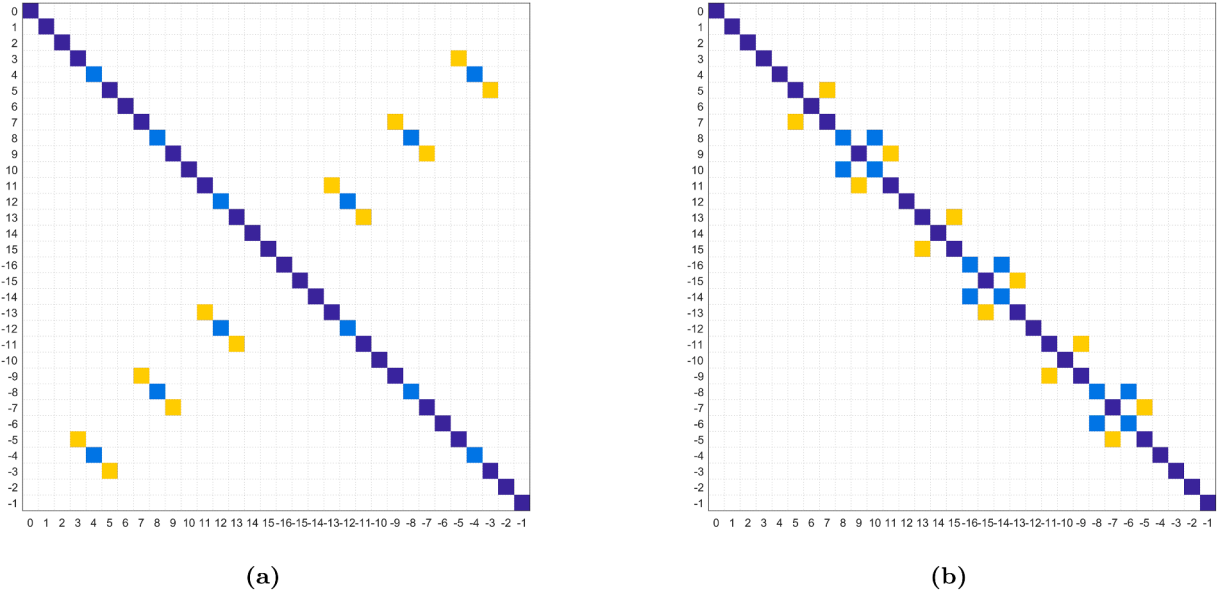


Fig. 7. (a) Magnitude of elements in  $T_G$  with  $N = 32, B = 4$ . (b) Magnitude of elements in  $V_G$  with  $N = 32, B = 4$ .

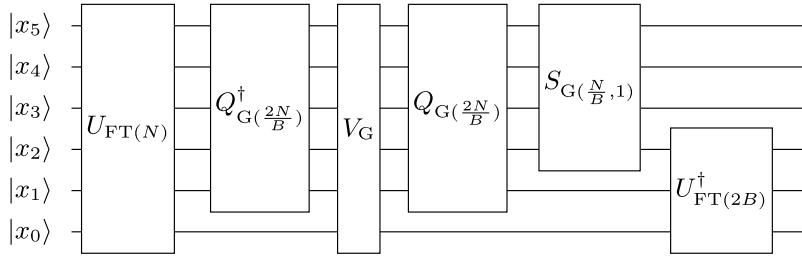


Fig. 8. The circuit of  $U_{GB}$  when  $N = 64$  and  $B = 4$ .

Notice that (2.17) and (2.4) share the same form, so we can reuse the circuit in Section 2.1, and conclude that the complete circuit from blended Gabor atom transform  $U_{GB} : f \mapsto a$  is given by

$$U_{GB} = (I_{2^{n-b-1}} \otimes U_{FT(2B)}^\dagger) S_{G(N,B)} T_G U_{FT(N)}, \tag{2.18}$$

where  $T_G$  is the transition matrix from  $\hat{f}$  to  $h$ , and can be written down following the definitions in (2.15) and (2.16).

Let us now focus on the implementation of  $T_G$ . Since  $T_G$  has the potential block-diagonal structure, as shown in Fig. 7(a), we shall first rearrange the order of the computational basis to reveal this structure. At this stage, we do not need to deal with the last  $b - 1$  qubits. For an integer  $M = 2^m$ , we may define the permutation matrix  $Q_{G(M)}$  as

$$\begin{aligned} Q_{G(M)} |j\rangle &= |2j\rangle, & \text{for } j \in [M/2], \\ Q_{G(M)} |M - 1\rangle &= |M - 1\rangle, \\ Q_{G(M)} |M/2\rangle &= |1\rangle, \\ Q_{G(M)} |M - 2j\rangle &= |4j + 1\rangle, & \text{for } j = 1, 2, \dots, M/4 - 1, \\ Q_{G(M)} |M - 2j - 1\rangle &= |4j - 1\rangle, & \text{for } j = 1, 2, \dots, M/4 - 1. \end{aligned} \tag{2.19}$$

The detailed implementation of  $Q_{G(M)}$  is shown in Section 2.2.1. We can use the permutation matrix  $Q_{G(\frac{2N}{B})} \otimes I_{2^{b-1}}$  to rearrange the nonzero elements in  $T_G$  and define a new matrix

$$V_G = (Q_{G(\frac{2N}{B})} \otimes I_{2^{b-1}})^\dagger T_G (Q_{G(\frac{2N}{B})} \otimes I_{2^{b-1}}), \tag{2.20}$$

which is a block-diagonal matrix as shown in Fig. 7(b). A detailed description of  $V_G$  and its implementation is discussed in Section 2.2.2. Combining (2.18) and (2.20), we can draw the complete circuit of  $U_{GB}$  in Fig. 8.

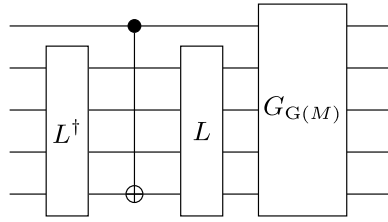


Fig. 9. The circuit of  $Q_{G(M)}$  for  $M = 32$ .

2.2.1. The implementation of  $Q_{G(M)}$

Let  $L$  be the  $M \times M$  shift matrix

$$L = \begin{bmatrix} 0 & 0 & \dots & \dots & 1 \\ 1 & 0 & 0 & \ddots & 0 \\ \vdots & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \tag{2.21}$$

which can be implemented easily within  $O(m^2)$  gate complexity, as shown in [33, Figure 6]. If one is allowed to use one or more ancilla qubits, then  $L$  can be implemented within  $O(m)$  gate complexity as in [28,34].

Notice that  $Q_{G(M)}$  has the same  $M/2$  columns as the matrix  $G_{G(M)}$  introduced earlier, so it is natural to reuse the circuit of  $G_{G(M)}$  in the construction of  $Q_{G(M)}$ . Straightforward calculations show that  $G_{G(M)}^{-1}Q_{G(M)}$  is a permutation matrix that switches  $|M - 2j\rangle$  and  $|M - 2j - 1\rangle$  for  $j = 1, 2, \dots, \frac{M}{4} - 1$ , and also switches  $|M - 1\rangle$  and  $|M/2\rangle$ . Therefore, we have  $G_{G(M)}^{-1}Q_{G(M)} = |0\rangle\langle 0| \otimes I_{M/2} + |1\rangle\langle 1| \otimes I_{M/4} \otimes X(I_2 \otimes L)^\dagger$ . Finally, noticing that the controls on  $L$  and  $L^\dagger$  are unnecessary since they will be canceled, we conclude

$$Q_{G(M)} = G_{G(M)}(I_2 \otimes L)(|0\rangle\langle 0| \otimes I_{M/2} + |1\rangle\langle 1| \otimes I_{M/4} \otimes X)(I_2 \otimes L)^\dagger, \tag{2.22}$$

where  $(|0\rangle\langle 0| \otimes I_{M/2} + |1\rangle\langle 1| \otimes I_{M/4} \otimes X)$  is just the CNOT gate on the first and the last qubits. Fig. 9 illustrates the circuit implementation of  $Q_{G(M)}$ .

2.2.2. Implementation of  $V_G$

To help the following discussion, we define two diagonal matrices  $D_+ = \text{diag}\{0, \frac{1}{B}, \frac{2}{B}, \dots, \frac{(B/2-1)}{B}\}$  and  $D_- = D_+ - \frac{1}{2}I$ .

As illustrated in Fig. 7(b), the matrix  $V_G$  is a block-diagonal matrix with  $\frac{N}{B}$  blocks, with each block of size  $B \times B$ . By the definitions in (2.15) and (2.16) together with the permutation matrix  $Q_{G(M)}$ , we can write down the blocks explicitly.

$$\begin{aligned} \text{The 0th block} &= \hat{K}_e := I_2 \otimes e^{\frac{i}{2}(\pi\beta(D_-)+\pi D_-)}. \\ \text{The } (2j)\text{th block} &= K_e := \begin{bmatrix} \cos(\frac{\pi}{2}\beta(D_-))e^{\frac{i}{2}\pi D_-} & i \sin(\frac{\pi}{2}\beta(D_-))e^{\frac{i}{2}\pi D_-} \\ i \sin(\frac{\pi}{2}\beta(D_-))e^{\frac{i}{2}\pi D_-} & \cos(\frac{\pi}{2}\beta(D_-))e^{\frac{i}{2}\pi D_-} \end{bmatrix} \\ &= (H \otimes I_{B/2}) \begin{bmatrix} e^{\frac{i}{2}(\pi\beta(D_-)+\pi D_-)} & 0 \\ 0 & e^{\frac{i}{2}(-\pi\beta(D_-)+\pi D_-)} \end{bmatrix} (H \otimes I_{B/2}), \\ &\text{for } j = 1, 2, \dots, \frac{N}{2B} - 1. \\ \text{The } (2j - 1)\text{th block} &= K_o := \begin{bmatrix} \cos(\frac{\pi}{2}\beta(D_+))e^{\frac{i}{2}\pi D_+} & -i \sin(\frac{\pi}{2}\beta(D_+))e^{\frac{i}{2}\pi D_+} \\ -i \sin(\frac{\pi}{2}\beta(D_+))e^{\frac{i}{2}\pi D_+} & \cos(\frac{\pi}{2}\beta(D_+))e^{\frac{i}{2}\pi D_+} \end{bmatrix} \\ &= (H \otimes I_{B/2}) \begin{bmatrix} e^{\frac{i}{2}(-\pi\beta(D_+)+\pi D_+)} & 0 \\ 0 & e^{\frac{i}{2}(\pi\beta(D_+)+\pi D_+)} \end{bmatrix} (H \otimes I_{B/2}), \\ &\text{for } j = 1, 2, \dots, \frac{N}{2B} - 1. \\ \text{The } (\frac{N}{B} - 1)\text{th block} &= \hat{K}_o := I_2 \otimes e^{\frac{i}{2}(-\pi\beta(D_+)+\pi D_+)}. \end{aligned} \tag{2.23}$$

The remaining challenge lies in implementing these diagonal matrices. Usually,  $\beta(x)$  will be chosen as a polynomial on  $[0, \frac{1}{2}]$  and then extend to  $[-1, 1]$  according to (1.3). Typical choices include  $\beta(x) = x$  or  $\beta(x) = 2x^2$  in the interval  $[0, \frac{1}{2}]$ . For such low-degree  $\beta$ , one can use the method introduced in Section 2.2.3 to implement the required diagonal matrices. If one wants to pursue higher regularity with more sophisticated  $\beta(x)$ , such as

$$\beta(x) = x^4(35 - 84x + 70x^2 - 20x^3), \quad \text{for } x \in [0, 1], \tag{2.24}$$

or even non-polynomial functions, the method in Section 2.2.4 could be used.

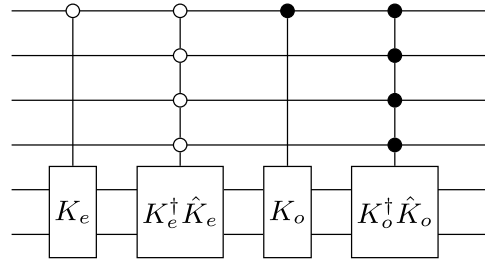


Fig. 10. The circuit of  $V_G$  when  $N = 64$  and  $B = 4$ .

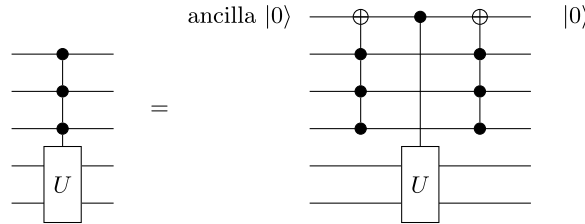


Fig. 11. The circuit of implementing multi-qubit controlled unitaries using one ancilla qubit.

Finally, the circuit of  $V_G$  is presented in Fig. 10. Here, we need to implement the multi-qubit controlled version of a unitary. A naive way is to do the multi-qubit controlled version of every gate in this unitary in order. However, this approach results in very high gate complexity. Instead, we may introduce an ancilla qubit to make the implementation using mainly single qubit controls, as demonstrated in Fig. 11.

2.2.3. Implementation of the exponential of a pure imaginary polynomial via bit manipulations

Now we focus on the general problem of implementing diagonal matrices  $\exp(iq(A))$ , where  $A = \text{diag}\{0, 1, \dots, M - 1\}$  for some  $M = 2^m$ , and  $q$  is a real polynomial. Since polynomials are the sum of monomials, we only need to consider the special case  $q(A) = rA^s$ , where  $r \in \mathbb{R}$  and  $s \geq 0$  is an integer. The case  $s = 0$  is trivial. The case  $s = 1$  is discussed in [35, Section 5.1], which exploits the following observation. Let  $x \in [M]$  and  $x = (x_{m-1} \dots x_0)_2$  be its binary representation, then

$$\exp(irA) |x\rangle = \exp(irx) |x\rangle = \bigotimes_{j=0}^{m-1} (\exp(irx_j 2^j) |x_j\rangle) = \bigotimes_{j=0}^{m-1} (R_z(2^j r) |x_j\rangle). \tag{2.25}$$

Where  $R_z(\theta) := \begin{pmatrix} 1 & \\ & e^{i\theta} \end{pmatrix}$  is the single-qubit rotation gate. This means  $\exp(irA)$  can be implemented by simply applying an  $R_z(2^j r)$  on the  $j$ th qubit.

For general  $s$ , a similar approach can be utilized, but it may involve multi-qubit gates due to the cross terms. Specifically, we have

$$\begin{aligned} x^s &= \left( \sum_{j=0}^{m-1} 2^j x_j \right)^s = \sum_{k_0+k_1+\dots+k_{m-1}=s; k_j \geq 0} \binom{n}{k_0, k_1, \dots, k_{m-1}} \prod_{j=0}^{m-1} (2^{jk_j} x_j^{k_j}) \\ &= \sum_{k_0+k_1+\dots+k_{m-1}=s; k_j \geq 0} \binom{n}{k_0, k_1, \dots, k_{m-1}} \left( \prod_{j=0}^{m-1} 2^{jk_j} \right) \prod_{k_j > 0} x_j. \end{aligned} \tag{2.26}$$

Therefore, to implement  $\exp(irx^s)$ , we only need to implement the terms that look like  $\exp(ic \prod_{j \in J} x_j)$ , where  $J$  is an index set, and  $c$  is a real number. This  $2^{|J|}$  dimensional matrix can be realized as a controlled version of  $R_z(c)$ , which is

$$\text{diag}\{1, 1, \dots, 1, e^{ic}\} = |1^{|J|-1}\rangle \langle 1^{|J|-1}| \otimes R_z(c) + (I - |1^{|J|-1}\rangle \langle 1^{|J|-1}|) \otimes I. \tag{2.27}$$

Note that it does not matter which  $|J| - 1$  qubits serve as control qubits since this matrix will not change under different choices.

To demonstrate this clearly, we give an example for  $m = 3, s = 3, r = 1$ . In this case,

$$\exp(ix^3) |x\rangle = \exp(i(x_0 + 8x_1 + 64x_2 + 18x_0x_1 + 60x_0x_2 + 144x_1x_2 + 48x_0x_1x_2)) |x_2x_1x_0\rangle, \tag{2.28}$$

which can be implemented as Fig. 12.

This method's advantage is that it does not use any ancilla qubit and does not introduce any approximation, so it is desirable as long as we have a moderate value of  $s$ .

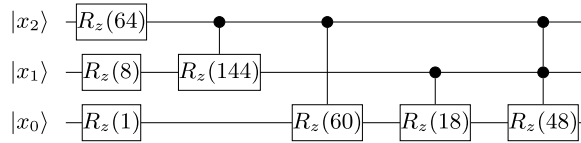


Fig. 12. The implementation of the unitary  $\sum_{x=0}^7 \exp(ix^3) |x\rangle\langle x|$ .

2.2.4. Implementation of the exponential of a pure imaginary polynomial via QSVT

The complexity of the implementation given in Section 2.2.3 is  $O(m^s)$ , where  $m$  is the number of qubits, and  $s$  is the degree of the polynomial  $q$ . This complexity is unaffordable when the degree  $s$  is large.

For polynomial  $q$  of higher degrees, one may resort to QSVT [36] for better asymptotic complexity scaling at the expense of introducing approximations and extra ancilla qubits. To be specific, we shall construct a  $(\sqrt{2}, 3, \epsilon)$ -block encoding of the desired unitary matrix using QSVT, where  $\epsilon$  is the tolerable error, and then use the “perfect amplitude amplification” [23] to extract the desired matrix. (One may refer to Section 1.3 for the terminology regarding block encoding.)

Recall that  $\beta$  is a polynomial on  $[-1, 1]$  satisfying (1.3), and a typical example is (2.24). Therefore,  $\gamma(x) := 2\beta(x + \frac{1}{2}) - 1$  is an odd polynomial. The diagonal matrices that we need to implement in (2.23) are  $e^{\frac{i}{2}(\pi\beta(D_-)+\pi D_-)}$ ,  $e^{\frac{i}{2}(-\pi\beta(D_-)+\pi D_-)}$ ,  $e^{\frac{i}{2}(-\pi\beta(D_+)+\pi D_+)}$ , and  $e^{\frac{i}{2}(\pi\beta(D_+)+\pi D_+)}$ .

Now we consider  $e^{\frac{i}{2}(\pi\beta(D_-)+\pi D_-)}$  as an example since the implementations for the other three matrices are analogous. In this case, the number of qubits is  $m = b - 1 = \log_2 B - 1$ . Noticing that

$$e^{\frac{i}{2}\pi(\beta(D_-)+D_-)} = e^{\frac{i}{2}\pi(-\beta(-D_-)+D_-)} = e^{\frac{i}{2}\pi(-\frac{1}{2}(\gamma(-D_- - \frac{1}{2})+1)+D_-)} = e^{-\frac{i\pi}{4}\gamma(-D_- - \frac{1}{2})} e^{-\frac{i\pi}{4} + \frac{i\pi}{2} D_-},$$

where  $e^{-\frac{i\pi}{4} + \frac{i\pi}{2} D_-}$  can be easily implemented using the method in Section 2.2.3, we focus on

$$e^{-\frac{i\pi}{4}\gamma(W)} = \cos(\frac{\pi}{4}\gamma(W)) - i \sin(\frac{\pi}{4}\gamma(W)),$$

where  $W := -D_- - \frac{1}{2}$  for the simplicity of notations. We need to first implement the  $(1, 2, \epsilon)$ -block encodings of  $\cos(\frac{\pi}{4}\gamma(W))$  and  $\sin(\frac{\pi}{4}\gamma(W))$ , and then use LCU [30, Section 7.3] to get a  $(\sqrt{2}, 3, \epsilon)$ -block encoding of  $e^{-\frac{i\pi}{4}\gamma(W)}$ . Below, we explain this idea in more detail.

For the  $(1, 2, \epsilon)$ -block encoding of  $\cos(\frac{\pi}{4}\gamma(W))$ , we outline the procedure following the methodology detailed in [35, Section 5.1]. First, we can use the method in Section 2.2.3 to construct the  $(1, 1, 0)$ -block encoding of  $\sin(W)$ , given by

$$\begin{bmatrix} \sin(W) & -i \cos(W) \\ -i \cos(W) & \sin(W) \end{bmatrix} = (H \otimes I) \begin{bmatrix} -iI & 0 \\ 0 & iI \end{bmatrix} \begin{bmatrix} e^{iW} & 0 \\ 0 & e^{-iW} \end{bmatrix} (H \otimes I).$$

As  $-\frac{1}{2}I \leq W \leq \frac{1}{2}I$ , we seek an  $\epsilon$  accurate polynomial approximation of the even function  $\cos(\frac{\pi}{4}(\gamma(\arcsin(x))))$  for  $x \in [-\sin(\frac{1}{2}), \sin(\frac{1}{2})]$  to utilize QSVT on  $\sin(W)$  to obtain a  $(1, 2, \epsilon)$ -block encoding of  $\cos(\frac{\pi}{4}\gamma(W))$ . Let  $m_\gamma = \max_{-\frac{1}{2} \leq x \leq \frac{1}{2}} |\gamma'(x)|$ . By truncating the Taylor series of  $\arcsin(x)$  at  $x = 0$  to degree  $O(\log(\frac{2m_\gamma}{\epsilon}))$ , we get a polynomial  $w(x)$  satisfying  $|w(x) - \arcsin(x)| \leq \frac{\epsilon}{2m_\gamma}$  for  $|x| \leq \sin(\frac{1}{2})$ . Consequently,  $|\gamma(w(x)) - \gamma(\arcsin(x))| \leq \frac{\epsilon}{2}$ . We can also expand  $\cos(\frac{\pi}{4}y)$  as Taylor series at  $y = 0$  and truncate at degree  $O(\log \frac{1}{\epsilon})$  to obtain a polynomial  $r(y)$  such that  $|r(y) - \cos(\frac{\pi}{4}y)| \leq \frac{\epsilon}{2}$  for  $|y| \leq 1$ . Given that  $|\gamma(x)| \leq 1$  for  $x \in [-\frac{1}{2}, \frac{1}{2}]$ , we finally derive

$$\begin{aligned} & \left| r(\gamma(w(x))) - \cos(\frac{\pi}{4}(\gamma(\arcsin(x)))) \right| \\ & \leq \left| r(\gamma(w(x))) - \cos(\frac{\pi}{4}\gamma(w(x))) \right| + \left| \cos(\frac{\pi}{4}\gamma(w(x))) - \cos(\frac{\pi}{4}(\gamma(\arcsin(x)))) \right| \\ & \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} \max_y \left| \frac{d \cos(\frac{\pi}{4}y)}{dy} \right| \leq \epsilon. \end{aligned}$$

Therefore, this polynomial  $r(\gamma(w(x)))$  of degree  $O(\log(\frac{1}{\epsilon}) \log(\frac{2m_\gamma}{\epsilon}) \deg(\gamma))$  is an  $\epsilon$  accurate polynomial approximation of the even function  $\cos(\frac{\pi}{4}(\gamma(\arcsin(x))))$ . After this polynomial is constructed, we can find the corresponding phase factors and implement QSVT, which we refer to [37–39] for details. Typically,  $\gamma$  is a smooth function of moderate degree, so we may regard  $m_\gamma$  and  $\deg \gamma$  as  $O(1)$  and conclude that the total cost of implementing a  $(1, 2, \epsilon)$ -block encoding of  $\cos(\frac{\pi}{4}\gamma(W))$  is  $O(b \log(\frac{1}{\epsilon})^2)$ , taking the encoding cost of  $\sin(W)$  into account. The  $(1, 2, \epsilon)$ -block encoding of  $\sin(\frac{\pi}{4}\gamma(W))$  follows the same routine, and the only difference is one shall approximate the odd function  $\sin(\frac{\pi}{4}(\gamma(\arcsin(x))))$ . Using LCU, we can implement a  $(\sqrt{2}, 3, \epsilon)$ -block encoding of  $e^{\frac{i}{2}(\pi\beta(D_-)+\pi D_-)}$  within  $O(b \log(\frac{1}{\epsilon})^2)$  cost. When measuring the ancilla qubits, we get the desired matrix for probability  $\frac{1}{2}$ . A naive way to extract the desired unitary matrix is to repeat the whole process until success. However, a better way is to use the “perfect amplitude amplification” [23] to boost the success probability to 1 deterministically with a constant overhead. Therefore, we conclude that we can implement  $e^{\frac{i}{2}(\pi\beta(D_-)+\pi D_-)}$  within  $O(b \log(\frac{1}{\epsilon})^2) = O(n \log(\frac{1}{\epsilon})^2)$  cost.

### 2.3. Complexity analysis

The circuit complexity of our Gabor atom implementation is analyzed below, assuming single-qubit and two-qubit gates as elementary building blocks. The circuit of the sharp Gabor atoms, shown in Fig. 5, is  $O(n^2)$ , comprising  $O(n)$  operations flanked by two Fourier transform circuits, each with  $O(n^2)$  complexity. For blended Gabor atoms, the complexity remains at most  $O(\text{poly}(n))$ , even if we always use the exact implementations and do not use any ancilla qubit as in Fig. 11. For instance, selecting  $\beta(x)$  as  $x$  or  $2x^2$  for  $0 \leq x \leq \frac{1}{2}$  and employing the method in Section 2.2.3 for diagonal matrix implementation still yields  $O(n^2)$  complexity, since we can check that each block in Fig. 8 costs no more than  $O(n^2)$ .

If we allow for an  $\epsilon$  error and use three ancilla qubits, then the overall complexity is only  $O(n(\log n + \log(\epsilon^{-1})^2))$ . A step-by-step examination of Fig. 8 verifies this. The Fourier transform can be performed using an  $O(n \log(n\epsilon^{-1}))$  algorithm [40].  $Q_{G(M)}$  has an  $O(n)$  complexity since  $G_{G(M)}$  is  $O(n)$  and  $L$  can also be implemented within  $O(n)$  complexity using one ancilla qubit [28]. In the implementation of  $V_G$ , the multiple qubit controls can be implemented as in Fig. 11, and the required diagonal matrices can be implemented within  $O(n \log(\epsilon^{-1})^2)$  for general  $\beta(x)$  using three ancilla qubits as discussed in Section 2.2.4.

## 3. Wavelets

This section gives the discretization of the (sharp) Shannon wavelets and (blended) Meyer wavelets and presents the corresponding quantum circuits. The discretization of the Shannon wavelet is straightforward, and its circuit implementation can be viewed as a recursive process. For the Meyer wavelets, a phase adjustment is necessary to ensure orthogonality when imposing periodic boundary conditions. When implementing its quantum circuit, we first reallocate the Fourier coefficients  $\hat{f}(k)$  into the correct positions and then apply the circuit for the Shannon wavelet, akin to our approach with Gabor atoms. However, the reallocation process involves multiple levels of control, and sometimes the control occurs at  $\frac{N}{3 \cdot 2^j}$  rather than powers of 2, which makes it more technical.

### 3.1. Shannon wavelets (sharp)

In the continuous setting, the Shannon wavelet basis  $\widehat{\psi}_{j,p}$  is given by (1.5) and (1.6). The Fourier transforms of discrete Shannon wavelets are given by

$$\widehat{\psi}_{j,p}(k) = \frac{1}{\sqrt{2^{n-j}}} e^{2\pi i \frac{pk}{2^{n-j}}} [\chi_{[-2^{n-j}, -2^{n-j-1}]}(k) + \chi_{[2^{n-j-1}, 2^{n-j}]}(k)], \tag{3.1}$$

for  $j = 1, 2, \dots, n$  and  $p \in [2^{n-j}]$ . In particular, when  $j = n$  and  $p = 0$ ,  $\widehat{\psi}_{j,p} = \chi_{\{-1\}}$ . In addition, the last basis functions are given by the scaling function

$$\widehat{\phi}_{n,0} = \chi_{\{0\}}. \tag{3.2}$$

We remark that, similar to the sharp Gabor atom case, it holds that  $\widehat{\psi}_{j,p}(k) = \sqrt{\frac{2\pi}{N}} \widehat{\psi}_{j,p}(\frac{2\pi}{N}k)$  for  $k = -N/2, \dots, N/2 - 1$ .

All the  $\widehat{\psi}_{j,p}$ 's together with  $\widehat{\phi}_{n,0}$  form an orthonormal basis of  $\mathbb{C}^N$  since the supports of  $\widehat{\psi}_{j,p}$  at different levels  $j$  are disjoint. Some version of Shannon wavelets may have extra phases multiplied on these  $\chi_{[-2^{n-j}, -2^{n-j-1}]}(k)$  and  $\chi_{[2^{n-j-1}, 2^{n-j}]}(k)$  [1, Chapter 7.2], but we do not adopt them here.

By orthogonality, the wavelet coefficients of the signal  $f$  are

$$a_{j,p} = \langle f, \psi_{j,p} \rangle = \langle \hat{f}, \widehat{\psi}_{j,p} \rangle = \left( \sum_{k=2^{n-j-1}}^{2^{n-j}-1} + \sum_{k=-2^{n-j}}^{-2^{n-j}-1} \right) \frac{1}{\sqrt{2^{n-j}}} e^{-2\pi i \frac{pk}{2^{n-j}}} \hat{f}(k). \tag{3.3}$$

and

$$a_{n+1,0} = \langle f, \phi_{n,0} \rangle = \langle \hat{f}, \widehat{\phi}_{n,0} \rangle = \hat{f}(0). \tag{3.4}$$

The Shannon wavelet transform is defined as

$$U_{WS} : \mathbb{C}^N \rightarrow \mathbb{C}^N : f \mapsto a = (a_{1,0}, \dots, a_{1,N/2-1}, a_{2,0}, \dots, a_{2,N/4-2}, \dots, a_{n-1,0}, a_{n-1,1}, a_{n,0}, a_{n+1,0})^T. \tag{3.5}$$

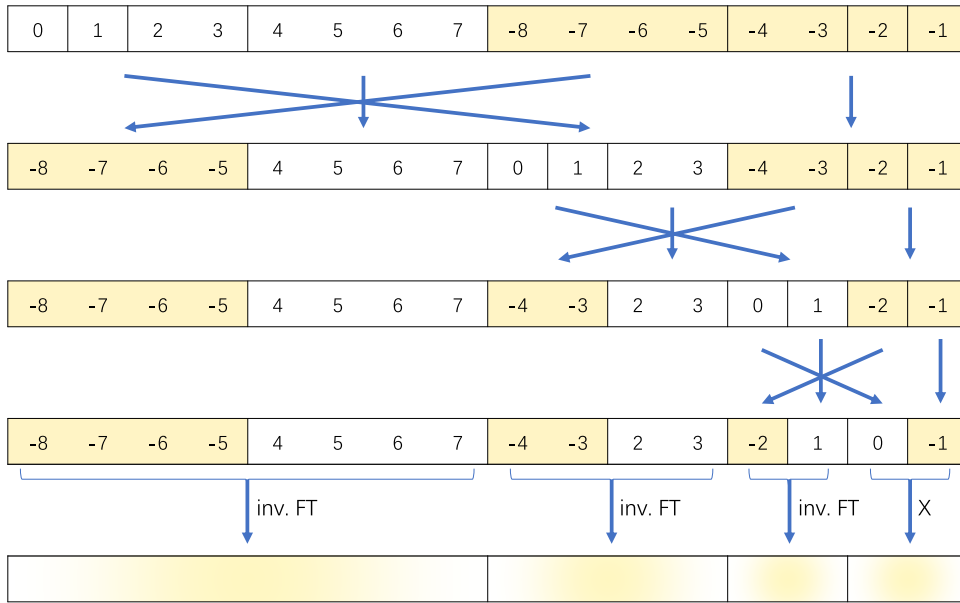
Since the calculation (3.3) first involves a Fourier transform of  $f$ , we introduce

$$S_{W(N)} : \mathbb{C}^N \rightarrow \mathbb{C}^N : \hat{f} \mapsto a, \tag{3.6}$$

which satisfies  $U_{WS} = S_{W(N)} U_{FT(N)}$ . Notice that when  $N = 2$ ,  $S_{W(2)} = X$  is the Pauli X matrix.

Consider the rearrangement of the index of  $\hat{f}$  as follows:

$$\begin{aligned} G_{W(N)} : & (\hat{f}(0), \dots, \hat{f}(N/2 - 1), \hat{f}(-N/2), \dots, \hat{f}(-1))^T \\ \mapsto & (\hat{f}(-N/2), \dots, \hat{f}(-N/4 - 1), \hat{f}(N/4), \dots, \hat{f}(N/2 - 1), \hat{f}(0), \dots, \hat{f}(N/4 - 1), \hat{f}(-N/4), \dots, \hat{f}(-1))^T, \end{aligned}$$



**Fig. 13.** The illustration of  $S_{W(K)}$ . It contains a reshuffling process, followed by inverse Fourier transforms. The inverse Fourier transforms can also be performed right after the corresponding reshuffling process.

which is the transposition of the first and the third quarter of the indices. The matrix form of  $G_{W(N)}$  can be written explicitly as  $G_{W(N)} = P_{1,3} \otimes I_{N/4}$ , where

$$P_{1,3} := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \text{---} \oplus \text{---}$$

is just a  $|0\rangle$  controlled-NOT gate. After this rearrangement, when  $j = 1$ , (3.3) becomes

$$a_{1,p} = \sum_{k=0}^{2^{n-1}-1} \frac{1}{\sqrt{2^{n-1}}} e^{-2\pi i \frac{pk}{2^{n-1}}} (G_{W(N)} \hat{f})(k), \tag{3.7}$$

which is an inverse Fourier transform of size  $\frac{N}{2}$  on the first half of  $G_{W(N)} \hat{f}$ .

Moreover, the latter half of  $G_{W(N)} \hat{f}$  is exactly the middle half of  $\hat{f}$ , i.e., the second and third quarters of  $\hat{f}$ . By definition (3.3), we notice that the second half of the coefficient vector  $a$ , denoted as

$$a(N/2 : N - 1) = (a_{2,0}, \dots, a_{2,N/4-2}, \dots, a_{n-1,0}, a_{n-1,1}, a_{n,0}, a_{n,1})^T, \tag{3.8}$$

only depends on  $(G_{W(N)} \hat{f})(N/2 : N - 1) = \hat{f}(-N/4 : N/4 - 1)$ , and this dependence follows the same pattern as (3.3) except for  $N$  is replaced by  $N/2$ . More specifically,

$$S_{W(N/2)}(G_{W(N)} \hat{f})(N/2 : N - 1) = a(N/2 : N - 1). \tag{3.9}$$

Therefore,  $S_{W(N)}$  can be implemented recursively, as shown in Fig. 13, and we have

$$S_{W(K)} = \left( |0\rangle\langle 0| \otimes U_{FT(K/2)}^\dagger + |1\rangle\langle 1| \otimes S_{W(K/2)} \right) G_{W(K)} \tag{3.10}$$

for  $K = N, N/2, N/4, \dots, 4$ , with the base case  $S_{W(2)} = X$  being the Pauli X matrix. This can be implemented in the quantum circuit as Fig. 14.

As an example, Fig. 15 shows the fully expanded circuit for  $U_{WS}$  when  $N = 16$ . Its recursive structure appears in the sub-circuits for  $S_{W(8)}$  (red box) and  $S_{W(4)}$  (blue box). This circuit also contains multi-qubit controlled unitary matrices, which can be implemented as in Fig. 11.

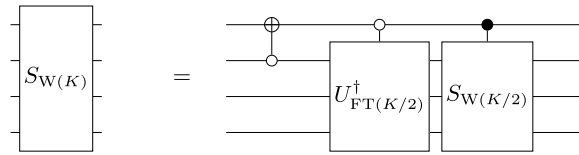


Fig. 14. The recursive definition of  $S_{W(K)}$ . The base case is  $S_{W(2)} = X$ .

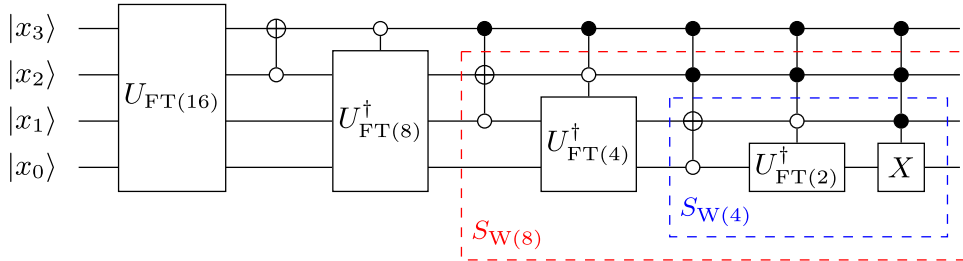


Fig. 15. The fully expanded circuit of  $U_{WS}$  when  $N = 16$ . Here, for multiple qubit control gates, the controlling qubits may be located both on the upper wires and lower wires of the controlled qubit.

### 3.2. Meyer wavelets (blended)

In the continuous setting, the Meyer wavelets are given by (1.7). In order to impose a periodic boundary condition here and maintain the orthogonality, we need to introduce a phase shift and define

$$\hat{\psi}_{ms}(\omega) = \begin{cases} e^{i\frac{\pi}{4} - i\frac{\omega}{2}} g\left(\frac{3\omega}{2} - 2\pi\right) & \text{if } \frac{2\pi}{3} \leq \omega \leq \frac{4\pi}{3}, \\ e^{i\frac{\pi}{4} - i\frac{\omega}{2}} g\left(\frac{3\omega}{4} - \pi\right) & \text{if } \frac{4\pi}{3} \leq \omega \leq \frac{8\pi}{3}, \\ 0 & \text{if } 0 \leq \omega \leq \frac{2\pi}{3} \text{ or } \omega \geq \frac{8\pi}{3}, \\ \hat{\psi}_{ms}(-\omega)^* & \text{if } \omega < 0. \end{cases} \tag{3.11}$$

Now we can define the discrete Meyer wavelet base functions  $\psi_{j,p}$  by assigning their Fourier coefficients as

$$\widehat{\psi}_{j,p}(k) = \frac{1}{\sqrt{2^{n-j}}} e^{2\pi i \frac{pk}{2^{n-j}}} \sum_{q \in \mathbb{Z}} \hat{\psi}_{ms}\left(2^{j+1}\pi\left(\frac{k}{N} + q\right)\right), \quad \left(k = -\frac{N}{2}, -\frac{N}{2} + 1, \dots, \frac{N}{2} - 1\right), \tag{3.12}$$

for  $j = 1, 2, \dots, n$  and  $p \in [2^{n-j}]$ . One can check that these  $N - 1$  functions, together with the scaling function

$$\widehat{\phi}_{n,0} = \chi_{\{0\}} \tag{3.13}$$

form an orthonormal basis of  $\mathbb{C}^N$ . While the orthogonality can be calculated directly, it can also be seen as a corollary of the following discussions, in which we will illustrate that the expansion under this basis is unitary. When  $k = -\frac{N}{2}, -\frac{N}{2} + 1, \dots, \frac{N}{2} - 1$ , (3.12) can also be formulated as

$$\widehat{\psi}_{j,p}(k) = \begin{cases} \frac{1}{\sqrt{2^{n-j}}} e^{2\pi i \frac{pk}{2^{n-j}}} \sum_{q=-1}^1 \hat{\psi}_{ms}\left(2^{j+1}\pi\left(\frac{k}{N} + q\right)\right) & \text{if } j = 1, \\ \frac{1}{\sqrt{2^{n-j}}} e^{2\pi i \frac{pk}{2^{n-j}}} \hat{\psi}_{ms}\left(2^{j+1}\pi\frac{k}{N}\right) & \text{if } j = 2, 3, \dots, n. \end{cases} \tag{3.14}$$

We also point out that if we ignore the range of  $k$  in (3.12), it still holds  $\widehat{\psi}_{j,p}(k) = \widehat{\psi}_{j,p}(k + N)$ , which is compatible with our convention. Our objective is to calculate the wavelet coefficients

$$a_{j,p} = \langle f, \psi_{j,p} \rangle = \langle \hat{f}, \widehat{\psi}_{j,p} \rangle \tag{3.15}$$

and

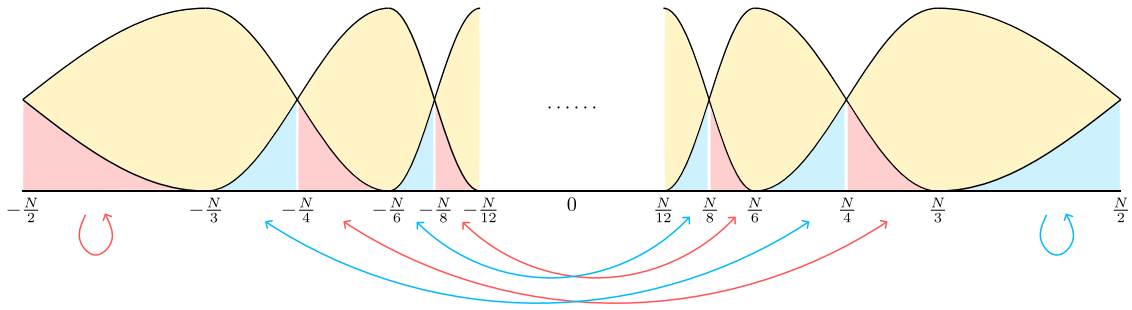
$$a_{n+1,0} := \langle f, \phi_{n,0} \rangle = \langle \hat{f}, \widehat{\phi}_{n,0} \rangle = \hat{f}(0). \tag{3.16}$$

Then the wavelet transform in a matrix form is

$$U_{WB} : \mathbb{C}^N \rightarrow \mathbb{C}^N : f \mapsto a = (a_{1,0}, \dots, a_{1,N/2-1}, a_{2,0}, \dots, a_{2,N/4-2}, \dots, a_{n-1,0}, a_{n-1,1}, a_{n,0}, a_{n+1,0})^T. \tag{3.17}$$

Similar to the blended Gabor atom case, the supports of basis functions at adjacent levels overlap. Therefore, we reallocate a portion of each  $\hat{f}(k)$ , which is illustrated in Fig. 16. To write down this reallocation process precisely, we introduce an intermediate variable  $h \in \mathbb{C}^N$ , which is given by

$$h\left(\frac{N}{2j} - q\right) = -e^{-i\frac{\pi}{4} - i\frac{\pi q 2^j}{N}} g\left(\frac{\pi}{2} - \frac{3\pi q 2^{j-1}}{N}\right) \hat{f}\left(\frac{N}{2j} - q\right) - e^{i\frac{\pi}{4} - i\frac{\pi q 2^j}{N}} g\left(-\frac{\pi}{2} - \frac{3\pi q 2^{j-1}}{N}\right) \hat{f}\left(-\frac{N}{2j} - q\right), \tag{3.18}$$



**Fig. 16.** This figure demonstrates the reallocations of  $\hat{f}(k)$  in the first few levels.  $\widehat{\psi}_{j,p}$  is shown as two bumps supported at  $[-\frac{2^{n-j+2}}{3}, -\frac{2^{n-j}}{3})$  and  $[\frac{2^{n-j}}{3}, \frac{2^{n-j+2}}{3})$ . After the transpositions, the relevant  $\hat{f}(k)$ 's would be placed at  $[-2^{n-j}, -2^{n-j-1}) \cup [2^{n-j-1}, 2^{n-j})$  with correct proportions.

$$h(-\frac{N}{2j} - q) = e^{i\frac{\pi}{4} - i\frac{\pi q 2^{j-1}}{N}} g(-\frac{\pi}{2} - \frac{3\pi q 2^{j-1}}{N}) \hat{f}(\frac{N}{2j} - q) + e^{-i\frac{\pi}{4} - i\frac{\pi q 2^{j-1}}{N}} g(\frac{\pi}{2} - \frac{3\pi q 2^{j-1}}{N}) \hat{f}(-\frac{N}{2j} - q), \tag{3.19}$$

for  $j = 2, 3, \dots, n$  and  $1 \leq q < \frac{N}{3 \cdot 2^j}$ . Similarly, we define

$$h(\frac{N}{2j} + q) = e^{i\frac{\pi}{4} + i\frac{\pi q 2^{j-1}}{N}} g(-\frac{\pi}{2} + \frac{3\pi q 2^{j-1}}{N}) \hat{f}(\frac{N}{2j} + q) + e^{-i\frac{\pi}{4} + i\frac{\pi q 2^{j-1}}{N}} g(\frac{\pi}{2} + \frac{3\pi q 2^{j-1}}{N}) \hat{f}(-\frac{N}{2j} + q), \tag{3.20}$$

$$h(-\frac{N}{2j} + q) = -e^{-i\frac{\pi}{4} + i\frac{\pi q 2^j}{N}} g(\frac{\pi}{2} + \frac{3\pi q 2^{j-1}}{N}) \hat{f}(\frac{N}{2j} + q) - e^{i\frac{\pi}{4} + i\frac{\pi q 2^j}{N}} g(-\frac{\pi}{2} + \frac{3\pi q 2^{j-1}}{N}) \hat{f}(-\frac{N}{2j} + q), \tag{3.21}$$

for  $j = 2, 3, \dots, n$  and  $0 \leq q < \frac{N}{3 \cdot 2^j}$ .

For the first level  $j = 1$ , there is a little bit of difference since there is no higher level and the indices  $-\frac{N}{2} + q$  and  $\frac{N}{2} + q$  are the same. Therefore, we only adopt (3.18) and (3.21) for  $j = 1$  and  $0 \leq q < \frac{N}{3 \cdot 2^j}$ . Finally, let  $h(0) = \hat{f}(0)$  for the scaling function component. We denote this transformation from  $\hat{f}$  to  $h$  as  $T_W$ , which is  $h = T_W \hat{f}$ .

We may calculate that

$$\left( \sum_{k=2^{n-j-1}}^{2^{n-j}-1} + \sum_{k=-2^{n-j}}^{-2^{n-j-1}-1} \right) \frac{1}{\sqrt{2^{n-j}}} e^{-2\pi i \frac{pk}{2^{n-j}}} h(k) = \langle \hat{f}, \widehat{\psi}_{j,p} \rangle = a_{j,p}. \tag{3.22}$$

The proof is in Section 4.2. Noticing that the left-hand side of (3.22) is exactly the same as applying  $S_{W(N)}$  on  $h$ , we conclude

$$U_{WB} = S_{W(N)} T_W U_{FT(N)}. \tag{3.23}$$

Therefore, the only remaining problem is to implement the matrix  $T_W$ .

### 3.3. The implementation of $T_W$

The idea of implementing  $T_W$  is similar to  $T_G$  in the Gabor atom implementation, exploiting the block diagonal structure and using the technique of encoding the exponential of pure imaginary polynomials. However, the main challenge here is that the ranges of indices are not powers of 2. A plausible choice to get around this difficulty is to further restrict the support of  $g(x)$  to  $[-\frac{7\pi}{8}, \frac{7\pi}{8}]$ , and also adjust the phase rotation factors  $e^{-i\frac{q\pi}{2}}$  in (1.7) in order to restrict the non-trivial part of the operations in blocks of size of power of 2. However, because this choice makes the wavelet different from what is commonly used, we choose to present a method that can implement  $T_W$  without any restrictions.

From the formulas (3.18)–(3.21), we see that  $T_W$  has a block-diagonal structure. This can also be seen from Fig. 17. Let us write the unitary block explicitly as

$$T_W = \prod_{j=1}^n R_j \prod_{j=1}^n L_j \tag{3.24}$$

where  $R_j$  is the unitary corresponding to (3.20) and (3.21) for level  $j$ , and  $L_j$  is the unitary corresponding to (3.18) and (3.19) for level  $j$ . Notice that  $R_1$  and  $L_1$  are only defined by (3.21) and (3.18), respectively. Since the support of all  $R_j$ 's and  $L_j$ 's are disjoint, they are all commutative, and thus the order of the product does not matter.

By definition,  $R_j$  is acting non-trivially on the indices  $\pm \frac{N}{2j} + q$  for  $0 \leq q < \frac{N}{3 \cdot 2^j}$ . However, this range is not convenient for quantum implementation since it is not a power of 2. Therefore, we may first enlarge the range of  $q$  to  $0 \leq q < \frac{N}{2^j}$  and pad the extra slots by identity. In the binary representation, this range contains the binary strings beginning with  $\underbrace{0 \dots 0}_j$  and  $\underbrace{1 \dots 1}_j$ . When  $j \geq 2$ , we

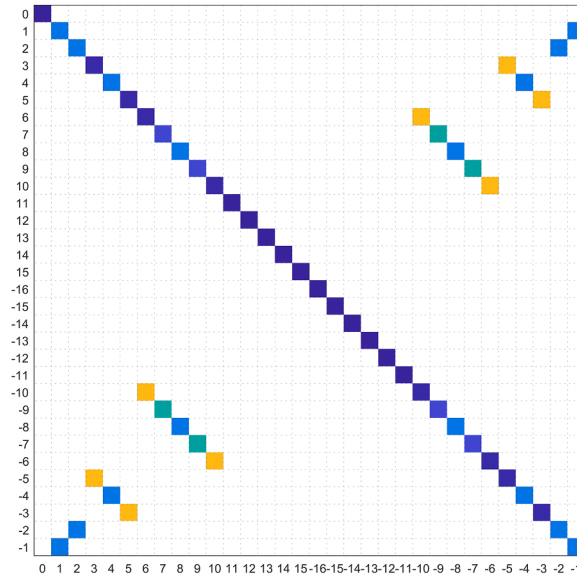


Fig. 17. Magnitude of elements in  $T_W$  with  $N = 32$ .

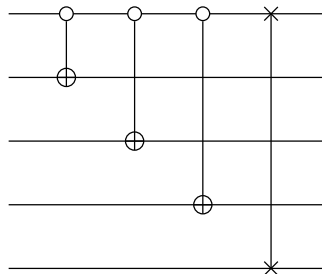


Fig. 18. The circuit for  $Q_{W(32)}$ .

shall first group these two ranges together and then perform the desired unitary transformation. To be specific, we need a permutation matrix  $Q_{W(2^j)}$  such that

$$Q_{W(2^j)} | \underbrace{0 \dots 0}_{(j-1) \text{ of } 0\text{'s}} 1 \rangle = | \underbrace{1 \dots 1}_{(j-1) \text{ of } 1\text{'s}} 0 \rangle \quad \text{and} \quad Q_{W(2^j)} | \underbrace{1 \dots 1}_{(j-1) \text{ of } 1\text{'s}} 0 \rangle = | \underbrace{1 \dots 1}_{(j-1) \text{ of } 1\text{'s}} 1 \rangle. \tag{3.25}$$

There are different choices of  $Q_{W(2^j)}$ , and we adopt the following

$$\begin{aligned} & |x_{n-1} x_{n-2} \dots x_{n-j+1} x_{n-j} \rangle \\ \mapsto & |x_{n-1}(x_{n-2} \oplus x_{n-1} \oplus 1) \dots (x_{n-j+1} \oplus x_{n-1} \oplus 1)(x_{n-j} \oplus x_{n-1} \oplus 1) \rangle \\ \mapsto & |(x_{n-j} \oplus x_{n-1} \oplus 1)(x_{n-2} \oplus x_{n-1} \oplus 1) \dots (x_{n-j+1} \oplus x_{n-1} \oplus 1)x_{n-1} \rangle, \end{aligned} \tag{3.26}$$

which is illustrated in Fig. 18.

After applying  $Q_{W(2^j)}$ , the relevant indices begin with  $\underbrace{1 \dots 10}_{(j-1) \text{ of } 1\text{'s}}$  and  $\underbrace{1 \dots 11}_{(j-1) \text{ of } 1\text{'s}}$ . Therefore, when  $j \geq 2$ , the matrix  $R_j$  can be written as

$$R_j = (Q_{W(2^j)}^\dagger \otimes I_{2^{n-j}})(I_{N-2^{n-j+1}} \oplus K_{2^{n-j+1}})(Q_{W(2^j)} \otimes I_{2^{n-j}}) \tag{3.27}$$

where  $K_{2^{n-j+1}}$  is a  $2^{n-j+1}$  dimensional matrix that can be determine according to (3.20) and (3.21). Let  $D$  be a  $\lceil \frac{N}{3 \cdot 2^j} \rceil$  dimensional diagonal matrix  $D = \text{diag}\{-\frac{1}{2} + \frac{3q2^{j-1}}{N} : 0 \leq q < \frac{N}{3 \cdot 2^j}\}$ , then

$$K_{2^{n-j+1}} = \begin{bmatrix} e^{i\pi(\frac{1}{3}D + \frac{5}{12})} \cos(\frac{\pi}{2}\beta(D)) & 0 & -e^{i\pi(\frac{1}{3}D - \frac{1}{12})} \sin(\frac{\pi}{2}\beta(D)) & 0 \\ 0 & I & 0 & 0 \\ e^{i\pi(\frac{2}{3}D + \frac{1}{12})} \sin(\frac{\pi}{2}\beta(D)) & 0 & -e^{i\pi(\frac{2}{3}D + \frac{7}{12})} \cos(\frac{\pi}{2}\beta(D)) & 0 \\ 0 & 0 & 0 & I \end{bmatrix}, \tag{3.28}$$

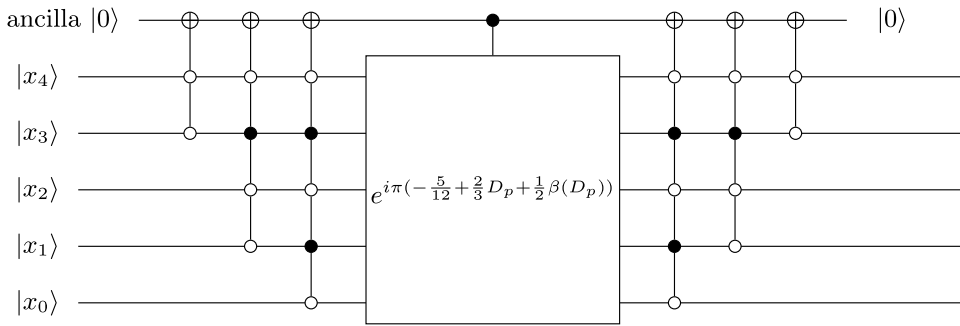


Fig. 19. The circuit for  $\widetilde{K}_{32}$ .

where  $I$  stands for the  $\frac{N}{2^j} - \lceil \frac{N}{3 \cdot 2^j} \rceil$  dimensional identity matrix. Note that as this matrix only depends on the value of  $n - j$ , it makes sense to write  $K_{2^{m+1}}$  without specifying  $n$  and  $j$ .

When  $j = 1$ , the matrix  $R_1$  is different from the (3.27). According to (3.21), we have

$$R_1 = I_{N/2} \oplus \widetilde{K}_{N/2} \tag{3.29}$$

with

$$\widetilde{K}_{N/2} = \begin{bmatrix} e^{i\pi(-\frac{5}{12} + \frac{2}{3}D + \frac{1}{2}\beta(D))} & 0 \\ 0 & I \end{bmatrix}, \tag{3.30}$$

where  $D = \text{diag}\{-\frac{1}{2} + \frac{3q}{N} : 0 \leq q < \frac{N}{6}\}$ , and  $I$  stands for the  $\frac{N}{2} - \lceil \frac{N}{6} \rceil$  dimensional identity matrix.

Next, we focus on implementing (3.27) and (3.30). We first discuss the latter one since it has a simpler form. The idea is to use an ancilla qubit to filter the binary representations of numbers smaller than  $\frac{N}{6}$ . Recall that  $\widetilde{K}_{N/2}$  is of dimension  $\frac{N}{2}$  and the largest number less than  $\frac{N}{6}$  is given by the binary representation  $\underbrace{0101 \dots}_{(n-1) \text{ bits}}$ . Therefore, all numbers less than  $\frac{N}{6}$  have the binary representation beginning

with 00, or 0100, or 010100, or so on. Once we use control gates to encode these conditional statements into the ancilla qubit, we can perform a controlled version of  $e^{i\pi(-\frac{5}{12} + \frac{2}{3}D_p + \frac{1}{2}\beta(D_p))}$ , where  $D_p$  is the padded version of  $D$ , namely  $D_p = \text{diag}\{-\frac{1}{2} + \frac{3q}{N} : 0 \leq q < \frac{N}{2}\}$ . This  $D_p$  can be implemented using the method discussed in Section 2.2.3. Finally, one should go through the uncompute process to restore the ancilla to  $|0\rangle$  and discard it. The complete circuit is shown in Fig. 19.

Next, let us construct the circuit of matrix  $K_{2^{n-j+1}}$ . We similarly introduce the padded diagonal matrix  $D_p = \text{diag}\{-\frac{1}{2} + \frac{3q2^{j-1}}{N} : 0 \leq q < \frac{N}{2^j}\}$ . One may also introduce an ancilla qubit to record whether the binary number given by the last  $n - j$  qubits is less than  $\frac{N}{3 \cdot 2^j}$ . Therefore, under the control of the ancilla qubit, the matrix to be implemented is

$$\begin{bmatrix} e^{i\pi(\frac{1}{3}D_p + \frac{5}{12})} \cos(\frac{\pi}{2}\beta(D_p)) & -e^{i\pi(\frac{1}{3}D_p - \frac{1}{12})} \sin(\frac{\pi}{2}\beta(D_p)) \\ e^{i\pi(\frac{2}{3}D_p + \frac{1}{12})} \sin(\frac{\pi}{2}\beta(D_p)) & -e^{i\pi(\frac{2}{3}D_p + \frac{7}{12})} \cos(\frac{\pi}{2}\beta(D_p)) \end{bmatrix}, \tag{3.31}$$

which can be further decomposed as

$$\begin{bmatrix} e^{i\pi(\frac{1}{3}D_p + \frac{5}{12})} & 0 \\ 0 & e^{i\pi(\frac{2}{3}D_p - \frac{5}{12})} \end{bmatrix} \begin{bmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & -\frac{I}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} e^{i\frac{\pi}{2}\beta(D_p)} & 0 \\ 0 & e^{-i\frac{\pi}{2}\beta(D_p)} \end{bmatrix} \begin{bmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & -\frac{I}{\sqrt{2}} \end{bmatrix}. \tag{3.32}$$

These four unitary matrices are either Hadamard gates or the exponentials of pure imaginary polynomials, which are constructed in Section 2.2.3. Therefore, we only need to perform the controlled version of them and finally do the uncomputation. The complete circuit is shown in Fig. 20.

Having described the implementation of all the building blocks, we conclude by drawing the circuit of  $\prod_{j=1}^n R_j = R_n R_{n-1} \dots R_1$  in Fig. 21. Note that there would be cancellations at the consecutive  $(Q_{W(2^{j+1})}^\dagger \otimes I_{2^{n-j-1}})(Q_{W(2^j)} \otimes I_{2^{n-j}})$ , which saves some gates.

For the implementation of  $\prod_{j=1}^n L_j$ , the method of implementation is very similar to  $\prod_{j=1}^n R_j$ . Therefore, we omit the details.

### 3.4. Complexity analysis

Here, we briefly discuss the complexity of the quantum wavelet transform circuits. For the exact implementation, the complexity is of  $O(\text{poly}(n))$ , and the degree may depend on the degree of  $\beta$ , which is similar to the Gabor atom case.

If we allow for an  $\epsilon$  error, it is possible to reduce the complexity as in the Gabor atom case. In particular, we may use the inexact Fourier transform [40], implement the multi-qubit control by Fig. 11, and implement the required diagonal matrices using the method in Section 2.2.4. For the Shannon wavelet, the corresponding complexity is  $O(n^2 \log(n\epsilon^{-1}))$  since we need to perform  $n$  (inverse) Fourier transforms. For the Meyer wavelet, we claim that the complexity is  $O(n^2(\log n + \log(\epsilon^{-1})^2))$  by examining each term

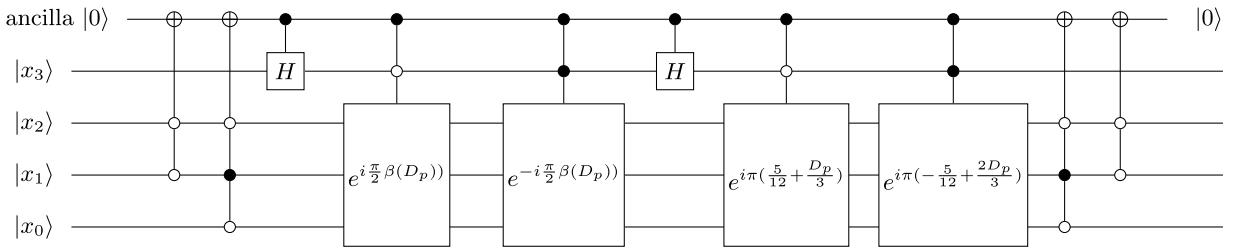


Fig. 20. The circuit for  $K_{16}$ .

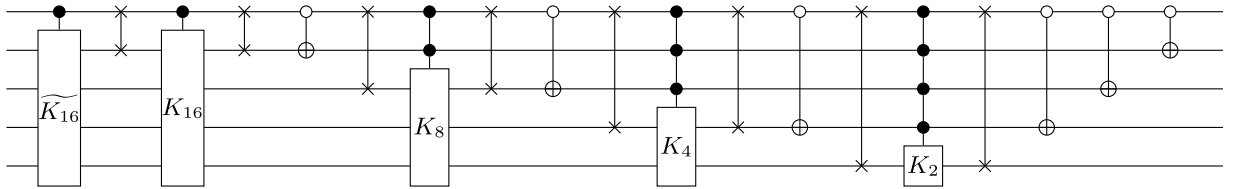


Fig. 21. The circuit of  $\prod_{j=1}^n R_j$  when  $N = 32$ .

in (3.23) as follows. The  $S_{W(N)}$  is of complexity  $O(n^2 \log(ne^{-1}))$  as discussed in the Shannon wavelet. The  $T_W$  part is mainly composed of implementing  $O(n)$  diagonal matrices, and each of them costs  $O(n \log(e^{-1})^2)$  as discussed in Section 2.2.4.

Notice that for the implementation of a required diagonal matrix, one needs an ancilla qubit, as shown in Figs. 19 and 20. At the same time, one extra ancilla qubit is needed for unwrapping the multi-qubit control on the diagonal matrices. Therefore, in total, two extra ancilla qubits are required.

#### 4. Proofs

##### 4.1. Proof of Eq. (2.17)

We may further split the sum of the left-hand side into four terms

$$\left( \sum_{k \in [jB, (j+\frac{1}{2})B]} + \sum_{k \in [(j+\frac{1}{2})B, (j+1)B]} + \sum_{k \in [-(j+1)B, -(j+\frac{1}{2})B]} + \sum_{k \in [-(j+\frac{1}{2})B, -jB]} \right) \frac{1}{\sqrt{2B}} e^{-2\pi i \frac{pk}{2B}} h(k). \tag{4.1}$$

The right-hand side is the sum on the support of  $\widehat{\psi}_{2Bj+p}$

$$\begin{aligned} \langle \hat{f}, \widehat{\psi}_{2Bj+p} \rangle &= \left( \sum_{k \in [(j-\frac{1}{2})B, (j+\frac{3}{2})B]} + \sum_{k \in [-(j+\frac{3}{2})B, -(j+\frac{1}{2})B]} \right) \hat{f}(k) \widehat{\psi}_{2Bj+p}(k)^* \\ &= \left( \sum_{\substack{k \in [jB, (j+\frac{1}{2})B] \\ \cup [-jB, -(j+\frac{1}{2})B]}} + \sum_{\substack{k \in [(j+\frac{1}{2})B, (j+1)B] \\ \cup [-(j+\frac{3}{2})B, -(j+1)B]}} + \sum_{\substack{k \in [-(j+1)B, -(j+\frac{1}{2})B] \\ \cup [(j+1)B, (j+\frac{3}{2})B]}} + \sum_{\substack{k \in [-(j+\frac{1}{2})B, -jB] \\ \cup [(j-\frac{1}{2})B, jB]}} \right) \hat{f}(k) \widehat{\psi}_{2Bj+p}(k)^*. \end{aligned} \tag{4.2}$$

Using the definition of  $h$ , we can check that the four summations in (4.1) and (4.2) are equal, respectively. For instance, the first summation

$$\begin{aligned} \sum_{\substack{k \in [jB, (j+\frac{1}{2})B] \\ \cup [-jB, -(j+\frac{1}{2})B]}} \hat{f}(k) \widehat{\psi}_{2Bj+p}(k)^* &= \sum_{q \in [0, \frac{B}{2})} \hat{f}(jB+q) \widehat{\psi}_{2Bj+p}(jB+q)^* + \sum_{q \in [0, \frac{B}{2})} \hat{f}(-jB+q) \widehat{\psi}_{2Bj+p}(-jB+q)^* \\ &= \sum_{q \in [0, \frac{B}{2})} \hat{f}(jB+q) \frac{1}{\sqrt{2B}} e^{-2\pi i \frac{p(jB+q)}{2B}} e^{\frac{i}{2}(-\frac{\pi}{2}+q\frac{\pi}{B})} g\left(-\frac{\pi}{2}+q\frac{\pi}{B}\right) \\ &\quad + \sum_{q \in [0, \frac{B}{2})} \hat{f}(-jB+q) \frac{1}{\sqrt{2B}} e^{-2\pi i \frac{p(-jB+q)}{2B}} e^{\frac{i}{2}(\frac{\pi}{2}+q\frac{\pi}{B})} g\left(\frac{\pi}{2}+q\frac{\pi}{B}\right) \\ &= \sum_{q \in [0, \frac{B}{2})} \frac{1}{\sqrt{2B}} e^{-2\pi i \frac{p(jB+q)}{2B}} h(jB+q) \quad (\text{used (2.15)}) \end{aligned} \tag{4.3}$$

$$= \sum_{k \in [jB, (j+\frac{1}{2})B)} \frac{1}{\sqrt{2B}} e^{-2\pi i \frac{pk}{2B}} h(k).$$

The other three summations can be checked similarly. Therefore, we have proved Eq. (2.17).

4.2. Proof of Eq. (3.22)

To verify this, we may further split the sum of the left-hand side into

$$\left( \sum_{k \in [-\frac{4}{3}2^{n-j-1}, -2^{n-j-1})} + \sum_{k \in [-2^{n-j}, -\frac{2}{3}2^{n-j})} + \sum_{k \in [\frac{2}{3}2^{n-j}, 2^{n-j})} + \sum_{k \in [2^{n-j-1}, \frac{4}{3}2^{n-j-1})} \right) \frac{1}{\sqrt{2^{n-j}}} e^{-2\pi i \frac{pk}{2^{n-j}}} h(k). \tag{4.4}$$

The right-hand side is the sum on the support of  $\widehat{\psi}_{j,p}$

$$\begin{aligned} \langle \hat{f}, \widehat{\psi}_{j,p} \rangle &= \left( \sum_{k \in [\frac{2}{3}2^{n-j-1}, \frac{4}{3}2^{n-j})} + \sum_{k \in [-\frac{4}{3}2^{n-j}, -\frac{2}{3}2^{n-j-1})} \right) \hat{f}(k) \widehat{\psi}_{j,p}(k)^* \\ &= \left( \sum_{\substack{k \in [-\frac{4}{3}2^{n-j-1}, -2^{n-j-1}) \\ \cup [\frac{2}{3}2^{n-j-1}, 2^{n-j-1})}} + \sum_{\substack{k \in [-2^{n-j}, -\frac{2}{3}2^{n-j}) \\ \cup [2^{n-j}, \frac{4}{3}2^{n-j})}} + \sum_{\substack{k \in [\frac{2}{3}2^{n-j}, 2^{n-j}) \\ \cup [-\frac{4}{3}2^{n-j}, -2^{n-j})}} + \sum_{k \in [2^{n-j-1}, \frac{4}{3}2^{n-j-1})} \right) \hat{f}(k) \widehat{\psi}_{j,p}(k)^*. \end{aligned} \tag{4.5}$$

We shall show that the four summations in (4.4) and (4.5) are equal, respectively. For the first summation, we have

$$\begin{aligned} \sum_{\substack{k \in [-\frac{4}{3}2^{n-j-1}, -2^{n-j-1}) \\ \cup [\frac{2}{3}2^{n-j-1}, 2^{n-j-1})}} \hat{f}(k) \widehat{\psi}_{j,p}(k)^* &= \sum_{q \in (0, \frac{1}{3}2^{n-j-1})} \hat{f}(-2^{n-j-1} - q) \widehat{\psi}_{j,p}(-2^{n-j-1} - q)^* + \sum_{q \in (0, \frac{1}{3}2^{n-j-1})} \hat{f}(2^{n-j-1} - q) \widehat{\psi}_{j,p}(2^{n-j-1} - q)^* \\ &= \sum_{q \in (0, \frac{1}{3}2^{n-j-1})} \frac{1}{\sqrt{2^{n-j}}} \hat{f}(-2^{n-j-1} - q) e^{2\pi i p(-2^{n-j-1} - q)/2^{n-j}} \widehat{\psi}_{ms}(2^{j+1-n} \pi(-2^{n-j-1} - q))^* \\ &\quad + \sum_{q \in (0, \frac{1}{3}2^{n-j-1})} \frac{1}{\sqrt{2^{n-j}}} \hat{f}(2^{n-j-1} - q) e^{2\pi i p(2^{n-j-1} - q)/2^{n-j}} \widehat{\psi}_{ms}(2^{j+1-n} \pi(2^{n-j-1} - q))^* \\ &= \sum_{q \in (0, \frac{1}{3}2^{n-j-1})} \frac{1}{\sqrt{2^{n-j}}} \hat{f}(-2^{n-j-1} - q) e^{2\pi i p(-2^{n-j-1} - q)/2^{n-j}} e^{-i\frac{\pi}{4} - i\frac{\pi q 2^j}{N}} g\left(\frac{\pi}{2} - \frac{3\pi q 2^j}{N}\right) \\ &\quad + \sum_{q \in (0, \frac{1}{3}2^{n-j-1})} \frac{1}{\sqrt{2^{n-j}}} \hat{f}(2^{n-j-1} - q) e^{2\pi i p(2^{n-j-1} - q)/2^{n-j}} e^{i\frac{\pi}{4} - i\frac{\pi q 2^j}{N}} g\left(-\frac{\pi}{2} - \frac{3\pi q 2^j}{N}\right) \\ &= \sum_{q \in (0, \frac{1}{3}2^{n-j-1})} \frac{1}{\sqrt{2^{n-j}}} e^{2\pi i p(-2^{n-j-1} - q)/2^{n-j}} h(-2^{n-j-1} - q) \\ &= \sum_{k \in [-\frac{4}{3}2^{n-j-1}, -2^{n-j-1})} \frac{1}{\sqrt{2^{n-j}}} e^{-2\pi i \frac{pk}{2^{n-j}}} h(k). \end{aligned} \tag{4.6}$$

Therefore, the first summation in (4.4) and (4.5) are equal. The remaining three summations are also equal, respectively, by a similar argument. Thus, we have proved (3.22).

5. Conclusion and discussions

This paper presents the quantum circuit implementation of wave packets, such as Gabor atoms and wavelets, with compact frequency support. The implementations for those with sharp frequency windows are rather straightforward, with the help of the quantum Fourier transform. For those with blended frequency windows, our method first reallocates the Fourier coefficients to the correct location in the frequency domain, followed by applying the transformation for the sharp-windowed versions.

There are many other kinds of wave packets, such as wave atoms [4] and curvelets [2], which give tiling in the space-frequency diagram different from Fig. 1. Our method can be extended to accommodate these wave packets as well. Another possible future direction is about higher-dimensional wave packet transforms, which could find applications in solving wave equations or the Schrödinger equation. While 2D or 3D versions of compactly supported wavelets have been discussed [25,26], little has been explored regarding Meyer-type wavelets or wave packets in higher dimensions.

Another, more detailed possible improvement is in the implementation of the unitary diagonal matrices discussed in Section 2.2.4. We anticipate the complexity can be reduced to  $O(n \log(\epsilon^{-1}))$ . Additionally, considering that QSVT can potentially handle complex

polynomials directly rather than splitting them into real and imaginary parts and using LCU to combine them [36], it may be feasible to find a polynomial that can approximate the even function  $e^{-\frac{i\pi}{4}\gamma(\arcsin(x))}$  directly while satisfying the conditions in [36, Theorem 4]. The QET-U technique [41] may also help since we start from a unitary matrix  $e^{iD}$ .

## Data availability

No data was used for the research described in the article.

## References

- [1] S. Mallat, *A Wavelet Tour of Signal Processing*, Elsevier, 1999.
- [2] E.J. Candes, L. Demanet, The curvelet representation of wave propagators is optimally sparse, *Commun. Pure Appl. Math.* 58 (11) (2005) 1472–1528.
- [3] S. Lintner, F. Malgouyres, Solving a variational image restoration model which involves  $L^\infty$  constraints, *Inverse Probl.* 20 (3) (2004) 815.
- [4] L. Demanet, L. Ying, Wave atoms and sparsity of oscillatory patterns, *Appl. Comput. Harmon. Anal.* 23 (3) (2007) 368–387.
- [5] E. Candes, L. Demanet, Curvelets and Fourier integral operators, *C.R. Math.* 336 (5) (2003) 395–398.
- [6] L. Demanet, L. Ying, Wave atoms and time upscaling of wave equations, *Numer. Math.* 113 (2009) 1–71.
- [7] F. Yan, A.M. Ilyyasu, S.E. Venegas-Andraca, A survey of quantum image representations, *Quantum Inf. Process.* 15 (1) (2016) 1–35. <https://doi.org/10.1007/s11128-015-1195-6>
- [8] S. Zhao, F. Yan, A.M. Ilyyasu, A.S. Salama, K. Hirota, Dual-level template for enhancing resolution of quantum images, *J. Adv. Comput. Intell. Inf.* 26 (3) (2022) 431–440. <https://doi.org/10.20965/jaciii.2022.p0431>
- [9] N.R. Zhou, L.X. Huang, L.H. Gong, Q.W. Zeng, Novel quantum image compression and encryption algorithm based on DQWT and 3D hyper-chaotic Henon map, *Quantum Inf. Process.* 19 (9) (2020) 21. <https://doi.org/10.1007/s11128-020-02794-3>
- [10] S. Chakraborty, S.H. Shaikh, A. Chakrabarti, R. Ghosh, An image denoising technique using quantum wavelet transform, *Int. J. Theor. Phys.* 59 (11) (2020) 3348–3371. <https://doi.org/10.1007/s10773-020-04590-2>
- [11] Y.M. Yu, J. Gao, X.Y. Mu, S.M. Wang, Adaptive LSB quantum image watermarking algorithm based on Haar wavelet transforms, *Quantum Inf. Process.* 22 (5) (2023) 21. <https://doi.org/10.1007/s11128-023-03926-1>
- [12] X.H. Song, S. Wang, S. Liu, A. Abd El-Latif, X.M. Niu, A dynamic watermarking scheme for quantum images using quantum wavelet transform, *Quantum Inf. Process.* 12 (12) (2013) 3689–3706. <https://doi.org/10.1007/s11128-013-0629-2>
- [13] X.Y. Mu, H.W. Wang, R.Y. Bao, S.M. Wang, H.Y. Ma, An improved quantum watermarking using quantum Haar wavelet transform and Qsobel edge detection, *Quantum Inf. Process.* 22 (5) (2023) 22. <https://doi.org/10.1007/s11128-023-03964-9>
- [14] S. Heidari, M. Naseri, R. Gheibi, M. Baghfalaki, M.R. Pourarian, A. Farouk, A new quantum watermarking based on quantum wavelet transforms, *Commun. Theor. Phys.* 67 (6) (2017) 732–742. <https://doi.org/10.1088/0253-6102/67/6/732>
- [15] C. Gao, R.G. Zhou, X. Li, A quantum steganography scheme using quantum wavelet transforms, *Mod. Phys. Lett. B* 36 (19) (2022) 18. <https://doi.org/10.1142/s0217984921506260>
- [16] M. Jeng, S.I. Ul Islam, D. Levy, A. Riachi, M. Chaudhary, M.A.I. Nobel, D. Kneidel, V. Jha, J. Bauer, A. Maurya, N. Mahmud, E. El-Araby, Improving quantum-to-classical data decoding using optimized quantum wavelet transform, *J. Supercomput.* 79 (18) (2023) 20532–20561. <https://doi.org/10.1007/s11227-023-05433-7>
- [17] A. Moshtaghpour, J.M. Bioucas-Dias, L. Jacques, Close encounters of the binary kind: signal reconstruction guarantees for compressive hadamard sampling with Haar wavelet basis, *IEEE Trans. Inf. Theory* 66 (11) (2020) 7253–7273. <https://doi.org/10.1109/tit.2020.2992852>
- [18] B.T. Kiani, G. De Palma, D. Englund, W. Kaminsky, M. Marvian, S. Lloyd, Quantum advantage for differential equation analysis, *Phys. Rev. A* 105 (2) (2022) 15. <https://doi.org/10.1103/PhysRevA.105.022415>
- [19] M. Bagherimehrab, K. Nakaji, N. Wiebe, A. Aspuru-Guzik, Fast quantum algorithm for differential equations, *arXiv preprint arXiv:2306.11802* (2023).
- [20] Y.-K. Liu, Quantum algorithms using the curvelet transform, in: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, 2009, pp. 391–400.
- [21] A. Fijany, C.P. Williams, Quantum wavelet transforms: fast algorithms and complete circuits, in: *Quantum Computing and Quantum Communications: First NASA International Conference, QCQC'98 Palm Springs, California, USA February 17–20, 1998 Selected Papers*, Springer, 1999, pp. 10–33.
- [22] H.S. Li, P. Fan, H.Y. Xia, S.X. Song, Quantum multi-level wavelet transforms, *Inf. Sci.* 504 (2019) 113–135. <https://doi.org/10.1016/j.ins.2019.07.057>
- [23] M. Bagherimehrab, A. Aspuru-Guzik, Efficient quantum algorithm for all quantum wavelet transforms, *Quantum Sci. Technol.* 9 (3) (2024) 035010.
- [24] Z.G. Zhang, M.A. Kon, Wavelet matrix operations and quantum transforms, *Appl. Math. Comput.* 428 (2022) 26. <https://doi.org/10.1016/j.amc.2022.127179>
- [25] H.S. Li, G.Q. Li, H.Y. Xia, Three-dimensional quantum wavelet transforms, *Front. Comput. Sci.* 17 (5) (2023) 14. <https://doi.org/10.1007/s11704-022-1639-y>
- [26] H.S. Li, P. Fan, H.L. Peng, S.X. Song, G.L. Long, Multilevel 2-D quantum wavelet transforms, *IEEE Trans. Cybern.* 52 (8) (2022) 8467–8480. <https://doi.org/10.1109/tycb.2021.3049509>
- [27] H.S. Li, P. Fan, H.Y. Xia, S.X. Song, X.J. He, The multi-level and multi-dimensional quantum wavelet packet transforms, *Sci. Rep.* 8 (2018) 23. <https://doi.org/10.1038/s41598-018-32348-8>
- [28] A. Klappenecker, Wavelets and wavelet packets on quantum computers, in: *Wavelet Applications in Signal and Image Processing VII*, 3813, SPIE, 1999, pp. 703–713.
- [29] F.A. Chaurra-Gutierrez, C. Feregrino-Urbe, J.C. Perez-Sansalvador, G. Rodriguez-Gomez, QIST: One-dimensional quantum integer wavelet S-transform, *Inf. Sci.* 622 (2023) 999–1013. <https://doi.org/10.1016/j.ins.2022.12.002>
- [30] L. Lin, Lecture notes on quantum algorithms for scientific computation, *arXiv preprint arXiv:2201.08309* (2022).
- [31] M.A. Nielsen, I. Chuang, *Quantum computation and quantum information*, 2000.
- [32] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* 52 (5) (1995) 3457.
- [33] D. Camps, L. Lin, R. Van Beeumen, C. Yang, Explicit quantum circuits for block encodings of certain sparse matrices, *arXiv preprint arXiv:2203.10236* (2022).
- [34] C. Gidney, Halving the cost of quantum addition, *Quantum* 2 (2018) 74.
- [35] H. Li, H. Ni, L. Ying, On efficient quantum block encoding of pseudo-differential operators, *Quantum* 7 (2023) 1031.
- [36] A. Gilyén, Y. Su, G.H. Low, N. Wiebe, Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, *arXiv:1806.01838* (2018).
- [37] L. Ying, Stable factorization for phase factors of quantum signal processing, *Quantum* 6 (2022) 842.
- [38] Y. Dong, L. Lin, H. Ni, J. Wang, Infinite quantum signal processing, *arXiv preprint arXiv:2209.10162* (2022).
- [39] Y. Dong, L. Lin, H. Ni, J. Wang, Robust iterative method for symmetric quantum signal processing in all parameter regimes, *arXiv preprint arXiv:2307.12468* (2023).
- [40] Y. Nam, Y. Su, D. Maslov, Approximate quantum Fourier transform with  $O(n \log(n))$  T gates, *npj Quantum Inf.* 6 (1) (2020) 26.
- [41] Y. Dong, L. Lin, Y. Tong, Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices, *PRX Quantum* 3 (4) (2022) 040305.