大域的グラフ構造を利用した GNN による IT/NW システム構成の高精度な優劣推定手法

丸山貴志† 黒田貴之† 里田浩三†

†NEC 〒211-8666 神奈川県川崎市中原区下沼部 1753
E-mail: †{49takashi, kuroda, satoda}@nec.com

あらまし 本稿では，IT/NW システムの自動設計技術におけるシステムトポロジの優劣推定タスクに対し，大域的グラフ構造を利用したグラフニューラルネットワーク（GNN）モデルを提案し，提案モデルの有効性の検証結果を報告する．本稿で対象とするデータ構造はアプリケーションやサーバ，ファイアウォール，ルータに代表される IT/NW 機器で構成されるシステムを表現するグラフ構造であり，システムとしての冗長性や頑健性の担保を目的に対称的に配置された IT/NW 機器の位置情報を捕捉するため，提案モデルではグラフ縮約技術に基づいたグラフプーリング技術を採用する．評価実験では，自動設計技術の問題設定に従い，強化学習アルゴリズム下で提案モデルと従来モデルそれぞれの性能比較を行った．評価の結果，GNN モデルの収束効率と汎化能力との両観点について，提案モデルがシステムの優劣推定精度を大幅に改善させられることを確認した．

キーワード グラフニューラルネットワーク，自動設計，IT/NW システム，強化学習

1. はじめに

AI と IoT の技術進化を背景にデジタル・トランスフォーメーション（DX）が世界的な潮流となりつつあり，企業内の業務効率化に向けた IT/NW 技術の活用の場が従来のバックオフィスからフロントオフィスへと移り変わり始めている [1]．業務形態が頻繁に変わるフロントオフィスでは，IT/NW 技術の動作を担保するシステムに求められる要件は多様であり，変化しやすいことから，要件の多様な変化に追従するためにシステムの迅速な提供が求められる [2]．

IT システムの提供を抜本的に変革する技術領域にシステムの自動運用技術がある．近年では，機械学習技術に基づく自動化手法が数多く提案されており，無線ネットワーク[3,4,5]，IoT ネットワーク[6, 7, 8]，有線ネットワーク[9,10,11]，仮想化ネットワーク[12,13,14]など，対象とされるシステムは多岐にわたる．これらの自動運用技術では，システムグラフとよばれる，IT/NW 機器間の通信による相互作用を表現したグラフがデータとして扱われており，機械学習技術はシステムグラフに対する所定のメトリックに関する値の推定や学習に利用される．例えば，運用技術の一手法である最適化技術に基づくシステム設計手法であれば，システムグラフを探索する過程で参照される目的（コスト）関数の学習に機械学習技術が利用される．

とりわけ最近の研究では，システムグラフの構造的特徴をより有効に利用するため，グラフニューラルネットワーク（GNN）を利用した手法が考えられはじめている[4, 5, 6, 9, 11, 12, 13]．GNN は，システムグラフに限らず SNS や文献グラフ，分子化合物など多種多様なグラフに対して利用されており，回帰や分類に代表されるタスクに対して頂点やグラフの表現ベクトルを

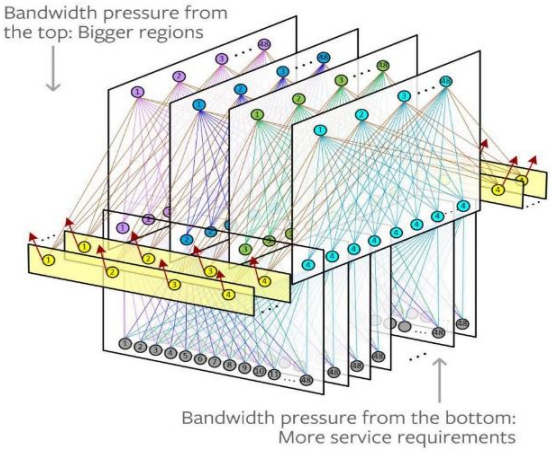

図 1：IT/NW システムの例であるデータセンターのネットワークトポロジ[15]．頂点だけでなく部分構造レベルの対称性も有する．

獲得する役割を担う．GNN モデルは，この表現を得るにあたり，入力されるグラフに対して，頂点の表現が頂点の近傍構造に依存するという性質を仮定している．この性質はホモフィリーと呼ばれ，例えば SNS アカウントや文献で構成されるグラフといった，隣り合う頂点どうしが同一のコミュニティに属する傾向の高いグラフで特に顕著にみられる．この仮定の下，頂点の表現には近傍構造を，グラフの表現には各頂点の表現ベクトルの集合を利用するアプローチが採られている．

しかし，ホモフィリーを前提とする GNN モデルでは，システムグラフに対するメトリックの値の正確な

This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.

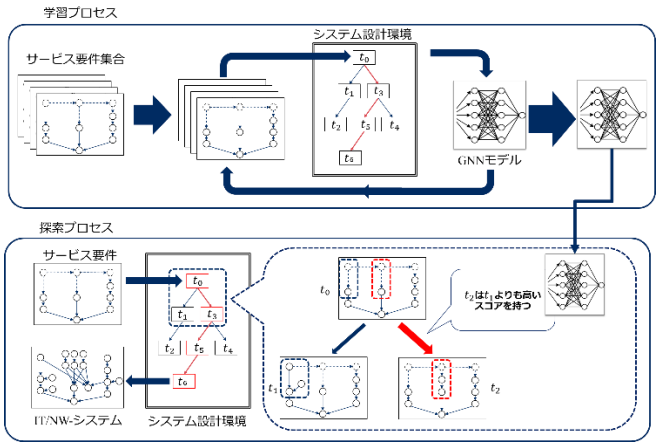

図 2 ：自動設計技術 Weaver の全体像．

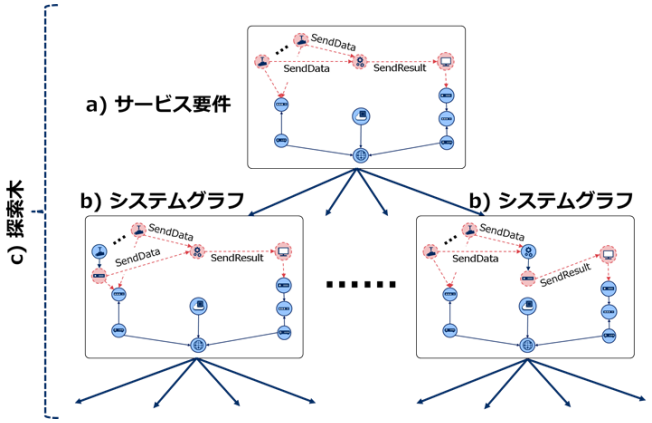

図 3：探索アルゴリズム内の探索木の模式図．破線で図示された頂点及びエッジが欠損エンティティを，実線で図示された頂点及びエッジが完備エンティティをそれぞれ表す．

推定が困難である．これは，システムの高い対称性によって，非同型なシステムグラフが同一の表現を持つことがあるからである．一般的に，IT/NW システムは高い保守性や冗長性を担保するために対称的な構造をとることが多い．対称的な位置関係にある頂点は同一の近傍構造をもつ（図 1 ）ため，ホモフィリーを前提とした GNN は同一の近傍構造を有する頂点を全て同一の表現ベクトルへと変換する．つまり，対称的な位置関係にある頂点は位置関係の違いに関わらず同一の表現を持つ．よって，システムグラフそれ自身は多数の頂点の位置関係が無視されたものとして表現されることとなるため，頂点の位置関係にしか差がみられないシステムグラフはすべて同一のグラフ表現を持つ．

そこで本稿では，近傍情報だけでは見落とされる頂点の位置情報を捕捉可能にするため，GNN の一処理としてグラフ縮約を採用したモデルを提案する．GNN モデルへグラフ縮約技術を導入することにより，巨視的観点から見た粗い粒度のグラフ構造を捕捉できるようになるため，近傍構造に含まれない遠くに位置する頂点であっても頂点の表現ベクトルの獲得に利用することが可能となる．モデルの有効性の検証では，提案モデルを検討する元の動機となった我々のシステム自動設計技術[6]の問題設定に倣い，強化学習環境下でモデルの学習効率・推定精度を評価する．評価の結果，GNN によるシステムグラフの表現に，頂点間の位置情報を扱うことが有効であることを示す結果が得られた．

本稿の構成を以下に述べる：第 2 節では，本稿の前提技術であるシステム自動設計技術[6]と GNN モデルの技術詳細を説明する．第 3 節では，システム自動設計技術における GNN モデルの課題を明らかにし，4 節で課題に対する改善手法を提案する．第 5 節では，提案手法の妥当性及び有効性を検証するための実験内容及び実験結果を説明し，第 6 節にてまとめを述べる．

2. 背景技術

本節では，既存の IT/NW システム自動設計技術[6]（以下，Weaver と呼ぶ）の技術詳細と，Weaver で利用される GNN モデルを説明する．

2.1. IT/NW システム自動設計技術

Weaver は，入力として IT/NW ユーザの所望するサービスが表現されたシステムに対する要件（以降，サービス要件と呼ぶ）を受け取り，そのサービス要件を完全に満足するシステムグラフを導出する自動設計技術である（図 2）．入力であるサービス要件はグラフデータで表現され，グラフの頂点が IT/NW 機器を表わし，頂点間の辺が http 接続や tcp 接続といった所望する IT/NW 機器間の関係性を表わす．

Weaver の設計処理は探索処理が担っており，全ての探索状態は探索木で管理される．探索木内の各探索状態は，入力のサービス要件を部分的に或いは完全に満足するシステムグラフである（探索木内では，サービス要件もシステムグラフとして扱われる）．また，探索状態の生成は，現在状態にあたるシステムグラフに対し，システムとして動かすには不足している情報や構造を部分的に補完することとして定義されている．以降，補完の必要な頂点やエッジを欠損エンティティと呼び，完全に補完されきったエンティティを完備エンティティと呼ぶ．図 3 の b)が示すシステムグラフは，a)に含まれる欠損エンティティそれぞれに補完を行った後のシステムグラフである．探索木内でサービス要件から順番に子を辿っていくと，サービス要件が部分的にかつ段階的に補完されていく様子が観察される．Weaver の出力は，全てのエンティティが完備なシステムグラフである．

探索処理内の現在状態から次状態への遷移は，まず現在状態にあたるシステムグラフから子にあたるシス

テムグラフを生成し，生成されたシステムグラフの中から 1 つグラフを選択することによって定義されている．このシステムグラフの選択はシステムにまつわるメトリックの値の優劣に基づいて行われ，優劣はメトリックの値の大小で定義されている．メトリックの例としては，システム全体のコスト（金額）やシステムの稼働率などが挙げられ，生成されたシステムグラフのうちメトリックの値の最も大きいものが次状態として選択される．

システムグラフをシステムメトリックで比較する際には，メトリックの具体的な値を評価する関数が利用される．この関数を評価関数と呼び，以下ではf_{ev}で評価関数を表わすものとする．Weaver では，f_{ev}を強化学習アルゴリズムによって獲得する．一般に，強化学習技術の目的は，探索問題において探索状態を最適化することと同時に，最適な探索状態へと導くことが可能な評価関数を学習することにある[16]．詳細には，探索状態sとsに対する行動aとのペア(s,a)の優劣を判断する関数$r(s,a)$（報酬関数）が所与の状況において，sとa，そして$r(s,a)$からなる系列（エピソードと呼ぶ）

$$s_0, a_0, r(s_0,a_0), s_1, a_1, \ldots, s_{t-1}, a_{t-1}, r(s_{t-1},a_{t-1}), s_t$$

で，収益$\sum_{i=0}^{t} r(s_{i-1},a_{i-1})$が最大となるものを見つけ出すことが強化学習の 1 つの目的であり，この過程で，探索状態sから探索を始めたときに達成可能な最も高い期待収益を出力する評価関数を学習する：

$$V(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s\right].$$

ここで，$R_{t+1} = E[r(s,a)]$で，$\gamma(\in [0,1])$は割引率を表す．Weaver では，探索木内のシステムグラフを状態とみなし，またシステムグラフから異なるシステムグラフへの遷移を行動とみなした上で，評価関数$f_{ev}(s)(=V(s))$の訓練を行う．

2.2. グラフニューラルネットワークモデル

Weaver の探索処理で利用される評価関数f_{ev}は GNN モデルによってモデリングされている．一般に，GNN モデルはグラフGと頂点vの属性ベクトルX_vを利用し，頂点のベクトル表現h_v或いはグラフそれ自身のベクトル表現h_Gを学習する．GNN モデルは標準的に近傍集約に基づいたベクトル表現を採用しており，そこでは頂点の近傍構造を反復的に集約することによってベクトル表現を更新する[17,18,19]．形式的には，反復回数がk回目の時点の処理は

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{h_u^{(k-1)} \mid u \in N(v)\right\}\right) \quad \ldots (1),$$

$$h_v^{(k)} = \text{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right) \quad \ldots (2),$$

と表現される．ここで，$h_v^{(k)}$は反復回数がk回目時点の

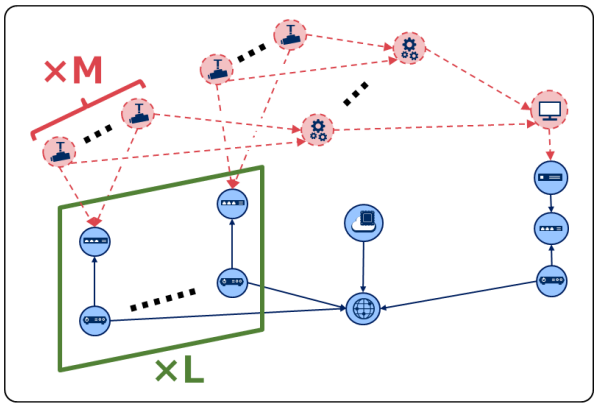

図 4：[6]の評価実験で利用したサービス要件．既存システム上へカメラを利用した不審者監視・検知サービスを追加することを表現した要件．

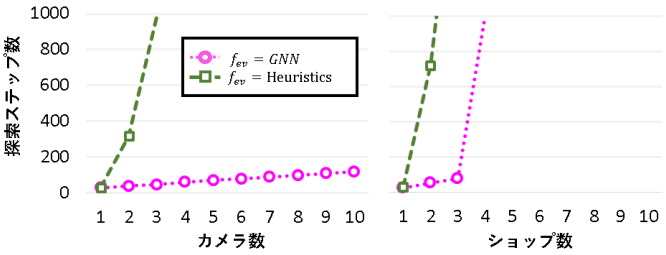

図 5：評価関数$\boldsymbol{f_{ev}}$を GNN とヒューリスティクスとしたときの探索ステップ数．いずれの図も図 4 のサービス要件を入力とした場合の探索ステップ数を表しており，左側はカメラ数$\boldsymbol{M}$を増減させた場合，右側が図 4 のショップ数 L を増減させた場合のステップ数を表す．

頂点vの特徴ベクトルを表わす．$k=0$の場合$h_v^{(k)} = X_v$であり，$N(v)$はvに隣接する頂点集合を表す．このことから$h_v^{(k)}$はvからkホップ以内に位置する頂点の情報が圧縮された表現と言える．AGGREGATE は総和，平均，最大・最小などの作用素，COMBINE は直和や射影作用素などの形態をとる．Weaver においてシステムグラフGに対するf_{ev}の評価値は，Gの頂点全ての特徴ベクトル集合を変換することによって定義される：

$$f_{ev}(G) = \text{summarize}\left(\left\{h_v^{(k)}\right\}_{v \in V_G}\right), \ldots \quad (3)$$

ここで，V_GはGの頂点集合を表し，summarizeは総和や平均，またそれらの作用素を通したのち更に多層パーセプトロンを通すなどの形態をとる．

3. Weaver において GNN の抱える課題

Weaver では，複数のサービス要件を初期状態として，GNN でモデリングされた評価関数$f_{ev}^{(GNN)}$を強化学習で訓練している．図 4 は[6]の実験で用いられたサービス要件を表し、図 5 は強化学習で訓練した$f_{ev}^{(GNN)}$による探索性能をヒューリスティクス$f_{ev}^{(heu)}$のそれと比較した図を表す．$f_{ev}^{(GNN)}$の訓練時には，強化学習アルゴリズムの初期状態s_0として，$(M,L) = (1,1),(1,2),(2,1)$の場合の図 4 のサービス要件が利用されている．

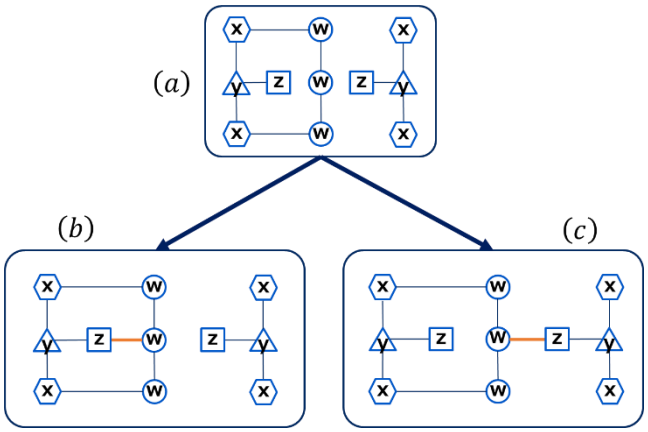

図 6：図 4 のサービス要件の部分構造(a)とサービス要件から生成されるシステムグラフの部分構造(b)(c). (b)と(c)は異なる NW ドメインに所属する IT/NW 機器への接続を行うか否かを表現しており，各頂点の文字は頂点の属性ベクトルを表わす．

図 5 から，$f_{ev}^{(GNN)}$をモデリングする GNN モデルはサービス要件の局所的な構造の変化に敏感である一方，グラフの大域的な構造変化の捕捉を不得手とすることが推察される．実際，訓練後の$f_{ev}^{(GNN)}$がカメラ数Mの増減に対して頑健な探索を実現できることが伺える一方，ショップ数Lを増減させた場合$L \geq 4$で探索ステップ数が爆発的に増加することがみてとれる．カメラ数とショップ数の変化はそれぞれ図 4 のサービス要件の構造変化を引き起こし，カメラ数は局所的な変化を，ショップ数は大域的な変化を引き起こす．

実際，GNN モデルは大域的構造の捕捉を苦手としており，特にシステムグラフのような部分構造レベルで高い対称性を有するグラフの識別を苦手とする．この弱点は GNN の集約処理が近傍構造にのみ依拠した処理であることに起因している．この集約処理方式が実際に問題として見て取れる例が図 6 である．図 6 は，図 4 のサービス要件から生成される探索木の部分木を表し，図 6 の(b)と(c)は IT/NW システムとして稼働するかどうかを左右するシステムグラフを簡素化したものである．このとき，(c)はその子孫に完備なシステムグラフを持たないが，子孫に完備なシステムグラフを持つ(b)と，(b)と(c)は非同型であるのにも関わらず，GNN によって同一視される．以降の残りのパラグラフで，実際に(b)と(c)が同一視される理由を説明する：(b)と(c)は類似したグラフでありそれぞれ高い対称性を有するため，位置関係の異なる頂点が存在するのにもかかわらず，(b)と(c)の頂点の近傍集合の多重集合が一致する：

$$\{N(v)\}_{v\in V_G} = \{N(v)\}_{v\in V_{G'}}.$$

[1] 簡素化される前のグラフでは$k = 2$以下で成立。

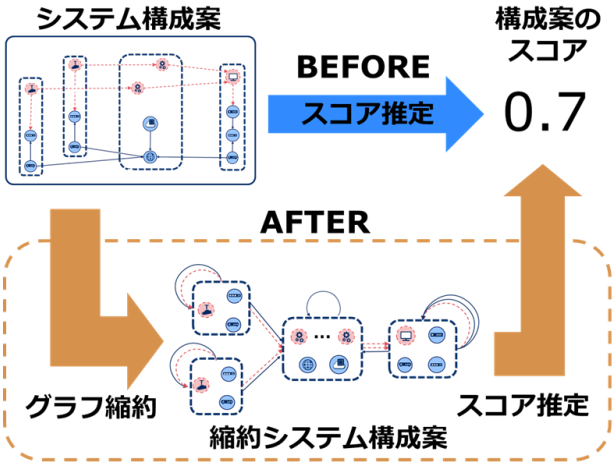

図 7：従来のスコア推定手法と提案スコア推定手法．

この等号関係は，特に$k = 1$のとき[1]，特徴ベクトルとのペアでも成立する：

$$\left\{\left(h_v^{(k-1)}, N(v)\right)\right\}_{v\in V_G} = \left\{\left(h_v^{(k-1)}, N(v)\right)\right\}_{v\in V_{G'}}.$$

すると，グラフそれぞれを(1)式と(2)式へ通して得られる頂点の特徴ベクトルの多重集合は一致する：

$$\left\{h_v^{(k)}\right\}_{v\in V_G} = \left\{h_v^{(k)}\right\}_{v\in V_{G'}}$$

評価関数$f_{ev}^{(GNN)}$はグラフ上の全ての頂点の特徴ベクトルを入力に持つ関数であるため，(3)式より(b)と(c)は$f_{ev}^{(GNN)}$によって同じスコアを持つ．

素朴に考えれば，反復回数kを増やし集約範囲を拡大すればより広域に位置する情報を集められるため，結果としてグラフの識別が可能になると考えられるが，この手法は現実的な解決策とは言えない．その理由は 2 点挙げられる：kを大きく設定することで GNN から抽出される特徴量は異なり得るが，kは GNN モデルの訓練や推論を行う際には固定されているパラメタである．そのため，GNN モデルから出力される特徴量の捕捉に必要な反復回数がグラフによって異なる場合，前述したような，頂点の特徴ベクトル集合が一致する問題が起こり得る．もう 1 点の理由は，GNN のモデルとしての学習能力に関して問題があり，[17]でも報告されているようにkの増加に伴い学習が不安定になるからである．この現象は，パラメタ更新の際にパラメタが過度に平滑化されてしまうこと(over-smoothing，[21])や集約処理において不可欠な構造的情報以外の情報が指数関数的に集まること(over-squashing，[22])などの GNN モデル特有の問題により引き起こされることが知られている．

4. 提案手法

我々は，近傍構造だけでは見落とされてしまう頂点

の位置関係を捕捉するため，グラフの巨視構造を集約する GNN モデルを提案する（図 7）．提案手法では，グラフの縮約処理を採用することでグラフの巨視的な構造を扱えることを可能とし，もともと遠い位置にあった頂点を近い位置にいるよう GNN に認識させることができるため，位置関係の捕捉が容易になる．

4.1. 縮約処理

我々はグラフの縮約処理を，GNN の各反復処理で特徴ベクトル$h_v^{(k)}$が得られる度に行うこととする．具体的には以下の演算によって縮約処理を実施する：

$$H_{G^{(k+1)}} = S_k^{\mathrm{T}} H_{G^{(k)}} \quad \dots \quad (4),$$

$$A_{G^{(k+1)}} = S_k^{\mathrm{T}} A_{G^{(k)}} S_k \quad \dots \quad (5).$$

ここで，$H_{G^{(k)}}$はグラフ$G^{(k)}$の頂点の特徴ベクトルh_v^{T}を行方向に並べた行列であり，$A_{G^{(k)}}$はグラフ$G^{(k)}$の隣接行列，$G^{(k+1)}$は$G^{(k)}$の縮約後のグラフ，S_kは行が$G^{(k)}$の頂点，列が$G^{(k+1)}$の頂点ぞれぞれで添え字付けられた$(G^{(k)}$の頂点数$)\times(G^{(k+1)}$の頂点数$)$行列で，(i,j)成分は$G^{(k)}$の頂点iがグラフ$G^{(k+1)}$の頂点jへ縮約される時に 1，そうでなければ 0 であるもの，である．(4)式は縮約先の各クラスタ内に割り当てられる頂点の属性ベクトルの総和を表す．(5)式は隣接行列の生成式であり，(5)式によって$G^{(k)}$内の頂点u,vがそれぞれ$G^{(k+1)}$の頂点U,Vへ割り当てられるとき，u,v間に辺が存在すればU,V間にも辺が存在するようグラフの隣接行列が生成される．

4.2. 頂点割当行列

4.1 節で述べた縮約を実現するためには，$G^{(k)}$の頂点から$G^{(k+1)}$の頂点への割当て行列S_kを構成する必要がある．その際特に，縮約後のグラフ$G^{(k+1)}$には，頂点間の位置情報を捕捉するために必要な本質的構造が現れていることが肝要である．そのために我々はバックボーン構造を利用する[23]．バックボーン構造はグラフの中の頂点の属するコミュニティを決定づける頂点から構成される部分グラフである．我々は，$G^{(k)}$の頂点のうち，物理レイヤに近い頂点を抽出し，その頂点集合をバックボーン構造の頂点集合V_bとした．その上で，V_bの各頂点を縮約先のクラスタとみなし，$G^{(k)}$の各頂点をV_bの頂点のうち一番近いものへ割り当てることによってS_kを定義することとした[24]．S_kの各成分は，形式的には以下のように定める：

$$s_k^{(i,j)} = \begin{cases} 1, & d_{geo}(i,j) \le d_{geo}(k,j), \quad \forall k \in V_b, \\ 0, & \text{otherwise.} \end{cases}$$

ここで，$d_{geo}(i,j)$で頂点i,j間の測地距離を表す．

5. 評価

本節では，[6]の問題設定に従い，強化学習アルゴリズム下で提案モデルと従来モデルそれぞれの性能比較評価を行う．

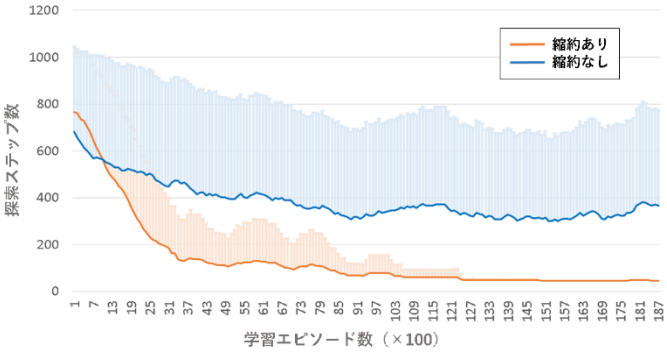

図 8：従来 GNN モデル（縮約なし）と提案 GNN モデル（縮約あり）の，学習回数に対する探索ステップ数．

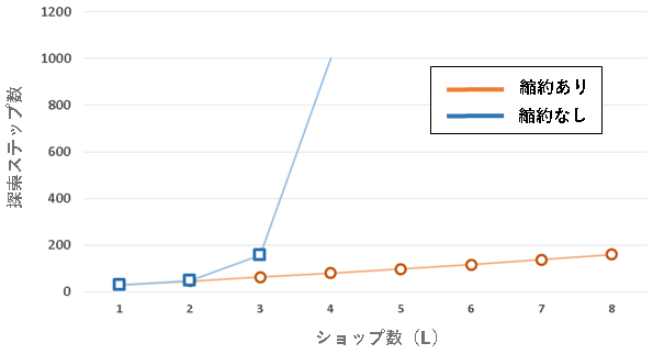

図 9：訓練後の従来 GNN モデル（縮約なし）と提案 GNN モデル（縮約あり）それぞれの，ショップ数の増減に対する探索ステップ数．

5.1. 問題設定

今回の実験では，GNN モデルを収束効率と汎化能力との 2 つの観点で評価する．収束効率については，強化学習アルゴリズム下で，従来の評価関数では学習を収束させられなかったサービス要件$(M,L)=(1,2)$を基に評価を行うこととする．ここでの評価基準は，f_{ev}の学習回数に対する，f_{ev}による探索にかかるステップ数とする．従来 GNN モデルと提案 GNN モデルのそれぞれによってモデリングしたf_{ev}を学習回数ごとに探索ステップ数で比較する．

汎化能力の評価は，[6]で行われた実験のうちの，ショップ数の増減に対するf_{ev}による探索ステップ数の変化を評価することによって行う．この評価では，$(M,L)=(1,1),(1,2),(1,3)$の場合の図 4 のサービス要件でf_{ev}を訓練することにより，ショップ数Lの増減に対してどれほど頑健な探索ができるようになるか評価する．

実験で用いる GNN モデルは GNN ライブラリの 1 つである GraphNets[18]で構築することとし，従来モデル，提案モデル共に 2 層の GN-block（言い換えると，反復回数$k=2$）で構成されるモデルを採用した．また，縮約処理は$k=0$で行うこととした．モデルパラメタの更新に関わるオプションについては，損失関数に 2 乗誤差関数，オプティマイザに Adam をそれぞれ採用した．

5.2. 収束効率評価

この実験では、従来モデルと提案モデルそれぞれで10回強化学習を回し、各施行で得られる特定の学習エピソード数時点での探索ステップ数の平均及び分散を比較し評価する。

図8は従来モデルと提案モデルそれぞれを学習回数に対する探索ステップ数で比較した図である．図から伺えるように，従来モデルは学習回数を増加させても探索ステップ数とその分散が途中から減少しなくなるが，提案モデルでは学習回数が増えるにつれ探索ステップ数の平均及び分散が減少することが確認された．このことから，提案モデルは頂点の位置関係を捕捉できるようになったと考えられる．

5.3. 汎化能力評価

図9は，訓練済みの従来モデルと提案モデルそれぞれによる探索ステップ数をショップ数Lで比較したものである．図からわかるように従来モデルでは$L \geq 4$で探索ステップ数が 1000 以上に達するのに対し，提案モデルでは，$L \geq 4$ 以降でも探索ステップ数が十分に，そして線形に抑えられることが確認された．この結果は従来モデルでは識別不能なシステムグラフが識別されるようになったことを示唆しており，学習モデルの汎化に縮約処理が効果的であることを示している．

6. まとめ

本稿では，IT/NW システムの自動設計技術におけるシステムトポロジの優劣推定タスクに対し，グラフプーリング（縮約）を利用した GNN モデルを提案した．従来のモデルでは見落とされてしまう頂点の位置関係を捕捉するため，グラフの巨視構造を集約する GNN モデルを提案し，評価実験では，提案モデルの収束効率と汎化能力との両観点で，提案モデルがシステムの優劣推定精度を大幅に改善させられることを確認した．

今後は，本稿ではドメイン知識に基づいて決定したバックボーン構造を，自動的に獲得する手法の検討を行う．また，頂点割当行列についてもグラフの構造に応じて自動的に構成する手法を検討する．

謝　辞

本研究開発は，総務省の「革新的 AI ネットワーク統合基盤技術の研究開発（JPMI00316）」によって実施した成果を含みます．

文　献

[1] “企業 IT 動向調査 2020（2019 年度調査）～データで探るユーザー企業の IT 動向～，”一般社団法人日本情報システム・ユーザー協会, April 2020.

[2] “なせ，いまアジャイルが必要か?，”情報処理推進機構, February 2018.

[3] A. Zappone et al., "Wireless Networks Design in the Era of Deep Learning:Model-Based, AI-Based, or Both?," IEEE Transactions on Communications, 2019.

[4] H. Wang et al., "A Graph Neural Network-based Digital Twin for Network Slicing Management,", IEEE Transactions on Industrial Informatics.

[5] Y. Shao et al., "Graph Attention Network-based DRL for Network Slicing Management in Dense Cellular Networks," 2021 WCNC.

[6] 黒田 貴之 他, "ICT システムの設計に関する知識の機械学習による獲得", 電子情報通信学会論文誌 B Vol.J104-B No.3 pp.140-151, 2021.

[7] C. Zhang et al., "Deep Learning in Mobile and Wireless Networking: A Survey," in IEEE Communications Surveys & Tutorials, 2019.

[8] J. Wang et al., "Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks," in IEEE Communications Surveys & Tutorials, 2020.

[9] W. W. Lo et al., “E-graphsage: A graph neural network based intrusion detection system,” arXiv preprint arXiv:2103.16329.

[10] Y. Liu et al., “On Dynamic Service Function Chain Reconfiguration in IoT Networks,” IEEE Internet of Things Journal, 2020.

[11] J. Suárez-Varela et al., “Challenging the generalization capabilities of graph neural networks for network modeling,” Proceedings of the ACM SIGCOMM, 2019.

[12] M. Ferriol-Galmés et al., “Applying graph-based deep learning to realistic network scenarios,” arXiv preprint arXiv:2010.06686.

[13] Fabien Geyer, “Performance Evaluation of Network Topologies using Graph-Based Deep Learning,” Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools.

[14] A. Rkhami et al., "Learn to improve: A novel deep reinforcement learning approach for beyond 5G network slicing," 2021 IEEE 18th Annual Consumer Communications & Networking Conference.

[15] FACEBOOK engineering, https://engineering.fb.com/

[16] R. Sutton et al., Introduction to reinforcement learning, volume 2. MIT press Cambridge, 1998.

[17] T. Kipf et al., “Semi-Supervised Classification with Graph Convolutional Networks”, In Proceedings of the 5th International Conference on Learning Representations, ICLR ’17, 2017.

[18] Peter W. Battaglia et al., “Relational inductive biases, deep learning, and graph networks”, CoRR, abs/1806.01261, 2018.

[19] W. Hamilton et al., “Inductive Representation Learning on Large Graphs”, NIPS 2017.

[20] K. Xu et al., “How Powerful Are Graph Neural Networks?”, ICLR, 2019.

[21] K. Oono et al., “Graph Neural Networks Exponentially Lose Expressive Power for Node Classification”, ICLR, 2020.

[22] U. Alon et al., “On the Bottleneck of Graph Neural Networks and its Practical Implications”, ICLR ’21.

[23] M. Weber et al., “Detecting the Coarse Geometry of Networks”, NIPS 2018.

[24] C. Cai et al., “Graph Induced Complex on Point Data”, In Proceedings of the 29th annual symposium on Computational geometry, 2013.