Curriculum Graph Machine Learning: A Survey

Haoyang Li[1] , **Xin Wang**[1,2] and **Wenwu Zhu**[1,2]
[1]Department of Computer Science and Technology, Tsinghua University
[2]Beijing National Research Center for Information Science and Technology, Tsinghua University
lihy18@mails.tsinghua.edu.cn, {xin_wang, wwzhu}@tsinghua.edu.cn

Abstract

Graph machine learning has been extensively studied in both academia and industry. However, in the literature, most existing graph machine learning models are designed to conduct training with data samples in a random order, which may suffer from suboptimal performance due to ignoring the importance of different graph data samples and their training orders for the model optimization status. To tackle this critical problem, curriculum graph machine learning (Graph CL), which integrates the strength of graph machine learning and curriculum learning, arises and attracts an increasing amount of attention from the research community. Therefore, in this paper, we comprehensively overview approaches on Graph CL and present a detailed survey of recent advances in this direction. Specifically, we first discuss the key challenges of Graph CL and provide its formal problem definition. Then, we categorize and summarize existing methods into three classes based on three kinds of graph machine learning tasks, i.e., node-level, link-level, and graph-level tasks. Finally, we share our thoughts on future research directions. To the best of our knowledge, this paper is the first survey for curriculum graph machine learning.

1 Introduction

Graph structured data is ubiquitous in the real world, which has been widely used to model the complex relationships and dependencies among various entities. In the past decade, graph machine learning approaches, especially graph neural networks (GNNs) [Kipf and Welling, 2017; Veličković *et al.*, 2018; Hamilton *et al.*, 2017], have drawn ever-increasing attention in both academia and industry, which make great progress in a variety of applications across wide-ranging domains, ranging from physics [Sanchez-Gonzalez *et al.*, 2020] to chemistry [Gilmer *et al.*, 2017], and from neuroscience [de Vico Fallani *et al.*, 2014] to social science [Zhang and Tong, 2016]. Other areas, such as recommender systems [Wu *et al.*, 2022], knowledge graphs [Wang *et al.*, 2017], molecular prediction [Hu *et al.*, 2020a], medical detection [Horry *et al.*, 2020] etc., also provide an increasing demand for the applications of graph machine learning.

Despite the popularity of graph machine learning approaches, the existing literature generally trains graph models by feeding the data samples in a random order during the training process. For example, when training GNNs, the widely adopted mini-batch stochastic gradient descent optimization strategy as well as its variants select the data samples in each mini-batch randomly. However, such training strategies largely ignore the importance of different graph data samples and how their orders can affect the optimization status, which may result in suboptimal performance of the graph learning models [Wei *et al.*, 2022; Wang *et al.*, 2021b]. Typically, humans tend to learn much better when the data examples are organized in a meaningful order rather than randomly presented, e.g., learning from basic easy concepts to advanced hard concepts resembling the "curriculum" taught in schools [Elman, 1993; Rohde and Plaut, 1999]. To this end, curriculum learning (CL) [Bengio *et al.*, 2009; Wang *et al.*, 2021a; Soviany *et al.*, 2022] is proposed to mimic human's learning process, and has been proved to be effective in boosting the model performances as well as improving the generalization capacity and convergence of learning models in various scenarios including computer vision [Guo *et al.*, 2018; Jiang *et al.*, 2014], natural language processing [Platanios *et al.*, 2019; Tay *et al.*, 2019] etc.

Curriculum graph machine learning (Graph CL), which combines the strength of graph machine learning and curriculum learning, has become a promising research direction and attracted an increasing number of interests from the community, spanning over a variety of graph learning methodologies and applications recently. Facing opportunities as well as challenges, we believe it is the right time to review and promote the studies of Graph CL approaches. In this survey, we provide a comprehensive and systematic review of Graph CL. Specifically, we first analyze the key challenges of Graph CL and introduce the basic formulations. Then, we summarize the existing methodologies into three categories based on the granularity of graph tasks, i.e., node-level, link-level, and graph-level tasks, and elaborate representative approaches in each category. Last but not least, we discuss potential future research topics, which could shed light on the development of this promising area. We deeply hope this survey may provide insights to promote the Graph CL research in the community.

2 Challenges and Problem Formulation

Curriculum graph machine learning (Graph CL), as an emerging research topic in the machine learning community, which non-trivially combines the strength of graph machine learning and curriculum learning, faces the following challenges.

- **Uniqueness of curriculum graph learning problem.** Unlike image or text, graph data lies in a non-Euclidean space. Besides, there exist complex relationships and dependencies between entities in graphs. Therefore, the non-Euclidean nature and complexity of graph-structured data bring unique challenges in tackling Graph CL problem.
- **Complexity of curriculum graph learning method.** Although general CL algorithms [Wang *et al.*, 2021a; Soviany *et al.*, 2022] have been extensively studied, it is non-trivial to combine the advantages of graph machine learning and curriculum learning into one unified framework, especially when customizing the design of the key CL components such as the difficulty measurer and the training scheduler that are compatible with graph models.
- **Diversity of curriculum graph learning task.** Curriculum graph learning tasks range from node-level and link-level to graph-level problems with different settings, objectives, constraints, and domain knowledge. Therefore, it is important yet difficult to develop graph CL approaches tailored for different graph tasks or applications.

The methodologies reviewed in this paper target on dealing with at least one of these three key challenges. Before introducing the problem formulation of curriculum graph machine learning (Graph CL), we briefly describe the background of graph machine learning and curriculum learning.

2.1 Graph Machine Learning

Existing graph machine learning models can be generally divided into the following two groups: network embedding (NE) [Cui *et al.*, 2018] and graph neural networks (GNNs) [Zhang *et al.*, 2020]. Specifically, GNNs as the current state-of-the-art in graph machine learning, have been widely adopted to serve as the backbone of a graph curriculum learning method.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with the node set $\mathcal{V}$ and the edge (link) set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. GNNs first learn node representations by the following message-passing [Gilmer *et al.*, 2017; Zhang *et al.*, 2022a] function:

$$\mathbf{h}_v^{(l+1)} = \mathrm{COM}\left(\mathbf{h}_v^{(l)}, \left[\mathrm{AGG}\left(\left\{\mathbf{h}_u^{(l)} \mid u \in \mathcal{N}_v\right\}\right)\right]\right), \quad (1)$$

where $\mathbf{h}_v^{(l)}$ is the node representation of node v at the l^{th} layer, which is usually initialized as node feature at the first layer. $\mathcal{N}_v$ denotes the neighbors of node v. $\mathrm{AGG}(\cdot)$ and $\mathrm{COM}(\cdot)$ denote the aggregation and combination function of GNNs [Wu *et al.*, 2020a]. After deriving the node representations, the link representations can be obtained based on the representations of the two connected nodes. Furthermore, graph-level representations can be computed by the readout (or pooling) [Xu *et al.*, 2019] operation on all nodes in this graph: $\mathbf{h}_G^{(l)} = \mathrm{READOUT}\left\{\mathbf{h}_v^{(l)} \mid v \in \mathcal{V}\right\}$. Finally, the representations of nodes, links, or graphs can be applied to different levels of graph tasks.

2.2 Curriculum Learning

Curriculum learning (CL), which mimics the human's learning process of learning data samples in a meaningful order, aims to enhance the machine learning models by using designed training curriculum, typically following an easy-to-hard pattern. As a general and flexible plug-in, the CL strategy has demonstrated its power in improving the model performance, generalization, robustness, and even convergence in a wide range of scenarios [Liu *et al.*, 2023a]. In general, the framework of curriculum learning consists of a *difficulty measurer* and a *training scheduler* [Wang *et al.*, 2021a]. The difficulty measurer is adopted to calculate the difficulty score for each data sample and the training scheduler aims to arrange data samples in a meaningful order as the curriculum for training, based on the judgment of the difficulty measurer.

The existing CL methods fall into two groups: (1) *predefined CL* that manually designs heuristic-based policies to decide the training order, and (2) *automatic CL* that relies on computable metrics (e.g., the training loss) to automatically design the curriculum for model training. Predefined CL can utilize expert knowledge but ignores the model's feedback, and automatic CL is domain-agnostic and more general as well as able to consider the model's feedback. That is to say, automatic CL is more data-driven or model-driven instead of human-driven, and more dynamically adaptive to the current training status, than predefined CL. In addition, the training schedulers can be divided into *discrete* and *continuous* schedulers. The discrete schedulers adjust training after every fixed number (> 1) of epochs or convergence on the current data samples, while the continuous schedulers adjust the training at every epoch.

2.3 Curriculum Graph Machine Learning

Let $\mathcal{C} = \langle Q_1, \dots, Q_t, \dots, Q_T \rangle$ denote a curriculum for training the graph model, which consists of a sequence of training data subsets over T training steps. Each data subset Q_t contains current training samples to be fed into the learning model at time step t. The order of all data subsets is determined by the difficulty measurer and the training scheduler. Therefore, the problem of curriculum graph machine learning (Graph CL) can be formulated in the following:

Problem 1. *(Curriculum Graph Machine Learning). Given the training set of data instances (i.e., nodes, links, or graphs) $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{|\mathcal{D}|}$, where X_i is the input instance and Y_i denotes the label, the goal is to optimize an optimal graph learning model guided by the curriculum $\mathcal{C}$, so as to achieve the best performance on testing graph data.*

3 Categorization

To tackle the non-trivial challenges introduced in Section 2, considerable efforts have been made in the literature, which integrate the strengths of graph machine learning and curriculum learning, and further propose tailored methods. Next, we will comprehensively review the existing methodologies by categorizing them into three groups based on the granularity of graph tasks, i.e., node-level, link-level, and graph-level tasks, followed by elaborations on representative approaches.

Method	Graph CL Type	Difficulty Measurer	Training Scheduler	Task	Need Label
Node-level Graph CL					
CLNode [2022]	Predefined	Label Distribution	Continuous	Node Classification	✓
GNN-CL [2022c]	Predefined	Sample Similarity	Continuous	Node Classification	✓
SMMCL [2019]	Predefined	Label Distribution	Discrete	Node Classification	✓
DiGCL [2021]	Predefined	Laplacian Perturbation	Continuous	Node Classification	✗
HSAN [2023b]	Predefined	Sample Similarity	Discrete	Node Classification	✗
MTGNN [2020b]	Predefined	Step Length	Discrete	Time Series Forecasting	✓
MentorGNN [2022a]	Automatic	Attention Weight	Discrete	Node Classification	✗
RCL [2023]	Automatic	Self-supervised Loss	Continuous	Node Classification	✓
DRL [2018]	Automatic	Cumulative Reward	Discrete	Node Classification	✓
GAUSS [2022]	Automatic	Sample Loss	Discrete	Node Classification	✓
CGCT [2021]	Automatic	Sample Similarity	Discrete	Image Classification	✓
Link-level Graph CL					
GCN-WSRS [2018]	Predefined	Sample Similarity	Continuous	Link Prediction	✓
TUNEUP [2022]	Predefined	Node Degree	Discrete	Link Prediction	✓
CHEST [2023]	Predefined	Pretraining Task	Discrete	Link Prediction	✗
GTNN [2022]	Automatic	Sample Loss	Discrete	Relation Extraction	✓
Graph-level Graph CL					
CurGraph [2021b]	Predefined	Label Distribution	Discrete	Graph Classification	✓
CuCo [2021]	Predefined	Sample Similarity	Continuous	Graph Classification	✗
HACL [2022]	Predefined	Sample Size	Discrete	Graph Classification	✓
Dual-GCN [2021]	Predefined	BLEU Metric	Discrete	Image Captioning	✓
CurrMG [2022]	Automatic	Domain Knowledge	Continuous	Graph classification	✓
HAC-TSP [2022b]	Automatic	Solution Cost	Continuous	Travelling Salesman Problem	✗

Table 1: A summary of curriculum graph machine learning methods. "Graph CL Type" denotes the curriculum learning type that each method belongs to, including predefined and automatic graph CL. "Difficulty Measurer" indicates the principle to design difficulty metric. "Training Scheduler" denotes the scheduler type that each method adopts. "Task" means the learning task of each method, including general tasks and specific applications. "Need Label" represents whether the method relies on labels during the training process.

Furthermore, for each category, we divide the methodologies into *predefined graph CL* and *automatic graph CL*, based on the type of curriculum learning strategy, i.e., whether adopting manually designed heuristic-based policies or automatically computable metrics to derive the curriculum for training. The categorization and characteristics of surveyed graph CL methods are summarized in Table 1.

4 Node-level Graph CL

Node is the fundamental unit for graph formation. Typically, the key to tackle node-level tasks is to learn node representations by training graph models (e.g., GNNs). Therefore, several studies on node-level graph CL are proposed to train graph models by starting with easy nodes and gradually include harder nodes during the training process, including both predefined and automatic node-level graph CL approaches.

4.1 Predefined Node-level Graph CL

Some methods define heuristic metrics to measure the difficulty of nodes in advance of the training process, e.g., by considering nodes' properties in terms of the topology, the feature, or the label.

CLNode (Curriculum Learning for Node Classification) [Wei *et al.*, 2022] is a curriculum learning framework for node-level representation learning of GNNs. It boosts the performance of backbone GNNs via incrementally introducing nodes into the training process, starting with easy nodes and progressing to harder nodes. The multi-perspective difficulty measurer is proposed to measure the difficulty of training nodes based on the label information. Specifically, the local perspective's difficulty measurer calculates local label distribution to recognize inter-class difficult nodes whose neighbors have diverse labels. And the global perspective's difficulty measurer recognizes mislabeled difficult nodes in terms of node feature. The continuous training scheduler is introduced to select appropriate training nodes in each epoch to mitigate the detrimental effect of difficult nodes. CLNode can be compatible with most existing GNNs and boost their performances without increasing the time complexity just by feeding training nodes in order from easy to hard.

GNN-CL (Graph Neural Network with Curriculum Learning) [Li *et al.*, 2022c] introduces curriculum learning into the imbalanced node classification task by controlling the training procedure from easy to hard, which consists of two modules. The first one is an adaptive graph oversampling module, which interpolates the most significant samples related to the original structure, so as to dynamically adjust the data distribution in the graph from imbalance to balance. The second one is a neighbor-based metric learning module. The distances between nodes and the connected neighbors are regularized based on the pseudo labels, which can dynamically adjust the position of the embeddings of minority class nodes in feature space. GNN-CL balances the label classification loss and neighbor-based triplet loss [Wang *et al.*, 2019] in the whole training process. The curriculum strategy consisting of two opposite learning curves is adopted. At the beginning of the training process, it pays more attention to optimizing feature propagation as well as reducing biased noises in the soft feature space. And it gradually focuses more on the average accuracy in each class, leading to strong accuracy on imbalanced node classification datasets.

SMMCL (Soft Multi-Modal Curriculum Learning) [Gong *et al.*, 2019] proposes a graph CL method for the label propagation on graphs [Iscen *et al.*, 2019]. The goal is to learn the labels for predictions of unlabeled samples on graphs. Specifically, the authors assume that different unlabeled samples have different difficulty levels for propagation, so it should follow an easy-to-hard sequence with updated curriculum for label propagation. They also claim that the real-world graph data is often in multiple modalities [Gao *et al.*, 2017], where each modality should be equipped with a "teacher" that not only evaluates the difficulties of samples from its own viewpoint, but cooperates with other teachers to generate the overall simplest curriculum samples for propagation. They take the curriculum of the teachers as a whole, so that the common preference (i.e., commonality) of teachers on selecting samples can be captured. Finally, an accurate curriculum sequence is established and the propagation quality can thus be improved, leading to more accurate label prediction results.

In addition to the supervised methods above that heavily rely on labels for training, there exist some self-supervised contrastive node-level graph CL methods.

DiGCL (Directed Graph Contrastive Learning) [Tong *et al.*, 2021] considers multi-task curriculum learning to progressively learn from multiple easy-to-hard contrastive views in directed graph contrastive learning. Specifically, to maximize the mutual information between the representations of different contrastive views [You *et al.*, 2020] and produce informative node representations, it introduces a generalized dynamic-view contrastive objective. The multi-task curriculum learning strategy is proposed to divide multiple contrastive views into sub-tasks with various difficulties and progressively learn from easy-to-hard sub-tasks. The different contrastive view pairs generated from Laplacian perturbation on the input graph are scored by a difficulty measurer that is a predefined function in terms of the Laplacian perturbation [Tong *et al.*, 2020]. And three common function families are considered as the training scheduler, including logarithmic, exponential, and linear functions. Note that this method is first to introduce curriculum learning in directed graph contrastive learning for node-level representation learning.

HSAN (Hard Sample Aware Network) [Liu *et al.*, 2023b] also considers curriculum learning scheme in contrastive clustering on graphs. Specifically, it first introduces a similarity measure criterion between training pairs in graph contrastive learning, which measures the difficulty considering both attribute and structure information and improves the representativeness of the selected hard negative samples. Besides, to overcome the drawback of classical graph contrastive learning that the hard node pairs are treated equally, it proposes a dynamic weight modulating function to adjust the weights of sample pairs during training, which can up-weight the hard node pairs and down-weight the negative ones. The focusing factor controls the down-weighting rate of easy sample pairs, as the training scheduler of this method. Thus, the discriminative capability of learned representations is enhanced, leading to better performances.

MTGNN (Multivariate Time Series GNN) [Wu *et al.*, 2020b] is a graph CL method designed specifically for multivariate time series data [Box *et al.*, 2015], which advocates a curriculum learning strategy to find a better local optimum of the GNN and splits multivariate time series into subgroups during training. Specifically, since directly optimizing the traditional objective enforces the model to focus too much on improving the accuracy of long-term predictions while ignoring short-term ones, this method proposes a curriculum learning strategy for the multi-step forecasting task. The training is scheduled by starting with solving the easiest problem, i.e., only predicting the next one-step, which helps the model to find a good starting point. As the training progresses, the prediction length of the model is gradually increased, so that the model can learn the hard task step by step. Overall, it is one effective trial combining GNN and curriculum learning in the application of multivariate time series forecasting.

4.2 Automatic Node-level Graph CL

We next describe automatic node-level graph CL methods, which consider the model's feedback during training to dynamically adapt to the optimization status.

MentorGNN [Zhou *et al.*, 2022a] derives a curriculum for pre-training GNNs to learn informative node representations. In order to tailor complex graph signals to boost the generalization performances, it develops a curriculum learning paradigm that automatically reweights graph signals for good generalization of the pre-trained GNNs in the target domain. Specifically, a teacher model that is a graph signal reweighting scheme gradually generates a domain-adaptive curriculum to guide the pre-training process of the student model that is a GNN architecture, so that the generalization performance in the node classification tasks can be enhanced. The curriculum is a sequence of graph signals that are extracted from the given graph. And the learned sample weighting scheme specifies a curriculum under which the GNNs are pre-trained gradually from the easy samples to the hard samples. The difficulty of training samples is measured by the teacher model and the training process is scheduled by the introduced learning threshold controlling the sample selection, so it belongs to automatic graph CL. The accuracy of this method on node classification in the graph transfer setting [Hu *et al.*, 2020b] is largely enhanced.

RCL (Relational Curriculum Learning) [Zhang *et al.*, 2023] claims that existing GNNs learn suboptimal node representations since they usually consider every edge of the input graph equally. Also, most graph CL methods simply consider nodes as independent samples for training, and introduce curriculum learning for these independent samples, which largely ignore the fundamental and unique dependency information behind the graph topology structure, and thus can not well deal with the correlation between nodes. To tackle this problem, it proposes a graph CL method, which leverages the various underlying difficulties of data dependencies, to train better GNNs that can improve the quality of learned node representations. Specifically, it considers the relation between nodes gradually into training based on the relation's difficulty from easy to hard, where the degree of difficulty is measured by a self-supervised learning paradigm instead of a predefined heuristic-based metric. Then it develops an optimization model to iteratively increment the training structure according to the model training status and a theoretical guarantee of the convergence on the optimization algorithm is provided. Finally, it presents an edge reweighting strategy to smooth the transition of the training structure between iterations, and reduce the influence of edges that connect nodes with relatively low confidence embeddings.

DRL (Learn curricula with Deep Reinforcement Learning) [Qu *et al.*, 2018] studies learning curriculum for node representations in heterogeneous star network [Sun *et al.*, 2009] that has a center node type linked with multiple attribute node types through different types of edges, namely learning a sequence of edges of different types for node representation learning. A curriculum is defined as a sequence of edge types used for training, so the problem is formulated as a Markov decision process [Puterman, 1990]. It learns the optimal curriculum by estimating the Q value of each state-action pair, namely the expected cumulative reward after taking the action from the state, by a planning module and a learning module. Finally, the meaningful curriculum can be learned with high accuracy and low time costs for enhancing the performance of node classification.

GAUSS (Graph ArchitectUre Search at Scale) [Guan *et al.*, 2022] is one large-scale node-level representation learning method by searching the GNN's architecture with curriculum learning. Since this method focuses on the large-scale graph whose efficiency issue becomes the main obstacle, it proposes a graph sampling-based single-path one-shot supernet [Pham *et al.*, 2018] to reduce the computation burden. To address the consistency collapse issues, it explicitly considers the joint architecture-graph sampling via a GNN architecture curriculum learning mechanism on the sampled sub-graphs and an architecture importance sampling algorithm [Tokdar and Kass, 2010]. Specifically, it first forms a curriculum learning group for the GNN's architecture, and then makes the best learner as the teacher to decide a smoother learning objective for the group. The importance sampling is also utilized to reduce the variance during architecture sampling to form better learning group. Experiments of node classification on the large-scale graph show the effectiveness of this graph CL method.

CGCT (Curriculum Graph Co-Teaching) [Roy *et al.*, 2021] presents a graph CL method for the multi-target domain adaptation based on feature aggregation and curriculum learning. The authors claim that learning robust and generalized representations in a unified space is just one prerequisite for tackling minimum risk across multiple target domains, where GNNs can play an important role in aggregating semantic information from neighbors across different domains on graphs represented as the source and target nodes. Then, the curriculum learning strategy is advocated in the proposed co-teaching framework to obtain pseudo-labels in an episodic fashion for mitigating information absence for the target nodes. Furthermore, an easy-to-hard curriculum learning strategy for domain selection is also proposed, where the feature alignment starts with the target domain that is closest to the source (easy one) and then gradually progresses towards the hard one, making the feature alignment process smoother. The experiments show that the performance on multi-target domain adaptation settings is improved by this curriculum graph co-teaching scheme.

5 Link-level Graph CL

Links explicitly interconnect nodes in a graph, representing the relations and dependencies between nodes. Compared with node-level graph CL, link-level graph CL measures the difficulty and schedules the training on links.

5.1 Predefined Link-level Graph CL

GCN-WSRS (GCN for Web-Scale Recommender Systems) [Ying *et al.*, 2018] presents a curriculum learning for negative sampling in link prediction tasks, whose effectiveness is validated in recommendation datasets. Specifically, the GNN model for learning link representations is fed easy-to-hard samples during the training process, resulting in performance gains. At the first training epoch, there are no hard negative links used for training, so the GNN model can quickly converge to find an area in the parameter space where the loss is small. Note that the hard samples are those related to the query samples, but not as related as the positive samples in link predictions. The training scheduler outputs the rank of items in a graph according to the Personalized PageRank scores [Gasteiger *et al.*, 2018] with respect to the query sample. As training proceeds, the harder negative links are added to the training set in subsequent epochs, which encourages the GNN model turns to learn how to distinguish highly related samples from only slightly related ones. In particular, $e-1$ hard negative samples are added to the training set at the epoch e.

TUNEUP [Hu *et al.*, 2022] is a two-stage curriculum learning strategy for better training GNNs, showing competitive performance gains against traditional GNNs in predicting new links in a graph given existing links on citation networks, protein-protein networks, and recommendation benchmarks. The link scores are produced by the inner product of the two connected nodes. Therefore, TUNEUP first trains a GNN to perform well on relatively easy head nodes (nodes with large degrees) and then proceeds to finetune the GNN to also perform well on hard tail nodes (nodes with small degrees). The node degrees are used to measure the difficulty of training samples. Specifically, in the first stage, TUNEUP randomly presents training samples to train the GNN model for obtaining a strong base GNN model. This base GNN model is encouraged to learn better representations for the head nodes, but performs poorly on the tail nodes. Thus, to handle this problem, in the second training stage, it further finetunes the base GNN model with increased supervisions on tail nodes, which follow the two steps: synthesizing more tail node inputs and adding target supervisions on the synthetic tail nodes. In addition to improving the performance of link predictions, it also performs well in learning node representations.

CHEST (Curriculum pre-training based HEterogeneous Subgraph Transformer) [Wang *et al.*, 2023] designs a curriculum graph pre-training strategy to gradually learn from both local and global contexts in the subgraph, which helps the GNN model to more effectively capture useful information for link prediction in recommendation. Compared with the supervised methods above, this method focuses on curriculum graph pre-training tailored to the link prediction. Specifically, the difficulty measurer is predefined on several pre-training tasks. Three easy pre-training curricula are related to node, edge and meta-path [Sun *et al.*, 2011], focusing on local context information within the subgraph. And a hard pre-training curriculum is a subgraph contrastive learning task, focusing on global context information at subgraph level for user-item interaction. The training scheduler is hand-crafted that schedules the pre-training tasks from the curricula in an "easy-to-hard" order, which is necessary to model complex data relations. The learned GNN model after curriculum learning can produce representations that are aggregated into obvious clusters, by gradually extracting useful information for user-item interaction, to improve the link prediction tasks.

5.2 Automatic Link-level Graph CL

GTNN (Graph Text Neural Network) [Vakil and Amiri, 2022] is one representative automatic link-level graph CL method. It trains GNN with trend-aware curriculum learning, which improves the performance on relation extraction that is one type of link-level task. Inspired by the SuperLoss (SL) [Castells *et al.*, 2020] which is a generic curriculum learning approach that dynamically learns a curriculum from model status, this method further designs Trend-SL curriculum learning approach, which belongs to self-paced curriculum learning [Wang *et al.*, 2021a]. Specifically, SuperLoss ignores the trend of instantaneous losses at sample-level that can not only improve the difficulty estimations of the model by making them local, sample dependent and potentially more precise, but also enable the model to distinguish samples with similar losses based on their known loss trajectories. In contrast, Trend-SL takes into account the loss information from the local time window before each iteration for capturing a form of momentum of loss on rising or falling trends and producing individual sample weights. Trend-SL adopts trend dynamics to shift the difficulty boundaries and adjust global difficulty using local sample-level loss trends. With the help of the Trend-SL, the performance of relation extraction can be improved on several benchmarks.

6 Graph-level Graph CL

Compared with node-level and link-level graph CL, graph-level graph CL focuses more on the global high-level properties of the whole graph to design the difficulty measurer and training scheduler for curriculum learning.

6.1 Predefined Graph-level Graph CL

CurGraph [Wang *et al.*, 2021b] proposes a curriculum learning method for graph classification tasks via easy-to-hard curriculum. It first obtains graph-level embeddings via unsupervised GNN scheme InfoGraph [Sun *et al.*, 2019] and further derives a neural density estimator to model embedding distributions. Then, to tackle the challenges of evaluating the difficulty of graphs induced by high irregular nature of graph data, it analyzes the graph difficulty in the high-level semantic feature space. The difficulty scores of graphs are calculated by a predefined difficulty measurer based on the intra-class and inter-class distributions of their embeddings. For the training scheduler, a smooth-step method is proposed to provide a soft transition from easy to hard graphs for GNNs. At each training step, the trained GNNs focus on the samples that are near the border of capability and neither too easy nor too hard, to expand the border gradually. Finally, the performances of graph classification are enhanced without extra inference cost by feeding the graphs in an easy-to-hard order for training.

CuCo [Chu *et al.*, 2021] incorporates curriculum learning into self-supervised graph-level representation learning. Compared with the supervised graph CL methods, this method focuses more on designing curriculum for selecting and training negative samples effectively, which are important steps in learning self-supervised graph representations. Similarly, it follows the learning process of humans by starting with easy negative samples when learning a new model and then learning difficult negative samples gradually. The difficulty measurer evaluates the difficulty of negative samples in the training dataset, which is calculated based on the embedding's similarity between the negative and positive samples. In addition, the training scheduler is proposed to decide how the negative samples are introduced to the training procedure by utilizing common function families: logarithmic, linear, quadratic, and root [Wang *et al.*, 2021a; Soviany *et al.*, 2022]. This method is the first to study the impact of negative samples in graph-level contrastive representation learning by introducing curriculum learning.

HACL (Hyper-graph based Attention Curriculum Learning) [Ahmed *et al.*, 2022] is a graph attention curriculum learning approach to learn hypergraph representations. Note that the hypergraphs are converted from the text written by the patients, which can be used to identify depressive symptoms of the patients. Based on the semantic vectors from an emotion-driven context extraction technique and the structural hypergraph, this method separates the important boundary elements from the unlabeled sample and then incorporates them into the curriculum learning mechanism for the training process. Finally, the performances of classifying the nine particular symptoms are highly boosted due to the tailored designs on graph CL.

Dual-GCN [Dong *et al.*, 2021] is one GNN model encoding structure information from both local and global levels with curriculum learning as the training strategy, whose capability is improved on the image captioning task. Specifically, the local object-level GCN first converts one image into one graph, where the region to region relation in the image is modeled by the graph topology. Besides, it also introduces auxiliary information to take into account the image's similarity relation by the global image-level GCN. To train this Dual-GCN, curriculum learning is adopted as the training strategy, where a cross-review scheme is introduced to distinguish the difficulty of the training samples. The difficulty is measured by a predefined metric, which is used to sort the training samples and divide them into several subsets. After that, all the training samples are scheduled in the order from easy subsets to difficult subsets.

6.2 Automatic Graph-level Graph CL

CurrMG [Gu *et al.*, 2022] designs a curriculum learning strategy to learn representations for molecular graphs. It arranges the easy-to-hard training curriculum in predefined, (automatic) transfer teacher, and hybrid types [Hacohen and Weinshall, 2019]. To design molecular graph CL with high robustness and applicability, six optional difficulty measurers, inspired by chemical domain knowledge and prior task information, are proposed, which can be divided into structure-specific, task-specific and hybrid types. For the training scheduler, it is infeasible to split the training data based on molecular difficulties explicitly, as the difference of difficulty distributions is calculated by different difficulty measurers. Therefore, a monotonically increasing smooth curve called competence function is introduced to control the data sampling space. The competence value is treated as a threshold for decreasing the data sampling space continuously, which is used for sampling current batch data during the training process. Finally, the training scheduler encourages the GNN model to pay attention to the easy graphs in the early training stage, and gradually broaden the learning scope by contacting those difficult graphs.

HAC-TSP [Zhang *et al.*, 2022b] proposes to design hardness-adaptive curriculum for solving travelling salesman problem (TSP) [Applegate *et al.*, 2011] from the perspective of graphs. Compared with classical graph prediction tasks, defining a quantitative hardness measurement is challenging since obtaining the ground-truth solution of a TSP instance is NP-hard [Hochba, 1997]. Therefore, the difficulty measurer calculates the hardness as greedy self-improving potentials by comparing the current cost with a surrogate model, avoiding the unbearable computational costs of calculating the ground-truth optimal solution for TSP. Then, a hardness-adaptive generator is designed to efficiently and continuously generate instances with different levels of hardness tailored for model training. Finally, it proposes a curriculum learner to fully utilize the hardness-adaptive TSP instances. By learning instance weights, this method can train the TSP solvers more efficiently through curriculum learning. This method moves an important step towards graph CL for solving NP-hard problems.

7 Future Directions

Curriculum graph machine learning (Graph CL) is an emerging research topic. Although significant progresses have been made for Graph CL, there still remain plenty of research directions worthy of future explorations.

- **Theoretical guarantees**: Although various graph CL methods have been proposed and demonstrated effectiveness empirically, it remains to be further explored to derive more fundamental theoretical analysis on graph CL. A promising direction is to develop such theoretical analysis inspired by general curriculum learning from the perspective of optimization problem [Weinshall *et al.*, 2018] or data distribution [Gong *et al.*, 2016] for better understanding the mechanism and effectiveness of the graph CL methods.
- **More principled Graph CL methods**: Although the existing works studied how curriculum learning methods are extended to graphs, it is still worth investigating to develop more principled models for graph curriculum learning by considering more detailed graph assumptions (e.g., homophily, heterophily), more complex graph types (e.g., attributed graphs, heterogeneous graphs, signed graphs, multiplex graphs), or more specific graph properties (e.g., data dependencies), into the model design. These methods are expected to further boost the model capacity.
- **Generalization and transferability**: Currently, most existing graph CL methods are overly dependent on the graph labels, so that learned graph models tend to inherit a strong inductive bias for new testing tasks. However, for real-world graphs, there will inevitably be scenarios with distribution shifts between testing and training graph data [Li *et al.*, 2022b; Li *et al.*, 2022a], which can induce significant performance drop for most existing approaches lacking the ability of generalization and transferability. One interesting direction is to learn label-irrelevant, generalizable and transferable representations via self-supervised learning via unsupervised pretext tasks, alleviating excessively relying on labeled data [Chu *et al.*, 2021]. The other feasible direction is to explicitly consider distribution shifts in the design of graph CL methods for learning better graph representations [Zhu *et al.*, 2021].
- **Comprehensive evaluation protocols**: While the graph CL methods have made great progress in performance boost, few works have made efforts on evaluating them with general graph benchmarks. The adopted datasets and evaluation metrics of existing works mainly follow classical settings. It is essential to develop a unified benchmark with unified metrics to evaluate and compare different methods, which should carefully incorporate datasets with different hardness (e.g., different levels of sparsity, heterogeneity, noises), and different evaluation metrics (e.g., relative performance boost, convergence speedup, additional computational costs). Besides, publicly available graph CL libraries are also important to facilitate and advance the research, which can be developed upon the existing library [Zhou *et al.*, 2022b].
- **Broader applications**: Although graph CL methods have been applied on several tasks as discussed in this work, it is worth further exploring the potential capacity in more diverse real-world applications, including recommendation [Ying *et al.*, 2018], healthcare [Ahmed *et al.*, 2022], biochemistry [Gu *et al.*, 2022], etc., for more effective and satisfying predictions. One of the major challenges is how to incorporate proper domain knowledge as additional priors to guide the model design.

Acknowledgments

This work was supported by the National Key Research and Development Program of China No. 2020AAA0106300, National Natural Science Foundation of China (No. 62250008, 62222209, 62102222), BNRist Grant No. BNR2023RC01003 and Beijing Key Lab of Networked Multimedia.

Contribution Statement

Haoyang Li and Xin Wang contribute equally. Xin Wang and Wenwu Zhu are the corresponding authors.

References

[Ahmed *et al.*, 2022] Usman Ahmed, Jerry Chun-Wei Lin, and Gautam Srivastava. Hyper-graph-based attention curriculum learning using a lexical algorithm for mental health. *Pattern Recognition Letters*, 2022.

[Applegate *et al.*, 2011] David L Applegate, Robert E Bixby, Vašek Chvátal, and William J Cook. The traveling salesman problem. In *The Traveling Salesman Problem*. 2011.

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

[Box *et al.*, 2015] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. 2015.

[Castells *et al.*, 2020] Thibault Castells, Philippe Weinzaepfel, and Jerome Revaud. Superloss: A generic loss for robust curriculum learning. *NeurIPS*, 2020.

[Chu *et al.*, 2021] Guanyi Chu, Xiao Wang, Chuan Shi, and Xunqiang Jiang. Cuco: Graph representation with curriculum contrastive learning. In *IJCAI*, 2021.

[Cui *et al.*, 2018] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE TKDE*, 2018.

[de Vico Fallani *et al.*, 2014] Fabrizio de Vico Fallani, Jonas Richiardi, Mario Chavez, and Sophie Achard. Graph analysis of functional brain networks: practical issues in translational neuroscience. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2014.

[Dong *et al.*, 2021] Xinzhi Dong, Chengjiang Long, Wenju Xu, and Chunxia Xiao. Dual graph convolutional networks with transformer and curriculum learning for image captioning. In *ACM MM*, 2021.

[Elman, 1993] Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 1993.

[Gao *et al.*, 2017] Yue Gao, Hanwang Zhang, Xibin Zhao, and Shuicheng Yan. Event classification in microblogs via social tracking. *ACM TIST*, 2017.

[Gasteiger *et al.*, 2018] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv:1810.05997*, 2018.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

[Gong *et al.*, 2016] Tieliang Gong, Qian Zhao, Deyu Meng, and Zongben Xu. Why curriculum learning & self-paced learning work in big/noisy data: A theoretical perspective. *Big Data and Information Analytics*, 2016.

[Gong *et al.*, 2019] Chen Gong, Jian Yang, and Dacheng Tao. Multi-modal curriculum learning over graphs. *ACM TIST*, 2019.

[Gu *et al.*, 2022] Yaowen Gu, Si Zheng, Zidu Xu, Qijin Yin, Liang Li, and Jiao Li. An efficient curriculum learning-based strategy for molecular graph learning. *Briefings in Bioinformatics*, 2022.

[Guan *et al.*, 2022] Chaoyu Guan, Xin Wang, Hong Chen, Ziwei Zhang, and Wenwu Zhu. Large-scale graph neural architecture search. In *ICML*, 2022.

[Guo *et al.*, 2018] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *ECCV*, 2018.

[Hacohen and Weinshall, 2019] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *ICML*, 2019.

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

[Hochba, 1997] Dorit S Hochba. Approximation algorithms for np-hard problems. *ACM Sigact News*, 1997.

[Horry *et al.*, 2020] Michael J Horry, Subrata Chakraborty, Manoranjan Paul, Anwaar Ulhaq, Biswajeet Pradhan, Manas Saha, and Nagesh Shukla. Covid-19 detection through transfer learning using multimodal imaging data. *Ieee Access*, 2020.

[Hu *et al.*, 2020a] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *NeurIPS*, 2020.

[Hu *et al.*, 2020b] 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.*, 2022] Weihua Hu, Kaidi Cao, Kexin Huang, Edward W Huang, Karthik Subbian, and Jure Leskovec. Tuneup: A training strategy for improving generalization of graph neural networks. *arXiv:2210.14843*, 2022.

[Iscen *et al.*, 2019] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, 2019.

[Jiang *et al.*, 2014] Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. In *ACM MM*, 2014.

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

[Li *et al.*, 2022a] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE TKDE*, 2022.

[Li *et al.*, 2022b] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *arXiv:2202.07987*, 2022.

[Li *et al.*, 2022c] Xiaohe Li, Lijie Wen, Yawen Deng, Fuli Feng, Xuming Hu, Lei Wang, and Zide Fan. Graph neural network with curriculum learning for imbalanced node classification. *arXiv:2202.02529*, 2022.

[Liu *et al.*, 2023a] Hanwen Liu, Peilin Zhao, Tingyang Xu, Yatao Bian, Junzhou Huang, Yuesheng Zhu, and Yadong Mu. Curriculum graph poisoning. In *The WebConf*, 2023.

[Liu *et al.*, 2023b] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. Hard sample aware network for contrastive deep graph clustering. *AAAI*, 2023.

[Pham *et al.*, 2018] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *ICML*, 2018.

[Platanios *et al.*, 2019] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. Competence-based curriculum learning for neural machine translation. *arXiv:1903.09848*, 2019.

[Puterman, 1990] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 1990.

[Qu *et al.*, 2018] Meng Qu, Jian Tang, and Jiawei Han. Curriculum learning for heterogeneous star network embedding via deep reinforcement learning. In *WSDM*, 2018.

[Rohde and Plaut, 1999] Douglas LT Rohde and David C Plaut. Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, 1999.

[Roy *et al.*, 2021] Subhankar Roy, Evgeny Krivosheev, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Curriculum graph co-teaching for multi-target domain adaptation. In *CVPR*, 2021.

[Sanchez-Gonzalez *et al.*, 2020] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *ICML*, 2020.

[Soviany *et al.*, 2022] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *IJCV*, 2022.

[Sun *et al.*, 2009] Yizhou Sun, Yintao Yu, and Jiawei Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *SIGKDD*, 2009.

[Sun *et al.*, 2011] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB Endowment*, 2011.

[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.

[Tay *et al.*, 2019] Yi Tay, Shuohang Wang, Luu Anh Tuan, Jie Fu, Minh C Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. *arXiv:1905.10847*, 2019.

[Tokdar and Kass, 2010] Surya T Tokdar and Robert E Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2010.

[Tong *et al.*, 2020] Zekun Tong, Yuxuan Liang, Changsheng Sun, Xinke Li, David Rosenblum, and Andrew Lim. Digraph inception convolutional networks. *NeurIPS*, 2020.

[Tong *et al.*, 2021] Zekun Tong, Yuxuan Liang, Henghui Ding, Yongxing Dai, Xinke Li, and Changhu Wang. Directed graph contrastive learning. *NeurIPS*, 2021.

[Vakil and Amiri, 2022] Nidhi Vakil and Hadi Amiri. Generic and trend-aware curriculum learning for relation extraction in graph neural networks. *arXiv:2205.08625*, 2022.

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

[Wang *et al.*, 2017] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE TKDE*, 2017.

[Wang *et al.*, 2019] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *ICCV*, 2019.

[Wang *et al.*, 2021a] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE TPAMI*, 2021.

[Wang *et al.*, 2021b] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Curgraph: Curriculum learning for graph classification. In *The WebConf*, 2021.

[Wang *et al.*, 2023] Hui Wang, Kun Zhou, Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. Curriculum pre-training heterogeneous subgraph transformer for top-n recommendation. *TOIS*, 2023.

[Wei *et al.*, 2022] Xiaowen Wei, Weiwei Liu, Yibing Zhan, Du Bo, and Wenbin Hu. Clnode: Curriculum learning for node classification. *arXiv:2206.07258*, 2022.

[Weinshall *et al.*, 2018] Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *ICML*, 2018.

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

[Wu *et al.*, 2020b] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *SIGKDD*, 2020.

[Wu *et al.*, 2022] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 2022.

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

[Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*, 2018.

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

[Zhang and Tong, 2016] Si Zhang and Hanghang Tong. Final: Fast attributed network alignment. In *SIGKDD*, 2016.

[Zhang *et al.*, 2020] Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep learning on graphs: A survey. *IEEE TKDE*, 2020.

[Zhang *et al.*, 2022a] Chuxu Zhang, Kaize Ding, Jundong Li, Xiangliang Zhang, Yanfang Ye, Nitesh V Chawla, and Huan Liu. Few-shot learning on graphs: A survey. *IJCAI*, 2022.

[Zhang *et al.*, 2022b] Zeyang Zhang, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning to solve travelling salesman problem with hardness-adaptive curriculum. In *AAAI*, 2022.

[Zhang *et al.*, 2023] Zheng Zhang, Junxiang Wang, and Liang Zhao. Relational curriculum learning for graph neural networks, 2023.

[Zhou *et al.*, 2022a] Dawei Zhou, Lecheng Zheng, Dongqi Fu, Jiawei Han, and Jingrui He. Mentorgnn: deriving curriculum for pre-training gnns. In *CIKM*, 2022.

[Zhou *et al.*, 2022b] Yuwei Zhou, Hong Chen, Zirui Pan, Chuanhao Yan, Fanqi Lin, Xin Wang, and Wenwu Zhu. Curml: A curriculum machine learning library. In *ACM MM*, 2022.

[Zhu *et al.*, 2021] Qi Zhu, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. Shift-robust gnns: Overcoming the limitations of localized graph training data. *NeurIPS*, 2021.