Graph Self-Supervised Learning: A Survey

Yixin Liu[1] , **Shirui Pan**[1] , **Ming Jin**[1] , **Chuan Zhou**[2] , **Feng Xia**[3] , **Philip S. Yu**[4]
[1]Department of Data Science & AI, Faculty of IT, Monash University, Australia
[2]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China
[3]School of Engineering, Information Technology and Physical Sciences, Federation University, Australia
[4]Department of Computer Science, University of Illinois at Chicago, USA
{yixin.liu, shirui.pan, ming.jin}@monash.edu, zhouchuan@amss.ac.cn,
f.xia@federation.edu.au, psyu@uic.edu

Abstract

Deep learning on graphs has attracted significant interest recently. However, most of the works have focused on (semi-) supervised learning, resulting in shortcomings including heavy label reliance, poor generalization, and weak robustness. To address these issues, self-supervised learning (SSL), which extracts informative knowledge through well-designed pretext tasks without relying on manual labels, has become a promising and trending learning paradigm for graph data. Different from other domains like computer vision/natural language processing, SSL on graphs has an exclusive background, design ideas, and taxonomies. Under the umbrella of *graph self-supervised learning*, we present a timely and comprehensive review of the existing approaches which employ SSL techniques for graph data. We divide these into four categories according to the design of their pretext tasks. We further discuss the remaining challenges and potential future directions in this research field.

1 Introduction

In recent years, deep learning on graphs [Kipf and Welling, 2017; Hamilton *et al.*, 2017] has become increasingly popular for the artificial intelligence research community since graph-structured data is ubiquitous in numerous domains. In general, most deep learning works on graphs have focused on (semi-) supervised learning scenarios, where the model is trained by a specific downstream task with manual labels. Despite the success of (semi-) supervised graph learning, it still has several shortcomings due to its heavy reliance on labels: the prohibitive cost of accessing ground-truth labels, poor generalization owing to over-fitting, and weak robustness under label-related adversarial attacks [Liu *et al.*, 2020].

Self-supervised learning (SSL) is a promising learning paradigm to address the shortcomings of (semi-) supervised learning. By training the model to solve well-designed *pretext tasks*, SSL helps the model learn more generalized representations from unlabeled data, so it can achieve better performance and generalization on *downstream tasks* [You *et al.*, 2020b]. In SSL, the pretext task is a series of handcrafted auxiliary tasks where the supervision signals are automatically acquired from the data itself instead of manual annotation. As a decisive element in SSL, the design of the pretext tasks often depends on domain-specific knowledge. Following the immense success of SSL on computer vision (CV) and natural language processing (NLP) [Jing and Tian, 2020; Liu *et al.*, 2020], very recently, there has been increasing interest in applying SSL to graph-structured data.

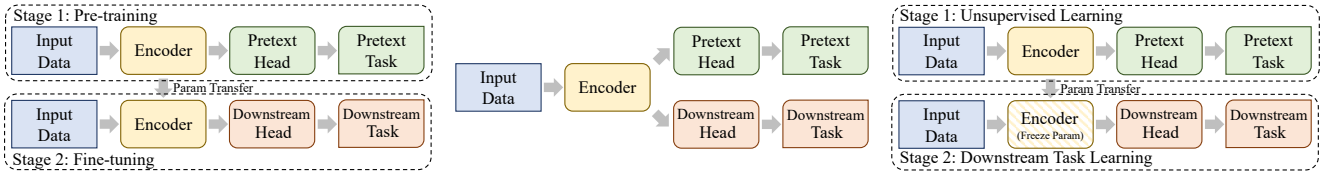
Applying SSL to the graph domain is of great significance and also has significant potential and research prospects. First of all, most works on graph learning overly emphasize the role of labels but ignore the underlying rich structural and attributive information, where various SSL pretext tasks can be designed to mitigate this gap. Moreover, labels on graphs are normally expensive to access, which prevents most of the existing methods to be applied in real-world data. SSL, in contrast, mitigate the reliance on manual labels. Furthermore, the graph domain is more suitable to construct various SSL pretext tasks to acquire supervision signals than in CV/NLP domains since it has a more general and complex data structure in a non-Euclidean space.

In this survey, we review the recent progress in *Graph SSL* [1]. The core contributions of this paper are threefold.

- We provide unified problem formulations and clear definitions to the concepts related to SSL on graphs.
- We provide a timely review and systematically categorize the existing works according to the design of pretext tasks.
- We point out the technical limitations of current research and provide promising directions for future works.

Compared to the existing surveys on SSL [Jing and Tian, 2020; Liu *et al.*, 2020; Jaiswal *et al.*, 2021], our work purely focuses on SSL for graph domains and gives a more scientific and detailed taxonomy according to the characteristics of graphs. In addition, our work pinpoints new challenges for this direction and opens up new directions for both graph learning and self-supervised learning.

[1]In 2020, more than twenty papers that studied graph SSL were published in top international journals/conferences.



(a) Pre-training and Fine-tuning (b) Joint Learning (c) Unsupervised Representation Learning

Figure 1: Pipelines of three types of learning schemes for SSL.

2 Preliminary and Background

In this section, we introduce the preliminaries of graph SSL, including the definitions of different types of graphs, graph neural networks, downstream tasks, and SSL training schemes.

2.1 Notations and Taxonomies of Graphs

Generally, an (unattributed) graph is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of nodes and $\mathcal{E} = \{e_1, \ldots, e_m\}$ is the set of edges, and naturally we have $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The topology of the graph is represented as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where $\mathbf{A}_{i,j} = 1$ means there is a link between nodes v_i and v_j, otherwise $\mathbf{A}_{i,j} = 0$.

Attributed graph, an opposite concept to the unattributed one, refers to a graph where nodes and edges are associated with their own features (a.k.a attributes). Concretely, the feature matrices of nodes and edges are represented as $\mathbf{X_{node}} \in \mathbb{R}^{n \times d_{node}}$ and $\mathbf{X_{edge}} \in \mathbb{R}^{n \times d_{edge}}$ respectively. In a more common scenario, only nodes have features, so we use $\mathbf{X} \in \mathbb{R}^{n \times d}$ to denote the node feature matrix for short. As a special type of attributed graph, a *spatial-temporal graph* can be regarded as an attributed graph with dynamic features at different time steps. Specifically, at each time step t, the dynamic feature matrix is denoted as $\mathbf{X^{(t)}} \in \mathbb{R}^{n \times d}$.

In addition to features, the type of nodes and edges is another dimension for taxonomy. For a graph with more than one type of node or edge, we denote it as a *heterogeneous graph*, otherwise, it is a *homogeneous graph*. There are also some special types of heterogeneous graphs: a *bipartite graph* is a heterogeneous graph with two types of nodes and a single type of edge, while a *multiplex graph* has one type of node and multiple types of edges.

2.2 Downstream Graph Analysis Tasks

We divide downstream tasks into node-, link-, and graph-level tasks. Neural networks often serve as encoders to generate embedding from the input graph in each task. Then, the embedding is fed into the output head to perform specific downstream tasks.

Node-level tasks mainly target the property of nodes in graph(s), so the node representation is indispensable to these tasks. Node classification is a typical node-level task, where each node $v_i \in \mathcal{V}$ has a label y_i. Given the labels of partial nodes for training, the goal is to predict the labels of the rest. A typical output head for node classification is a multi-class classifier with the embedding of node as input.

Link-level tasks often infer the property of edges, and the representation of nodes (in pairs) is the main focus. Taking link prediction as an example, given two nodes, the goal is to discriminate if there is a connection (i.e., edge) between them. An output head could be a binary classifier with the embeddings of two nodes as input.

Graph-level tasks learn from multiple graphs in a dataset and predict the property of a single graph. Therefore, these tasks often rely on the representation of graphs. For instance, in the graph classification task, each graph $\mathcal{G}_i$ has its label y_i, and the objective is to train a model to predict the labels of the input graphs. A general solution is to aggregate the node embeddings into a graph embedding via a readout function and feed the graph embedding into the classifier.

2.3 Graph Neural Networks

Graph neural networks (GNNs) [Kipf and Welling, 2017; Veličković *et al.*, 2018; Xu *et al.*, 2019] are a family of neural networks that have been widely applied to graph analysis tasks recently. Since GNNs are the backbone encoder for most of the reviewed works in this paper, we introduce a general GNN framework in this subsection.

Given an attributed graph $\mathcal{G}$ with its feature matrix $\mathbf{X}$ where $\mathbf{x_i} = \mathbf{X}[i,:]^T$ is a d-dimensional feature vector of the node v_i, the goal of GNNs is to learn a node representation $\mathbf{h_i}$ for each node $v_i \in \mathcal{V}$. Considering a K-layer GNN, the formulation of the k-th layer is represented as:

$$\mathbf{a_i^{(k)}} = \mathrm{AGGREGATE}^{(k)}\left(\left\{\mathbf{h_j^{(k-1)}} : v_j \in \mathcal{N}(v_i)\right\}\right),$$
$$\mathbf{h_i^{(k)}} = \mathrm{COMBINE}^{(k)}\left(\mathbf{h_i^{(k-1)}}, \mathbf{a_i^{(k)}}\right), \tag{1}$$

where $\mathbf{h_i^{(k)}}$ is the latent vector of node v_i at the k-th iteration/layer with $\mathbf{h_i^{(0)}} = \mathbf{x_i}$ and $\mathbf{h_i^{(K)}} = \mathbf{h_i}$, $\mathcal{N}(v_i)$ is a set of nodes adjacent to v_i, $\mathrm{AGGREGATE}^{(k)}(\cdot)$ and $\mathrm{COMBINE}^{(k)}(\cdot)$ are component functions of the GNN layer.

For node-level tasks, the node representation $\mathbf{h_i}$ is used for downstream tasks directly. For graph-level tasks, an extra readout function that aggregates node features to obtain the entire graph's representation $\mathbf{h_\mathcal{G}}$ is needed:

$$\mathbf{h_\mathcal{G}} = \mathrm{READOUT}\left(\left\{\mathbf{h_i}^{(K)} \mid v_i \in \mathcal{V}\right\}\right) \tag{2}$$

The design of component functions is crucial to different GNNs. For the sake of space, here we do not discuss the operations of specific GNNs. For a thorough review, we refer the reader to the recent survey [Wu *et al.*, 2020].

2.4 Self-Supervised Training Schemes

According to the relationship among graph encoders, self-supervised pretext tasks, and downstream tasks, we investigate three types of self-supervised training schemes. The brief pipelines are given in Figure 1.

Pre-training and Fine-tuning (PT&FT) In the PT&FT scheme, the encoder is first pre-trained with pretext tasks, which can be viewed as an initialization for the encoder's parameters. After this, the pre-trained encoder is fine-tuned together with a prediction head under the supervision of specific downstream tasks.

Joint Learning (JL) In the JL scheme, the encoder is jointly trained with the pretext and downstream tasks. The loss function consists of both the self-supervised and downstream task loss functions, where a trade-off hyper-parameter controls their contribution. This can be considered as a kind of multi-task learning or the pretext task is served as a regularization of the downstream task.

Unsupervised Representation Learning (URL) The first stage of the URL scheme is similar to that of PT&FT. The difference is, in the second stage, the encoder's parameters are frozen when the model is trained with the downstream task. Compared with other schemes, URL is more challenging since there is no supervision during the encoder training.

3 Methods of Graph Self-Supervised Learning

The design of the pretext tasks is the core of Graph SSL. In this section, we divide the existing works into four categories by their underlying motivations of the pretext task design, whose sketch maps are given in Figure 2. A brief summary of all the surveyed works is given in Table 1.

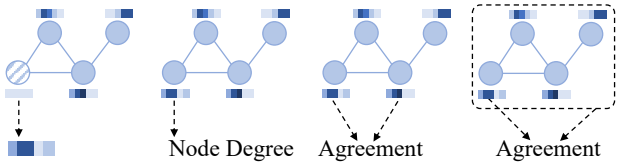
3.1 Masked Feature Regression (MFR)

This branch of approaches is motivated by image inpainting in computer vision [Yu *et al.*, 2018], which aims to fill the masked pixels of an image. For input graph data, the features of nodes and/or edges are masked with zero or certain tokens. Then, the objective is to recover the masked features according to the unmasked information via GNNs.

You et al. [2020b] first defined masked node feature regression as Graph Completion and the intuition is to enable GNN to extract features from the context. Jin et al. [2020] propose AttributeMask, which aims to reconstruct the dense feature matrix processed by Principle Component Analysis (PCA). Hu et al. [2020a] present AttrMasking, which replaces the edge and node attributes with special masks and then forces the GNNs to rebuild them simultaneously. Manessi et al. [2020] present three reconstruction tasks, e.g., reconstructing raw features from clean inputs, reconstructing raw features from corrupted inputs, and reconstructing embeddings from corrupted embeddings, which train the encoder in a joint learning way.

3.2 Auxiliary Property Prediction (APP)

Apart from MFR, the underlying graph structural and attributive information can be further explored to construct diverse pretext tasks and provide self-supervision signals. We refer to this branch of methods as auxiliary property prediction, including regression- and classification-based approaches.



(a) MFR (b) APP (c) SSC (d) CSC

Figure 2: Sketch maps for four types of graph SSL pretext tasks.

Regression-based Approach (R-APP)

Similar to but different from MFR, the underlying idea of this branch of methods primarily lies in predicting other extensive numerical structural and attributive properties within graphs. To extract rich self-supervision signals by leveraging the inherent graph structure, Jin et al. [2020] first consider predicting some representative node properties, such as node degree, to form a local-structure aware pretext task called NodeProperty. In contrast to node-specific properties, global structural information will not be ignored as well. Distance2Cluster [Jin *et al.*, 2020] predicts the distance between unlabeled nodes and pre-defined clusters in the graph, which encourages the node representation to be aware of its global positioning while training. PairwiseAttrSim [Jin *et al.*, 2020] enforces the similarity of a pair of nodes to be close to their feature similarity on a set of representative node pairs (i.e., with the highest or lowest attribute similarity). The intuition behind this is to enhance feature transformation in the local structure while avoiding over-smoothing.

Classification-based Approach (C-APP)

In contrast to R-APP, classification-based approaches typically rely on constructing pseudo labels to facilitate model training. For methods built on the idea of clustering, M3S [Sun *et al.*, 2020] iteratively trains an encoder by leveraging DeepCluster [Caron *et al.*, 2018] to assign pseudo labels to those unlabeled nodes at each training stage. Similar to M3S, You et al. [2020b] propose Node Clustering to use the indices of pre-computed clusters as self-supervised labels, which is similar to the Cluster Preserving introduced by Hu et al. [2019]. In addition to feature-based clustering, You et al. [2020b] introduce another structural-aware task named Graph Partitioning to group nodes based on the inherent topology where the connections across subsets are minimized. After this, partition indices are assigned as node labels in different subsets as in Node Clustering. To absorb the advantages of both attributive- and structural-based clustering, CAGNN [Zhu *et al.*, 2020b] first assigns cluster indices as pseudo labels and then refines the clusters by minimizing the inter-cluster edges.

Different from the clustering-based approaches, graph property prediction is another promising way to provide extra self-supervision. GROVER [Rong *et al.*, 2020] considers two learning tasks named contextual property and motif prediction. The first task aims to predict the statistical properties of anchor nodes' subgraphs (e.g., node-edge-counts), while the

Approach	Pretext Task Category	Data Type of Graph	Downstream Task level	Training Scheme	Encoder
Graph Completion [You *et al.*, 2020b]	MFR	Attributed	Node	PT&FT/JL	GCN
AttributeMask [Jin *et al.*, 2020]	MFR	Attributed	Node	PT&FT/JL	GCN
AttrMasking [Hu *et al.*, 2020a]	MFR	Attributed	Node	PT&FT	GCN
[Manessi and Rozza, 2020]	MFR	Attributed	Node	JL	GCN
NodeProperty [Jin *et al.*, 2020]	R-APP	Attributed	Node	PT&FT/JL	GCN
Distance2Cluster [Jin *et al.*, 2020]	R-APP	Attributed	Node	PT&FT/JL	GCN
PairwiseAttrSim [Jin *et al.*, 2020]	R-APP	Attributed	Node	PT&FT/JL	GCN
M3S [Sun *et al.*, 2020]	C-APP	Attributed	Node	JL	GCN
Node Clustering [You *et al.*, 2020b]	C-APP	Attributed	Node	PT&FT/JL	GCN
Cluster Preserving [Hu *et al.*, 2019]	C-APP	Attributed	Node/Link/Graph	PT&FT	GCN
Graph Partitioning [You *et al.*, 2020b]	C-APP	Attributed	Node	PT&FT/JL	GCN
CAGNN [Zhu *et al.*, 2020b]	C-APP	Attributed	Node	URL	GCN
GROVER [Rong *et al.*, 2020]	C-APP	Attributed	Node/Link/Graph	PT&FT	Transformer
Centrality Score Ranking [Hu *et al.*, 2019]	C-APP	Attributed	Node/Link/Graph	PT&FT	GCN
DeepWalk [Perozzi *et al.*, 2014]	C-SSC	Unattributed	Node	URL	Shallow NN
node2vec [Grover and Leskovec, 2016]	C-SSC	Unattributed	Node	URL	Shallow NN
GraphSAGE [Hamilton *et al.*, 2017]	C-SSC	Attributed	Node	URL	SAGE$^{\#}$
LINE [Tang *et al.*, 2015]	C-SSC	Unattributed	Node	URL	Shallow NN
GAE/VGAE [Kipf and Welling, 2016]	C-SSC	Attributed	Link	URL	GCN
EdgeMask [Jin *et al.*, 2020]	C-SSC	Attributed	Node	PT&FT/JL	GCN
Denoising Link Reconstruction [Hu *et al.*, 2019]	C-SSC	Attributed	Node/Link/Graph	PT&FT	GCN
S^2GRL [Peng *et al.*, 2020a]	C-SSC	Attributed	Node/Link	URL	GCN
SuperGAT [Kim and Oh, 2021]	C-SSC	Attributed	Node	JL	SuperGAT$^{\#}$
SELAR [Hwang *et al.*, 2020]	C-SSC	Heterogeneous	Node/Link	JL	GNNs (5)
GCC [Qiu *et al.*, 2020]	A-SSC	Unattributed	Node/Graph	URL/PT&FT	GIN
GRACE [Zhu *et al.*, 2020a]	A-SSC	Attributed	Node	URL	GCN
GCA [Zhu *et al.*, 2021]	A-SSC	Attributed	Node	URL	GCN
GraphCL [You *et al.*, 2020a]	A-SSC	Attributed	Graph	URL	GCN
CSSL [Zeng and Xie, 2021]	A-SSC	Attributed	Graph	URL/PT&FT/JL	HGP-SL
IGSD [Zhang *et al.*, 2020a]	A-SSC	Attributed	Graph	URL/JL	GCN/GIN
DGI [Veličković *et al.*, 2018]	CSC	Attributed	Node	URL	GCN/SAGE
MVGRL [Hassani and Khasahmadi, 2020]	CSC	Attributed	Node	URL	GCN
Subg-Con [Jiao *et al.*, 2020]	CSC	Attributed	Node	URL	GCN
Context Prediction [Hu *et al.*, 2020a]	CSC	Attributed	Node	PT&FT	GCN
GIC [Mavromatis and Karypis, 2020]	CSC	Attributed	Node/Link	URL	GCN
InfoGraph [Sun *et al.*, 2019]	CSC	Attributed	Graph	URL	GIN
MICRO-Graph [Zhang *et al.*, 2020c]	CSC	Attributed	Graph	URL	DeeperGCN
SUGAR [Sun *et al.*, 2021]	CSC	Attributed	Graph	JL	GCN
HDGI [Ren *et al.*, 2020]	CSC	Heterogeneous	Node	URL	GCN/GAT
SLiCE [Wang *et al.*, 2021]	CSC	Heterogeneous	Link	PT&FT	Shallow NN
DMGI [Park *et al.*, 2020]	CSC	Multiplex	Node/Link	URL	GCN
BiGI [Cao *et al.*, 2021]	CSC	Bipartite	Node/Link	URL	BiGI-GNN$^{\#}$
STDGI [Opolka *et al.*, 2019]	CSC	Spatial-temporal	Node	URL	GCN
GPT-GNN [Hu *et al.*, 2020b]	Hybrid	Heterogeneous	Node/Link	PT&FT	HGT
GMI [Peng *et al.*, 2020b]	Hybrid	Attributed	Node/Link	URL	GCN
Graph-Bert [Zhang *et al.*, 2020b]	Hybrid	Attributed	Node	PT&FT	Transformer
CG^3 [Wan *et al.*, 2021]	Hybrid	Attributed	Node	JL	GCN/HGCN

Table 1: Summary of the surveyed papers. The classification is organized along the underlying motivations of the pretext tasks. We also list other properties of the reviewed methods, including data type, downstream task level, training scheme, and graph encoder. In the last column, the marker $^{\#}$ indicates that the encoder is also proposed by this paper, and the item "GNNs (5)" means that this paper experiments with 5 types of GNNs, namely GCN, GAT, GIN, SGC, and HGN.

second task is to identify whether a graph contains specific pre-determined motifs by providing graph embedding. Centrality Score Ranking [Hu *et al.*, 2019] leverages node centrality (e.g., eigencentrality and subgraph centrality) to rank nodes and estimate the ranking score for any two nodes in the graph to preserve different-level graph information.

3.3 Same-Scale Contrasting (SSC)

Different from the aforementioned two types of methods which build tasks on a single element (e.g. a single node), contrastive learning methods learn by predicting the agreement between two elements in a graph. Specifically, the agreement between samples with similar semantic informa-

tion (denoted as a positive pair) is maximized, while those with unrelated semantic information (denoted as a negative pair) are minimized. As subdivision methods, SSC contrasts two graph elements in a similar or equal scale, e.g., node-node and graph-graph contrasting. Figure 2(c) shows a toy example for node-node contrasting. Here, we further divide the SSC approaches into two categories according to their definition of positive/negative pairs.

Context-Based Approaches (C-SSC)

The main idea of the C-SSC approach is to pull the contextual nodes closer in the embedding space. The contextual nodes often have adjacent geometric positions in the graph structure. The intuition behind such a definition is the Homophily hypothesis [McPherson *et al.*, 2001], i.e., entities in the graph with similar semantic information are likely to interconnect.

An efficient way to define context is to rely on random walk to generate sequences of nodes as similar semantic sets. The node pairs which stay close in a walk are denoted as positive pairs, while negative pairs are acquired by the negative sampling strategy. DeepWalk [Perozzi *et al.*, 2014] is a classic node embedding method for unattributed graphs. It samples random walks as "sequences", and learns low-dimensional node representation based on the Skip-Gram model [Mikolov *et al.*, 2013a; Mikolov *et al.*, 2013b]. On the basis of DeepWalk, node2vec [Grover and Leskovec, 2016] uses a biased random walk procedure to efficiently explore diverse neighbors as contexts. While the aforementioned methods are for unattributed graphs and utilize shallow neural networks as their encoder, GraphSAGE [Hamilton *et al.*, 2017] is an advanced approach for representation learning on attributed graphs. A novel type of GNN (here we denote as SAGE) is designed for node encoding, and a random walk-based sampling strategy is employed to generate context for each node.

Instead of sampling contextual nodes with random walk, a simpler way is to utilize the graph structure directly. For each node, we can set its adjacent k-hop neighbors as positive samples, while the negative samples are acquired from the remainder. Note that this solution can also be regarded as a reconstruction of graph structure, as proposed in [Liu *et al.*, 2020]. LINE [Tang *et al.*, 2015] considers both first-order and second-order neighbors as context information to jointly optimize the node embedding. GAE/VGAE [Kipf and Welling, 2016] and EdgeMask [Jin *et al.*, 2020] consider the one-hop neighbors as the positive contexts of each node, and the negative examples are sampled by randomly negative sampling. The Denoising Link Reconstruction in [Hu *et al.*, 2019] has a similar learning objective. S^2GRL [Peng *et al.*, 2020a] builds k decoders to contrast a node with its 1-hop to k-hop neighbors respectively, and these decoders are learned jointly. SuperGAT [Kim and Oh, 2021] carries out structure reconstruction on the latent outputs of every layer to ensure that the graph attention units learn the correct attention weights. On heterogeneous graphs, SELAR [Hwang *et al.*, 2020] first generates meta-paths and then facilitates the primary task by introducing an auxiliary task to predict various meta-paths with node embeddings.

Augmentation-Based Approaches (A-SSC)

Motivated by recent breakthroughs in contrastive visual feature learning [He *et al.*, 2020; Chen *et al.*, 2020], A-SSC generates augmented examples of original data samples, and views two augmented examples which are from the same original sample as a positive pair, while those from different original samples are negatives. The inherent contrasting mechanism of these methods is on top of the *mutual information (MI) estimation* [Hjelm *et al.*, 2019] with *InfoNCE* as their estimator [Oord *et al.*, 2018]. For A-SSC, the definition of data augmentation is the most significant factor.

For node-level tasks, GCC [Qiu *et al.*, 2020] focuses on universal unattributed graphs. It samples subgraphs with Random Walk with Restart for each node as augmentations, and the node features are artificially-designed positional node embedding. GRACE [Zhu *et al.*, 2020a] adapts two augmentation strategies, removing edges and masking node features, to generate augmented views of graph data. It jointly considers both intra-view and inter-view negative pairs for contrast purposes. GCA [Zhu *et al.*, 2021] further improves the augmentation strategies of GRACE into adaptive augmentation.

For graph-level tasks, GraphCL [You *et al.*, 2020a] considers four graph-level augmentations: node dropping, edge perturbation, attribute masking and subgraph sampling. A SimCLR-like [Chen *et al.*, 2020] framework is utilized to contrast graph examples from different augmentations. CSSL [Zeng and Xie, 2021] augments examples with the addition/deletion operation of nodes/edges, and studies the performance under three types of training schemes. IGSD [Zhang *et al.*, 2020a] adopts graph diffusion [Klicpera *et al.*, 2019] on input graphs to generate augmented views and leverages the idea of BYOL [Grill *et al.*, 2020] to contrast anchor graphs with other instances in a Siamese network. As a structural augmentation strategy, graph diffusion provides the global views of the underlying structure where the rich global information can be encoded during SSL.

3.4 Cross-Scale Contrasting (CSC)

Different from the SSC methods, this type of approach learns representations by contrasting the elements in different scales of graph data, e.g., node-subgraph and node-graph contrasting (as Figure 2(d) shown). A readout function is usually adopted to acquire the summary for a graph/subgraph. Similar to A-SSC, most of these methods inherent the idea of MI maximization but leverage the *Jensen-Shannon divergence* as their MI estimator. [Hjelm *et al.*, 2019].

DGI [Veličković *et al.*, 2018] is one of the most representative works. It learns node representation by maximizing the MI between patch representations and corresponding high-level summary of graphs. To generate negative samples on a single graph, DGI corrupts the original graph by randomly scrambling node features while keeping the structure unchanged. Following DGI, MVGRL [Hassani and Khasahmadi, 2020] suggests a multi-view contrasting where the original graph structure and graph diffusion are regarded as two different views. The target is the MI maximization between the cross-view representations of nodes and large-scale graphs. Subg-Con [Jiao *et al.*, 2020] contrasts the embedding of a node and its context subgraphs to learn re-

gional graph structure information. Context Prediction [Hu *et al.*, 2020a] enhances the agreement between nodes and context subgraphs whose embedding is acquired by an auxiliary GNN. With the help of a differentiable clustering layer, GIC [Mavromatis and Karypis, 2020] maximizes the intra-cluster MI which is denoted as the agreement between nodes and their corresponding cluster centroids. InfoGraph [Sun *et al.*, 2019] extends such an idea to graph-level tasks, which maximizes the MI between graph representations and substructures at different levels. MICRO-Graph [Zhang *et al.*, 2020c] employs EM clustering to learn meaningful motifs iteratively, where the objective is to enhance the embedding similarity between a graph and its motifs. SUGAR [Sun *et al.*, 2021] pulls the local subgraph representations and the global graph representation closer to discriminate the subgraph representations among graphs, which benefits the reinforcement pooling mechanism.

The idea of CSC can be expanded to different types of graphs. [Ren *et al.*, 2020] presents a local-global MI maximization for heterogeneous graph representation learning, where the local representation is an attention-based combination of nodes within a meta-path, and the global representation is the summary of local representation. Similarly, SLiCE [Wang *et al.*, 2021] introduces a contextual node prediction task that maximizes the occurrence probability between a context graph and a randomly masked node within this subgraph. To learn the representation for multiplex graphs, DMGI [Park *et al.*, 2020] maximizes the agreements of specific node embedding and graph-level summary representation for each relation graph respectively. For bipartite graph embedding, BiGI [Cao *et al.*, 2021] also adopts the local-global MI maximization. The global representation is the concatenation of two prototype representations, while the local representation is an attention-based k-hop subgraph aggregation for two nodes of each edge. STDGI [Opolka *et al.*, 2019] extends the idea of contrastive learning to spatial-temporal graphs. By maximizing the agreement of node embedding in time step t and the raw features of the same node at a future time step $t + k$, the encoder can capture the information that is relevant for predicting its future features.

3.5 Hybrid Self-supervised Learning

Rather than utilizing a single task, some methods combine different types of pretext tasks in a multi-task learning fashion to better leverage their advantages. We name these methods hybrid self-supervised learning.

GPT-GNN [Hu *et al.*, 2020b] integrates MFR and dropped edge prediction (C-SSC) into a graph generation task to pre-train the GNN. GMI [Peng *et al.*, 2020b] proposes to jointly maximize feature MI (between the node's embedding and raw features of its neighbors) and edge MI (embedding of two adjacent nodes) for graph representation learning. Graph-Bert [Zhang *et al.*, 2020b] leverages node feature reconstruction (MFR) and graph structure recovery (C-SSC) to pre-train a graph transformer model. CG^3 [Wan *et al.*, 2021] considers augmentation-based and context-based SSCs as self-supervised learning signals which are simultaneously learned with the downstream node classification task.

4 Challenges and Future Directions

In this section, we analyze the existing challenges in Graph SSL and pinpoint a few future research directions aiming to address the shortcomings.

4.1 Theoretical Foundation for Graph SSL

Despite the great success of SSL in various domains, it still lacks a theoretical foundation. The existing methods are mostly designed with intuition and their performance gain is evaluated by empirical experiments. Although MI estimation theory [Hjelm *et al.*, 2019] supports some of the works on contrastive learning, the choice of the MI estimator still relies on empirical study [Hassani and Khasahmadi, 2020]. Setting up a solid theoretical foundation for graph SSL is urgently needed. It is promising to bridge the gap between empirical SSL and a series of graph theories, including graph signal processing and spectral graph theory.

4.2 Augmentation for Graph Contrastive Learning

In recent breakthroughs in visual contrastive learning [Chen *et al.*, 2020], data augmentation is essential to maintaining the representation invariance during contrastive learning. Due to the nature of graph-structured data (e.g., complex and non-Euclidean structure), data augmentation schemes on graphs are not often explored and thus compromise the effectiveness of graph augmentation-based approaches as discussed in Section 3.3. Most of the existing graph augmentations consider uniformly shuffling node features, dropping edges, or other alternative ways like subgraph sampling and graph diffusion [Klicpera *et al.*, 2019]. To open future research directions, adaptively performing graph augmentations [Zhu *et al.*, 2021] or jointly considering stronger augmented samples [Wang and Qi, 2021] by mining the rich underlying structural- and attributive-information could be promising approaches.

4.3 Pretext Tasks for Complex Types of Graph

As shown in Table 1, most of the current works concentrate on SSL for attributed graphs, and few focuses on complex graph types, e.g., heterogeneous or spatial-temporal graphs. The main challenge is that the pretext task design needs domain knowledge on specific graph data types. Most of the existing methods apply the idea of MI maximization [Veličković *et al.*, 2018] to complex graph learning, which is limited in its ability to leverage rich information from data. A future opportunity is to produce various SSL tasks for complex graph data, where specific data characteristics are the main focus. Furthermore, extending SSL to more ubiquitous graph types (e.g., dynamic or hyper- graphs) is also a promising direction.

5 Conclusion

In this work, we presented a survey on the topic of self-supervised learning on graph-structured data. We first introduce the related preliminary definitions exhaustively. We review the recent works and categorize them with a systematic taxonomy. More importantly, we dive deeper into the research topic and unveil the critical challenges and analyze the possible directions in the future. We believe that graphs SSL will continue to be an active and promising research area with broad potential applications.

References

[Cao *et al.*, 2021] Jiangxia Cao, Xixun Lin, Shu Guo, Luchen Liu, Tingwen Liu, and Bin Wang. Bipartite graph embedding via mutual information maximization. In *WSDM*, 2021.

[Caron *et al.*, 2018] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119, pages 1597–1607. PMLR, 13–18 Jul 2020.

[Grill *et al.*, 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864, 2016.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034, 2017.

[Hassani and Khasahmadi, 2020] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, volume 119, pages 4116–4126. PMLR, 13–18 Jul 2020.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[Hjelm *et al.*, 2019] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[Hu *et al.*, 2019] Ziniu Hu, Changjun Fan, Ting Chen, Kai-Wei Chang, and Yizhou Sun. Pre-training graph neural networks for generic structural feature extraction. *arXiv preprint arXiv:1905.13728*, 2019.

[Hu *et al.*, 2020a] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *ICLR*, 2020.

[Hu *et al.*, 2020b] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *KDD*, pages 1857–1867, 2020.

[Hwang *et al.*, 2020] Dasol Hwang, Jinyoung Park, Sunyoung Kwon, KyungMin Kim, Jung-Woo Ha, and Hyunwoo J Kim. Self-supervised auxiliary learning with meta-paths for heterogeneous graphs. In *NeurIPS*, volume 33, 2020.

[Jaiswal *et al.*, 2021] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021.

[Jiao *et al.*, 2020] Yizhu Jiao, Yun Xiong, Jiawei Zhang, Yao Zhang, Tianqi Zhang, and Yangyong Zhu. Sub-graph contrast for scalable self-supervised graph representation learning. *arXiv preprint arXiv:2009.10273*, 2020.

[Jin *et al.*, 2020] 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. *arXiv preprint arXiv:2006.10141*, 2020.

[Jing and Tian, 2020] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE TPAMI*, 2020.

[Kim and Oh, 2021] Dongkwan Kim and Alice Oh. How to find your friendly neighborhood: Graph attention design with self-supervision. In *ICLR*, 2021.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Variational graph auto-encoders. In *NeurIPS Workshop*, 2016.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Klicpera *et al.*, 2019] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In *NeurIPS*, pages 13354–13366, 2019.

[Liu *et al.*, 2020] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218*, 1(2), 2020.

[Manessi and Rozza, 2020] Franco Manessi and Alessandro Rozza. Graph-based neural network models with multiple self-supervised auxiliary tasks. *arXiv preprint arXiv:2011.07267*, 2020.

[Mavromatis and Karypis, 2020] Costas Mavromatis and George Karypis. Graph infoclust: Leveraging cluster-level node information for unsupervised graph representation learning. *arXiv preprint arXiv:2009.06946*, 2020.

[McPherson *et al.*, 2001] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *NeurIPS*, 26:3111–3119, 2013.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Opolka *et al.*, 2019] Felix L Opolka, Aaron Solomon, Cătălina Cangea, Petar Veličković, Pietro Liò, and R Devon Hjelm. Spatio-temporal deep graph infomax. In *ICLR Workshop*, 2019.

[Park *et al.*, 2020] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. Unsupervised attributed multiplex network embedding. In *AAAI*, pages 5371–5378, 2020.

[Peng *et al.*, 2020a] Zhen Peng, Yixiang Dong, Minnan Luo, Xiao-Ming Wu, and Qinghua Zheng. Self-supervised graph representation learning via global context prediction. *arXiv preprint arXiv:2003.01604*, 2020.

[Peng *et al.*, 2020b] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization. In *WWW*, pages 259–270, 2020.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710, 2014.

[Qiu *et al.*, 2020] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*, pages 1150–1160, 2020.

[Ren *et al.*, 2020] Yuxiang Ren, Bo Liu, Chao Huang, Peng Dai, Liefeng Bo, and Jiawei Zhang. Hdgi: An unsupervised graph neural network for representation learning in heterogeneous graph. In *AAAI Workshop*, 2020.

[Rong *et al.*, 2020] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *NeurIPS*, 33, 2020.

[Sun *et al.*, 2019] Fan-Yun Sun, Jordan Hoffman, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2019.

[Sun *et al.*, 2020] Ke Sun, Zhouchen Lin, and Zhanxing Zhu. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *AAAI*, pages 5892–5899, 2020.

[Sun *et al.*, 2021] Qingyun Sun, Hao Peng, Jianxin Li, Jia Wu, Yuanxing Ning, Phillip S Yu, and Lifang He. Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In *WWW*, 2021.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[Veličković *et al.*, 2018] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *ICLR*, 2018.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Wan *et al.*, 2021] Sheng Wan, Shirui Pan, Jian Yang, and Chen Gong. Contrastive and generative graph convolutional networks for graph-based semi-supervised learning. In *AAAI*, 2021.

[Wang and Qi, 2021] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations, 2021.

[Wang *et al.*, 2021] Ping Wang, Khushbu Agarwal, Colby Ham, Sutanay Choudhury, and Chandan K Reddy. Self-supervised learning of contextual embeddings for link prediction in heterogeneous networks. In *WWW*, 2021.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE TNNLS*, 2020.

[Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[You *et al.*, 2020a] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *NeurIPS*, 33, 2020.

[You *et al.*, 2020b] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. When does self-supervision help graph convolutional networks? In *ICML*, pages 10871–10880. PMLR, 2020.

[Yu *et al.*, 2018] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, pages 5505–5514, 2018.

[Zeng and Xie, 2021] Jiaqi Zeng and Pengtao Xie. Contrastive self-supervised learning for graph classification. In *AAAI*, 2021.

[Zhang *et al.*, 2020a] Hanlin Zhang, Shuai Lin, Weiyang Liu, Pan Zhou, Jian Tang, Xiaodan Liang, and Eric P Xing. Iterative graph self-distillation. *arXiv preprint arXiv:2010.12609*, 2020.

[Zhang *et al.*, 2020b] Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*, 2020.

[Zhang *et al.*, 2020c] Shichang Zhang, Ziniu Hu, Arjun Subramonian, and Yizhou Sun. Motif-driven contrastive learning of graph representations. *arXiv preprint arXiv:2012.12533*, 2020.

[Zhu *et al.*, 2020a] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep Graph Contrastive Representation Learning. In *ICML Workshop*, 2020.

[Zhu *et al.*, 2020b] Yanqiao Zhu, Yichen Xu, Feng Yu, Shu Wu, and Liang Wang. Cagnn: Cluster-aware graph neural networks for unsupervised graph representation learning. *arXiv preprint arXiv:2009.01674*, 2020.

[Zhu *et al.*, 2021] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *WWW*, 2021.