A Survey on Graph Neural Networks and Graph Transformers in Computer Vision: A Task-Oriented Perspective

Chaoqi Chen, Yushuang Wu, Qiyuan Dai, Hong-Yu Zhou, Mutian Xu, Sibei Yang, Xiaoguang Han, and Yizhou Yu, *Fellow, IEEE*

Abstract—Graph Neural Networks (GNNs) have gained momentum in graph representation learning and boosted the state of the art in a variety of areas, such as data mining (*e.g.,* social network analysis and recommender systems), computer vision (*e.g.,* object detection and point cloud learning), and natural language processing (*e.g.,* relation extraction and sequence learning), to name a few. With the emergence of Transformers in natural language processing and computer vision, graph Transformers embed a graph structure into the Transformer architecture to overcome the limitations of local neighborhood aggregation while avoiding strict structural inductive biases. In this paper, we present a comprehensive review of GNNs and graph Transformers in computer vision from a task-oriented perspective. Specifically, we divide their applications in computer vision into five categories according to the modality of input data, *i.e.,* 2D natural images, videos, 3D data, vision + language, and medical images. In each category, we further divide the applications according to a set of vision tasks. Such a task-oriented taxonomy allows us to examine how each task is tackled by different GNN-based approaches and how well these approaches perform. Based on the necessary preliminaries, we provide the definitions and challenges of the tasks, in-depth coverage of the representative approaches, as well as discussions regarding insights, limitations, and future directions.

Index Terms—Graph neural networks, graph Transformers, computer vision, vision and language, point clouds and meshes, medical image analysis.

1 Introduction

Deep learning [1] has brought many breakthroughs to computer vision, where convolutional neural networks (CNN) take a dominant position and become the fundamental infrastructure of many modern vision systems. In particular, a number of state-of-the-art CNN models, such as AlexNet [2] and ResNet [3], have been proposed and achieved unprecedented advances in a variety of vision problems, including image classification, object detection, semantic segmentation, and image processing to name a few. Moreover, existing vision systems take various input modalities as humans do, such as 2D images (*e.g.,* natural and medical images), videos, 3D data (*e.g.,* point clouds and meshes), as well as multimodal inputs (*e.g.,* image + text).

Despite the proliferation of CNN-based methods that excel at dealing with input data defined on regular grids, such as images, there is an emerging sense in the computer vision community that visual information with an irregular topology is crucial for representation learning but is yet to be thoroughly studied. Upon observing that human capacity for combinatorial generalization largely relies on their cognitive mechanisms for representing structures and reasoning about relations [4], mimicking human learning and decision-making processes could improve the performance of vision models. For instance, in the task of object recognition, state-of-the-art neural networks prefer to focus on perceiving separate objects, whereas the dependencies and interactions among different objects have received scant attention.

Moreover, compared to natural graph data, such as social networks and biological protein-protein networks, that has intrinsic edge connections and the notion of nodes, there is a shortage of principled methods for the construction of graphs (*e.g.,* relation graphs) from regular grid data (*e.g.,* images and temporal signals), and domain knowledge is critical for the success. On the other hand, certain visual data formats, such as point clouds and meshes, are naturally not defined on a Cartesian grid and involve sophisticated relational information. In that sense, both regular and irregular visual data formats would benefit from the exploration of topological structures and relations especially for challenging scenarios, such as understanding complex scenes, learning from limited experiences, and transferring knowledge across domains.

In the past few years, GNNs [5] have demonstrated ground-breaking performance in modeling graph structures under the umbrella of recent advancements in deep learning. In the scope of computer vision, much of current GNN-related research has one of the following two objectives: (1) a mixture of GNN and CNN backbones, and (2) a pure GNN architecture for representation learning. The former typically seeks to improve the long-range modeling ability

- *C. Chen, H.-Y. Zhou, and Y. Yu are with the Department of Computer Science, The University of Hong Kong, Hong Kong. (email: cqchen1994@gmail.com, whuzhouhongyu@gmail.com, yizhouy@acm.org).*
- *Y. Wu, M. Xu, and X. Han are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen. Y. Wu and X. Han are also with the Future Network of Intelligence Institute, CUHK-Shenzhen. (email: yushuangwu@link.cuhk.edu.cn, mutianxu@link.cuhk.edu.cn, hanxiaoguang@cuhk.edu.cn).*
- *Q. Dai and S. Yang are with the School of Information Science and Technology, ShanghaiTech University, Shanghai. S. Yang is also with Shanghai Engineering Research Center of Intelligent Vision and Imaging. (email: daiqy2022@shanghaitech.edu.cn, yangsb@shanghaitech.edu.cn).*
- *C. Chen, Y. Wu, Q. Dai, and H.-Y. Zhou contributed equally to this work. Corresponding authors: S. Yang, X. Han, and Y. Yu.*

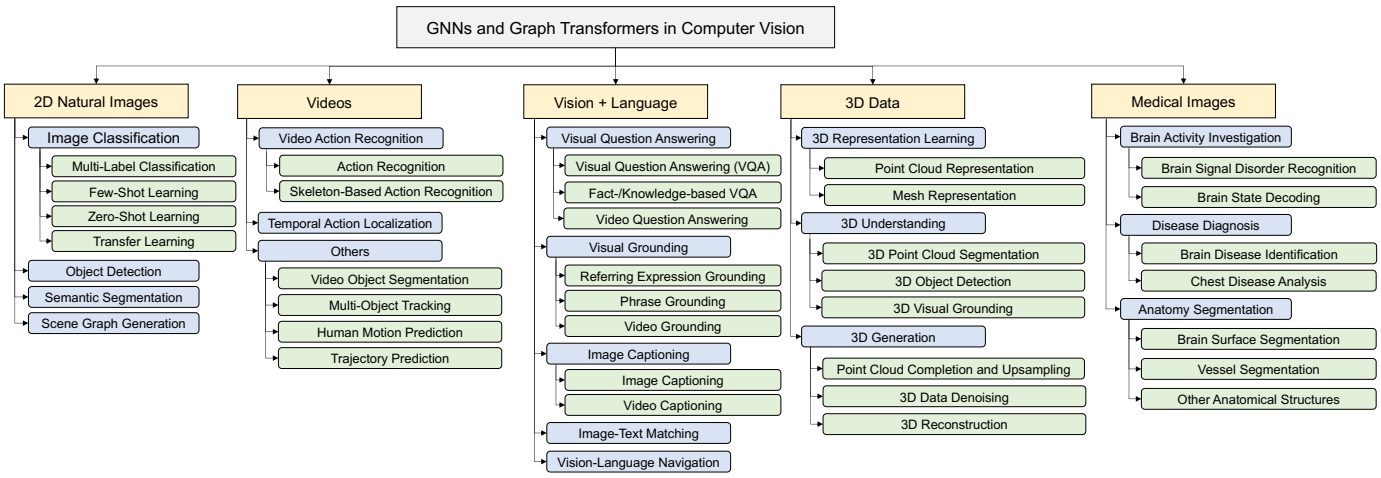

Fig. 1: Overview of the landscape of GNNs and graph Transformers in computer vision.

of CNN-based features and applies to vision tasks that were previously solved using pure CNN architectures, such as image classification and semantic segmentation. The latter serves as a feature extractor for certain visual data formats, such as point clouds, and was developed in parallel to other approaches. For instance, for the classification of 3D shapes represented as point clouds [6], there exist three major approaches, namely, pointwise MLP methods, convolution-based methods, and graph-based methods.

Despite the fruitful advancements, there still does not exist a survey to systematically and timely review how GNN-based computer vision has progressed. Specifically, we present this literature review as a complete introduction of GNNs in computer vision from a task-oriented perspective, including *(i)* the definitions and challenges of the tasks, *(ii)* in-depth coverage of the representative approaches, and *(iii)* systematic discussions regarding insights and future directions. In particular, we divide the applications of GNNs in computer vision into five categories according to the modality of input data. In each category, we further divide the applications according to the computer vision tasks they perform. We also review graph Transformers used in vision tasks considering their similarity with GNNs in terms of architecture [7], [8]. The organization of this survey is shown in Fig. 1. While several surveys (*e.g.*, [9], [10]) have previously reviewed the application of GNNs in certain vision tasks, we provide a more comprehensive and detailed examination of GNNs and graph Transformers in vision, a better taxonomy of the literature, and present discussions regarding insights, limitations, and potential directions for future research.

2 Background and Categorization

In this section, we recap GNNs and graph Transformers used in computer vision. Readers could refer to several previous GNN surveys [11], [12], [13] that comprehensively introduce the development of GNNs. In addition, we would like to emphasize that many existing GNN-based vision approaches actually use a mixture of CNNs and GNNs, whereas we focus on the GNN side.

2.1 Recurrent GNNs

GNN was initially developed in the form of recurrent GNNs. Earlier work [5] in this regime tries to extract node representations from directed acyclic graphs by recurrently using the same set of weights over iterations. Scarselli *et al.* [14] extended such neural networks to process more types of graphs, such as cyclic and undirected graphs. They recurrently update the hidden state $\mathbf{h}$ of a node as follows,

$$\mathbf{h}_i^{(t+1)} = \sum_{v_j \in \mathcal{N}(v_i)} f\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_{ij}^{\mathbf{e}}, \mathbf{h}_j^{(t)}\right), \tag{1}$$

where $\mathcal{N}(v_i)$ represents the neighborhood of node v_i, $f(\cdot)$ is a feedforward neural network, $\mathbf{x}_i \in \mathbb{R}^d$ denotes the features at v_i, $\mathbf{x}_{ij}^{\mathbf{e}} \in \mathbb{R}^c$ denotes the features at the edge between v_i and v_j, and t is the iteration number.

2.2 Convolutional GNNs

Inspired by the astonishing progress of CNNs in the deep learning era, many research efforts have been devoted to generalizing convolution to the graph domain. Among them, there are two series of approaches (cf. Fig. 2) that garner the most attention in recent years, namely, the spectral approaches [15], [16], [17], [18], [19], [20] and the spatial approaches [21], [22], [23], [24], [25], [26].

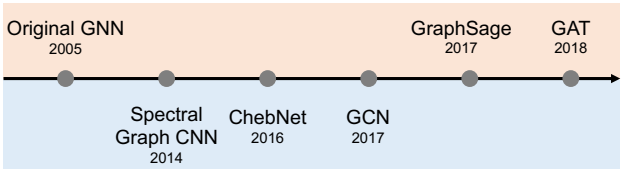

Fig. 2: Two types of graph convolutional operations.

2.2.1 Spectral Approaches

Spectral approaches rely on the Laplacian spectrum to define graph convolution. For an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, $\mathbf{A}$ is the adjacency matrix, and $\mathbf{D}$ is the diagonal degree matrix, $\mathbf{D}_{ii} = \sum_j^N \mathbf{A}_{ij}$. $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ represents the normalized Laplacian matrix of $\mathcal{G}$, and $\mathbf{L}$ can be decomposed as $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is the matrix of eigenvectors and $\mathbf{\Lambda} = diag[\lambda_1, ..., \lambda_N]$ is the diagonal matrix of eigenvalues.

Let $\mathbf{Z} \in \mathbb{R}^{N \times d}$ ($N = |\mathcal{V}|$) be the feature matrix of $\mathcal{G}$, and $\mathbf{z} \in \mathbb{R}^N$ be one of the columns of $\mathbf{Z}$ ($d = 1$). The graph Fourier transform of $\mathbf{z}$ is formulated as $\mathcal{F}(\mathbf{z}) = \mathbf{U}^T\mathbf{z}$, and the inverse graph Fourier transform is $\mathcal{F}^{-1}(\hat{\mathbf{z}}) = \mathbf{U}\hat{\mathbf{z}}$, where $\hat{\mathbf{z}} = \mathcal{F}(\mathbf{z})$. Then, the convolution of $\mathbf{z}$ with a filter $\mathbf{g} \in \mathbb{R}^N$ is defined as $\mathbf{z} *_\mathcal{G} \mathbf{g} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{z}) \odot \mathcal{F}(\mathbf{g})) = \mathbf{U}((\mathbf{U}^T\mathbf{z}) \odot (\mathbf{U}^T\mathbf{g}))$, where $*$ is the graph convolution operator and $\odot$ is the Hadamard product. By defining $\mathbf{g}_\theta = diag(\mathbf{U}^T\mathbf{g})$, which is a function of $\mathbf{\Lambda}$, we have

$$\mathbf{z} *_\mathcal{G} \mathbf{g} = \mathbf{U} diag(\mathbf{U}^T\mathbf{g})\mathbf{U}^T\mathbf{z} = \mathbf{U}\mathbf{g}_\theta\mathbf{U}^T\mathbf{z}. \quad (2)$$

Chebyshev Spectral CNN (ChebNet) [17] uses Chebyshev polynomials to approximate the filtering operation $\mathbf{g}_\theta$. $\mathbf{g}_\theta \approx \sum_{i=0}^K \theta_i T_k(\tilde{\mathbf{L}})$, where $\tilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{max} - \mathbf{I}$ is the scaled Laplacian matrix, λ_{max} is the largest eigenvalue of $\mathbf{L}$, and θ_i's are learnable parameters. The Chebyshev polynomials can be defined recursively by $T_i(\mathbf{z}) = 2\mathbf{z}T_{i-1}(\mathbf{z}) - T_{i-2}(\mathbf{z})$ with $T_0(\mathbf{z}) = 1$ and $T_1(\mathbf{z}) = \mathbf{z}$. Then, the filtering operation is formulated as

$$\mathbf{z} *_\mathcal{G} \mathbf{g} \approx \mathbf{U}\left(\sum_{i=0}^K \theta_i T_i(\tilde{\mathbf{L}})\right)\mathbf{U}^T\mathbf{z} \approx \sum_{i=0}^K \theta_i T_i(\tilde{\mathbf{L}})\mathbf{z}. \quad (3)$$

Graph Convolutional Networks (GCNs) [18] introduce a first-order approximation of ChebNet ($K = 1$). A GCN iteratively aggregates information from neighbors, and the feed forward propagation regarding node v_i is conducted as

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{v_j \in \mathcal{N}(v_i) \cup \{v_i\}} \hat{a}(v_i, v_j)\mathbf{W}^{(l)}\mathbf{h}_j^{(l)}\right), \quad (4)$$

where $\sigma(\cdot)$ is a nonlinear activation function, $\hat{\mathbf{A}} = (\hat{a}(v_i, u_j))$ denotes the re-normalized adjacency matrix $\mathbf{A}$, and $\mathbf{W}^{(l)}$ is a learnable transformation matrix in the l-th layer. GCNs can also be interpreted from a spatial view [13].

2.2.2 Spatial Approaches

GraphSAGE [23] is a general inductive framework that updates node states by sampling and aggregating hidden states from a fixed number of local neighbors. Formally, it performs graph convolutions in the spatial domain,

$$\begin{aligned} \mathbf{h}_{\mathcal{N}_s(v_i)}^{(l+1)} &= \texttt{Aggregator}_{l+1}\left(\{\mathbf{h}_j^l, \forall v_j \in \mathcal{N}_s(v_i)\}\right), \\ \mathbf{h}_i^{(l+1)} &= \sigma\left(\mathbf{W}^{(l+1)} \cdot [\mathbf{h}_{v_i}^l \oplus \mathbf{h}_{\mathcal{N}_s(v_i)}^{(l+1)}]\right), \end{aligned} \quad (5)$$

where $\mathcal{N}_s(v_i)$ is a subset of nodes sampled from the full neighborhood $\mathcal{N}(v_i)$, and $\oplus$ is the concatenation operator. As suggested by [23], the aggregation function $\texttt{Aggregator}_l$ can be the mean aggregator, max aggregator, LSTM aggregator, or pooling aggregator.

Graph Attention Networks (GAT) [24] introduces a self-attention mechanism to learn dynamic weights between connected nodes. It updates the hidden state of a node by attending to its neighbors,

$$\begin{aligned} \mathbf{h}_i^{(l+1)} &= \sigma\left(\sum_{v_j \in \mathcal{N}(v_i) \cup \{v_i\}} \alpha_{ij}\mathbf{W}^{(l)}\mathbf{h}_j^{(l)}\right), \\ \alpha_{ij} &= \frac{\exp\left(\texttt{LReLU}(\mathbf{a}_{(l)}^T[\mathbf{W}^{(l)}\mathbf{h}_i^{(l)} \oplus \mathbf{W}^{(l)}\mathbf{h}_j^{(l)}]\right)}{\sum_{v_k \in \mathcal{N}(v_i)} \exp\left(\texttt{LReLU}(\mathbf{a}_{(l)}^T[\mathbf{W}^{(l)}\mathbf{h}_i^{(l)} \oplus \mathbf{W}^{(l)}\mathbf{h}_k^{(l)}]\right)} \end{aligned} \quad (6)$$

where α_{ij} is the pairwise attention weight and $\mathbf{a}$ is a vector of learnable parameters. To increase the model's capacity and make the process of self-attention stable, GAT employs multi-head self-attention in practice.

2.2.3 New GNN Techniques

Deeper GNNs. A few recent works [27], [28], [29] delve into building deep GCNs for a variety of basic graph-oriented tasks, such as node prediction and link prediction. DeepGCNs [28] introduce commonly used concepts in CNNs, *i.e.*, residual connections, dense connections, and dilated convolutions, to make GCNs go deeper as CNNs, such as a 56-layer GCN for point cloud semantic segmentation. To alleviate the over-fitting and over-smoothing problems of deep GCNs, DropEdge [29] proposes to randomly drop out a certain proportion of edges of the input graph for each training iteration. Moreover, Li *et al.* [30] systemically investigate the effects of reversible connections, group convolutions, weight tying, and equilibrium models for improving the memory and parameter efficiency of GNNs, empirically revealing that combining reversible connections with deep network architectures could enable the training of extremely deep and wide GNNs, such as 1001 layers with 80 channels each and 448 layers with 224 channels each.

Graph Pooling. Graph pooling is a critical operation for modern GNN architectures. Inspired by the traditional CNN-based pooling, existing methods typically formulate graph pooling as a cluster assignment problem and explore the concept of local patches in the context of graphical structures. Defferrard *et al.* [17] achieve pooling with pre-defined subgraphs produced by a graph cut algorithm. DiffPool [31] is a differentiable graph pooling module capable of generating hierarchical graph representations, seamlessly integrating with various GNN architectures in an end-to-end manner. EigenPooling [32] incorporates node features and local structures to obtain better assignment matrices. Graph U-Nets *et al.* [33] presents a U-shaped architecture to implement pooling and up-sampling operations for GNNs.

Vision GNN. ViG [34] directly represents an image as a graph, aiming to learn graph-level features for downstream vision tasks. They first split the input image into a set of regularly-shaped patches and regards each patch as a graph node. Graph edges are constructed using K-nearest neighbors of each node. Then, multi-head graph convolution and positional encoding are performed at every node to jointly learn the topological structure and node features, and a feed-forward network (FFN) is used to mitigate the over-smoothing of node features and enhance the feature transformation capacity. In experiments, ViG outperforms DeiT by 1.7% (top-1 accuracy) on ImageNet classification and Swin-T by 0.3% (mAP) on MSCOCO object detection.

2.3 Graph Transformers for 3D Data

Point Transformer [35] designs a local vector self-attention mechanism for point cloud analysis. In contrast, a related work, Point Cloud Transformer [36], adopts global attention. The local vector self-attention operation in each Transformer block of the Point Transformer is defined as,

$$\mathbf{h}_i^{(l+1)} = \sum_{v_j \in \mathcal{N}_s(v_i)} \rho\left(\gamma\left(\mathbf{W}_Q^{(l+1)}\mathbf{h}_i^{(l)} - \mathbf{W}_K^{(l+1)}\mathbf{h}_j^{(l)} + \delta\right)\right) \odot \left(\mathbf{W}_V^{(l+1)}\mathbf{h}_j^{(l)} + \delta\right),$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are shared parameter matrices to compute the query, key and value for attention-based aggregation, $\odot$ is the element-wise product, δ is a position encoding function, γ is a nonlinear mapping function (*e.g.* MLP), and ρ is a normalization function (*e.g.* softmax). Recently, Fast Point Transformer [37] introduces a lightweight local self-attention architecture with voxel hashing to significantly improve efficiency. Stratified Transformer [38] samples distant points as additional keys to enlarge the receptive field, thereby modeling long-range dependencies. Point Transformer V2 [39] introduces grouped vector attention, improved position encoding, and partition-based pooling to enhance efficiency.

Mesh Graphormer [40] develops a graph Transformer for mesh reconstruction from images. It exploits graph convolution and self-attention to learn local interactions within neighborhoods and non-local relations, respectively. Each Graphormer encoder block consists of five components, *i.e.*, a Layer Norm, a Multi-Head Self-Attention (MHSA) module, a Graph Residual Block, a second Layer Norm, and an MLP at the end. Specifically, the MHSA module with P heads accepts an input sequence $\mathbf{H} = \{\mathbf{h}_i\} \in \mathbb{R}^{n\times d}, i \in \{1, 2, \cdots, n\}$ of n tokens, and outputs $\{\mathbf{h}_i^p\}$ for each token, where $p \in \{1, 2, \cdots, P\}$ is the head index. Each $\mathbf{h}_i^p$ is computed as

$$\mathbf{h}_i^p = \mathbf{Att}\left(\mathbf{Q}_i^p,\ \mathbf{K}^p\right) \cdot \mathbf{V}^p \quad \in \mathbb{R}^{\frac{d}{P}}, \tag{7}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are computed as $\mathbf{HW}_Q$, $\mathbf{HW}_K$, $\mathbf{HW}_V$, respectively, $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$ are trainable parameter matrices in each layer, $\mathbf{Att}(\cdot)$ computes the original self-attention in [41], and $\mathbf{Q}_i^p \in \mathbb{R}^{\frac{d}{P}}$, $\mathbf{K}^p, \mathbf{V}^p \in \mathbb{R}^{n\times\frac{d}{P}}$. Graph convolution (*i.e.* Eq. 4) with a residual connection is further applied to $\mathbf{H}$.

GNNs *v.s.* Vision Transformers (ViTs). Given that ViTs resemble GNNs, especially GAT, in accounting for relations among spatially distributed entities, ViTs can be perceived as a special case of GNNs. Technically, however, GNNs can use arbitrary relational inductive biases via a graph, *i.e.*, we possess the flexibility to design any form of connectivity (regular or irregular connectivity) and are capable of incorporating multiple types of relations concurrently. In comparison, ViTs focus on modeling relations in fully-connected global or local graphs. Therefore, if the data inherently exhibits a graph structure with irregular connectivity, such as scene graphs, GNNs are often a good choice.

3 2D Natural Images

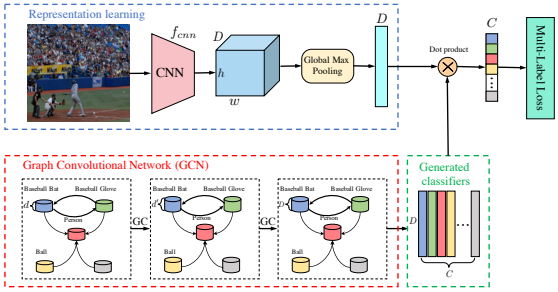

Fig. 3: ML-GCN (Figure used courtesy of [42]).

3.1 Image Classification

Thanks to the strong relational inductive biases provided by GNNs, *e.g.*, arbitrary connectivity patterns (one-to-many and many-to-many), recent approaches focus on modeling structural dependencies among different categories. This enables better visual and semantic understanding and improves transparency of the prediction process and the overall performance. Given a single or a set of image(s) I, we have

$$\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{X}) = \Pi(I) \tag{8}$$

where Π represents the graph construction process, and $\mathbf{X}$ is the feature matrix. Then, a common choice to define the adjacency matrix $\mathbf{A}$ based on $\mathcal{G}$ is Radial Basis Function,

$$\mathbf{A}_{ij} = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{2\delta^2}\right) \tag{9}$$

where $d(\cdot,\cdot)$ is a distance measure (*e.g.*, Euclidean distance and cosine similarity), δ is a scaling factor, and $\mathbf{x}$ can be either a class prototype or a specific instance. Note that different types of adjacency matrices, *e.g.*, language-based label dependencies and image structural priors, can be jointly considered. In addition to using pre-defined adjacency matrices, we can also opt for learnable ones by introducing additional parameters. These matrices will be merged into a unified matrix. After that, we can perform learning process,

$$\hat{\mathbf{X}} = \mathrm{GNN}(\mathcal{G}, \mathbf{A}) \tag{10}$$

where $\hat{\mathbf{X}}$ is the updated feature matrix which can be projected to the visual domain, combined with other feature extraction modules, or directly used for task-specific prediction.

TABLE 1: Performance (mAP) on two multi-label classification benchmarks, *i.e.*, MS-COCO and VOC 2007.

Method	Reference	MS-COCO	VOC 2007
ResNet-101 [3]	CVPR'16	77.3	89.9
ML-GCN [42]	CVPR'19	83.0	94.0
SSGRL [43]	ICCV'19	83.8	93.4
MS-CMA [47]	AAAI'20	83.8	-
ADD-GCN [46]	ECCV'20	85.2	93.6
TDRG [49]	ICCV'21	86.0	95.0

3.1.1 Multi-Label Classification

This task aims to recognize a set of objects within a single image. The key idea for GNN-based methods is to construct a per-image label graph for relational modeling and reasoning, ensuring that different categories are no longer isolated. Semantic object representations and label relationships can be learned simultaneously or independently. For the former, a representative work [42] (cf. Fig. 3) builds a directed graph over the label space, where each node is a word (label) embedding, and the connections denote their relations. GCN maps the constructed graph into a set of interdependent binary classifiers, and a label correlation matrix is designed to guide information propagation among nodes. For the latter, Chen *et al.* [43] introduced statistical label co-occurrence to directly guide the propagation of semantic features of different object regions. Follow-up approaches [44], [45], [46], [47], [48], [49] focus on modeling semantic label dependencies via more elaborate architectures, such as hypergraph neural networks [45], cross-modality attention [47], graph Transformer [48], and Transformer [49]. The performance of these algorithms is summarized in Tab. 1.

3.1.2 Few-Shot Learning (FSL)

FSL aims to generalize to new tasks that only have a few samples. GNN was introduced to compensate for the lack of semantic correlations. TPN [50] constructs graphs in the embedding space to exploit the manifold structure of novel classes. Label information is propagated from the support set to the query set based on the constructed graphs. Bottom-up and top-down reasoning modules [51] are introduced to explore the hierarchical relations of semantic classes. Compared to the node-based labeling frameworks, [52] proposes an edge-labeling GNN that learns to predict edge labels, explicitly constraining intra- and inter-class similarities. Instead of using a GNN to perform label propagation, Yu *et al.* [53] introduce an instance GNN and a prototype GNN as feature embedding task adaptation modules for quickly adapting learned features to new tasks. Moreover, meta-learning-based approaches [54], [55], [56] can directly solve data scarcity by imitating the distribution shift during training and performing class-based information transfer.

3.1.3 Zero-Shot Learning (ZSL)

ZSL aims to classify samples from classes that have not been seen during training, and predefined or learnable knowledge graphs are used to represent semantic relations. Due to the absence of training data, they resort to certain prior knowledge, such as language-based relation graphs, to assist in understanding information unseen during training. To enhance knowledge propagation across distant nodes in a graph network, [57] leverages the hierarchical structure of the graph, such as the relations between different levels of species classification. To promote structural knowledge transfer (*e.g.* semantic descriptions of classes) from seen to unseen classes, Xie *et al.* [58] introduce a region-based graph to model the visual relations among different regions within an input image, which are expected to generalize well on unseen classes. Instead of using pre-defined attributes to bridge seen and unseen classes, [59] resort to the inter-class relations to generate attribute vectors and propagate the dependencies among them via graph convolution. Recent studies strive to improve the efficacy of knowledge propagation across different classes, such as joint visual and semantic prototype propagation on auto-generated graphs [60], GATs for exploiting appearance relations among local regions [61], as well as explicit compositional relation modeling [62].

3.1.4 Transfer Learning

Domain Adaptation (DA) and Domain Generalization (DG), both sub-branches of transfer learning, benefit from knowledge graphs depicting relations among classes. Such structural knowledge is expected to be transferable across domains and thus can be reused for novel environments. Wang *et al.* [63] propose an adversarial GCN for DA, where a GCN is established over densely connected instance graphs using mini-batch samples to encode data structure information. For DG, Chen *et al.* [64] build global prototypical relation graphs and introduce a graph self-attention mechanism to model long-range dependencies among different categories. Other works introduce more elaborate training algorithms to aggregate and propagate information in both intra- and inter-domains, such as curriculum learning to achieve progressive aggregation [65], progressive graph learning to select reliable pseudo-labels [66], prototype alignment to assist the structural knowledge transfer between domains [67], [68].

Discussion. GNNs used for image classification have been mostly explored for multi-object, data- or label-efficient scenarios (*e.g.*, zero-shot and few-shot learning), aiming to model the complex relations among different objects/classes for compensating the shortage of training samples or supervision signals. Current work focuses on extracting ad-hoc knowledge graphs from the data for a certain task, which is heuristic and relies on the human prior. Future work is expected to *(i)* develop general and automatic graph construction procedures, *(ii)* enhance the interactions between abstract graph structures and task-specific classifiers, and *(iii)* excavate more fine-grained building blocks (node and edge) to increase the capability of constructed graphs.

3.2 Object Detection

Object detection [69] aims to localize and recognize all object instances of given classes in input images. Despite the fruitful progress, modern object detectors often overlook the relationships and interactions among object instances by treating each object individually. Thus, two challenges arise: (1) reasoning over semantic dependencies, co-occurrence, and relative locations of objects in addition to perceiving individual objects, and (2) embedding object-level dependencies into the detection pipeline when data distributions exhibit structural object relations. In general, given an input image I with a set of objects $\mathcal{O}$, the goal of GNN-based methods is to perform both local and global relational reasoning over these objects or parts of objects,

$$\mathcal{G} = \Pi(I, \mathcal{O}),\ \hat{\mathbf{X}} = \mathrm{GNN}(\mathcal{G}, \mathbf{A}). \tag{11}$$

Reasoning-RCNN [70] presents an adaptive global reasoning network for large-scale object detection by incorporating commonsense knowledge (category-wise knowledge graph) and propagating visual information globally, thereby endowing detection networks with the capability of reasoning. SGRN [71] (cf. Fig. 4) goes one step further by adaptively discovering semantic and spatial relationships without requiring prior handcrafted linguistic knowledge. The relation learner learns a sparse adjacency matrix to encode contextual relations among different regions. A spatial graph reasoning module utilizes a learned adjacency matrix and learnable spatial Gaussian kernels to perform feature aggregation and relational reasoning. RelationNet [72] proposes an adapted attention module for detection head networks, explicitly learning relations among objects through encoding long-range dependencies. RelationNet++ [73] presents a self-attention based decoder module to embrace the strengths of different object/part representations within a single detection framework. Li *et al.* [74] introduce a heterogeneous graph to jointly model object-object and object-scene relations. Recently, GraphFPN [75] presents a graph feature pyramid network, which explores contextual and hierarchical structures of an input image using a superpixel hierarchy, enabling within-scale and cross-scale feature interactions with the help of spatial and channel attention mechanisms.

In addition to improving the in-distribution performance, domain adaptive object detection (DAOD) has received a great deal of attention from many real-world applications.

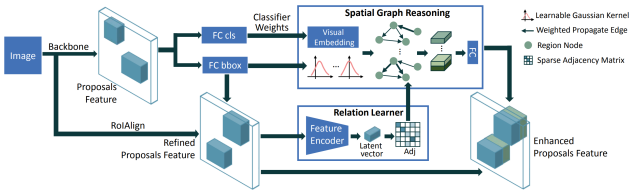

Fig. 4: SGRN (Figure used courtesy of [71]).

A natural choice for modeling cross-domain relations is a bipartite graph $\mathcal{G} = \{\mathcal{V}_s, \mathcal{V}_t, \mathcal{E}\}$, where vertices are divided into two disjoint sets and edges connect vertices from different sets. In practice, Chen *et al.* [76], [77] first builds intra- and inter-domain relation graphs in virtue of cyclic between-domain consistency without assuming any prior knowledge about the target distribution. It then incorporates bipartite GCNs and graph attention mechanisms to model homogeneous and heterogeneous object dependencies and interactions in both pixel and semantic spaces. In addition, SIGMA [78] formulates DAOD as a graph matching problem by setting up cross-image graphs to model class-conditional distributions on both domains. SRR-FSD [79] introduces a semantic relation reasoning module, where each class has a corresponding node in a dynamic relation graph, to integrate semantic relations between base and novel classes for novel object detection. By doing so, semantic information is propagated through graph nodes, endowing an object detector with semantic reasoning ability.

Discussion. Owing to the long-range modeling capability, GNN-based detection methods exploit between-object [71], cross-scale [75] or cross-domain [76] relationships, as well as relationships between base and novel classes [79]. Due to the introduction of additional learning modules, however, the optimization process becomes harder than vanilla detection models. Moreover, the compatibility between Euclidean and non-Euclidean structures will be challenged if not properly harmonized. In future, researchers could *(i)* design better region-to-node feature mapping methods, *(ii)* incorporate Transformer (or pure GNN) encoders to improve the expressive power of initial node features, and *(iii)* instead of resorting to forward and backward feature mappings, directly perform reasoning in the original feature space to better preserve the intrinsic structure of images.

3.3 Image Segmentation

Semantic segmentation divides an image into several semantically meaningful regions to perform pixel-wise labeling. While CNN-based architectures have made significant advancements, their intrinsic limitations, such as the inability to reason about distant regions of arbitrary shapes, pose challenges in achieving a holistic understanding of a scene. In this regard, GNNs offer a unified framework for modeling both object appearances $\mathcal{R}_a$ and image contexts $\mathcal{R}_c$.

$$\mathcal{G} = \Pi(I),\ \mathbf{A} = \mathtt{E}(\mathcal{R}_a, \mathcal{R}_c),\ \hat{I} = \mathtt{P}(\mathrm{GNN}(\mathcal{G}, \mathbf{A})), \quad (12)$$

where $\mathtt{E}(\cdot)$ is a function that encodes both $\mathcal{R}_a$ and $\mathcal{R}_c$, $\mathtt{P}(\cdot)$ is a pixel-wise predictor, and $\hat{I}$ represents prediction results.

A common goal is to globally model contextual and semantic relations in the backbone feature space. Zhang *et al.* [80] propose a dual GCN framework, where a coordinate space GCN models spatial relations among pixels, and a feature space GCN models dependencies among channel dimensions of a feature map. After that, these two features are mapped back to the original coordinate space. To improve local feature aggregation, Chen *et al.* [81] design a global reasoning unit by projecting features that are globally aggregated in the coordinate space to the node domain and performing relational reasoning in a fully connected graph. To model long-range dependencies and avoid constructing fully connected graphs, DGMN [82] dynamically samples the neighborhood of a node and predicts node dependencies, filter weights, and affinities for information propagation. Similarly, Yu *et al.* [83] dynamically sample representative nodes for relational modeling. Instead of constructing additional semantic interaction spaces (projection and re-projection), Li *et al.* [84] propose an improved Laplacian formulation that enables graph reasoning in the original feature space, fully exploiting contextual relations at different feature scales. Hu *et al.* [85] introduce a class-wise dynamic graph convolution module to perform graph reasoning over pixels that belong to the same class to dynamically aggregate features. We summarize the performance in Tab. 2.

TABLE 2: Performance (mIOU) on segmentation benchmarks, *i.e.*, Cityscapes, PASCAL-Context, and COCO Stuff.

Method	Reference	Cityscapes	PASCAL	COCO Stuff
DGCNet [80]	BMVC'19	80.5	53.7	-
GloRe [81]	CVPR'19	80.9	-	-
DGMN [82]	CVPR'20	81.6	-	-
SpyGR [84]	CVPR'20	81.6	52.8	39.9
RGNet [83]	ECCV'20	81.5	53.9	-
CDGCNet [85]	ECCV'20	82.0	-	40.7

For one-shot semantic segmentation, Zhang *et al.* [86] introduce a pyramid graph attention module to model the connection between query and support feature maps, associating unlabeled pixels in the query set with semantic and contextual information. For few-shot semantic segmentation, Xie *et al.* [87] propose a scale-aware GNN to perform cross-scale relational reasoning over both support and query images. A self-node collaboration mechanism is introduced to perceive different resolutions of the same object. Zhang *et al.* [88] propose an affinity attention GNN for weakly supervised semantic segmentation. Specifically, an image is first converted to a weighted graph via an affinity CNN network, and then an affinity attention layer is devised to model long-range interactions in the constructed graph and propagate semantic information to unlabeled pixels. In addition to semantic segmentation, Wu *et al.* [89] design a bidirectional graph reasoning network to bridge a pair of "things" and "stuff" branches for panoptic segmentation. They first build things and stuff graphs to represent relations within corresponding branches. Bidirectional graph reasoning is then performed to propagate semantic representations both within and across the two branches.

Discussion. Many research efforts in semantic segmentation are devoted to exploring contextual information in the local- or global-level with pyramid pooling, dilated convolutions, or the self-attention mechanism. For instance, non-local networks [90] and their variants achieve this goal by adopting the self-attention mechanism, but are computationally expensive as comparing every pixel to all other pixels in an image has a quadratic time complexity. In addition, probabilistic graphical models, such as Conditional Random Fields (CRFs) and Markov Random Fields (MRFs), can be

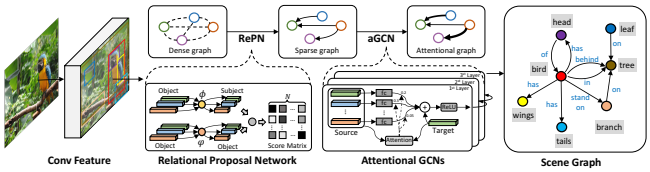

Fig. 5: Graph R-CNN (Figure used courtesy of [91]).

used to model scene-level semantic contexts but have very limited representation ability. In comparison to these prior efforts, GNN-based methods show clear superiority in terms of relation modeling and training efficiency.

3.4 Scene Graph Generation (SGG)

SGG [92], [93] refers to the task of detecting object pairs and their relations in an image to generate a visually-grounded scene graph, which provides a high-level understanding of a visual scene. A scene graph is the result of image parsing by representing objects in the image as nodes and their relations as edges. This aligns well with the nature of GNNs, which do not require prior knowledge to construct task-specific graph structures from inputs, unlike other computer vision tasks. GNN-based methods for SGG typically consist of three stages: object detection, relation graph construction and refinement, and relation prediction, *i.e.*,

$$P(\mathcal{S}|I) = P(\mathcal{V}|\mathcal{I})P(\mathcal{E}|\mathcal{V},I)P(\mathcal{R},\mathcal{O}|\mathcal{V},\mathcal{E},I), \quad (13)$$

where I is an image, $\mathcal{S} = (\mathcal{V},\mathcal{E},\mathcal{O},\mathcal{R})$ is a scene graph, and $\mathcal{O}$ and $\mathcal{R}$ are object and relationship labels respectively.

Their key idea is to align visual and textual entities (including their topological homogeneity) in a shared latent space. Graph R-CNN [91] (cf. Fig. 5) first obtains a sparse candidate graph by pruning the densely connected graph generated from RPN via a relation proposal network, then an attentional GCN is introduced to aggregate contextual information and update node features and edge relations. Li *et al.* [94] utilizes a spatially weighted message-passing structure to refine features of objects and subgroups by passing messages among them with attention-like schemes. Qi *et al.* [95] propose attentive relational networks, which first transform label embeddings and visual features into a shared semantic space, and then rely on a GAT to perform feature aggregation for final relation inference. Li *et al.* [96] use a bipartite GNN to estimate and propagate relation confidence in a multi-stage manner. Suhail *et al.* [97] propose an energy-based framework, which relies on a graph message passing algorithm to compute the energy of configurations.

Discussion. Due to the inherent connection between SGG and GNNs, many advanced graph representation paradigms can be effectively applied to SGG. However, current approaches in SGG often adopt a stage-wise manner, and exploring methods that can directly generate a scene graph in a single stage is an open question. Further research can focus on developing novel generative models, such as graph-based diffusion models, to better exploit the relational inductive bias in generating scene graphs.

4 Video Understanding

Researchers have explored utilizing GNNs in video understanding tasks, including video action recognition [98], [99], temporal action localization [100], [101], and video object segmentation [102], [103]. As spatio-temporal relations are critical in video understanding, GNNs are naturally applied to perform spatial and semantic relation reasoning over time among visual constituents. Specifically, GNNs are adapted in video understanding tasks to facilitate modeling (i) spatio-temporal relations of objects among single or multiple frames, ranging from first-order relations to higher-order relations [98], (ii) dependencies between frames at multiple time scales [104], (iii) dependencies among joints in the same or consecutive frames in skeleton-based approaches [99].

4.1 Video Action Recognition

Video human action recognition is one of the fundamental tasks in video processing and understanding, which aims to identify and classify human actions in RGB/depth videos or skeleton data. Regardless of the data modality, modeling spatio-temporal contexts using humans, objects and joints is critical in identifying human action. Typically, one or more graphs $\mathcal{G}$ are constructed from objects, humans, video frames or skeletons. Then, GNNs are employed to model the relationships among these elements and make predictions regarding human actions as follows:

$$\mathcal{G} = \Pi(\hat{\mathcal{V}},\hat{\mathcal{E}}),\ \mathcal{F} = \mathrm{GNN}(\mathcal{G}), \quad (14)$$

where $\hat{\mathcal{V}}$ denotes the nodes that can represent objects, humans, video frames, or skeletons, $\hat{\mathcal{E}}$ denotes the edges that can capture spatial relations between objects in a single frame or temporal relations among multiple frames, and $\mathcal{F}$ represents the aggregated information for final prediction. Some of these methods employ RNN-style structures for temporal modeling in this context.

Action Recognition. Wang *et al.* [98] propose to capture long-range temporal contexts via graph-based reasoning over human-object and object-object relations. As shown in Fig. 6, they connect all the humans and objects in all the frames to construct a space-time region graph, where edges based on appearance similarity connect every pair of objects while edges based on spatio-temporal relations connect spatially overlapping objects in consecutive frames. The results from the GCNs built on this graph are used to predict human action. Ou *et al.* [105] improve the spatio-temporal graph in [98] by constructing actor-centric object-level graphs and applying GCNs to capture the contexts among objects in an actor-centric way. In addition, a relation-level graph is built to model the contexts of relation nodes. Instead of spatio-temporal modeling, Zhang *et al.* [104] propose multi-scale reasoning over the temporal graph of a video, where each node is a video frame, and the pairwise relations between nodes are represented as a learnable adjacency matrix. To capture both short-term and long-term dependencies, both GAT and the temporal adjacency matrix have multiple heads, each of which investigates one kind of temporal relations.

Some other studies have explored alternative approaches, such as leveraging knowledge graphs for zero-shot action recognition [106] and employing graph-based higher-order relation modeling for long-term action recognition [107]. Besides human action recognition, GNNs have also been applied to group activity recognition [108] and action performance assessment [109].

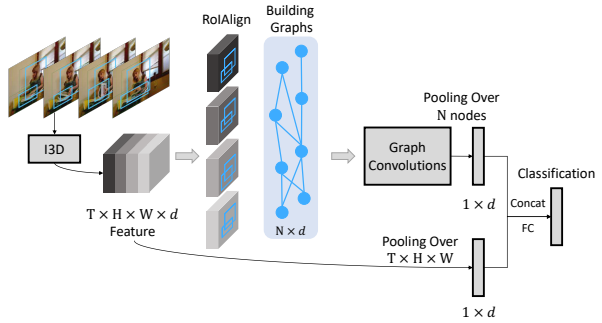

Fig. 6: Long-range temporal modeling via GCNs (Figure used courtesy of [98]).

Skeleton-Based Action Recognition. Skeletal data plays a vital role in action recognition because spatio-temporal dependencies among human body parts indicate human motion and action. Skeleton-based action recognition recognizes human actions in a skeleton sequence extracted from a video. Considering the human skeleton has a graph structure with natural joint connections, Yan *et al.* [99] propose an ST-GCN network that first connects joints in the same frame according to the natural connectivity in the human skeleton, and then connects the same joints in two consecutive frames to maintain temporal information. They run GCNs on the joint graph to learn both spatial and temporal action patterns. As shown in Tab. 3, their ST-GCN network improves on previous methods significantly. Shi *et al.* [110] improve [99] by introducing a fully-connected graph with learnable edge weights between joints and a data-dependent graph learned from the input skeleton. They build GCNs on all the three graphs. Similar to [110], Li *et al.* [111] and Li *et al.* [112] also connect physically-apart skeleton joints to capture the patterns of collaboratively moving joints. Different from [111], Zhao *et al.* [113] improve joint connectivity in a single frame [99] by adding edges between limbs and head. In addition, they use GCNs to capture relations between joints in individual frames and adopt LSTM [114] to capture temporal dynamics. Instead of only updating node features in GCNs, Shi *et al.* [115] maintain edge features and learn both node and edge feature representations via directed graph convolution.

Unlike previous works that model local spatial and temporal contexts in consecutive frames, Liu *et al.* [116] capture long-range spatio-temporal dependencies. They first construct multiple dilated windows over the temporal dimension. Then, in each window, separate GCNs are used on multiple graphs with different scales. Finally, the results of GCNs are aggregated on all the graphs in all the windows to capture multi-scale and long-range dependencies. Some additional studies [117], [118], [119] have explored combining or modifying standard graph convolution layers with other modules (*e.g.*, LSTM and shift CNNs [120]) to enhance their suitability for action recognition tasks.

4.2 Temporal Action Localization

Temporal action localization aims to localize temporal intervals and recognize action instances in an untrimmed video. Video contexts play an important role in action detection as they can be used to infer potential actions [101]. This section reviews methods that utilize GNNs to obtain video contexts. Zeng *et al.* [100] follow a two-stage pipeline for temporal action localization, which first generates temporal proposals, then classifies them and regresses their temporal boundaries. In the second stage, pairs of proposals are connected according to their temporal intersection over union and the L_1 distance over union to form a proposal graph. GCNs are run on the graph to capture the relations among proposals, and the output from the GCNs is used to predict the boundary and category of every proposal again. Unlike [100] taking temporal proposals as nodes, Xu *et al.* [101] represent a video sequence as a snippet graph, where each node is a snippet and each edge corresponds to the correlation between two snippets. Both temporal edges between snippets in two consecutive time steps and semantic edges learned from snippet features are used to build graph connectivity. A dynamic graph convolution operation [124] is performed to obtain temporal and semantic contexts for snippets. Zhao *et al.* [125] introduce a video self-stitching graph network to address the scaling curse in [101] by utilizing the graph network to exploit multi-level correlations among cross-scale snippets. Besides temporal action localization, GNNs are utilized in [126] and [127] for temporal action proposal generation and video action segmentation, respectively.

TABLE 3: Performance comparison on datasets for action recognition. GNN-based methods are marked with ⋆.

Method	Charades [121] mAP	NTU-RGB+D [122] Accuracy(%)		Kinetics Skeleton [123] Accuracy(%)	
		X-Sub	X-View	Top1	Top5
Action Recognition					
NL + I3D [90]	37.5	-	-	-	-
STRG [98]⋆	39.7	-	-	-	-
O^2R^2G [105]⋆	44.9	-	-	-	-
TS-GCN [106]⋆	40.2	-	-	-	-
GHRM [107]⋆	38.3	-	-	-	-
Skeleton-Based Action Recognition					
Deep LSTM [122]	-	60.7	67.3	16.4	35.3
ST-GCN [99]⋆	-	81.5	88.3	30.7	52.8
2s-AGCN [110]⋆	-	88.5	95.1	36.1	58.7
STGR [111]⋆	-	86.9	92.3	33.6	56.1
AS-GCN [112]⋆	-	86.8	94.2	34.8	56.5
Zhao et al. [113]⋆	-	81.8	89.0	-	-
DGNN [115]⋆	-	89.9	96.1	36.9	59.6
MS-G3D Net [116]⋆	-	91.5	96.2	38.0	60.9

4.3 Others

Video Object Segmentation Wang *et al.* [102] adopt GNNs to capture higher-order relations among video frames to help object segmentation from a global perspective. In addition, they adapt the convolution operation in classical GNNs to the pixel-level segmentation task. Lu *et al.* [103] extend [102] by introducing a graph memory network to store memory in a graph structure with memory cells as nodes. And the query of a video frame in the memory graph facilitates segmentation mask prediction of that frame.

Multi-Object Tracking aims to track all the objects in a video. Two-step tracking-by-detection methods set up detection graphs where object detection instances in all frames are taken as nodes and an edge connects two instances in consecutive frames. The trajectory of an object forms a connected component in such a graph. Braso *et al.* [128] propose a time-aware message passing network on the detection graph to encode its temporal structure and predict whether an edge is active or non-active to obtain the connected components. Weng *et al.* [129] suggest first connecting tracked objects and

detected objects in two consecutive frames according to their spatial distances. Then, GNNs [124] are used to aggregate features from neighbors and predict the status of edges to obtain the connectivity between the tracked objects and the detected ones.

Human motion prediction forecasts future poses conditioned on the past motions in videos. Like skeleton-based action recognition, spatio-temporal correlations among body parts are crucial cues for accurate motion prediction. Li *et al.* [130] propose dynamic multi-scale GNNs to capture physical constraints and movement relations among body components at multiple scales and achieve effective motion prediction. Instead of modeling spatial correlation, Sofianos *et al.* [131] encode spatial joint-joint, temporal time-time, and spatio-temporal joint-time interactions via the proposed separable space-time GNNs. In addition to human motion prediction, Cai *et al.* [132] exploit GNNs to estimate 3D human poses from consecutive 2D poses.

Trajectory Prediction aims to predict the future trajectory of a pedestrian on the basis of his/her existing trajectory and the complex interactions between the pedestrian and the environment/other pedestrians. Mohamed *et al.* [133] propose a graph representation of the pedestrian trajectory based on relative locations among pedestrians and their temporal information. Furthermore, GCNs are applied to capture the complex interactions and the spatio-temporal graph representation. Similar to [133], Yu *et al.* [134] propose a spatio-temporal graph Transformer to conduct crowd trajectory prediction according to relative locations among pedestrians and temporal dependencies among trajectories. In addition, they extend GATs by using Transformer's self-attention mechanism [41]. Beyond the distance between pedestrians, Sun *et al.* [135] utilize social-related annotations, historical trajectories, and human contexts to construct a social behavior graph with pedestrians' historical trajectories as nodes and social relations as edges. Then, they build a GCN on the graph to learn higher-order social relations to facilitate future trajectory generation. Shi *et al.* [136] use the same GCNs in [133] to capture spatial and temporal dependencies but they introduce two sparse directed spatial and temporal graph representations for trajectories. Different from previous methods, Li *et al.* [137] construct a multi-scale graph to model the spatial information and surrounding area of pedestrians in multiple scales. In addition, they use scene semantic segmentation to help model the relations between pedestrians and the scene.

4.4 Discussion

Compared to images, videos involve temporal connectivity patterns among consecutive frames. The spatio-temporal dependencies and contexts in a video play an essential role in video understanding. Therefore, since the introduction of GNNs for capturing the spatial and semantic relations among visual constituents over time, video understanding has achieved significant progress. However, existing methods usually either capture partial dependencies at the frame level or local contexts at the region level for several consecutive frames. A meaningful direction for future research would be effectively capturing long-range global dependencies or crucial contexts without redundant information.

5 Vision + Language

In addition to a single modality, there has been growing interest in applying GNNs to vision-and-language tasks, such as visual question answering [138], visual grounding [139], and image captioning [140]. GNNs facilitate modeling the structure of visual and linguistic components to help with cross-modal semantic alignment and joint understanding, which are mainly utilized to (i) learn the spatial or semantic relation between visual components, such as detected objects in images [141]; (ii) model the dependencies among noun phrases in the input questions or expressions [142]; (iii) learn the relation among both images and texts jointly by combining the above two methods or performing graph convolution on the language-conditioned visual graph [143].

5.1 Visual Question Answering

The input to visual question answering (VQA) is an image and a question expressed as text in a natural language, and the output is an answer to the question. Since questions usually describe not only the appearances and attributes of the visual constituents of the image but also their relations, capturing the relations and aligning them with the question plays a vital role in VQA. Compared with conventional CNN-based or LSTM-based methods, GNN-based methods are able to capture complex object relationships, adaptively focus on relevant image parts, and conduct higher-order reasoning about the visual content. This enables more context-sensitive and detailed answers. In VQA, graphs are constructed by leveraging the objects presented in image $\hat{I}$ and the textual content of question $\hat{Q}$. In the case of fact-based VQA, an additional graph is created to incorporate external knowledge $\hat{K}$. The general pipeline of GNN-based methods for the VQA task can be outlined as follows:

$$\mathcal{G} = \Pi(\hat{I}, \hat{Q}, \hat{K}),\ \mathcal{F} = \mathrm{P}(\mathrm{GNN}(\mathcal{G})), \tag{15}$$

where $\mathrm{P}(\cdot)$ refers to the final prediction layer, and $\mathcal{F}$ represents the final answer.

Static Graphs. A typical solution sets up the graph representations of images and questions, and further aligns the constructed graphs [141]. In the image graph, visual objects become nodes, and relative spatial relations between objects become edge weights; the question graph represents the dependencies among words via a dependency parser [144]. A recurrent unit [145] followed by an attention mechanism similar to cosine similarity with learnable transformations is used to process and align the two graphs for prediction. Instead of the language-irrelevant image graph construction, Norcliffe *et al.* [146] use a graph learner to generate a sparse graph representation conditioned on the question for the image, followed by a graph CNN [25] to capture language-relevant spatial relations between objects.

Dynamic Graphs. Instead of constructing static graphs, Hu *et al.* [147] generate a dynamic visual graph where the edges between objects are updated during each GNN-based message-passing iteration. The update is conditioned on the textual command vector extracted for each iteration (see Fig. 7). Jing *et al.* [148] adopt a similar architecture to answer multiple questions for compositional VQA. Li *et al.* [149] propose three graphs for representing semantic relations, spatial relations, and implicit relations between objects, respectively. A GAT is

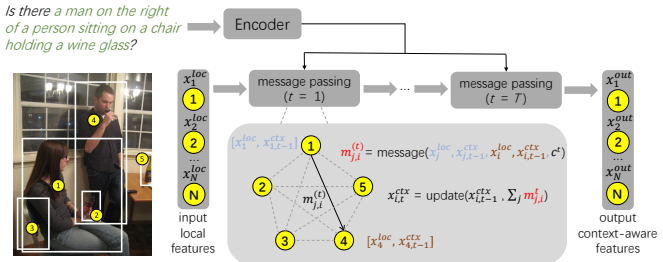

Fig. 7: Language-Conditioned Graph Networks [147].

TABLE 4: Accuracy on two datasets for visual question answering. GNN-based methods are marked with *.

Method	VQA v2.0 [154]		GQA [155]		
	std	dev	val	test-dev	test
Bottom-Up [159]	65.67	-	52.2	-	49.7
Graph Learner [146]*	66.18	-	-	-	-
single-hop + LCGN [147]*	-	-	63.9	55.8	56.1
Jing *et al.* [148]*	-	-	-	-	59.6
ReGAT [149]*	70.58	70.27	-	-	-
HANs [150]*	-	69.10	-	69.5	-

run on each graph to assign importance to nodes and learn relation-aware node representations. Results from the GATs are aggregated to predict the answer.

Symbolic Graphs. Compared with feature-based graphs, symbolic graphs provide a clear and interpretable representation through symbolic labels, and excel in reasoning tasks. Kim *et al.* [150] suggest using symbolic graphs generated through the scene graph parser [92] and dependency parser, respectively, to represent the image and question. Nodes of such symbolic graphs represent semantic units (*e.g.*, attributes and objects) in a textual form, and edges represent relations (*e.g.*, predicates) between semantic units. The message passing neural network [151] is applied to the two symbolic graphs to obtain informative representations and align their sub-graphs. Saqur *et al.* [152] further utilize the graph isomorphism network [153] to perform graph matching.

A comparison of these GNN-based methods on the VQA v2.0 [154] and the GQA [155] datasets are shown in Tab. 4. In addition, GNNs have been applied to TextVQA [156] and visual commonsense reasoning [157], [158], which extend the VQA task by reading texts in images and enabling commonsense reasoning, respectively.

Fact-/Knowledge-based VQA. Compared to classical VQA, fact-based VQA [160] has an external knowledge base of facts to facilitate fact-based reasoning related to the image and question. A supporting fact is required to predict the correct answer for an image-question pair. Narasimhan *et al.* [161] first parse and retrieve relevant facts from the external knowledge base using the image and question, and the retrieved facts are used to construct a fact graph. Then, they utilize a GCN to perform reasoning over the retrieved facts to identify the supporting fact and answer. In addition to fact-based VQA, Singh *et al.* [162] extend the classical gated GNN [163] to integrate information from the knowledge base, question, and image for knowledge-enabled VQA. They further use the knowledge base to interpret text in the image. Marino *et al.* [164] construct a symbolic knowledge graph from the knowledge base and perform explicit reasoning on the graph via a relational GCN [165]. Then, the results of such explicit reasoning and Transformer-based implicit reasoning are integrated for answer prediction.

5.2 Visual Grounding

Visual grounding aims to locate the referent in an input image given a natural language expression. As in the VQA task, GNNs are used to capture dependencies in visual and linguistic components. One type [166] of GNN-based pipeline for visual grounding is similar to the one in VQA (Eq. 15) but replaces the prediction layer with a bounding box matching branch or bounding box refiner. Alternatively, other pipelines [142] involve employing a separate text encoder, followed by the fusion of textual information into the visual graph in the following manner:

$$\mathcal{G} = \Pi(\hat{I}),\ T = \mathrm{E}(\hat{T}),\ \mathcal{F} = \mathrm{P}(\mathrm{GNN}(\mathcal{G}, T)), \qquad (16)$$

where $\mathrm{E}(\cdot)$ denotes the text encoder, and T is the corresponding text embedding. GNNs encode structural information in both visual and linguistic data by effectively capturing the dependencies among distinct objects in two modalities, thus enhancing their ability to handle tasks involving diverse object interactions.

Referring Expression Grounding. As shown in Fig. 8, Yang *et al.* [143] construct a language-guided visual relation graph, that is conditioned on the expression, to represent the image. They apply a gated GCN to the graph to learn visual-linguistic fusion and capture multi-order semantic contexts. Yang *et al.* [166] follow the same graph construction [143], but perform dynamic graph attention and convolution on the graph to achieve stepwise graph reasoning under the guidance of the linguistic structure of the expression. Similar to [166], Hu *et al.* [147] also generate a dynamic visual graph conditioned on the expression to capture complex relations. Instead of multi-order relations considered in [143], [167], Wang *et al.* [168] form a visual graph using direct intra-relations and inter-relations between neighboring objects. They use language-guided graph attention and aggregation to update node features. Different from previous methods, Chen *et al.* [169] construct a multi-modal graph with randomly initialized bounding boxes as nodes and semantic relations between nearest neighboring nodes as edges. They utilize a GAT to update node features and bounding boxes, and introduce a graph Transformer to prune the nodes and edges iteratively. In addition, GNNs [18], [124] have been utilized for referring expression segmentation [170], [171] and 3D visual grounding on point clouds [172] in a similar manner as for referring expression grounding.

Phrase Grounding. Phrase grounding locates the referent in the image for each phrase in the expression. The referent of one phrase may depend on the referents of other phrases. Bajaj *et al.* [142] first construct a phrase graph, where nodes correspond to noun phrases and edges connect pairs of noun phrases, and then apply a gated GNN [163] to capture the dependencies among phrases. They also perform similar graph computations on a visual graph and a visual-linguistic fusion graph to jointly predict the referent for each phrase. Unlike the approach in [142], Yang *et al.* [173] perform dependency parsing on the expression to construct a language scene graph with noun phrases as nodes and relations between phrases as edges. They develop a one-stage graph-based propagation network for phrase grounding. The

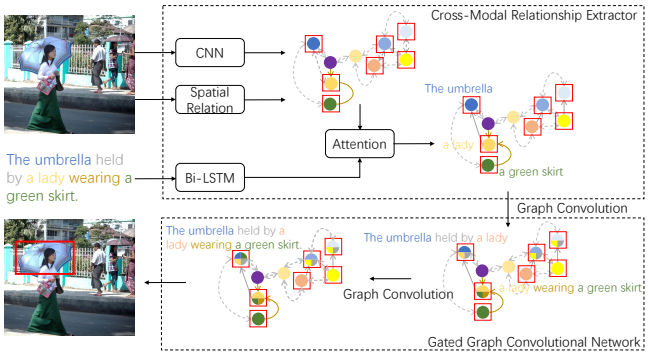

Fig. 8: Cross-Modal Relationship Inference Network [143]. Different colors correspond to different proposals after fusing with language. Then graph convolution is utilized to model the semantic contexts with multi-order relationships. The output is the proposal corresponding to the node most matching the language.

methods in [174], [175] use the same language scene graph in [173], and apply GCNs and attention mechanisms to the graph to learn visual features for graph nodes. In addition to the language scene graph, the method in [176] constructs a visual scene graph [177] and extends the standard GCN to learn disentangled representations for various motifs in the graphs and capture motif-aware contexts.

Video Grounding. For video grounding, it is vital to capture temporal or spatial-temporal correlation. Zhang *et al.* [178] construct a spatial relation graph over objects in each frame and a temporal dynamic graph where nodes correspond to objects in all frames and edges connect the same objects in consecutive frames. Zhao *et al.* [179] develop a different way of constructing a temporal graph, where each node represents an interval, and each direct edge represents the overlapping relation between two intervals.

5.3 Image and Video Captioning

Image Captioning. Image captioning aims to generate a complete and natural description of an image. The relations between objects in the image are natural priors for image description. Since GNNs model data using graphs, enabling them to capture contextual information among objects in images to enhance contextual awareness, which empowers GNNs to generate more accurate and coherent captions in comparison to traditional methods. Yao *et al.* [180] construct a semantic graph and a spatial graph to represent objects and their semantic and spatial relations. Then, they extend the GCN [18] on the graphs to obtain object representations with inherent visual relations, and use the attention LSTM [181] to decode the caption from such representations. Yao *et al.* [182] use the same graph construction and GCN as in [180] and extend single-level object relation modeling into three hierarchical levels including the image, regions and objects. Unlike previous work only capturing visual relations, Yang *et al.* [183] use a GCN and sentence reconstruction to capture the language inductive bias to form a dictionary at the training stage. At the inference stage, they utilize attention between an image scene graph and the dictionary to embed the language inductive bias into visual representations. Unlike previous methods directly decoding GNN-based context representations of the image scene graph into sentences, the method in [184] decomposes a scene graph into a set of sub-graphs and chooses important sub-graphs to decode at each time step. To achieve user desired captioning and improve the diversity at a fine-grained level, Chen *et al.* [185] use an abstract scene graph to control the described objects, attributes, and relations in the caption. A multi-relational GCN [165] is used to encode the abstract scene graph grounded in the image. Nguyen *et al.* [186] combine an image scene graph with a human-object interaction graph extracted from [187] to enhance salient parts of the scene graph. Then, they use a similar GCN-LSTM architecture as in [180] to generate the caption.

Video Captioning. Video captioning requires generating a description for a video. To learn the interactions among objects over time, Zhang *et al.* [188] construct an object relation graph to connect each object with top-k corresponding objects in all frames and use a GCN to learn the relations on the graph. As for image captioning, they generate descriptions corresponding to object relational features via an attention LSTM [181]. With the same motivation as [188], Pan *et al.* [189] construct a spatio-temporal graph and adopt a GCN to update node features. For graph construction, they first take objects in individual frames as nodes and normalized IOU between objects as edge weights, then connect semantically similar objects in two consecutive frames. Contrary to previous GCN-LSTM architectures, Chen *et al.* [190] aggregate information in a temporal graph and feed the aggregation result into an LSTM decoder, which adjusts the graph structure according to its hidden state and, in the meantime, updates the hidden state according to the graph representation.

5.4 Others

Image-Text Matching. Image-text matching aims to measure semantic similarity between a pair of image and sentence, which plays a vital role in visuo-linguistic cross-modal content retrieval. Since local similarity between regions and words contributes to global semantic similarity, Huang *et al.* [191] utilize a cross-modal GCN to align visual regions and word representations for global similarity computation. Unlike the method in [191], Li *et al.* [192] first learn a relation-enhanced global image representation based on local regions and their relations, and then match this image representation with the sentence representation. A GCN is used to perform relation reasoning on image regions to generate relation-enhanced features. Similar to the method in [192], Yan *et al.* [193] also use a GCN to reason about semantic relations between image regions. In addition, they propose a discrete-continuous policy gradient algorithm to transform images and texts into a common space. Wang *et al.* [194] improve region-based relation reasoning by introducing commonsense knowledge into the reasoning process. They construct a concept correlation graph and learn consensus-aware concept representations via stacked GCN layers. Besides constructing a relation graph on image regions, Liu *et al.* [195] and Li *et al.* [196] build a textual graph for the sentence via a dependency parser and perform semantic matching between the two graphs via a GCN and

an attention mechanism. In addition, GNNs have also been utilized for video-text matching [197], [198], [199].

Vision-Language Navigation. Vision-language navigation requires an agent to navigate an unknown environment following a natural language instruction [200]. Deng *et al.* [201] develop an evolving graphical planner model, which iteratively and dynamically expands a trajectory graph built on the current node, visited nodes, potential action nodes and their relations, and runs a GNN on the graph to predict an action. Unlike the method in [201], Chen *et al.* [202] use a GNN to capture the visual appearance and connectivity of the environment from its topological map. Gao *et al.* [203] involve external knowledge in action prediction and run a GCN on external and internal environment knowledge graphs to extract external knowledge used for cooperating with internal knowledge. Chen *et al.* [204] generate a layout graph at each step according to the instruction and use graph convolution to learn the current navigation state from the current layout graph and history layout memory. The navigation state is used to predict the action probability distribution at the current step. Like the method in [201], Chen *et al.* [205] construct an environment graph on visited, navigable, and current nodes. Furthermore, they aggregate information in the graph using self-attention to predict the navigation score for each node.

Video Question Answering. Similar to the VQA approach in [141], Park *et al.* [206] represent the question as a parse question graph. To capture temporal interactions in a video, they also construct a frame-based appearance graph and a motion graph. Then, GCNs [18] are used to learn question-appearance and question-motion interactions on the three graphs. Unlike previous methods that use GNNs to learn relation-embedded node representations, Yu *et al.* [207] utilize GNNs to adaptively propagate predicted probabilities among samples in the video to obtain consistent results for samples with high visual similarity. Instead of exploiting frame or motion-level information, some works [208], [209] suggest constructing object-level visual graphs. In particular, Liu *et al.* [208] combine GNNs and memory networks [210] to perform dynamic relation reasoning on visual and semantic graphs at the object level.

5.5 Discussion

Natural language sentences contain natural dependency structures, while images can take the form of spatial, semantic, or scene graph representations. GNNs thus are utilized naturally by the vision-and-language community to capture the relations among visual, linguistic, or multi-modal components, or perform cross-modal alignment between visual and linguistic graphs. Nevertheless, there still exist open problems in the exploitation of GNNs in vision-and-language research. For example, is it possible to design a unified GNN-based architecture for different vision-and-language tasks, especially in view of the similar roles played by GNNs in different tasks? How do we better explore the reasoning ability and explainability of GNNs beyond neighborhood aggregation and message passing? How do we further incorporate knowledge graphs for achieving knowledge-aware graph reasoning?

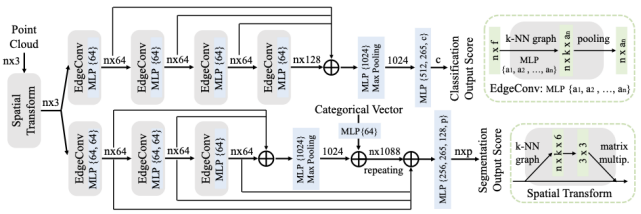

Fig. 9: Dynamic Graph Convolution Neural Network (Figure used courtesy of [124]).

6 3D Data Analysis

6.1 3D Representation Learning

Among various 3D data representations, point clouds and meshes receive increasing attention due to their strong ability to represent irregular structures and complicated 3D shapes. In this task, GNNs can efficiently capture complex geometric and topological relationships inherent in 3D structures and are versatile in handling varying sizes and resolutions of 3D data, making them ideal for tasks that require robust and scalable 3D analysis.

6.1.1 Point Cloud Representation

PointNet [211] is a pioneering work for point cloud representation learning that takes the raw point cloud as the input and processes each point independently for feature representation. Such point-based methods include PointNet++ [212], PointCNN [213], etc. Another type of methods is geometry-based, including KPConv [214] and MinkowskiNet [215], which use mathematical models to represent the shape and structure of the point cloud. However, both point-based and geometry-based methods treat each point as an individual entity yet do not directly consider the relationships between points, which limits robustness when applied to more noisy, complex, or incomplete point clouds.

Naturally, GNN-based approaches model the input point cloud as a spatial graph and apply a GNN or graph Transformer to characterize local or global interaction among points. A pioneering graph-based approach, ECC [216], generalizes convolutions on regular grids to point cloud based graphs. ECC computes adaptive convolution kernel weights using specific edge labels in the neighborhood of a vertex to better utilize edge information. A recent work, AdaptConv [217], also focuses on generating adaptive kernels but according to learned features. In the above methods, the graph structure is determined by the input point cloud, more specifically, local point neighborhoods, which limits flexibility and non-local relation modeling. To address these issues, DGCNN [124] (cf. Fig. 9) designs an EdgeConv operator and proposes to construct graphs in the feature space and dynamically update graphs in each layer so that both global and local interactions can be flexibly captured. Following ECC and DGCNN, a series of works are contributed to improving the spatial-domain graph convolutions for point cloud representation, including [218], [219], [220], [221], [222], [223] Different from the mainstream approach, a random walk based method, CurveNet [224], generates sequences of point segments as non-local descriptors to better depict point cloud geometry.

In contrast to performing graph processing operations in the spatial domain, spectral methods build on a mathe-

TABLE 5: Overall Accuracy (oAcc.) on the point cloud classification task of the ModelNet40 dataset.

	Method	Reference	oAcc. (%)
non-GNN-based	PointNet [211]	CVPR'17	89.2
	PointCNN [243]	NeurIPS'18	92.2
	KPConv [214]	ICCV'19	92.5
GNN-based	LocalSpecGCN [229]	ECCV'18	92.1
	DGCNN [246]	TOG'19	92.9
	DeepGCN [28]	ICCV'19	93.6
	PT [35]	ICCV'21	93.7
	CurveNet [224]	ICCV'21	94.2

matically elegant approach that defines graph convolution as spectral filtering, which relies on the eigen-decomposition of the graph Laplacian [20], [225], [226], [227], [228].LocalSpecGCN [229] performs spectral filtering on dynamically built local graphs while recursively clustering spectral coordinates to support graph pooling, which improves PointNet++ [212] via alleviating point isolation, and also addresses the limitation of traditional spectral GCNs that the graph Laplacian and pooling hierarchy need to be pre-computed for the full graph. HGNN [230] designs a hyperedge convolution operation to learn high-order data correlation for representing the complex structure in a point cloud. Instead of using a fixed view, DiffConv [231] is operated under adaptive views to match local point cloud density. Wiersma *et al.* [232] design a graph-based anistropic convolutional operator by combining a set of geometric operators defined on scalar and vector fields to encode the directional information of each surface point.

In addition to graph convolution based methods, another line of work aims to improve graph attention neural networks for point cloud analysis [233], [234], [235], [236]. In such work, during graph feature aggregation, attention weights are computed from both neighbors' positions and learnable features to adaptively attend important graph nodes or regions. In addition, graph-RNN is explored in [237] to consider relations between both spatially and temporally neighboring points for point cloud sequence analysis.

As Transformers have been extensively explored in computer vision, graph-based Transformers designed for 3D point cloud representation learning have also been proposed [35], [36]. The core mechanism of Transformers, self-attention, naturally enables the network to build and represent global relations. PT [35] computes self-attention on local k-NN graphs, which leads to high-quality local representations while global relations are captured in deep layers. In comparison, PCT [36] adapts global self-attention to learn both local and non-local interactions, which sacrifices computational efficiency for representation capability. Recently, Lu *et al.* [238] further proposed 3DCTN to combine a Transformer and graph convolutions to achieve both efficient and powerful 3D point representations. As the agglomeration process of point clouds (including point sampling, grouping, and pooling) is complex, some methods [223], [239], [240], [241], [242] have been proposed for simplifying the procedure or promoting the efficiency. The performance of some GNN-based algorithms on the point cloud classification task is given in Tab. 5 for a comparison with non-GNN-based ones [211], [214], [243]. On the ModelNet40 dataset, advanced GNN-based methods [35], [224], [244], [245] achieve SOTA performance of point cloud classification.

6.1.2 Mesh Representation

A polygonal mesh discretely represents the surface of a 3D shape with faces and vertices. A mesh has a set of vertices that are connected by edges to form faces, therefore, a mesh can be directly represented as an undirected graph.

A line of work explores representation learning over meshes by performing operations such as convolution and pooling on different mesh components, *e.g.*, vertices [247], edges [248], faces [249], or edges and vertices [250]. FeaStNet [247] performs graph convolution in each local neighborhood centered at a vertex to aggregate the features at vertices connected to the center vertex. As local neighborhoods in a mesh have irregular structures, FeaStNet learns multiple weight matrices for graph convolution and dynamically chooses one of the weight matrices for each vertex in a neighborhood based on the vertex feature. MeshCNN [248] adopts a different approach, where mesh convolution, pooling, and unpooling are operated on edges since every edge is adjacent to two triangle faces defined by additional four nearby edges, as illustrated in Fig. 10. Such edge-oriented operations have a fixed neighborhood size and derive a simple network for mesh feature extraction. Besides, MeshNet [249] introduces convolution operated on mesh facets as well as feature splitting to obtain spatial and structural mesh features. DCM-Net [250] performs graph-based operations on both edges and vertices by jointly exploiting geodesic and Euclidean convolutions, which encourages contextual information flow among spatially or geodesically nearby patches. PD-MeshNet [251] constructs a pair of primal and dual graphs on an input 3D mesh to connect the features of mesh faces and vertices and employs a graph attention network to dynamically aggregate these features.

Recently, many studies have focused on designing new GNN operators to improve feature extraction. SpiralNet++ [252] improves [253] with a fast and efficient intrinsic mesh convolution operator that fuses features from adjacent nodes with multi-scale local geometric information. A stacked dilated mesh convolution is proposed to inflate the receptive field of graph convolution kernels [254]. PolyNet [255] develops convolution and pooling operations that are invariant to the scale, size, and perturbations of local patches. He *et al.* [256] improve the graph convolution operation by learning direction sensitive geometric features from mesh surfaces. GET [257] builds an efficient Transformer with both gauge equivariance and rotation invariance on triangle meshes. Recently, SubdivNet [258] performs representation learning on meshes using connectivity from the Loop subdivision sequence, by building hierarchical mesh pyramids. Dong *et al.* [259] consider graph convolutions and pooling in the spectral domain, and maps mesh features in the Euclidean space to the multi-dimensional Laplacian-Beltrami space. Different from the custom of constructing graphs with regular neighborhood structures, MeshWalker [260] takes a similar approach as in [224] for mesh analysis, and random walks along edges are used for 3D shape information extraction. AttWalk [261] enhances [260] by exploring meaningful interactions among different walks.

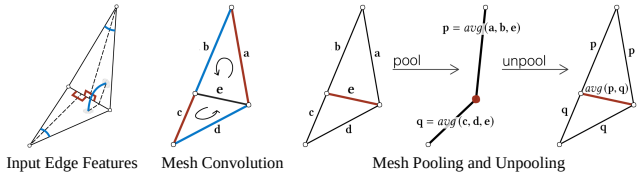

Fig. 10: MeshCNN (Figure used courtesy of [248]).

TABLE 6: Performance (mIoU) on the point cloud semantic segmentation task of the S3DIS Area-5 dataset.

	Method	Reference	mIoU (%)
non-GNN-based	PointNet [211]	CVPR'17	41.1
	PointCNN [243]	NeurIPS'18	57.3
	KPConv [214]	ICCV'19	67.1
GNN-based	SPG [262]	CVPR'19	61.7
	Point-Edge [263]	ICCV'19	61.9
	GACNet [234]	CVPR'19	62.9
	PT [35]	ICCV'21	70.4
	PTv2 [264]	NeurIPS'22	72.6

6.2 3D Understanding

GNNs are powerful neural networks capable of representing both local and global regions, which leads to accurate context-aware semantic understanding. 3D understanding tasks that can exploit GNN algorithms include 3D segmentation, detection, and visual grounding. Based on GNN-based 3D representation learning approaches, the methods proposed for 3D understanding tasks further consider instance-level and/or scene-level features, and GNNs play an important role in learning inter-instance and contextual information. Denote the 3D input as I, a graph $\mathcal{G}$ is constructed first at the input level, and the GNN-based methods serve to extract low-level features $\mathcal{F}$ from $\mathcal{G}$. In some methods, the features $\mathcal{F}$ are further aggregated via constructing a region-level or instance-level graph $\mathcal{G}'$, on which GNN-based methods are employed to incorporate the high-level relations into a more representative feature $\mathcal{F}'$ for the downstream task, such as segmentation or detection. The whole process can be formulated as:

$$\mathcal{F} = \text{GNN}(\Pi(I)), \ \mathcal{F}' = \text{GNN}(\Pi(\mathcal{F})). \tag{17}$$

6.2.1 3D Point Cloud Segmentation

3D segmentation, *e.g.*, semantic segmentation, needs to understand both global scene semantics and fine-grained point-wise signatures. GNNs can efficiently capture complex inter-point, inter-region, and inter-object relationships, enabling robust feature extraction and scene understanding that adapt to irregular geometric structures. Existing work primarily focuses on how to construct graphs and achieve better local feature aggregation and global interaction. Note that some studies may involve both aspects, but we categorize them based on their core contributions.

Graph Construction. Qi *et al.* [265] make the first attempt to apply GCNs to this task. They construct a large-scale graph at the point level for each 3D scene to incorporate geometric relations in a learning-based way, so that semantic information from the RGB images and geometric information from the point clouds can be combined through a GNN. Differently, SPG [266] constructs graphs at the superpoint level. Each superpoint is a cluster of nearby points that have similar semantic meanings. A graph constructed on such geometric partitions has a simpler structure and smaller scale, giving rise to high-quality and efficient segmentation. Landrieu *et al.* [262] further propose a graph-structured contrastive loss and a cross-partition weighting strategy to better oversegment a point cloud into superpoints in a supervised manner, which results in high contrast at object boundaries and a compact representation for each segment.

Feature Aggregation. To better handle irregular structures of point clouds, GACNet [267] performs graph attention convolutions on local point subsets, where attention weights are learned from both 3D coordinates and point features, so that the network can attend most relevant neighboring points to better model the 3D structure of a point cloud. To utilize all available local contextual information, PointWeb [268] densely connects all point pairs in a local region and updates point features using nonlinearly transformed pairwise feature differences. Compared to focusing each local graph at a point, PointWeb's feature aggregation provides an improved depiction of local regions via enhanced interactions among adjacent points. Jiang *et al.* [263] construct a hierarchical graph, where a point branch and an edge branch interact with each other. PyramNet is equipped with a Graph Embedding Module [269] to strengthen its ability to represent local geometric details. Combining depthwise graph convolutions and pointwise convolutions, HDGCN [270] customizes DG-Conv to better extract local features. Lei *et al.* [271] build each local graph as a ball space around a center point, so that the weights of its neighboring points can be efficiently determined for convolution by its location pre-defined by a volumetric division in the ball. Grid-GCN [272] leverages the efficiency of a grid space to significantly accelerate the process of structuring point clouds. Point2Node [273] explores the correlation among graph nodes at different scales. RGCNN [226] applies graph convolution in the spectral domain to part segmentation. SegGroup [274] runs GCNs to decrease the difference among nodes from the same instance and increase the variance among nodes from different instances.

The performance of GNN-based methods and non-GNN-based ones [211], [243], [275] on the point cloud semantic segmentation task is given in Tab. 6. As shown, on the S3DIS-Area5 dataset, point-transformer-based methods [35], [264] achieve SOTA segmentation accuracy.

6.2.2 3D Object Detection

3D object detection aims to localize and recognize objects in point clouds or RGBD data. Compared with traditional 3D object detection methods, GNN-based methods improve feature extraction at the point level by capturing cross-point relations to facilitate non-local learning. Liang *et al.* [276] focused on multi-sensor detection by fusing image information from cameras with point clouds from LiDAR, and proposed continuous fusion layers, which compute dynamic convolution kernels for better integration between different modalities [277]. Shi *et al.* [278] proposed Point-GNN for one-stage 3D object detection, where neighboring points within a fixed radius are connected, and an auto-registration method is designed to reduce the translation variance by dynamically refining point positions. HGNet [279] adopts a GCN with a hierarchical structure to adequately exploit multi-level semantics for object detection.

GNNs are also employed to capture inter-object relations through object-level graphs, which have been proven beneficial to the understanding of 3D scenes with multiple objects. Feng *et al.* [280] built an inter-object relation graph to enhance proposal features for higher-quality detection. Object DGCNN [281] models object relations via dynamic graph convolutions [246] to learn better features for object queries. PointRGCN [282] introduces both point- and object-level GNNs to thoroughly model inter-point and inter-object relations as well as contextual information.

Discussion. GCN-based methods are the most popular solutions for 3D understanding, where both local neighborhoods and global contexts can be modeled. Recent works [35], [36] proposed Transformer architectures for stronger feature extraction via the self-attention mechanism but at the cost of efficiency. However, the problem of a good combination of graph convolution and self-attention for 3D understanding has been less explored. A future direction would be a comprehensive framework that performs point-level, object-level, and scene-level understanding, and leverages the complementary advantages of MLP, graph convolution, and self-attention to balance between efficiency and performance. In addition to point clouds and meshes, automated panoptic symbol spotting [283], [284], which is crucial for creating 3D prototypes in architecture, engineering, and construction industries, introduces graphical networks to model symbol-wise dependencies and has received growing attention.

6.3 3D Generation

Approaches for 3D generation usually use an encoder for input representation and a decoder to transform latent features into 3D outputs. Besides being used in the encoder for feature extraction as in 3D understanding tasks (Eq. 17), GNNs are widely adopted to (i) learn required manipulations of a 3D input, *e.g.*, adjusting point positions for the task of point cloud denoising; (ii) generate a 3D shape from a reference input, *i.e.*, 3D reconstruction.

6.3.1 Point Cloud Completion and Upsampling

Scanned data from 3D sensors are usually sparse, non-uniform, and incomplete. Point cloud completion and upsampling aim to produce complete and dense point clouds from real scans. Denote the sparse or incomplete input as I, it aims to output more points $\triangle I$ to obtain a dense or complete one $I' = \{I, \triangle I\}$. A typical pipeline uses a feature extractor and an upsampling network as follows:

$$\mathcal{F}' = \texttt{Feature}(I),\ I' = \texttt{Upsample}(\mathcal{F}'). \tag{18}$$

In this task, GNNs can efficiently and dynamically capture the geometric structure of point clouds, providing both global and local topological information to avoid generating inconsistent content.

The challenge of completion lies in recovering complete object shapes from incomplete information. ECG [285] is a two-stage framework that exploits skeleton as the intermediate structural representation and introduces a hierarchical encoder based on graph convolution to extract multi-scale edge features. In [286], leap-type EdgeConv [124] is used to capture local geometric features, and a cross-cascade module is used to hierarchically combine local and global features. Shi *et al.* [287] regard point cloud completion as a generation task, and cast the input data and intermediate generation as controlling and supporting points, and a GCN is utilized to guide the optimization process.

For the task of upsampling, a multi-step point cloud upsampling network is proposed in [288] to compute the k-NN graph for feature aggregation. Unlike PU-GCN [289] that integrates densely connected graph convolutions into an Inception module, AR-GCN [290] incorporates residual connections into a GCN to exploit the correlation between point clouds with different resolutions, as well as a graph adversarial loss to capture characteristics of high-resolution ones. PUGeo-Net [291] extends DGCNN by two modules for feature recalibration and point expansion.

6.3.2 3D Data Denoising

Denote a 3D input with noise as I, the denoising task aims to estimate the noise component N, which leads to the denoised one $I' = I - N$. A typical pipeline uses a feature extractor and a denoising network:

$$\mathcal{F}' = \texttt{Feature}(I),\ I' = \texttt{Denoise}(\mathcal{F}'), \tag{19}$$

For this task, GNNs can adaptively learn from local neighborhood structures for noise reduction, are robust to irregular sampling, and integrate multiple features (*e.g.*, position, normal, color) of each point in the data.

Given that the denoising task is more concerned with local representations of point neighborhoods, Pistilli *et al.* [292] introduce a new dynamic graph framework to achieve better feature aggregation. To address this issue, DMR [293] builds on DGCNN [124] to learn the underlying manifold of the noisy input from differentiably subsampled points and their local features for less perturbation. GPDNet [292] proposes to build hierarchies of local and non-local features to regularize the underlying noise in the input point cloud. For 3D mesh denoising, Armando *et al.* develop a multi-scale GCN in [294], where the algorithm is built on CNN-based image denoising techniques. More recently, GCN-Denoiser [295] learns a rotation-invariant graph representation for local surface patches, and performs graph convolutions over both static graph structures of local patches and dynamic learnable structures. GeoBi-GNN [296] excavates the dual-graph structure in meshes to capture both position and normal noises through a GNN-based U-Net.

6.3.3 3D Reconstruction

3D reconstruction recovers 3D point clouds or meshes from lower-dimensional inputs. Mainstream paradigm adopts an encoder-decoder architecture, where a low-dimensional input I is encoded into latent features and then decoded into a 3D output:

$$\mathcal{F}' = \texttt{Encode}(I),\ O = \texttt{Decode}(\mathcal{F}'), \tag{20}$$

For unconditional generation, only `Decode` is needed to decode a random code sampled from a pre-defined distribution over 3D outputs. Compared with other methods, GNNs can provide sparse dependency modeling and more detailed geometric description for 3D reconstruction.

Point Cloud Reconstruction. FoldingNet [297] has a graph-based auto-encoder that deforms a 2D grid into a point cloud with delicate structures. In [298], a unified latent space

is defined for graph representations, and a GNN named StructureNet is able to encode or generate point clouds. Inspired by the success of GANs, some works focus on unsupervised point cloud generation, which incorporates graph convolution into the decoder [299], [300] or even the self-attention mechanism [301] for better generation quality.

Mesh Reconstruction. Pixel2Mesh [302] employs a GNN for mesh reconstruction from a single image. Instead of learning a direct mapping from images to meshes, targeted at multi-view mesh reconstruction, Pixel2Mesh++ [303] uses a GCN to predict a series of deformations to refine a coarsely generated shape. Mesh R-CNN [304] considers reconstruction from images with multiple objects. It augments Mask R-CNN [305] with a GCN, which performs 3D shape inference by refining a coarsely cubified mesh. For the task of mesh reconstruction from image volumes, Voxel2Mesh [306] employs a GCN in the decoder to refine a sphere mesh to match a target shape. As a more editable input, scene graphs are considered in [307] to generate and manipulate 3D scene meshes via a GCN-based variational auto-encoder.

Human-related Mesh Reconstruction. For human body reconstruction, Kolotouros *et al.* [308] improve the template-based mesh regression approach through a GCN to explicitly encode the template mesh structure and aggregate image features. A bilayer graph is designed in [309] to represent the body shape and pose simultaneously, with a GCN deployed to jointly perform pose estimation and mesh regression. DC-GNet [310] also adopts a GCN and further takes into account occlusion cases where human bodies are incomplete in images. The graph Transformer is exploited in Mesh Graphormer [40] for human mesh reconstruction. For reconstruction from 2D human pose, Pose2Mesh [311] and GTRS [312] respectively builds on GCNs or graph Transformers for pose-guided mesh regression. Besides human body, the face and hand reconstruction are also explored where GCN is a popular structure [313], [314], [315].

Discussion. Recently, the self-attention mechanism has been adopted for capturing global interactions [35], [40]. However, graph Transformers have not been adequately explored for general 3D object and scene reconstruction, and more graph Transformers are expected to be developed. In addition, in view of the large data size in 3D generation tasks, a practical direction would be promoting the efficiency of graph convolutional networks and graph Transformers.

7 Medical Image Analysis

The applications of GNNs on medical image analysis include, but are not limited to, brain activity investigation, disease diagnosis, and anatomy segmentation.

7.1 Brain Activity Investigation

Approaches for brain activity analysis can be roughly divided into two categories: subject- and region-based, according to the granularity of graph nodes. For subject-based methods (cf. Fig. 11), each node comprises characteristics of a subject, and GNNs are applied to graphs consisting of $\hat{N}$ subjects:

$$\begin{aligned} S_i &= \Pi_s(\hat{I}_i), \\ \mathcal{F} &= \texttt{GNN}(\{S_i\}_{i=1}^{\hat{N}}), \end{aligned} \tag{21}$$

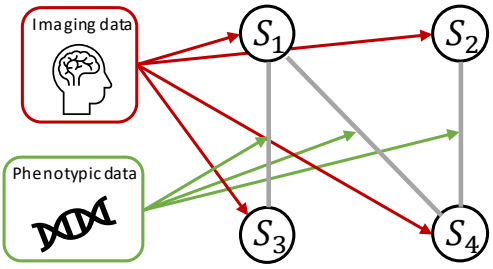

Fig. 11: Subject-based brain analysis. S_i denotes the i-th subject. Note that the node and edge features may vary depending on the task.

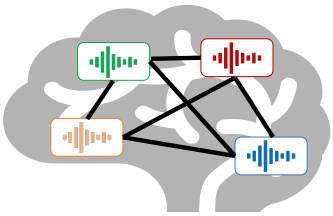

Fig. 12: Region-based brain analysis. Colors denote brain regions with different functions/structures.

where $\Pi_s(\cdot)$ denotes the graph construction process on top of N subjects. $\hat{I}_i$ and S_i represent the subject-level raw data and model input, respectively. $\mathcal{F}$ stands for the extracted graph representations. In comparison, region-based approaches construct graphs based on brain regions (cf. Fig. 12), where each node contains the representation of a specific brain area. Similar to subject-based methodologies, region-based methods can be formulated as follows:

$$\begin{aligned} R_i &= \Pi_r(\tilde{I}_i), \\ \mathcal{F} &= \texttt{GNN}(\{R_i\}_{i=1}^{\tilde{N}}), \end{aligned} \tag{22}$$

where $i = 1, 2, ..., \tilde{N}$. $\Pi_r(\cdot)$ denotes the graph construction process on top of brain areas. $\tilde{I}_i$ and R_i stand for the raw data and graph feature input of the i-th brain area, respectively.

7.1.1 Brain Signal Disorder Recognition

Autism spectrum disorder (ASD). Autism spectrum disorder is a type of developmental disorder that often adversely affects people's social communication and interaction. Recently, GNNs have been employed to identify ASD in functional magnetic resonance imaging (f-MRI) [316].

Subject-based methods. Parisot *et al.* [317] introduced spectral GCNs to conduct brain analysis in populations. The imaging feature of each subject serves as the vertex in the graph. The graph edge would be built if the similarity score of two subjects' imaging and non-imaging (e.g., phenotypic data) features is above an artificially defined threshold. The proposed GCN framework outperforms the archetypical ridge classifier by at least 5% on different datasets. Anirudh *et al.* [318] addressed the instability problem in the step of population graph construction (with fewer edges) and introduced the bootstrap approach to construct an ensemble of multiple GCNs, which provides performance gains over [317]. Rakhimberdina and Murata [319] replaced the spectral GCNs in [317] with linear GCNs and optimized the edge construction step. Jiang *et al.* [320] proposed hybrid GCNs that maintain a hierarchical architecture to jointly

incorporate information from both subject and population levels. Peng *et al.* [321] applied federated learning to train GCNs on inter-institutional data by first inpainting local networks with masked nodes and then training a global network across institutions.

Region-based methods. Ktena *et al.* [322] used siamese GCNs to conduct distance metric learning on functional graphs of single subjects, outperforming the Principal Component Analysis (PCA) based baseline by over 10% on different imaging sites. Based on the work [322], Yao *et al.* [323] proposed triplet GCNs that contrast matching and non-matching functional connectivity graphs at the same time. In addition, multi-scale templates [323] were employed to advance the graph representation learning results. Li *et al.* [324] modified the ranking-based pooling methods by adding the regularization, enabling better node selection and prediction interpretation. In BrainGNN [325], ROI-aware graph convolution was introduced to encode the locality information into associated node embeddings. Kazi *et al.* [326] introduced inception GCN, which integrates spectral convolution with different kernel sizes to capture the intra- and inter-graph structural information.

Other disorders. Brain imaging can help distinguish patients with major depressive disorder (MDD) from normal ones. In [327], Yao *et al.*proposed an adaptive GCN framework to incorporate both spatial and temporal information from associated time-series f-MRI data. Yao *et al.* [328] aligned multi-index representations of f-MRI and encoded them into a shared latent feature space for making MDD predictions. For the diagnosis of bipolar disorder, Yang *et al.* [329] employed GAT networks and hierarchical pooling strategies to deal with variable-sized inputs. Rakhimberdina and Murata [319] applied linear GCNs to tackle the diagnosis issue of schizophrenia that is a mental illness related to the barrier of perceiving the reality.

7.1.2 Brain State Decoding

Zhang *et al.* [330] used deep graph convolution to interpret Resting-State Functional MRI (rsf-MRI) data into associated cognitive task results spanning six cognitive domains. In [331], the spatial-temporal graph convolution was employed to predict age and gender based on the BOLD signal of rsf-MRI data. Kim *et al.* [332] proposed to learn dynamic graph representations using spatial-temporal attention while Transformer serves as the encoder model to process a short sequence of brain connectome graphs.

7.2 Disease Diagnosis

In this section, we mainly review the related literature on diagnosis of brain and chest diseases using GNNs. Specifically, for brain diseases, we involve Alzheimer's and Parkinson's diseases as two representatives. As for chest pathologies, we do not restrict the type of disease.

7.2.1 Brain Disease Identification

Alzheimer's disease (AD). Alzheimer's disease is a type of neurodegenerative disease that gradually destroys brain cells. Methods for AD diagnosis are mostly based on subject-level information (cf. Fig. 11 and Eq. 21).

Parisot *et al.* [333] formulated the subject diagnosis issue as a graph labeling problem, where a spectral GCN was introduced to make diagnoses based on population analysis. As shown in Fig. 11, imaging data provide node features, while phenotypic data are interpreted as edge weights. The proposed approach surpasses previous state-of-the-arts by about 5% at least. Zhao *et al.* [334] used a GCN-based approach to detect Mild cognitive impairment (MCI) in AD. In [335], Liu *et al.*deployed a GCN model to identify early mild cognitive impairment (EMCI, an early stage of AD) with a multi-task feature selection method on both imaging and non-imaging data. Yu *et al.* [336] designed a multi-scale GCN framework that integrates each subject's structural information, functional information, and demographics (i.e., gender and age) for diagnosing EMCI based on population. Huang *et al.* [337] constructed an adaptive population graph with variational edges. The uncertainty of the adaptive graph was estimated using the proposed Monte-Carlo edge dropout [337]. Ma *et al.* [338] incorporated local and global attention modules into GNNs. Besides, an attention-guided random walk strategy [338] was applied to extract features from dynamic graphs (with variable-sized nodes). Wee *et al.* [339] developed a cortical GNN model to take into account the cortical geometry and thickness information for identifying AD or MCI.

Parkinson's disease (PD). Approaches for PD diagnosis mainly construct graphs based on brain regions (cf. Fig. 12 and Eq. 22). Zhang *et al.* [340] implemented a multi-modal GCN model that fuses different brain image modalities for distinguishing PD from the healthy control, outperforming traditional diagnosis machine learning models by large margins. Kazi *et al.* [341] developed a LSTM-based attention mechanism to learn subject-specific features by ranking multi-modal features, and the fused representation was used for making final decisions. In [342], an end-to-end deep GCN framework was introduced to encode the cross-modality relationship into the graph while learning the mapping from structures to brain functions.

7.2.2 Chest Disease Analysis

Chest disease analysis methods mainly build graphs upon anatomies or textual attributes, both of which are similar to the region-based schema (cf. Fig. 12 and Eq. 22). Yu *et al.* [343] treated each CT scan as a node and connected nodes that share highest similarities. Features extracted from CNNs are passed to a one-layer GCN to help capture the relationship between similar samples. Similarly, Wang *et al.* [344] proposed to fuse image-level and relation-aware features via a GCN. Here, the GCN model is used to gather neighborhood information for deciding whether the input CT scan has COVID-19 or not. Zhang *et al.* [345] built a knowledge graph of chest abnormality based on prior knowledge on chest findings. The obtained graph was integrated with a graph embedding module, which was appended to a off-shelf CNN feature extractor. The output features of the graph embedding module can be used for both disease classification and radiology report generation. Hou *et al.* [346] pre-trained the label co-occurrence graph on radiology reports and integrated the resulting label embeddings with image-level features. The fused representations are

passed to the Transformer encoder and the following GCN layers for feature fusion. Liu *et al.* [347], [348] introduced an anatomy-aware GCN model for mammogram detection. The anatomy-aware framework mainly involves two modules: i) a bipartite GCN to model the intrinsic geometrical and structural relations of ipsilateral views. ii) an inception GCN to model the structural similarities of bilateral views. The proposed anatomy-aware GCNs not only achieved state-of-the-art results on mammogram classification but also provided interpretable diagnosis results. Lian *et al.* [349] proposed a series of relation modules to capture the relations among thoracic diseases and anatomical structures. Chen *et al.* [350] proposed an instance importance-aware GCN for applying multi-instance learning to 3D medical diagnosis, which helps to learn complementary representations by exploiting importance- and feature-based topologies. Zhao *et al.* [351] proposed a general attribute-based medical image diagnosis framework by coupling probabilistic reasoning (Bayesian network) and neural reasoning (GCN) modules to model the causal relationships between attributes and diseases.

7.3 Anatomy Segmentation

Based on the segmentation targets, we roughly divide GNN-based segmentation approaches into three groups: brain surface segmentation, vessel segmentation, and other anatomical structure segmentation. Most of these segmentation approaches are built on top of certain structures, which can be regarded as extensions of the region-based approaches (cf. Fig. 12 and Eq. 22).

7.3.1 Brain Surface Segmentation

Cucurull *et al.* [352] exploited GCNs and GATs for mesh-based cerebral cortex segmentation. They found that either GCNs or GATs can obviously outperform traditional mesh segmentation approaches, sometimes by over 5%. Gopinath *et al.* [353] used GCNs to learn spectral embeddings directly in a space defined by surface basis functions for brain surface parcellation. Specifically, the proposed GCN model employs spectral filters to handle intrinsic surface representations, which were also validated in the task of brain parcellation. To better handle impaired topology, Wu *et al.* [354] proposed to directly conduct surface parcellation on top of the cortical manifold with the help of GCNs, which produced satisfactory results on surfaces that violate the spherical topology. He *et al.* [355] used the spectral graph Transformer to align multiple brain surfaces in the spectral domain, outperforming traditional GCN-based models by observable margins.

7.3.2 Vessel Segmentation

Wolterink *et al.* [356] used graph convolutional networks (GCNs) to forecast the spatial placement of vertices in a tubular surface mesh that divides the coronary artery lumen. Zhai *et al.* [357] combined GCNs with CNNs to categorize pulmonary vascular trees into either arteries or veins. In practice, the hybrid vessel segmentation model is trained on constructed vessel graphs, where the label of each node is whether it belongs to the artery class or the vein. Shin *et al.* [358] proposed a unified model that consists of both CNNs and GNNs to leverage graphical connectivity for vessel segmentation. Specifically, the unified framework exploits local and global information from perspectives of appearances and structures, respectively. Noh *et al.* [359] extracted vessel connectivity matrices from fundus and fluorescein angiography images via a hierarchical GNN, which were used to assist the classification of retinal artery/vein classification. To tackle high variations in intracranial arteries, Chen *et al.* [360] introduced a GNN segmentation model and a hierarchical refinement module to integrate structural information with relational prior knowledge for the segmentation of intracranial arteries. Considering vessels in 3D images often have diverse sizes and shapes, Yao *et al.* [361] proposed a GCN-based point cloud network to incorporate the tubular prior knowledge into vessel segmentation, which not only helps to capture the global shape but also the local vascular structure. Yang *et al.* [362] introduced a partial-residual GCN to take into account both position features of coronary arteries and associated imaging features to conduct automated anatomical segmentation. Zhang *et al.* [363] integrated the intra-artery relation with the inter-organ dependencies for capturing anatomical dependencies. Zhao *et al.* [364] proposed a cascaded deep neural network, where cross-network multi-scale feature fusion is performed between a CNN-based U-Net and a graph U-Net to effectively support high-quality vessel segmentation.

7.3.3 Segmentation of Other Anatomical Structures

Selvan [365] proposed to use a GNN-based auto-encoder to learn node features for airway extraction while a decoder was employed to predict edges between nodes. Garcia-Uceda Juarez *et al.* [366] modified UNet [367] by replacing the convolution layers in the bottleneck with graph convolution layers, which helps encode the node connectivity into latent embeddings. The resulting hybrid UNet outperformed the classic UNet in the task of airway segmentation. In [368], a GNN-based graph refinement module was derived for to extract airway structures more accurately. Mukul *et al.* [369] formulated the post-processing procedure in pancreas and spleen segmentation as a semi-supervised graph labeling problem and proposed a GCN-based refinement strategy to replace the classic conditional random fields (CRFs) for segmentation mask post-processing. Tian *et al.* [370] applied GCNs to the interactive segmentation problem of prostate, where GCNs were responsible for refining the coarse contour produced by CNNs. Meng *et al.* [371], [372] developed a hybrid U-shape model for optic cup/disc segmentation, where the encoder consists of convolution layers while the decoder comprises graph convolution layers. The encoder and decoder is connected via attention modules, which help the decoder to leverage the more location information from the convolutional encoder. Chao *et al.* [373] presented a GNN that incorporates the global spatial priors of lymph node tumor into its learning process to further model the relations between lymph nodes. Yan *et al.* [374] first transform 3D MRI images into supervoxels, which were then passed to graph convolution layers to capture interconnections for brain tissue segmentation.

7.4 Discussion

Graph representation learning is the core idea behind GNN-based brain signal analysis, while self-supervised learning

has been shown to improve the generalization ability of representations in medical imaging [375], [376], [377], [378]. Therefore, it is promising to incorporate self-supervised learning into existing GNN-based analysis frameworks. For instance, neighboring nodes can be contrasted with non-neighboring ones to learn invariant and discriminative node features using noise-contrastive estimation [379].

Multi-disease graphs are necessary for capturing inter-disease relations to achieve reliable diagnoses. For example, a knowledge graph [380], [381] can be integrated into a hierarchical multi-disease diagnosis system, where each node includes both imaging and non-imaging features of a specific disease. The knowledge graph can be built using domain knowledge or clinical descriptions associated with images (such as radiology reports [382], [383], [384]).

GNN-based approaches resort to building graphs for encoding relations into latent representations. In practice, these graphs can be incorporated into label-efficient learning to assist medical image segmentation with limited annotations. For instance, we can apply GNNs to aggregate the representations of neighboring classes with rich annotations to improve the representations of rare diseases. Moreover, conducting graph-based reasoning [385] within GNN-based segmentation frameworks is also a promising direction, which helps improve the performance of segmentation models on unseen organs and diseases.

8 Conclusions

Despite the ground-breaking progress in perception, how to endow deep learning models with reasoning ability remains a formidable challenge for modern computer vision systems. In this regard, GNN and graph Transformers have demonstrated significant flexibility and superiority in dealing with 'relational' tasks. To this end, we have presented the first comprehensive survey of GNN and graph Transformers in computer vision from a task-oriented perspective. Specifically, a variety of classical and up-to-date algorithms are grouped into five categories according to the modality of input data, such as image, video, and point cloud. By systematically sorting out the methodologies for each task, we hope that this survey can shed light on more progress in the future. By providing discussion regarding key innovations, limitations, and potential research directions, we hope that readers can obtain new insights and go a further step towards human-like visual understanding.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[4] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[5] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE TNN*, vol. 8, 1997.

[6] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE TPAMI*, vol. 43, 2020.

[7] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," *arXiv preprint arXiv:2012.09699*, 2020.

[8] D. Chen, L. O'Bray, and K. Borgwardt, "Structure-aware transformer for graph representation learning," in *ICML*, 2022.

[9] M. Krzywda, S. Łukasikt, and A. H. Gandomi, "Graph neural networks in computer vision-architectures, datasets and common approaches," in *IEEE IJCNN*, 2022.

[10] P. Pradhyumna, G. Shreya *et al.*, "Graph neural network (gnn) in image and video understanding using deep learning for computer vision applications," in *IEEE ICESC*, 2021.

[11] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE TNNLS*, vol. 32, 2020.

[12] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, 2020.

[13] Y. Rong, T. Xu, J. Huang, W. Huang, H. Cheng, Y. Ma, Y. Wang, T. Derr, L. Wu, and T. Ma, "Deep graph learning: Foundations, advances and applications," in *KDD*, 2020.

[14] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE TNN*, vol. 20, 2008.

[15] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *ICLR*, 2014.

[16] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[17] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *NeurIPS*, vol. 29, 2016.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[19] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *IEEE TSP*, vol. 67, 2018.

[20] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *AAAI*, 2018.

[21] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," *NeurIPS*, vol. 29, 2016.

[22] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *ICML*, 2016.

[23] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *NeurIPS*, vol. 30, 2017.

[24] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[25] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *CVPR*, 2017.

[26] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive sampling towards fast graph representation learning," *NeurIPS*, vol. 31, 2018.

[27] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," *arXiv preprint arXiv:1806.03536*, 2018.

[28] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *ICCV*, 2019.

[29] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *ICLR*, 2020.

[30] G. Li, M. Müller, B. Ghanem, and V. Koltun, "Training graph neural networks with 1000 layers," in *ICML*, 2021.

[31] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *NeurIPS*, vol. 31, 2018.

[32] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang, "Graph convolutional networks with eigenpooling," in *KDD*, 2019.

[33] H. Gao and S. Ji, "Graph u-nets," in *ICML*, 2019.

[34] K. Han, Y. Wang, J. Guo, Y. Tang, and E. Wu, "Vision gnn: An image is worth graph of nodes," in *NeurIPS*, 2022.

[35] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *ICCV*, 2021.

[36] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, pp. 187–199, 2021.

[37] C. Park, Y. Jeong, M. Cho, and J. Park, "Fast point transformer," in *CVPR*, 2022.

[38] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia, "Stratified transformer for 3d point cloud segmentation," in *CVPR*, 2022.

[39] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," in *NeurIPS*, 2022.

[40] K. Lin, L. Wang, and Z. Liu, "Mesh graphormer," in *ICCV*, 2021.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, vol. 30, 2017.

[42] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *CVPR*, 2019.

[43] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, "Learning semantic-specific graph representation for multi-label image recognition," in *ICCV*, 2019.

[44] J. Lanchantin, A. Sekhon, and Y. Qi, "Neural message passing for multi-label classification," in *ECML-PKDD*, 2019.

[45] X. Wu, Q. Chen, W. Li, Y. Xiao, and B. Hu, "Adahgnn: Adaptive hypergraph neural networks for multi-label image classification," in *ACM MM*, 2020.

[46] J. Ye, J. He, X. Peng, W. Wu, and Y. Qiao, "Attention-driven dynamic graph convolutional network for multi-label image recognition," in *ECCV*, 2020.

[47] R. You, Z. Guo, L. Cui, X. Long, Y. Bao, and S. Wen, "Cross-modality attention with semantic graph embedding for multi-label classification," in *AAAI*, 2020.

[48] H. D. Nguyen, X.-S. Vu, and D.-T. Le, "Modular graph transformer networks for multi-label image classification," in *AAAI*, 2021.

[49] J. Zhao, K. Yan, Y. Zhao, X. Guo, F. Huang, and J. Li, "Transformer-based dual relation graph for multi-label image recognition," in *ICCV*, 2021.

[50] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," in *ICLR*, 2019.

[51] C. Chen, K. Li, W. Wei, J. T. Zhou, and Z. Zeng, "Hierarchical graph neural networks for few-shot learning," *TCSVT*, 2021.

[52] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *CVPR*, 2019.

[53] T. Yu, S. He, Y.-Z. Song, and T. Xiang, "Hybrid graph neural networks for few-shot learning," in *AAAI*, 2022.

[54] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *ICLR*, 2018.

[55] L. Yang, L. Li, Z. Zhang, X. Zhou, E. Zhou, and Y. Liu, "Dpgn: Distribution propagation graph network for few-shot learning," in *CVPR*, 2020.

[56] Y. Luo, Z. Huang, Z. Zhang, Z. Wang, M. Baktashmotlagh, and Y. Yang, "Learning from the past: Continual meta-learning via bayesian graph modeling," in *AAAI*, 2020.

[57] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing, "Rethinking knowledge graph propagation for zero-shot learning," in *CVPR*, 2019.

[58] G.-S. Xie, L. Liu, F. Zhu, F. Zhao, Z. Zhang, Y. Yao, J. Qin, and L. Shao, "Region graph embedding network for zero-shot learning," in *ECCV*, 2020.

[59] L. Liu, T. Zhou, G. Long, J. Jiang, and C. Zhang, "Attribute propagation network for graph zero-shot learning," in *AAAI*, 2020.

[60] L. Liu, T. Zhou, G. Long, J. Jiang, X. Dong, and C. Zhang, "Isometric propagation network for generalized zero-shot learning," in *ICLR*, 2021.

[61] S. Chen, Z. Hong, G. Xie, Q. Peng, X. You, W. Ding, and L. Shao, "Gndan: Graph navigated dual attention network for zero-shot learning," *IEEE TNNLS*, 2022.

[62] M. Mancini, M. F. Naeem, Y. Xian, and Z. Akata, "Learning graph embeddings for open world compositional zero-shot learning," *IEEE TPAMI*, 2022.

[63] X. Ma, T. Zhang, and C. Xu, "Gcan: Graph convolutional adversarial network for unsupervised domain adaptation," in *CVPR*, 2019, pp. 8266–8276.

[64] C. Chen, J. Li, X. Han, X. Liu, and Y. Yu, "Compound domain generalization via meta-knowledge encoding," in *CVPR*, 2022.

[65] S. Roy, E. Krivosheev, Z. Zhong, N. Sebe, and E. Ricci, "Curriculum graph co-teaching for multi-target domain adaptation," in *CVPR*, 2021.

[66] Y. Luo, Z. Wang, Z. Huang, and M. Baktashmotlagh, "Progressive graph learning for open-set domain adaptation," in *ICML*, 2020.

[67] Z. Wang, Y. Luo, Z. Huang, and M. Baktashmotlagh, "Prototype-matching graph network for heterogeneous domain adaptation," in *ACM MM*, 2020.

[68] H. Wang, M. Xu, B. Ni, and W. Zhang, "Learning to combine: Knowledge aggregation for multi-source domain adaptation," in *ECCV*. Springer, 2020.

[69] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *IJCV*, vol. 128, no. 2, pp. 261–318, 2020.

[70] H. Xu, C. Jiang, X. Liang, L. Lin, and Z. Li, "Reasoning-rcnn: Unifying adaptive global reasoning into large-scale object detection," in *CVPR*, 2019.

[71] H. Xu, C. Jiang, X. Liang, and Z. Li, "Spatial-aware graph relation network for large-scale object detection," in *CVPR*, 2019.

[72] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *CVPR*, 2018.

[73] C. Chi, F. Wei, and H. Hu, "Relationnet++: Bridging visual representations for object detection via transformer decoder," *NeurIPS*, vol. 33, 2020.

[74] Z. Li, X. Du, and Y. Cao, "Gar: Graph assisted reasoning for object detection," in *WACV*, 2020.

[75] G. Zhao, W. Ge, and Y. Yu, "Graphfpn: Graph feature pyramid network for object detection," in *ICCV*, 2021.

[76] C. Chen, J. Li, H.-Y. Zhou, X. Han, Y. Huang, X. Ding, and Y. Yu, "Relation matters: Foreground-aware graph-based relational reasoning for domain adaptive object detection," *IEEE TPAMI*, 2022.

[77] C. Chen, J. Li, Z. Zheng, Y. Huang, X. Ding, and Y. Yu, "Dual bipartite graph learning: A general approach for domain adaptive object detection," in *ICCV*, 2021.

[78] W. Li, X. Liu, and Y. Yuan, "Sigma: Semantic-complete graph matching for domain adaptive object detection," in *CVPR*, 2022.

[79] C. Zhu, F. Chen, U. Ahmed, Z. Shen, and M. Savvides, "Semantic relation reasoning for shot-stable few-shot object detection," in *CVPR*, 2021.

[80] L. Zhang, X. Li, A. Arnab, K. Yang, Y. Tong, and P. Torr, "Dual graph convolutional network for semantic segmentation," in *BMVC*, 2019.

[81] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *CVPR*, 2019.

[82] L. Zhang, D. Xu, A. Arnab, and P. H. Torr, "Dynamic graph message passing networks," in *CVPR*, 2020.

[83] C. Yu, Y. Liu, C. Gao, C. Shen, and N. Sang, "Representative graph neural network," in *ECCV*, 2020.

[84] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, "Spatial pyramid based graph reasoning for semantic segmentation," in *CVPR*, 2020.

[85] H. Hu, D. Ji, W. Gan, S. Bai, W. Wu, and J. Yan, "Class-wise dynamic graph convolution for semantic segmentation," in *ECCV*, 2020.

[86] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation," in *ICCV*, 2019.

[87] G.-S. Xie, J. Liu, H. Xiong, and L. Shao, "Scale-aware graph neural network for few-shot semantic segmentation," in *CVPR*, 2021.

[88] B. Zhang, J. Xiao, J. Jiao, Y. Wei, and Y. Zhao, "Affinity attention graph neural network for weakly supervised semantic segmentation," *IEEE TPAMI*, 2021.

[89] Y. Wu, G. Zhang, Y. Gao, X. Deng, K. Gong, X. Liang, and L. Lin, "Bidirectional graph reasoning network for panoptic segmentation," in *CVPR*, 2020.

[90] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018, pp. 7794–7803.

[91] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *ECCV*, 2018.

[92] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *CVPR*, 2015.

[93] G. Zhu, L. Zhang, Y. Jiang, Y. Dang, H. Hou, P. Shen, M. Feng, X. Zhao, Q. Miao, S. A. A. Shah *et al.*, "Scene graph generation: A comprehensive survey," *arXiv preprint arXiv:2201.00443*, 2022.

[94] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang, "Factorizable net: an efficient subgraph-based framework for scene graph generation," in *ECCV*, 2018.

[95] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo, "Attentive relational networks for mapping images to scene graphs," in *CVPR*, 2019.

[96] R. Li, S. Zhang, B. Wan, and X. He, "Bipartite graph network with adaptive message passing for unbiased scene graph generation," in *CVPR*, 2021.

[97] M. Suhail, A. Mittal, B. Siddiquie, C. Broaddus, J. Eledath, G. Medioni, and L. Sigal, "Energy-based learning for scene graph generation," in *CVPR*, 2021.

[98] X. Wang and A. Gupta, "Videos as space-time region graphs," in *ECCV*, 2018.

[99] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI*, 2018.

[100] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *ICCV*, 2019.

[101] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, "G-tad: Sub-graph localization for temporal action detection," in *CVPR*, 2020.

[102] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *ICCV*, 2019.

[103] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. V. Gool, "Video object segmentation with episodic graph memory networks," in *ECCV*, 2020.

[104] J. Zhang, F. Shen, X. Xu, and H. T. Shen, "Temporal reasoning graph for activity recognition," *IEEE TIP*, vol. 29, 2020.

[105] Y. Ou, L. Mi, and Z. Chen, "Object-relation reasoning graph for action recognition," in *CVPR*, 2022.

[106] J. Gao, T. Zhang, and C. Xu, "I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs," in *AAAI*, 2019.

[107] J. Zhou, K.-Y. Lin, H. Li, and W.-S. Zheng, "Graph-based high-order relation modeling for long-term action recognition," in *CVPR*, 2021.

[108] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu, "Learning actor relation graphs for group activity recognition," in *CVPR*, 2019.

[109] J.-H. Pan, J. Gao, and W.-S. Zheng, "Action assessment by joint relation graphs," in *ICCV*, 2019.

[110] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *CVPR*, 2019.

[111] B. Li, X. Li, Z. Zhang, and F. Wu, "Spatio-temporal graph routing for skeleton-based action recognition," in *AAAI*, 2019.

[112] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *CVPR*, 2019.

[113] R. Zhao, K. Wang, H. Su, and Q. Ji, "Bayesian graph convolution lstm for skeleton based action recognition," in *ICCV*, 2019.

[114] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, 1997.

[115] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *CVPR*, 2019.

[116] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *CVPR*, 2020.

[117] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *CVPR*, 2019.

[118] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *CVPR*, 2020.

[119] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling gcn with dropgraph module for skeleton-based action recognition," in *ECCV*, 2020.

[120] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao, N. Golmant, A. Gholaminejad, J. Gonzalez, and K. Keutzer, "Shift: A zero flop, zero parameter alternative to spatial convolutions," in *CVPR*, 2018.

[121] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *ECCV*, 2016.

[122] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *CVPR*, 2016.

[123] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[124] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM TOG*, vol. 38, 2019.

[125] C. Zhao, A. K. Thabet, and B. Ghanem, "Video self-stitching graph network for temporal action localization," in *ICCV*, 2021.

[126] Y. Bai, Y. Wang, Y. Tong, Y. Yang, Q. Liu, and J. Liu, "Boundary content graph neural network for temporal action proposal generation," in *ECCV*, 2020.

[127] Y. Huang, Y. Sugano, and Y. Sato, "Improving action segmentation via graph-based temporal reasoning," in *CVPR*, 2020.

[128] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," in *CVPR*, 2020.

[129] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning," in *CVPR*, 2020.

[130] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, "Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction," in *CVPR*, 2020.

[131] T. Sofianos, A. Sampieri, L. Franco, and F. Galasso, "Space-time-separable graph convolutional network for pose forecasting," in *ICCV*, 2021.

[132] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in *ICCV*, 2019.

[133] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *CVPR*, 2020.

[134] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *ECCV*, 2020.

[135] J. Sun, Q. Jiang, and C. Lu, "Recursive social behavior graph for trajectory prediction," in *CVPR*, 2020.

[136] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua, "Sgcn: Sparse graph convolution network for pedestrian trajectory prediction," in *CVPR*, 2021.

[137] L. Li, M. Pagnucco, and Y. Song, "Graph-based spatial transformer with memory replay for multi-future pedestrian trajectory prediction," in *CVPR*, 2022.

[138] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *ICCV*, 2015.

[139] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "Referitgame: Referring to objects in photographs of natural scenes," in *EMNLP*, 2014.

[140] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *ECCV*, 2010.

[141] D. Teney, L. Liu, and A. van Den Hengel, "Graph-structured representations for visual question answering," in *CVPR*, 2017.

[142] M. Bajaj, L. Wang, and L. Sigal, "G3raphground: Graph-based language grounding," in *ICCV*, 2019.

[143] S. Yang, G. Li, and Y. Yu, "Cross-modal relationship inference for grounding referring expressions," in *CVPR*, 2019.

[144] M.-C. De Marneffe and C. D. Manning, "The stanford typed dependencies representation," in *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, 2008.

[145] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014, pp. 1724–1734.

[146] W. Norcliffe-Brown, S. Vafeias, and S. Parisot, "Learning conditioned graph structures for interpretable visual question answering," *NeurIPS*, vol. 31, 2018.

[147] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko, "Language-conditioned graph networks for relational reasoning," in *ICCV*, 2019.

[148] C. Jing, Y. Jia, Y. Wu, X. Liu, and Q. Wu, "Maintaining reasoning consistency in compositional visual question answering," in *CVPR*, 2022.

[149] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-aware graph attention network for visual question answering," in *ICCV*, 2019.

[150] E.-S. Kim, W. Y. Kang, K.-W. On, Y.-J. Heo, and B.-T. Zhang, "Hypergraph attention networks for multimodal learning," in *CVPR*, 2020.

[151] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, 2017.

[152] R. Saqur and K. Narasimhan, "Multimodal graph networks for compositional generalization in visual question answering," *NeurIPS*, vol. 33, 2020.

[153] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[154] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *CVPR*, 2017, pp. 6904–6913.

[155] D. A. Hudson and C. D. Manning, "Gqa: a new dataset for compositional question answering over real-world images," *arXiv preprint arXiv:1902.09506*, vol. 3, 2019.

[156] D. Gao, K. Li, R. Wang, S. Shan, and X. Chen, "Multi-modal graph neural network for joint reasoning on vision and scene text," in *CVPR*, 2020.

[157] A. Wu, L. Zhu, Y. Han, and Y. Yang, "Connective cognition network for directional visual commonsense reasoning," *NeurIPS*, vol. 32, 2019.

[158] W. Yu, J. Zhou, W. Yu, X. Liang, and N. Xiao, "Heterogeneous graph learning for visual commonsense reasoning," *NeurIPS*, vol. 32, 2019.

[159] D. Teney, P. Anderson, X. He, and A. Van Den Hengel, "Tips and tricks for visual question answering: Learnings from the 2017 challenge," in *CVPR Challenge*, 2018.

[160] P. Wang, Q. Wu, C. Shen, A. Dick, and A. Van Den Hengel, "Fvqa: Fact-based visual question answering," *IEEE TPAMI*, vol. 40, 2017.

[161] M. Narasimhan, S. Lazebnik, and A. Schwing, "Out of the box: Reasoning with graph convolution nets for factual visual question answering," *NeurIPS*, vol. 31, 2018.

[162] A. K. Singh, A. Mishra, S. Shekhar, and A. Chakraborty, "From strings to things: Knowledge-enabled vqa model that can read and reason," in *ICCV*, 2019.

[163] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[164] K. Marino, X. Chen, D. Parikh, A. Gupta, and M. Rohrbach, "Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa," in *CVPR*, 2021.

[165] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *ECCV*, 2018.

[166] S. Yang, G. Li, and Y. Yu, "Dynamic graph attention for referring expression comprehension," in *ICCV*, 2019.

[167] ——, "Relationship-embedded representation learning for grounding referring expressions," *IEEE TPAMI*, vol. 43, 2020.

[168] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel, "Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks," in *CVPR*, 2019.

[169] S. Chen and B. Li, "Multi-modal dynamic graph transformer for visual grounding," in *CVPR*, 2022.

[170] S. Huang, T. Hui, S. Liu, G. Li, Y. Wei, J. Han, L. Liu, and B. Li, "Referring image segmentation via cross-modal progressive comprehension," in *CVPR*, 2020.

[171] T. Hui, S. Liu, S. Huang, G. Li, S. Yu, F. Zhang, and J. Han, "Linguistic structure guided context modeling for referring image segmentation," in *ECCV*, 2020.

[172] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. Guibas, "Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes," in *ECCV*, 2020.

[173] S. Yang, G. Li, and Y. Yu, "Propagating over phrase relations for one-stage visual grounding," in *ECCV*, 2020.

[174] Y. Liu, B. Wan, L. Ma, and X. He, "Relation-aware instance refinement for weakly supervised visual grounding," in *CVPR*, 2021.

[175] Y. Liu, B. Wan, X. Zhu, and X. He, "Learning cross-modal context graph for visual grounding," in *AAAI*, 2020.

[176] Z. Mu, S. Tang, J. Tan, Q. Yu, and Y. Zhuang, "Disentangled motif-aware graph learning for phrase grounding," in *AAAI*, 2021.

[177] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *CVPR*, 2018.

[178] Z. Zhang, Z. Zhao, Y. Zhao, Q. Wang, H. Liu, and L. Gao, "Where does it exist: Spatio-temporal video grounding for multi-form sentences," in *CVPR*, 2020.

[179] Y. Zhao, Z. Zhao, Z. Zhang, and Z. Lin, "Cascaded prediction network via segment tree for temporal video grounding," in *CVPR*, 2021.

[180] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *ECCV*, 2018, pp. 684–699.

[181] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018.

[182] T. Yao, Y. Pan, Y. Li, and T. Mei, "Hierarchy parsing for image captioning," in *ICCV*, 2019.

[183] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *CVPR*, 2019.

[184] Y. Zhong, L. Wang, J. Chen, D. Yu, and Y. Li, "Comprehensive image captioning via scene graph decomposition," in *ECCV*, 2020.

[185] S. Chen, Q. Jin, P. Wang, and Q. Wu, "Say as you wish: Fine-grained control of image caption generation with abstract scene graphs," in *CVPR*, 2020.

[186] K. Nguyen, S. Tripathi, B. Du, T. Guha, and T. Q. Nguyen, "In defense of scene graphs for image captioning," in *ICCV*, 2021.

[187] O. Ulutan, A. Iftekhar, and B. S. Manjunath, "Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions," in *CVPR*, 2020.

[188] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, and Z.-J. Zha, "Object relational graph with teacher-recommended learning for video captioning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 278–13 288.

[189] B. Pan, H. Cai, D.-A. Huang, K.-H. Lee, A. Gaidon, E. Adeli, and J. C. Niebles, "Spatio-temporal graph for video captioning with knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 870–10 879.

[190] S. Chen and Y.-G. Jiang, "Motion guided region message passing for video captioning," in *ICCV*, 2021.

[191] Y. Huang and L. Wang, "Acmm: Aligned cross-modal memory for few-shot image and sentence matching," in *ICCV*, 2019.

[192] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Visual semantic reasoning for image-text matching," in *ICCV*, 2019.

[193] S. Yan, L. Yu, and Y. Xie, "Discrete-continuous action space policy gradient-based attention for image-text matching," in *CVPR*, 2021.

[194] H. Wang, Y. Zhang, Z. Ji, Y. Pang, and L. Ma, "Consensus-aware visual-semantic embedding for image-text matching," in *ECCV*, 2020.

[195] C. Liu, Z. Mao, T. Zhang, H. Xie, B. Wang, and Y. Zhang, "Graph structured network for image-text matching," in *CVPR*, 2020.

[196] Y. Li, D. Zhang, and Y. Mu, "Visual-semantic matching by exploring high-order attention and distraction," in *CVPR*, 2020.

[197] S. Chen, Y. Zhao, Q. Jin, and Q. Wu, "Fine-grained video-text retrieval with hierarchical graph reasoning," in *CVPR*, 2020.

[198] Y. Zeng, D. Cao, X. Wei, M. Liu, Z. Zhao, and Z. Qin, "Multi-modal relational graph for cross-modal video moment retrieval," in *CVPR*, 2021.

[199] J. Li, S. Tang, L. Zhu, H. Shi, X. Huang, F. Wu, Y. Yang, and Y. Zhuang, "Adaptive hierarchical graph reasoning with semantic coherence for video-and-language inference," in *ICCV*, 2021.

[200] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR*, 2018.

[201] Z. Deng, K. Narasimhan, and O. Russakovsky, "Evolving graphical planner: Contextual global planning for vision-and-language navigation," *NeurIPS*, vol. 33, 2020.

[202] K. Chen, J. K. Chen, J. Chuang, M. Vázquez, and S. Savarese, "Topological planning with transformers for vision-and-language navigation," in *CVPR*, 2021.

[203] C. Gao, J. Chen, S. Liu, L. Wang, Q. Zhang, and Q. Wu, "Room-and-object aware knowledge reasoning for remote embodied referring expression," in *CVPR*, 2021.

[204] J. Chen, C. Gao, E. Meng, Q. Zhang, and S. Liu, "Reinforced structured state-evolution for vision-language navigation," in *CVPR*, 2022.

[205] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Think global, act local: Dual-scale graph transformer for vision-and-language navigation," in *CVPR*, 2022.

[206] J. Park, J. Lee, and K. Sohn, "Bridge to answer: Structure-aware graph interaction network for video question answering," in *CVPR*, 2021.

[207] W. Yu, H. Zheng, M. Li, L. Ji, L. Wu, N. Xiao, and N. Duan, "Learning from inside: Self-driven siamese sampling and reasoning for video question answering," *NeurIPS*, vol. 34, 2021.

[208] F. Liu, J. Liu, W. Wang, and H. Lu, "Hair: Hierarchical visual-semantic relational reasoning for video question answering," in *ICCV*, 2021.

[209] D. Huang, P. Chen, R. Zeng, Q. Du, M. Tan, and C. Gan, "Location-aware graph convolutional networks for video question answering," in *AAAI*, 2020.

[210] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," *NeurIPS*, vol. 28, 2015.

[211] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017.

[212] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *NeurIPS*, vol. 30, 2017.

[213] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *NeurIPS*, 2018.

[214] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *ICCV*, 2019.

[215] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *CVPR*, 2019.

[216] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *CVPR*, 2017.

[217] H. Zhou, Y. Feng, M. Fang, M. Wei, J. Qin, and T. Lu, "Adaptive graph convolution for point cloud analysis," in *ICCV*, 2021.

[218] K. Zhang, M. Hao, J. Wang, X. Chen, Y. Leng, C. W. de Silva, and C. Fu, "Linked dynamic graph cnn: Learning through point cloud by linking hierarchical features," in *2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, 2021, pp. 7–12.

[219] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *CVPR*, 2018.

[220] K. Hassani and M. Haley, "Unsupervised multi-task feature learning on point clouds," in *ICCV*, 2019.

[221] C. Chen, G. Li, R. Xu, T. Chen, M. Wang, and L. Lin, "Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis," in *CVPR*, 2019.

[222] Z.-H. Lin, S.-Y. Huang, and Y.-C. F. Wang, "Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis," in *CVPR*, 2020.

[223] H. Lei, N. Akhtar, and A. Mian, "Spherical kernel for efficient graph convolution on 3d point clouds," *IEEE TPAMI*, vol. 43, 2020.

[224] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: Learning curves for point clouds shape analysis," in *ICCV*, 2021.

[225] L. Yi, H. Su, X. Guo, and L. J. Guibas, "Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation," in *CVPR*, 2017.

[226] G. Te, W. Hu, A. Zheng, and Z. Guo, "Rgcnn: Regularized graph cnn for point cloud segmentation," in *ACM MM*, 2018.

[227] G. Pan, J. Wang, R. Ying, and P. Liu, "3dti-net: Learn inner transform invariant 3d geometry features using dynamic gcn," *arXiv preprint arXiv:1812.06254*, 2018.

[228] Y. Zhang and M. Rabbat, "A graph-cnn for 3d point cloud classification," in *ICASSP*, 2018.

[229] C. Wang, B. Samari, and K. Siddiqi, "Local spectral graph convolution for point set feature learning," in *ECCV*, 2018.

[230] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *AAAI*, vol. 33, 2019.

[231] M. Lin and A. Feragen, "diffconv: Analyzing irregular point clouds with an irregular view," in *ECCV*, 2022.

[232] R. Wiersma, A. Nasikun, E. Eisemann, and K. Hildebrandt, "Deltaconv: Anisotropic operators for geometric deep learning on point clouds," *ACM TOG*, 2022.

[233] C. Chen, L. Z. Fragonara, and A. Tsourdos, "Gapnet: Graph attention based point neural network for exploiting local feature of point cloud," *CoRR*, vol. abs/1905.08705, 2019.

[234] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *CVPR*, 2019.

[235] L. Xun, X. Feng, C. Chen, X. Yuan, and Q. Lu, "Graph attention-based deep neural network for 3d point cloud processing," in *ICME*, 2021.

[236] C.-Q. Huang, F. Jiang, Q.-H. Huang, X.-Z. Wang, Z.-M. Han, and W.-Y. Huang, "Dual-graph attention convolution network for 3-d point cloud classification," *TNNLS*, 2022.

[237] P. Gomes, S. Rossi, and L. Toni, "Spatio-temporal graph-rnn for point cloud prediction," in *ICIP*. IEEE, 2021.

[238] D. Lu, Q. Xie, K. Gao, L. Xu, and J. Li, "3dctn: 3d convolution-transformer network for point cloud classification," *IEEE TITS*, 2022.

[239] J. Liu, B. Ni, C. Li, J. Yang, and Q. Tian, "Dynamic points agglomeration for hierarchical point sets learning," in *ICCV*, 2019.

[240] Q. Xu, X. Sun, C.-Y. Wu, P. Wang, and U. Neumann, "Grid-gcn for fast and scalable point cloud learning," in *CVPR*, 2020.

[241] J.-F. Zhang and Z. Zhang, "Point-x: A spatial-locality-aware architecture for energy-efficient graph-based point-cloud deep learning," in *IEEE/ACM International Symposium on Microarchitecture*, 2021.

[242] Y. Li, H. Chen, Z. Cui, R. Timofte, M. Pollefeys, G. Chirikjian, and L. Van Gool, "Towards efficient graph convolutional networks for point cloud handling," in *ICCV*, 2021.

[243] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *NeurIPS*, vol. 31, 2018.

[244] S. S. Mohammadi, Y. Wang, and A. Del Bue, "Pointview-gcn: 3d shape classification with multi-view point clouds," in *ICIP*. IEEE, 2021, pp. 3103–3107.

[245] X. Yan, H. Zhan, C. Zheng, J. Gao, R. Zhang, S. Cui, and Z. Li, "Let images give you more: Point cloud cross-modal training for shape analysis," *NeurIPS*, 2022.

[246] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, 2019.

[247] N. Verma, E. Boyer, and J. Verbeek, "Feastnet: Feature-steered graph convolutions for 3d shape analysis," in *CVPR*, 2018.

[248] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "Meshcnn: a network with an edge," *ACM TOG*, vol. 38, 2019.

[249] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "Meshnet: Mesh neural network for 3d shape representation," in *AAAI*, 2019.

[250] J. Schult, F. Engelmann, T. Kontogianni, and B. Leibe, "Dualconvmesh-net: Joint geodesic and euclidean convolutions on 3d meshes," in *CVPR*, 2020.

[251] F. Milano, A. Loquercio, A. Rosinol, D. Scaramuzza, and L. Carlone, "Primal-dual mesh convolutional neural networks," *NeurIPS*, vol. 33, 2020.

[252] S. Gong, L. Chen, M. Bronstein, and S. Zafeiriou, "Spiralnet++: A fast and highly efficient mesh convolution operator," in *ICCV Workshops*, 2019.

[253] G. Bouritsas, S. Bokhnyak, S. Ploumpis, M. Bronstein, and S. Zafeiriou, "Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation," in *ICCV*, 2019.

[254] V. V. Singh, S. V. Sheshappanavar, and C. Kambhamettu, "Mesh classification with dilated mesh convolutions," in *ICIP*, 2021.

[255] M. Yavartanoo, S.-H. Hung, R. Neshatavar, Y. Zhang, and K. M. Lee, "Polynet: Polynomial neural network for 3d shape recognition with polyshape representation," in *3DV*. IEEE, 2021.

[256] W. He, Z. Jiang, C. Zhang, and A. M. Sainju, "Curvanet: Geometric deep learning based on directional curvature for 3d shape analysis," in *KDD*, 2020.

[257] L. He, Y. Dong, Y. Wang, D. Tao, and Z. Lin, "Gauge equivariant transformer," in *NeurIPS*, 2021.

[258] S.-M. Hu, Z.-N. Liu, M.-H. Guo, J.-X. Cai, J. Huang, T.-J. Mu, and R. R. Martin, "Subdivision-based mesh convolution networks," *TOG*, vol. 41, 2022.

[259] Q. Dong, Z. Wang, J. Gao, S. Chen, Z. Shu, and S. Xin, "Laplacian2mesh: Laplacian-based mesh understanding," *arXiv preprint arXiv:2202.00307*, 2022.

[260] A. Lahav and A. Tal, "Meshwalker: Deep mesh understanding by random walks," *TOG*, vol. 39, 2020.

[261] R. B. Izhak, A. Lahav, and A. Tal, "Attwalk: Attentive cross-walks for deep mesh analysis," in *WACV*, 2022.

[262] L. Loic and B. Mohamed, "Point cloud oversegmentation with graph-structured deep metric learning," in *CVPR*, 2019.

[263] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia, "Hierarchical point-edge interaction network for point cloud semantic segmentation," in *ICCV*, 2019.

[264] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," *NeurIPS*, 2022.

[265] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3d graph neural networks for rgbd semantic segmentation," in *ICCV*, 2017.

[266] L. Loic and S. Martin, "Large-scale point cloud semantic segmentation with superpoint graphs," in *CVPR*, 2018.

[267] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *CVPR*, 2019.

[268] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *CVPR*, 2019.

[269] K. Zhiheng and L. Ning, "Pyramnet: Point cloud pyramid attention network and graph embedding module for classification and segmentation," *arXiv preprint arXiv:1906.03299*, 2019.

[270] Z. Liang, M. Yang, L. Deng, C. Wang, and B. Wang, "Hierarchical depthwise graph convolutional neural network for 3d semantic segmentation of point clouds," in *ICRA*, 2019.

[271] H. Lei, N. Akhtar, and A. Mian, "Seggcn: Efficient 3d point cloud segmentation with fuzzy spherical kernel," in *CVPR*, 2020.

[272] Q. Xu, X. Sun, C.-Y. Wu, P. Wang, and U. Neumann, "Grid-gcn for fast and scalable point cloud learning," in *CVPR*, 2020.

[273] W. Han, C. Wen, C. Wang, X. Li, and Q. Li, "Point2node: Correlation learning of dynamic-node for point cloud feature modeling," in *AAAI*, 2020.

[274] A. Tao, Y. Duan, Y. Wei, J. Lu, and J. Zhou, "SegGroup: Seg-level supervision for 3D instance and semantic segmentation," *IEEE TIP*, 2022.

[275] F. Engelmann, T. Kontogianni, and B. Leibe, "Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds," in *ICRA*, 2020.

[276] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *ECCV*, 2018.

[277] S. Wang, S. Suo, W. Ma, A. Pokrovsky, and R. Urtasun, "Deep parametric continuous convolutional neural networks," in *CVPR*, 2018.

[278] W. Shi and R. R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *CVPR*, 2020.

[279] J. Chen, B. Lei, Q. Song, H. Ying, D. Z. Chen, and J. Wu, "A hierarchical graph network for 3d object detection on point clouds," in *CVPR*, 2020.

[280] M. Feng, S. Z. Gilani, Y. Wang, L. Zhang, and A. Mian, "Relation graph network for 3d object detection in point clouds," *TIP*, vol. 30, 2020.

[281] Y. Wang and J. M. Solomon, "Object dgcnn: 3d object detection using dynamic graphs," in *NeurIPS*, 2021.

[282] J. Zarzar, S. Giancola, and B. Ghanem, "Pointrgcn: Graph convolution networks for 3d vehicles detection refinement," *arXiv preprint arxiv:1911.12236*, 2019.

[283] Z. Fan, L. Zhu, H. Li, X. Chen, S. Zhu, and P. Tan, "Floorplancad: a large-scale cad drawing dataset for panoptic symbol spotting," in *ICCV*, 2021.

[284] Z. Fan, T. Chen, P. Wang, and Z. Wang, "Cadtransformer: Panoptic symbol spotting transformer for cad drawings," in *CVPR*, 2022.

[285] L. Pan, "Ecg: Edge-aware point cloud completion with graph convolution," *IEEE RA-L*, vol. 5, 2020.

[286] L. Zhu, B. Wang, G. Tian, W. Wang, and C. Li, "Towards point cloud completion: Point rank sampling and cross-cascade graph cnn," *Neurocomputing*, vol. 461, 2021.

[287] J. Shi, L. Xu, L. Heng, and S. Shen, "Graph-guided deformation for point cloud completion," *IEEE RA-L*, vol. 6, 2021.

[288] W. Yifan, S. Wu, H. Huang, D. Cohen-Or, and O. Sorkine-Hornung, "Patch-based progressive 3d point set upsampling," in *CVPR*, 2019.

[289] G. Qian, A. Abualshour, G. Li, A. Thabet, and B. Ghanem, "Pu-gcn: Point cloud upsampling using graph convolutional networks," in *CVPR*, 2021.

[290] H. Wu, J. Zhang, and K. Huang, "Point cloud super resolution with adversarial residual graph networks," *arXiv preprint arXiv:1908.02111*, 2019.

[291] Y. Qian, J. Hou, S. Kwong, and Y. He, "Pugeo-net: A geometry-centric network for 3d point cloud upsampling," in *ECCV*, 2020.

[292] F. Pistilli, G. Fracastoro, D. Valsesia, and E. Magli, "Learning graph-convolutional representations for point cloud denoising," in *ECCV*, 2020.

[293] S. Luo and W. Hu, "Differentiable manifold reconstruction for point cloud denoising," in *ACM MM*, 2020.

[294] M. Armando, J.-S. Franco, and E. Boyer, "Mesh denoising with facet graph convolutions," *IEEE TVCG*, 2020.

[295] Y. Shen, H. Fu, Z. Du, X. Chen, E. Burnaev, D. Zorin, K. Zhou, and Y. Zheng, "Gcn-denoiser: Mesh denoising with graph convolutional networks," *ACM TOG*, vol. 41, 2022.

[296] Y. Zhang, G. Shen, Q. Wang, Y. Qian, M. Wei, and J. Qin, "Geobi-gnn: Geometry-aware bi-domain mesh denoising via graph neural networks," *Computer-Aided Design*, vol. 144, 2022.

[297] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *CVPR*, 2018.

[298] K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. J. Mitra, and L. J. Guibas, "Structurenet: hierarchical graph networks for 3d shape generation," *ACM TOG*, vol. 38, 2019.

[299] D. Valsesia, G. Fracastoro, and E. Magli, "Learning localized generative models for 3d point clouds via graph convolution," in *ICLR*, 2018.

[300] D. W. Shu, S. W. Park, and J. Kwon, "3d point cloud generative adversarial network based on tree structured graph convolutions," in *ICCV*, 2019.

[301] Y. Li and G. Baciu, "Hsgan: Hierarchical graph learning for point cloud generation," *IEEE TIP*, vol. 30, 2021.

[302] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," in *ECCV*, 2018.

[303] C. Wen, Y. Zhang, Z. Li, and Y. Fu, "Pixel2mesh++: Multi-view 3d mesh generation via deformation," in *ICCV*, 2019.

[304] G. Gkioxari, J. Malik, and J. Johnson, "Mesh r-cnn," in *ICCV*, 2019.

[305] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[306] U. Wickramasinghe, E. Remelli, G. Knott, and P. Fua, "Voxel2mesh: 3d mesh model generation from volumetric data," in *MICCAI*, 2020.

[307] H. Dhamo, F. Manhardt, N. Navab, and F. Tombari, "Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs," in *ICCV*, 2021.

[308] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *CVPR*, 2019.

[309] X. Yu, J. van Baar, and S. Chen, "Joint 3d human shape recovery and pose estimation from a single image with bilayer graph," in *3DV*, 2021.

[310] S. Zhou, M. Jiang, S. Cai, and Y. Lei, "Dc-gnet: Deep mesh relation capturing graph convolution network for 3d human shape reconstruction," in *ACM MM*, 2021.

[311] H. Choi, G. Moon, and K. M. Lee, "Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose," in *ECCV*, 2020.

[312] C. Zheng, M. Mendieta, P. Wang, A. Lu, and C. Chen, "A lightweight graph transformer network for human mesh reconstruction from 2d human pose," *arXiv preprint arXiv:2111.12696*, 2021.

[313] J. Lin, Y. Yuan, T. Shao, and K. Zhou, "Towards high-fidelity 3d face reconstruction from in-the-wild images using graph convolutional networks," in *CVPR*, 2020.

[314] S. Cheng, G. Tzimiropoulos, J. Shen, and M. Pantic, "Faster, better and more detailed: 3d face reconstruction with graph convolutional networks," in *ACCV*, 2020.

[315] M. Li, L. An, H. Zhang, L. Wu, F. Chen, T. Yu, and Y. Liu, "Interacting attention graph for single image two-hand reconstruction," in *CVPR*, 2022.

[316] A. S. Heinsfeld, A. R. Franco, R. C. Craddock, A. Buchweitz, and F. Meneguzzi, "Identification of autism spectrum disorder using deep learning and the abide dataset," *NeuroImage: Clinical*, vol. 17, 2018.

[317] S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. G. Moreno, B. Glocker, and D. Rueckert, "Spectral graph convolutions for population-based disease prediction," in *MICCAI*, 2017.

[318] R. Anirudh and J. J. Thiagarajan, "Bootstrapping graph convolutional neural networks for autism spectrum disorder classification," in *ICASSP*, 2019.

[319] Z. Rakhimberdina and T. Murata, "Linear graph convolutional model for diagnosing brain disorders," in *International Conference on Complex Networks and Their Applications*, 2019.

[320] H. Jiang, P. Cao, M. Xu, J. Yang, and O. Zaiane, "Hi-gcn: a hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction," *Computers in Biology and Medicine*, vol. 127, 2020.

[321] L. Peng, N. Wang, N. Dvornek, X. Zhu, and X. Li, "FedNI: Federated graph learning with network inpainting for population-based disease prediction," *IEEE TMI*, 2022.

[322] S. I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, and D. Rueckert, "Distance metric learning using graph convolutional networks: Application to functional brain networks," in *MICCAI*, 2017.

[323] D. Yao, M. Liu, M. Wang, C. Lian, J. Wei, L. Sun, J. Sui, and D. Shen, "Triplet graph convolutional network for multi-scale analysis of functional connectivity using functional mri," in *International Workshop on Graph Learning in Medical Imaging*. Springer, 2019, pp. 70–78.

[324] X. Li, Y. Zhou, N. C. Dvornek, M. Zhang, J. Zhuang, P. Ventola, and J. S. Duncan, "Pooling regularized graph neural network for fmri biomarker analysis," in *MICCAI*, 2020.

[325] X. Li, Y. Zhou, N. Dvornek, M. Zhang, S. Gao, J. Zhuang, D. Scheinost, L. H. Staib, P. Ventola, and J. S. Duncan, "BrainGNN: Interpretable brain graph neural network for fmri analysis," *Medical Image Analysis*, vol. 74, 2021.

[326] A. Kazi, S. Shekarforoush, S. Arvind Krishna, H. Burwinkel, G. Vivar, K. Kortüm, S.-A. Ahmadi, S. Albarqouni, and N. Navab, "Inceptiongcn: receptive field aware graph convolutional network for disease prediction," in *IPMI*, 2019.

[327] D. Yao, J. Sui, E. Yang, P.-T. Yap, D. Shen, and M. Liu, "Temporal-adaptive graph convolutional network for automated identification of major depressive disorder using resting-state fmri," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2020, pp. 1–10.

[328] D. Yao, E. Yang, H. Guan, J. Sui, Z. Zhang, and M. Liu, "Tensor-based multi-index representation learning for major depression disorder detection with resting-state fmri," in *MICCAI*, 2021.

[329] H. Yang, X. Li, Y. Wu, S. Li, S. Lu, J. S. Duncan, J. C. Gee, and S. Gu, "Interpretable multimodality embedding of cerebral cortex using attention graph network for identifying bipolar disorder," in *MICCAI*, 2019.

[330] Y. Zhang, L. Tetrel, B. Thirion, and P. Bellec, "Functional annotation of human cognitive states using deep graph convolution," *NeuroImage*, vol. 231, 2021.

[331] S. Gadgil, Q. Zhao, A. Pfefferbaum, E. V. Sullivan, E. Adeli, and K. M. Pohl, "Spatio-temporal graph convolution for resting-state fmri analysis," in *MICCAI*, 2020.

[332] B.-H. Kim, J. C. Ye, and J.-J. Kim, "Learning dynamic graph representation of brain connectome with spatio-temporal attention," *NeurIPS*, vol. 34, 2021.

[333] S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker, and D. Rueckert, "Disease prediction using graph convolutional networks: application to autism spectrum disorder and alzheimer's disease," *Medical image analysis*, vol. 48, 2018.

[334] X. Zhao, F. Zhou, L. Ou-Yang, T. Wang, and B. Lei, "Graph convolutional network analysis for mild cognitive impairment prediction," in *ISBI*, 2019.

[335] J. Liu, G. Tan, W. Lan, and J. Wang, "Identification of early mild cognitive impairment using multi-modal data and graph convolutional networks," *BMC bioinformatics*, vol. 21, 2020.

[336] S. Yu, S. Wang, X. Xiao, J. Cao, G. Yue, D. Liu, T. Wang, Y. Xu, and B. Lei, "Multi-scale enhanced graph convolutional network for early mild cognitive impairment detection," in *MICCAI*, 2020.

[337] Y. Huang and A. Chung, "Edge-variational graph convolutional networks for uncertainty-aware disease prediction," in *MICCAI*, 2020.

[338] J. Ma, X. Zhu, D. Yang, J. Chen, and G. Wu, "Attention-guided deep graph neural network for longitudinal alzheimer's disease analysis," in *MICCAI*, 2020.

[339] C.-Y. Wee, C. Liu, A. Lee, J. S. Poh, H. Ji, A. Qiu, A. D. N. Initiative *et al.*, "Cortical graph neural network for ad and mci diagnosis and transfer learning across populations," *NeuroImage: Clinical*, vol. 23, 2019.

[340] X. Zhang, L. He, K. Chen, Y. Luo, J. Zhou, and F. Wang, "Multi-view graph convolutional network and its applications on neuroimage analysis for parkinson's disease," in *AMIA Annual Symposium Proceedings*, 2018.

[341] A. Kazi, S. Shekarforoush, S. Arvind Krishna, H. Burwinkel, G. Vivar, B. Wiestler, K. Kortüm, S.-A. Ahmadi, S. Albarqouni, and N. Navab, "Graph convolution based attention model for personalized disease prediction," in *MICCAI*, 2019.

[342] W. Zhang, L. Zhan, P. Thompson, and Y. Wang, "Deep representation learning for multimodal brain networks," in *MICCAI*, 2020.

[343] X. Yu, S. Lu, L. Guo, S.-H. Wang, and Y.-D. Zhang, "Resgnet-c: A graph convolutional neural network for detection of covid-19," *Neurocomputing*, vol. 452, 2021.

[344] S.-H. Wang, V. V. Govindaraj, J. M. Górriz, X. Zhang, and Y.-D. Zhang, "Covid-19 classification by fgcnet with deep feature fusion from graph convolutional network and convolutional neural network," *Information Fusion*, vol. 67, 2021.

[345] Y. Zhang, X. Wang, Z. Xu, Q. Yu, A. Yuille, and D. Xu, "When radiology report generation meets knowledge graph," in *AAAI*, 2020.

[346] D. Hou, Z. Zhao, and S. Hu, "Multi-label learning with visual-semantic embedded knowledge graph for diagnosis of radiology imaging," *IEEE Access*, vol. 9, 2021.

[347] Y. Liu, F. Zhang, Q. Zhang, S. Wang, Y. Wang, and Y. Yu, "Cross-view correspondence reasoning based on bipartite graph convolutional network for mammogram mass detection," in *CVPR*, 2020.

[348] Y. Liu, F. Zhang, C. Chen, S. Wang, Y. Wang, and Y. Yu, "Act like a radiologist: Towards reliable multi-view correspondence reasoning for mammogram mass detection," *IEEE TPAMI*, 2021.

[349] J. Lian, J. Liu, S. Zhang, K. Gao, X. Liu, D. Zhang, and Y. Yu, "A structure-aware relation network for thoracic diseases detection and segmentation," *IEEE TMI*, vol. 40, 2021.

[350] Z. Chen, J. Liu, M. Zhu, P. Y. Woo, and Y. Yuan, "Instance importance-aware graph convolutional network for 3d medical diagnosis," *Medical Image Analysis*, vol. 78, 2022.

[351] G. Zhao, Q. Feng, C. Chen, Z. Zhou, and Y. Yu, "Diagnose like a radiologist: Hybrid neuro-probabilistic reasoning for attribute-based medical image diagnosis," *IEEE TPAMI*, 2022.

[352] G. Cucurull, K. Wagstyl, A. Casanova, P. Veličković, E. Jakobsen, M. Drozdzal, A. Romero, A. Evans, and Y. Bengio, "Convolutional neural networks for mesh-based parcellation of the cerebral cortex," in *Proceedings of the Medical Imaging with Deep Learning*, 2018.

[353] K. Gopinath, C. Desrosiers, and H. Lombaert, "Adaptive graph convolution pooling for brain surface analysis," in *IPMI*, 2019.

[354] Z. Wu, F. Zhao, J. Xia, L. Wang, W. Lin, J. H. Gilmore, G. Li, and D. Shen, "Intrinsic patch-based cortical anatomical parcellation using graph convolutional neural network on surface manifold," in *MICCAI*, 2019.

[355] R. He, K. Gopinath, C. Desrosiers, and H. Lombaert, "Spectral graph transformer networks for brain surface parcellation," in *ISBI*, 2020.

[356] J. M. Wolterink, T. Leiner, and I. Išgum, "Graph convolutional networks for coronary artery segmentation in cardiac ct angiography," in *International Workshop on Graph Learning in Medical Imaging*, 2019.

[357] Z. Zhai, M. Staring, X. Zhou, Q. Xie, X. Xiao, M. Els Bakker, L. J. Kroft, B. P. Lelieveldt, G. J. Boon, F. A. Klok *et al.*, "Linking convolutional neural networks with graph convolutional networks: application in pulmonary artery-vein separation," in *International Workshop on Graph Learning in Medical Imaging*, 2019.

[358] S. Y. Shin, S. Lee, I. D. Yun, and K. M. Lee, "Deep vessel segmentation by learning graphical connectivity," *Medical image analysis*, vol. 58, 2019.

[359] K. J. Noh, S. J. Park, and S. Lee, "Combining fundus images and fluorescein angiography for artery/vein classification using the hierarchical vessel graph network," in *MICCAI*, 2020.

[360] L. Chen, T. Hatsukami, J.-N. Hwang, and C. Yuan, "Automated intracranial artery labeling using a graph neural network and hierarchical refinement," in *MICCAI*, 2020.

[361] L. Yao, P. Jiang, Z. Xue, Y. Zhan, D. Wu, L. Zhang, Q. Wang, F. Shi, and D. Shen, "Graph convolutional network based point cloud for head and neck vessel labeling," in *International Workshop on Machine Learning in Medical Imaging*, 2020.

[362] H. Yang, X. Zhen, Y. Chi, L. Zhang, and X.-S. Hua, "Cpr-gcn: Conditional partial-residual graph convolutional network in automated anatomical labeling of coronary arteries," in *CVPR*, 2020.

[363] X. Zhang, Z. Cui, J. Feng, Y. Song, D. Wu, and D. Shen, "Corlab-net: Anatomical dependency-aware point-cloud learning for automatic labeling of coronary arteries," in *MLMI*, 2021.

[364] G. Zhao, K. Liang, C. Pan, F. Zhang, X. Wu, X. Hu, and Y. Yu, "Graph convolution based cross-network multi-scale feature fusion for deep vessel segmentation," *IEEE TMI*, 2023.

[365] R. Selvan, T. Kipf, M. Welling, J. H. Pedersen, J. Petersen, and M. de Bruijne, "Extraction of airways using graph neural networks," in *Proceedings of the Medical Imaging with Deep Learning*, 2018.

[366] A. Garcia-Uceda Juarez, R. Selvan, Z. Saghir, and M. d. Bruijne, "A joint 3d unet-graph neural network-based method for airway segmentation from chest cts," in *MLMI*, 2019.

[367] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[368] R. Selvan, T. Kipf, M. Welling, A. G.-U. Juarez, J. H. Pedersen, J. Petersen, and M. de Bruijne, "Graph refinement based airway

extraction using mean-field networks and graph neural networks," *Medical Image Analysis*, vol. 64, 2020.
[369] R. D. S. Mukul, N. Navab, S. Albarqouni *et al.*, "An uncertainty-driven gcn refinement strategy for organ segmentation," *Machine Learning for Biomedical Imaging*, vol. 1, 2020.
[370] Z. Tian, X. Li, Y. Zheng, Z. Chen, Z. Shi, L. Liu, and B. Fei, "Graph-convolutional-network-based interactive prostate segmentation in mr images," *Medical physics*, vol. 47, 2020.
[371] Y. Meng, M. Wei, D. Gao, Y. Zhao, X. Yang, X. Huang, and Y. Zheng, "Cnn-gcn aggregation enabled boundary regression for biomedical image segmentation," in *MICCAI*, 2020.
[372] Y. Meng, W. Meng, D. Gao, Y. Zhao, X. Yang, X. Huang, and Y. Zheng, "Regression of instance boundary by aggregated cnn and gcn," in *ECCV*, 2020.
[373] C.-H. Chao, Z. Zhu, D. Guo, K. Yan, T.-Y. Ho, J. Cai, A. P. Harrison, X. Ye, J. Xiao, A. Yuille *et al.*, "Lymph node gross tumor volume detection in oncology imaging via relationship learning using graph neural network," in *MICCAI*, 2020.
[374] Z. Yan, K. Youyong, W. Jiasong, G. Coatrieux, and S. Huazhong, "Brain tissue segmentation based on graph convolutional networks," in *ICIP*, 2019.
[375] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "Self-supervised learning for medical image analysis using image context restoration," *Medical image analysis*, vol. 58, 2019.
[376] H.-Y. Zhou, S. Yu, C. Bian, Y. Hu, K. Ma, and Y. Zheng, "Comparing to learn: Surpassing imagenet pretraining on radiographs by comparing image representations," in *MICCAI*, 2020.
[377] Z. Zhou, V. Sodha, J. Pang, M. B. Gotway, and J. Liang, "Models genesis," *Medical image analysis*, vol. 67, 2021.
[378] H.-Y. Zhou, C. Lu, C. Chen, S. Yang, and Y. Yu, "A unified visual information preservation framework for self-supervised pre-training in medical image analysis," *IEEE TPAMI*, vol. 45, 2023.
[379] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, 2010.
[380] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs." *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48, 2016.
[381] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE TKDE*, vol. 29, 2017.
[382] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. P. Langlotz, "Contrastive learning of medical visual representations from paired images and text," *arXiv preprint arXiv:2010.00747*, 2020.
[383] S.-C. Huang, L. Shen, M. P. Lungren, and S. Yeung, "Gloria: A multimodal global-local representation learning framework for label-efficient medical image recognition," in *ICCV*, 2021.
[384] H.-Y. Zhou, X. Chen, Y. Zhang, R. Luo, L. Wang, and Y. Yu, "Generalized radiograph representation learning via cross-supervision between images and free-text radiology reports," *Nature Machine Intelligence*, vol. 4, 2022.
[385] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *CVPR*, 2018.