VisionGNN: A Survey on Lightweight Graph Models and Visual Representation Learning in Computer Vision

www.surveyx.cn

Abstract

VisionGNN represents a transformative framework that integrates Graph Neural Networks (GNNs) with lightweight graph models and graph structural inductive biases to enhance visual representation learning. This survey systematically explores the multifaceted applications of VisionGNN in computer vision, organized into four parts: foundational concepts, methods, frontier topics, and applications. Key advancements include addressing over-smoothing in GNNs and improving model scalability, particularly in low homophily scenarios. VisionGNN leverages innovative techniques such as gated graph convolutional networks and structural message-passing to enhance model interpretability and performance. Applications span diverse domains, including 3D point cloud segmentation, semantic segmentation, and medical imaging, with significant improvements in accuracy and model efficiency. The survey highlights the critical role of structural biases in improving model interpretability and performance, as evidenced by enhanced predictive capabilities in tasks like multi-label land cover classification. Future research directions suggest expanding VisionGNN's capabilities through dual hypergraph neural networks, scalable architectures, and advanced optimization techniques. By synthesizing these insights, the survey underscores VisionGNN's potential to advance computer vision technologies and its applicability across diverse visual tasks.

1 Introduction

1.1 Structure of the Survey

This survey comprises four parts, each aimed at providing a thorough understanding of VisionGNN and its applications in computer vision. Part I introduces essential concepts, defining a conceptual framework that integrates lightweight graph models with graph structural inductive bias. Part II explores foundational methods, focusing on the advancements and challenges of Graph Neural Networks (GNNs) in visual data modeling [1]. Part III addresses emerging topics, discussing the development and optimization of lightweight graph models and the incorporation of graph structural inductive biases to improve model interpretability and performance. Lastly, Part IV highlights applications, presenting real-world implementations and case studies in areas such as 3D point cloud segmentation, semantic segmentation, and medical imaging [2]. This structured approach fosters a systematic and comprehensive understanding of the VisionGNN framework and its influence on visual representation learning.The following sections are organized as shown in Figure 1.

Generated by SurveyX.cn.

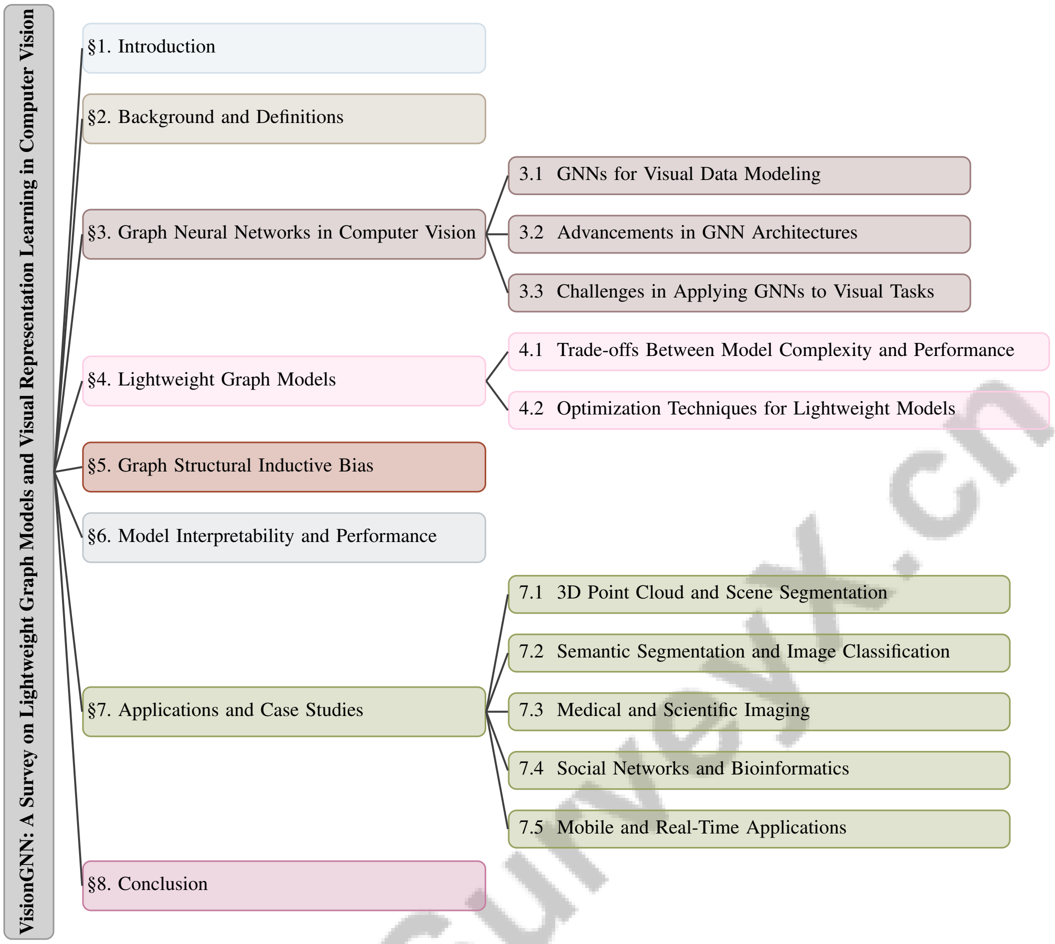

VisionGNN: A Survey on Lightweight Graph Models and Visual Representation Learning in Computer Vision
§1. Introduction
§2. Background and Definitions
§3. Graph Neural Networks in Computer Vision
3.1 GNNs for Visual Data Modeling
3.2 Advancements in GNN Architectures
3.3 Challenges in Applying GNNs to Visual Tasks
§4. Lightweight Graph Models
4.1 Trade-offs Between Model Complexity and Performance
4.2 Optimization Techniques for Lightweight Models
§5. Graph Structural Inductive Bias
§6. Model Interpretability and Performance
§7. Applications and Case Studies
7.1 3D Point Cloud and Scene Segmentation
7.2 Semantic Segmentation and Image Classification
7.3 Medical and Scientific Imaging
7.4 Social Networks and Bioinformatics
7.5 Mobile and Real-Time Applications
§8. Conclusion

Figure 1: chapter structure

2 Background and Definitions

2.1 Conceptual Framework and Definitions

The VisionGNN framework synergistically integrates graph neural networks (GNNs) with structural inductive biases and lightweight graph models to advance visual representation learning. GNNs are pivotal in capturing intricate dependencies within visual data, thus effectively modeling both local and global structural information crucial for tasks like scene understanding and object recognition [3]. By employing gated graph convolutional networks (GCNs) alongside deep structured feature embedding (DSFE), VisionGNN enhances segmentation accuracy and refines weak semantic predictions [4].

Addressing the challenge of over-smoothing in GNNs, especially in heterophilic graphs where connected nodes display dissimilar features, VisionGNN mitigates inefficient aggregation [5]. It enhances GNN performance in contexts where features misalign with the graph's topology, particularly in low homophily settings, by utilizing advanced model construction techniques and higher-order interactions to capture long-range dependencies [6].

VisionGNN improves GNN scalability for inference on unseen nodes, which typically necessitates fixed feature preprocessing [7]. It advances Node Local Distributions (NLD) modeling through innovative techniques, integrating diverse modalities into a unified graph representation that accommodates varying inductive biases [8]. Furthermore, it addresses standard GNN limitations in distinguishing non-isomorphic graphs, enhancing their applicability in graph learning tasks [9].

A central aspect of VisionGNN is the enhancement of performance through graph data, focusing on the quality and quantity of neighborhood information. Techniques capturing complex semantic relationships and content-based correlations among nodes, especially involving higher-order neighbors, are vital [10]. The Task-level Relation Module (TLRM) enriches the understanding of sample relations within tasks, bolstering VisionGNN's comprehensive modeling capabilities. Self-supervised methods are also employed for effective representation learning of nodes and graphs.

VisionGNN distinguishes nodes with identical structural roles but varying positional significances, particularly in molecular graphs, by reducing reliance on manually defined meta-paths for effective node embedding. The Inductive Graph Alignment Prompt (IGAP) method uses learnable prompts to align graph signals and structures, enhancing heterogeneous graph neural networks (HGNNs) by capturing high-order neighborhood information. This integration deepens information processing in heterogeneous graphs. Unlike traditional HGNNs that focus on first-order neighbors and often depend on meta-paths, VisionGNN incorporates both meta-paths and meta-graphs, utilizing a self-attention mechanism for a comprehensive exploration of content-based interactions among nodes [11, 12, 13].

VisionGNN also integrates Graph Deconvolution Networks (GDN) to learn latent graph distributions through a supervised framework, enhancing expressiveness and interpretability. The Dual Hypergraph Neural Network (Dual-HGNN) approach improves semi-supervised node classification by capturing high-order semantic correlations among data points. VisionGNN further enhances its representation capabilities through structural message-passing (SMP) methods, maintaining permutation equivariance while manipulating node identifiers. This allows for effective capture of local features and topological properties, improving generalization across diverse graph structures. By propagating one-hot encodings alongside node features, VisionGNN constructs a local context matrix that encapsulates rich structural information, leading to more accurate predictions of graph topological properties and superior performance in tasks like molecular graph regression [14, 15, 7, 12, 16]. The framework addresses inefficiencies in existing Graph-Informed (GI) layer implementations by leveraging adjacency matrix sparsity to reduce memory usage. Through these advancements, VisionGNN provides structured and interpretable visual data representations, significantly enhancing computer vision technologies and broadening GNN applicability across various visual tasks.

3 Graph Neural Networks in Computer Vision

Graph Neural Networks (GNNs) have revolutionized computer vision by enabling sophisticated modeling of visual data through graph-based structures. Their ability to capture intricate relationships within visual datasets surpasses traditional methods, enhancing tasks such as semantic segmentation and object recognition. This section delves into GNN applications in visual data modeling and explores the innovative methodologies and architectures that have emerged.

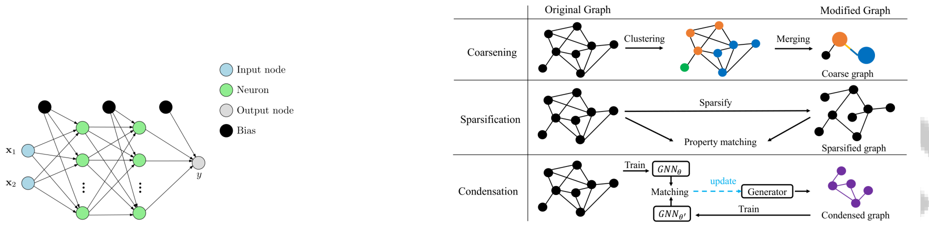
3.1 GNNs for Visual Data Modeling

GNNs excel in visual data modeling by capturing complex relationships using graph structures, effectively representing local and global information crucial for semantic segmentation and object recognition. Edge-featured Graph Neural Architecture Search (EGNAS) enhances performance by incorporating both entity and edge features [17]. Handcrafted matching priors facilitate graph matching without extensive training, showcasing GNNs' adaptability in complex data structures [18]. The Fea2Fea method enriches feature learning through advanced message passing [19]. Flexgrid2vec employs flexible grid-graphs to represent images as low-dimensional feature vectors, yielding compact visual representations [4]. Hierarchical Bipartite Graph Convolution (BiGraphNet) improves scalability by directly connecting input and output nodes [20]. The MAG framework uses a masking mechanism to tailor attention patterns, enhancing information propagation across diverse tasks [8]. Collectively, these methods demonstrate GNNs' robustness in visual data modeling, significantly improving interpretability and performance in computer vision applications.

3.2 Advancements in GNN Architectures

Recent advancements in GNN architectures have bolstered visual representation learning. Harada et al.'s dual convolutional framework integrates internal and external graph structures, capturing complex dependencies [21]. AnchorGT employs k-dominating sets as anchors, enhancing flexibility and efficiency in large-scale data handling [22]. GPNet combines dilated convolution, multi-channel learning, and self-attention to handle datasets with varying homophily [23]. SNAG simplifies architecture search, improving upon frameworks like GraphNAS and Auto-GNN [24]. Graph Feature

Network (GFN) and Graph Linear Network (GLN) streamline filtering and set function components, maintaining competitive performance with reduced computational overhead [25]. AERO-GNN enhances deep graph attention for nuanced inference [26]. The Cluster Information Transfer (CIT) mechanism enriches node representations by integrating cluster information [27]. Factor Graph Neural Networks (FGNN) utilize low-rank tensor decomposition for efficient message passing [28]. BiGraphNet generalizes graph convolutions for hierarchical graphs, optimizing computations for large datasets [20]. These advancements underscore the evolving GNN architectures, driving improvements in visual representation across applications.



(a) A neural network architecture with multiple layers of neurons and connections[29]

(b) Graph Modification Techniques: Coarsening, Sparsification, and Condensation[30]

Figure 2: Examples of Advancements in GNN Architectures

As depicted in Figure 2, GNNs offer transformative solutions in computer vision through innovative graph-based structures. The advancements in GNN architectures include complex neural network designs and novel graph modification techniques. The first subfigure illustrates a multi-layer neural network, highlighting intricate information flow. The second subfigure presents graph modification techniques essential for optimizing GNN structure and efficiency. These developments emphasize the dynamic evolution of GNN architectures, enhancing robustness and scalability in computer vision [29, 30].

3.3 Challenges in Applying GNNs to Visual Tasks

Applying GNNs to visual tasks involves several challenges affecting their effectiveness and scalability. Oversmoothing, where node representations become indistinguishable, reduces discriminative power [31]. Information over-squashing during message passing limits sensitivity to distant information [32]. Efficient computation on large and dynamic graphs remains complex [3], and inefficiencies in training methods, particularly with sparse computations, create bottlenecks [33]. Integrating topological features without performance loss is challenging, as many methods fail to leverage these effectively [34]. Capturing positional node information is crucial for tasks such as classification, yet current methods often fall short [35]. Learning fundamental graph properties like connectivity limits GNN effectiveness in structural tasks [16]. Determining beneficial feature dimensions for graph convolution can degrade performance [36]. Reliance on manually defined meta-paths can introduce inaccuracies [37], and methods using dummy nodes and edges increase computational complexity [5]. Addressing these challenges requires novel architectures and strategies that manage visual data complexity and diversity. Optimizing message-passing, enhancing interpretability, and developing flexible inference approaches are crucial. Strategies like decoupling node feature learning from graph topology and employing adaptive filters can enhance adaptability [38]. Improving Class Activation Mapping (CAM) methods is essential for accurate semantic representation. Existing methods struggle to scale and optimize computations in hierarchical networks, often relying on separate pooling layers that do not adapt to graph structure. Developing integrated approaches considering data hierarchy could mitigate these issues [20]. Overcoming these challenges will enable GNNs to better handle the complexities of visual tasks, leading to more robust and scalable models.

In recent years, the development of lightweight graph models has garnered significant attention due to their potential to balance model complexity and performance. Figure 3 illustrates the hierarchical categorization of these models, emphasizing the trade-offs involved and the optimization techniques that enhance their efficiency. The first category focuses on efficient memory management, adaptable models, and computational efficiency, which are crucial for ensuring that these models can operate effectively within constrained environments. In contrast, the second category highlights the integration of semantic information, dynamic propagation depth, efficient embedding construction, and hierarchical graph processing. Together, these concepts underscore the strategic design and application of lightweight graph models, ensuring high performance and adaptability across diverse applications.

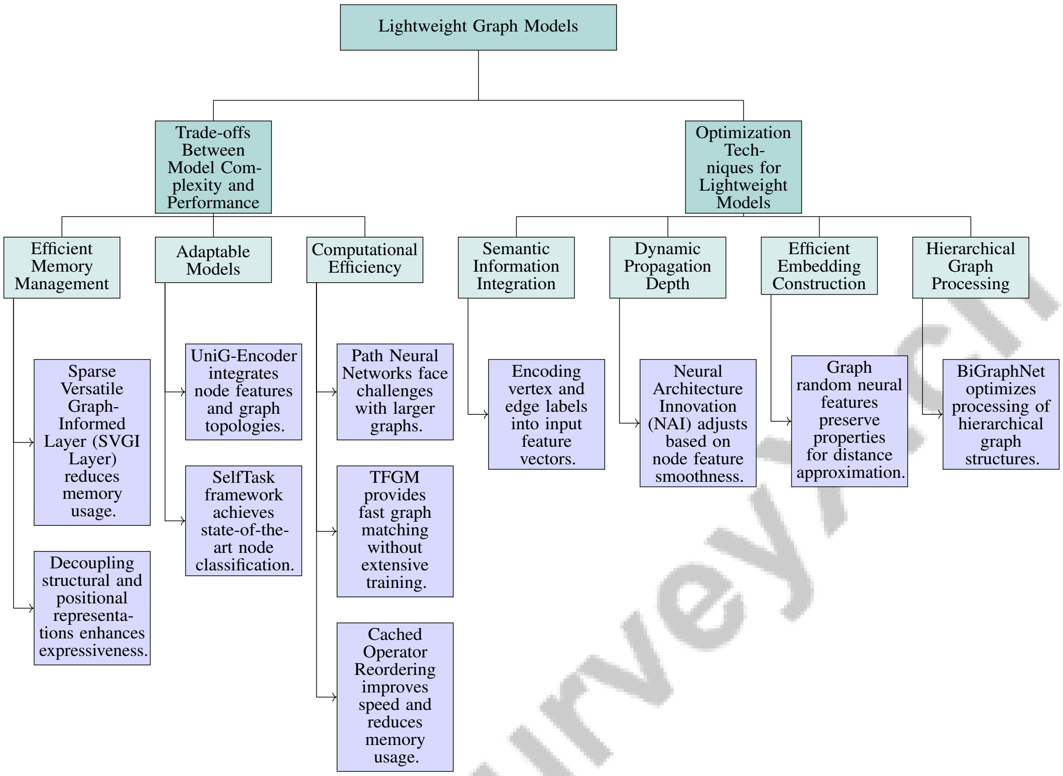

Lightweight Graph Models
Trade-offs Between Model Complexity and Performance
Optimization Techniques for Lightweight Models
Efficient Memory Management
Adaptable Models
Computational Efficiency
Semantic Information Integration
Dynamic Propagation Depth
Efficient Embedding Construction
Hierarchical Graph Processing
Sparse Versatile Graph-Informed Layer (SVGI Layer) reduces memory usage.
Decoupling structural and positional representations enhances expressiveness.
UniG-Encoder integrates node features and graph topologies.
SelfTask framework achieves state-of-the-art node classification.
Path Neural Networks face challenges with larger graphs.
TFGM provides fast graph matching without extensive training.
Cached Operator Reordering improves speed and reduces memory usage.
Encoding vertex and edge labels into input feature vectors.
Neural Architecture Innovation (NAI) adjusts based on node feature smoothness.
Graph random neural features preserve properties for distance approximation.
BiGraphNet optimizes processing of hierarchical graph structures.

Figure 3: This figure illustrates the hierarchical categorization of lightweight graph models, emphasizing the trade-offs between model complexity and performance, and the optimization techniques for enhancing efficiency. The first category delves into efficient memory management, adaptable models, and computational efficiency. The second category highlights semantic information integration, dynamic propagation depth, efficient embedding construction, and hierarchical graph processing. These concepts collectively underscore the strategic design and application of lightweight graph models to ensure high performance and adaptability across diverse applications.

4 Lightweight Graph Models

4.1 Trade-offs Between Model Complexity and Performance

Method Name	Complexity Management	Model Adaptability	Efficiency Optimization
LSPE[12]	-	Diverse Graph Structures	-
UGE[39]	Efficient Memory Management	Homophilic Heterophilic Environments	Optimize Computational Efficiency
ST[40]	-	Diverse Graph Structures	-
PathNN[41]	Computational Efficiency Challenges	Enhancing Expressiveness Beyond	Optimizing Computational Efficiency
TFGM[18]	Efficient Memory Management	Diverse Graph Structures	Reduce Resource Usage
TR-GNN[34]	Dual Propagation Strategy	Homophilic Heterophilic Environments	Reduce Resource Usage
AORC[33]	Efficient Memory Management	-	Optimize Computational Efficiency

Table 1: Comparison of various graph neural network methods in terms of complexity management, model adaptability, and efficiency optimization. The table highlights the diverse strategies employed by each method to balance computational efficiency and adaptability in different graph environments.

Balancing model complexity and performance is crucial for lightweight graph models, particularly in environments with limited resources. The Sparse Versatile Graph-Informed Layer (SVGI Layer) exemplifies this by retaining only non-zero elements of the adjacency matrix, reducing memory usage while maintaining efficiency [42]. This highlights the necessity of efficient memory management in model design.

Dwivedi et al. propose decoupling structural and positional representations to enhance GNN expressiveness without added complexity, offering a robust alternative to traditional methods [12]. The UniG-Encoder effectively integrates node features and graph topologies, excelling in both homophilic and heterophilic contexts, underscoring the need for adaptable models across diverse graph structures [39].

The SelfTask framework employs advanced pretext tasks using structural and attribute information, achieving state-of-the-art node classification performance while maintaining a lightweight architecture [40]. This demonstrates the potential for high performance with reduced complexity, suitable for various applications.

Path Neural Networks face challenges in computational efficiency with larger graphs and longer path lengths, resulting in increased time complexity [41]. This necessitates strategies to optimize computational efficiency without sacrificing performance.

TFGM provides a fast solution for graph matching without extensive training, showing that lightweight models can perform well in data-limited scenarios [18]. However, the topological regularization method may not universally enhance all models or datasets, especially when baseline performance is suboptimal, indicating the need for careful model applicability consideration [34].

The Cached Operator Reordering method significantly improves speed and reduces memory usage across various hardware platforms, emphasizing computational efficiency in managing the complexity-performance trade-off [33]. These strategies collectively stress the importance of balancing model complexity and performance to ensure lightweight graph models remain efficient and effective in diverse applications. Table 1 provides a comparative analysis of different graph neural network methods, illustrating their approaches to managing complexity, adaptability, and efficiency in the context of lightweight model design.

4.2 Optimization Techniques for Lightweight Models

Method Name	Model Efficiency	Adaptability Mechanisms	Resource Management
LPG2vec[43]	-	Dynamic Adjustments	-
NAI[44]	Node-Adaptive Inference	Dynamic Adjustment Methods	-
GRNF[45]	Performance And Scalability	Dynamic Adjustments	Cached Operator Reordering
AORC[33]	Significant Performance Improvements	Adaptive Operator Reordering	Adaptive Caching Strategies
BGN[20]	Enhanced Computational Efficiency	Bipartite Graph Convolutions	Reducing Complexity

Table 2: This table presents a comparative analysis of various optimization techniques for lightweight graph models, highlighting their model efficiency, adaptability mechanisms, and resource management strategies. The methods listed include LPG2vec, NAI, GRNF, AORC, and BGN, each contributing distinct approaches to enhance computational efficiency and adaptability in graph neural networks.

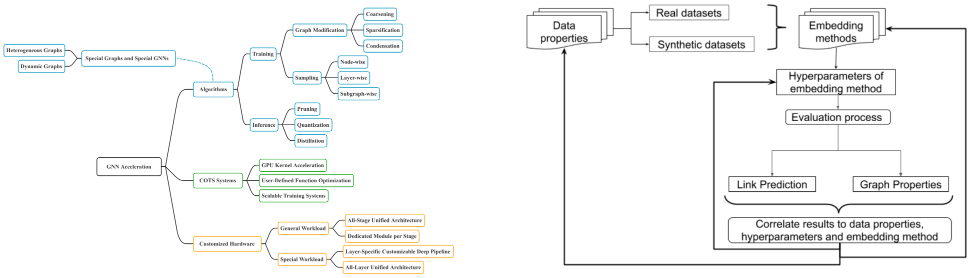
Optimization techniques are vital for enhancing the efficiency and scalability of lightweight graph models across applications. Encoding vertex and edge labels into input feature vectors allows GNNs to leverage rich semantic information effectively [43], integrating complex data structures without compromising performance.

Neural Architecture Innovation (NAI) dynamically adjusts propagation depth based on node feature smoothness, ensuring efficiency across various graph structures while optimizing computational resources [44]. The SNAG framework enhances adaptability with a novel search space and reinforcement learning-based search algorithm, enabling automatic discovery of data-specific GNN architectures suited for particular tasks [24].

Graph random neural features for embedding construction preserve essential properties for accurate distance approximation, allowing lightweight models to maintain efficacy in tasks requiring precise relational inferences [45]. The Cached Operator Reordering method optimizes memory and computation management, crucial for deploying lightweight models in resource-constrained environments [33].

BiGraphNet introduces operations similar to strided and dilated convolutions, optimizing the model's ability to efficiently process hierarchical graph structures, enhancing scalability and responsiveness of lightweight models to handle large-scale data with minimal computational overhead [20]. These optimization techniques underscore the importance of strategic model design in achieving high efficiency and performance in lightweight graph models.

Table 2 provides a detailed overview of optimization methods employed to enhance the efficiency and adaptability of lightweight graph models, as discussed in the preceding section.



(a) GNN Acceleration Techniques[30]

(b) An Overview of a Data Analysis Process[46]

Figure 4: Examples of Optimization Techniques for Lightweight Models

As depicted in Figure 4, optimization techniques are crucial for enhancing the efficiency and performance of lightweight graph models. The first figure, "GNN Acceleration Techniques," presents a comprehensive mind map detailing various strategies for expediting GNNs, including heterogeneous and dynamic graphs, special graphs and GNNs, graph modification, sampling, and training. The second figure, "An Overview of a Data Analysis Process," outlines a structured approach to data analysis, emphasizing the importance of leveraging both real and synthetic datasets. This process begins with identifying data properties and applying real-world datasets for robust training and testing of analytical methods. Together, these figures underscore the significance of tailored optimization techniques in developing and applying lightweight graph models, ensuring effectiveness and efficiency in handling complex data structures [30, 46].

5 Graph Structural Inductive Bias

5.1 Techniques for Incorporating Structural Bias

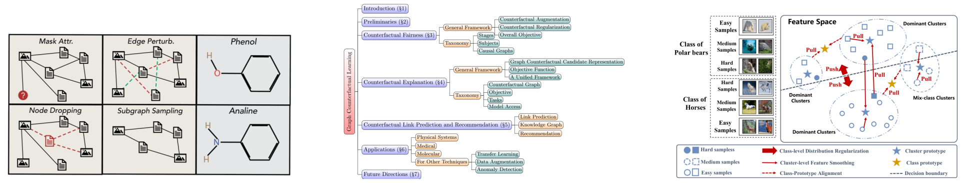
Integrating structural biases into Graph Neural Networks (GNNs) is pivotal for enhancing both interpretability and performance. The SelfTask framework exemplifies this by leveraging structural and attribute information to boost node classification [40]. This strategic use of structural data enables GNNs to capture essential graph characteristics. The flexgrid2vec method further emphasizes this by aggregating features from selected image patches while maintaining spatial relationships, thus enhancing representation learning [4].

Adaptive operator reordering with caching, as outlined by Bazinska et al., enhances GNN performance by reducing redundant computations and improving memory efficiency [33]. This method effectively capitalizes on the structural biases inherent in graph data, leading to more interpretable models. Similarly, the hierarchical bipartite graph convolution approach facilitates distinct input and output vertex sets, optimizing computational efficiency through new operations [20]. By exploiting structural biases, this method enhances the processing of hierarchical graph structures.

These techniques collectively highlight the importance of structural biases in GNNs for improved model interpretability and performance. Advanced methods, such as Class-level Structural Relation Modeling and graph convolution networks, leverage structural relationships in graph data to refine visual representation learning, modeling complex inter-class similarities and intra-class variations [47, 48, 15, 49].

As depicted in Figure 5, structural bias integration is crucial for enhancing model performance in graph-based learning. The figure showcases three techniques: "Graph-based methods for molecular structure prediction" illustrates diverse approaches for molecular formation prediction, "Graph Counterfactual Learning" details aspects of counterfactual reasoning in graph models, and "Class-Prototype Alignment for Domain Adaptation" demonstrates aligning class prototypes across domains.

AI-generated, for reference only.



(a) Graph-based methods for molecular structure prediction[50]

(b) Graph Counterfactual Learning[51]

(c) Class-Prototype Alignment for Domain Adaptation[49]

Figure 5: Examples of Techniques for Incorporating Structural Bias

These examples underscore the role of structural bias in advancing graph-based methodologies for accurate predictions and effective domain adaptations [50, 51, 49].

5.2 Performance Enhancement through Structural Bias

Incorporating structural bias within VisionGNN frameworks markedly improves performance in computer vision applications. The ViG architecture exemplifies this by significantly enhancing multi-label land cover classification through its capacity to capture complex visual data dependencies, thereby improving accuracy and robustness [52].

In visual representation learning, structural biases are essential for modeling intricate data relationships. VisionGNNs utilize frameworks like Class-level Structural Relation Modeling and Smoothing (CSRMS) to enhance representation learning via class-aware graph sampling and relational graph-guided techniques. These address intra-class visual diversity and inter-class similarity, boosting performance in tasks such as image captioning and visual question answering [47, 14, 15, 49]. Graph-based structures enable models to capture local and global dependencies, crucial for tasks requiring detailed semantic understanding. Structural bias incorporation not only enhances interpretability but also optimizes performance by preserving and utilizing critical data patterns and relationships.

Moreover, structural bias in VisionGNN frameworks supports the development of efficient models for complex visual tasks with reduced computational demands. By strategically incorporating structural information, these models achieve higher accuracy and efficiency, suitable for applications ranging from land cover classification to complex visual recognition. These advancements highlight the role of structural bias in optimizing Vision Graph Neural Networks (VisionGNNs) by leveraging local and global structural information, essential for improving tasks like image captioning, visual question answering, and image retrieval, contributing to the evolution of graph-based visual representation learning [14, 53, 15, 49].

6 Model Interpretability and Performance

Understanding how models generate predictions is essential for establishing trust in AI applications. This section explores the relationship between model interpretability and performance, particularly within VisionGNN frameworks. By examining how structured visual data representations enhance interpretability, we highlight their implications for model performance, providing a foundation for understanding their significance in practical applications.

6.1 Impact on Model Interpretability

In VisionGNN frameworks, structured visual data representations significantly enhance interpretability by elucidating how GNNs utilize structural information for predictions. The MATE method exemplifies this by embedding explanation tasks within the training process, thereby improving GNN explainability [54]. This approach ensures the model captures essential structural features while providing insights into its decision-making.

Model-agnostic explainability methods, as demonstrated by Pelletreau-Duris et al., further reveal how GNNs incorporate structural graph properties into their learned representations [55]. Such techniques allow researchers to dissect internal mechanisms, identifying influential structural features that shape

AI-generated, for reference only.

predictions. These insights are crucial for understanding model behavior and ensuring GNNs are both accurate and transparent.

The ability to interpret model decisions through structured representations is vital in applications where understanding the rationale behind predictions is as critical as the predictions themselves. VisionGNN frameworks thus support the development of trustworthy systems capable of delivering actionable insights in high-stakes fields such as medical imaging, autonomous driving, and environmental monitoring. For instance, the Gaze-Directed Vision GNN (GDViG) enhances interpretability by utilizing radiologists' gaze patterns to focus on disease-relevant areas, thereby reducing shortcut learning and increasing reliability in medical image analysis. Additionally, the Dynamic Axial Graph Construction (DAGC) method within the GreedyViG architecture illustrates how efficient graph construction can enhance model performance while maintaining interpretability, emphasizing the importance of robust interpretative frameworks in critical applications [56, 57, 58]. Consequently, structured representation not only improves performance but also aligns models with human understanding, promoting informed and responsible use of GNNs.

6.2 Evaluation Metrics for Model Performance

Benchmark	Size	Domain	Task Format	Metric
GSI-Bench[53]	45,000	Graph Representation Learning	Node Classification	Accuracy, MSE
GNN-ANF[59]	10	Graph Classification	Node Classification	Accuracy, F1-score
GRL-Bench[46]	5,000	Graph Theory	Graph Property Prediction	Micro-F1, R2
GNN-Emb[15]	3,000	Image Classification	Node Classification	Accuracy
CMVRL[60]	1,000	Graph Classification	Node Classification	Accuracy, NMI
GNN-VIS[61]	15,000	3D Object Detection	Object Recognition	Top-1

Table 3: This table presents an overview of representative benchmarks used in graph-based learning, detailing their size, domain, task format, and evaluation metrics. The benchmarks, including GSI-Bench, GNN-ANF, GRL-Bench, GNN-Emb, CMVRL, and GNN-VIS, are critical for assessing model performance across various graph-related tasks, offering insights into accuracy, classification, and prediction capabilities.

Evaluating graph-based models in computer vision requires comprehensive metrics that assess various aspects of effectiveness and efficiency. Mean Intersection over Union (mIOU) is a primary metric for segmentation performance, offering a robust measure of model accuracy in delineating different image regions [62]. This metric is particularly valuable in semantic segmentation, where precise boundary delineation is crucial.

Alongside mIOU, precision, recall, and F1 score provide a holistic view of model effectiveness, especially in classification tasks [52]. Precision reflects the accuracy of positive predictions, recall assesses the model's ability to identify relevant instances, and the F1 score balances precision and recall, accounting for both false positives and negatives.

For clustering and link prediction tasks, metrics such as Accuracy, Normalized Mutual Information (NMI), Adjusted Random Index (ARI), and Area Under the Curve (AUC) evaluate model performance [63]. These metrics are essential for understanding a model's ability to group similar data points and predict node connections, critical capabilities in graph-based learning.

To address over-smoothing challenges, metrics like row-diff, col-diff, group distance ratio, and instance information gain provide insights into the model's capacity to maintain distinct node representations [64]. These metrics are vital for preserving critical structural information during the learning process.

Performance evaluation also encompasses computational efficiency, with metrics such as the runtime of forward and backward passes crucial for assessing graph-based model efficiency [33]. Comparing these runtimes against baseline implementations highlights the computational advantages of novel methods, particularly in resource-constrained environments.

The integration of various performance metrics establishes a robust framework for assessing graph-based models in computer vision, ensuring accuracy, effectiveness, efficiency, and scalability across diverse applications. This framework includes innovative metrics like Topological Feature Informativeness (TFI) and smoothness metrics to enhance feature selection and graph information utilization. Moreover, the incorporation of Large Language Models (LLMs) into GNNs via frameworks such as E-LLaGNN demonstrates a scalable approach to optimizing node features, thereby maximizing computational resources while maintaining high performance across various datasets [36, 7, 65]. Table 3 provides a comprehensive overview of key benchmarks in graph-based learning, highlighting their relevance in evaluating model performance across diverse domains and tasks.

7 Applications and Case Studies

7.1 3D Point Cloud and Scene Segmentation

VisionGNN frameworks are pivotal in enhancing 3D point cloud and scene segmentation by leveraging the structural advantages of Graph Neural Networks (GNNs) to capture complex spatial relationships essential for object recognition and scene understanding [3]. Edge-featured architectures in GNNs, as demonstrated by Cai et al., improve the model's ability to discern fine-grained details crucial for applications like autonomous driving and robotics [17]. Hierarchical bipartite graph convolution methods further optimize large-scale 3D data processing by enhancing node connectivity and reducing computational overhead [20]. By incorporating structural biases, VisionGNN frameworks significantly enhance segmentation results, offering robust solutions for complex 3D environments [4].

7.2 Semantic Segmentation and Image Classification

In semantic segmentation and image classification, VisionGNN frameworks utilize graph-based structures to capture local and global dependencies, crucial for delineating semantic boundaries and categorizing image content. The Edge-featured Graph Neural Architecture Search (EGNAS) exemplifies the enhancement of GNN performance through architectural optimization for segmentation tasks [17]. Handcrafted matching priors and the Fea2Fea method further showcase GNNs' adaptability and ability to exploit structural feature relationships [18, 19]. Flexible grid-graphs, as in the flexgrid2vec approach, and the Hierarchical Bipartite Graph Convolution (BiGraphNet) method streamline graph convolutions, enhancing scalability and efficiency [4, 20]. The MAG framework's masking mechanism highlights GNNs' versatility in various visual tasks [8].

7.3 Medical and Scientific Imaging

VisionGNN frameworks significantly improve precision and efficiency in medical and scientific imaging by identifying intricate spatial and relational patterns, crucial for accurate disease diagnosis and anomaly detection. By minimizing shortcut learning risks and integrating expert visual patterns, GNNs enhance diagnostic accuracy and reliability [56, 14, 36, 66, 57]. In medical imaging, GNNs model complex relationships within high-dimensional data, improving anomaly detection and facilitating tasks like image captioning [14, 66]. Hierarchical graph convolution techniques optimize large-scale dataset processing, reducing computational overhead [20]. In scientific imaging, VisionGNNs effectively analyze complex datasets, enhancing segmentation and classification results [4].

7.4 Social Networks and Bioinformatics

VisionGNN frameworks advance the analysis of complex relational data in social networks and bioinformatics by modeling intricate user interactions and biological networks. In social networks, they aid in community detection and influence prediction [3]. In bioinformatics, VisionGNNs capture dependencies within biological systems, enhancing our understanding of critical pathways and functional modules [67, 3, 12, 53]. Integrating structural biases improves predictive accuracy and interpretability, essential for applications like personalized medicine [4]. Hierarchical graph convolution techniques optimize processing of large-scale datasets, maintaining performance while reducing computational overhead [20].

7.5 Mobile and Real-Time Applications

VisionGNN frameworks, particularly MobileViG and PointViG models, enhance mobile and real-time applications by improving efficiency and functionality in computer vision tasks. MobileViG employs

AI-generated, for reference only.

a hybrid CNN-GNN architecture with a graph-based sparse attention mechanism, achieving high performance metrics on mobile devices [68, 69, 70]. PointViG enhances point cloud analysis, demonstrating competitive accuracy and efficiency in resource-constrained environments. Cached Operator Reordering techniques further enhance computational efficiency [33]. In real-time applications, VisionGNN's rapid processing capabilities provide timely insights critical for decision-making in dynamic environments [20]. The adaptability and efficiency of VisionGNNs make them ideal for mobile and real-time applications, offering substantial performance improvements over traditional methods [14, 69, 70, 57, 52].

8 Conclusion

8.1 Future Directions and Innovations

The trajectory of VisionGNN research presents numerous prospects for advancing the integration of graph models within visual representation learning. Exploring Dual Hypergraph Neural Networks (DualHGNN) for larger datasets and extending their application to graph-level tasks, such as classification, represents a promising direction. Enhancing feature sets with additional structural features and leveraging frameworks like Fea2Fea on expansive datasets can potentially elevate model performance.

Further examination of the Cluster Information Transfer (CIT) mechanism for graph-level tasks, alongside a deeper theoretical exploration, could provide significant insights into invariant representation learning. Additionally, advancing large-scale pre-training and transfer learning within contrastive methods may broaden their utility across various domains.

The development of scalable Graph Neural Network (GNN) architectures is crucial for improving generalization capabilities, with potential applications in emerging fields like social network analysis and bioinformatics. Employing tailored search strategies for graph neural architectures, such as Edge-featured Graph Neural Architecture Search (EGNAS), could enhance architectural efficiency and innovation.

Incorporating flexgrid2vec in few-shot learning, visual segmentation, and object detection, along with the integration of sophisticated loss functions, offers another promising research avenue. Optimizing graph convolution operations and expanding the applications of BiGraphNet could significantly widen the scope of VisionGNN frameworks.

Lastly, refining caching strategies and enhancing compilation support for diverse GNN architectures are essential for improving computational efficiency and scalability. By pursuing these diverse research paths, VisionGNNs can continue to evolve, driving substantial advancements in visual representation learning and extending the influence of graph-based models across a wide array of applications.

References

[1] Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Xiaojie Guo. Graph neural networks: foundation, frontiers and applications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4840–4841, 2022.

[2] Yu Zhou, Haixia Zheng, Xin Huang, Shufeng Hao, Dengao Li, and Jumin Zhao. Graph neural networks: Taxonomy, advances and trends, 2022.

[3] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications, 2021.

[4] Ali Hamdi, Du Yong Kim, and Flora D. Salim. flexgrid2vec: Learning efficient visual representations vectors, 2021.

[5] Xin Liu, Jiayang Cheng, Yangqiu Song, and Xin Jiang. Boosting graph structure learning with dummy nodes, 2022.

[6] Seyun Bae, Hoyoon Byun, Changdae Oh, Yoon-Sik Cho, and Kyungwoo Song. Graph perceiver io: A general architecture for graph structured data, 2025.

[7] Ajay Jaiswal, Nurendra Choudhary, Ravinarayana Adkathimar, Muthu P. Alagappan, Gaurush Hiranandani, Ying Ding, Zhangyang Wang, Edward W Huang, and Karthik Subbian. All against some: Efficient integration of large language models for message passing in graph neural networks, 2024.

[8] David Buterez, Jon Paul Janet, Dino Oglic, and Pietro Lio. An end-to-end attention-based approach for learning on graphs, 2024.

[9] Veronika Thost and Jie Chen. Directed acyclic graph neural networks. *arXiv preprint arXiv:2101.07965*, 2021.

[10] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pages 4116–4126. PMLR, 2020.

[11] Nan Wu and Chaofan Wang. Heterogeneous graph tree networks, 2022.

[12] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations, 2022.

[13] Jianxin Li, Hao Peng, Yuwei Cao, Yingtong Dou, Hekai Zhang, Philip S. Yu, and Lifang He. Higher-order attribute-enhancing heterogeneous graph neural networks, 2021.

[14] Henry Senior, Gregory Slabaugh, Shanxin Yuan, and Luca Rossi. Graph neural networks in vision-language image understanding: A survey, 2024.

[15] S. Purchase, A. Zhao, and R. D. Mullins. Revisiting embeddings for graph neural networks, 2022.

[16] Clement Vignac, Andreas Loukas, and Pascal Frossard. Building powerful and equivariant graph neural networks with structural message-passing. *Advances in neural information processing systems*, 33:14143–14155, 2020.

[17] Shaofei Cai, Liang Li, Xinzhe Han, Zheng jun Zha, and Qingming Huang. Edge-featured graph neural architecture search, 2021.

[18] Zhiyuan Liu, Yixin Cao, Fuli Feng, Xiang Wang, Jie Tang, Kenji Kawaguchi, and Tat-Seng Chua. Training free graph neural networks for graph matching, 2022.

[19] Jiaqing Xie and Rex Ying. Fea2fea: Exploring structural feature correlations via graph neural networks, 2021.

[20] Marcel Nassar. Hierarchical bipartite graph convolution networks, 2018.

AI-generated, for reference only.

[21] Shonosuke Harada, Hirotaka Akita, Masashi Tsubaki, Yukino Baba, Ichigaku Takigawa, Yoshihiro Yamanishi, and Hisashi Kashima. Dual convolutional neural network for graph of graphs link prediction, 2018.

[22] Wenhao Zhu, Guojie Song, Liang Wang, and Shaoguo Liu. Anchorgt: Efficient and flexible attention architecture for scalable graph transformers, 2024.

[23] Xun Liu, Alex Hay-Man Ng, Fangyuan Lei, Yikuan Zhang, and Zhengmin Li. Gpnet: Simplifying graph neural networks via multi-channel geometric polynomials, 2022.

[24] Huan Zhao, Lanning Wei, and Quanming Yao. Simplifying architecture search for graph neural network, 2020.

[25] Ting Chen, Song Bian, and Yizhou Sun. Are powerful graph neural nets necessary? a dissection on graph classification. *arXiv preprint arXiv:1905.04579*, 2019.

[26] Soo Yong Lee, Fanchen Bu, Jaemin Yoo, and Kijung Shin. Towards deep attention in graph neural networks: Problems and remedies, 2023.

[27] Donglin Xia, Xiao Wang, Nian Liu, and Chuan Shi. Learning invariant representations of graph neural networks via cluster generalization, 2024.

[28] Zhen Zhang, Mohammed Haroon Dupty, Fan Wu, Javen Qinfeng Shi, and Wee Sun Lee. Factor graph neural networks, 2023.

[29] Giannis Nikolentzos, Siyun Wang, Johannes Lutzeyer, and Michalis Vazirgiannis. Graph neural machine: A new model for learning with tabular data, 2024.

[30] Shichang Zhang, Atefeh Sohrabizadeh, Cheng Wan, Zijie Huang, Ziniu Hu, Yewen Wang, Yingyan, Lin, Jason Cong, and Yizhou Sun. A survey on graph neural network acceleration: Algorithms, systems, and customized hardware, 2023.

[31] Takeshi D. Itoh, Takatomi Kubo, and Kazushi Ikeda. Multi-level attention pooling for graph neural networks: Unifying graph representations with multiple localities, 2021.

[32] Lorenzo Giusti. Topological neural networks: Mitigating the bottlenecks of graph neural networks via higher-order interactions, 2024.

[33] Julia Bazinska, Andrei Ivanov, Tal Ben-Nun, Nikoli Dryden, Maciej Besta, Siyuan Shen, and Torsten Hoefler. Cached operator reordering: A unified view for fast gnn training, 2023.

[34] Rui Song, Fausto Giunchiglia, Ke Zhao, and Hao Xu. Topological regularization for graph neural networks augmentation, 2021.

[35] Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks, 2019.

[36] Yilun Zheng, Xiang Li, Sitao Luan, Xiaojiang Peng, and Lihui Chen. Is graph convolution always beneficial for every feature?, 2024.

[37] Zhaoliang Chen, Zhihao Wu, Luying Zhong, Claudia Plant, Shiping Wang, and Wenzhong Guo. Attributed multi-order graph convolutional network for heterogeneous graphs, 2023.

[38] Andrea Apicella, Francesco Isgrò, Andrea Pollastro, and Roberto Prevete. Adaptive filters in graph convolutional neural networks, 2023.

[39] Minhao Zou, Zhongxue Gan, Yutong Wang, Junheng Zhang, Dongyan Sui, Chun Guan, and Siyang Leng. Unig-encoder: A universal feature encoder for graph and hypergraph node classification, 2023.

[40] Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. Self-supervised learning on graphs: Deep insights and new direction, 2020.

[41] Gaspard Michel, Giannis Nikolentzos, Johannes Lutzeyer, and Michalis Vazirgiannis. Path neural networks: Expressive and accurate graph neural networks, 2023.

[42] Francesco Della Santa. Sparse implementation of versatile graph-informed layers, 2024.

[43] Maciej Besta, Patrick Iff, Florian Scheidl, Kazuki Osawa, Nikoli Dryden, Michal Podstawski, Tiancheng Chen, and Torsten Hoefler. Neural graph databases, 2022.

[44] Xinyi Gao, Wentao Zhang, Junliang Yu, Yingxia Shao, Quoc Viet Hung Nguyen, Bin Cui, and Hongzhi Yin. Accelerating scalable graph neural network inference with node-adaptive propagation, 2023.

[45] Daniele Zambon, Cesare Alippi, and Lorenzo Livi. Graph random neural features for distance-preserving graph representations, 2020.

[46] Antonia Gogoglou, C. Bayan Bruss, Brian Nguyen, Reza Sarshogh, and Keegan E. Hines. Quantifying challenges in the application of graph representation learning, 2020.

[47] Yi-Zhan Xu, Chih-Yao Chen, and Cheng-Te Li. Suvr: A search-based approach to unsupervised visual representation learning, 2023.

[48] Mital Kinderkhedia. Learning representations of graph data – a survey, 2019.

[49] Zitan Chen, Zhuang Qi, Xiao Cao, Xiangxian Li, Xiangxu Meng, and Lei Meng. Class-level structural relation modelling and smoothing for visual representation learning, 2023.

[50] Puja Trivedi, Ekdeep Singh Lubana, Yujun Yan, Yaoqing Yang, and Danai Koutra. Augmentations in graph contrastive learning: Current methodological flaws towards better practices, 2022.

[51] Zhimeng Guo, Teng Xiao, Zongyu Wu, Charu Aggarwal, Hui Liu, and Suhang Wang. Counterfactual learning on graphs: A survey, 2024.

[52] Giovanni Sciortino. *Vision Graph Neural Networks for Remote Sensing*. PhD thesis, Politecnico di Torino, 2023.

[53] Davide Buffelli and Fabio Vandin. The impact of global structural information in graph neural networks applications, 2021.

[54] Indro Spinelli, Simone Scardapane, and Aurelio Uncini. A meta-learning approach for training explainable graph neural networks, 2022.

[55] Tom Pelletreau-Duris, Ruud van Bakel, and Michael Cochez. Do graph neural network states contain graph properties?, 2024.

[56] Shaoxuan Wu, Xiao Zhang, Bin Wang, Zhuo Jin, Hansheng Li, and Jun Feng. Gaze-directed vision gnn for mitigating shortcut learning in medical image. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 514–524. Springer, 2024.

[57] Mustafa Munir, William Avery, Md Mostafijur Rahman, and Radu Marculescu. Greedyvig: Dynamic axial graph construction for efficient vision gnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6118–6127, 2024.

[58] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. *arXiv preprint arXiv:2201.12872*, 2022.

[59] Hejie Cui, Zijie Lu, Pan Li, and Carl Yang. On positional and structural node features for graph neural networks on non-attributed graphs, 2022.

[60] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs, 2020.

[61] Felipe Macías Granado and Lama Alkhaled. How gnns can be used in the vehicle industry. In *Artificial Intelligence and Applications*, 2022.

[62] Yi Lu, Yaran Chen, Dongbin Zhao, and Jianxin Chen. Graph-fcn for image semantic segmentation. In *International symposium on neural networks*, pages 97–105. Springer, 2019.

[63] Marinos Poiitis, Pavlos Sermpezis, and Athena Vakali. Pointspectrum: Equivariance meets laplacian filtering for graph representation learning, 2021.

[64] Yeskendir Koishekenov. Reducing over-smoothing in graph neural networks using relational embeddings, 2023.

[65] Yifan Hou, Jian Zhang, James Cheng, Kaili Ma, Richard T. B. Ma, Hongzhi Chen, and Ming-Chang Yang. Measuring and improving the use of graph information in graph neural networks, 2022.

[66] Chaoqi Chen, Yushuang Wu, Qiyuan Dai, Hong-Yu Zhou, Mutian Xu, Sibei Yang, Xiaoguang Han, and Yizhou Yu. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[67] Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9211–9219, 2019.

[68] Qiang Zheng, Yafei Qi, Chen Wang, Chao Zhang, and Jian Sun. Pointvig: A lightweight gnn-based model for efficient point cloud analysis. *arXiv preprint arXiv:2407.00921*, 2024.

[69] Mustafa Munir, William Avery, and Radu Marculescu. Mobilevig: Graph-based sparse attention for mobile vision applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2211–2219, 2023.

[70] Mustafa Munir, William Avery, and Radu Marculescu. Mobilevig: Graph-based sparse attention for mobile vision applications, 2023.

AI-generated, for reference only.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.