Deformable Protein Shape Classification Based on Deep Learning, and the Fractional Fokker–Planck and Kähler–Dirac Equations

Eric Paquet, Herna L. Viktor, Kamel Madi, and Junzheng Wu

Abstract—The classification of deformable protein shapes, based solely on their macromolecular surfaces, is a challenging problem in protein–protein interaction prediction and protein design. Shape classification is made difficult by the fact that proteins are dynamic, flexible entities with high geometrical complexity. In this paper, we introduce a novel description for such deformable shapes. This description is based on the bifractional Fokker–Planck and Dirac–Kähler equations. These equations analyse and probe protein shapes in terms of a scalar, vectorial and non-commuting quaternionic field, allowing for a more comprehensive description of the protein shapes. An underlying non-Markovian Lévy random walk establishes geometrical relationships between distant regions while recalling previous analyses. Classification is performed with a multiobjective deep hierarchical pyramidal neural network, thus performing a multilevel analysis of the description. Our approach is applied to the SHREC'19 dataset for deformable protein shapes classification and to the SHREC'16 dataset for deformable partial shapes classification, demonstrating the effectiveness and generality of our approach.

Index Terms—Classification, fractional, Fokker–Planck, Dirac–Kähler, deep learning, wavelets, deformable shape, Lévy distribution, macromolecular surface, multiobjective optimisation, partial shape, protein, non-Markovian process, pyramidal neural network

1 Introduction

Proteins perform many functions at the cellular level, including metabolic reaction catalysis, DNA replication, stimulus response, molecular transportation, and organism structure [1]. Proteins consist of linear chains of amino acid residues called polypeptides. The sequence of amino acids in a protein, determined by genetics, distinguishes one protein from another. The amino acid sequence allows a protein to form in a specific 3-D structure that determines its activity, known as a native conformation. Proteins may shift into different conformations to perform their functions, making some structures available for a reaction and hiding others. The transitions between shape states are called conformational changes. The structure is generally stabilized by nonlocal interactions, most commonly the formation of a hydrophobic core, but also through salt bridges, hydrogen bonds, disulfide bonds, and even posttranslational modifications [1].

As proteins are articulated or otherwise deformable, research is actively under way to develop invariant descriptions for such deformable shapes; various isometrically invariant descriptions have been proposed [2]. An isometric transformation, also known as a non-elastic deformation, is a transformation that preserves geodesic distances between points [2]. These transformations play a key role in shape analysis since several deformations are either isometric as with articulated objects, or quasi-isometric, as in the case of proteins [3]. Here, the geodesic distance refers to the length of the shortest path in between two points, with the path entirely within the manifold defined by the shape. Most procedures for isometrically invariant shape description rely either on geodesic distance or on physics-based modelling [2]. When using physics-based modelling, the shape under consideration is assimilated to a manifold over which some physical field (such as heat) propagates according to its associated physical process [4]. Analysing the field distribution at various time intervals allows for the definition of multiresolution features which, after dimensional reduction, constitute an invariant and, hopefully, informative and discriminative descriptor (also known as signature) [5]. In this paper, a new approach, based on the bifractional Fokker–Planck and bifractional Dirac–Kähler equations, is proposed. The standard heat equation may be formulated in terms of a Markovian–Gaussian random walk [6], which means that the exploration, analysis and understanding of a shape are constrained by the very nature of this walk. In turn, this implies that only small steps are permitted (the Gaussian exponentially suppresses large steps) and the walk is memoryless [6]. Consequently, the exploration is intrinsically local (due to the small steps) and non-contextual (small steps and memoryless). Our objective is to perform a contextual analysis of a shape so as to overcome the limitations mentioned above. Our method allows distant

- *Eric Paquet is with National Research Council, Ottawa, ON K1A 0R6, Canada. E-mail: eric.paquet@nrc-cnrc.gc.ca.*
- *Herna L. Viktor and Junzheng Wu are with the University of Ottawa, Ottawa, ON K1N 6N5, Canada. E-mail: {hviktor, jwu220}@uottawa.ca.*
- *Kamel Madi is with the Umanis, Research and Innovation, 92300 Levallois-Perret, France. E-mail: kmadi@umanis.com.*

Manuscript received 1 Apr. 2021; revised 13 Nov. 2021; accepted 18 Jan. 2022. Date of publication 27 Jan. 2022; date of current version 5 Dec. 2022.
(Corresponding author: Junzheng Wu.)
Recommended for acceptance by O. Winther.
Digital Object Identifier no. 10.1109/TPAMI.2022.3146796

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

points to be put into relation by large steps while the exploration proceeds according to a non-Markovian process (i.e. with memory). The aim is to establish an interrelation between more distant simplices (vertices, edges, triangles), as well as a geometrical context [7]. Indeed, the context is lost when steps are too small, or if intermediary steps are not remembered. Such an objective may be achieved by replacing the Gaussian transition probability with a Lévy distribution [6] and by substituting the temporal derivative with the fractional one. The result of these substitutions is the bifractional Fokker–Planck equation. The bifractional Fokker–Planck equation, like its non-fractional counterpart, is restricted to real scalar fields such as heat. If more complex fields are employed, complementary features may be extracted which either are inaccessible or at least inadequately probed by heat. For instance, heat may be substituted by probability whose propagation is described by the Schrödinger equation, from which a shape description may be inferred: the celebrated wave kernel signature (WKS) [8]. We propose complementing the Fokker–Planck equation using the Dirac-Kähler equation [9], [10], which is solved in terms of quaternions [11], a generalisation of complex numbers. Quaternion fields anticommute and interfere, as opposed to heat fields which do not. Therefore, their realms are distinct and complementary. In the present work, the differential operators associated with the Fokker–Planck (de Rham) and the Dirac–Kähler equations are expressed in terms of discrete differential calculus [12]. As for standard discrete operators, such as the Laplace–Beltrami operator [13], they constitute an approximation of the continuous operator but, in contrast to the latter, they are topologically exact which means that topological features, such as holes, and geometrical identities, such as Gauß's theorem [12] remain entirely valid. Discrete differential operators are defined solely in terms of discrete geometry. They involve, for instance, the incidence matrices associated with the various simplices: vertices, edges, and triangles. On the other hand, the Laplace–Beltrami operator is approximated with finite element methods, which inherently involves interpolations [13]. Such approximations may alter or even destroy topological features. Since topology is of paramount importance for shape description, we employ discrete differential calculus to define the discrete de Rham and Dirac–Kähler operators. Classification is performed with a new multiresolution and multiobjective deep neural network. Its architecture is inspired by the pyramid neural network [14], which is an extension of the Siamese neural network [15], but differs in many fundamental ways. For instance, the coupling between the two branches is only partial. Moreover, each branch is characterised by a block architecture which performs a multiresolution analysis of the input. A loss or cost function is associated with each resolution level. As a result, the learning process involves multiobjective or Pareto stochastic optimisation [16] which ensures that classification is properly performed at each resolution level. In addition, a learnable attention mechanism selects the best combination of features for classification from the shape description.

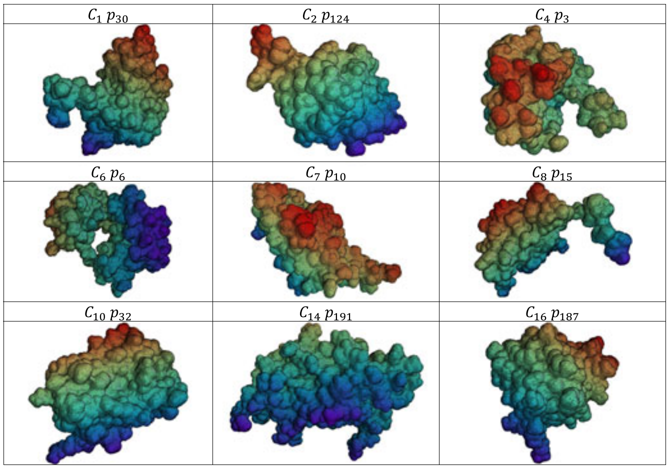

Fig. 1. Macromolecular surfaces sampled from the SHREC'19 dataset. Both the surface and the underlying mesh (graph) are shown.C_ip_j refers to protein j belonging to class i.

Our most important contributions are:

- a topologically exact formulation of discrete differential operators in terms of discrete differential calculus
- the replacement of the Laplace–Beltrami operator on an undirected graph by the de Rham operator on a directed graph
- a Lévy random walk and a non-Markovian fractional time derivative for shape contextual analysis with memory
- the bifractional Fokker–Planck equation, both for vertices and edges, for shape description
- the introduction of the Dirac–Kähler and a quaternion fields for shape description, and
- a multiobjective deep neural network for classification.

We deployed our system for macromolecular shape classification as described in the SHREC'19 Protein Shape Retrieval Context, some of which are illustrated in Fig. 1, and compared our results to the state of the art [17]. The SHREC'19 dataset consists of 5,298 proteins extracted from 211 entries from the Protein Data Bank [18]. They are classified according to the Structural Classification of Proteins-extended (SCOPe) [19] in order to evaluate our system's performance in terms of conformation retrieval. The task is made difficult by the fact that protein deformability is due both to their multiple degrees of freedom, and also to interaction with the surrounding environment: two proteins with a same conformation may not have the same shape.

The paper is organised as follows. Section 2 reviews background material and introduces differential forms, discrete differential calculus and the de Rham operator. The fractional Fokker–Planck equation is derived in Section 3, from the Lévy random walk. Non-Markovian processes (with memory) and fractional time derivatives are introduced in Section 4. This is followed, in Section 5, by the bifractional Fokker–Planck equation and its solution in terms of the heat kernel. The bifractional Dirac–Kähler equation, the Dirac-Kähler kernel, and the bifractional Planck–Dirac signature are described in Section 6. Bifractional Planck-Dirac wavelets are introduced in Section 7. The numerical evaluation of the bifractional Planck-Dirac wavelet signatures is addressed in Section 8. Section 9 describes a new multiresolution and multiobjective pyramid deep neural network for proteins classification. Our experimental results are reported in Sections 10 and 11, and

Section 12 concludes the paper by identifying promising directions for future work.

2 Background

2.1 Computer Vision

Non-Euclidean, geometrical structures, such as manifolds and graphs, are pervasive in numerous fields such as social media [20] and computer vision [21], [22], [23], [24]. Usually, 3D shapes are represented by triangular meshes or graphs which comprise vertices, edges, and faces (mostly triangles). This section briefly reviews various approaches for describing and classifying 3D shapes. The numerous feature-based approaches [25], [26], [27], [28], [29] include spectral methods which employ the eigenvalues and eigenvectors associated with a graph operator [30] such as the Laplace–Beltrami operator [31]. The heat kernel signature (HKS) [32] and the wave kernel signatures (WKS) [33] are among the best known spectral 3D shape descriptors. They provide an intrinsic (no external reference frame), isometrically invariant description which is robust against perturbations. The HKS characterises heat diffusion: it is governed by the heat equation, while WKS describes probability propagation, as determined by the Schrödinger equation. Other HKS variants have been proposed such as scale-invariant HKS (SI-HKS) [34] and volumetric HKS [35] [36]. An optimal heat kernel was proposed in [37], aimed at minimising the ratio between the diffusion distances associated with corresponding and non-corresponding points respectively. A generalisation of HKS and WKS was proposed by [38]. An optimal heat kernel was obtained by minimizing a distance function which simultaneously minimised and maximised the distances between the features associated with corresponding and non-corresponding points between two objects. This method was further improved by [39] by employing a dense, deep neural network for optimisation. Nonlinear mapping was used in order to address non-isometric deformations. In [40], spectral descriptors, namely HKS and WKS, were matched with the help of a Siamese neural network. The network applied a data fusion process to both signatures while maximising and minimising their interclass and intra-class distances, respectively. In [41], descriptors were learned, with Bayesian techniques, from the probability density function associated with the normalised eigenvectors of the Laplace–Beltrami operator while ensuring dimensionality reduction. Finally, convolutional neural networks were generalised to Riemannian manifolds. Indeed, the absence of translation invariance inherent to Riemannian geometry was addressed by either employing the spectral theorem and graph Fourier transform, or by defining a local tangent space in which the convolution could be performed with the standard approach [36], [42], [43], [44], [45]. Graphs provide a powerful and flexible modelling tool for representing shape properties, as well as for capturing the interrelations between substructures. The specific representation is determined by the motivating application. Indeed, vertices may be associated to points, to geometrical regions, or to any substructure resulting from a segmentation or decomposition process while the edges describe topological relationships between the vertices such as proximity and adjacency. Various graph comparison techniques have been proposed, either focusing on the specific nature of the graphs, or addressing specific application-related issues [46]. For example, 3D shapes may be converted to skeletons via a thinning procedure [47]. It is also possible to use a mapping function directly on a manifold, in order to generate a Reeb graph [48]. A shape may also be divided into substructures using segmentation techniques; interrelations between elements resulting from segmentation or the decomposition can then be represented by a graph [49]. Other graph based methods, such as [50], [51], [52], [53], are based on an approximation of the Graph Edit Distance, through new approaches for decomposing a 3D shape mesh into a set of substructures called triangles-stars. The resulting triangles-stars are then subsequently matched using an assignment algorithm such as the Hungarian algorithm in [50], [51], [53] and the stable marriage algorithm in [52]. Graph embedding techniques unify and combine complementary properties associated with statistical and structural methods [54], [55]. Graph embedding techniques map graphs onto a vector space, thus representing graphs with a set of vectors. Graph embedding techniques can be divided into two different classes [56]. The first class maps the graph vertices (or substructures) onto a set of points in a vector space, associating a vector representation to each vertex, where similar vertices (or substructures) are mapped to nearby points in the vector space [57]. The second class maps whole graphs onto points in a vector space, where similar graphs correspond, in the vector space, to neighbouring points. Among the most salient works are: [58] for isometric embedding; [59], [60] for spectral embedding; [61], [62], [50], [51] for prototype-based embedding. Several strategies for selecting the graph prototypes have been proposed, such as [63], [51]. Indeed, the authors of [51] propose a new graph embedding-based approach for addressing the problem of classifying deformable 3D protein shapes. Proteins functions are strongly constrained by their shape [64] making shape classification an important task. For instance, the importance of shape complementarity for macromolecular docking is well known [65] due to the importance of macromolecular docking in drug and antibody design. The proposed approach defines a metric space using graph embedding techniques, where each object is represented by a set of selected prototypes. The authors propose new approaches for prototype selection based on a multi-criteria algorithm, and feature reduction. Classification is performed with a naïve Bayes classifier. For this, we will require discrete differential calculus and the de Rham operators.

2.2 Discrete Differential Calculus

In order to be applicable to meshes and tessellations, the de Rham operator [12], [13], which is a generalisation of the Laplacian, must be discretised: the reason being that three-dimensional shapes are represented by a finite number of vertices, edges and triangles and, as a result, discrete operators must follow the same representation. Most common approaches for discretisation rely either on finite difference or finite element methods [13]. In this section, we introduce a discrete formulation of the de Rham operator, based on discrete differential calculus [12]. Our objective is to define the de Rham operator from only the discrete vertices, edges and triangles forming a macromolecular surface. Our method differs from other approaches, such as finite different and finite element methods, by the fact that no interpolation is required: the operators are defined solely by the discrete simplices (vertices, edges, triangles, etc.) without any interpolation. The main advantage of this approach is that topological properties are preserved, which it is not the case for methods based on interpolation [12]. Indeed, with discrete differential calculus, it is possible to formulate the de Rham operator in such a way that the topological properties (for instance, the number of holes) and related theorems are enforced at all times. Among these properties are Stokes's theorem, Gauß's theorem, as well as topological invariants such as the Euler characteristic:

$$\chi = \sum_{i=1}^{p} (-1)^i b_i = 2 - 2g, \tag{1}$$

where b_p, the Betti number, corresponds to the number of connected p-simplices [12]. For instance, b_1 is the number of connected vertices, b_2 is the number of connected triangles, and so forth. In order to introduce discrete differential calculus, some notions of differential geometry are required. For an exhaustive review, the reader is referred to [12]. In differential geometry, the Laplacian or the de Rham operator is given by

$$\Delta f = (dd^* + d^*d)f, \tag{2}$$

where d is the exterior derivative, d^* is the codifferential, and f is a k-form (a totally antisymmetric function). Formally, the exterior derivative and the k-form are defined as

$$df = \frac{\partial f_{\mu_1 \ldots \mu_k}}{\partial x^\mu} dx^\mu \wedge dx^{\mu_1} \wedge \ldots dx^{\mu_k}, \tag{3}$$

where $f_{\mu_1 \ldots \mu_k}$is a k-form (a 0-form being a scalar function) and $\wedge$ is the exterior product. The indices associated with the k-form are always antisymmetric while the exterior product corresponds to an antisymmetrisation of the indices associated with the differential elements:

$$\begin{gathered} f_{\mu_1 \ldots \mu_k} \equiv f_{[\mu_1 \ldots \mu_k]} \\ dx^\mu \wedge dx^{\mu_1} \wedge \ldots dx^{\mu_k} \overset{\wedge}{=} dx^{[\mu} dx^{\mu_1} \ldots dx^{\mu_k]}, \end{gathered} \tag{4}$$

where [] is the antisymmetrisation operator e.g. $[x^{\mu_1}, x^{\mu_2}] = \frac{1}{2}(x^{\mu_1} x^{\mu_2} - x^{\mu_2} x^{\mu_1})$.

The codifferential, for its part, is defined as

$$d^* \equiv \delta = -1^{kn+n+1} * \ d \ , \tag{5}$$

where $*$ is the dual operator, given by

$$*\omega_{\mu_1 \ldots \mu_{n-k}} = \frac{1}{k!} \ \omega^{\nu_1 \ldots \nu_k} \sqrt{|\det \mathbf{G}|} \ \varepsilon_{\nu_1 \ldots \nu_k \mu_1 \ldots \mu_{n-k}}, \tag{6}$$

where $\varepsilon_{\nu_1 \ldots \nu_k \mu_1 \ldots \mu_{n-k}}$ is the Levi-Civita symbol, which is totally antisymmetric with $\varepsilon_{1,2,\ldots n} \overset{\wedge}{=} 1$, and $\mathbf{G} = [g_{\mu\nu}]$ is the Riemannian metric tensor [12]. In a nutshell, the dual operator transforms a differential form into a tensor and vice versa. In the case of a 0-form (scalar function), the de Rham operator reduces to

$$\Delta = d^* d, \tag{7}$$

The correspondence in between differential calculus and discrete differential calculus lies in the incidence matrix [12]. The latter is defined as

$$\left(\mathbf{N}_p^T\right)_{ij} = \begin{cases} 0 & \Leftrightarrow \quad \sigma_j^{p-1} \notin \partial \sigma_i^p \\ 1 & \Leftrightarrow \quad \mathcal{O}\left(\sigma_j^{p-1}\right) = \mathcal{O}(\sigma_i^p) \\ -1 & \Leftrightarrow \quad \mathcal{O}\left(\sigma_j^{p-1}\right) = -\mathcal{O}(\sigma_i^p) \end{cases}, \tag{8}$$

where σ_i^p is a p-simplex with label and orientation $\mathcal{O}$. The incidence matrix implicitly implies that the underlying meshes, tessellation or graphs are directed, implying directed graphs whose simplices have well-defined directions. This is to be put in contrast with most current approaches, based on the Laplacian, in which the graphs are undirected. The directions may be determined by the underlying geometry, labelling, or physical properties. For instance, the triangles' orientations are determined by the directions of the normal vectors: a positive orientation corresponds to an outwards-pointing normal vector. Digraphs thus allow a richer, more informative description of the surface, making it possible to distinguish between cavities and protuberances, features that play critical roles in protein interaction [66]. Such concavity or convexity can be ascertained directly from the triangles' normal vectors.

Discrete differential operators may be expressed in terms of the incidence matrix. Indeed, it may be demonstrated [12] that the transpose of the incidence matrix corresponds to the discrete differential operator:

$$d_p = \mathbf{N}_p, \tag{9}$$

Consequently, the discrete codifferential [12] is given by

$$d_p^* \equiv \delta \Rightarrow \left(\mathbf{N}_p\right)^* = *\mathbf{N}_p^T * = \mathbf{G}_{p-1} \mathbf{N}_p^T \mathbf{G}_p^{-1}, \tag{10}$$

where $\mathbf{G}_p$ is the metric associated with the p-simplex. In this paper, the metric associated with the edges is defined as

$$[\mathbf{G}_1]_{ij} = \exp\left(-\frac{(x_i - x_j)^T (x_i - x_j)}{\alpha}\right), \tag{11}$$

where x_i corresponds to the Cartesian coordinates of a vertex labelled with index i and where α is a scaling parameter representative of typical edge length. On the other hand, the metric associated with the vertices is defined as

$$[\mathbf{G}_0]_{ii} = \frac{1}{8} \sum_{\triangle \in \mathcal{N}(i)} \mathcal{A}(\triangle), \quad [\mathbf{G}_0]_{ij} = 0 \quad \forall_{i \neq j}, \tag{12}$$

where $\mathcal{A}(\triangle)$ is the area of a given triangle while $N(i)$ corresponds to the set of triangles belonging to the 1-ring neighbourhood of vertex i. From the previous results, it follows that the discrete de Rham operator is given by

$$\Delta \equiv dd^* + d^*d \Rightarrow \mathbf{L}_p = \mathbf{N}_p \mathbf{N}_p^* + \mathbf{N}_{p+1}^* \mathbf{N}_{p+1}. \tag{13}$$

For the vertices (0-simplices),

$$\mathbf{L}_0 = \mathbf{N}_0\mathbf{N}_0^* + \mathbf{N}_1^*\mathbf{N}_1 = \mathbf{G}_0\mathbf{N}_0^T\mathbf{G}_1^{-1}\mathbf{N}_0. \tag{14}$$

while, for the edges (1-simplices), the de Rham operator reduces to

$$\begin{aligned}\mathbf{L}_1 &= \mathbf{N}_1\mathbf{N}_1^* + \mathbf{N}_2^*\mathbf{N}_2\\ &= \mathbf{N}_1\mathbf{G}_0\mathbf{N}_1^T\mathbf{G}_1^{-1} + \mathbf{G}_1\mathbf{N}_2^T\mathbf{G}_2^{-1}\mathbf{N}_2.\end{aligned} \tag{15}$$

The latter is also known as the vectorial Laplacian. In the next section, the (continuous) fractional Fokker–Planck equation and the fractional de Rham operator are introduced by means of the Lévy random walk, which will subsequently allow us to establish relationships between more distant regions on the macromolecular surface These relationships are important as they strongly constrain protein functionality and shape.

3 Lévy Random Walk and Fractional Fokker–Planck Equation

The next steps of our analysis require the concepts of 'random walk' and 'random process'. According to the Kolmogorov–Chapman formula [6], if the transition probability between two adjacent (connected by an edge) vertices i and j is known, the transition probability between two remote vertices may be obtained from a path integral over the former:

$$\begin{aligned}&P(x,t|x_0,t_0)\\ &\quad= \prod_{j=0}^{N-1}\int P\big(x_{j+1},t_{j+1}\big|x_j,t_j\big)\ dx_{j+1}.\end{aligned} \tag{16}$$

Most methods employ the Gaussian transition probability which exponentially suppresses large transitions, so the random walk explores only a relatively small neighbourhood around each point. In the present work, the neighbourhood is extended by considering heavy-tailed transition probability distributions for which large transitions are more likely. We have chosen the Lévy or stable distribution [6] as it has proven itself effective in modelling the large transitions required for describing spatial relations between distant regions on the molecular surface [67]. Indeed, the suppression of large steps by the Gaussian distribution is exponential while that by the Lévy distribution is polynomial. Large steps are required to relate distant or disconnected regions; the Lévy distribution being a generalisation of the Gaussian distribution, it encompasses the latter [6]. This distribution does not have a closed (analytic) form, but its characteristic function — its Fourier transform — does. Employing the Lévy distribution for the transition probability, the Kolmogorov–Chapman formula becomes:

$$\begin{aligned}P_L(x,t|x_0,t_0) &= \frac{1}{2\pi}\lim_{\substack{N\to\infty\\ \triangle t\to 0}}\int_{(x_0,t_0)}^{(x,t)}\prod_{j=0}^{N-1}dx_j\\ &\int_{-\infty}^{\infty}dk\,\exp\big(ik\big(x_{j+1}-x_j\big)\big)\ \exp(-\triangle t\ \gamma^\alpha|k|^\alpha).\end{aligned} \tag{17}$$

The integral in Eq. (17) corresponds to the characteristic function. As in Eq. (16), the path integral is taken over all possible paths. The parameter $\alpha\in[0,2]$corresponds to the exponent of the heavy-tailed power law: the smaller the exponent, the more likely large transitions are while, for a value of two, the distribution reduces to the Gaussian or normal distribution.

The stable distribution is a solution of Eq. (18), the inhomogeneous fractional Fokker–Planck equation [6]:

$$\left[\frac{\partial}{\partial t} - R_x^\alpha\right]P_L(x,t|x_0,t_0) = \delta(x-x_0)\ \delta(t-t_0), \tag{18}$$

where $\delta(t-t_0)$is the Dirac delta function, $P_L(x,t|x_0,t_0)$is the transition probability, and R_x^α is the fractional Riesz derivative [6] which is defined as

$$\begin{aligned}R_x^\alpha\circ P_L(x,t|x_0,t_0)&\overset{\triangle}{=}\frac{b}{\pi}\Gamma(\alpha+1)\sin\left(\frac{\pi\alpha}{2}\right)\\ &\times\int_{-\infty}^{\infty}\frac{P_L(x',t|x_0,t_0)}{|x'-x|^{1+\alpha}}\,dx'.\end{aligned} \tag{19}$$

The fractional derivative may be defined in a more intuitive way via the Fourier transform. Indeed, it is well-known from Fourier analysis [68] that the fractional gradient operator may be obtained by raising the Fourier frequencies by the fractional exponent and then taking the inverse Fourier transform:

$$\nabla^\alpha\cdot = -\frac{1}{2\pi}\int d^nk\ \exp(ikx)\ \|k\|^\alpha\cdot \tag{20}$$

As a result, the (continuous) fractional Laplacian becomes:

$$\Delta^{\alpha/2}\cdot = -\frac{1}{2\pi}\int d^nk\ \exp(ikx)\ \|k\|^\alpha\cdot \tag{21}$$

These operators have, to some extent, a relationship with fractal geometry. Indeed, it may be demonstrated [6] that the fractal dimension associated with the fractal de Rham operator is given by

$$d_F = \alpha:\quad 1\ <\ \alpha\le 2. \tag{22}$$

Fractal operators are particularly well suited to characterise multiscale self-similarity [69], [70] which is commonplace in highly complex shapes [69], [70]. Outside this range, however, the operator is not fractal. Some macromolecular surfaces present highly complex substructures that may be fractal but, in general, the shape is not [71].

In the next section, the concept of fractional derivate is extended to time in order to obtain non-Markovian processes, i.e. random walks with memory, as opposed to the memoryless, non-fractional time derivatives. Our working hypothesis is that a random walk with memory should allow for a more thorough and informative exploration of a shape, by exploiting the important information remembered during exploration, hence accounting for the substructure interrelations that are of paramount importance for protein description, but which are readily lost during memoryless random walks.

4 Fractional Time Derivative and Non-Markovian Process: The Bifractional Fokker–Planck Equation

Non-fractional time derivatives are memoryless; therefore, the evolution process associated with a trajectory is purely

Markovian [6]. In the present work, it is assumed that a random walk is more informative if the information associated with trajectories is not discarded. Such non-Markovian behaviour may be obtained by introducing a fractional time derivative into the inhomogeneous fractional Fokker–Planck equation to derive the bifractional Fokker–Planck equation:

$$\begin{aligned} &D_t^{\beta}(u(x,t)) - u(x,t_0)\frac{t^{-\beta}}{\Gamma(1-\beta)} - \Delta^{\alpha/2}\, u(x,t) \\ &= \delta(x-x_0)\ \delta(t-t_0), \end{aligned} \tag{23}$$

where D_t^{β} is the fractional time derivative. The latter is defined from the Riemann–Liouville fractional integral [6]:

$$D_t^{\beta}(f(t)) \triangleq \frac{1}{\Gamma(-\beta)} \int_{-\infty}^{t} \frac{f(\tau)}{(t-\tau)^{1+\beta}}\ d\tau. \tag{24}$$

The second term in the left member of Eq. (23) is required for consistency. The temporal fractional derivative is characterised by its fractional exponent β which is also known as the Hurst exponent:

$$H = \beta - 1/2. \tag{25}$$

Without a temporal fractional derivative, the temporal evolution is a Markovian process and, therefore, memoryless. On the other hand, employing a fractional temporal derivative makes the evolution process non-Markovian, with a memory. As a result, the temporal variance is not infinite and may be estimated by

$$\begin{aligned} &\left\langle (u(t)-u(0))^2 \right\rangle \\ &\simeq \frac{4\kappa\ t^{2\beta-1}}{\beta\ \Gamma(\beta)^2}\ {}_2F_1(1;1-\beta,1+\beta:1), \end{aligned} \tag{26}$$

where ${}_2F_1$ is the hypergeometric function [72], a function of only the fractional exponent β.

From the variance, one may distinguish two regimes: the persistent regime ($\beta < 1/2$) and the anti-persistent regime($\beta > 1/2$). Under the persistent regime, the evolution remains strongly correlated over time while, for the anti-persistent, it does not. When $\beta = 1/2$, the variance is infinite and the process becomes Markovian i.e. memoryless. The temporal variance is associated with the temporal fractional derivative. As stated in Eq. (26), it depends solely on the time and the fractional exponent β. The variance becomes infinite when the fractional exponent is equal to 0.5 irrespectively if the surface is tessellated or not. The various regimes associated with the fractional operators are outlined in Table 1.

TABLE 1
Various Regimes Associated With the Fractional Operators

Regime	α(de Rham)	β(temporal)
Lévy random walk	$0 < \alpha \leq 2$	
Fractal random walk	$1 < \alpha \leq 2$	
Superdiffusion	$\alpha < 2$	
Diffusion (Gaussian)	$\alpha = 2$	
Subdiffusion	$\alpha > 2$	
Persistent process (non-Markovian)		$\beta < 1/2$
Markovian process (memoryless)		$\beta = 1/2$
Anti-persistent process (non-Markovian)		$\beta > 1/2$
Sombrero wavelets (admissibility conditions)	$1/2 < \alpha \leq 2$	$1/2 \leq \beta \leq 1$

In the next section, the bifractional heat kernel and the bifractional Fokker–Planck signature (descriptor) are introduced to describe the shapes of deformable proteins.

5 Discrete Bifractional Fokker-Planck Equation and Kernel

The solution of the homogeneous heat equation is called the heat kernel [68]. Accordingly, the solution of the homogeneous bifractional Fokker–Plank equation is called the bifractional heat kernel because it requires two fractional parameters, the temporal fractional exponent and the Lévy fractional exponent, for its definition. The principal interest of the heat kernel lies in the fact that it contains all of the necessary information about heat propagation [68] to define the shape description. This may be explained by the fact that any solution of the inhomogeneous heat equation may be expressed in terms of the heat kernel [68]. The heat kernel is, by definition, the solution of the homogeneous Fokker–Planck equation:

$$\begin{aligned} &\frac{\partial K_\Delta(t)}{\partial t} = \Delta K_\Delta(t) \Rightarrow \\ &K_\Delta(t) = \exp(-\Delta t). \end{aligned} \tag{27}$$

As a result of both Eq. (27) and the spectral theorem [68], the heat kernel may be expressed in terms of the eigendecomposition of the de Rham operator:

$$K_\Delta(t) = \sum_{k=0}^{\infty} \phi_{\Delta;k}(x)\ \phi^{-1}_{\Delta;k}(x_0)\ \exp\bigl(-t\ \lambda_{\Delta;k}\bigr), \tag{28}$$

where $\{\lambda_{\Delta;k}\}_{k=0}^{\infty}$ and $\{\phi_{\Delta;k}(x)\}_{k=0}^{\infty}$are the eigenvalues and eigenvectors associated with the non-fractional de Rham operator:

$$\Delta\phi_{\Delta;\,k}(x) = \lambda_{\Delta;k}\phi_{\Delta;k}(x). \tag{29}$$

Similarly, the bifractional heat kernel is the solution of the homogeneous bifractional Fokker-Planck equation:

$$\begin{aligned} &D_t^{\beta} K_{\Delta;\alpha,\beta}(t) - \frac{t^{-\beta} K_{\Delta;\alpha,\beta}(t)}{\Gamma(1-\beta)} = \Delta^{\alpha/2}\ K_{\Delta;\alpha,\beta}(t) \Rightarrow \\ &K_{\Delta;\alpha,\beta}(t) = E_\beta\Bigl(-\ t^{\beta}\ \Delta^{\alpha/2}\Bigr), \end{aligned} \tag{30}$$

where

$$E_\beta(x) \triangleq \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(\beta k + 1)}, \tag{31}$$

is the Mittag–Leffler function (a generalisation of the exponential function). The bifractional heat kernel reduces to

$$K_{\Delta;\alpha,\beta}(t) = \sum_{k=1}^{N} \phi_{\Delta;k}\,\phi^{-1}_{\Delta;k}\ E_\beta\Bigl(-\ t^{\beta}\ \lambda^{\alpha/2}_{\Delta;k}\Bigr), \tag{32}$$

once the spectral decomposition is applied to the de Rham operator.

The discrete, inhomogeneous, fractional Fokker–Planck equation is obtained by replacing the continuous de Rham operator by its discrete counterpart:

$$D_t^{\beta}\mathbf{u} - \mathbf{u}_0 \frac{t^{-\beta}}{\Gamma(1-\beta)} - \left(\mathbf{L}_p\right)^{\alpha/2} \mathbf{u} = \mathbf{I}\,\delta(t-t_0), \tag{33}$$

where **I**is the identity matrix. As for its continuous de Rham operator, the discrete fractional de Rham operator has a spectral decomposition:

$$\begin{aligned} \left(\mathbf{L}_p\right)^{\alpha/2} &= \Phi_{\mathbf{L}_p}\ \Lambda_{\mathbf{L}_p}^{\alpha/2}\ \Phi_{\mathbf{L}_p}^{-1} \\ \Phi_{\mathbf{L}_p}\left[\phi_{\mathbf{L}_p;1}\middle|\cdots\middle|\phi_{\mathbf{L}_p;n}\right], \quad \Lambda_{\mathbf{L}_p}^{\alpha/2} &= \operatorname{diag}\left(\lambda_{\mathbf{L}_p;1}^{\alpha/2}, \ldots, \lambda_{\mathbf{L}_p;n}^{\alpha/2}\right), \end{aligned} \tag{34}$$

where $\Phi_{\mathbf{L}_p}$ is the matrix of eigenvectors (in which the eigenvectors each constitute a column) while $\Lambda_{\mathbf{L}_p}$is the corresponding eigenvalues matrix, in which the eigenvalues are along the diagonal. As a result, the bifractional heat kernel becomes:

$$\mathbf{K}_{\Delta;\alpha,\beta}(t) = \Phi_{\mathbf{L}_p}\ E_{\beta}\left(-t^{\beta}\Lambda_{\mathbf{L}_p}^{\alpha/2}\right)\ \Phi_{\mathbf{L}_p}^{-1}, \tag{35}$$

where the Mittag–Leffler function is applied to the diagonal elements only. In the next section, the bifractional Dirac–Kähler equation and the bifractional Planck–Dirac signature are introduced for deformable macromolecular shape description.

6 Bifractional Dirac–Kähler Equation and Bifractional Planck–Dirac Signature

We propose employing, in addition to the Fokker–Planck equation, the Dirac–Kähler equation for shape analysis. The notoriously complex and intricate shapes of proteins can be better targeted by combining multiple and complementary strategies. The Dirac–Kähler operator [73] is defined as

$$\mathfrak{D} \equiv d - \delta. \tag{36}$$

In essence, this operator is the square root of the de Rham operator:

$$\mathfrak{D}^2 = (d-\delta)^2 = d\delta + \delta d = \Delta. \tag{37}$$

The Dirac–Kähler equation [10] is defined as

$$\frac{\partial\Psi}{\partial t} = (d - \delta + 1)\Psi = (\mathfrak{D} + 1)\ \Psi. \tag{38}$$

This equation is completely distinct from the Fokker–Planck equation. Indeed, its solution is expressed in terms of quaternions [9], [10], [74], unlike the real solution of the Fokker–Planck equation. Quaternions are a generalisation of complex numbers, characterised by the following properties:

$$q = a + bi + cj + dk \Leftrightarrow \begin{cases} i^2 = j^2 = k^2 & \\ ij = k & ji = -k \\ jk = i & kj = -i \\ ki = j & ik = -j \end{cases}, \\ a, b, c, d \in \mathbb{R}; \quad i, j, k \in \mathbb{Q}. \tag{39}$$

In contrast to real numbers, quaternions do not commute under any arithmetic operation, and may interfere and interact. Therefore, instead of characterising a shape by analysing the non-interacting heat flow, the Dirac–Kähler equation propagates a field of non-commuting and interacting quaternions [10] . Quaternions are computationally expensive, but that cost may be circumvented by expressing them in terms of forms [9], [10]. For instance, the Dirac–Kähler field on a shape may be expressed as a sum of differential forms (totally antisymmetric functions) over the simplices:

$$\Psi = \psi(x) + \psi_{\mu}(x)dx^{\mu} + \frac{1}{2!}\psi_{\mu\nu}(x)dx^{\mu} \wedge dx^{\nu}. \tag{40}$$

As for the discrete Fokker–Planck equation, the discrete Dirac–Kähler equation [75], [76] may be obtained by replacing the exterior derivative and the codifferential by their discrete counterparts as defined by Eqs. (9) and (10):

$$\begin{aligned} \frac{\partial\Psi_p}{\partial t} &= \left(\mathfrak{D}_p + \mathbf{I}_p\right)\ \Phi_p \Rightarrow \\ \frac{\partial\Psi_p}{\partial t} &= \begin{bmatrix} 1 & -(\mathbf{N}_1)^* & & & & \\ \mathbf{N}_1 & 1 & -(\mathbf{N}_2)^* & & & \\ & \mathbf{N}_2 & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & -\left(\mathbf{N}_p\right)^* \\ & & & & \mathbf{N}_p & 1 \end{bmatrix} \Psi_p \end{aligned} \tag{41}$$

where

$$\Psi_p = \left[\psi_{\sigma_1}, \ldots, \psi_{\sigma_p}\right]^T. \tag{42}$$

For macromolecular surfaces, the Dirac–Kähler equation reduces to

$$\begin{aligned} & p = 2 \Rightarrow \\ & \frac{\partial\Psi_2}{\partial t} = (\mathfrak{D}_2 + \mathbf{I}_2)\ \Psi_2 \Rightarrow \\ & \frac{\partial\Psi_2}{\partial t} = \begin{bmatrix} 1 & -\mathbf{G}_0\mathbf{N}_1^T\mathbf{G}_1^{-1} & 0 \\ \mathbf{N}_1 & 1 & -\mathbf{G}_1\mathbf{N}_2^T\mathbf{G}_2^{-1} \\ 0 & \mathbf{N}_2 & 1 \end{bmatrix} \Psi_2, \end{aligned} \tag{43}$$

where the Dirac–Kähler field consists of the forms associated with the vertices, the edges and the faces:

$$\Psi_2 = [\psi_V, \psi_E, \psi_F]^T, \quad \Psi_2\ \in \mathbb{R}^{(|V|+|E|+|F|)} \tag{44}$$

As a result, the Dirac–Kähler operator is isomorphic to

$$\begin{aligned} \mathfrak{D}_2 &\cong \begin{bmatrix} 0 & V \times E & 0 \\ E \times V & 0 & E \times F \\ 0 & F \times E & 0 \end{bmatrix}, \\ \mathfrak{D}_2\ &\in \mathbb{R}^{(|V|+|E|+|F|)\times(|V|+|E|+|F|)} \end{aligned} \tag{45}$$

The fractional Dirac–Kähler equation is obtained in the same way as the Fokker–Planck equation:

$$D_t^{\beta}\Psi_p = \left(\left(\mathfrak{D}_p\right)^{\alpha} + \mathbf{I}_p\right)\Psi_p. \tag{46}$$

As for its Fokker–Planck counterpart, the fractional Dirac–Kähler operator is obtained by taking the fractional power of its spectral decomposition:

$$\left(\mathfrak{D}_p\right)^{\alpha} = \mathbf{Q}_{\mathfrak{D}} \Lambda_{\mathfrak{D}}^{\alpha} \mathbf{Q}_{\mathfrak{D}}^{-1}. \tag{47}$$

It follows immediately that the discrete bifractional Dirac–Kähler kernel has the form:

$$\mathbf{K}_{\mathfrak{D}_p;\alpha,\beta}(t) = \Phi_{\mathfrak{D}_p}\ E_{\beta}\Big(-t^{\beta}\Lambda_{\mathfrak{D}_p}^{\alpha/2}\Big)\ \Phi_{\mathfrak{D}_p}^{-1}. \tag{48}$$

We propose a new isometrically invariant shape descriptor or signature, called the bifractional Planck–Dirac signature (BPDS), which generalises the heat kernel signature [3]. It is defined as

$$\begin{aligned} \mathbf{K}_{\mathbf{L},\mathfrak{D}} \triangleq & \bigcup_{p=p_{\min}}^{p_{\max}} \bigcup_{\alpha>1/2}^{2} \bigcup_{\beta=1/2}^{1} \bigcup_{t=t_{\min}}^{t_{\max}} \mathfrak{H}\big(\mathrm{diag}\big(\mathbf{K}_{\mathbf{L}_p;\alpha,\beta}(t)\big)\big)\Bigg] \\ & \bigcup \left[\bigcup_{p'=p'_{\min}}^{p'_{\max}} \bigcup_{\alpha>1/2}^{2} \bigcup_{\beta=1/2}^{1} \bigcup_{t=t_{\min}}^{t_{\max}} \mathfrak{H}\Big(\mathrm{diag}\Big(\mathbf{K}_{\mathfrak{D}_{p'};\alpha,\beta}(t)\Big)\Big)\right] \end{aligned} \tag{49}$$

where *diag* indicates the diagonal element of the kernels, while $\mathfrak{H}$ is their probability distribution function (PDF). The BPDS is essentially a concatenation of the PDFs of the diagonal element of the bifractional heat kernels and the Dirac–Kähler kernel for a discrete set of times and fractional exponents. It is important to note that the descriptors are entirely based on geometric information, meaning that no physicochemical properties are employed whatsoever. We plan to explore the benefits of concurrently employing geometric information and physicochemical properties in future research. As reported in [68], the information present in the heat kernel is entirely contained within its diagonal. As a result, the computational complexity is reduced without information loss.

Wavelets have been successfully applied in multiresolution invariant description of deformable shapes [77], [78], [79], [80], [81]. In the next section, bifractional wavelets are introduced in order to generalise Eq. (49) for multiresolution analysis. Indeed, multiresolution analysis is often necessary to describe highly complex shapes [82] such as protein surfaces.

7 Bifractional Planck–Dirac Wavelets

Wavelets have compact supports, meaning that they are localised in both the spatial and the spectral domain; therefore, they are especially suitable for multiresolution shape analysis [77], [78], [79], [80], [81]. The continuous sombrero (Mexican hat) wavelet is defined as the temporal derivative of the kernel (either heat or Dirac–Kähler) [68]. As a result, the bifractional Fokker–Planck wavelets and the bifractional Dirac–Kähler wavelets are given by

$$\begin{aligned} \xi_{\mathbf{O}_p;t}(x,y) &\equiv -\frac{\partial}{\partial t} K_{\mathbf{O}_p}(t) \\ &= \sum_{k=0}^{n-1} \hat{\xi}_{\mathbf{O}_p;t}(k)\ \phi_{\mathbf{O}_p;k}(x)\phi_{\mathbf{O}_p;k}(y) \end{aligned} \tag{50}$$

where $\xi_{\mathbf{O}_p;t}(x,y)$ is the wavelet in the spatial domain, while $\hat{\xi}_{\mathbf{O}_p;t}(k)$ is its counterpart in the Fourier or spectral domain [68]. The operator $\mathbf{O}_p$stands either for the fractional de Rham operator, or the fractional Dirac–Kähler operator. It follows from Eq. (50) that

$$\begin{aligned} \hat{\xi}_{\mathbf{O}_p;t}(k) &= -\frac{\partial E_{\beta}\Big(-\ t^{\beta}\ \lambda_{\mathbf{O}_p;k}^{\alpha}\Big)}{\partial t} \Rightarrow \\ \hat{\xi}_{\mathbf{O}_p;t}(k) &= t^{-1+\beta}\lambda_{\mathbf{O}_p;k}^{\alpha}\ E_{\beta,\beta}\Big(-t^{\beta}\lambda_{\mathbf{O}_p;k}^{\alpha}\Big), \end{aligned} \tag{51}$$

where $E_{\alpha,\beta}(x)$ is the generalised Mittag–Leffler function [6]:

$$E_{\alpha,\beta}(x) = \sum_{k=0}^{\infty} \frac{\overset{\wedge}{x^k}}{\Gamma(\beta k+\alpha)}, \tag{52}$$

Consequently, our approach may be extended to wavelets through a simple substitution:

$$\begin{aligned} & E_{\beta}\Big(-\ t^{\beta}\ \lambda_{\mathbf{O}_p;k}^{\alpha/2}\Big) \mapsto \\ & t^{-1+\beta}\lambda_{\mathbf{O}_p;k}^{\alpha}\ E_{\beta,\beta}\Big(-t^{\beta}\lambda_{\mathbf{O}_p;k}^{\alpha}\Big). \end{aligned} \tag{53}$$

When this substitution is performed in Eq. (49), one obtains the Planck–Dirac wavelet signatures (BPDWS) which entail a bifractional multiresolution shape description. This signature, described by both Eqs. (49) and (53), is employed in our framework for macromolecular shape description.

Wavelets must satisfy an admissibility condition [68]. Indeed, their Fourier spectrum must be bounded and consequently localised [68]:

$$\begin{aligned} & \sum_{k=0}^{\infty} \frac{\left|\hat{\xi}_{\mathbf{O}_p;t}(k)\right|^2}{k} < \infty \Rightarrow \\ & \qquad \sum_{k=0}^{\infty} k^{2\alpha-1}\ t^{2\beta-2}\ E_{\beta,\beta}^2\big(-k^{\alpha}t^{\beta}\big)\ <\ \infty. \end{aligned} \tag{54}$$

The admissibility condition constrains, in turn, the values for which the temporal and spatial fractional exponents are valid. These values have been estimated numerically from Eq. (54) and are reported in the last row of Table 1.

The kernels are evaluated for a discrete subset of times as well as for a discrete subset of temporal and spatial (Lévy) fractional exponents. As for standard heat kernel-based approaches [68], the time scales control the extent to which the neighbourhood of a given vertex is explored: that is, heat is propagated. The Lévy fractional exponents α determine the extent to which large transitions are allowed during exploration. In the present paper, α is restricted to the interval $[1/2\ ,1]$ in which the admissibility condition for the wavelet is satisfied. In turn, the temporal fractional exponents β determine whether the random walk associated with the bifractional heat kernel is memoryless (Markovian) or not (non-Markovian). Once more, β is restricted to the interval $[1/2\ ,1]$ in which the admissibility condition for the wavelet is satisfied. As for p, that determines which operator is employed. In this paper, we consider the de Rham operators associated with the vertices $(p=0)$ and edges $(p=1)$ as well as the Dirac–Kähler operator associated with the triangles$(p=2)$.

In the next section, we describe Fractio: the subsystem in charge of evaluating the bifractional Planck–Dirac wavelets signatures.

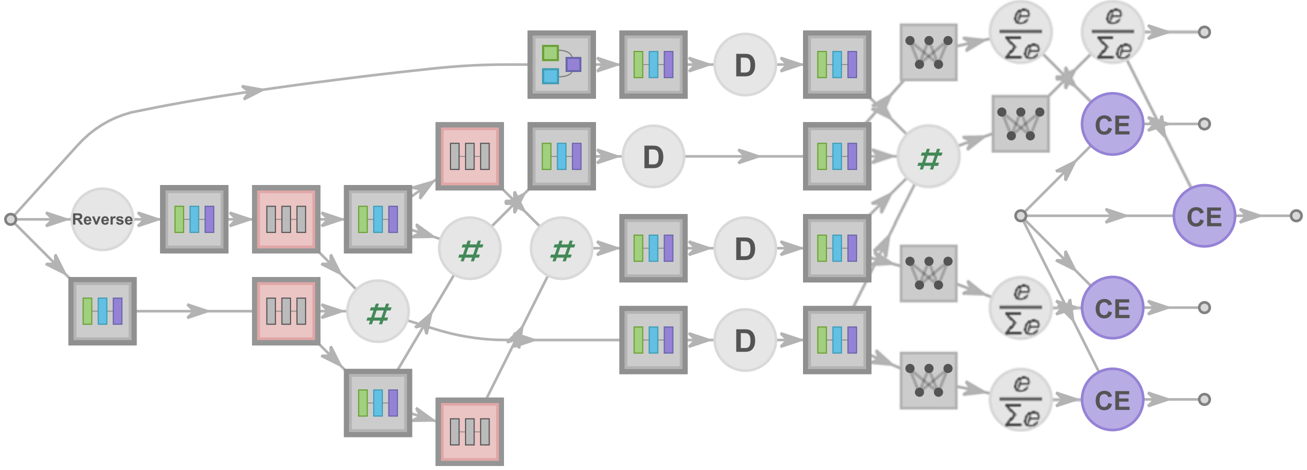

Fig. 2. Outline of the architecture of Gauß.

8 Bifractional Planck–Dirac Wavelets Signature Evaluation

Our system consists of two components: Fractio, which evaluates the Planck–Dirac wavelet signatures, and Gauß, the protein-classification deep neural network; both programmes are written in the Mathematica$^{\mathrm{TM}}$ language. The OFF files corresponding to the macromolecular surfaces are stored in a Bitcoin-like blockchain [83]. Each object is encrypted with an AES256 symmetric key [84]. The blockchain provides many advantages over a conventional file system or database: it is decentralised, secure, and tamper-proof by design, with Byzantine fault tolerance [83] in addition to providing a unique address or transaction ID for each protein, thus eliminating ambiguities and mismatches associated with version control and naming [83]. Fractio processes the proteins in parallel on forty Dual Intel Xeon Gold 6140, 2.3 GHz, 18-core, 140 W Skylake processors. The OFF files are first retrieved from the blockchain and, for each one of them, the vertex and edge incidence matrices are evaluated with Eq. (8) while the corresponding metrics are calculated with Eqs. (12) and (11) respectively. The triangle metric is simply a diagonal matrix in which each element corresponds to the area of a particular triangle. The vertex and edge de Rham operators, and the Dirac–Kähler operators, are evaluated according to Eqs. (14), (15), and (43) respectively. Since the matrices associated with these operators are sparse by construction, sparse arrays are employed for their storage which alleviates both computational complexity and memory usage. Since the number of vertices may vary from one macromolecular surface to the other, and in order to standardise the dimensions of the operators, the de Rham and Dirac–Kähler operators are resampled with the optimal sampling of maximal order and minimal support method (MOMS) [85]. Their spectral decompositions (the eigenvectors and eigenvalues) are then evaluated with the Arnoldi iterative method [86] up to the sixtieth order. Finally, the fractional Planck–Dirac wavelet signatures are evaluated according to Eq. (49) with $\alpha = \{3/5\ ,1,3/2,2\ \}$, $\beta = \{1/2\ ,3/4,1\ \}$ with $p = \{0,1\}$for the de Rham operators and $p = 2$ for the Dirac–Kähler operator. The deep neural network employed for protein classification is described in the next section.

9 Multiobjective Deep Neural Network

The input for Gauß, our multiobjective deep neural network, is the bifractional Planck–Dirac wavelet signature defined earlier. This descriptor may be assimilated to a sequence or pseudo-time series:

$$\begin{aligned}\mathbf{K}_{\mathrm{L},\mathfrak{D}} \cong \Bigg\{ & \bigcup_{\alpha>1/2}^{2} \bigcup_{\beta=1/2}^{1} \left[\bigcup_{t=t_{\min}}^{t_{\max}} \mathfrak{H}\big(\mathrm{diag}\big(\mathbf{K}_{\mathrm{L}_0;\alpha,\beta}(t)\big)\big) \right]^T \Bigg\| \\ & \bigcup_{\alpha>1/2}^{2} \bigcup_{\beta=1/2}^{1} \left[\bigcup_{t=t_{\min}}^{t_{\max}} \mathfrak{H}\big(\mathrm{diag}\big(\mathbf{K}_{\mathrm{L}_1;\alpha,\beta}(t)\big)\big) \right]^T \Bigg\| \\ & \bigcup_{\alpha>1/2}^{2} \bigcup_{\beta=1/2}^{1} \left[\bigcup_{t=t_{\min}}^{t_{\max}} \mathfrak{H}\big(\mathrm{diag}\big(\mathbf{K}_{\mathfrak{D}_2;\alpha,\beta}(t)\big)\big) \right]^T \Bigg\}, \end{aligned} \tag{55}$$

where $\|$ stands for concatenation. The sequence is vectorial: each element has the form:

$$\left[\mathfrak{H}\big(\mathrm{diag}\big(\mathbf{K}_{\mathrm{O}_p;\alpha,\beta}(t_{\min})\big)\big)\big\| \dots \big\|\mathfrak{H}\big(\mathrm{diag}\big(\mathbf{K}_{\mathrm{O}_p;\alpha,\beta}(t_{\max})\big)\big)\right]^T \tag{56}$$

The smallest time increment [68] supported by the graph is

$$\min_{\substack{\eta \\ E}} \|E\| = 2\sqrt{2t_{\min}} \tag{57}$$

where $\min_\eta$is a robust estimate of the minimum, obtained by discarding the $\eta - 1$ smallest values, while represents an edge. The time intervals for which the BPDWS are evaluated correspond to harmonics of this time increment:

$$t = \bigcup_{\ell=0}^{n-1} 2^{\ell} t_{\min}, \tag{58}$$

where $n = 6$ in our application.

Our deep neural network architecture is inspired by pyramid Siamese networks [87]. The architecture of Gauß is outlined in Fig. 2 and fully described in Fig. 3. As for any Siamese network [15], our network consists of two branches. The input, for both branches, is the bifractional Planck–Dirac wavelet signature associated with the protein under consideration, while the output is the corresponding SCOPe class. Each branch of Gauß consists of four blocks. In accordance with Siamese network practices, the second block (as

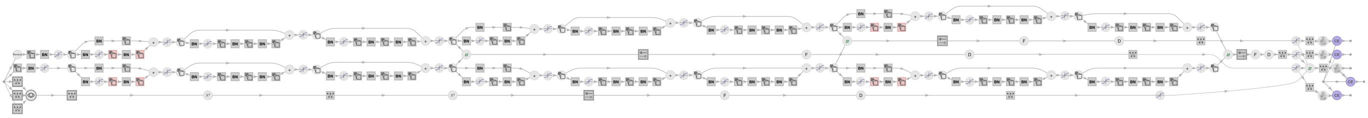

Fig. 3. Detailed architecture of Gauß.

well as the fourth block) of the upper and lower branch share the same parameters.

The periodical coupling in between the two branches reduces the number of parameters, and consequently, the risk of overfitting. Additionally, these couplings act as constraints which aim at facilitating the learning process by applying additional restriction on the loss function [88].

For their part, the first and the third block of the upper and lower branch do not share parameters, ensuring that each branch may learn from different perspectives. The lower branch has, just after the input layer, a sequence-reversal layer to eliminate learning biases associated with the sequence ordering.

The corresponding blocks of the upper and lower branch are linked by connections which are described below. The various blocks ensure a multiresolution analysis of the inputs while the connections associate an independent classifier to each resolution level. As a result, our network is multiobjective and the training must be performed with stochastic Pareto optimisation.

Each coupled block contains a convolutional sub-block, and two threading sub-blocks. Each convolutional sub-block comprises a batch normalization layer, a activation function, a convolutional layer of size five with ten filters, a batch normalization layer, and a convolutional layer of size five with ten filters. These values were settled upon though extensive experimentation. The first convolutional networks of each branch share common parameters; the second convolutional networks of the upper branch also share common weights with the second convolutional network of the lower branch. Essentially, the role of the convolutional layer is to perform a multiresolution analysis as the convolution may be assimilated to a band-pass filter. As opposed to a dense layer, convolutional layers strongly reduce the number of parameters that must be learned, thus reducing the risk of overfitting. The batch normalisation layers stabilise and accelerate the network by learning the input data mean and variance which makes it possible to re-centre and re-scale the data.

Each threading block consists of two links. The first link consists of a batch normalisation layer, activation function, a convolutional layer of size five with ten filters, a batch normalisation layer, a convolutional layer of size five with ten filters, and a batch normalisation layer followed by a convolutional layer of size ten with ten filters. The second link consists of an additive threading layer which acts on both the output from the first link and also on the input of the threading block. This is followed by the activation function, and a maximum pooling layer of size three which is defined as

$$\max\big[\mathfrak{H}_k\big(\mathrm{diag}\big(\mathbf{K}_{\mathbf{L};\alpha,\beta,2}(t)\big)\big), \mathfrak{H}_k\big(\mathrm{diag}\big(\mathbf{K}_{\mathbf{L};\alpha,\beta,2}(t)\big)\big),\\ \mathfrak{H}_k\big(\mathrm{diag}\big(\mathbf{K}_{\mathfrak{D};\alpha,\beta,2}(t)\big)\big)\big]. \tag{59}$$

where $\mathfrak{H}_k$ is the kth element of the discrete probability distribution function. As stated earlier, the first and the third blocks do not share parameters. The first block consists of a convolutional network of size five with ten filters, a batch normalization layer, a activation function, and a maximum pooling layer of size three for summarisation as defined by Eq. (59). As for the third block, it has the same structure as the second and fourth blocks except for the fact that it does not share parameters between the upper and lower branches.

The second block of the upper branch is connected to the second block of the lower branch; the third and fourth block follow the same rule. The connection consists of a multiplicative $\tanh(o_1 \times o_2)$ threading layer, a bidirectional long short-term memory (LSTM) recurrent network [89] in order to introduce a long and short term memory mechanism, a dropout layer (probability=0.1) for regularisation, and a dense layer (number of neurons = number of classes), followed by a activation function.

The fourth and last connection consists of an attention network, followed by a transpose layer, a dense layer, another transpose layer, a bidirectional long short-term memory recurrent layer, a dropout layer for regularisation, and a dense layer. The outputs from the four connections are combined by a subnetwork consisting of an additive $\tanh(o_1 + o_2 + o_3 + o_4)$ threading layer and a dense layer (number of neurons = number of classes) with a activation function. The attention layer allows the network to learn which elements of the input are the most relevant when determining the class, and which weights should be attributed to them [90]. Therefore, the fourth connection serves as a multiresolution classifier with attention. Each classifier has a confidence threshold: if the class occurrence probability is greater than the threshold, the outcome is considered conclusive, or inconclusive otherwise.

Finally, each connection is completed by a dense layer, a softmax layer, and a cross-entropy loss layer. As a result, each connection acts as a classifier, so that a classifier is associated with each resolution level. The first three connections correspond to three different resolution levels, while the fourth connection encompasses all resolution levels. Each classifier has its own objective function, so multiobjective (Pareto) stochastic optimisation must be performed

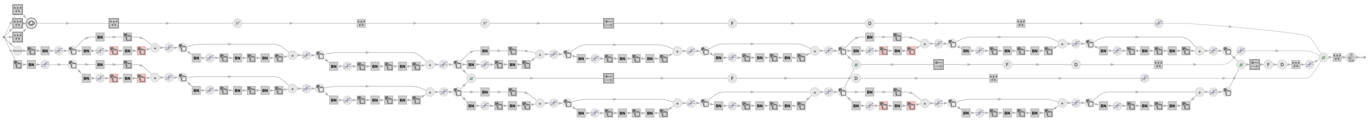

Fig. 4. Trained Gauß network.

TABLE 2
Parameters for the Adam Algorithm

ρ_1	ρ_2	ε	δ
0.9	0.999	0.000001	0

while learning [91]. This ensures that each level of detail is properly learned. A small quadratic regularisation term is added to the multiobjective cost function for regularisation.

Once the network is trained, only the layers between the input and the final output are retained, i.e. only the multiresolution classifier remains, as illustrated in Fig. 4. The network is trained with stochastic gradient descent with the adaptive moment estimation (Adam) algorithm [92]: this is an extension of RMSProp which is summarised in Algorithm 1. This algorithm, which is outlined in Eq. (60), evaluates and accumulates, for each iteration, the expectations of both the gradient and its second moment. The accumulated expectations are rescaled as time flows. Finally, a correction is applied to the learnable parameters. In this work, the size of the mini batches has been fixed to 64. All sizes were set through inspection.

$$
\begin{gathered}
\nabla \leftarrow \frac{1}{m} \nabla_\theta \sum\nolimits_i L\Big(f\Big(\mathbf{x}^{(i)};\theta\Big), \mathbf{y}^{(i)}\Big), \quad \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \sim \mathcal{D} \\
t \leftarrow t+1 \\
\mathbf{s} \leftarrow, \rho_1 \mathbf{s} + (1-\rho_1)\nabla \\
\mathbf{r} \leftarrow \rho_2 \mathbf{r} + (1-\rho_2)\nabla \odot \nabla \\
\hat{\mathbf{s}} \leftarrow \frac{\mathbf{s}}{1-\rho_1^t} \\
\hat{\mathbf{r}} \leftarrow \frac{\mathbf{r}}{1-\rho_2^t} \\
\Delta\theta = -\varepsilon \hat{\mathbf{s}} \varnothing \Big(\sqrt{\hat{\mathbf{r}}} + \delta\Big) \\
\theta \leftarrow \theta + \Delta\theta. \qquad (60)
\end{gathered}
$$

Algorithm 1: Summarisation of the Adaptive moment estimation (Adam) algorithm: s is the accumulated gradient, r is the accumulated moment, $\rho_1$and $\rho_2$are the exponential decay rates, $\emptyset$ is the element wise division while and $\delta$are stability parameters. The values employed for these parameters are reported in Table 2.

The calculations were performed by a Nvidia Tesla V100 (32 GB) GPU. The weights and the bias were initially estimated with Xavier's method [93], in which the parameters are chosen such as to preserve variance when propagated through subsequent layers. The whole approach may be summarised as follows:

1) For each protein, reduce the mesh size to 1000 polygons
2) For each protein, evaluate the BPDWS signature with Eq. (49) and (53), that is with Fractio
3) Generate the 10-folds datasets
4) For each fold
   a) Train, with Adam, the multiobjective Gauß network with the training BPDWS signatures evaluated with Fractio
   b) Prune the trained network from all its outputs except the last one
   c) Evaluate the precision, recall, accuracy, Van Rijsbergen's effectiveness measure and precision-recall curve with the corresponding training set
5) Evaluate the average of the metrics for all ten folds

Experimental results are presented in the next section.

10 Experimental Results

The SHREC'19 Protein Shape Retrieval Contest [17] dataset was employed in our experiments. It consists of 5,298 proteins extracted from 211 entries in the Protein Data Bank [18], each of which has been classified according to the Structural Classification of Proteins—extended (SCOPe) [19] based on their conformation (the shape of their macromolecular surface).The Structural Classification of Proteins – Extended (SCOPe) classifies protein domains according to their structural and evolutionary properties [94]. The classification is performed according to class and fold for the structural properties, and according to superfamily, family, protein and species for their evolutionary counterparts. The SHREC'19 data was obtained by selecting proteins from the SCOPe database according to the following criterions: 1) macromolecular surfaces (3-D structure) were resolved by nuclear magnetic resonance spectroscopy, 2) the conformers (deformable shapes) must have the same number of atoms, 3) each entry must have at least four ortholog proteins, and 4) the resulting dataset was randomly subsampled in order to obtain 5298 domains. The macromolecular surfaces (solvent excluded surface) were calculated with EDTSurf [95]. The dataset consists of 17 classes. Classification is made challenging by the fact that the macromolecular surfaces associated with proteins are so geometrically complex, as illustrated in Fig. 1. Furthermore, they are subject to deformations as a result of the interaction with both their external environment and their multiple degrees of freedom, all of which contributes to altering their conformation. The dataset consists of 17 classes, the largest of which has 1160 proteins; there is an average of 311 proteins per class [17]. Therefore, the dataset is highly unbalanced. Our Fractio-Gauß algorithm was compared against thirteen algorithms and variants which are described, in detail, in the SHREC'19 Protein Shape Retrieval Contest [17]. These include three variants of the convolutional neural network–linear dynamical system framework (ConvLDSNet), the 3D Zernike moments framework (3DZM), three variants of the 3D Zernike descriptors framework (3DZD), three variants of the histogram of area projection transform framework (HAPT), the vector of locally aggregated descriptors framework (VLAD), the globally aligned spatial distribution framework (GASD), and the hybrid GASD–VLAD framework, all of which are described in [17].

Essentially, ConvLDSNet combines convolutional neural networks with linear dynamical systems: the numbering referring to the training dataset; 3DZM employs 3D Zernike moments: a prior alignment of protein pairs with fast Fourier transform is required; 3DZD relies on 3D Zernike descriptors: the numbering referring to the classification metric; HAPT determines the probability for a point to be the centre of a spherical symmetry: the numbering referring to the parametrisation of the algorithm; Ft-PSSC is based on

TABLE 3
E-Measure, Mean Average Precision, and First and Second Tier for the SHREC'19 Dataset

Framework	$\mathcal{E}$	$\bar{\mathcal{M}}$	$\mathcal{T}_1$	$\mathcal{T}_2$
Fractio-Gauß	**0.090**	**0.928**	**0.947**	**0.945**
3DZD 1	0.580	0.638	0.579	0.729
3DZD 2	0.510	0.712	0.658	0.789
3DZD 3	0.520	0.720	0.665	0.802
3DZDM	0.650	0.649	0.583	0.706
ConvLDSNet 1	0.730	0.329	0.303	0.458
ConvLDSNet 2	0.740	0.324	0.296	0.457
ConvLDSNet 3	0.730	0.328	0.301	0.458
Ft-PSSC + GASD	0.720	0.417	0.372	0.506
Ft-PSSC + GASD + VLAD	0.740	0.315	0.315	0.481
Ft-PSSC + VLAD	0.750	0.206	0.226	0.380
HAPT 1	0.540	0.659	0.616	0.734
HAPT 2	0.530	0.666	0.624	0.738
HAPT 3	0.550	0.658	0.613	0.732

a fast point feature histogram (FPFH) combined with either the globally aligned spatial distribution (GASD), the vector of locally aggregated descriptors (VLAD), or both.

Our framework was evaluated using 10-fold cross-validation: the original dataset was randomly partitioned into ten equal-sized subsamples. Of the ten subsamples, a single subsample was retained as the validation data for testing the model, and the remaining nine subsamples were employed as training data. The cross-validation process was repeated ten times, with each of the ten subsamples used exactly once as the validation data. The ten results were averaged to produce a single estimate. Evaluating the Planck–Dirac wavelet signatures was completed in 71 minutes while the total training and testing time for all ten folds was 28 minutes.

The performance was evaluated and compared with five metrics: Van Rijsbergen's effectiveness measure (also known as E-measure: $\mathcal{E}$), the mean average precision $\bar{\mathcal{M}}$, and the first ($\mathcal{T}_1$) and second ($\mathcal{T}_2$) tier [96]. The E-measure is defined as

$$\mathcal{E} = 1 - \frac{2}{\frac{1}{P} + \frac{1}{R}}, \tag{61}$$

where P is the precision and R is the recall, which are defined as

$$\begin{cases} P = \frac{TP}{TP+FP} \\ R = \frac{TP}{TP+FN} \end{cases}, \tag{62}$$

where TP refers to true positive classification count, FP refers to false positive classification count and FN refers to false negative classification count. Meanwhile, $\bar{\mathcal{M}}$ is the area under the precision-recall curve and characterises the classifier performance on average. In addition, it is also important to characterise the behaviour of the neural network for a short list of the best results. This is essential, for instance, if the classifier acts as a search engine in which case only the first best results (short list) are relevant. The characterisation of the performance for the first best results may be achieved with $\mathcal{T}_1$ and $\mathcal{T}_2$; the former considers the average precision for the first $|\mathcal{C}| - 1$ results while the latter extends the average to the first $2(|\mathcal{C}| - 1)$ matches, respectively, with $|\mathcal{C}|$ being the number of instances per class.

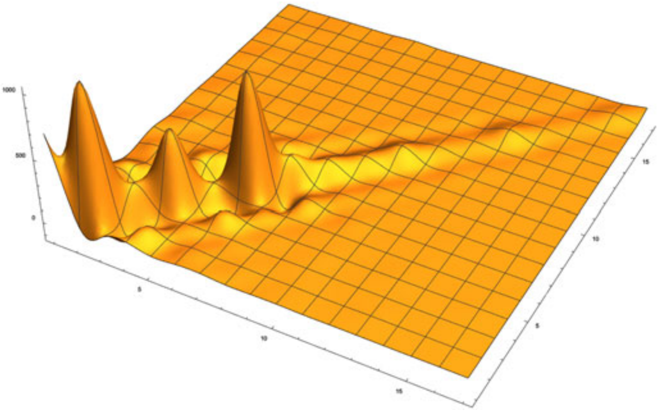

Fig. 5. Confusion matrix for the proposed approach for the SHREC'19 dataset.

Our results are reported in Table 3 and in Fig. 7 respectively. Table 3 provides the Van Rijsbergen effectiveness measure, as defined by Eq. (61), for all the frameworks: a value of zero corresponds to perfect classification, while a value of one corresponds to a classification indistinguishable from random. From the table, it may be concluded that our method outperforms the others. Indeed, Van Rijsbergen's effectiveness measure is 0.09 in our case while the next smallest is 0.51 for 3DZD 2. The same may be said about $\bar{\mathcal{M}}$, and $\mathcal{T}_1$ and $\mathcal{T}_2$. For $\bar{\mathcal{M}}$, a value of 0.928 was obtained with the proposed approach while the best $\bar{\mathcal{M}}$ for SHREC'19 was 0.720 (3DZD 3). Values of 0.947 and 0.945 were obtained with the proposed approach for $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, while the best corresponding SHREC'19 values were 0.665 and 0.802 respectively; both with 3DZD3. The confusion matrix [97], which appears in Fig. 5, further demonstrates the efficiency of the proposed method. The results are illustrated, in Fig. 6, by showing two proteins belonging to the same class. Fig. 7 shows the precision–recall curves both for SHREC'19 frameworks as well as for our Fractio–Gauß framework. Our approach has a higher precision than any SHREC'19 method irrespective of the recall rate. This demonstrates again the performance of our framework.

11 Application to Deformable Partial Shapes

Although our main objective was protein classification, it is important to note that our approach is applicable to multiple domains. For this reason, we illustrate how to apply our

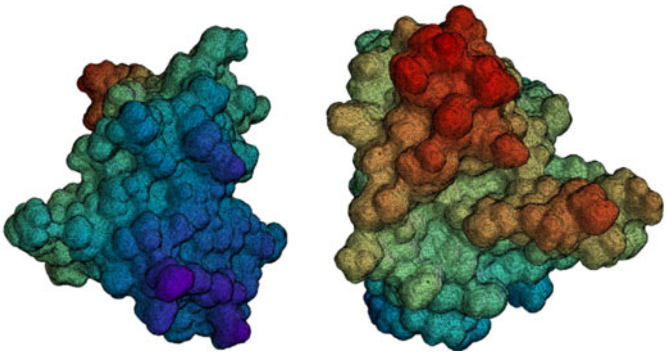

Fig. 6. Illustration of two proteins belonging to the same class (133_C13 and 324_C13).

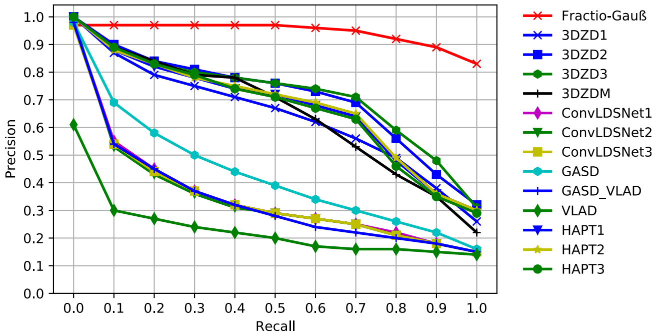

Fig. 7. Precision–recall curve for Fractio–Gauß as well as for the ones proposed in SHREC'19.

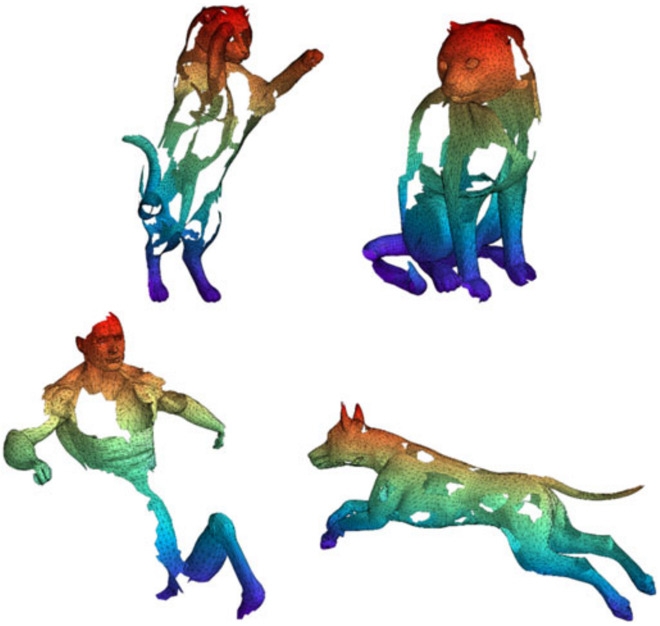

Fig. 8. A sample of deformable partial shapes from the SHREC'16 dataset: the cat in two postures (deformable shape), and a pose of the centaur and the dog.

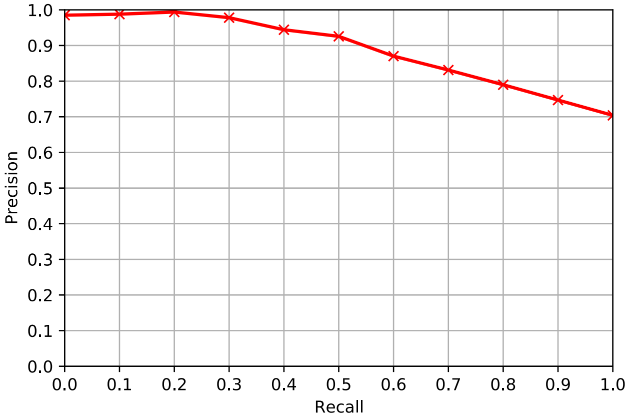

Fig. 9. Precision-recall curve for Fractio–Gauß for the SHREC'16 dataset.

TABLE 4
E-Measure, Mean Average Precision, and First and Second Tier of the Proposed Approach for the SHREC'16 Dataset

$\mathcal{E}$	0.163
$\bar{\mathcal{M}}$	0.890
$\mathcal{T}_1$	0.985
$\mathcal{T}_2$	0.989

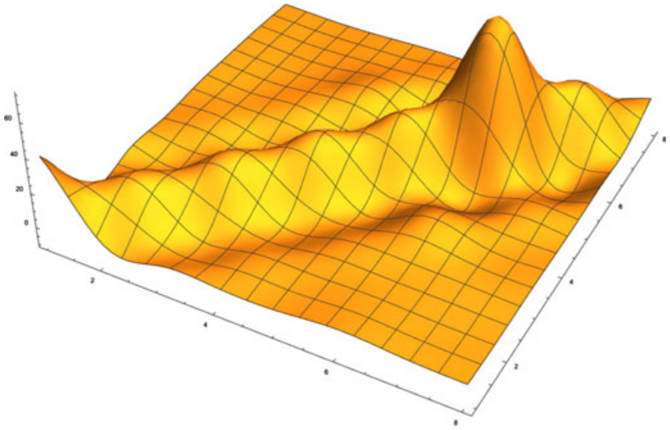

Fig. 10. Confusion matrix for the proposed approach for the SHREC'16 dataset.

approach to classification of deformable partial shapes. In order to make the problem challenging, the reader should notice that no prior knowledge on the full shape is assumed, meaning that training and testing is entirely performed with partial shapes. The dataset employed is the SHREC'16 partial matching of deformable shapes [98] which was initially created for partial matching. This dataset consists of 599 deformable partial shapes divided into 8 classes: dog and wolf (the two being relatively similar), cat, David and Michael (again, the two being very similar), horse, centaur (a hybrid creature combining Michael and a horse), and Victoria. In order to generate partial shapes, full shapes of each class are either cut with a plane at six different orientations or eroded from a seed hole. In addition, for each class, irregular holes are created keeping either 40%, 70% and 90% of the original area. This is illustrated in Fig. 8.

In regard to the proteins, the same descriptors and neural network were employed with the same hyperparameters. The only distinction was the patience (period after which training is interrupted if the loss function does not change significantly), which has been increased by a twofold factor in order to reflect the difficulty involved in learning deformable partial shapes. The same training procedure and metrics employed for the proteins were employed for the evaluation: precision-recall curve, E-measure, MAP, and first and second tier. The precision-recall curve is reported in Fig. 9 and the remaining metrics appear in Table 4. The confusion matrix, which appears in Fig. 10, further demonstrates the efficiency of the proposed method.

These results, which are comparable in terms of performances to those obtained for proteins, demonstrate the generality of the proposed method. The fact that the differential operators (de Rham, Kähler–Dirac) preserve topological properties, such as holes (see Section 2.2), make them more suitable for the learning of deformable partial shapes. Meanwhile, the multiobjective cost function allows for more efficient learning of the various resolution levels (latent representations) associated with the pyramid neural network because each level has its own cost function.

12 Conclusion and Future Work

We have introduced a new framework for deformable macromolecular surface classification based on the bifractional Fokker–Planck and Dirac–Kähler equations, as well as a new multiobjective pyramid neural network. Macromolecular surfaces were analysed in terms of a non-Markovian Lévy random walk which makes it possible to establish geometrical relationships between distant simplices while remembering previous analyses. A multi-perspective analysis was performed with the Fokker–Planck equation which propagates a non-interacting scalar, and with the Dirac–Kähler equations which propagate a non-commuting, interfering quaternion field. A new deep multiobjective pyramid neural network was employed for multiresolution shape analysis. Multiobjective learning was performed with Pareto stochastic optimisation, ensuring that each resolution level was properly learned. Multiobjective learning is utilised since the learning process could potentially be biased toward a particular level if a unique objective function is employed [99]. In addition, we found that multiobjective optimisation typically reduces the number of epochs required to train the network by half. Our method was applied successfully to the SHREC'19 dataset, and is applicable to any kind of shape.

In addition to the Fokker–Planck and the Dirac–Kähler equations, one may alternatively consider the Schrödinger equation. The shape descriptor associated with Schrödinger's equation is the celebrated wave kernel signature (WKS) [8]. The Schrödinger equation, like the Fokker-Planck equation, has a fractional counterpart [100], [101] which is given by

$$i\hbar\frac{\partial\psi}{\partial t} = -\eta_\alpha(\hbar\nabla)^\alpha\psi \tag{63}$$

where ∇^α is the fractional gradient operator, which is defined by Eq. (20), and η_α is a constant. The non-fractional Schrödinger equation corresponds to $\alpha = 2$. Both the fractional and the non-fractional Schrödinger equations satisfy the continuity equation [100], [101]:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot j_\alpha = I_\alpha \tag{64}$$

where ρ is a probability density function:

$$\rho = \psi^*\psi \tag{65}$$

and j_α is the probability current:

$$j_\alpha = -\,i\eta_\alpha\hbar^{\alpha-1} \left[\psi^*(-\Delta)^{\alpha/2\,-1}\nabla\psi - \psi(-\Delta)^{\alpha/2\,-1}\nabla\psi^*\right] \tag{66}$$

For the non-fractional Schrödinger equation, the right member of Eq. (66) is zero meaning that the probability is conserved locally [100], [101]. This is a sine qua non condition for the wave kernel signature to be defined [8]. Yet, this member is different from zero for the fractional Schrödinger equation:

$$I_\alpha = -i\eta_\alpha\hbar^{\alpha-1} \left[\nabla\psi^*(-\Delta)^{\alpha/2\,-1}\nabla\psi - \nabla\psi(-\Delta)^{\alpha/2\,-1}\nabla\psi^*\right] \tag{67}$$

This means that the probability is not conserved locally as occurs in the non-fractional case. Nevertheless, it is conserved globally [100], [101]. Therefore, the approach developed for the WKS is not directly applicable to the fractional Schrödinger equation. Nonetheless, the form of the continuity equation would suggest that the fractional Schrödinger equation would be a suitable candidate for a description which could benefit from the intricate interplay between local and global features. The application of the fractional Schrödinger equation to invariant deformable shape description shall therefore be the foundation of our future work.

References

[1] J. M. Berg, J. L. Tymoczko, L. Stryer, and G. J. Gregory, *Biochemistry*. New York, NY, USA: WH Freeman, 2019.

[2] G. Shamai and R. Kimmel, "Geodesic distance descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3624–3632.

[3] E. Paquet and H. L. Viktor, "Isometrically invariant description of deformable objects based on the fractional heat equation," in *Proc. 15th Int. Conf. Comput. Anal. Images Patterns*, 2013, pp. 135–143.

[4] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1704–1711.

[5] R. Litman and A. M. Bronstein, "Learning spectral descriptors for deformable shape correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 171–180, Jan. 2014.

[6] B. West, M. Bologna, and P. Grigolini, *Physics of Fractal Operators*. Berlin, Germany: Springer, 2012.

[7] M. Sun, S. Y. Bao, and S. Savarese, "Object detection using geometrical context feedback," *Int. J. Comput. Vis.*, vol. 100, no. 2, pp. 154–169, 2012.

[8] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2011, pp. 1626–1633.

[9] K. Crane, *Conformal Geometry Processing*. Pasadena, CA, USA: California Inst. Technol., 2013.

[10] S. Kruglov, "Dirac-Kähler equation," *Int. J. Theor. Phys.*, vol. 41, no. 4, pp. 653–687, 2002.

[11] J. Vince, *Quaternions for Computer Graphics*. Berlin, Germany: Springer, 2011.

[12] L. J. Grady and J. R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Berlin, Germany: Springer, 2010.

[13] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo, "Discrete Laplace–Beltrami operators for shape analysis and segmentation," *Comput. Graph.*, vol. 33, no. 3, pp. 381–390, 2009.

[14] X. Chen, X. Chen, Y. Zhang, X. Fu, and Z. J. Zha, "Laplacian pyramid neural network for dense continuous-value regression for complex scenes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 5034–5046, Nov. 2021.

[15] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2720–2729.

[16] M. Gobbi, F. Levi, and G. Mastinu, "Multi-objective stochastic optimisation of the suspension system of road vehicles," *J. Sound Vib.*, vol. 298, no. 4–5, pp. 1055–1072, 2006.

[17] F. Langenfeld *et al.*, "Protein shape retrieval contest," in *Proc. 12th Eurographics Workshop Object Retrieval*, 2019, pp. 25–31.

[18] S. K. Burley *et al.*, "RCSB Protein Data Bank: Biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D464–D474, 2018.

[19] J.-M. Chandonia, N. K. Fox, and S. E. Brenner, "SCOPe: Classification of large macromolecular structures in the structural classification of proteins–extended database," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D475–D481, 2018.

[20] W. Fan, "Graph pattern matching revised for social network analysis," in *Proc. 15th Int. Conf. Database Theory*, 2012, pp. 8–21.

[21] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. Int. Symp. Circuits Syst.*, 2010, pp. 253–256.

[22] D. Conte, P. Foggia, C. Sansone, and M. Vento, "How and why pattern recognition and computer vision applications use graphs," in *Applied Graph Theory in Computer Vision and Pattern Recognition*, Berlin, Germany: Physica-Verlag, 2007, pp. 85–135.

[23] R. C. Wilson, E. R. Hancock, E. Pekalska, and R. P. W. Duin, "Spherical and hyperbolic embeddings of data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2255–2269, Nov. 2014.

[24] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on Riemannian manifolds with Gaussian RBF kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2464–2477, Dec. 2015.

[25] E. Paquet, M. Rioux, A. Murching, T. Naveen, and A. Tabatabai, "Description of shape information for 2-D and 3-D objects," *Signal Process. Image Commun.*, vol. 16, no. 1, pp. 103–122, 2000.

[26] M. Yu, I. Atmosukarto, W. K. Leow, Z. Huang, and R. Xu, "3D model retrieval with morphing-based geometric and topological feature maps," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. 656–661.

[27] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A reflective symmetry descriptor for 3D models," *Algorithmica*, vol. 38, no. 1, pp. 201–225, 2004.

[28] M. Körtgen, G.-J. Park, M. Novotni, and R. Klein, "3D shape matching with 3D shape contexts," in *Proc. 7th Central. Eur. Seminar Comput. Graph.*, 2003, pp. 5–17.

[29] J. Xie, G. Dai, F. Zhu, E. K. Wong, and Y. Fang, "DeepShape: Deep-learned shape descriptor for 3D shape retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1335–1345, Jul. 2017.

[30] H. Zhang, O. van Kaick, and R. Dyer, "Spectral mesh processing," *Comput. Graph. Forum*, vol. 29, no. 6, pp. 1865–1894, 2010.

[31] B. Levy, "Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry," in *Proc. IEEE Int. Conf. Shape Model. Appl.*, 2006, pp. 13–21.

[32] J. Sun, M. Ovsjanikov, and L. J. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[33] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2011, pp. 1626–1633.

[34] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1704–1711.

[35] D. Raviv, M. M. Bronstein, A. M. Bronstein, and R. Kimmel, "Volumetric heat kernel signatures," in *Proc. ACM Workshop D Object Retrieval*, 2010, pp. 39–44.

[36] R. Rostami, F. S. Bashiri, B. Rostami, and Z. Yu, "A survey on data-driven 3D shape descriptors," *Comput. Graph. Forum*, vol. 38, no. 1, pp. 356–393, 2019.

[37] Y. Aflalo, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Deformable shape retrieval by learning diffusion kernels," in *Scale Space Variational Methods Computer Vision*. Ein-Gedi, Israel: Springer, 2011, pp. 689–700.

[38] R. Litman and A. M. Bronstein, "Learning spectral descriptors for deformable shape correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 171–180, Jan. 2014.

[39] J. Xie, M. Wang, and Y. Fang, "Learned binary spectral shape descriptor for 3D shape correspondence," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3309–3317.

[40] X. Shi, J. Zhao, L. Zhang, and X. Ye, "Non-rigid 3D object retrieval with a learned shape descriptor," in *Proc. 9th Int. Conf. Image Graph.*, 2017, pp. 24–37.

[41] Z. Wang and H. Lin, "3D shape retrieval based on Laplace operator and joint Bayesian model," *Vis. Inf.*, vol. 4, no. 3, pp. 69–76, 2020.

[42] R. Socher, B. Huval, B. P. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3D object classification," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 665–673.

[43] B. Shi, S. Bai, Z. Zhou, and X. Bai, "DeepPano: Deep panoramic representation for 3-D shape recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2339–2343, Dec. 2015.

[44] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5648–5656.

[45] C. Wang, M. Cheng, F. Sohel, M. Bennamoun, and J. Li, "NormalNet: A voxel-based CNN for 3D object classification and retrieval," *Neurocomputing*, vol. 323, pp. 139–147, 2019.

[46] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," in *Proc. Shape Model. Appl.*, 2004, pp. 145–156.

[47] J. Czajkowska, C. Feinen, M. Grzegorzek, M. Raspe, and R. Wickenhöfer, "Skeleton graph matching vs. maximum weight cliques aorta registration techniques," *Computerized Med. Imag. Graph.*, vol. 46, pp. 142–152, 2015.

[48] V. Barra and S. Biasotti, "3D shape retrieval using kernels on extended Reeb graphs," *Pattern Recognit.*, vol. 46, no. 11, pp. 2985–2999, 2013.

[49] Y. Kleiman, O. van Kaick, O. Sorkine-Hornung, and D. Cohen-Or, "Shed: Shape edit distance for fine-grained shape similarity," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–11, 2015.

[50] K. Madi, E. Paquet, and H. Kheddouci, "New graph distance for deformable 3D objects recognition based on triangle-stars decomposition," *Pattern Recognit.*, vol. 90, pp. 297–307, 2019.

[51] K. Madi and E. Paquet, "New graph embedding approach for 3D protein shape classification," in *Proc. Joint 9th Int. Conf. Inf. Electron. Vis., 4th Int. Conf. Imag. Vis. Pattern Recognit.*, 2020, pp. 1–7.

[52] K. Madi, E. Paquet, and H. Kheddouci, "New graph distance based on stable marriage formulation for deformable 3D objects recognition," in *Proc. 16th IEEE/ACS Int. Conf. Comput. Syst. Appl.*, 2019, pp. 1–8.

[53] K. Madi, E. Paquet, H. Seba, and H. Kheddouci, "Graph edit distance based on triangle-stars decomposition for deformable 3D objects recognition," in *Proc. Int. Conf. on 3D Vis.*, 2015, pp. 55–63.

[54] H. Bunke and K. Riesen, "Recent advances in graph-based pattern recognition with applications in document analysis," *Pattern Recognit.*, vol. 44, no. 5, pp. 1057–1067, 2011.

[55] B. Horst and R. Kaspar, "Towards the unification of structural and statistical pattern recognition," *Pattern Recognit. Lett.*, vol. 33, no. 7, pp. 811–825, 2012.

[56] P. Foggia, G. Percannella, and M. Vento, "Graph matching and learning in pattern recognition in the last 10 years," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 28, no. 1, 2014, Art. no. 1450001.

[57] F. Escolano, B. Bonev, and M. A. Lozano, "Information-geometric graph indexing from bags of partial node coverages," in *Proc. Int. Workshop Graph-Based Representations Pattern Recognit.*, 2011, pp. 52–61.

[58] S. Jouili and S. Tabbone, "Graph embedding using constant shift embedding," in *Proc. Int. Conf. Pattern Recognit. Recognizing Patterns Signals Speech Images Videos*, 2010, pp. 83–92.

[59] X. Bai, E. R. Hancock, and R. C. Wilson, "Graph characteristics from the heat kernel trace," *Pattern Recognit.*, vol. 42, no. 11, pp. 2589–2606, 2009.

[60] X. Bai, Y. Song, and P. M. Hall, "Learning invariant structure for object identification by using graph methods," *Comput. Vis. Image Underst.*, vol. 115, no. 7, pp. 1023–1031, 2011.

[61] W. Lee and R. P. W. Duin, "A labelled graph based multiple classifier system," in *Proc. 8th Int Workshop Mult. Classifier Syst.*, 2009, pp. 201–210.

[62] W. Lee, R. P. W. Duin, and H. Bunke, "Selecting structural base classifiers for graph-based multiple classifier systems," in *Proc. 9th Int. Workshop Mult. Classifier Syst.*, 2010, pp. 155–164.

[63] E. Z. Borzeshi, M. Piccardi, K. Riesen, and H. Bunke, "Discriminative prototype selection methods for graph embedding," *Pattern Recognit.*, vol. 46, no. 6, pp. 1648–1657, 2013.

[64] B. Alberts *et al.*, *Molecular Biology of the Cell*. New York, NY, USA: W.W. Norton & Company, 2014.

[65] Y. Yan and S.-Y. Huang, "Pushing the accuracy limit of shape complementarity for protein-protein docking," *BMC Bioinf.*, vol. 20, no. 25, Dec. 2019, Art. no. 696.

[66] Z. Wu, Q. Liao, and B. Liu, "A comprehensive review and evaluation of computational methods for identifying protein complexes from protein–protein interaction networks," *Brief. Bioinf.*, vol. 21, no. 5, pp. 1531–1548, 2020.

[67] E. Paquet and H. L. Viktor, "Toward a better description of deformable macromolecular surfaces: isometrically invariant and allosterically aware shape signature based on the fractional fokker–planck equation," in *Proc. 4th Comput. Vaccinol. Workshop at ISV*, 2014, pp. 1–7.

[68] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[69] S. Butera and M. Di Paola, "A physically based connection between fractional calculus and fractal geometry," *Ann. Phys.*, vol. 350, pp. 146–158, 2014.

[70] M. E. Farmer, *Application of Chaos and Fractals to Computer Vision*. Al Sharjah, UAE: Bentham Science Publishers, 2015.

[71] M. Lewis and D. C. Rees, "Fractal surfaces of proteins," *Sci.*, vol. 230, no. 4730, pp. 1163–1165, 1985.

[72] K. Aomoto, M. Kita, T. Kohno, and K. Iohara, *Theory of Hypergeometric Functions*. Berlin, Germany: Springer, 2011.

[73] K. C. Hsueh-Ti Derek Liu, and Alec Jacobson, "A Dirac operator for extrinsic shape analysis," *Comput. Graph. Forum*, vol. 36, no. 5, pp. 139–149, 2017.

[74] K. Morita, "Quaternions, Lorentz group and the Dirac theory," *Prog. Theor. Phys.*, vol. 117, no. 3, pp. 501–532, 2007.

[75] V. Sushch, "A discrete model of the Dirac-Kähler equation," *Rep. Math. Phys.*, vol. 73, no. 1, pp. 109–125, 2014.

[76] V. Sushch, "A discrete Dirac-Kähler equation using a geometric discretisation scheme," *Adv. Appl. Clifford Algebras*, vol. 28, no. 4, pp. 1–17, 2018.

[77] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.

[78] W. Hwa Kim, M. K. Chung, and V. Singh, "Multi-resolution shape analysis via non-Euclidean wavelets: Applications to mesh segmentation and surface alignment problems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2139–2146.

[79] N. Li, S. Wang, M. Zhong, Z. Su, and H. Qin, "Generalized local-to-global shape feature detection based on graph wavelets," *IEEE Trans. Visual Comput. Graphics*, vol. 22, no. 9, pp. 2094–2106, Sep. 2016.

[80] C. Li and A. B. Hamza, "A multiresolution descriptor for deformable 3D shape retrieval," *Vis. Comput.*, vol. 29, no. 6, pp. 513–524, 2013.

[81] T. Hou and H. Qin, "Admissible diffusion wavelets and their applications in space-frequency processing," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 1, pp. 3–15, Jan. 2013.

[82] Y. Zhao, D. Stoffler, and M. Sanner, "Hierarchical and multi-resolution representation of protein flexibility," *Bioinf.*, vol. 22, no. 22, pp. 2768–2774, 2006.

[83] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?–a systematic review," *Public Lib. Sci. One*, vol. 11, no. 10, pp. 1–27, 2016.

[84] A. Biryukov and D. Khovratovich, "Related-key cryptanalysis of the full AES-192 and AES-256," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Secur.*, 2009, pp. 1–18.

[85] T. Blu, P. Thcvenaz, and M. Unser, "MOMS: Maximal-order interpolation of minimal support," *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1069–1080, Jul. 2001.

[86] K. Dookhitram, R. Boojhawon, and M. Bhuruth, "A new method for accelerating Arnoldi algorithms for large scale eigenproblems," *Math. Comput. Simul.*, vol. 80, no. 2, pp. 387–401, 2009.

[87] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[88] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[89] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[90] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[91] S. RojasGonzalez, H. Jalali, and I. Van Nieuwenhuyse, "A multi-objective stochastic simulation optimization algorithm," *Eur. J. Oper. Res.*, vol. 284, no. 1, pp. 212–226, 2020.

[92] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[93] L. Datta, "A survey on activation functions and their relation with xavier and he normal initialization," 2020, *arXiv: 2004.06632*.

[94] A. Andreeva, E. Kulesha, J. Gough, and A. G. Murzin, "The scop database in 2020: Expanded classification of representative family and superfamily domains of known protein structures," *Nucleic Acids Res.*, vol. 48, no. D1, pp. D376–D382, 2020.

[95] D. Xu and Y. Zhang, "Generating triangulated macromolecular surfaces by euclidean distance transform," *PLoS One*, vol. 4, no. 12, 2009, Art. no. e8140.

[96] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[97] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sens. Environ.*, vol. 62, no. 1, pp. 77–89, 1997.

[98] L. Cosmo, E. Rodolaá, M. M. Bronstein, A. Torsello, D. Cremers, and Y. Sahillioglu, "Partial matching of deformable shapes," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2016, pp. 61–67.

[99] T. Eftimov and P. Korošec, "Deep statistical comparison for multi-objective stochastic optimization algorithms," *Swarm Evol. Computation*, vol. 61, 2021, Art. no. 100837.

[100] Y. Wei *et al.*, "Comment on "Fractional quantum mechanics" and "Fractional Schrödinger equation"," *Phys. Rev. E*, vol. 93, no. 6, 2016, Art. no. 066103.

[101] N. Laskin, "Fractional Schrödinger equation," *Phys. Rev. E*, vol. 66, no. 5, 2002, Art. no. 056108.

Eric Paquet is currently a senior research officer with the National Research Council of Canada and an adjunct professor with the School of Electrical Engineering and Computer Science, University of Ottawa. He is currently pursuing research on geometric deep learning, machine learning, and protein-protein interaction with the National Research Council of Canada. He is the principal investigator of the Artificial Intelligence Protein Design for Drugs and Gene Therapies Project, which is part of the AI for Design Challenge Program (NRC). He is also the principal investigator of the FoodScan Project, combining recent pathogen microbiological developments with the latest advances in data science technology, which is part of the joint Canada-U.K. Enhanced Industrial Productivity Program, this project has been endorsed by the Eureka High-Level Group, the key decision-making body of Eureka. He is an international expert for the ISO/TC 307 Blockchain and Distributed Ledger Technologies Committee and also for the ISO/IEC JTC 1/SC 42 Artificial Intelligence Committee. He is the author of more than 150 publications and holds many patents. His research interests include deep learning, computational geometry, and data streams, with applications in structural proteomics, anthropometric databases, quantitative finance, and cultural heritage.

Herna L. Viktor is currently a full professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Canada. She is the author of more than 150 journal articles, conference papers, and book chapters and serves on the editorial board of three journals. Her research focuses on data-driven discovery, with an emphasis on creating learning algorithms to extract meaning from fast-evolving streams. The end results of her research have been applied across numerous and diverse domains, including the study of pediatric anemia patients in collaboration with the Hospital for Sick Children in Toronto, exploring the evolving media discourse regarding the Alberta oil sands debate, intelligent placement of data analytics applications on cloud services, sentiment analysis for opinion mining in elections, and a study of content-based detection of online influence campaigns. Her work was the recipient of numerous recognitions and awards.

Kamel Madi received the master's degree in computer science (artificial intelligence) from Montpellier 2 University, France, and the PhD degree in computer science (computer vision and artificial intelligence) from Claude Bernard Lyon 1 University, France, in December 2016. He is currently a research scientist of computer vision and machine learning with the Department of Research and Innovation (R&I) of Umanis. During his PhD, he was a visiting researcher for six months with Ottawa University and the National Research Council Canada, Ottawa, Canada. From September 2016 to August 2018, he was an assistant professor and a researcher of computer science with Jean Moulin Lyon 3 University and the LIRIS Laboratory, Lyon, France. He is the author of numerous publications in international journals and conferences. His research interests include graph-based approaches (graph matching, graph edit distance, graph metric learning, graph decomposition, and graph-based modeling) combined with machine learning and deep learning techniques, for 2D and 3D pattern recognition, addressing various applications, such as 3D deformable objects and protein shapes classification.

Junzheng Wu received the BS degree in software engineering from Southwest Jiaotong University, China, in 2018. He is currently working toward the PhD degree with the Department of Computer Science, University of Ottawa, Canada. His research focuses on deep learning applications for biology.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.