

多様な人工グラフを用いたGNNによるノード分類の実証研究

前川 政司[†] 野田 昂希^{††} 佐々木 勇和[†] 鬼塚 真[†]

[†] 大阪大学大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

^{††} 株式会社 TDAI Lab 〒103-0026 東京都中央区日本橋兜町 5-1

E-mail: [†]{maekawa.seiji,sasaki,onizuka}@ist.osaka-u.ac.jp, ^{††}koki.noda@tdailab.com

あらまし グラフ深層学習 (GNN) はノード分類タスクにおいて、大きな成功を収めている。GNN の開発および評価に広く関心が寄せられているにもかかわらず、それらは限られたベンチマークデータセットによってのみ評価されている。その結果、既存の GNN の評価は、多様な特性を持つグラフによる詳細な分析を欠いている。この問題に取り組むために、本論文では、グラフの多様な特性を制御可能な人工グラフ生成器を用いることで、クラスラベルを持つ実グラフが示す四つの主要な特性に関して GNN の利点と欠点を明らかにする: 1) クラスサイズ分布 (balanced vs. imbalanced), 2) クラス間のエッジ接続割合 (homophily vs. heterophily), 3) 属性値 (biased vs. random), 4) グラフサイズ (small vs. large). 加えて、GNN の今後の研究を推進するために、我々は多様なグラフで多様な GNN を評価可能であるコードベースを公開した。

キーワード グラフ深層学習, 人工グラフ, ノード分類, 属性付きグラフ, heterophily

1 はじめに

半教師ありノード分類タスクは、注目を浴びているトピックの一つである。その目的は、部分的にラベル付されたネットワークが与えられたときに、グラフ構造とノード属性を用いてノードの未知ラベルを予測することである。直近数年では、グラフ深層学習 (GNN) [1–12] は、化学 [13,14] や物理 [15], 社会科学 [1], 神経科学 [16] などの分野で広く関心を集めている。分類精度の更なる向上のために、ほとんどの研究は表現力の高い強力な GNN を開発することに注力している。

GNN の実践的なユースケースに向けて、研究者や開発者は精度や効率性、パラメータへの感受性などの多様な観点から、GNN の強みと弱みを深く理解する必要がある。この目的のために異なる特性を持つ多様なグラフを用いた、広範な実験を行うことは重要である。

GNN の既存評価の課題. 既存の研究 [14,17] では、GNN のパフォーマンスのベンチマークに取り組んできた。しかし、それらの研究では Cora や Citeseer, Pubmed [18] などの広く知られているが限られたベンチマークデータセットのみでほとんどのモデルが評価されているため、GNN の包括的な評価は困難である。最近の研究 [19,20] がデータセット不足を解消するために実データセットを提供しているが、多様なモデルアーキテクチャの利点と欠点を理解するためには、これらのデータセットは未だに不十分である。具体的には、GNN を評価するために以下に示す属性付きグラフ¹の四つの特性に関して、多様なグラフが必要である: 1) クラスサイズ分布, 2) クラス間のエッジ接続割合, 3) 属性値, 4) グラフサイズ。

異なる特性を持つグラフで GNN を評価するために、既存研

究 [17] は広く利用されているグラフ生成器である SBM [21] を用いる。しかし、SBM は現実的なグラフを生成することできず、またノード属性の生成をサポートしていないことから、生成される人工グラフの質と多様性は限定的である。この問題点のため、実グラフと人工グラフを用いる既存の GNN のベンチマーク研究は、詳細な分析を欠いている。例えば、一つや少数のターゲット特性のみを変化させ、他の特性については固定したグラフを用いた評価などである。私たちが知る限り、GNN の開発や評価には多大な関心が寄せられているにも関わらず、クラスに関して制御された特性を持つ多様な人工グラフによる GNN の包括的な評価を行った研究はない。

貢献点. GNN の利点と欠点を理解するために、私たちはグラフの一つもしくは少数の特性を人工的に変化させることによって、多様なグラフにおいて広範な実験を行い、GNN のパフォーマンスを実証的に調査する。この詳細な分析は、限られた種類しかない実ベンチマークデータセットでは実施することができない。GNN の詳細な分析のために、私たちは広く利用されているグラフ生成器である SBM よりも柔軟にノードラベルを持つグラフの特性を制御できる最先端のグラフ生成器 GenCAT [22] を採用する。上記の四つの主要な特性、1) クラスサイズ分布, 2) クラス間のエッジ接続割合, 3) 属性値, 4) グラフサイズに関して、私たちは下記の質問に答えることを目的とする: **Q1.** クラスサイズ分布はどれほど大きく GNN のパフォーマンスに影響するか? **Q2.** 多様なクラス間のエッジ接続割合を持つグラフにおいて、どれだけ効果的に GNN は動作するか? **Q3.** 属性値はどれほど大きく GNN のパフォーマンスに貢献するか? **Q4.** 多様なサイズを持つグラフにおいて、GNN はどれだけ効果的に動作するか?

私たちの実証研究のキーとなる知見を以下にまとめる。

- 最先端の手法を含む GNN は、マルチクラス分類のパ

¹: 本研究では、ノード分類について考えるため、各ノードは所属先のクラスを示すラベルを持つ。

パフォーマンスはクラス不均等問題によって、一般的に悪化する。興味深いことに、クラスサイズが不均等な設定では最近の複雑な GNN アルゴリズムではサイズの大きいクラスに過適合するため、最も単純なアルゴリズムである SGC [23] が複雑なアルゴリズムを上回るパフォーマンスを示した。

- 各クラス内でほとんどエッジを持たないグラフ (heterophilic グラフと呼ぶ) を考慮する GNN は、heterophily 設定においてグラフを無視する分類器である MLP からわずかな改善しか示さなかった。これはそのような GNN が heterophily 設定では、構造情報をほぼ無視していることを示唆している。

私たちの評価フレームワークの利用によって、既存のベースライン GNN との比較や新しいアルゴリズムの開発にかかる手間が大きく削減されることが期待される。我々のコードベース²は、MIT ライセンスのもと公開されている。

2 事前準備

ノードラベルを持つ属性付きグラフは $G = (\mathbf{A}, \mathbf{X}, \mathbf{C})$ と表され、 $\mathbf{A} \in \{0, 1\}^{n \times n}$ は隣接行列、 $\mathbf{X} \in \mathbb{R}^{n \times d}$ はノードに属性を割り当てる属性行列、 $\mathbf{C} \in \{0, 1\}^{n \times y}$ は各ノードのクラス情報を保持するクラス行列であり、 n, d, y はそれぞれノード数、属性数、クラス数を表す。同じラベルを持つノードの集合をクラスと呼び、ラベル l を持つノードの集合を Ω_l と定義する。

問題定義 (ノード分類). ノード分類では、ノードを訓練/検証/テスト集合に分割する。そして、隣接行列 \mathbf{A} 、属性行列 \mathbf{X} 、訓練と検証集合に含まれるノードのクラス情報を持つ部分的なクラス行列 \mathbf{C}' が与えられたときに、テスト集合に含まれるノードのラベルを予測する。

3 ノード分類のためのグラフ深層学習

グラフ畳込み処理は、グラフベースの深層学習において最も中核的なトピックである。畳込みニューラルネットワーク (CNN) から着想を受け、複数の畳込み層を用いた GNN は局所的と大域的な構造パターンを学習する。グラフ畳込みネットワークに基づいた多くの手法が提案されている。本章では、いくつかのキーとなる GNN の手法を簡単に説明する。

3.1 典型的な畳込み GNN

複数層のグラフ畳込みネットワーク (GCN) は [1] に提案され、標準的なグラフ畳込みネットワークモデルである。グラフ畳込みは、グラフ信号処理において数学的に裏付けされたグラフスペクトラル分析と深く関連する。グラフ信号処理は典型的に固有値分解を用い、それは $O(n^3)$ の計算量を要する。計算量を削減するため、ChebNet [24] は k 乗の Chebyshev 多項式を用いて、固有値分解を近似する。このとき、 k は局所的な特徴伝播のホップ数を示す。混合モデルネットワーク (MoNet) [2] は、各ノードとその近傍との間の相対位置を用いることによって、隣接ノードへの重みを割り当てるアプローチを提案した。GCN [1] などの

いくつかの既存のアプローチは、Monet の特殊ケースとして一般化できる。グラフアテンションネットワーク (GAT) [4] は接続している各ノードペアの間の相対重みを学習するアテンションメカニズムを採用する。また、GAT はマルチヘッドアテンションメカニズムを用いることによって、その表現力を増強している。Jumping Knowledge network (JKnet) [25] は異なる畳込み層の間にスキップ接続を導入する。SGC [23] は畳込み層から非線形性を取り除くことによって、GCN を簡略化する。SGC は GCN と同程度のパフォーマンスを示す一方で、計算量の観点でより効率的かつ少ないパラメータを学習する。Graph isomorphism networks (GIN) [6] は証明可能な高い表現力を持つ単純なアーキテクチャを持ち、それは Weisfeiler-Lehman グラフ同型テストと同等の識別性能がある。

サンプリングベース GNN. サンプリングベース GNN [3,10,26] は入力グラフより抽出したサブグラフからノード表現を計算する。GraphSAGE [3] は各ノードに対して近傍から固定数のノードを一様にサンプリングする。GraphSAINT [26] はサブグラフをサンプリングし、そのサブグラフの中でノード埋込を伝播する。GraphSAINT はたくさんのサブグラフの情報を一緒に組合せることによって、全体グラフの良い表現を学習する。Shadow-GNN [10] は GNN の層数とサブグラフ抽出の範囲を分離する。これによって、Shadow-GNN はサブグラフのサイズを小さくしながら、多くのホップ数でノード埋込を伝播することを可能にする。そのため、Shadow-GNN は他の既存手法に比べて低いハードウェアコストで、最先端の分類精度を達成した。

3.2 Heterophilic グラフを考慮した GNN

多くの GNN モデルは暗黙的にグラフ内の homophily を仮定している。すなわち、同じクラスラベルを持つノードは互いに接続する傾向がある。しかし、実世界では、“反対のもの同士が惹かれ合う”異なる設定も存在する。それは heterophily 設定と呼ばれ、接続するノードは異なるクラスに所属する割合が高いケースや異なる属性を持つケースがある。例えば、オンライン購買ネットワークでは、詐欺師たちは、他の詐欺師たちと接続する場合に比べて、通常のユーザと接続する傾向がある。

ほとんどの既存の GNN はグラフの heterophily を捉えることに失敗しているが、最近の研究 [7, 11, 12] は homophily と heterophily を一般化することに取り組んできた。Heterophily を捉えるために、H2GCN [7] は三つのデザインを組合せる：1) 自己と近傍の埋込の分離、2) 高次の近傍ノード、3) 中間表現の組合せ。これらのデザインによって、H2GCN は複雑な heterophily 設定で GNN の表現力を向上させ、heterophily を無視した既存手法よりも良い結果を獲得した。FSGNN [11] は特徴伝播の深さとニューラルネットワークの層数を分離することによってモデルパラメータの数を減らし、高次の近傍情報を柔軟に利用することを可能にした。また、FSGNN は heterophilic グラフにおいて動作するために、ノード分類にとって重要なホップに大きな重みを与える特徴選択パラメータを採用する。GPRGNN [12] は特徴伝播の各ホップに関連付けられた generalized PageRank パラメータを自動で学習する。FSGNN の特徴選択パラメータ

²: <https://github.com/seijimaekawa/empirical-study-of-GNNs>

と同様、このパラメータは異なるホップの貢献度を学習する。LINKX [27] は Multi-Layer Perceptrons (MLP) に基づいた単純かつスケラブルな手法であり、heterophilic グラフ（その論文中では non-homophilious グラフと呼ばれる）での学習において最先端のパフォーマンスを獲得している。しかし、LINKX は homophily を捉えることに適したグラフ畳込みを採用していないため、既存の GNN と比べて homophilic グラフで低い精度を示す。

4 実ベンチマークデータセットとその課題

新しい GNN アルゴリズムと既存のアルゴリズムを正当に評価するために、ベンチマークデータセット重要な役割を担っている。本章では、まずベンチマークデータセットの要約を提供し、その後 GNN の詳細な分析に向けたベンチマークデータセットの課題について述べる。

4.1 ベンチマークデータセット

Homophilic graphs. Cora, Pubmed, および Citeseer [18] は広く用いられている三つの引用関係グラフである。また、特定の領域での Amazon 購買ネットワーク (Computers と Photo) は、いくつかの研究 [12, 28] で用いられている。Reddit [3] はソーシャルネットワークのデータセットであり、そこではノードはポストを表し、同じユーザがコメントを残したポストは接続されている。大規模な homophilic グラフの集合が Open Graph Benchmark プロジェクト [19] によって最近公開された。それには二つの引用関係グラフ (ogbn-arxiv と ogbn-papers100M) と Amazon の購買ネットワーク (ogbn-products) が含まれている。

Heterophilic graphs. Texas, Wisconsin および Cornell は対応する大学³の Web ページ間のリンクを表現するグラフである。Actor [29] は、各ノードが俳優を表し、同じ Wikipedia ページに共起したことをエッジとする共起ネットワークである。Chameleon と Squirrel [29] は特定のトピックにおける Wikipedia ネットワークである。多様な領域からなる大規模な heterophilic グラフの集合⁴が最近の研究 [20, 27] によって公開されている。それは五つのオンラインソーシャルネットワーク (Penn94, Pokec, genius, Deezer-Europe, および Twitch-Gamers), 二つの引用関係グラフ (arXiv-year と snap-patents), Wikipedia ウェブページネットワーク (Wiki), ホテルとレストランのレビューデータセット (YelpChi) が含まれている。

4.2 GNN の現在の評価の課題

ノード分類の結果は、ノードラベルを持つグラフの主要な四つの特徴に依存する： 1) クラスサイズ分布 (balanced vs. imbalanced), 2) クラス間のエッジ接続割合 (homophily vs.

heterophily), 3) 属性値 (biased vs. random), 4) グラフサイズ (small vs. large). いくつかの実グラフが公開されているが、それらを使った評価は詳細な分析を行うことに関して課題がある。すなわち、クラスや構造、属性などの入力グラフが持つ個別の特徴のうち、どれがノード分類の結果に大きな影響を与えたかを調べることは困難である。それは実グラフは複数の特徴について違いに異なっており、一つの特徴に注目して調査を行うことができないからである。そのような GNN アルゴリズムに関する深い分析や理解は、多大な注目を集める新しい GNN の開発の一助になる。

5 人工グラフ生成器

我々はノード分類のための GNN の詳細な分析に取り組むことを目的とするため、我々が採用するグラフ生成器は以下の二つの要件を満たす必要がある：1) 属性とラベルがノードに関連付けられているグラフを生成すること、2) 生成グラフの特徴を柔軟に制御すること。

上記の二つの要件を満たすただ一つの手法である GenCAT グラフ生成器 [22] を採用する。現在の最先端の手法 [21, 30] は GenCAT によって満たされている要件のいずれかをサポートできない。最近の研究 FastSGG [30] は数兆規模のエッジを効率的に生成するが、クラス間の接続を柔軟に制御することができない。すなわち、4.2 章で議論した二つ目の特徴を掴むことができない。他の例は SBM [21] であり、それは [22] で示されるように実グラフに似たグラフを生成できない。また、属性の生成をサポートしないため、三つ目の特徴を掴むことができない。cSBM [31] は [12] において GNN を評価するために人工グラフ生成器として利用されている。しかし、cSBM はグラフ内に二つのクラスがある場合という単純な設定のみに対応している。実グラフは多様なクラス数を持ちうるため、cSBM は包括的な評価には適していない。

GenCAT は最先端の属性付きグラフ生成器であり、4.2 章で述べたグラフの四つの主要な特徴を柔軟に制御することを可能とする。GenCAT はクラスと属性、構造の間の関係を掴むため、生成されたグラフの属性と構造はクラス構造を共有する。また GenCAT はそれ自体に変更を加えることなく、既存手法 SBM を模倣することができる。より詳細かつ正確な手続きについては [22] を参照のこと。

5.1 GenCAT グラフ生成器の主要な入力

属性付きグラフが持つ多様な特徴を捉えるために、GenCAT はクラス特徴とグラフ特徴と呼ばれるいくつかのパラメータを入力とする。ユーザの労力を軽減するために、GenCAT はノードラベルを持つ入力グラフからパラメータを抽出する機能を提供する。グラフ生成に関する変数一覧を表 1 に示す。

クラス特徴. 最初にクラスサイズ分布 $\rho \in \mathbb{R}_+^k$ を導入する。それは各クラス l に関して $\rho_l = |\Omega_l|/n$ を満たす。次に、homophily/heterophily を表現するために、クラス間のエッジ接続割合を表すクラス接続平均 $M \in \mathbb{R}^{k \times k}$ を導入する。クラ

3 : <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwwkb/>

4 : これらのデータセットは論文内 [20, 27] では、non-homophilious データセットと呼ばれている

表 1: 変数一覧

変数	説明
$A \in \{0, 1\}^{n \times n}$	隣接行列
$X \in \mathbb{R}^{n \times d}$	属性行列
$C \in \{0, 1\}^{n \times y}$	クラス行列
$\Omega_l \in \{1, \dots, n\}^*$	ラベル l を持つノードの集合
$n, m, d, y \in \mathbb{N}$	ノード数, エッジ数, 属性数, クラス数
$\rho \in \mathbb{R}_+^k$	クラスサイズ分布
$M \in \mathbb{R}^{k \times k}$	クラス接続平均
$H \in \mathbb{R}^{d \times k}$	属性クラス相関

ス l_1 とクラス l_2 の間のクラス接続平均を以下のように定式化する。

$$M_{l_1 l_2} = \frac{1}{|\Omega_{l_1}|} \sum_{i \in \Omega_{l_1}} \left(\sum_{j \in \Omega_{l_2}} A_{ij} / \sum_{j=1}^n A_{ij} \right). \quad (1)$$

最後に、属性とクラスの間の特異性を表す属性クラス相関 $H \in \mathbb{R}^{d \times k}$ を導入する。クラス l に属するノードの属性 δ の平均値として、属性 δ とクラス l の間の属性クラス相関を以下のように定式化する。

$$H_{\delta l} = \frac{1}{|\Omega_l|} \sum_{i \in \Omega_l} (X_{i\delta}). \quad (2)$$

グラフ特徴。 ノード数やエッジ数, 属性数など, グラフの基礎的な統計量がいくつかある。それらを入力することによって, GenCAT は多様なサイズを持つグラフを生成可能である。

6 実証研究

ノードラベルを持つグラフの以下の四つの主要な特徴の観点から GNN の強みと弱みを明らかにすることを目的とする: 1) クラスサイズ分布, 2) クラス間のエッジ接続割合, 3) 属性値, 4) グラフサイズ。最初に, 実証研究の設定を記述する。次に, ノード分類パフォーマンスについての実験を示し, その結果について詳細に議論する。最後に, GNN の効率性を調査するために多様なサイズを持つ人工グラフにおいて, GNN のエポックあたりの訓練時間を調べる⁵。

実験設定。 3章で述べた GPRGNN [12] や shaDow-GNN [10] などの最先端の手法を含む 17 種類の GNN を用いる⁶。また, グラフの構造がどれだけ大きく分類パフォーマンスに寄与するかを評価するために, グラフ構造を利用せずノード属性だけを用いた分類器である MLP についても実験を行う。データセットの分割については, [12] に従い, 訓練/検証/テストそれぞれ 60%/20%/20% の割合にランダム分割を行う。それぞれの設定で生成グラフについて GNN のハイパーパラメータチューニングのために, グリッドサーチを行う。実験の再現性のために, ハイパーパラメータのサーチスペースと各設定のベストパラメー

5: 全体の実行時間はエポックあたりの訓練時間と収束までのエポック数の積であるため, early stopping のための基準や学習率に大きく依存する。このことから全体の実行時間を直接分析することは困難であるため, 本実証研究ではエポックあたりの訓練時間について実験および考察を行う。

6: 紙面の制限から, GIN [6] や SIGN [8] などのいくつかの既存手法を用いていないが, 最先端の手法については漏れなく利用している。

タを, 我々のコードベースにて報告する。ランダム性を軽減するために, グラフ生成の各設定に対して三つのグラフを生成し, 各生成グラフに対して三回のリスタートで GNN を実行する。平均値を示すとともに, 標準偏差をエラーバーとして報告する。我々の実験では, Cora データセットからグラフ生成に用いるパラメータを抽出する。すなわち, Cora からクラス特徴とグラフ特徴を得て, GenCAT のデフォルトパラメータとする。Cora は 2708 ノード, 5278 エッジ, 1433 属性, および 7 クラスを持つグラフである。その後, 各設定のために一つもしくは少数のパラメータの調整を行う⁷。訓練時間の計測には, NVIDIA Tesla V100S GPU (32GB memory) および Intel(R) Xeon(R) Gold 5220R CPU×2 (378GB memory) を用いる。行列演算には GPU を使用し, それ以外の処理では CPU を使用する。

6.1 多様な特性を持つ人工グラフでの分類精度の比較

6.1.1 多様なクラスサイズ分布

ノード分類のための GNN に関するほとんどの研究は, 評価の中でクラスサイズ分布を考慮していない。実際に, 既存研究では分類精度の評価に accuracy のみが用いられている。しかし, 実践的なユースケースではクラスサイズ分布は典型的に不均等 [32] である。そのため accuracy を用いる代わりに, サイズの大きいクラスとサイズの小さいクラスを公平に扱うことに適した f1-macro を我々は用いる。公平性と accuracy のトレードオフは盛んに研究されている [33, 34]。

詳細な実験設定。 クラスが不均等であることがどれだけ分類精度に影響するかを調べるために, 少数のクラスが大部分のノードを含むクラスサイズ分布を持つグラフを用いて実験を行う。具体的には, 下記に示す調整したクラスサイズ分布 ρ^{conf} を持つ人工グラフを用いて実験を行う。

$$\rho_l^{\text{conf}} = \begin{cases} \alpha & (l = 1) \\ \alpha * (1 - \sum_{l'=1}^{l-1} \rho_{l'}^{\text{conf}}) & (1 < l < y) \\ 1 - \sum_{l'=1}^{l-1} \rho_{l'}^{\text{conf}} & (l = y) \end{cases}, \quad (3)$$

ここでの α は最大クラスのサイズを意味する。既存手法が無視している不均等なサイズのクラスを持つグラフで GNN の動作を調査するために, $\alpha \in [0.4, 0.5, 0.6, 0.7]$ の範囲で実験を行う。Cora データセットは 7 クラス持つため, $\alpha = 0.4, 0.5, 0.6, 0.7$ のとき生成グラフのクラスは不均等である。クラスサイズが均等な場合とも比較するために, 完全に均等なクラスを持つグラフについても実験を行う。

考察。 図 1 において, 多様なクラスサイズでのノード分類精度 (f1-macro) を示す。クラスサイズが均等, 不均等の両方において他の手法を一貫して上回る手法は存在しない。均等な設定では, 最新の手法である GPRGNN が, 高いモデル表現力を持つため最高の f1-macro を獲得している (図中, 最も左のデータポイント参照)。一方で, 興味深いことに不均等な設定 ($\alpha = 0.4, 0.5, 0.6, 0.7$) では, 最も単純なアルゴリズムである SGC が他の複雑な GNN を上回っている。

7: 明示的に言及していないパラメータについては, Cora から抽出した値を用いる。

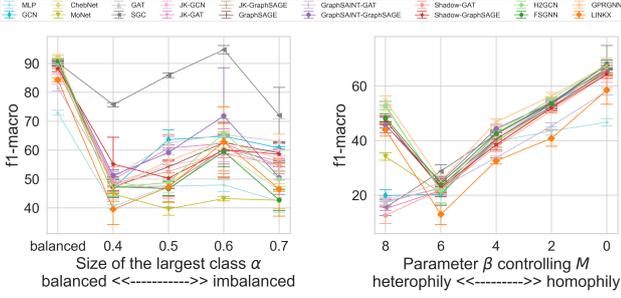


図 1: 多様なクラスサイズを持つグラフにおける分類精度。 図 2: 多様なエッジ接続割合を持つグラフにおける分類精度。

6.1.2 多様なクラス間のエッジ接続割合

多様なクラス間のエッジ接続割合を持つグラフでどれだけ効果的に GNN が動作するかを詳細に明らかにするために、エッジ接続割合について人工グラフの詳細なパターンを用いて実験を行う。

詳細な実験設定. 多様なエッジ接続割合を持つグラフを生成するために、以下の調整したクラス接続平均 M^{conf} を用いる。

$$M_{l_1 l_2}^{\text{conf}} = \begin{cases} \max(M_{l_1 l_2}^{\text{Cora}} - 0.1 * \beta, 0) & (l_1 = l_2) \\ M_{l_1 l_2}^{\text{Cora}} + 0.1 * \beta / (k - 1) & (l_1 \neq l_2) \end{cases}, \quad (4)$$

ここでの β はグラフの homophily/heterophily の度合いを調整するパラメータであり、 M^{Cora} は Cora から抽出したクラス接続平均を意味する。 M^{Cora} の対角成分の平均は 0.81 であることから、調整を実施しなかった人工グラフは Cora と同様 homophilic グラフである。もし β が小さいとき、クラスは多くのクラス内エッジを持ち、生成グラフは homophilic グラフになる。対照的に β が大きい場合は、クラス内エッジが少なくなり、生成グラフは heterophilic グラフになる。エッジ接続割合について多様なグラフを生成するために、 $\beta \in [0, 2, 4, 6, 8]$ の範囲で実験を行う。

考察. 図 2 は多様なクラス間のエッジ接続割合 (クラス接続平均) を持つグラフにおける、ノード分類精度を示す。最初に、homophily 設定では全てのモデルが良い精度を示している (図中、最も右のデータポイント参照)。一方で、heterophily 設定では、典型的な GNN である GCN や GAT が低い精度を示すが、heterophilic グラフを考慮する H2GCN や FSGNN, GPRGNN は高い精度を示す (図中、最も左のデータポイント参照)。しかし、それらの GNN は複雑なデザインを持つにも関わらず、heterophily 設定において MLP からわずかな改善しか得ることができていない。これは heterophily を考慮する GNN はほとんど構造情報を使えていないことを意味しており、たとえばグラフが強い heterophily の傾向を持っていたとしてもその情報をほとんど無視している。 $\beta = 8$ のとき M^{conf} の対角成分の平均は 0.03 であり、これはクラス内にほとんどエッジを持たないことを意味する。GraphSAGE は自己と近傍の埋込の分離を採用しているため、heterophily 設定において比較的高い f1-macro を得ている。最近の手法である LINKX は f1-macro において最先端の精度を得ることができなかった。本実験では

その論文内で示されているハイパーパラメータサーチスペースをそのまま使用したため、より広いサーチが分類精度を上げる可能性がある。最後に、クラス間にほとんどランダムにエッジが生成された場合、全てのモデルは低い f1-macro スコアを得た ($\beta = 6$ のときのデータポイント参照)。その理由は、このケースではクラス接続平均の値がほとんど一様な値になっているからである⁸。

6.1.3 多様な属性値

属性値について人工グラフの詳細なパターンを用いることによって、属性値がどれほど大きく GNN のパフォーマンスに影響するかを明らかにすることを目的とする。

詳細な実験設定. 多様な属性クラス相関 (式 (2)) を持つグラフを生成するために、Cora から抽出した属性クラス相関 H^{Cora} と一様な分布をある度合い γ で混ぜ合わせる。すなわち、クラスに関してバイアスのある属性から、ランダムな属性まで多様な属性値を持つグラフを生成する。この調整を行った属性クラス相関 H^{conf} は下記の混合計算によって定式化する。

$$H^{\text{conf}} = (H^{\text{Cora}} + \gamma c) / (1 + \gamma), \quad (5)$$

ここでの c は H^{Cora} の平均値を表し、 $c = \sum_{i=1}^d \sum_{j=1}^k H_{ij}^{\text{Cora}} / (d * k)$ として計算できる。もし $\gamma = 0$ のとき、 H^{conf} はオリジナルの属性クラス相関と一致する。もし γ が大きいとき、 H^{conf} は一様分布に近づく。結果として、 H^{conf} はノードラベルの予測に関して情報が少なくなる。本実験では $\gamma \in [16, 4, 1, 0]$ の範囲で調整した属性クラス相関を持つグラフを生成する。上記の設定とランダムな属性を比べるために、一様な属性クラス相関 (H^{conf} の全ての要素が c) を持つグラフについても実験を行う。

考察. 図 3 は多様な属性値におけるノード分類精度を示す。属性値がランダムに近いとき (γ が大きいとき)、ほとんどのモデルは低い f1-macro を得た。SGC がいくつかのケースで良い精度を得る理由は、モデルの単純さのために大きなクラスに過適合を避けることができるからである。6.1.1 章で述べたように、Cora は弱い不均等クラスを持つため、上記の結果が得られている。しかし、SGC は多様な設定で安定しない。他の考察としては、最新の手法である GPRGNN は他の手法と比べて一貫して高い精度を得ている。最後に、MLP は最も左と最も右のデータポイント (ランダムとクラスへのバイアスを持つ属性) 間でほとんどの GNN よりも大きな精度向上を獲得した。これは構造と属性のノード分類への貢献が幾分か重複していることを示している。

6.1.4 多様なグラフサイズ

グラフサイズがどのように GNN のパフォーマンスに影響するかを明らかにするために、多様なサイズを持ち、サイズ以外は同じ特徴を持つグラフについて実験を行う。

詳細な実験設定. 多様なサイズのグラフを生成するために、ノード数とエッジ数のペア (n, m) を $[(3000, 5000), (6000, 10000), (9000, 15000), (12000, 20000), (15000, 25000)]$ の範囲で実験を

⁸: GenCAT のエッジと属性生成ではクラス構造を共有するため、このケースでは属性もほとんどランダムに生成されている。

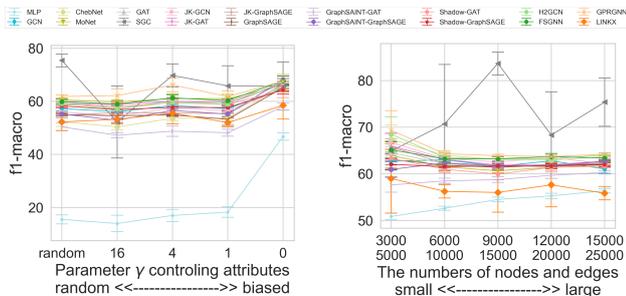


図 3: 多様な属性値を持つグラフにおける分類精度.

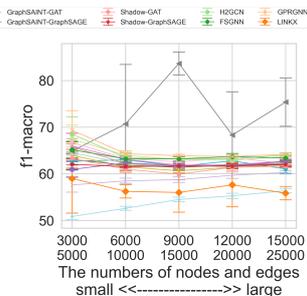


図 4: 多様なグラフサイズを持つグラフにおける分類精度.

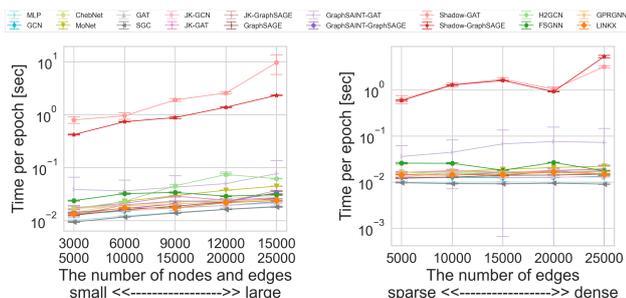


図 5: 多様なサイズを持つグラフにおける GNN のエポックごとの訓練時間.

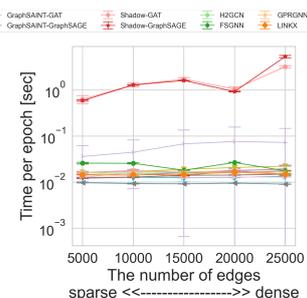


図 6: 多様なエッジ数を持つグラフにおける GNN のエポックごとの訓練時間.

行う。比較を簡単にするため、これらを最小のグラフサイズ (3000, 5000) の定数倍のサイズに設定している。

考察. 図 4 において、多様なグラフサイズを持つグラフにおける分類精度を示す。グラフが大きくなると、いくつかの GNN は f1-macro が低下している。ほとんどの GNN の損失関数は accuracy を増加させるために設計されているため、より大きいグラフがモデル訓練により多くの部分グラフパターンを提供するとき GNN は大きいクラスに過適合し、小さいクラスの優先度が低くなる。実際に accuracy を計測したとき、グラフが大きくなるとほとんどの GNN はより高いスコアを獲得している (我々のコードベースにおいて、accuracy の結果も提供している)。6.1.3 節と同じ理由で、SGC はいくつかのケースで良い精度を示すが、多様なグラフサイズにおいて安定した精度を示すことはできない。MLP はグラフサイズが大きくなると、分類精度を向上させている。これは大きなグラフからより多い訓練データを用いることにより、ノード属性によく適合しているからである。

6.2 多様なサイズを持つ人工グラフでの訓練効率の比較

多様なノード数とエッジ数. ノード数とエッジ数が訓練効率にどれだけ影響するかを調べるために、多様なサイズを持つ人工グラフでモデル訓練をしたときの GNN の実行時間を計測する。この実験は 6.1.4 節で用いたグラフと同じものを使用する。グリッドサーチを用いて、f1-macro を最大化するように GNN ハイパーパラメータをチューニングする。

図 5 は多様なサイズを持つグラフにおける GNN のエポック

ごとの訓練時間を示す。構造と属性の情報を保持する行列のサイズが大きくなるため、全てのモデルはグラフサイズが大きくなると、より長いエポックあたりの時間が必要になる。Shadow-GAT と Shadow-GraphSAGE はサンプリングした部分グラフにおいて大量の畳込み処理を行う必要があるため、他の GNN に比べてより長い実行時間を必要とする。

多様なエッジ数と固定のノード数. グラフのエッジ密度が訓練効率にどのような影響を与えるかを調査するために、モデル訓練にかかる GNN の実行時間を計測する。エッジ数は [5000, 10000, 15000, 20000, 25000] の範囲とし、ノード数は Cora が持つノード数と同じ値にする。

図 6 は多様なエッジ密度を持つグラフにおける GNN のエポックあたりの訓練時間である。隣接行列のサイズが同じであるため、ほとんどの GNN はグラフが疎である場合と密である場合で似たエポックあたりの訓練時間を必要とする。これらの手法はグラフ全体が GPU メモリに保持可能な限り、効率的に動作する⁹。サンプリングベース GNN である GraphSAGE, GraphSAINT, および Shadow-GNN は、より密なグラフのノードがより多くの隣接ノードを持つため、エッジ密度が高くなるとより長いエポックあたりの訓練時間を必要とする傾向がある。Shadow-GAT と Shadow-GraphSAGE のエポックあたりの訓練時間が 20000 エッジを持つグラフのときに減少している理由は、f1-macro を最大化するハイパーパラメータチューニングで小さい近傍サンプリング数が選ばれたからである。また、FSGNN について 15000 と 25000 エッジを持つグラフでエポックあたりの訓練時間が減少している理由は、他の設定に比べて小さい特徴集約のホップ数が選ばれたからである。

7 関連研究

GNN のベンチマーク. いくつかの研究 [14, 17, 35] で GNN のベンチマークは取り組まれてきた。研究 [17] は、ノード分類、グラフ分類、リンク予測の多様なグラフに関するタスクをサポートしている。しかし、それぞれのタスクに一つか二つのデータセットのみを用いているため、ノード分類に関して深い分析は提供されていない。GraphWorld [35] は GNN によるノード分類に関して限られた知見しか提供していない。それは以下の三つの観点を無視しているからである。一つ目は、GraphWorld は最近注目を集めている heterophilic グラフを考慮する GNN [7, 11, 12, 27] を無視していることである。二つ目は、クラスサイズ分布が分類結果に大きな影響を与えるにも関わらず、その研究ではクラスサイズを考慮しないことである。また、クラス不均等問題はグラフマイニング [36–38] を含む機械学習分野で研究されている問題である。三つ目は、機械学習手法の大きな関心ごとの一つである効率性について、GraphWorld は調査をしていないことである。他の研究 [14] は材料化学のための GNN にフォーカスしている。データセット

⁹: 8 章において、GPU メモリのサイズを超えるような大規模グラフに GNN を適用することを簡単に議論する。

の特徴のため、この研究ではグラフ回帰問題のみを検討している。

GNN についてのサーベイ GNN についての多くのサーベイ [39–42] が行われてきた。それらは既存の GNN をいくつかのカテゴリに分類する。例えば研究 [40] では、再帰グラフ深層学習、畳込みグラフネットワーク、グラフ自己符号化器、時空間グラフ深層学習のカテゴリを提案している。また、研究 [43] は GNN の近年の外観を説明している。サーベイ [42] は heterophily を考慮する GNN の最先端技術を要約することに取り組んでいる。これらの目的は GNN の研究の歴史を要約することと潜在的な研究方針を議論することであり、実験結果は提供されていない。

以上のことから、既存の研究では GNN の適用可能性や課題を明らかにするために、多様な特徴を持つグラフを用いた実験は行われていない。

8 終わりに

制御された特徴を持つ多様な人工グラフによって、我々の実証研究では将来の GNN アルゴリズム開発に役立つであろういくつかの興味深い知見を得た。これは既存の限られたベンチマークデータセットでは提供できない詳細な分析である。また GNN の将来の研究のために、オープンソースの PyTorch ベースで実装したライブラリを公開する。

8.1 未解決の問題。

我々の実証研究を通して、ノード分類のための新しい GNN の開発に関していくつかの未解決の問題を明らかにした。

クラス不均等。 我々は、ほとんどの GNN においてクラス不均等問題は分類精度を低下させることを示した。これは既存の GNN の損失関数がクラス不均等設定での分類精度を最大化するために設計されていないからである。数少ない研究 [36–38] が GNN によるノード分類タスクにおけるクラス不均等問題に取り組んでいるが、それらの主要なフォーカスは homophilic であり小規模なグラフである。すなわち、クラス不均等, heterophily, 大規模グラフの組合せなどのより複雑な設定を無視している。例えば、広く用いられている heterophilic グラフ, Texas, Wisconsin, や Cornell は全てグラフ内に 5 種類のクラスを持つにも関わらず、最大クラスはそれぞれ 55%, 55%, 47% のノードを持っているため、クラスサイズが不均等である。そのため、クラス不均等, heterophily, 大規模グラフの組合せのような多様で複雑な設定において、良いパフォーマンスを発揮する GNN をどのように開発するかは未解決の問題である。

Heterophily 設定。 最近の GNN [7, 11, 12] は homophilic/heterophilic グラフの両方をサポートするために提案されてきた。しかし、強い heterophily を示すグラフでは MLP のパフォーマンスが、そのような GNN と同程度であった (図 2 の最も左のポイントを参照)。これは GNN が heterophilic グラフからの構造情報をほとんど無視しており、heterophily 設定においてパフォーマンス改善の余地があることを示唆してい

る。そのため、homophilic グラフで最先端のパフォーマンスを維持しつつも、heterophilic グラフからクラス構造をよりよく掴むことができる GNN をどのように開発するかは未解決の問題である。

8.2 本研究の限界。

紙面が限られているため、本稿ではいくつかの課題を検討していない。例えば、ノード分類は GNN の最も注目を浴びているタスクの一つであるが、本研究はノード分類一つだけにフォーカスして、他のタスクを考慮していない。一方でノード分類に注力することによって、我々は GNN の強みと弱みを明らかにする深い分析を行うことを可能にした。

また、本研究はエッジの向きやエッジラベル、時間変化についてサポートしていない。いくつかの最近の研究 [44–46] は異種グラフ (heterogeneous graphs) と呼ばれる、多様な意味で異なるタイプを持つオブジェクトが互いに相互作用するグラフにおいて動作する GNN の開発に取り組んでいる。多様な異種グラフを用いて実験を行い、そのような GNN をベンチマークすることは重要な課題である。

本研究では、GNNs によるノード分類において結果の分析に広く用いられているクラス、構造および属性の関係に注目して実証研究を実施した。しかし、これらの関係だけでなくグラフには、クラスタ係数やグラフ半径といったトポロジー特徴量も存在する。多様なグラフトポロジー指標について、GNN のパフォーマンスを調べることは我々の今後の課題である。

近年では、GNN のスケーラビリティを改善するためにいくつかの GNN アルゴリズム [8, 10] が提案されている。我々の実証研究では、グラフ全体とノード属性が同時に GPU メモリに保持できる、小規模グラフのための設定のみを考慮している。大規模グラフにおける効率性と精度について、スケーラブルな GNN のベンチマーキングに取り組むことは我々の今後の課題である。

謝辞。 本研究は JSPS 科研費 JP20H00583 および JST さきさき JPMJPR18UD の助成を受けたものです。

文 献

- [1] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [2] Federico Monti, Davide Boscaioli, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, 2017.
- [3] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.
- [4] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.
- [5] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *ICLR*, 2019.
- [6] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ICLR*, 2019.

- [7] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Le-man Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NeurIPS*, 2020.
- [8] Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Benjamin Chamberlain, Michael Bronstein, and Federico Monti. Sign: Scalable inception graph neural networks. In *ICML 2020 Workshop on Graph Representation Learning and Beyond*, 2020.
- [9] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. Graph neural networks with heterophily. In *AAAI*, 2021.
- [10] Hanqing Zeng, Muhan Zhang, Yinglong Xia, Ajitesh Srivastava, Andrey Malevich, Rajgopal Kannan, Viktor K. Prasanna, Long Jin, and Ren Chen. Deep graph neural networks with shallow subgraph samplers. *CoRR*, Vol. abs/2012.01380, , 2020.
- [11] Sunil Kumar Maurya, Xin Liu, and Tsuyoshi Murata. Improving graph neural networks with simple architecture design. *arXiv preprint*, 2021.
- [12] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *ICLR*, 2021.
- [13] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NeurIPS*, Vol. 28, 2015.
- [14] Victor Fung, Jiaxin Zhang, Eric Juarez, and Bobby G Sumpter. Benchmarking graph neural networks for materials chemistry. *npj Computational Materials*, Vol. 7, No. 1, pp. 1–8, 2021.
- [15] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *ICML*. PMLR, 2020.
- [16] Peixiang Zhong, Di Wang, and Chunyan Miao. Eeg-based emotion recognition using regularized graph neural networks. *IEEE Transactions on Affective Computing*, 2020.
- [17] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.
- [18] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*. PMLR, 2016.
- [19] 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. *arXiv preprint arXiv:2005.00687*, 2020.
- [20] Derek Lim, Xiuyu Li, Felix Hohne, and Ser-Nam Lim. New benchmarks for learning on non-homophilous graphs. In *Workshop on Graph Learning Benchmarks (GLB 2021) at WWW*, 2021.
- [21] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 2017.
- [22] Seiji Maekawa, Yuya Sasaki, George Fletcher, and Makoto Onizuka. Gencat: Generating attributed graphs with controlled relationships between classes, attributes, and topology. *Information Systems*, Vol. 115, p. 102195, 2023.
- [23] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *ICML*, 2019.
- [24] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, Vol. 29, 2016.
- [25] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *ICML*, 2018.
- [26] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. GraphSAINT: Graph sampling based inductive learning method. In *ICLR*, 2020.
- [27] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *NeurIPS*, Vol. 34, 2021.
- [28] Aleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.
- [29] Hongbin Pei, Bingzhen Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *ICLR*, Vol. abs/2002.05287, 2020.
- [30] Chaokun Wang, Binbin Wang, Bingyang Huang, Shaoxu Song, and Zai Li. Fastsgg: Efficient social graph generation using a degree distribution generation model. In *ICDE*. IEEE, 2021.
- [31] Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. Contextual stochastic block models. *NeurIPS*, Vol. 31, , 2018.
- [32] Haibo He and Eduardo A Garcia. Learning from imbalanced data. *IEEE TKDE*, Vol. 21, No. 9, pp. 1263–1284, 2009.
- [33] Tameem Adel, Isabel Valera, Zoubin Ghahramani, and Adrian Weller. One-network adversarial fairness. In *AAAI*, 2019.
- [34] Pieter Delobelle, Paul Temple, Gilles Perrouin, Benoît Frénay, Patrick Heymans, and Bettina Berendt. Ethical adversaries: Towards mitigating unfairness with adversarial machine learning. *KDD Explorations Newsletter*, Vol. 23, No. 1, 2021.
- [35] John Palowitch, Anton Tsitsulin, Brandon Mayer, and Bryan Perozzi. Graphworld: Fake graphs bring real insights for gnn. *KDD*, 2022.
- [36] Zheng Wang, Xiaojun Ye, Chaokun Wang, Jian Cui, and S Yu Philip. Network embedding with completely-imbalanced labels. *IEEE TKDE*, Vol. 33, No. 11, 2020.
- [37] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *WSDM*, 2021.
- [38] Liang Qu, Huaisheng Zhu, Ruiqi Zheng, Yuhui Shi, and Hongzhi Yin. Imagn: Imbalanced network embedding via generative adversarial graph networks. In *KDD*, 2021.
- [39] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, Vol. 1, pp. 57–81, 2020.
- [40] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, Vol. 32, No. 1, pp. 4–24, 2020.
- [41] Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2020.
- [42] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2022.
- [43] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Vol. 14, No. 3, pp. 1–159, 2020.
- [44] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In *NeurIPS*, Vol. 32, 2019.
- [45] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun.

Heterogeneous graph transformer. In *The Web Conference*, 2020.

- [46] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *KDD*, 2021.