

Research Article

Fractional Spectral Graph Wavelets and Their Applications

Jiasong Wu ^{1,2,3,4}, Fuzhi Wu,¹ Qihan Yang,¹ Yan Zhang,¹ Xilin Liu,⁵ Youyong Kong ^{1,3,4},
Lotfi Senhadji ^{2,3,4} and Huazhong Shu ^{1,3,4}

¹Laboratory of Image Science and Technology (LIST), Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing 210096, China

²Univ Rennes, INSERM, LTSI-UMR 1099, 35042 Rennes, France

³Centre de Recherche en Information Biomédicale Sino-Français (CRIBs), Univ Rennes, INSERM, Rennes, France

⁴Southeast University, Nanjing, China

⁵College of Data Science, Taiyuan University of Technology, Taiyuan 030024, China

Correspondence should be addressed to Jiasong Wu; jswu@seu.edu.cn

Received 14 June 2020; Revised 18 September 2020; Accepted 15 October 2020; Published 6 November 2020

Academic Editor: Bogdan Smolka

Copyright © 2020 Jiasong Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the key challenges in the area of signal processing on graphs is to design transforms and dictionary methods to identify and exploit structure in signals on weighted graphs. In this paper, we first generalize graph Fourier transform (GFT) to spectral graph fractional Fourier transform (SGFRFT), which is then used to define a novel transform named spectral graph fractional wavelet transform (SGFRWT), which is a generalized and extended version of spectral graph wavelet transform (SGWT). A fast algorithm for SGFRWT is also derived and implemented based on Fourier series approximation. Some potential applications of SGFRWT are also presented.

1. Introduction

In traditional signal processing, the most commonly used tools are transforms, among which Fourier transform and wavelet transform play key roles [1]. Fourier transform and wavelet transform were generalized and extended in many contexts in recent years.

On the one hand, Fourier transform and wavelet transform were extended to fractional domain, obtaining fractional Fourier transform (FRFT) [2–23] and fractional wavelet transform (FRWT) [24–30]. From a mathematical viewpoint, the FRFT can be seen as a parametric Fourier transform [7], whose fractional order θ is a free parameter which offers more flexibility and generalization properties compared to the classical Fourier transform. From a signal processing point of view, the FRFT can be interpreted as a rotation in the time-frequency plane, i.e., the unified time-frequency transform. With the order θ increasing from 0 to 1, the FRFT can show the characteristics of the signal changing from the time domain to the frequency domain

[8]. There are roughly three main research directions for investigating FRFT: firstly, the application of FRFT to deal with many signal processing problems [9], for example, filtering, compression, image encryption, digital watermarking, pattern recognition, edge detection, antennas, radar and sonar, and communication; secondly, the discretization algorithms of the FRFT [10–17]; and lastly, the extension of the fractional idea of FRFT to other transforms, for example, fractional cosine, sine and Hartley transforms [18–21], fractional Krawtchouk transform [22], and short-time FRFT [23]. By cascading of the FRFT and the ordinary wavelet transform, Mendlovic et al. [24] first proposed FRWT. Recently, by introducing a new structure of the fractional convolution [25] associated with the FRFT, Shi et al. [26] proposed a simplified definition of the FRWT which analyzes the signal in time-frequency-FRFD domain, where FRFD denotes fractional Fourier domain. The definition of FRWT in [26] was further improved by Prasad et al. [27], who analyzed the signal only in time-FRFD domain. More recently, Dai et al. [28] presented a new FRWT which

displays the time and FRFD-frequency information jointly in the time-FRFD-frequency plane. Srivastava et al. [29] proposed a certain family of fractional wavelet transformations and Shi et al. [30] proposed a novel fractional wavelet packet transform. In general, compared to traditional transforms, fractional transforms may lead to (1) better signal filtering results due to the rotation of the time–frequency plane [9]; (2) better image compression ratios [31], image recognition results [32], and image segmentation results [33] due to the flexible choices of the fractional orders; and (3) better image encryption [34] and image watermarking [35] performances since the fractional order can be used as an additional secret key.

On the other hand, Fourier transform and wavelet transform were extended to graph domain, obtaining graph Fourier transform (GFT) [36–43] and graph wavelet transform (GWT) [44–49] to handle signal defined on the vertices of weighted graphs. Two basic approaches to signal processing on graphs have been considered: the first is rooted in the spectral graph theory [50] and builds upon the graph Laplacian matrix [36]. Although spectral decomposition of generalized graph Laplacian with negative edges has been considered for the construction of graph frequency domain [51, 52], in this paper, we limit this approach to undirected graphs with real and nonnegative edge weights for the sake of clarity. The second approach, discrete signal processing on graphs [40, 41], is rooted in the algebraic signal processing theory [53, 54] and builds on the graph shift operator. The latter works as the elementary operator that generates all linear shift-invariant filters for signals with a given structure. In particular, the graph Fourier transform in this framework expands a graph signal into a basis of eigenvectors of the adjacency matrix, and the corresponding spectrum is then given by the associated eigenvalues. Besides graph-based transforms [36–49], recent research works on graph also include, among others, sampling and interpolation on graphs [55, 56], graph signal recovery [57–60], semi-supervised classification on graphs [61, 62], graph dictionary learning [63, 64], and graph convolutional neural networks [65–72]. Please refer to [73] for more references on graph signal processing.

To the authors' knowledge, there are only few research works done for discrete signal processing on fractional graph domain, which is a combination of fractional transform domain and graph transform domain. Wang et al. [74, 75] proposed a definition of fractional Fourier transform on graphs named graph fractional Fourier transform (GFRFT), which is based on algebraic signal processing theory [53, 54]. However, similar to Fourier transform, GFRFT is a global transform and does not provide useful localization properties in fractional graph domain. Furthermore, GFRFT does not allow multiresolution analysis of graph signals. In this paper, based on spectral graph theory, we first propose a new spectral graph fractional Fourier transform (SGFRFT), which is then used to define a new spectral graph fractional wavelet transform (SGFRWT), which can be seen as an extended version of spectral graph wavelet transform (SGWT) [44].

The remainder of the paper is organized as follows. Section 2 recalls the main foundations of fractional wavelet transform, spectral graph theory, and spectral graph wavelet transform (SGWT). The spectral graph fractional Fourier transform (SGFRFT) and the spectral graph fractional wavelet transform (SGFRWT) are defined in Section 3. Section 4 is dedicated to the Fourier series approximation-based fast algorithm for forward and inverse SGFRWT. Several applications of SGFRWT for addressing different problems are shown in Section 5. Section 6 concludes the paper.

2. Preliminary

In the following, we first recall the forward and inverse fractional Fourier transform (FRFT) [2–5] and then the forward and inverse fractional wavelet transform (FRWT) [24, 26–28] and also show how scaling operator may be expressed in the FRFT domain. Table 1 summarizes the definitions of various transforms for classical signal processing and also those for graph signal processing.

2.1. Fractional Fourier Transform (FRFT). The θ -order forward FRFT [2–5] is the decomposition of a function f according to the fractional Fourier kernel, i.e.,

$$\widehat{f}_\theta(u) = \langle f, K_\theta^* \rangle = \int_{-\infty}^{+\infty} f(x) K_\theta(x, u) dx, \quad (1)$$

where

$$K_\theta(x, u) = \begin{cases} A_\theta \exp[(j/2)(x^2 + u^2) \cot \theta - jxu \csc \theta], & \theta \neq n\pi, \\ \delta(x - u), & \theta = 2n\pi, \\ \delta(x + u), & \theta = (2n + 1)\pi, \end{cases} \quad (2)$$

$$A_\theta = \sqrt{\frac{1 - j \cot \theta}{2\pi}}, \quad (3)$$

where θ indicates the rotation angle of the transform and $\delta(x)$ represents the Dirac distribution.

The inverse FRFT is given by [2–5]

$$f(x) = \langle \widehat{f}_\theta, K_\theta \rangle = \int_{-\infty}^{+\infty} \widehat{f}_\theta(u) K_\theta^*(x, u) du. \quad (4)$$

The kernel in equation (2) can be obtained by its spectral expansion [5]:

$$K_\theta(x, u) = \sum_{k=0}^{\infty} \exp(-j\theta k) \xi_k(u) \xi_k(x). \quad (5)$$

$\xi_k(x)$ in equation (5) is the k th-order normalized Hermite function [11, 12], which is the eigenfunction of the Fourier transform. From (5), we can see that FRFT is a generalized version of the Fourier transform, since they share the same eigenfunction $\xi_k(x)$ and the eigenvalues of the FRFT are the θ th root of the eigenvalues of the Fourier

transform [2]. When the rotation angle $\theta = \pi/2$, the FRFT defaults to Fourier transform.

The discrete form of equation (5) can be expressed as follows [11, 12]:

$$\mathbf{F}_N^\theta = \mathbf{V}_N \mathbf{D}_N^\theta \mathbf{V}_N^T = \mathbf{V}_N \text{diag}([1, e^{-j\theta}, \dots, e^{-j(N-1)\theta}]) \mathbf{V}_N^T, \quad (6)$$

where $\text{diag}(\cdot)$ denotes a diagonal matrix formed from its vector argument. \mathbf{F}_N^θ denotes the θ -order forward FRFT matrix of size $N \times N$. $\mathbf{V}_N = [\mathbf{v}_0 \ \mathbf{v}_1 \ \dots \ \mathbf{v}_{N-1}]$, \mathbf{v}_k is the k th-order DFT Hermite eigenvector. Note that \mathbf{F}_N^θ becomes identity matrix when $\theta = 0$ and Fourier matrix when $\theta = \pi/2$.

2.2. Fractional Wavelet Transform (FRWT). There are several definitions for fractional wavelet transform (FRWT)

$$W_f^\theta(s, a) = (T_\theta^s f)(a) = \sqrt{\frac{2\pi}{1-j \cot \theta}} \int_{-\infty}^{\infty} e^{j/2 s^2 u^2 \cot \theta} \hat{f}_\theta(u) \hat{\psi}_\theta^*(su) K_\theta^*(u, a) du, \quad (9)$$

where \hat{f}_θ stands for the θ -order FRFT of f . Here also, as for the classical CWT [1], the scaling is operated in the transformed domain “ u ” and not directly in the time domain.

The inverse of the FRWT is given by [28]

$$f(x) = \frac{1}{2\pi \sin \theta C_{\theta, \psi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f^\theta(s, a) \psi_{\theta, s, a}(x) \frac{ds da}{s}, \quad (10)$$

when the following admissibility condition is satisfied [28]:

$$\int_0^{\infty} \frac{|\hat{\psi}_\theta(s)|^2}{s} ds = C_{\theta, \psi} < \infty. \quad (11)$$

2.3. Spectral Graph Theory and Spectral Graph Wavelet Transform. In this section, we briefly review the spectral graph theory and also the spectral graph wavelet transform [44].

2.3.1. Spectral Graph Theory. We consider an undirected, connected, weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, W\}$, composed of a finite set of vertices \mathcal{V} (with $\text{Card}(\mathcal{V}) = N$), a set of edges \mathcal{E} , and an adjacency symmetric and positive-valued matrix \mathbf{W} . A real-valued signal f defined on the vertices \mathcal{V} of the graph \mathcal{G} is an $N \times 1$ vector where each entry is the value $f(n)$ assigned to the vertex n . The (nonnormalized) graph Laplacian operator is the matrix defined by $\mathbf{L} = \mathbf{D} - \mathbf{W} \in \mathbb{R}^{N \times N}$, where \mathbf{D} is the diagonal degree matrix defined by $d_{m,m} = \sum_n w_{m,n}$. $d_{m,m}$ is the degree of vertex m .

As the matrix \mathbf{L} is real and symmetric, then it is diagonalizable and its eigenvectors $\{\chi_0, \chi_1, \dots, \chi_{N-1}\}$, sorted according to the ascending order of the corresponding eigenvalues $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{N-1}$, form an orthonormal basis. Therefore, with the unitary column matrix:

[24, 26–28]. In this part, we choose the definition in [28] since it displays the signal in the joint time-FRFD-frequency plane.

The θ -order forward FRWT is defined as [28]

$$W_f^\theta(s, a) = \langle f, \psi_{\theta, s, a} \rangle = \int_{-\infty}^{\infty} f(x) \psi_{\theta, s, a}^*(x) dx, \quad (7)$$

where

$$\psi_{\theta, s, a}(x) = e^{-j/2 (x^2 - a^2 - (x-a/s)^2) \cot \theta} \psi_{s, a}(x), \quad (8)$$

with $\psi_{s, a}(x) = (1/s) \psi(x - a/s)$.

The unitarity property of the FRFT applied to (7) leads to an equivalent form of FRWT:

$$\boldsymbol{\chi} = [\chi_0, \chi_1, \dots, \chi_{N-1}], \quad (12)$$

and the diagonal matrix $\Lambda = \text{diag}([\lambda_0, \lambda_1, \dots, \lambda_{N-1}])$, we have

$$\mathbf{L} = \boldsymbol{\chi} \Lambda \boldsymbol{\chi}^H, \quad (13)$$

where the superscript H denotes the Hermitian transpose operation. The graph Fourier transform (GFT) of the signal f is then the sequence provided by the scalar products [36]:

$$\hat{f}(\ell) = \langle f, \chi_\ell \rangle = \sum_{n=1}^N f(n) \chi_\ell^*(n), \quad \ell = 0, 1, \dots, N-1, \quad (14)$$

and f can be recovered (inverse GFT) by means of the reconstruction formula:

$$f(n) = \langle \hat{f}, \chi_n^* \rangle = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(n), \quad n = 1, \dots, N. \quad (15)$$

Using vectors and matrices notations, equations (14) and (15) become

$$\begin{aligned} \hat{f} &= [\hat{f}(0) \ \hat{f}(1) \ \dots \ \hat{f}(N-1)]^T = \boldsymbol{\chi}^H f, \\ f &= [f(1) \ f(2) \ \dots \ f(N)]^T = \boldsymbol{\chi} \hat{f}. \end{aligned} \quad (16)$$

The graph Fourier transform obeys the Parseval theorem; that is, for any signals f and h defined on the graph \mathcal{G} , we have

$$\langle f, h \rangle = \langle \hat{f}, \hat{h} \rangle. \quad (17)$$

2.3.2. Spectral Graph Wavelet Transform (SGWT). The spectral graph wavelet transform (SGWT) of the signal f with the kernel g is defined by [44]

$$W_f(s, n) = (T_g^s f)(n) = \sum_{\ell=0}^{N-1} g(s\lambda_\ell) \widehat{f}(\ell) \chi_\ell(n), \quad n = 1, \dots, N, \quad (18)$$

where

$$T_g^s = g(sL), \quad (19)$$

and the kernel g is continuous positive-valued function defined on \mathbb{R}^+ satisfying

$$g(0) = \lim_{x \rightarrow +\infty} g(x) = 0 \int_0^{+\infty} \frac{g(x)}{x^2} dx = C_g \in \mathbb{R}^+. \quad (20)$$

Using equation (14), the SGWT becomes

$$W_f(s, n) = (T_g^s f)(n) = \sum_{m=1}^N f(m) \psi_{s,n}^*(m) = \langle f, \psi_{s,n} \rangle, \quad n = 1, \dots, N, \quad (21)$$

with

$$\psi_{s,n}(m) = \sum_{\ell=0}^{N-1} g(s\lambda_\ell) \chi_\ell(m) \chi_\ell^*(n), \quad m = 1, \dots, N. \quad (22)$$

The signal f can be recovered up to its mean value using the inverse formula [44]:

$$f(m) - \langle f, \chi_0 \rangle \chi_0(m) = \frac{1}{C_g} \sum_{n=1}^N \int_0^{+\infty} W_f(s, n) \psi_{s,n}(m) \frac{ds}{s}, \quad m = 1, \dots, N. \quad (23)$$

3. Spectral Graph Fractional Transforms

We first propose a novel spectral graph fractional Fourier transform (SGFRFT) in Section 3.1 and then give the definition of spectral graph fractional wavelet transform (SGFRWT) in Section 3.2.

3.1. Spectral Graph Fractional Fourier Transform. Similar to (13), we define the graph fractional Laplacian operator \mathbf{L}_θ as follows:

$$\mathbf{L}_\theta = \boldsymbol{\gamma} \mathbf{R} \boldsymbol{\gamma}^H, \quad (24)$$

where $0 < \theta \leq 1$ and $\boldsymbol{\gamma}$ is the power matrix given by

$$\boldsymbol{\gamma} = [\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_{N-1}] = \boldsymbol{\chi}^\theta, \quad (25)$$

with χ being the unitary (i.e., inverse graph Fourier transform matrix) shown in (12). Note that as χ is unitary, then $\boldsymbol{\gamma} = \boldsymbol{\chi}^\theta$ is also unitary. And

$$\mathbf{R} = \text{diag}([r_0, r_1, \dots, r_{N-1}]) = \boldsymbol{\Lambda}^\theta, \quad (26)$$

that is,

$$r_\ell = \lambda_\ell^\theta, \quad \ell = 0, 1, \dots, N-1. \quad (27)$$

Considering $\lambda_\ell \in [0, \lambda_{N-1}]$, and from (27), we can easily get $r_\ell \in [0, r_{N-1}]$. The reason why we choose \mathbf{L}_θ as the

definition of graph fractional Laplacian operator will be discussed in Section 3.2.

The forward spectral graph fractional Fourier transform (SGFRFT) of any signal f defined on the vertices \mathcal{V} of the graph \mathcal{G} is defined by

$$\widehat{f}_\theta(\ell) := \langle f, \chi_\ell \rangle = \sum_{n=1}^N f(n) \gamma_\ell^*(n), \quad \ell = 0, 1, \dots, N-1, \quad (28)$$

or by the following matrix form:

$$\widehat{\mathbf{f}}_\theta = [\widehat{f}_\theta(0) \ \widehat{f}_\theta(1) \ \dots \ \widehat{f}_\theta(N-1)]^T = \boldsymbol{\gamma}^H \mathbf{f}. \quad (29)$$

For $\theta = 1$, the proposed SGFRFT in (29) is no more than the GFT in (18) defined by Shuman et al. [36]. For other values of θ , we can compute $\boldsymbol{\gamma} = \boldsymbol{\chi}^\theta$ by using Schur–Padé algorithm [76, 77] which is found to be superior in accuracy and stability to several alternatives, including eigendecomposition method and also the approaches based on the formula $\boldsymbol{\chi}^\theta = e^{\theta \log(\boldsymbol{\chi})}$.

The inverse SGFRFT is given by

$$f(n) = \langle \widehat{\mathbf{f}}_\theta, \boldsymbol{\gamma}_\ell^* \rangle = \sum_{\ell=0}^{N-1} \widehat{f}_\theta(\ell) \gamma_\ell(n), \quad n = 1, \dots, N, \quad (30)$$

or by the following matrix form:

$$\mathbf{f} = [f(1) \ f(2) \ \dots \ f(N)]^T = \boldsymbol{\gamma} \widehat{\mathbf{f}}_\theta. \quad (31)$$

The graph fractional Fourier transform obeys the Parseval theorem; that is, for any signals f and h defined on the graph \mathcal{G} , we have

$$\langle f, h \rangle = \langle \widehat{\mathbf{f}}_\theta, \widehat{\mathbf{h}}_\theta \rangle. \quad (32)$$

Note that the forward and inverse SGFRFTs defined here are different from the forward and inverse graph fractional Fourier transforms (GFRFTs) [74, 75], which are, respectively, given by

$$\widehat{\mathbf{f}}'_\theta = [\widehat{f}'_\theta(0) \ \widehat{f}'_\theta(1) \ \dots \ \widehat{f}'_\theta(N-1)]^T = \boldsymbol{\gamma}' \mathbf{f}, \quad (33)$$

$$\mathbf{f} = [f(1) \ f(2) \ \dots \ f(N)]^T = (\boldsymbol{\gamma}')^{-1} \widehat{\mathbf{f}}'_\theta, \quad (34)$$

where

$$\boldsymbol{\sigma}' = (\boldsymbol{\chi}')^\theta, \quad (35)$$

and $\boldsymbol{\chi}' = \mathbf{V}^{-1}$ is obtained by the Jordan decomposition of the adjacency matrix \mathbf{W} as follows [74, 75]:

$$\mathbf{W} = \mathbf{V} \mathbf{J}_\mathbf{W} \mathbf{V}^{-1}, \quad (36)$$

where \mathbf{V} is a matrix composed of eigenvector and $\mathbf{J}_\mathbf{W}$ is a block-diagonal matrix. For more details, please refer to Appendix A of [74].

From (13), (29), (31), and (33)–(36), we can see that the differences are that the proposed SGFRFTs are based on the decomposition of the graph Laplacian matrix \mathbf{L} , while the GFRFTs are based on the decomposition of the adjacency matrix \mathbf{W} .

3.2. Spectral Graph Fractional Wavelet Transform (SGFRWT).

$$W_f(\theta, s, n) = (T_{g_\theta}^s f)(n) = \sum_{m=1}^N f(m) \psi_{\theta, s, n}^*(m) = \langle f, \psi_{\theta, s, n} \rangle, \quad n = 1, \dots, N, \quad (37)$$

where

$$T_{g_\theta}^s = g(s\mathbf{L}_\theta), \quad (38)$$

$$\psi_{\theta, s, n}(m) = \sum_{l=0}^{N-1} g(s\lambda_l^\theta) \gamma_l(m) \gamma_l^*(n), \quad m = 1, \dots, N. \quad (39)$$

Why we choose \mathbf{L}_θ in (24) as the definition of graph fractional Laplacian operator and choose $\psi_{\theta, s, n}(m)$ in (39) as the basis of SGFRWT? There are two reasons. (1) We want to define the basis of SGFRWT $\psi_{\theta, s, n}(m)$ based on a new function $\gamma_l(n)$ instead of the same function $\chi_l(n)$ as basis of SGWT $\psi_{s, n}(m)$ in (23). The SGFRWT and SGWT will have different characteristics since they are based on different functions $\gamma_l(n)$ and $\chi_l(n)$, respectively. (2) We want to define the basis of SGFRWT $\psi_{\theta, s, n}(m)$ which changes directly with λ_l^θ . Since the parameter θ acts on the eigenvalues of the Fourier matrix in the form of exponents in the spectral expansion of FRFT in (5), similarly, we also want to obtain the form that the parameter θ can directly affect λ_l in the proposed definition of SGFRWT. Two other possible definitions of SGFRWT are also discussed in Appendices A and B, respectively.

From (37), we can see that the proposed SGFRWT is a generalization of SGWT in (21) defined by Hammond et al. [44]; that is, when $\theta=1$, the proposed SGFRWT defaults to the SGWT. The proposed SGFRWT can be seen as a parametric generalization of SGWT [7]. Furthermore, the proposed SGFRWT converts a graph signal into an intermediate domain between vertex and spectral graph wavelet. With the order θ increasing from 0 to 1, the SGFRWT can provide much more domain choices of expressions for a graph signal.

The scaling functions are then obtained by

$$\phi_\theta(n) = T_{h_\theta} \delta_n = h(\mathbf{L}_\theta) \delta_n, \quad (40)$$

and the coefficients by

$$S_{\theta, f}(n) = \langle \phi_{\theta, n}, f \rangle. \quad (41)$$

Stable recovery of the original signal \mathbf{f} from the SGFRWT coefficients will be assured if the quantity $G(r) = h^2(r) + \sum_{j=1}^J g^2(s_j r)$, where s_j denotes j th scale, is bounded away from zero on the fractional Laplacian \mathbf{L}_θ .

Similar to (21), we can define the spectral graph fractional wavelet transform (SGFRWT) operator $T_{g_\theta}^s$ as follows:

Moreover, under the same conditions, Lemmas 5.1, 5.2, 5.3, and 5.4 and Theorems 5.5 and 5.6 in [44] are verified by the SGFRWT, whose properties are shown in Appendix C.

4. Fourier Series Approximation and Fast SGFRWT

If we compute the SGFRWT by using equation (37) directly, we have to compute the entire set of eigenvectors and eigenvalues of \mathbf{L}_θ shown in (24). Both the computational complexity and the memory consumption become unacceptable when the data is larger than hundreds of thousands or millions of dimensions. Therefore, in this section, we propose a fast algorithm for computing the SGFRWT based on approximating the scaled generating kernels g_θ by low-order Fourier series. It should be noted that the proposed fast SGFRWT algorithm is an extension of the fast SGWT algorithm from real domain to complex domain. The proposed fast SGFRWT algorithm is different from the fast graph Fourier transform proposed by Magoarou et al. [39], who use the modified Jacobi eigenvalue algorithm for approximating the graph Laplacian matrix. Note that the proposed SGFRWT algorithm can approximate an arbitrary matrix while the algorithm in [39] is only practicable for approximating symmetric matrix.

The following lemma shows that the polynomial approximation of $g(sx)$ may be taken over a finite range containing the spectrum of \mathbf{L}_θ .

Lemma 1. *Let $r_{\max} \geq r_{N-1}$ be any upper bound on the fractional spectrum of \mathbf{L}_θ . For fixed $s > 0$, let $p(x)$ be a polynomial approximant of $g(sx)$ with L_∞ error $B = \sup_{x \in [0, r_{\max}]} |g(sx) - p(x)|$. Then the approximate SGFRWT coefficients $\tilde{W}_f(\theta, s, n) = (p(\mathbf{L}_\theta)f)_n$ satisfy $|W_f(\theta, s, n) - \tilde{W}_f(\theta, s, n)| \leq B \|f\|$.*

The proof of Lemma 1 is similar to Lemma 6.1 in [44].

In [44], Hammond et al. approximated $g(s\mathbf{L})$ shown in (19) by truncated Chebyshev expansions. The reason is that $g(s\mathbf{L})$ is real as the original \mathbf{L} shown in (13) is a real matrix. However, the Chebyshev polynomial approximation method is not suitable for our problem of approximating $g(s\mathbf{L}_\theta)$ shown in (38), because $g(s\mathbf{L}_\theta)$ may be complex as

$g(s\mathbf{L}_\theta)$ shown in (33) may be a complex matrix. Therefore, in this part, we will use the truncated Fourier series expansion instead of Chebyshev expansions to approximate $g(s\mathbf{L}_\theta)$ shown in (38).

The Fourier series of a general function $f(x)$ is given by [78]

$$f(y) = \sum_{k=-\infty}^{\infty} c_k \exp(iky), \quad (42)$$

where

$$c_k = \frac{1}{2\pi} \int_0^{2\pi} f(y) \exp(-iky) dy = \begin{cases} a_0, & k = 0, \\ \frac{(a_k - ikb_k)}{2}, & k > 0, \\ \frac{(a_k + ikb_k)}{2}, & k < 0, \end{cases}$$

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(y) dy,$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(y) \cos(ky) dy,$$

$$b_k = \frac{1}{\pi} \int_0^{2\pi} f(y) \sin(ky) dy.$$

(43)

We now assume a fixed set of wavelet scales s_n . For each n , approximating $g(s_n x)$ for $x \in [0, r_{\max}]$ can be done by scaling the domain of $\exp(-iky)$, $y \in [0, 2\pi]$, using the transformation $x = yr_{\max}/(2\pi)$. Then, $g(s_n x)$ can be written by

$$g(s_n x) = \sum_{k=-\infty}^{\infty} c_{n,k} F_k(x), \quad x \in [0, r_{\max}], \quad (44)$$

with

$$c_{n,k} = \frac{1}{2\pi r_{\max}} \int_0^{r_{\max}} g(s_n x) F_{-k}(x) dx, \quad (45)$$

$$F_k(x) = \exp\left(\frac{i2\pi kx}{r_{\max}}\right). \quad (46)$$

Note that

$$F_k(x) = (F_{-k}(x))^*. \quad (47)$$

The above equation is due to the fact that

$$\int_0^{r_{\max}} F_k(x) F_{-l}(x) \frac{1}{r_{\max}} dx = \delta_{kl}. \quad (48)$$

For each scale s_j , $g(s_j x)$ can be approximated by the truncated Fourier expansion (42) to $2M_j + 1$ terms; that is,

$$p_j(x) = \sum_{k=-M_j}^{M_j} c_{n,k} F_k(x), \quad x \in [0, r_{\max}]. \quad (49)$$

We may choose an appropriate M_j in practice to achieve a balance between accuracy and computational complexity. We can also use exactly the same scheme to approximate the scaling function kernel h_θ shown in (40) by p_0 . Then, the approximated SGFRWT wavelet and scaling function coefficients are, respectively, given by

$$\tilde{W}_f(\theta, s_j, n) = \left(\sum_{k=-M_j}^{M_j} c_{j,k} F_k(\mathbf{L}_\theta) \mathbf{f} \right)_n, \quad (50)$$

$$\tilde{S}_f(\theta, n) = \left(\sum_{k=-M_0}^{M_0} c_{0,k} F_k(\mathbf{L}_\theta) \mathbf{f} \right)_n, \quad (51)$$

where

$$F_k(\mathbf{L}_\theta) = \exp\left(\frac{i2\pi k \mathbf{L}_\theta}{r_{\max}}\right). \quad (52)$$

Note that (52) is easily obtained by applying the function $F_k(\cdot)$ in (46) to the matrix \mathbf{L}_θ , and we have

$$F_k(\mathbf{L}_\theta) = (F_{-k}(\mathbf{L}_\theta))^*. \quad (53)$$

The efficiency of this algorithm depends on the following recursive formula:

$$F_k(\mathbf{L}_\theta) \mathbf{f} = F_1(\mathbf{L}_\theta) (F_{k-1}(\mathbf{L}_\theta) \mathbf{f}), \quad k \in [-M_j, M_j]. \quad (54)$$

As the signal $\mathbf{f} \in \mathbb{R}^N$, taking equation (53) into consideration, equation (54) can only be computed for $k \in [1, M_j]$, and the other $k \in [-M_j, -1]$ can be obtained by conjugate operation.

The computational complexity of the proposed fast SGFRWT algorithm is analyzed as follows:

- (1) The computation of γ from χ in (25) by using Schur–Padé algorithm [76, 77] needs $O(N^3)$ operations.
- (2) The computation cost of each $F_k(\mathbf{L}_\theta) \mathbf{f}$ in (54) is $O(6|E|)$, where $|E|$ is the number of nonzero edges in the graph, and $6|E|$ because $F_1(\mathbf{L}_\theta)$ is a complex matrix and a complex multiplication requires 4 real multiplications and 2 real additions (6 operations). Therefore, the computation of all $F_k(\mathbf{L}_\theta) \mathbf{f}$, $k = 1, 2, \dots, M_j$ needs $O(6M_j|E|)$ operations.
- (3) It requires $O(6N)$ operations at scale j for each $k \leq M_j$ in (50) and (51). Therefore, the computation of (50) and (51) needs $O(6N \sum_{j=0}^J M_j)$ operations.

Therefore, the total computational complexity of the fast SGFRWT algorithm is $O(N^3 + 6M_j|E| + 6N \sum_{j=0}^J M_j)$.

Implementation of (54) requires memory of size $2N$. The total memory requirement for the proposed fast SGFRWT is $2N(J+1) + 2N$.

4.1. Fast Computation of Adjoint. Given a fixed set of wavelet scales $\{s_j\}_{j=1}^J$, and including the scaling functions ϕ_n , the SGFRWT $\mathbf{W}_\theta \mathbf{f} = ((\mathbf{T}_{h_\theta} \mathbf{f})^T, (\mathbf{T}_{g_\theta^{s_1}} \mathbf{f})^T, \dots, (\mathbf{T}_{g_\theta^{s_J}} \mathbf{f})^T)^T$ can be seen as a linear map $\mathbf{W}_\theta: \mathbb{R}^N \rightarrow \mathbb{C}^{N(J+1)}$, where \mathbb{C} denotes the complex domain. Let $\tilde{\mathbf{W}}_\theta \mathbf{f} = (((p_0(\mathbf{L}_\theta)) \mathbf{f})^T, (p_1(\mathbf{L}_\theta) \mathbf{f})^T, \dots, (p_J(\mathbf{L}_\theta) \mathbf{f})^T)^T$ denote the approximated fractional wavelet transform by using the fast SGFRWT algorithm. In this section, we show that both the adjoint $\mathbf{W}_\theta^*: \mathbb{C}^{N(J+1)} \rightarrow \mathbb{R}^N$ and the composition $\mathbf{W}_\theta^* \mathbf{W}_\theta: \mathbb{R}^N \rightarrow \mathbb{R}^N$ can be computed efficiently by using the Fourier series approximation.

For any $\boldsymbol{\eta} \in \mathbb{C}^{N(J+1)}$, we consider $\boldsymbol{\eta}$ as the concatenation $\boldsymbol{\eta} = (\eta_0^T, \eta_1^T, \dots, \eta_J^T)^T$ with each $\eta_j \in \mathbb{C}^N$ for $0 \leq j \leq J$; we have

$$\begin{aligned} \langle \boldsymbol{\eta}, \mathbf{W}_\theta \mathbf{f} \rangle_{N(J+1)} &= \langle \boldsymbol{\eta}_0, T_{h_\theta} \mathbf{f} \rangle + \sum_{j=1}^J \langle \boldsymbol{\eta}_j, T_{g_\theta^{s_j}} \mathbf{f} \rangle_N \\ &= \langle T_{h_\theta}^* \boldsymbol{\eta}_0, \mathbf{f} \rangle + \left\langle \sum_{j=1}^J (T_{g_\theta^{s_j}})^* \boldsymbol{\eta}_j, \mathbf{f} \right\rangle_N \quad (55) \\ &= \langle T_{h_\theta}^* \boldsymbol{\eta}_0 + \sum_{j=1}^J (T_{g_\theta^{s_j}})^* \boldsymbol{\eta}_j, \mathbf{f} \rangle_N. \end{aligned}$$

Considering $\langle \boldsymbol{\eta}, \mathbf{W}_\theta \mathbf{f} \rangle_{N(J+1)} = \langle \mathbf{W}_\theta^* \boldsymbol{\eta}, \mathbf{f} \rangle_N$, we have

$$\mathbf{W}_\theta^* \boldsymbol{\eta} = T_{h_\theta}^* \boldsymbol{\eta}_0 + \sum_{j=1}^J (T_{g_\theta^{s_j}})^* \boldsymbol{\eta}_j. \quad (56)$$

Similar to (50) and (51), the adjoint operator \mathbf{W}_θ^* in (56) can be approximated by

$$\begin{aligned} p_j^*(x) p_j(x) &= \left(\sum_{k=-M_j}^{M_j} c_{j,k} F_k(x) \right)^* \left(\sum_{l=-M_j}^{M_j} c_{j,l} F_l(x) \right) = \sum_{k=-M_j}^{M_j} \sum_{l=-M_j}^{M_j} c_{j,k}^* c_{j,l} F_k^*(x) F_l(x) = \sum_{k=-M_j}^{M_j} \sum_{l=-M_j}^{M_j} c_{j,k}^* c_{j,l} F_{l-k}(x), \\ &= \sum_{k=-M_j}^{M_j} \sum_{k'+k=-M_j}^{M_j} c_{j,k}^* c_{j,k'+k} F_{k'}(x) = \sum_{i=-M_j}^{M_j} \sum_{k+i=-M_j}^{M_j} c_{j,i}^* c_{j,k+i} F_k(x) = \sum_{i=-M_j}^{M_j} \sum_{k=-M_j-i}^{M_j-i} c_{j,i}^* c_{j,k+i} F_k(x), \quad (60) \\ &= \sum_{k=-2M_j}^{-1} \sum_{i=-M_j-i}^{M_j} c_{j,i}^* c_{j,k+i} F_k(x) + \sum_{k=0}^{2M_j} \sum_{i=-M_j}^{M_j-k} c_{j,i}^* c_{j,k+i} F_k(x) = \sum_{k=-M^*}^{M^*} d_{j,k} F_k(x). \end{aligned}$$

Note that $F_k^*(x) F_l(x) = F_{l-k}(x)$ is used in the above derivation.

Therefore,

$$\begin{aligned} P(x) &= \sum_{j=0}^J p_j^*(x) p_j(x) = \sum_{j=0}^J \sum_{k=-M^*}^{M^*} d_{j,k} F_k(x) \\ &= \sum_{k=-M^*}^{M^*} \sum_{j=0}^J d_{j,k} F_k(x) = \sum_{k=-M^*}^{M^*} d_k F_k(x). \quad (61) \end{aligned}$$

$$\tilde{\mathbf{W}}_\theta^* \boldsymbol{\eta} = \sum_{j=0}^J p_j^*(\mathbf{L}_\theta) \boldsymbol{\eta}_j. \quad (57)$$

In the following, we derive a fast algorithm for the computation of $\tilde{\mathbf{W}}_\theta^* \tilde{\mathbf{W}}_\theta$, which is an approximation of the composition $\mathbf{W}^* \mathbf{W}$. In general, when computing $\tilde{\mathbf{W}}_\theta^* \tilde{\mathbf{W}}_\theta$, we first apply $\tilde{\mathbf{W}}_\theta$ and then apply $\tilde{\mathbf{W}}_\theta^*$ by the fast SGFRWT, which needs $2J$ Fourier series expansions. However, the computational complexity of $\tilde{\mathbf{W}}_\theta^* \tilde{\mathbf{W}}_\theta$ can further be reduced. Note that

$$\tilde{\mathbf{W}}_\theta^* \tilde{\mathbf{W}}_\theta \mathbf{f} = \sum_{j=0}^J p_j^*(\mathbf{L}_\theta) (p_j(\mathbf{L}_\theta) \mathbf{f}) = \sum_{j=0}^J (p_j^*(\mathbf{L}_\theta) p_j(\mathbf{L}_\theta)) \mathbf{f}. \quad (58)$$

In the following, we derive a fast algorithm for the computation of $\tilde{\mathbf{W}}_\theta^* \tilde{\mathbf{W}}_\theta \mathbf{f}$.

Lemma 2. Set $P(x) = \sum_{j=0}^J p_j^*(x) p_j(x)$, which has degree $M^* = 2 \max\{M_j\}$. Then, $P(x)$ can be computed by

$$P(x) = \sum_{k=-M^*}^{M^*} d_k F_k(x), \quad (59)$$

where

$$d_k = \begin{cases} \sum_{i=-M_j-i}^{M_j} c_{j,i}^* c_{j,k+i}, & -2M_j \leq k \leq -1 \\ M_j - k \\ \sum_{i=-M_j}^{M_j} c_{j,i}^* c_{j,k+i}, & 0 < k \leq 2M_j \end{cases} \quad \text{and}$$

Proof

The proof of Lemma 2 is now completed.

Combining (58) and (59), we have

$$\tilde{\mathbf{W}}_\theta^* \tilde{\mathbf{W}}_\theta \mathbf{f} = P(\mathbf{L}_\theta) \mathbf{f} = \sum_{k=-M^*}^{M^*} d_k F_k(\mathbf{L}_\theta) \mathbf{f}. \quad (62)$$

From (62), we can see that $\tilde{\mathbf{W}}_\theta^* \tilde{\mathbf{W}}_\theta \mathbf{f}$ can be computed by a single Fourier series expansion with twice the length, which reduces the computational cost by a factor J comparing to the direct computation of $\tilde{\mathbf{W}}_\theta^* \tilde{\mathbf{W}}_\theta \mathbf{f}$. \square

4.2. Reconstruction by Using the Fast Adjoint Algorithm. In this part, we show the reconstruction of proposed SGFRWT by using the fast adjoint algorithm in Section 4.1.

The SGFRWT is an overcomplete transform since it maps an input signal \mathbf{f} of size N to the $N(J+1)$ fractional wavelets coefficients

$$\mathbf{c} = \mathbf{W}_\theta \mathbf{f}. \quad (63)$$

The pseudoinverse of the above equation can be obtained by solving the following square matrix equation:

$$(\mathbf{W}_\theta^* \mathbf{W}_\theta) \mathbf{f} = \mathbf{W}_\theta^* \mathbf{c}. \quad (64)$$

Equation (64) can be solved by the conjugate gradients method [79], whose computation in each step is dominated by applying $\mathbf{W}_\theta^* \mathbf{W}_\theta$ to a single vector. What is more, $\mathbf{W}_\theta^* \mathbf{W}_\theta \mathbf{f}$ can be fast approximated by $\tilde{\mathbf{W}}_\theta^* \tilde{\mathbf{W}}_\theta \mathbf{f}$ shown in (62). Therefore, we can reconstruct input signal \mathbf{f} from \mathbf{c} .

5. Application Examples

Similar to [44], in this section, we will show several application examples of the SGFRWT on different real and synthetic datasets by modifying the GSPBOX toolbox [80]. The first and the third experiments are implemented in MATLAB programming language on a PC machine, which sets up Microsoft Windows 7 operating system and has an Intel® Core™ i7-2600 CPU with speed of 3.40 GHz and 16 GB RAM. The second experiment is implemented using PyTorch on a PC machine, which sets up Ubuntu 16.04 operating system and has an Intel® Core™ i7-4790 CPU with speed of 4.00 GHz and 32 GB RAM and has also two NVIDIA GTX1080-Ti GPUs.

5.1. SGFRWT Design Details. We should design two kernels in SGFRWT. One is the wavelet kernel $g(x)$, and the other is the scaling function $h(x)$.

For the wavelet kernel $g(x)$, we choose the same cubic spline in [44] as follows:

$$g(x) = \begin{cases} x_1^{-\alpha} x^\alpha, & \text{for } x < x_1, \\ s(x), & \text{for } x_1 \leq x \leq x_2, \\ x_2^\beta x^{-\beta}, & \text{for } x > x_2, \end{cases} \quad (65)$$

where α and β are integers and they are used to determine the decay rate of spline. x_1 and x_2 are boundary of transitions region. Note that g is normalized such that $g(x_1) = g(x_2) = 1$. The coefficients of the cubic polynomial $s(x)$ are determined by $s(x_1) = s(x_2) = 1$, $s'(x_1) = \alpha/x_1$, and $s'(x_2) = -\beta/x_2$. All of the examples in this paper were produced using $\alpha = \beta = 2$, $x_1 = 1$, and $x_2 = 2$; in this case, $s(x) = -5 + 11x - 6x^2 + x^3$. For the SGFRWT, we can easily substitute x in (65) by \mathbf{L}_θ shown in (24) to get $g(\mathbf{L}_\theta)$.

For the scaling function kernel $h(x)$, we take $h(x) = \rho \exp(-(x/0.6\lambda_{\min})^4)$ with $\rho = h(0)$.

5.2. SGFRWT Application Examples

5.2.1. SGFRWT on Synthetic Graph Data. In the first example, we perform the proposed SGFRWT on “Swiss roll,” which is a dataset whose points are sampled randomly from a two-dimensional (2D) manifold as follows:

$$\vec{x}(t_1, t_2) = \left(t_1 \frac{\cos(t_2)}{4\pi}, t_1, t_2 \frac{\sin(t_2)}{4\pi} \right), \quad (66)$$

$$-1 \leq t_1 \leq 1, \quad \pi \leq t_2 \leq 4\pi.$$

Figure 1 shows the Swiss roll data cloud. We take 500 points sampled uniformly on the manifold and then build a weighted graph, whose edge weight is defined as follows:

$$a_{i,j} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right). \quad (67)$$

Note that the definition of the edge weight is a critical issue in graph construction. The most direct way to compute the edge weight is based on a given (dis)similarity measure. In practice, the edge weight is generally defined using different measures for better interpretability. As discussed in [81], Gaussian kernel and its variants (i.e., heat kernel or RBF kernel) are one of the most popular schemes to assign edge weights for a graph. Therefore, we take the Gaussian kernel in (67) to calculate the edge weight to simplify our discussion in this paper. We perform SGFRWT with fractional order $\theta \in [0.1, 1.0]$ on Swiss roll data cloud with $J = 5$ wavelet scales, and the results of some fractional orders $\theta \in \{0.1, 0.5, 0.8, 1.0\}$ are shown in Figure 2. The last column of Figure 2 shows the results of performing SGWT [44] on Swiss roll, from which we can see that SGWT [44] searches for the optimal expression domain only along the dimension of scale. However, the proposed SGFRWT searches for the optimal expression domain not only along the dimension of scale but also along the dimension of fractional order. That is, compared with SGWT, the proposed SGFRWT can express Swiss roll in higher dimensions, which makes the expression space larger and is therefore more likely to obtain a better expressive domain for the original problem. Just like [31–33], the fractional order θ is a hyperparameter depending on the original problem and you should search for a suitable value of θ . For example, the fractional order $\theta = 0.4$ is an appropriate value for gland segmentation when using fractional wavelet scattering network [33].

As a comparison, we also give the results of the proposed spectral graph fractional Fourier transform (SGFRFT) and graph fractional Fourier transform (GFRFT) proposed in [74, 75] on the “Swiss roll” in Figure 3, respectively. As we can see from Figure 3, the results of the proposed SGFRFT are different from those of GFRFT in [74, 75]. Furthermore, both SGFRFT and GFRFT are global transform and are not easily to be used for local analysis. However, let us see the third column of Figure 2, that is, the results of SGFRWT with $J = 5$ and fractional order $\theta = 0.8$; the localization is more and more obvious along scale dimension. Comparing the third column of Figure 2 with the third column of Figure 3, we can

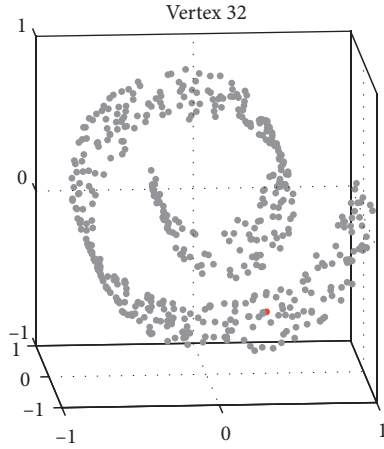


FIGURE 1: Spectral fractional graph wavelets on Swiss roll data cloud. The figure shows vertex at which SGFRWT is centered; that is, we will apply SGFRWT on the signal value of centered vertex.

see that, compared with SGFRFT and GFRFT, the proposed SGFRWT allows localization in fractional graph domain, which may lead to better local analysis in spectral graph domain. The experiment results of Figure 3 verify the local properties of SGFRWT shown in Appendix C (i.e., Lemmas C2–C4 and Theorem C5) to a certain extent.

5.2.2. SGFRWT on MNIST Database. In the second example, we will see that the proposed SGFRWT can be used as a data augmentation method in the preprocessing of convolutional neural networks, which require as many input images as possible. Because, as shown in the first example, the proposed SGFRWT allows deriving many images from the original image by considering several fractional orders. In the following, a series of experiments are performed on the MNIST dataset to verify the effectiveness of the proposed SGFRWT as a method of data augmentation.

We construct the graph for each image in MNIST dataset. The vertices of the graph are made up of pixels in the image and each pixel is connected with its vertical and horizontal neighbor pixel. The edge weight is calculated by a Gauss kernel weighting function [36] as follows:

$$w_{i,j} = \begin{cases} \exp\left(-\frac{[\text{dist}(i,j)]^2}{2\tau^2}\right), & \text{if } \text{dist}(i,j) \leq k, \\ 0, & \text{otherwise,} \end{cases} \quad (68)$$

where $w_{i,j}$ is the weight on the edge connection of the vertex v_i and v_j , and $\text{dist}(i,j)$ represents the distance between pixel i and pixel j .

In order to prove the effectiveness of SGWT and SGFRWT, we augment MNIST dataset by the following strategies: (i) The original MNIST database has a training set of 60,000 images and a testing set of 10,000 images, which are used as a control group. (ii) For each image of the MNIST training set, we perform SGWT [44] with $J=5$ wavelet scales

and generate a total of 360,000 images. The original 10,000 test images are still used as the testing set. (iii) For each image of the MNIST training set, we apply the proposed SGFRWT with $J=5$ wavelet scales and fractional order $\theta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, generating a total of 1,800,000 training images. The details of the generated four datasets are shown in Table 1. Figure 4 shows all sample images by processing an image of the digit “zero” by SGFRWT with $J=5$ wavelet scales and fractional order $\theta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$.

To evaluate the data augmentation performance of the proposed approach, we train and test a simple convolutional neural network, which is shown in Figure 5, using the three datasets mentioned above. The CNN model consists of 2 convolutional layers, 2 pooling layers, and a fully connected layer with a 10-way Softmax classifier. The inputs of the CNN are images of size 28×28 . The first convolutional layer has 32 kernels with size 5×5 , while the other layer has 64 kernels with size 3×3 . The Rectified Linear Units (ReLU) are chosen as the activation function. Every convolutional layer is followed by a nonoverlapping max-pooling with filter of size 2×2 . Stochastic gradient descent algorithm is employed to optimize the network with a learning rate of 0.001. When the network converges, we evaluate the classification accuracy on the testing set with an average value of 10 times. The last column of Table 2 shows the classification performance comparison of the three different datasets. As we can see from Table 2, if we use traditional data augmentation methods (flipping, rotating, and adding noise), then the recognition rate can achieve 98.41%. When the training dataset of MNIST is augmented by SGWT without traditional data augmentation, the recognition rate can increase to 98.71%. From the last row of Table 2, we can see that our proposed SGFRWT can further improve the accuracy.

5.2.3. Face Recognition by Fusing SGFRWT Features. In the third example, the application of SGFRWT to provide features for face recognition is investigated. We use four classical face databases, whose MATLAB formats are provided in [82, 83]. (1) *Yale Database* [84]. It contains 165 grayscale images (15 individuals, 11 images per individual). (2) *AR Database* [85]. We use a nonoccluded subset of AR database containing 700 images (50 male subjects and 50 female subjects, 7 images per subject). (3) *CMU PIE Face Database* [86]. We also use a nonoccluded subset of CMU PIE database which contains 11560 images (68 persons, 170 images per person). (4) *ORL Database* [87]. It contains 400 grayscale images (40 individuals, 10 images per individual). Note that all the images are resized to 32×32 pixels.

The experiment is carried out by the following three steps:

- (1) For each image of the four face databases, we apply the propo $J = \text{tr}(\mathbf{S}_b)/\text{tr}(\mathbf{S}_w)$,sed SGFRWT with

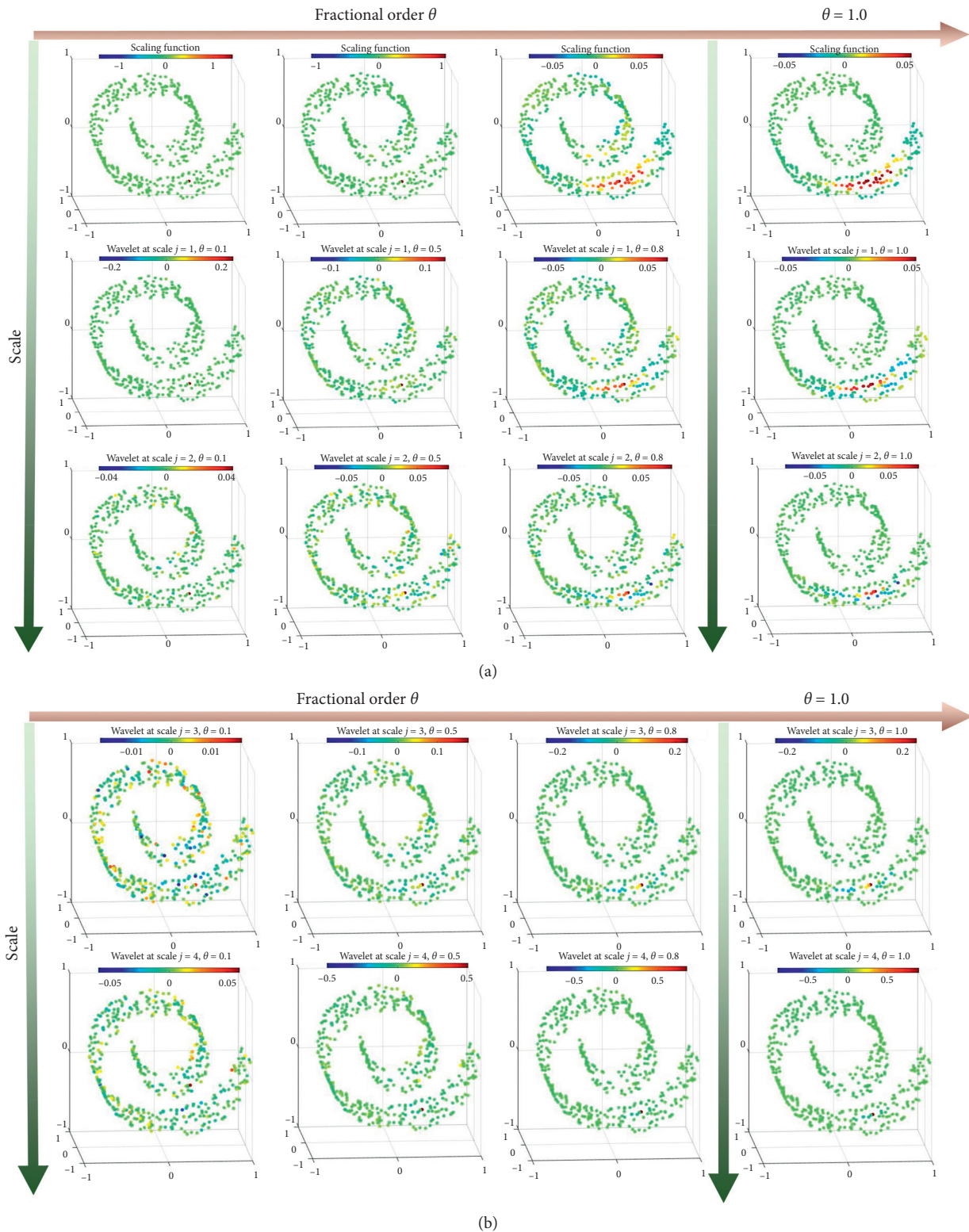


FIGURE 2: Spectral fractional graph wavelets on Swiss roll data cloud, with wavelet scale $J = 5$ and fractional order $\theta \in \{0.1, 0.5, 0.8, 1.0\}$. (a) Wavelet scale $j = 0, 1, 2$ and fractional order $\theta \in \{0.1, 0.5, 0.8, 1.0\}$. Note that $j = 0$ denotes the scaling function (b) wavelet scale $j = 3, 4$ and fractional order $\theta \in \{0.1, 0.5, 0.8, 1.0\}$.

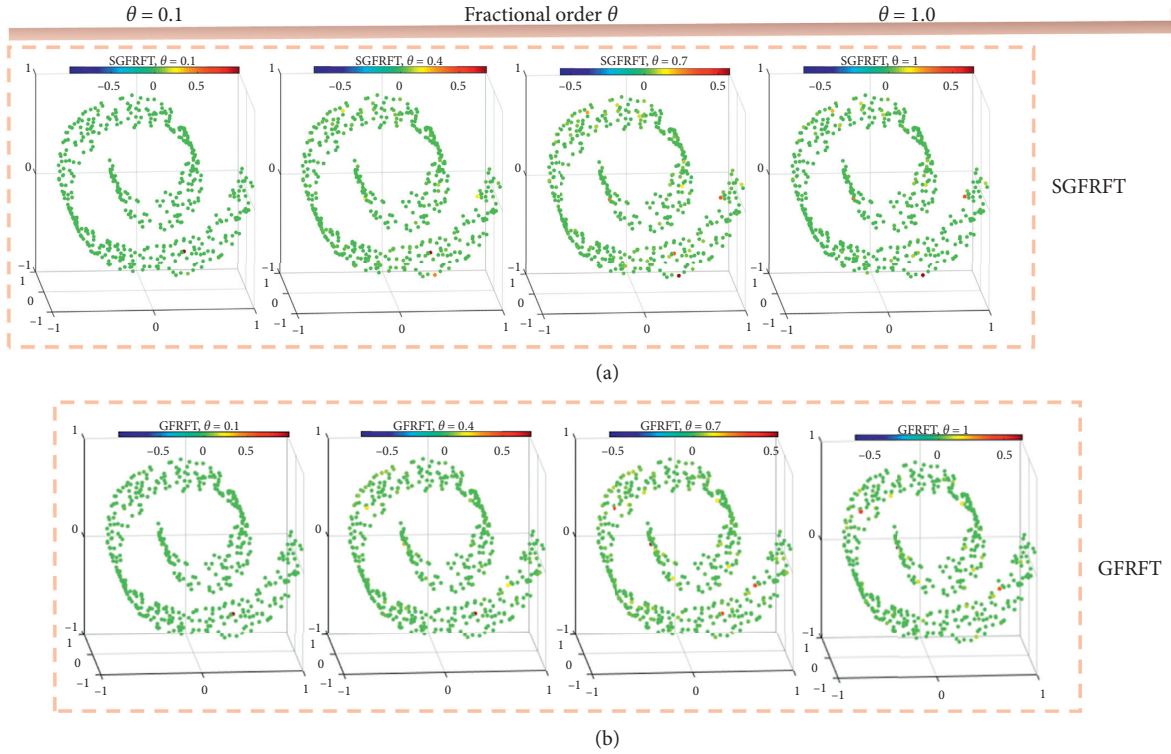


FIGURE 3: Spectral graph fractional Fourier transform (SGFRFT) and graph fractional Fourier transform (GFRFT) [74, 75] on Swiss roll data with fractional order $\theta \in \{0.1, 0.4, 0.7, 1.0\}$.

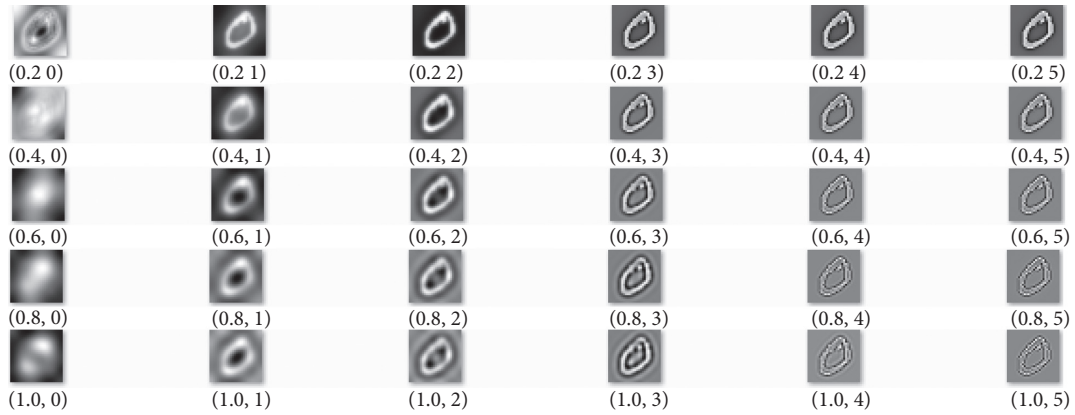


FIGURE 4: Spectral graph fractional wavelets on an image "zero," with $J = 5$ wavelet scales and fractional order $\theta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. (θ, j, θ) denotes the fractional order; j denotes the scale.

wavelet scale $J = 10$ and fractional order $\theta \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, obtaining 100 times the images of original datasets.

- (2) We then extract the SGFRWT features of three fractional orders which have the highest trace ration [88] as follows:

$$J = \frac{\text{tr}(\mathbf{S}_b)}{\text{tr}(\mathbf{S}_w)}, \quad (69)$$

where $\text{tr}(\cdot)$ denotes the trace. \mathbf{S}_b and \mathbf{S}_w are between-class scatter matrix and the within-class scatter matrix, respectively,

$$\begin{aligned} \mathbf{S}_b &= \sum_{i=1}^c N_i (\mathbf{A} - \mathbf{A}_i)^T, \\ \mathbf{S}_w &= \sum_{i=1}^c N_i (\mathbf{A}_i - |\mathbf{x}_{\theta, j}|) (\mathbf{A}_i - |\mathbf{x}_{\theta, j}|)^T, \end{aligned} \quad (70)$$

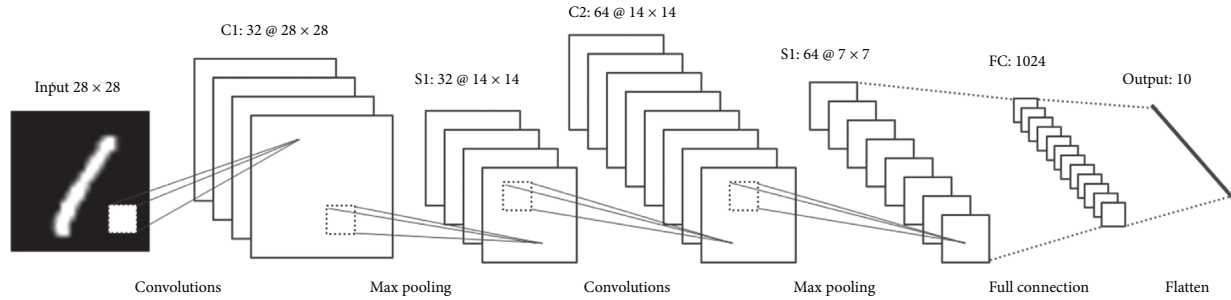


FIGURE 5: The convolutional neural network used in our experiment.

TABLE 2: Dataset generation implementation details. Comparison of classification performance of three different datasets with three different augmentation methods on the same CNN network.

Dataset	Augmentation details of training set	Validation set	Accuracy (%)
(i)	MNIST training set augmented with flipping, rotating, and adding noise.	Original MNIST testing set	98.41
(ii)	MNIST training set augmented with SGWT	Original MNIST testing set	98.71
(iii)	MNIST training set augmented with SGFRWT	Original MNIST testing set	98.86

TABLE 3: The recognition results of the proposed SGFRWT feature, the SGWT feature, and the baseline on four face databases (Yale, AR, ORL, and CMU PIE). We randomly selected several images per individual as training set and the remaining images as testing set. “*i* train” means that we randomly choose *i* images in each class for training. The accuracy is the average results of ten random testing experiments.

Dataset	Number of categories	Construction details	Baseline (SVM) (%)	SGWT + SVM (%)	The proposed SGFRWT + SVM (%)
<i>Yale: 165 grayscale images (15 individuals, 11 images per individual)</i>	15	1 train, 10 test each class	91.93	99.27	100
	15	2 train, 9 test each class	93.26	99.48	100
	15	3 train, 8 test each class	93.42	99.86	100
	15	4 train, 7 test each class	94.57	100	100
	15	5 train, 6 test each class	95.56	100	100
<i>AR: 700 images (50 male subjects and 50 female subjects, 7 images per subject)</i>	100	1 train, 6 test each class	49.53	90.92	96.52
	100	2 train, 5 test each class	70.78	93.24	98.14
	100	3 train, 4 test each class	78.53	96.58	98.4
<i>ORL: 400 grayscale images (40 individuals, 10 images per individual)</i>	40	1 train, 9 test each class	93.94	98.16	99.86
	40	2 train, 8 test each class	97	98.5	99.81
	40	3 train, 7 test each class	97.72	99.29	99.89
	40	4 train, 6 test each class	97.83	99.33	99.83
	40	5 train, 5 test each class	98	99.35	100
<i>CMU PIE: 11560 images (68 persons, 170 images per person)</i>	68	1 train, 169 test each class	15.1	77.9	81.13
	68	5 train, 165 test each class	40.9	89.22	90.09
	68	10 train, 160 test each class	52.45	91.11	91.95
	68	15 train, 155 test each class	60.28	91.14	92.36
	68	20 train, 150 test each class	64.81	91.67	92.7

where the superscript T denotes transposition. C denotes the number of classes. \mathbf{A} is the mean of \mathbf{A}_i and \mathbf{A}_i is the mean image of class X_i and N_i is the number of images in class X_i . $|\mathbf{X}_{(\theta, j)}|$ denotes the modulus of SGFRWT coefficient with the fractional order θ and the wavelet scale j .

- (3) The SGFRWT features of three fractional orders are fused by canonical correlation analysis (CCA) [89] to obtain the feature vector, whose dimension is then reduced by principal component analysis (PCA) and Linear Discriminant Analysis (LDA). Finally, we feed the feature vector to the support vector machine (SVM).

We conduct experiments on 18 different datasets which are constructed by four classical face databases mentioned above. Each set of datasets is compared with three different methods (baseline, SGWT, and SGFRWT), and we evaluate the classification accuracy on the testing set with an average value of 10 times. The recognition comparison results of the proposed SGFRWT feature building, the SGWT feature, and the baseline (only PCA + LDA + SVM used) on four face databases are shown in Table 3, from which we can see that the proposed SGFRWT feature building approach can achieve promising results and perform better than SGWT feature and also the SVM baseline.

6. Conclusion

This paper investigates the issue of extension of spectral graph wavelet transform (SGWT) to fractional domain. The main contributions of this paper can be summarized as follows. (1) A novel transform named spectral graph fractional wavelet transform (SGFRWT) is defined. (2) A Fourier series approximation-based fast algorithm for SGFRWT is derived and implemented since the SGFRWT includes complex domain computations compared to SGWT. (3) Applications of SGFRWT to synthetic and real datasets are also given to highlight its potential usefulness. In summary, the proposed fractional spectral graph wavelets provide a new choice for the graph signal processing. Further research may include the extension of the proposed SGFRWT for dealing with the directed graphs [40, 41] and the extension of the idea of SGFRWT to critically sampled graph wavelets like GraphBio [46].

Appendix

A. Definition of SGFRWT by Using the Idea of Traditional FRWT

According to (8), we can define the SGFRWT in (37) in another form by using the idea of traditional FRWT as follows:

$$W_f(\theta, s, n) = (T_{g_\theta}^s f)(n) = \sum_{m=1}^N f(m) \psi_{\theta, s, n}^*(m) = \langle f, \psi_{\theta, s, n} \rangle, \quad n = 1, \dots, N, \quad (\text{A.1})$$

where

$$T_{g_\theta}^s = e^{-j/2(m^2 - n^2 - (m-n/s)^2) \cot \theta} g(s\mathbf{L}), \quad (\text{A.2})$$

$$\psi_{\theta, s, n}(m) = e^{-j/2(m^2 - n^2 - (m-n/s)^2) \cot \theta} \sum_{\ell=0}^{N-1} g(s\lambda_\ell) \chi_\ell(m) \chi_\ell^*(n). \quad (\text{A.3})$$

From (A.3), we can see that this definition has two defects:

- (1) The basis of SGFRWT $\psi_{\theta, s, n}(m)$ is still based on $\chi_\ell(n)$; however, we want to define the basis of SGFRWT $\psi_{\theta, s, n}(m)$ based on a new function, for example, $\gamma_\ell(n)$. If we use the definition of (A.3), then the characteristic of $\psi_{\theta, s, n}(m)$ will be very similar to $\psi_{s, n}(m)$ in (22) since they are based on the same function $\chi_\ell(n)$.
- (2) We want to define the basis of SGFRWT $\psi_{\theta, s, n}(m)$ which changes directly with λ_ℓ^θ ; that is, the parameter θ should have an effect on λ_ℓ directly.

Therefore, we do not use (A.1)–(A.3) to define the SGFRWT.

B. Definition of SGFRWT by Using Another Graph Fractional Laplacian Operator L^θ

Different from the definition in (24), the graph fractional Laplacian operator can also be defined as follows:

$$\mathbf{L}^\theta = (\chi \Lambda \chi^H)^\theta = \chi \Lambda^\theta \chi^H. \quad (\text{B.1})$$

Comparing (B.1) and (13), we can see that the decomposition of \mathbf{L} and \mathbf{L}^θ shares the same χ . That is, if we define \mathbf{L}^θ as the graph fractional Laplacian operator, then, similar to (21), we can define the spectral graph fractional wavelet transform (SGFRWT) operator $T_{g_\theta}^s$ as follows:

$$W_f(\theta, s, n) = T(T_{g_\theta}^s f)(n) = \sum_{m=1}^N f(m) \psi_{\theta, s, n}^*(m) \langle f, \psi_{\theta, s, n} \rangle, \quad n = 1, \dots, N, \quad (\text{B.2})$$

where

$$T_{g_\theta}^s = g(s\mathbf{L}^\theta), \quad (\text{B.3})$$

$$\psi_{\theta, s, n}(m) = \sum_{\ell=0}^{N-1} g(s\lambda_\ell^\theta) \chi_\ell(m) \chi_\ell^*(n), \quad m = 1, 2, \dots, N. \quad (\text{B.4})$$

From (B.4), we can see that this definition has one defect: the basis of SGFRWT $\psi_{\theta, s, n}(m)$ is based on $\chi_\ell(n)$; however, we want to define the basis of SGFRWT $\psi_{\theta, s, n}(m)$ based on a new function, for example, $\gamma_\ell(n)$. If we use the definition of (B.4), then the characteristic of $\psi_{\theta, s, n}(m)$ will be very similar

to $\psi_{s,n}(m)$ in (21) since they are based on the same function $\chi_\ell(n)$.

Therefore, we do not use (B.1)–(B.4) to define the SGFRWT.

C. Some Properties of SGFRWT

In this appendix, we give some properties of proposed SGFRWT without proof, since the proof of these properties is very similar to that of those properties of SGWT in [44].

Lemma C.1. *If the SGFRWT kernel g_θ satisfies the following admissibility condition:*

$$\int_0^\infty \frac{g^2(x)}{x} dx = C_g < \infty, \quad (C.1)$$

and $g(0) = 0$, then

$$\frac{1}{C_g} \sum_{n=1}^N \int_0^\infty W_f(\theta, s, n) \psi_{\theta,s,n}(m) \frac{ds}{s} = f^\#(m), \quad (C.2)$$

where $\mathbf{f}^\# = \mathbf{f} - \langle \gamma_0, \mathbf{f} \rangle \gamma_0$. In particular, the complete reconstruction is given by $\mathbf{f} = \mathbf{f}^\# + \hat{f}(0) \gamma_0$.

Lemma c.1 shows that the mean of f may not be recovered from the zero-mean fractional graph wavelets. Lemma c.1 is a generalization of Lemma 5.1 in [44].

Lemma C.2. *Let \mathcal{G} be a weighted graph, L_θ the graph fractional Laplacian operator, and $t > 0$ an integer. For any two vertices m and n , if $d_{\mathcal{G}}(m, n) > t$ then $((L_\theta)^t)_{m,n} = 0$.*

Note that $d_{\mathcal{G}}(m, n)$ is the shortest-path distance, i.e., the minimum number of edges for any paths connecting m and n . Lemma c.2 shows the localization result for integer powers of the fractional Laplacian L_θ . Lemma c.2 is a generalization of Lemma 5.2 in [44].

Lemma C.3. *Let $\psi_{\theta,s,n} = T_{g_\theta}^s \delta_n$ and $\tilde{\psi}_{\theta,s,n} = T_{\tilde{g}_\theta}^s \delta_n$ be the fractional wavelets at scale s generated by the fractional kernels g_θ and \tilde{g}_θ . If $|g(sr) - \tilde{g}(sr)| \leq M(s)$ for all $r \in [0, r_{N-1}]$, then $|\psi_{\theta,s,n}(m) - \tilde{\psi}_{\theta,s,n}(m)| \leq M(s)$ for each vertex m . Additionally, $\|\psi_{\theta,s,n} - \tilde{\psi}_{\theta,s,n}\|_2 \leq \sqrt{N}M(s)$.*

Lemma c.3 shows that if two fractional kernels g_θ and \tilde{g}_θ are close to each other in some sense, then the resulting graph fractional wavelets should be close to each other. Lemma 3.3 is a generalization of Lemma c.3 in [44].

Lemma C.4. *Let g_θ be $K+1$ times continuously differentiable, satisfying $g(0) = 0, g^{(i)}(0) = 0$ for all $i < K$, and $g^{(K)}(0) = C \neq 0$. Assume that there is some $st > 0$ such that $|g^{(K+1)}(r)| \leq B$ for all $r \in [0, str_{N-1}]$. Then, for $\tilde{g}(sr) = (C/K!)(sr)^K$, we have $M(s) = \sup_{r \in [0, r_{N-1}]} |g(sr) - \tilde{g}(sr)| \leq s^{K+1} r_{N-1}^{K+1} / (K+1)!$ for all $s < s'$.*

Lemma c.4 shows that if a fractional kernel g_θ has a zero of integer multiplicity at the origin, then g_θ can be

approximated by a single monomial for small scales. Lemma c.4 is a generalization of Lemma 5.4 in [44].

Theorem C.1. *Let \mathcal{G} be a weighted graph with fractional Laplacian L_θ . Let g_θ be a kernel satisfying the hypothesis of Lemma 3.4, with constants s' and B . Let m and n be vertices of \mathcal{G} such that $d_{\mathcal{G}}(m, n) > K$. Then, there exist constants D and s'' , such that $\psi_{\theta,s,n}(m) / \|\psi_{\theta,s,n}\| \leq Ds$ for all $s < \min(st, s'')$.*

Theorem c.1 shows that the localization of the fractional wavelet $\psi_{\theta,s,n}$ must include a renormalization factor in the limit of small scales. In general, $\psi_{\theta,s,n}(m) \rightarrow 0$ as $s \rightarrow 0$ for all m and n , due to the normalization chosen for the graph fractional wavelets. Theorem c.1 is a generalization of Theorem 5.5 in [44].

Theorem C.2. *Given a set of scales $\{s_j\}_{j=1}^J$, the set $F = \{\phi_{\theta,n}\}_{n=1}^N \cup \{\psi_{\theta,s_j,n}\}_{j=1}^J$ forms a frame with bounds A, B given by $A = \min_{r \in [0, r_{N-1}]} G(r)$ and $B = \max_{r \in [0, r_{N-1}]} G(r)$, where $G(r) = |h(r)|^2 + \sum_j |g(s_j r)|^2$.*

Note that the basic definition of a frame is as follows: Given a Hilbert space \mathcal{H} , a set of vectors $\Gamma_k \in \mathcal{H}$ form a frame with frame bounds A and B if the inequality $A\|f\|^2 \leq \sum |\langle f, \Gamma_k \rangle|^2 \leq B\|f\|^2$ holds for all $f \in \mathcal{H}$. Theorem C.6 shows that the numerical stability of recovering the vector f from the inner product $\langle f, F_k \rangle$ depends on the frame bounds A and B . Theorem c.2 is a generalization of Theorem 5.6 in [44].

Data Availability

The data used in Section 5.2.1 are synthetic graph data. The data used in Section 5.2.2 are MNIST dataset, which are downloaded from the homepage of Yann LeCun (<http://yann.lecun.com/exdb/mnist/>). The data used in Section 5.2.3 are supported in [82, 83].

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61876037, 31800825, 61871117, 61871124, and 61773117, in part by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2018ZX10201002-003, in part by the Short-Term Recruitment Program of Foreign Experts under Grant WQ20163200398, and in part by INSERM under the Grant IAL.

References

- [1] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press, Cambridge, MA, USA, 3rd edition, 2008.

- [2] V. Namias, "The fractional order Fourier transform and its application to quantum mechanics," *IMA Journal of Applied Mathematics*, vol. 25, no. 3, pp. 241–265, 1980.
- [3] L. B. Almeida, "The fractional Fourier transform and time-frequency representations," *IEEE Transactions on Signal Processing*, vol. 42, no. 11, pp. 3084–3091, 1994.
- [4] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform with Applications in Optics and Signal Processing*, Wiley, New York, NY, USA, 2001.
- [5] E. Sejdić, I. Djurović, and L. J. Stanković, "Fractional Fourier transform as a signal processing tool: an overview of recent developments," *Signal Processing*, vol. 91, no. 6, pp. 1351–1369, 2011.
- [6] M. H. Annaby, H. A. Ayad, M. A. Rushdi, and E. A. Nehary, "Difference operators and generalized discrete fractional transforms in signal and image processing," *Signal Processing*, vol. 151, pp. 1–18, 2018.
- [7] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "New parametric discrete Fourier and Hartley transforms, and algorithms for fast computation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 3, pp. 562–575, 2011.
- [8] R. Tao, B. Deng, and Y. Wang, "Research progress of the fractional Fourier transform in signal processing," *Science in China Series F*, vol. 49, no. 1, pp. 1–25, 2006.
- [9] A. Bultheel and H. Martínez-Sulbaran, "Recent developments in the theory of the fractional Fourier and linear canonical transforms," *Bulletin of the Belgian Mathematical Society—Simon Stevin*, vol. 13, no. 5, pp. 971–1005, 2007.
- [10] H. M. Ozaktas, O. Arikan, M. A. Kutay, and G. Bozdagt, "Digital computation of the fractional Fourier transform," *IEEE Transactions on Signal Processing*, vol. 44, no. 9, pp. 2141–2150, 1996.
- [11] C. Candan, M. A. Kutay, and H. M. Ozaktas, "The discrete fractional Fourier transform," *IEEE Transactions on Signal Processing*, vol. 48, no. 5, pp. 1329–1337, 2000.
- [12] S. C. Pei, M. H. Yeh, and C. C. Tseng, "Discrete fractional Fourier transform based on orthogonal projections," *IEEE Transactions on Signal Processing*, vol. 47, no. 5, pp. 1335–1348, 1999.
- [13] S.-C. Pei, W.-L. Hsue, and J.-J. Ding, "Discrete fractional Fourier transform based on new nearly tridiagonal commuting matrices," *IEEE Transactions on Signal Processing*, vol. 54, no. 10, pp. 3815–3828, 2006.
- [14] S. C. Pei, J. J. Ding, W. L. Hsue, and K. W. Chang, "Generalized commuting matrices and their eigenvectors for DFTs, offset DFTs, and other periodic operations," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3891–3904, 2008.
- [15] A. Bultheel and H. E. Martínez Sulbaran, "Computation of the fractional Fourier transform," *Applied and Computational Harmonic Analysis*, vol. 16, no. 3, pp. 182–202, 2004.
- [16] R. Tao, B. Deng, W.-Q. Zhang, and Y. Wang, "Sampling and sampling rate conversion of band limited signals in the fractional Fourier transform domain," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 158–171, 2008.
- [17] R. Tao, F. Zhang, and Y. Wang, "Research progress on discretization of fractional Fourier transform," *Science in China Series F: Information Sciences*, vol. 51, no. 7, p. 859, 2008.
- [18] S. C. Pei and J. J. Ding, "Fractional cosine, sine, and Hartley transforms," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1661–1680, 2002.
- [19] C. C. Tseng, "Eigenvalues and eigenvectors of generalized DFT, generalized DHT, DCT-IV and DST-IV matrices," *IEEE Transactions on Signal Processing*, vol. 50, no. 4, pp. 866–877, 2002.
- [20] G. Cariolaro, T. Erseghe, and P. Kraniuskas, "The fractional discrete cosine transform," *IEEE Transactions on Signal Processing*, vol. 50, no. 4, pp. 902–911, 2002.
- [21] S. C. Pei and W. L. Hsue, "Tridiagonal commuting matrices and fractionalizations of DCT and DST matrices of types I, IV, V, and VIII," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2357–2369, 2008.
- [22] X. Liu, G. Han, J. Wu, Z. Shao, G. Coatrieux, and H. Shu, "Fractional Krawtchouk transform with an application to image watermarking," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1894–1908, 2017.
- [23] R. Tao, Y.-L. Li, and Y. Wang, "Short-time fractional Fourier transform and its applications," *IEEE Transactions on Signal Processing*, vol. 58, no. 5, pp. 2568–2580, 2010.
- [24] D. Mendlovic, Z. Zalevsky, D. Mas, J. García, and C. Ferreira, "Fractional wavelet transform," *Applied Optics*, vol. 36, no. 20, pp. 4801–4806, 1997.
- [25] J. Shi, Y. Chi, and N. Zhang, "Multichannel sampling and reconstruction of bandlimited signals in fractional Fourier domain," *IEEE Signal Processing Letters*, vol. 17, no. 11, pp. 909–912, 2010.
- [26] J. Shi, N. Zhang, and X. Liu, "A novel fractional wavelet transform and its applications," *Science China Information Sciences*, vol. 55, no. 6, pp. 1270–1279, 2012.
- [27] A. Prasad, S. Manna, A. Mahato, and V. K. Singh, "The generalized continuous wavelet transform associated with the fractional Fourier transform," *Journal of Computational and Applied Mathematics*, vol. 259, pp. 660–671, 2014.
- [28] H. Dai, Z. Zheng, and W. Wang, "A new fractional wavelet transform," *Communications in Nonlinear Science and Numerical Simulation*, vol. 44, pp. 19–36, 2017.
- [29] H. M. Srivastava, K. Khatterwani, and S. K. Upadhyay, "A certain family of fractional wavelet transformations," *Mathematical Methods in the Applied Sciences*, vol. 42, no. 9, pp. 3103–3122, 2019.
- [30] J. Shi, X. Liu, W. Xiang, M. Han, and Q. Zhang, "Novel fractional wavelet packet transform: theory, implementation, and applications," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4041–4054, 2020.
- [31] I. S. Yetik, M. A. Kutay, and H. M. Ozaktas, "Image representation and compression with the fractional Fourier transform," *Optics Communications* 4–6, vol. 197, pp. 275–278, 2001.
- [32] J. Liu, S. Chen, and X. Tan, "Fractional order singular value decomposition representation for face recognition," *Pattern Recognition*, vol. 41, no. 1, pp. 378–395, 2008.
- [33] L. Liu, J. Wu, D. Li, L. Senhadji, and H. Shu, "Fractional wavelet scattering network and applications," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 2, pp. 553–563, 2019.
- [34] B. Hennelly and J. T. Sheridan, "Fractional Fourier transform-based image encryption: phase retrieval algorithm," *Optics Communications*, vol. 226, no. 1–6, pp. 61–80, 2003.
- [35] I. Djurovic, S. Stankovic, and I. Pitas, "Digital watermarking in the fractional Fourier transformation domain," *Journal of Network and Computer Applications*, vol. 24, no. 2, pp. 167–173, 2001.
- [36] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

- [37] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, 2016.
- [38] I. Jestrovic, J. L. Coyle, and E. Sejdic, "A fast algorithm for vertex-frequency representations of signals on graphs," *Signal Processing*, vol. 131, pp. 483–491, 2017.
- [39] L. L. Magoarou, R. Gribonval, and N. Tremblay, "Approximate fast fourier transforms on graphs via multi-layer sparse approximations," 2016, <https://arxiv.org/abs/1612.04542>.
- [40] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.
- [41] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.
- [42] R. V. Mendes, H. C. Mendes, and T. Araujo, "Signals on graphs: transforms and tomograms," *Physica A: Statistical Mechanics and Its Applications*, vol. 450, pp. 1–17, 2016.
- [43] S. Chen, R. Varma, A. Singh, and J. Kovacevic, "Signal representations on graphs: tools and applications," 2015, <https://arxiv.org/abs/1512.05406>.
- [44] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
- [45] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2786–2799, 2012.
- [46] S. K. Narang and A. Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4673–4685, 2013.
- [47] D. I. Shuman, M. J. Faraji, and P. Vandergheynst, "A multiscale pyramid transform for graph signals," *IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 2119–2134, 2016.
- [48] A. Rezapour, A. Ortega, and M. Sahimi, "Upscaling of geological models of oil reservoirs with unstructured grids using lifting-based graph wavelet transforms," *Transport in Porous Media*, vol. 127, no. 3, pp. 661–684, 2019.
- [49] D. K. Hammond, P. Vandergheynst, and R. Gribonval, *The Spectral Graph Wavelet Transform: Fundamental Theory and Fast Computation. Vertex-Frequency Analysis of Graph Signals*, pp. 141–175, Springer, Cham, Switzerland, 2019.
- [50] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Soc., Providence, RI, USA, 1997.
- [51] G. Cheung, W.-T. Su, Y. Mao, and C.-W. Lin, "Robust semisupervised graph classifier learning with negative edge weights," *IEEE Transactions on Signal and Information Processing Over Networks*, vol. 4, no. 4, pp. 712–726, 2018.
- [52] W. T. Su, G. Cheung, and C. W. Lin, "Graph Fourier transform with negative edges for depth image coding," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1682–1686, Beijing, China, September 2017.
- [53] M. Puschel and J. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3572–3585, 2008.
- [54] M. Puschel and J. M. F. Moura, "Algebraic signal processing theory: 1-D space," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3586–3599, 2008.
- [55] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5445–5449, Florence, Italy, May 2013.
- [56] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3864–3868, Florence, Italy, May 2014.
- [57] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovacevic, "Signal recovery on graphs: variation minimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4609–4624, 2015.
- [58] S. Chen, A. Sandryhaila, G. Lederman et al., "Signal inpainting on graphs via total variation minimization," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8267–8271, Florence, Italy, May 2014.
- [59] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2432–2444, 2015.
- [60] X. Wang, M. Wang, and Y. Gu, "A distributed tracking algorithm for reconstruction of graph signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 728–740, 2015.
- [61] S. Chen, F. Cerda, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovacevic, "Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring," *IEEE Transactions on Signal Processing*, vol. 62, no. 11, pp. 2879–2893, 2014.
- [62] V. N. Ekambaram, G. Fanti, B. Ayazifar, and K. Ramchandran, "Wavelet-regularized graph semi-supervised learning," in *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 423–426, Austin, TX, USA, December 2013.
- [63] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.
- [64] D. Thanou, D. I. Shuman, and P. Frossard, "Learning parametric dictionaries for signals on graphs," *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3849–3862, 2014.
- [65] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.
- [66] L. Yi, H. Su, X. Guo, and L. Guibas, "Syncspeccnn: synchronized spectral CNN for 3 d shape segmentation," 2016, <https://arxiv.org/abs/1612.00606>.
- [67] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," 2015, <https://arxiv.org/abs/1501.06297>.
- [68] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3844–3852, Barcelona, Spain, December 2016.
- [69] F. P. Such, S. Sah, M. Dominguez et al., "Robust spatial filtering with graph convolutional neural networks," 2017, <https://arxiv.org/abs/1703.00792>.
- [70] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, <https://arxiv.org/abs/1609.02907>.
- [71] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," 2017, <https://arxiv.org/abs/1704.02901>.

- [72] Y. Hechtlinger, P. Chakravarti, and J. Qin, "Convolutional neural networks generalization utilizing the data graph structure," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, April 2017.
- [73] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.
- [74] Y. Q. Wang, B. Z. Li, and Q. Y. Cheng, "The fractional Fourier transform on graphs," in *Proceedings of the APSIPA Annual Summit and Conference*, pp. 105–110, Kuala Lumpur, Malaysia, December 2017.
- [75] Y. Q. Wang and B. Z. Li, "The fractional fourier transform on graphs: sampling and recovery," in *Proceedings of the 14th IEEE International Conference on Signal Processing (ICSP)*, pp. 1103–1108, Beijing, China, August 2018.
- [76] N. J. Higham and L. Lin, "A schur-padé algorithm for fractional powers of a matrix," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 3, pp. 1056–1078, 2011.
- [77] N. J. Higham and L. Lin, "An improved Schur-Padé algorithm for fractional powers of a matrix and their Fréchet derivatives," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 1341–1360, 2013.
- [78] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Courier Corporation, Chelmsford, MA, USA, 2001.
- [79] K. Grochenig, "Acceleration of the frame algorithm," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3331–3340, 1993.
- [80] N. Perraudin, J. Paratte, D. Shuman et al., "A toolbox for signal processing on graphs," 2014, <https://arxiv.org/abs/1408.5781>.
- [81] L. Qiao, L. Zhang, S. Chen, and D. Shen, "Data-driven graph construction and graph learning: a review," *Neurocomputing*, vol. 312, pp. 336–351, 2018.
- [82] A. Krizhevsky, "Learning multiple layers of features from tiny images," Technique Report, pp. 1–58, Department of Computer Science, University of Toronto, Toronto, Canada, 2009.
- [83] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3608–3614, 2006.
- [84] Yale University, *Yale Database*, Yale University, New Haven, CT, USA, 1997.
- [85] A. M. Martinez and R. Benavente, "The AR face database CVC Technical Report," 1998, <http://www2.ece.ohio-state.edu/%7Ealeix/ARdatabase.html>.
- [86] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, DC, USA, May 2002.
- [87] Cambridge University Computer Laboratory, *ORL Datasets*, Cambridge University Computer Laboratory, Cambridge, UK <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>.
- [88] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [89] M. Haghghat, M. Abdel-Mottaleb, and W. Alhalabi, "Fully automatic face normalization and single sample face recognition in unconstrained Environments," *Expert Systems With Applications*, vol. 47, pp. 23–34, 2016.