Computers in Biology and Medicine 182 (2024) 109192

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Using 3D point cloud and graph-based neural networks to improve the estimation of pulmonary function tests from chest CT

Jingnan Jia [a], Bo Yu [a,b,f], Prerak Mody [a], Maarten K. Ninaber [c], Anne A. Schouffoer [d], Jeska K. de Vries-Bouwstra [d], Lucia J.M. Kroft [e], Marius Staring [a], Berend C. Stoel [a,*]

[a] *Division of Image Processing, Department of Radiology, Leiden University Medical Center, PO Box 9600, 2300 RC, Leiden, The Netherlands*
[b] *School of Artificial Intelligence, Jilin University, 130015, Changchun, China*
[c] *Department of Pulmonology, Leiden University Medical Center, PO Box 9600, 2300 RC, Leiden, The Netherlands*
[d] *Department of Rheumatology, Leiden University Medical Center, PO Box 9600, 2300 RC, Leiden, The Netherlands*
[e] *Department of Radiology, Leiden University Medical Center, PO Box 9600, 2300 RC, Leiden, The Netherlands*
[f] *Engineering Research Center of Knowledge-Driven Human–Machine Intelligence, Ministry of Education, China*

ARTICLE INFO

Dataset link: https://github.com/Jingnan-Jia/PFT_regression

Keywords:
Pulmonary function test
Systemic sclerosis
Computed tomography
Deep learning
Point cloud
Graph
Lung vessels

ABSTRACT

Pulmonary function tests (PFTs) are important clinical metrics to measure the severity of interstitial lung disease for systemic sclerosis patients. However, PFTs cannot always be performed by spirometry if there is a risk of disease transmission or other contraindications. In addition, it is unclear how lung function is affected by changes in lung vessels. Therefore, convolution neural networks (CNNs) were previously proposed to estimate PFTs from chest CT scans (CNN-CT) and extracted vessels (CNN-Vessel). Due to GPU memory constraints, however, these networks used down-sampled images, which causes a loss of information on small vessels. Previous literature has indicated that detailed vessel information from CT scans can be helpful for PFT estimation. Therefore, this paper proposes to use a point cloud neural network (PNN-Vessel) and graph neural network (GNN-Vessel) to estimate PFTs from point cloud and graph-based representations of pulmonary vessel centerlines, respectively. After that, we combine different networks and perform multiple variable step-wise regression analysis to explore if vessel-based networks can contribute to the PFT estimation, in addition to CNN-CT. Results showed that both PNN-Vessel and GNN-Vessel outperformed CNN-Vessel, by 14% and 4%, respectively, when averaged across the intra-class correlation coefficient (ICC) scores of four PFTs metrics. In addition, compared to CNN-Vessel, PNN-Vessel used 30% of training time (1.1 h) and 7% parameters (2.1 M) and GNN-Vessel used only 7% training time (0.25 h) and 0.7% parameters (0.2 M). We combined CNN-CT, PNN-Vessel and GNN-Vessel with the weights obtained from multiple variable regression methods, which achieved the best PFT estimation accuracy (ICC of 0.748, 0.742, 0.836 and 0.835 for the four PFT measures respectively). The results verified that more detailed vessel information could provide further explanation for PFT estimation from anatomical imaging.

1. Introduction

Systemic sclerosis (SSc) is a rare immune-mediated connective tissue disease that affects different organs. Up to 90% of SSc patients have lung involvement, with pulmonary function abnormalities such as interstitial lung disease (ILD) [1]. To evaluate progression of SSc-ILD, various pulmonary function tests (PFTs) are used as key measures, such as diffusion capacity for carbon monoxide (DLCO), forced expiratory volume in 1 s (FEV_1), forced vital capacity (FVC) and total lung capacity (TLC) [1–3]. PFTs, however, cannot always be performed if there is a risk of disease transmission, e.g. in patients with COVID-19 [4,5] or other contraindications such as myocardial infarct, pulmonary embolism or ascending aortic aneurysm [6,7].

CT scans provide high-resolution details of the lungs, and is hence regarded the gold standard for diagnosing SSc-ILD [9,10]. Therefore, when PFTs are not possible and CT scans have been made previously of SSc patients for other purposes, it could be of interest to estimate PFTs from CT images. In addition, the clinical research community is interested to investigate the relation between structural (especially

---

* Corresponding author.
E-mail addresses: j.jia@lumc.nl (J. Jia), byu20@mails.jlu.edu.cn (B. Yu), P.P.Mody@lumc.nl (P. Mody), M.K.Ninaber@lumc.nl (M.K. Ninaber), A.A.Schouffoer@lumc.nl (A.A. Schouffoer), j.k.de_vries-bouwstra@lumc.nl (J.K. de Vries-Bouwstra), L.J.M.Kroft@lumc.nl (L.J.M. Kroft), m.staring@lumc.nl (M. Staring), b.c.stoel@lumc.nl (B.C. Stoel).

https://doi.org/10.1016/j.compbiomed.2024.109192
Received 19 March 2024; Received in revised form 23 August 2024; Accepted 21 September 2024

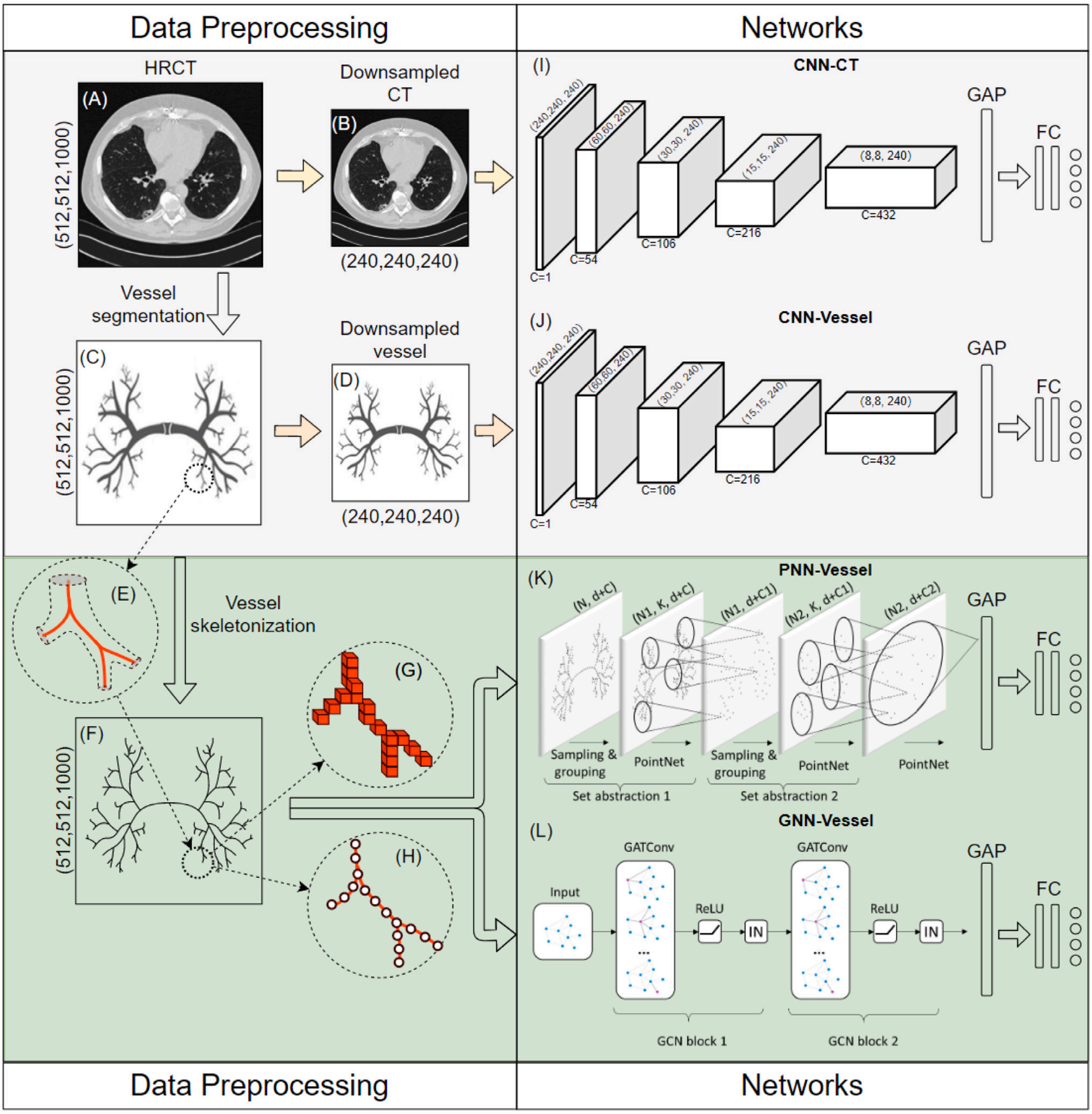

Fig. 1. Overview of the four PFT estimation methods. The data-preprocessing steps are on the left and neural network architectures on the right. The existing methods [8] are shown on the upper part (gray background), the work proposed in this paper is on the bottom part (green background). The shape of each grid image is marked next to it. Each of the four networks consists of an encoder and a global average pooling (GAP) layer, followed by a fully-connected (FC) block. The FC block includes two fully connected layers with 1024 nodes for each of them. The four output nodes represent DLCO, FEV$_1$, FVC and TLC, respectively.

vascular) changes due to SSc and pulmonary function. Previous work has extracted some quantitative biomarkers from chest CT of SSc patients, which correlated with PFTs [3,10–12]. For instance, two lung vascular tree-based biomarkers, extracted from the lung vessel radius histogram, were found to be correlated with DLCO [11]. However, these biomarkers are investigated only in terms of their correlations with PFTs but do not estimate PFT values directly. Thus, there is a need to investigate the use of high-resolution pulmonary vessel masks from CT scans to directly estimate PFTs.

Deep neural networks have shown some promise for PFT estimation from CT scans [4,8,13] due to their powerful capability to extract features from images. To the best of our knowledge, there are just three works on the PFT estimation using deep learning from CT scans. The first work is for patients with lung cancer [4]. A network was developed to estimate FEV$_1$ and FVC, which achieved R values of 0.73 and 0.82, respectively. Their network consisted of a ResNet-50 [14] for feature extraction and a bidirectional long short-term memory (BiLSTM [15]) network for PFTs prediction. The second work is for subjects at risk of lung cancer [13]. Two separate networks were trained to estimate FEV$_1$ and FVC, respectively. Both of the previous methods estimated FEV$_1$ and FVC only, lacking DLCO and TLC. In addition, none of them were developed for SSc patients. The third work for PFT estimation is for SSc patients [8]. The potential of convolution neural networks (CNN) was verified to estimate the complete set of PFTs. Intra-class correlation coefficient (ICC) values were achieved for DLCO, FEV$_1$, FVC and TLC as 0.71, 0.76, 0.80 and 0.81, respectively, from CT scans of SSc patients (see Fig. 1-I, CNN-CT). In addition, motivated by the manually extracted vascular tree-based biomarkers [11], it was verified that CNN trained by 3D binary vessels could also be used to estimate complete PFTs (with lower accuracy) [8] (see Fig. 1-J, CNN-Vessel).

The limitation of this previous work for SSc patients [8] is that the 3D images were down-sampled before being fed into networks, due to GPU memory limitation (see Fig. 1-B and D). This led to approximately 20× loss of detailed vessel information. Down-sampling is a compromise that has to be made, because even with a batch size of 1, the GPU usage was still at least 11 GB, and to use the original resolution CT, a single GPU of at least 220 GB is needed, which does not yet exist. Because the absence of small vessels was reported to influence PFTs significantly [11], we assumed that the better performance could be achieved by overcoming the information loss of small vessels during the development of PFT-estimation networks. Therefore, the goal of this study was to explore the possibility of improving the deep-learning-based PFT estimation performance by efficiently utilizing detailed vessel information.

Given that most voxels of 3D binary vessel images are background, we propose to convert binary vessel images to two different data formats — point clouds and graphs, to efficiently utilize the relevant vessel voxels. A point cloud is a sparse representation of the binary vessel image and contains structural information on the pulmonary vessels. Such a representation could help networks on extracting structural features relevant to PFT estimation. However, point clouds do not explicitly represent the geometric relationships between points. Given that points in vessels are not completely independent but belong to different sub-branches, such information could be useful to PFT-estimation networks. Therefore, we also built graphs of vessels by using edges to explicitly connect the points in the same sub-branches.

Thus, our contributions are as follows

- We, for the first time, propose to use a point cloud neural network (PNN-Vessel) and a graph neural network (GNN-Vessel) to estimate PFTs from pulmonary vessels.
- We explored the architectures and training strategies for PNN-Vessel and GNN-Vessel.
- We explored how pulmonary vasculature influences pulmonary function and verified that higher resolution of vessels, which include more small vessels, could lead to higher PFT estimation accuracy.
- We, for the first time, successfully combined CNN, PNN and GNN together and achieved the best PFT estimation performance.

The remaining paper is organized as follows. In Section 2, the related work was reviewed. Then, our solution for the PFT estimation based on point clouds and graphs is described in Section 3. Detailed experiment setting and results are shown in Section 4. Finally, in Section 5, the experiment results are discussed.

2. Related work

2.1. Point cloud and point neural network (PNN)

A 3D point cloud is a set of discrete data points in space, which is commonly used in 3D scanners, LIDAR and RGB-D cameras [16]. In a 3D point cloud, each point position has its set of Cartesian coordinates (x, y, z) and other extra features like color and depth [17]. A point cloud is stored as an unordered set of vectors with shape of (N, D+C), where N is the number of points, D is the dimension of the coordinates (normally 3 for points in 3D space) and C is the number of extra features.

In the past years, a great number of different neural networks were proposed for point cloud data. PointNet is a pioneering deep neural network that directly analyzes point clouds for both classification and segmentation [18]. The basic idea of PointNet is to learn a spatial encoding of each point through a sequence of shared multi-layer perceptions (MLPs) and then aggregate all individual point features to a global point cloud signature by a global pooling [18]. Before the global pooling, each point is processed identically and independently, which ensures invariance to permutations but also ignores the local context. To overcome this limitation, PointNet++ [19] was released subsequently and became one of the most influential networks by leveraging local neighborhoods at multiple scales. PointNet++ hierarchically abstracts features of point clouds using a number of set abstraction (SA) blocks. An SA block consists of a subsampling layer to down-sample the incoming points, a grouping layer to query neighbors for each point, and a simplified PointNet to extract and aggregate features. Because PointNet++ affirmed the significance of leveraging local neighboring information, more networks were proposed to aggregate spatially-local correlation information. In PointCNN [20], an X-Conv operator was introduced, which can weight and permute input features before they are processed by a typical convolution. This is the generalization of a CNN on the point cloud domain. Inspired by the success of transformers in natural language processing [21] and image analysis [22], Point Transformer [23] networks were designed with self-attention layers for point clouds.

The latest network, which achieved the best performance on most of the benchmarks, is PointNeXt [24]. It used the same design of PointNet++ [19] but applied different hyper-parameters and training strategies for different tasks. It concluded that by just optimizing training strategies (e.g. data augmentation and optimization techniques) for different tasks, PointNet++ could exceed the current state of the art. Inspired by this works, the PNN-Vessel described in this paper also uses the design of PointNet++ and explores the optimal training strategies and hyper-parameters for PFT estimation.

2.2. Graph and graph neural network (GNN)

A graph is a data structure that models a set of objects (nodes) and their relationships (edges) [25]. Because of the unstructured nature of graph data, CNNs cannot be applied directly to graphs. Therefore, a great number of networks were proposed to mimic the principle of CNNs by aggregating information from neighboring nodes using different aggregation strategies [26–30]. ChebConv [30] is one of the earliest attempts, and approximates spectral graph convolutions using Chebyshev polynomials. GCN [27] simplifies ChebConv by utilizing only the first two Chebyshev polynomials while still outperforming it on real-world datasets. The core operations of a GCN is aggregating (average or maximum) neighboring features, followed by an MLP layer to increase/decrease feature dimensions and a non-linear activation function. Simplifying Graph Convolution (SGConv) [26] is a simplified version of GCN, which aims to reduce computational costs without sacrificing too much performance. GraphConv [28] was proposed as a hierarchical version of k-GNNs, based on the k-dimensional Weisfeiler-Lehman (WL) algorithm, which is able to work with the fine- and coarse-grained structures of a given graph. Graph Isomorphism Network Convolution (GINConv) [29] implements graph isomorphism tests [31] in a neural network. Graph Attention Network Convolution (GATConv) [32] was introduced with the concept of self-attention mechanisms to graph convolutions. Each node computes attention coefficients with all of its neighbors, allowing different neighbors to contribute differently to the updated node representation. The key difference between PNN-Vessel and GNN-Vessel is that PNN-Vessel regards all neighboring points that have nearest distances, whereas GNN-Vessel only considers points as neighbors if they have an direct edge between them. This is critical in our task where two neighboring points in a vascular tree should belong to the same vessel branch. Therefore, we proposed the GNN-Vessel to estimate PFTs from our built vessel graph dataset.

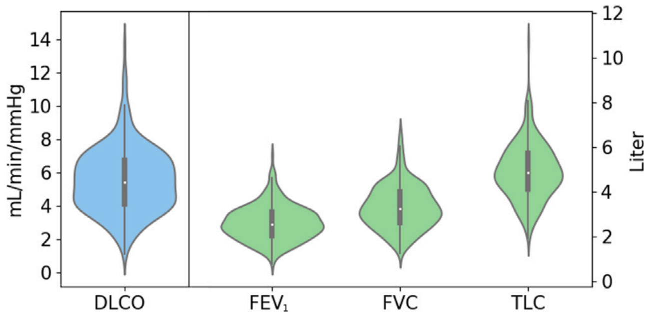

Fig. 2. Measured PFTs distribution on the whole dataset.

3. Methodology

In this section, we describe how to convert 3D grid vessel images to point cloud and graph datasets. We then separately illustrate our two proposed networks: PNN-Vessel and GNN-Vessel.

Two previously developed for PFT estimation are shown in Fig. 1-I and J, which were trained by down-sampled CT images and down-sampled binary vessel images separately. The two new methods developed in this paper are shown in Fig. 1-K and L. The high-resolution 3D grid vessel images were skeletonized resulting in two types of data. The first type of data is point cloud with coordinates and vessel radius as the features of each point. The second type of data is graph with coordinates and vessel radius as the features of each node, and edge between two adjacent nodes. PNN-Vessel and GNN-Vessel were then developed for the two datasets, respectively. The details of dataset preparation and network design are described below.

3.1. Acquisition of CT images and PFTs measurements

In this study, we retrospective selected a cohort of 316 patients referred to our specialized outpatient health care program (focused on combined care in systemic sclerosis) at Leiden University Medical Center. These patients, identified by the referring rheumatologists as having a definitive diagnosis of Systemic Sclerosis (SSc) or presenting with a strong clinical suspicion, were examined by high-resolution computed tomography (HRCT) scans and subsequent pulmonary function tests (PFTs) within 10 days. All subjects underwent scanning at full inspiration without contrast enhancement using an Aquilion 64 CT scanner (Canon Medical Systems), configured at 120 kVp, a median tube current of 140 mA, a rotation time of 0.4 s, a collimation of 64 × 0.5 mm and a helical beam pitch of 0.8; leading to a median CTDIvol of 8.2 mGy. The images were reconstructed with filtered back projection and an FC86 kernel, with a median pixel spacing of 0.64 mm × 0.64 mm, with a slice thickness and increment of 0.5 and 0.3 mm, respectively. PFTs were performed using a spirometer under ERS/ATS guidelines [33,34] including DLCO, FEV_1, FVC and TLC. While DLCO was measured in units of mm/Hg/min, FEV_1, FVC, TLC were measured in units of liter (Fig. 2). We divided this dataset into two distinct subsets: 253 CT-PFT pairs were allocated for four-fold training and cross-validation, while the remaining 63 pairs were reserved for the separate testing phase. Written informed consent was provided by all patients. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board of the LUMC under protocol numbers P09.003/SH/sh, REU 036/SH/sh, REU 043/SH/sh and B19.008/KB/kb. For the current specific analysis no separate research protocol was submitted.

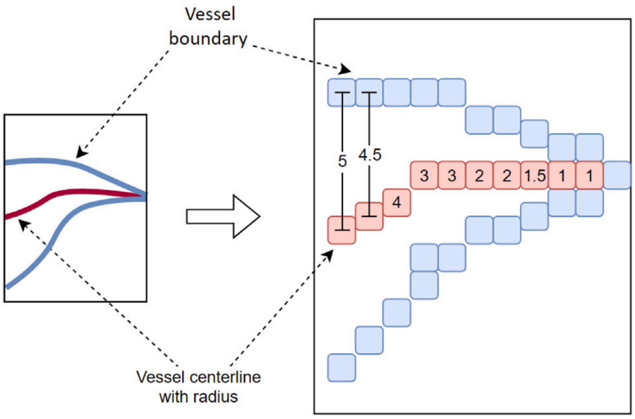

Fig. 3. Illustration of vessel centerline extraction. The left image is a vessel endpoint with the centerline in it. The right image shows the process of calculating the radius of the vessel. Each square represent a voxel in the 3D image. The number on the centerline indicates the radius of that position.

3.2. Dataset preparation

3.2.1. Dataset preparation for CNN-CT and CNN-Vessel

We reproduced previously developed CNN-CT and CNN-Vessel networks [8]. The dataset preparation for CNN-CT and CNN-Vessel are shown in Fig. 1-A, B, C, and D. After we obtained CT scans (Fig. 1-A), the corresponding binary vessels masks were automatically segmented by a graph-cut based method [35] (Fig. 1-C). This vessel segmentation method achieved the top three in VESSEL12 Challenge (Team name is “LKEBChina” in the VESSEL12 website https://vessel12.grand-challenge.org/Results). To develop CNN-CT and CNN-Vessel, we down-sampled all CT scans and binary vessel images to an isotropic spacing of 1.5 mm, then cropped 3D patches of fixed size (240 × 240 × 240 voxels, see Fig. 1-B and D).

3.2.2. Dataset preparation for PNN-Vessel

After we obtained the binary vessel masks, the centerlines of the vessel trees were extracted using skeletonization [36] with the calculated radius embedded in the centerline voxels, as illustrated in Fig. 1-E and F. The detailed calculation of vessel radius is shown in Fig. 3. At each location in the vascular tree, the radius was calculated by a skeletonization method (DtSkeletonization method of Mevislab 2.731). This method selects voxels that are located at the center of a blood vessel by eroding the extracted vessels, and the corresponding radius is estimated by measuring the distance between the vessel boundary and the center. After we obtained the centerlines, we converted the points on the centerlines to a list of vectors, each of which includes three elements of spatial coordinates and one element of the corresponding vessel radius (Fig. 1-G). Before the conversion, the average size of each 3D vessel grid image is about 512 MB with a mean size of 512 × 512 × 1000 pixels along x, y and z axes. After the conversion, the size of each image decreased dramatically to 0.3 MB. Each image is now represented as an array with a shape of $N \times 4$, where N is the number of points on the vessel centerlines, with a mean value of 75,000. 4 means the four features for each point: the coordinates along the x, y and z axes, and the radius at that point.

3.2.3. Dataset preparation for GNN-Vessel

As shown in Fig. 1-H, each vessel centerline tree is defined as a graph and the voxels of the centerlines are defined as nodes in a graph. Each node has four features: coordinates (x, y, z) and radius value (R). The edges exist between two adjacent voxels. The mean number of nodes is 75 k, with the mean number of edges of 132 k.

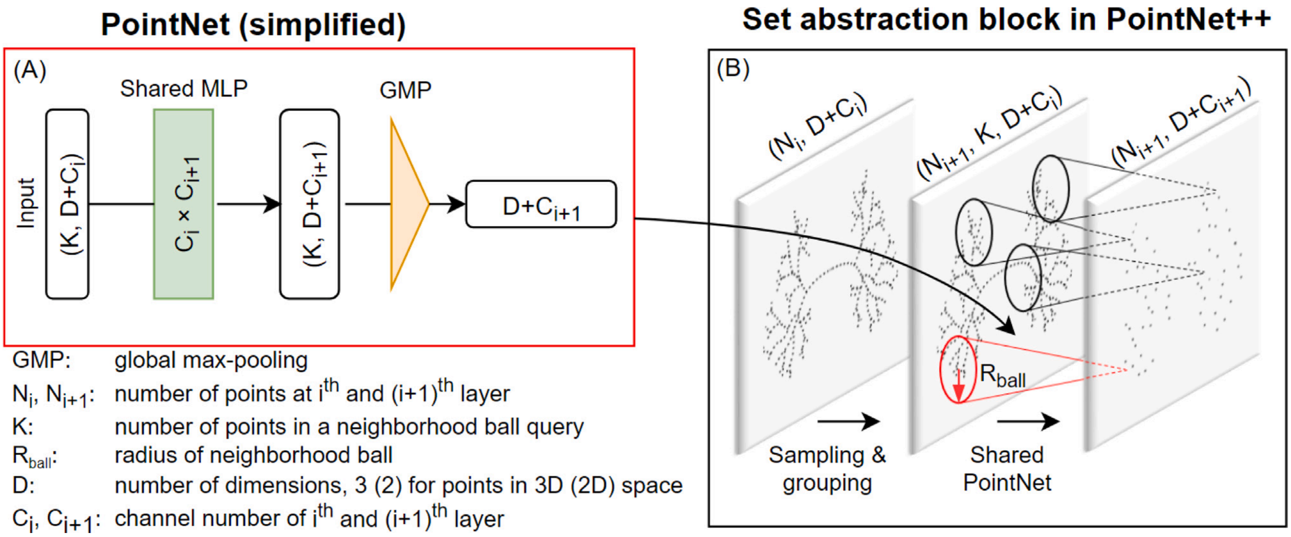

Fig. 4. Architecture of PointNet and PointNet++ [24].

3.3. PNN-Vessel design

We designed a point-cloud-based neural network named PNN-Vessel, which extracts features from 3D point clouds to regress to four PFTs values. As shown in Fig. 1 (K), it consists of an encoder, a global average pooling (GAP) layer and a decoder. The decoder consists of two FC layers with 1024 channels for each of them. The encoder consists of two set abstraction blocks. The details of the set abstraction blocks are shown in Fig. 4. The ith set abstraction take an input point cloud of size $N_i \times (D + C_i)$, representing N_i points with D-dim coordinates and C_i-dim feature for each point. It first subsamples N_{i+1} points by a farthest point sampling (FPS) strategy [19]. With the N_{i+1} points as the center, N_{i+1} groups of point sets were extracted and each group included K neighboring points. The N_{i+1} groups of point sets were then passed to a shared PointNet, resulting in N_{i+1} points with D-dim coordinates and C_{i+1}-dim features. Therefore, after a set abstraction block, the number of points was decreased from N_i to N_{i+1} while the number of features was increased from C_i to C_{i+1}. Finally, features from the last set abstraction layer are passed into a GAP layer that combines point features followed by two FC layers that output four PFTs values. Inspired by PointNeXt, we successively tuned the hyper-parameters to achieve the optimal combination of hyper-parameters.

3.4. GNN-Vessel design

Although the point cloud format offers detailed information about the structure of the pulmonary vessels, they do not leverage connectivity information, an aspect relevant to the tree hierarchy of pulmonary vessels. In the vessel centerlines, points that are physically close to each do not necessarily belong to the same vessel. To utilize the connectivity information, we built a dataset using graphs of vessel centerlines (see Section 3.2) and trained the graph neural networks to estimate PFTs. The structure of our designed GNN-Vessel is shown in Fig. 1 (L). It consists of an encoder, a global average pooling (GAP) layer and a decoder. The decoder consists of two FC layers with 1024 channels for each of them. The encoder consists of two graph convolution network (GCN) blocks. Each GCN block includes one graph attention convolution (GATConv) layer [32], followed by ReLU activation and instance normalization (IN) layers. The graph attention network (GAT) is a combination of a graph neural network and an attention layer. The attention layer helps focusing to the important information from the data instead of global data. As shown in Fig. 5, the GATConv layer consists of a shared MLP, an attention calculation block and a aggregation block, which correspond to the following equations:

$$\mathbf{x}_i' = F(\mathbf{x}_i^{in}) \tag{1}$$

$$\alpha_{ij} = softmax_j(\mathrm{ReLU}(\mathbf{x}_i \parallel \mathbf{x}_j) \cdot \mathbf{w}) \tag{2}$$

$$\mathbf{x}_i^{out} = \sum\nolimits_{j \in S_i} (\alpha_{ij} \cdot \mathbf{x}_i') \tag{3}$$

where $\mathbf{x}_i^{in}$, $\mathbf{x}_i'$, and $\mathbf{x}_i^{out}$ are the input, intermediate and output features of node i. $F(\cdot)$ represents the MLP layer which is shared by all nodes. α_{ij} is the attention weight between node i and j. $\parallel$ denotes the concatenation operation. N_i is the neighborhood of node i in the graph (see Fig. 5).

3.5. Combination of different networks

After we developed PNN-Vessel and GNN-Vessel, it is of great interest to see if the combination of different networks could result in better performance. We proposed two methods to verify this.

In the first method, we developed a combined network (see Fig. 6) which consists of several different encoders, corresponding to the pre-trained feature extractors for different data formats, and one decoder, which consists of two FC layers. A GAP layer, standardization layer and concatenation layer were used to fuse the features from different encoders. Because of GPU memory limitation, CNN-CT and CNN-Vessel could not be in the encoder at the same time. Given that all image information of vessel has already been included in CT, we chose to exclude CNN-Vessel from the combined network. The three encoders were initialized with the weights from the trained single CNN-CT, PNN-Vessel and GNN-Vessel, separately. The FC layers were initialized from scratch. The whole combined network was trained by 100 epochs with learning rate of 1e-4 and batch size of 1.

In the second method, we performed multiple variable regression analysis to evaluate if the vessel based networks could contribute further to the estimation of PFTs, in addition to the estimation of CNN-CT. Multiple variable stepwise linear regression (a statistical technique used to measure the impacts of several explanatory variables on a dependent variable) was performed with DLCO as the dependent variable and the estimated DLCO from CNN-CT, CNN-Vessel, PNN-Vessel and GNN-Vessel as independent variables. We performed the same analyses for FEV_1, FVC and TLC as dependent variables.

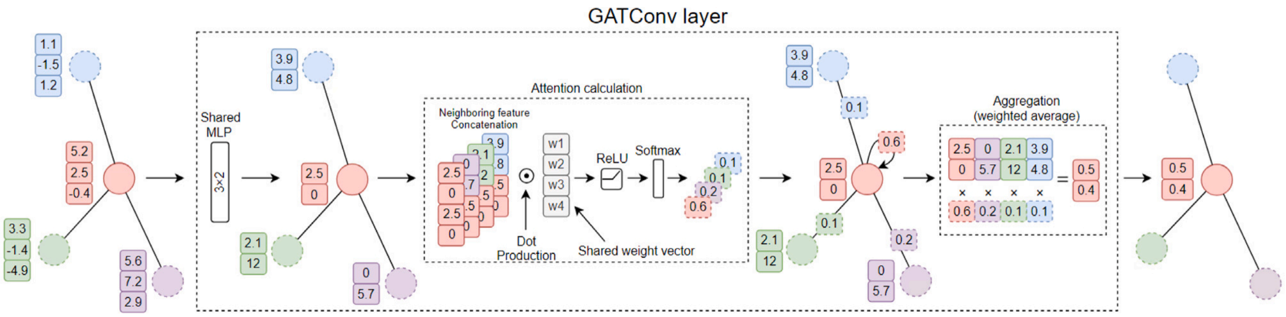

Fig. 5. Illustration of the graph attention convolution (GAT) layer [32]. Different colors correspond to different nodes. The feature vector of each node is shown next to the node with the same color.

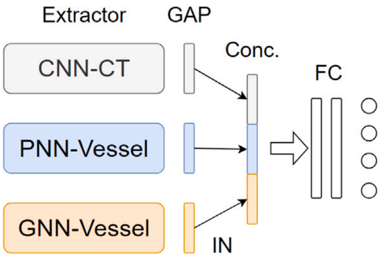

Fig. 6. Illustration of the combined network. Global average pooling (GAP) was performed on the features from different extractors. After that, these features become 1-dimension vectors. Each of them were then performed instance normalization (IN) before concatenation (Conc.). The concatenated 1-dimension features were then fed into two fully-connected (FC) layers.

3.6. Evaluation metrics and statistical analysis

We used various metrics to evaluate the agreement between our network output and measured values (from spirometry). The mean absolute error (MAE) was used to reflect the absolute agreement. Since the unit and scale of the four PFTs are different, we also used the mean absolute percentage error (MAPE), which is the ratio of MAE to the real measurements, to reflect the relative uncertainty of prediction. MAE and MAPE were calculated as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|, \tag{4}$$

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\frac{\left|Y_i - \hat{Y}_i\right|}{Y_i}, \tag{5}$$

where $i \in N$ is the index of samples, N represents the total number of samples, $\hat{Y}_i$ is the network's estimated value, and Y_i the measured PFTs value.

We used the Pearson correlation coefficient (R) to indicate the linear correlation. The coefficient R can be interpreted as negligible (R < 0.1), weak (0.1 to 0.39), moderate (0.4 to 0.69), strong (0.7 to 0.89) or very strong (over 0.9) [37]. We also used the intra-class correlation coefficient (ICC) as a measure of reliability, which represents not only the linear correlation, but also the absolute agreement. ICC was calculated by Pingouin 0.4.0 [38] based on a single-rating, absolute-agreement, 2-way mixed-effects model [24]. ICC values can be explained as poor (below 0.5), moderate (between 0.5 and 0.75), good (between 0.75 and 0.9) and excellent (above 0.9) reliability [39]. Bland-Altman plots were used to analyze the mean differences (bias) and limits of agreement. These statistical analyses were performed by an in-house python 3.8 script with corresponding libraries. All the following experiment results are based on four-fold cross-validation unless otherwise stated. The multi-variable step-wise linear regression provide the extra regression performance on the hold-out testing dataset.

4. Experiments and results

We conducted three sets of experiments to explore: **E1**. training strategies and hyper-parameters for PNN-Vessel, **E2**. training strategies and hyper-parameters for GNN-Vessel and **E3**. strategies to combine different networks.

4.1. Experimental settings

Our neural networks were implemented using PyTorch 1.11.0 (https://pytorch.org). Mixed-precision [40] was applied during training to minimize the occupation of GPU memory and accelerate training. The loss function was the mean squared error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{6}$$

where $i \in N$ is the index of samples, Y_i is the vector of measured PFTs, $\hat{Y}_i$ is the vector of predicted PFTs. The Adam optimizer was used with 100 epochs. Multiprocessing was used to accelerate on-the-fly data augmentation. The workstation for training and validation was equipped with an Intel(R) Xeon(R) CPU Gold 2.6 GHz with 90 GB memory and a NVIDIA GPU GeForce RTX 2080TI with 11 GB memory. The trained networks and source code for network development are publicly available at https://github.com/Jingnan-Jia/PFT_regression for the convenience of reproducing our method, applying our model to other datasets or using our model as a pre-trained model for other tasks.

4.2. E1: Training strategies and hyper-parameters for PNN-Vessel

Because this is the first work on point cloud based network for PFT estimation, it is unclear what are the optimal training strategy and data augmentation techniques. As we mentioned in Section 3.3, instead of inventing a new network architecture, we adopted PointNet++ as the backbone of PNN-Vessel and explored the best combination of different hyper-parameters and data augmentation techniques. Inspired by PointNeXt, the explored techniques include (see Table 1):

- **Random scaling** (±10%). A method for data augmentation, which randomly scales the coordinates of the **whole** point cloud by a factor in ±10%.
- **Random shifting** (±3 mm). A method for data augmentation, which randomly shifts the coordinates of the **whole** point cloud within 3 mm.
- **L2 Loss regularization (0.01)** [41]. It stabilizes training by reducing the sum of the squares of the trainable parameters. The equation of the loss function after the introduction of L2 Loss regularization with weight of 0.01 is

$$L = MSE + 0.01 \cdot \sum_{i=1}^{N} w_i^2, \tag{7}$$

where MSE is the original loss function of PNN-Vessel, N is the number of trainable parameters, and w_i is the weight of ith parameter, 0.01 is the weight of L2 loss regularization.

Table 1

Hyper-parameter optimization results for PNN-Vessel based on four-fold cross-validation.

Techniques	ICC				ΔICC[a]	Decision[b]
	DLCO	FEV_1	FVC	TLC		
Baseline[c]	0.379	0.427	0.473	0.449		✓
+ Random scaling (±10%)	0.266	0.301	0.368	0.344	−0.112	✗
+ Random shifting (±3 mm)	0.378	0.450	0.439	0.445	−0.004	✗
+ L2 Loss regularization (0.01)	0.331	0.455	0.446	0.420	−0.015	✗
+ CCNorm	0.395	0.483	0.540	0.562	+0.062	✓
Input points → 56 K	0.456	0.585	0.640	0.652	+0.089	✓
R_{ball} → 40 mm	0.520	0.602	0.652	0.693	+0.034	✓
LR decay → Cosine	0.531	0.583	0.630	0.664	−0.015	✗
Optimizer → AdamW	**0.544**	**0.610**	**0.683**	**0.726**	+0.024	✓

[a] ΔICC: Average change of ICC, comparing to the previous best performance.

[b] Decision: Include the technique (✓) in the following experiments if Δ ICC > 0; exclude the technique (✗) if Δ ICC ≤ 0.

[c] Baseline: batch size is 20; Number of input points is 7 K; R_{ball} is 10; Optimizer is Adam.

- **Coordinate center normalization (CCNorm).** A method for data augmentation, which makes the mean of the coordinates of the whole point be zero. The equation of CCNorm is

$$Y_{norm} = Y - \overline{Y} \tag{8}$$

where Y represents the original coordinates, and $\overline{Y}$ is the mean of coordinates.
- **Tuning the number of input points.** As we mentioned at the end of Section 3.2, the number of points for different samples are different. To align input data, we randomly selected a fixed number (N_i in Fig. 4) of points for each patient to feed PNN-Vessel for each training iteration. For points less than N_i, some points of the patient were repeatedly sampled. Higher number of input points means more detailed vessel information.
- **Tuning the radius of neighboring query balls** [19]. As shown in Fig. 4, increasing R_{ball} could increase the receptive field for each set abstraction layer. The principle of tuning R_{ball} for PNNs is just like adjusting the kernel size for CNNs.
- **Cosine learning rate (LR) decay** [42]. LR is decayed with a cosine annealing for each batch.
- **AdamW optimizer** [43]. It was reported to perform better optimization than Adam Optimizer [43].

We did not explore the influence of jittering or label smoothing, which were used in PointNeXt [24], because jittering would destroy the continuity of blood vessels and label smoothing is for classification tasks instead of our PFTs regression tasks.

Table 1 shows the influence of the different parameters. The techniques improving the PNN-Vessel performance included using CCNorm, increasing the number of input points from 7 K to 56 K, increasing R_{ball} from 10 mm to 40 mm, and replacing optimizer Adam with a new optimizer AdamW. The techniques that do not improve PNN-Vessel performance included random scaling (±10%), random shifting (±3 mm), L2 Loss regularization (0.01), cosine LR decay. Fig. 7 shows the scatter plot and Bland-Altman plot of the final optimal network. The output of PNN-Vessel shows moderate correlation with DLCO and FEV_1 (R=0.57 and 0.65) and strong correlation with FVC and TLC (R=0.71 and 0.74). The mean differences for all PFTs (0.03, 0.07, 0.09 and 0.01) are close to 0, indicating that there is no systematic bias between the estimated and measured PFTs.

When we compare the PFT estimation performance between CNN-Vessel (Table 2-i) and PNN-Vessel (Table 2-v), the ICC values of four PFTs were improved from 0.51, 0.52, 0.59 and 0.62 to 0.54, 0.61, 0.68 and 0.73, respectively. Nevertheless, because they have different network structures and different numbers of input points, it is not clear whether such an improvement is due to different network design or different input resolutions. Therefore, we conducted a series of experiments with the number of input vessel points gradually increasing from 7 K to 70 K (higher number of input points means higher resolution and more details of vessels). The results are shown in Table 2 (higher numbers than 70 K were not applied because of GPU memory limitation). When we gradually increase the number of input points from 7 K to 56 K, the PFT estimation performance is gradually increasing as well. When we increased the number of input points from 56 K to 70 K, the network did not show significant improvements anymore (DLCO and FVC predictions improved slightly while predictions for FEV_1 and TLC slightly decreased). When we compare Table 2-i and ii, we found that given the similar number of input vessel points, CNN-Vessel performed better than PNN-Vessel. However, when the number of points was increased gradually, the accuracy of PNN-Vessel grew steadily and finally outperformed CNN-Vessel.

4.3. E2: Training strategies and hyper-parameters for GNN-Vessel

Since training GNN is much faster than PNN-Vessel, we used Optuna (https://optuna.org) to automatically search the optimal hyper-parameters for GNN-Vessel. The search algorithm was set to Tree-structured Parzen Estimator (TPE) [44], which was reported to outperform both manual and random search algorithms. The hyper-parameters that may influence the performance of GNN-Vessel include learning rate, batch size, number of convolution layer, convolution kernel and normalization method. Possible data augmentation techniques include random scaling, random shifting and random sampling. Random scaling or shifting has already reported not to improve the PFT estimation performance (see Table 1). Random sampling (e.g. sampling 56 K nodes from the whole graph) would take substantially longer time, because each sampling of points require the re-indexing of remaining edges. Therefore, we decided not to introduce data augmentation techniques for GNN-Vessel. Therefore, the search space for possible hyper-parameters are shown in Table 3. The hyper-parameter search stopped after 100 search iterations. We selected some intermediate results to show the influence of different hyper-parameters in Table 4. The best PFT estimation performance was achieved by increasing the batch size from 8 to 32, increasing hidden channels from 32 to 128, increasing convolution layers from 1 to 2, increasing learning rate from 1e-3 to 1e-4, changing the normalization layer from batch normalization to instance normalization [45] and changing the convolution kernel from GCNConv to GATConv layer [32]. Fig. 8 shows the scatter plot and Bland-Altman plot of the final optimal GNN-Vessel. The output of GNN-Vessel shows moderate correlation with DLCO, FEV_1, FVC and TLC (R=0.54, 0.58, 0.62 and 0.68).

4.4. E3: Combination of different networks

The comparison of different networks proposed in this work (PNN-Vessel and GNN-Vessel) and existing methods (CNN-CT and CNN-Vessel) are shown in Table 5. CNN-CT and CNN-Vessel used the same training time (3.7 h) and have the same trainable parameters (30.1 M). This is because they have the same network design and input sizes. Compared to CNN-Vessel, PNN-Vessel used about 30% training time (1.1 h) and 7% parameters (2.1 M). GNN-Vessel used 7% training time (only 0.25 h) and 0.7% parameters (0.2 M). Notably, although GNN-Vessel receive additional edge information, its performance is worse than PNN-Vessel. Nevertheless, both PNN-Vessel and GNN-Vessel outperformed CNN-Vessel, by 14% and 4%, respectively, when averaged across the ICC scores of four PFTs metrics.

Table 6 shows the results of the combined network with different feature extractors. It shows that leaving out GNN-Vessel could achieve better performance than leaving out PNN-Vessel, which indicates the PNN-Vessel contributes more than GNN-Vessel. Although none of the combined networks outperformed the individual CNN-CT in Table 5 on the average ICC or R values. Most of the MAE values of the combined networks were lower than the counterparts in the individual CNN-CT in Table 5.

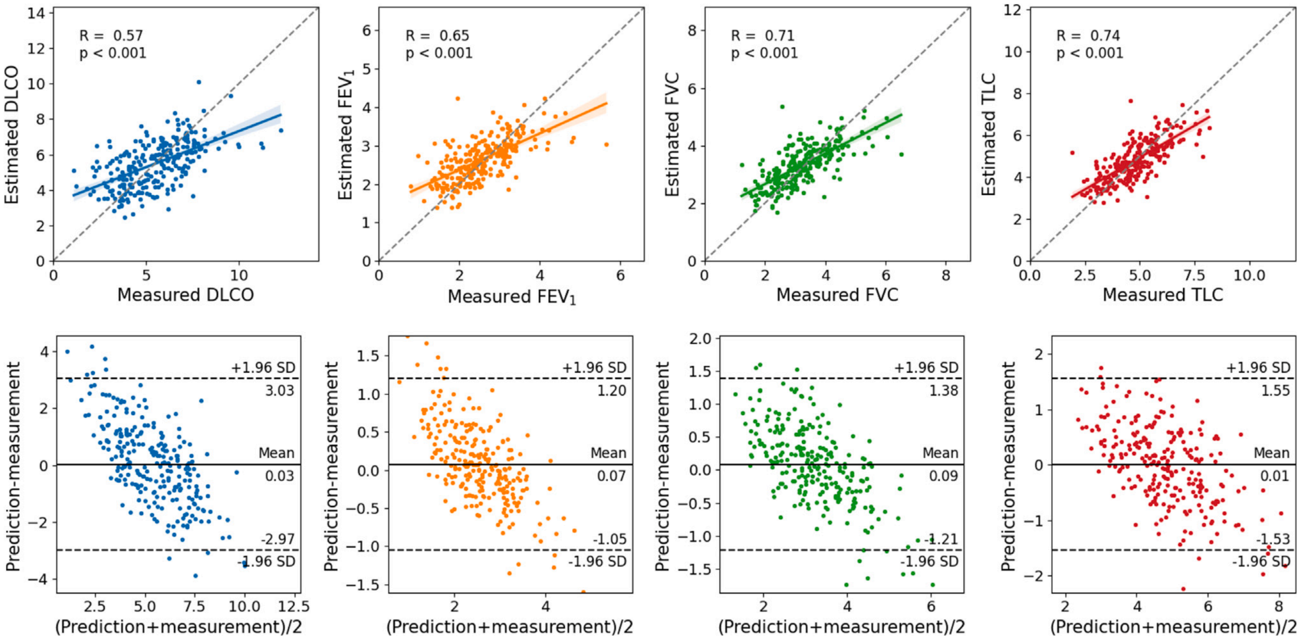

Fig. 7. PFTs estimation results of PNN-Vessel based on the optimized setting. Upper: scatter plots with the identify line (dotted line), regression line (solid line) and the 95% confidence intervals (shaded areas). Lower: corresponding Bland-Altman plots with the mean difference and the limits of agreement (mean ±1.96 × SD, where SD is the standard deviation of the differences).

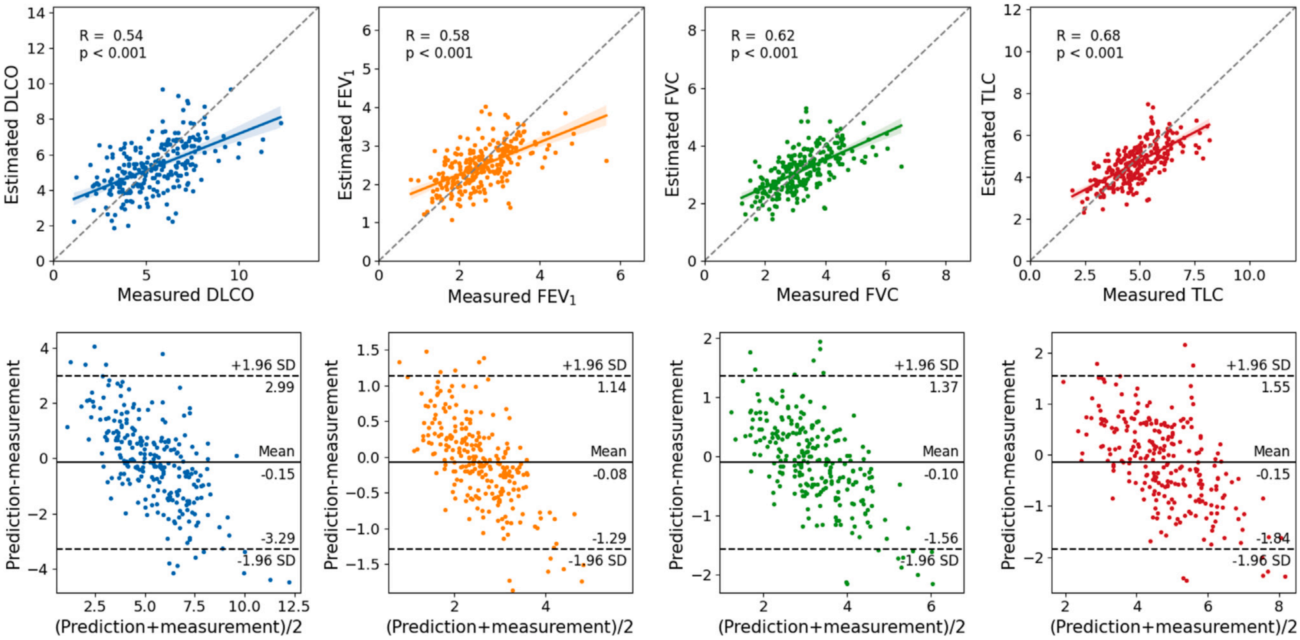

Fig. 8. PFTs estimation results of GNN-Vessel based on the optimized setting. Upper: scatter plots with the identify line (dotted line), regression line (solid line) and the 95% confidence intervals (shaded areas). Lower: corresponding Bland-Altman plots with the mean difference and the limits of agreement (mean ±1.96 × SD, where SD is the standard deviation of the differences).

Table 2
Influence of number of points in PNN-Vessel based on four-fold cross-validation.

Experiment	Network	Input points (K)	Training time (h)	Params. (M)	GPU memory (GB)	ICC DLCO	FEV$_1$	FVC	TLC
i	CNN-Vessel	≈ 7*	4.4	30.2	10.2	0.510	0.524	0.593	0.622
ii	PNN-Vessel	7	1.8	2.1	2.6	0.245	0.373	0.488	0.578
iii	PNN-Vessel	14	2.0	2.1	3.4	0.314	0.482	0.577	0.608
iv	PNN-Vessel	28	2.7	2.1	5.1	0.484	0.513	0.602	0.664
v	PNN-Vessel	56	4.0	2.1	8.8	0.544	0.610	0.683	0.726
vi	PNN-Vessel	70	4.2	2.1	10.8	0.550	0.602	0.694	0.720

* Average number of voxels for each down-sampled vessel image.

Table 3
Hyper-parameter search space setting for GNN-Vessel. **Bold** values were finally selected after hyper-parameter search.

Hyper-parameters	Search space (candidate values)
Learning rate	**1e-3**, 1e−4
Batch size	8, 16, **32**, 64
Hidden channels	32, 64, **128**, 256
Convolution layers	1, **2**, 3, 4
Convolution kernel	ChebConv [30], GINConv [29], GCNConv [27], **GATConv** [32], SGConv [26], GraphConv [28]
Normalization	BatchNorm [46], **InstanceNorm** [45], LayerNorm [47], GraphNorm [48], DiffGroupNorm [25]

Table 4
Hyper-parameter optimization process for GNN-Vessel based on four-fold cross-validation.

Techniques	ICC				ΔICC[b]
	DLCO	FEV_1	FVC	TLC	
Baseline[a]	0.431	0.480	0.504	0.525	
Batch size → 32	0.441	0.467	0.503	0.563	+0.009
Hidden channels → 128	0.478	0.511	0.510	0.591	+0.029
Convolution layers → 2	0.493	0.490	0.522	0.606	+0.005
Normalization → InstanceNorm	0.482	0.508	0.536	0.616	+0.008
Convolution kernel → GATConv	0.515	0.520	0.566	0.635	+0.024
Learning rate → 1e−3	**0.521**	**0.550**	**0.601**	**0.657**	+0.023

[a] Baseline hyper-parameters: learning rate of 1e−4, batch size of 8, hidden channels of 32, 1 GCNConv layer, and Batch normalization.

[b] ΔICC: Average change of ICC, comparing to the previous best performance.

Multivariable step-wise linear regression was performed via four-fold cross-validation dataset with DLCO as the dependent variable and the estimated DLCO from CNN-CT, CNN-Vessel, PNN-Vessel and GNN-Vessel as independent variables. We performed similar analyses for FEV_1, FVC and TLC (Table 7). The results in Table 7 showed that CNN-CT always performed the best among the four networks. The inclusion of CNN-Vessel could not improve the PFT estimation accuracy significantly (this is why it was not in Table 7). The inclusion of PNN-Vessel significantly improved the performance for all measures, while the additional inclusion of GNN-Vessel further improved the performance for DLCO and TLC. After we obtained regression equations from validation dataset, we applied them to the separate testing dataset (see Table 7). The testing performance shows the similar tendency with validation performance. We can also observe that the CNN-CT was always assigned the highest weights, followed by PNN-Vessel and GNN-Vessel. This indicates that CNN-CT contributes the most to PFT estimation, PNN-Vessel contributes less, and GNN-Vessel contributes the least.

5. Discussion

The value of binary vessel masks in estimating PFTs were preliminary verified in the literature by a manually designed method [11] and a CNN-based automatic method [8]. The estimation accuracy of the previous state-of-the-art CNN-based method was assumed to be (partly) limited by down-sampled vessels because of the GPU limitation. These finding and hypothesis motivated our research to use PNN and GNN to achieve higher estimation accuracy by utilizing higher-resolution vessels. In this paper, we converted high-resolution binary pulmonary vessels to point cloud and graph data, then proposed PNN-Vessel and GNN-Vessel to estimate PFTs based on these two datasets, respectively. After applying proper training strategies and hyper-parameters, both of them showed considerable improvement compared to the existing CNN-Vessel model, which was trained on low-resolution grids containing vessel masks.

Random scaling (±10%) decreased the accuracy of the PFT estimation significantly (see Section 4.2). This finding is contrary to the finding in PointNeXt [24]. This is because the tasks in [24] are independent on the scale of the point cloud data, while PFT estimation is dependent on the scale of lung vessels. Larger vessel trees normally represent larger lung sizes, which is directly related to higher PFT values [52].

When we compare Table 2-i and ii, CNN-Vessel achieved better performance than PNN-Vessel given a similar number of input points (7 K). However, when we gradually increased the number of input points for each point cloud data, the PFT estimation performance of PNN-Vessel gradually increased and finally outperformed CNN-Vessel (Table 2-i) using the same GPU (11 GB in RTX2080Ti). It verified our assumption that more detailed vessel information leads to better PFT estimation performance. However, when the number of input points was increased from 56 K to 70 K (GPU memory increased from 8.8 GB to 10.8 GB, almost reaching the GPU limit of 11 GB), not all estimations of the four PFTs increased anymore. This may suggest that 56 K points already include enough vessel details to estimate PFTs. Therefore, contrary to CNN-Vessel, PNN-Vessel, which uses only 2.1 MB trainable parameters, is able to receive high-resolution vessels and extract valuable features from them.

Although graph data include additional edge information compared to point cloud data, the GNN-Vessel performed worse than PNN-Vessel. One possible reason is their network structures differ, which leads to different learning capacities. Another reason may be because the training set of point cloud data for PNN-Vessel was augmented by random shuffling and sampling, while GNN-Vessel applied no augmentation. It means that PNN-Vessel was trained by numerous different samples, while GNN-Vessel was trained by limited number of different samples. The third reason may be because the pulmonary vessels used in this work were segmented by an automated graph-cut based method [35]. Thus, any imperfections in the segmentation would be amplified in graphs, due to imperfect connectivity. Since point clouds do not contain this explicit connectivity, PNN-Vessel may be able to deal with these imperfections.

Under the current network combination setting, the combined networks did not show significantly better performance than the single CNN-CT on the average ICC or R values. This may be because the training of different networks requires different learning rates, batch sizes, etc. For instance, we set batch size to 1 and learning rate to 1e-4 for the combined networks, while the optimal batch size for individual CNN-CT, PNN-Vessel and GNN-Vessel is 1, 20 and 32, respectively (the training of individual PNN-Vessel and GNN-Vessel could not be converged with batch size of 1) and the optimal learning rate is 1e-4, 1e-3, 1e-3, respectively (CNN-CT with learning rate of 1e-3 would encounter exploding gradients during training). Such conflicts of hyper-parameters may negatively affect the training of combined networks.

From the multi-variable step-wise regression analysis, we found that by including PNN-Vessel and GNN-Vessel, the estimation of all PFTs could be significantly improved to different extents. It verifies our assumption that more detailed vessel information could provide more explanation of PFTs. It is worth noting that the inclusion of CNN-Vessel did not improve the estimation performance of PFTs. This may be because that the information of 3D grid vessels are already included in the 3D CT images. The reason why simple multi-variable linear regression perform better than complex combined network may be that the combined network has too many features to learn. Although the features before FC layers are extracted by different networks, these features are all extracted from vessels or CT images containing vessels. Therefore, there may be a great number of repeated or similar features which increase the learning difficulty of combined networks.

PFT estimation is a topic that few researchers have explored. To the best of our knowledge, there are only three published works [4,8,13] for PFT estimation from chest CT (see Table 8). Two of them [4,13]

Table 5
PFT estimation performance comparison based on four-fold cross-validation between different networks.

Network	Training time (h)	Params. (M)	ICC				R				MAE			
			DLCO	FEV$_1$	FVC	TLC	DLCO	FEV$_1$	FVC	TLC	DLCO	FEV$_1$	FVC	TLC
CNN-CT	3.7	30.1	0.708	0.756	0.803	0.809	0.712	0.771	0.831	0.824	1.044	0.411	0.490	0.630
CNN-Vessel	3.7	30.1	0.512	0.524	0.589	0.615	0.568	0.545	0.641	0.680	1.201	0.515	0.621	0.764
PNN-Vessel	1.1	2.1	0.544	0.610	0.683	0.726	0.568	0.645	0.710	0.741	1.153	0.448	0.539	0.672
GNN-Vessel	0.3	0.2	0.521	0.550	0.601	0.657	0.544	0.579	0.617	0.676	1.241	0.490	0.593	0.715

Table 6
Ablation study for PFT estimation performance of combined network with different feature extractors based on four-fold cross-validation.

Feature extractor			ICC				R				MAE			
CNN-CT	PNN-Vessel	GNN-Vessel	DLCO	FEV$_1$	FVC	TLC	DLCO	FEV$_1$	FVC	TLC	DLCO	FEV$_1$	FVC	TLC
✓	✓		0.711	0.756	0.789	0.772	0.731	0.781	0.802	0.792	0.989	0.345	0.424	0.578
✓		✓	0.689	0.716	0.756	0.760	0.701	0.774	0.815	0.812	1.023	0.350	0.433	0.534
✓	✓	✓	0.671	0.718	0.746	0.768	0.714	0.779	0.810	0.821	1.011	0.355	0.434	0.520

Table 7
Multivariable stepwise linear regression analysis results on four-fold cross-validation and testing datasets. Numbers in bold indicates the best results for the corresponding PFTs.

PFTs	Weights of networks			Cross-validation dataset			Testing dataset		
	CNN-CT	PNN-Vessel	GNN-Vessel	R	ICC	MAE	R	ICC	MAE
DLCO	1			0.712	0.708	1.044	0.751	0.726	1.012
	0.754	0.246		0.730	0.717	0.977	**0.782**	**0.764**	**0.967**
	0.672	0.105	0.223	**0.750**	**0.732**	**0.967**	0.756	0.748	0.988
FEV$_1$	1			0.771	0.756	0.411	0.735	0.728	0.511
	0.641	0.359		**0.792**	**0.767**	**0.334**	**0.760**	**0.742**	**0.470**
FVC	1			0.831	0.803	0.490	0.821	0.801	0.480
	0.642	0.358		**0.842**	**0.825**	**0.378**	**0.827**	**0.836**	**0.374**
TLC	1			0.824	0.809	0.630	0.827	0.801	0.600
	0.504	0.496		0.840	0.832	0.489	0.828	0.787	0.625
	0.455	0.387	0.158	**0.845**	**0.837**	**0.478**	**0.851**	**0.835**	**0.489**

Table 8
Comparison between our method (combination of three networks based on multivariable linear regression results) and previous automatic methods for the estimation on PFTs. Because previous methods did not estimate DLCO and TLC, the corresponding results are not included.

Method	Study population	Backbone	#Subjects	MAE				R			
				DLCO	FEV$_1$	FVC	TLC	DLCO	FEV$_1$	FVC	TLC
[4]	Lung cancer	CNN (ResNet) [49]	546	–	0.329	0.370	–	–	0.729	0.822	–
[13]	Lung cancer risk	CNN (I3D) [50]	16 148	–	0.220	0.220	–	–	–	–	–
CNN-CT [8]	SSc	CNN (X3D) [51]	316	1.012	0.511	0.480	0.600	0.751	0.735	0.821	0.827
Our method	SSc	CNN, PNN, GNN	316	0.988	0.470	0.374	0.489	0.756	0.760	0.827	0.851

are for patients with (the risk of) lung cancer and only FEV$_1$ and FVC were estimated in the two works. The other one [8] is for patients with SSc, and it estimated four PFTs using a CNN (X3D) network. These three works used different kinds of CNNs (ResNet [7], I3D [9] and X3D [10] respectively) as the backbones. None of them proposed new networks as well. Compared to them, we are the first to estimate PFTs using PNN and GNN instead of CNN. Our method is also the first work which combined CNN, PNN and GNN together and achieved the better performance than CNN [8] alone. The comparison between our method and CNN-CT [8] is fair because we used the same dataset. However, the results of [4,13] are for reference only, since they are based on different datasets sizes, networks and diseases. Nevertheless, compared with the previous works which, our work estimates a more comprehensive set of PFT measurements for SSc patients with higher accuracy, rendering it more clinically relevant for SSC patients, that is likely of additional clinical value.

Our work may also provide a general solution to overcome the high GPU memory requirement in deep learning. By segmenting the objects of interest and then converting the results to point cloud or graph data type, higher prediction accuracy may be achieved using PNN, GNN or their combination.

There are some limitations to our research. The first limitation is that, in order to use automatic hyper-parameter search tool (Optuna), we did not apply data augmentation for GNN-Vessel. This may limit its performance. In the future, we will explore the efficient data augmentation methods for GNN-Vessel. Another limitation is that this paper explored the influence of three modalities (CT, point cloud and graph). If more image modalities are available in the future, we can explore the potential scalability of our methods on other modalities.

6. Conclusion

In this paper, we skeletonized pulmonary vessels and proposed PNN-Vessel and GNN-Vessel to estimate PFTs from point cloud and graph-based dataset, respectively. Compared with CNN-Vessel, which was developed based on 3D grid images, PNN-Vessel achieved significantly better performance with shorter training time and GNN-Vessel achieved slightly better performance with substantially shorter training time. The proper combination of the three networks (multiple variable step-wise regression analysis) verified that more detailed vessel information could provide more explanation of PFT estimation. We conclude that the detailed geometry of the vessels aids in the estimation of PFTs.

CRediT authorship contribution statement

Jingnan Jia: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Bo Yu:** Writing – review & editing, Writing – original draft, Methodology. **Prerak Mody:** Writing – review & editing, Writing – original draft, Methodology. **Maarten K. Ninaber:** Writing – review & editing, Validation, Data curation. **Anne A. Schouffoer:** Writing – review & editing, Validation, Data curation. **Jeska K. de Vries-Bouwstra:** Writing – review & editing, Validation, Data curation. **Lucia J.M. Kroft:** Writing – review & editing, Validation, Data curation. **Marius Staring:** Writing – review & editing, Validation, Supervision, Methodology. **Berend C. Stoel:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jingnan Jia reports financial support was provided by China Scholarship Council.

Data availability

All code and data used to develop and verify the deep neural networks in this study has been published at https://github.com/Jingnan-Jia/PFT_regression. The other materials used in the analysis can be available upon request for the purposes of reproducing or extending the analysis via the corresponding author, in accordance with local and institutional guidance and legal requirements.

Acknowledgments

This work is supported by the China Scholarship Council, China No. 202007720110.

References

[1] Jürgen Behr, Daniel E. Furst, Pulmonary function tests, Rheumatology 47 (suppl_5) (2008) v65–v67.

[2] Melissa Caron, Sabrina Hoa, Marie Hudson, Kevin Schwartzman, Russell Steele, Pulmonary function tests as outcomes for systemic sclerosis interstitial lung disease, Eur. Respir. Rev. 27 (148) (2018).

[3] Maarten K. Ninaber, Jan Stolk, Jasper Smit, Ernest J. Le Roy, Lucia J.M. Kroft, M. Els Bakker, Jeska K. de Vries Bouwstra, Anne A. Schouffoer, Marius Staring, Berend C. Stoel, Lung structure and function relation in systemic sclerosis: application of lung densitometry, Eur. J. Radiol. 84 (5) (2015) 975–979.

[4] Young Sang Choi, Jieun Oh, Seonhui Ahn, Yul Hwangbo, Jin-Ho Choi, Automated pulmonary function measurements from preoperative CT scans with deep learning, in: 2022 IEEE-EMBS International Conference on Biomedical and Health Informatics, BHI, IEEE, 2022, pp. 01–04.

[5] Aisling McGowan, Pierantonio Laveneziana, Sam Bayat, Nicole Beydon, P.W. Boros, Felip Burgos, Matjaž Fležar, Monika Franczuk, Maria Alejandra Galarza, Adrian H. Kendrick, Enrico Lombardi, Jellien Makonga-Braaksma, Meredith C. McCormack, Laurent Plantier, Sanja Stanojevic, Irene Steenbruggen, Bruce Thompson, Allan L. Coates, Jack Wanger, Donald W. Cockcroft, Bruce Culver, Karl Sylvester, Frans De Jongh, International consensus on lung function testing during the COVID-19 pandemic and beyond, ERJ Open Res. 8 (1) (2022).

[6] Brendan G. Cooper, An update on contraindications for lung function testing, Thorax 66 (8) (2011) 714–723.

[7] Hongjia Meng, Yun Liu, Xiaoyin Xu, Yuting Liao, Hengrui Liang, Huai Chen, A machine learning approach for preoperatively assessing pulmonary function with computed tomography in patients with lung cancer, Quant. Imaging Med. Surg. 13 (3) (2023) 1510.

[8] Jingnan Jia, Emiel R. Marges, Jeska K. De Vries-Bouwstra, Maarten K. Ninaber, Lucia J.M. Kroft, Anne A. Schouffoer, Marius Staring, Berend C. Stoel, Automatic pulmonary function estimation from chest CT scans using deep regression neural networks: The relation between structure and function in systemic sclerosis, IEEE Access 11 (2023) 135272–135282.

[9] Jingnan Jia, Marius Staring, Irene Hernández-Girón, Lucia J.M. Kroft, Anne A. Schouffoer, Berend C. Stoel, Prediction of lung CT scores of systemic sclerosis by cascaded regression neural networks, in: Medical Imaging 2022: Computer-Aided Diagnosis, Vol. 12033, SPIE, 2022, pp. 851–857.

[10] Nicole S.L. Goh, Sujal R. Desai, Srihari Veeraraghavan, David M. Hansell, Susan J. Copley, Toby M. Maher, Tamera J. Corte, Clare R. Sander, Jonathan Ratoff, Anand Devaraj, Gracijela Bozovic, Christopher P. Denton, Carol M. Black, Roland M. Du Bois, Athol U. Wells, Interstitial lung disease in systemic sclerosis: a simple staging system, Am. J. Respir. Crit. Care Med. 177 (11) (2008) 57–59.

[11] Zhiwei Zhai, Marius Staring, Maarten K. Ninaber, Jeska K. de Vries-Bouwstra, Anne A. Schouffoer, Lucia J. Kroft, Jan Stolk, Berend C. Stoel, Pulmonary vascular morphology associated with gas exchange in systemic sclerosis without lung fibrosis, J. Thorac. Imaging 34 (6) (2019) 373–379.

[12] Margreet E. Bakker, Maarten K. Ninaber, Jan Stolk, Lucia J.M. Kroft, Anne A. Schouffoer, Jeska K. De Vries Bouwstra, Suzanne E. Van Wijngaarden, Berend C. Stoel, Lung Density and Pulmonary Artery Diameter are Predictors of Pulmonary Hypertension in Systemic Sclerosis, J. Thorac. Imaging 32 (6) (2017) 391–397.

[13] Hyunjung Park, Jihye Yun, Sang Min Lee, Hye Jeon Hwang, Joon Beom Seo, Young Ju Jung, Jeongeun Hwang, Se Hee Lee, Sei Won Lee, Namkug Kim, Deep learning–based approach to predict pulmonary function at chest CT, Radiology 307 (2) (2023) e221488.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[15] Sima Siami-Namini, Neda Tavakoli, Akbar Siami Namin, The Performance of LSTM and BiLSTM in Forecasting Time Series, in: Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 3285–3292.

[16] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, Mohammed Bennamoun, Deep learning for 3D point clouds: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2021) 4338–4364.

[17] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, Matthias Nießner, Scannet: Richly-annotated 3d reconstructions of indoor scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5828–5839.

[18] Charles R. Qi, Hao Su, Matthias Niebner, Angela Dai, Mengyuan Yan, Leonidas J. Guibas, Volumetric and Multi-View CNNs for Object Classification on 3D Data, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016-December, IEEE Computer Society, 2016, pp. 5648–5656.

[19] Charles Ruizhongtai Qi, Li Yi, Hao Su, Leonidas J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. 30 (2017).

[20] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, Baoquan Chen, Pointcnn: Convolution on x-transformed points, Adv. Neural Inf. Process. Syst. 31 (2018).

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[22] Han Hu, Zheng Zhang, Zhenda Xie, Stephen Lin, Local relation networks for image recognition, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2019-October, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 3463–3472.

[23] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, Vladlen Koltun, Point Transformer, in: Proceedings of the IEEE International Conference on Computer Vision, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 16239–16248.

[24] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, Bernard Ghanem, Pointnext: Revisiting pointnet++ with improved training and scaling strategies, Adv. Neural Inf. Process. Syst. 35 (2022) 23192–23204.

[25] Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, Xia Hu, Towards deeper graph neural networks with differentiable group normalization, Adv. Neural Inf. Process. Syst. 2020-December (2020).

[26] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza, Christopher Fifty, Tao Yu, Kilian Q. Weinberger, Simplifying Graph Convolutional Networks, in: 36th International Conference on Machine Learning, Vol. 2019-June, ICML 2019, International Machine Learning Society (IMLS), 2019, pp. 11884–11894.

[27] Thomas N. Kipf, Max Welling, Semi-Supervised Classification with Graph Convolutional Networks, in: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, International Conference on Learning Representations, ICLR, 2016.

[28] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, Martin Grohe, Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks, in: 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, AAAI Press, 2018, pp. 4602–4609.

[29] Xuanang Xu, Fugen Zhou, Bo Liu, Dongshan Fu, Xiangzhi Bai, Efficient multiple organ localization in CT image using 3D region proposal network, IEEE Trans. Med. Imaging 38 (8) (2019) 1885–1898.

[30] Michaël Defferrard, Xavier Bresson, Pierre Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, Adv. Neural Inf. Process. Syst. (2016) 3844–3852.

[31] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, Karsten M Borgwardt, Weisfeiler-lehman graph kernels, J. Mach. Learn. Res. 12 (9) (2011) 2539–2561.

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, Graph attention networks, 2017, arXiv preprint arXiv:1710.10903.

[33] Martin R. Miller, J.A.T.S. Hankinson, Vito Brusasco, F. Burgos, R. Casaburi, A. Coates, R. Crapo, Pvd Enright, C.P.M. Van Der Grinten, P. Gustafsson, et al., Standardisation of spirometry, Eur. Respir. J. 26 (2) (2005) 319–338.

[34] Brian L. Graham, Vito Brusasco, Felip Burgos, Brendan G. Cooper, Robert Jensen, Adrian Kendrick, Neil R. MacIntyre, Bruce R. Thompson, Jack Wanger, 2017 ERS/ATS standards for single-breath carbon monoxide uptake in the lung, Eur. Respir. J. 49 (1) (2017).

[35] Zhiwei Zhai, Marius Staring, Berend C. Stoel, Lung vessel segmentation in CT images using graph-cuts, in: Medical Imaging 2016: Image Processing, Vol. 9784, SPIE, 2016, pp. 699–706.

[36] Zhiwei Zhai, Marius Staring, Irene Hernández Girón, Wouter J.H. Veldkamp, Lucia J. Kroft, Maarten K. Ninaber, Berend C. Stoel, Automatic quantitative analysis of pulmonary vascular morphology in CT images, Med. Phys. 46 (9) (2019) 3985–3997.

[37] Patrick Schober, Christa Boer, Lothar A. Schwarte, Correlation coefficients: appropriate use and interpretation, Anesth. Analg. 126 (5) (2018) 1763–1768.

[38] Raphael Vallat, Pingouin: statistics in Python, J. Open Source Softw. 3 (31) (2018) 1026.

[39] Terry K. Koo, Mae Y. Li, A guideline of selecting and reporting intraclass correlation coefficients for reliability research, J. Chiropr. Med. 15 (2) (2016) 155–163.

[40] Marion Dörrich, Mingcheng Fan, Andreas M. Kist, Impact of mixed precision techniques on training and inference efficiency of deep neural networks, IEEE Access (2023).

[41] Jonathon Byrd, Zachary Lipton, What is the effect of importance weighting in deep learning? in: International Conference on Machine Learning, PMLR, 2019, pp. 872–881.

[42] Ilya Loshchilov, Frank Hutter, Sgdr: Stochastic gradient descent with warm restarts, 2016, arXiv preprint arXiv:1608.03983.

[43] Ilya Loshchilov, Frank Hutter, Decoupled weight decay regularization, 2017, arXiv preprint arXiv:1711.05101.

[44] James Bergstra, Rémi Bardenet, Yoshua Bengio, Balázs Kégl, Algorithms for hyper-parameter optimization, in: Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011, 2011, pp. 1–9.

[45] Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky, Instance normalization: The missing ingredient for fast stylization, 2016, arXiv preprint arXiv:1607.08022.

[46] Sergey Ioffe, Christian Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in: 32nd International Conference on Machine Learning, Vol. 1, ICML 2015, International Machine Learning Society (IMLS), 2015, pp. 448–456.

[47] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton, Layer normalization, 2016, arXiv preprint arXiv:1607.06450.

[48] Jason C. Cai, Zeynettin Akkus, Kenneth A. Philbrick, Arunnit Boonrod, Safa Hoodeshenas, Alexander D. Weston, Pouria Rouzrokh, Gian Marco Conte, Atefeh Zeinoddini, David C. Vogelsang, Qiao Huang, Bradley J. Erickson, Fully Automated Segmentation of Head CT Neuroanatomy Using Deep Learning, Radiology: Artif. Intell. 2 (5) (2020) e190183.

[49] Noémie Le Gouellec, Alain Duhamel, Thierry Perez, Anne-Lise Hachulla, Vincent Sobanski, Jean-Baptiste Faivre, Sandrine Morell-Dubois, Marc Lambert, Pierre-Yves Hatron, Eric Hachulla, et al., Predictors of lung function test severity and outcome in systemic sclerosis-associated interstitial lung disease, PLoS One 12 (8) (2017) e0181692.

[50] Shingo Iwano, Tohru Okada, Hiroko Satake, Shinji Naganawa, 3D-CT Volumetry of the Lung Using Multidetector Row CT: Comparison with Pulmonary Function Tests, Academic Radiol. 16 (3) (2009) 250–256.

[51] Veronika Cheplygina, Cats or CAT scans: Transfer learning from natural or medical image source data sets? Curr. Opin. Biomed. Eng. 9 (2019) 21–27.

[52] Alejandro Talaminos Barroso, Eduardo Márquez Martín, Laura María Roa Romero, Francisco Ortega Ruiz, Factors affecting lung function: a review of the literature, Arch. Bronconeumología (Engl. Ed.) 54 (6) (2018) 327–332.