DEIM Forum 2023 3a-5-5

複数データセットを活用した連続時間グラフリンク予測手法の分析および予測精度の改善

菊地　良将† 　前川　政司† 　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
E-mail: †{kikuchi.ryosuke,maekawa.seiji,sasaki,onizuka}@ist.osaka-u.ac.jp

あらまし　連続時間グラフにおけるリンク予測は，推薦システムや異常検知に有用である．しかし，学習データ不足により学習データに対する過剰適合が起きることが課題であった．そこで，TLIM は複数データセットを用いた学習により学習データ不足を解決し，最高精度を達成した．TLIM では各データセットで共通の方法でグラフの構造・時間情報を捉えたリンク履歴シーケンスを作成し，複数データセットを用いた学習を可能にしている．しかし，複数データセット使用について分析不足など未検証な部分が残されている．そこで，本研究では最先端手法 TLIM を分析して得られた課題の解決により予測精度改善を目指した．複数データセットについての知見と，Fine-Tuning を行っても精度が改善しない理由が明らかになった．また，Positional Encoding によりシーケンス長の増加に伴う予測精度を改善した．

キーワード　リンク予測，時系列グラフ，グラフマイニング

1　はじめに

グラフを用いることでソーシャルネットワーク [1,2] や学術論文の引用関係 [3] などの実世界の様々な事柄を表現することができる．多くのグラフは潜在的に時系列情報を持つため，時間情報を活用する時系列グラフの分析に注目が集まっている [4–13]．特に，時系列グラフにおいてリンクの発生予測を行うことで，実世界で様々な応用が可能になる．例えばソーシャルネットワークを分析することで，得られたユーザ同士の交流傾向をもとに友人推薦や将来のリツイート予測が可能になる．

近年，深層学習を用いることにより時系列グラフの 1 種である連続時間グラフにおいてリンク予測の精度向上を実現する手法が提案されている [7–13]．ここで連続時間グラフとは作用し合う要素がノード，作用自体がリンクとして表現され，リンクの発生がソースノード (始点)，ターゲットノード (終点)，リンク発生時刻のタイムスタンプで表現されるものである．手法は主に 2 種類に大別され，node embedding を用いる手法と walk sampling を用いる手法がある [14]．node embedding は各ノードの周辺の構造情報やリンクが発生した時刻情報をもとに作成され，リンクの発生に伴い動的に更新される．算出された node embedding をもとにグラフの構造・時間的情報を捉えることによりリンク予測を行う．一方で，walks を用いる手法では，node を匿名化した複数の walk を集約することで，三角閉包などといったグラフ変化に頻出するモチーフを捉える．

しかし，既存手法では単一のデータセットを前提に設計されているため，過適合しやすいという問題があった．そこで，大規模深層学習モデル Transformer をベースとした複数データセットを用いて学習可能な連続時間グラフのためのリンク予測手法である TLIM が提案された [15]．複数データセットを使用した学習により，学習データが不足する問題を解消することで，学習データに対する過剰適合を解消している．TLIM では，グラフの構造的・時間的情報を捉えるためにリンク履歴シーケンスを作成する．これは，ノードの周辺構造や時間情報をもとに動的に変化する node embedding を連結したものである．また，複数のデータセットでモデルを学習できるように，各データセットで共通したシーケンス作成方法をとっているため，複数データセットを用いた学習を可能にした．グラフの構造的・時間的情報を捉えたリンク履歴シーケンスを大規模深層学習モデルである Transformer で学習することにより，Hits@10 において既存手法に比べて最高精度を達成した．

しかし，TLIM には未検証な箇所が残されている．1 つ目の未検証な箇所としては複数データセットの使用について詳細な分析がなされていないことが挙げられる．具体的には，データセット数による精度の比較や，データセットの組み合わせによる精度変化が検証されていないことである．2 つ目の未検証な箇所としては Fine-Tuning を行っても精度が上がらない理由についてである．これらが未検証であるために，TLIM が高精度の予測を行えた根拠が不明瞭である．また，未検証な箇所の他に課題が残されており，主要なものにシーケンス長の増加 (モデルにより過去の情報を与える) に伴い予測精度が低下してしまうことが挙げられる．

本研究では TLIM の未検証な箇所を解明し，更に課題の解決をすることでさらなる予測精度の向上を目指す．未検証部分を明らかにするために，データセット数および組合せについて実証的に分析を行う．分析から得られた知見は以下の 3 つである．(1) 学習に使用するデータセットの個数を増やすことで予測精度の改善が見られる．(2) データセットには組み合わせの良し悪しがあり，それにより予測精度が変化する．(3) TLIM は最後に入力したデータに対して過剰適合が起きやす

いため Fine-Tuning を行っても精度が向上しなかった．また，TLIM の長いシーケンスを活用できない課題を解決するために，Positional Encoding によって効果的なシーケンスを導入した．

2 事前準備

2.1 連続時間グラフと問題定義

時刻 t における連続時間グラフは $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}^t)$ と表現される．ここで $\mathcal{V}$ はグラフ上の全ノードの集合であり，$\mathcal{E}^t$ は時刻 t まで発生した全リンクの集合である．また，リンクは $e_i = (v^{src}, v^{tgt}, t_i) \in \mathcal{E}^t$ と表される．ただし，$v^{src}, v^{tgt} \in \mathcal{V}$ である．

本研究では連続時間グラフ $\mathcal{G}(t)$，予測時刻である $T \geqq t$ と予測対象のソース・ターゲットノードペア (v^{src}, v^{tgt}) を入力として，時刻 T における予測対象ノードペア間にリンクが発生する確率を算出する．

2.2 連続時間グラフにおけるリンク予測

近年連続時間グラフを対象としたリンク予測の手法が数多く提案されている [7–13, 16]．静的グラフを対象とするグラフニューラルネットワーク (GNN) を，動的グラフである連続時間グラフに拡張した手法も多い．主に構造情報をもとに学習する静的グラフ上での手法とは異なり，連続時間グラフを対象とする場合はその時間的情報も共に学習する．代表的なアプローチは構造的/時間的情報を捉えた表現である node embedding を用いるものである．

最新の連続時間グラフを活用したリンク予測手法 [9,10,17] ではノードの構造的情報とリンクの発生時刻の 2 つの側面から学習することで，構造的/時間的な傾向を捉えた node embedding を生成する．ノード分類やリンク予測といったタスクに生成した node embedding を使用する．

また，別の手法では walk sampling をベースとした手法が提案されている [12, 13]．それらは三角閉包 (triadic closure) などのグラフが時間変化する際の傾向を捉えることで高精度なリンク予測を可能にしている．具体的にはノードの ID 情報を匿名化する random walks である Anonymous Walks [18] を時系列グラフに拡張することで構造/時間的情報を捉えた高精度なリンク予測を実現している．しかし，既存の手法は単一のデータセット前提に設計されており，学習データ不足のために，学習データに対して過剰適合しやすいという問題がある．

2.3 TLIM

複数データセットを用いた学習により学習データ不足を解決する TLIM [15] が提案された．TLIM の特徴は以下の二つである． 1) リンク履歴をもとに node embedding を連結したリンク履歴シーケンスを用いることで，時系列グラフの特徴を Transformer で捉えることを可能にする．2) リンク履歴シーケンスに匿名化を施すことで，複数データセットを用いた学習が可能になる

このような特徴により，ほとんどのデータセットにおいて TLIM は最先端の予測精度を達成した．しかし，TLIM には問題が残されている．複数データセットの使用について未検証な箇所が多いことや，Transformer への入力シーケンス長の増加に伴い予測精度が低下するという課題が解決されていないことである．

そこで，本研究では最先端手法である TLIM の分析と課題の解決をすることで，連続時間グラフにおけるリンク予測の精度改善を目指す．

なお，本節は文献 [15] を参考に作成している．

2.4 事前知識

2.4.1 TDGNN

TDGNN [8] は node embedding を用いた連続時間グラフにおけるリンク予測手法である．TLIM ではリンク履歴シーケンスの作成において，各時刻の node embedding の算出に使用している．任意の時刻の node embedding を算出することができ，それを用いてリンク予測を行うことができる．TDGNN は，入力の時刻 t のノード v の embedding を得る場合，時刻 t 以前の連続時間グラフからノード v の近傍ノードの属性を伝播することでノード v の embedding を取得する．リンクが発生した時刻情報をによる重みづけを用いて近傍ノードから embedding を集約する．これにより，より直近にリンクが発生したノードに embedding に類似する．

2.4.2 Transformer

ここで，TLIM のベースモデルとして利用されている Transformer について説明する．Transformer [19] は自然言語処理や画像認識などの機械学習分野で用いられている深層学習モデルであり，畳み込みニューラルネットワーク (CNN) や RNN に代わる新しい技術として注目されている．Transformer は自然言語処理の分野では様々なタスクにおいて高い性能を示しており，近年では BERT [20] や GPT-3 [21] などの言語モデルにおける基盤技術としても利用されている．Transformer は入力シーケンスから潜在表現を獲得するエンコーダ層と目的に応じたベクトルを出力するデコーダ層の 2 種類の層からなる．どちらの層も Multi-head attentnion 層を含んでおり，そこで入力シーケンスの各要素間での関連性を表す self-attention を計算する．各要素間での関連性を利用することで長期的な依存関係を捉えることが可能になっている．時系列順に処理しなければならない RNN とは異なり，時系列に関係なく並列して計算可能であるために高速な処理が可能となっている．TLIM では構造的/時間的情報を捉えた潜在表現を獲得するために，リンク履歴シーケンスを Transforomers のエンコーダ層に入力している．潜在表現から入力シーケンスが一意に定まらない匿名化が施されており，複数データセットを用いた学習を可能にしている．

3 分析対象手法 TLIM

3.1 TLIM のアプローチ

TLIM は複数データセットから得られた構造的/時間的情報を Transformer により学習することで，連続時間グラフにおけ

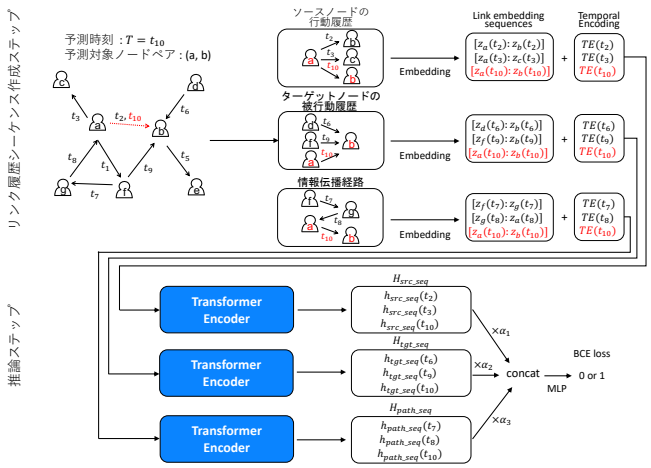


図 1: TLIM の概要図．図は文献 [15] より引用．リンク履歴シーケンス作成ステップでは，予測対象ノードペアごとにソースノードの行動履歴シーケンスとターゲットノードの被行動履歴シーケンス，直近の情報伝播経路シーケンスの 3 つのをリンク履歴シーケンスを作成する．推論ステップでは，作成した 3 つのシーケンスをそれぞれ異なる Transformer encoder に入力し，ソースノードの行動傾向とターゲットノードの被行動傾向，情報の伝播傾向を捉える．

るリンク予測の精度向上を実現する．構造/時間情報を捉えるために，ノードの直近のリンク発生対象を用いてリンク履歴シーケンスを作成し，Transformer を用いて推論を行う．TLIM の全体図を図 1 に示す．TLIM は主に 2 つのステップからなる．(1) 各ノードの行動履歴から匿名化された構造/時間的情報を捉えたリンク履歴シーケンスを作成するステップである．(2) リンク履歴シーケンスを用いて Transformer の学習・推論を行うステップである．

(1) リンク履歴シーケンスは 3 種類存在する．ソースノードの行動履歴，ターゲットノードの被行動履歴，ターゲットノードへの情報伝搬経路を用いたものである．これら 3 種類のリンク履歴シーケンスを用いることによってソース，ターゲットノードそれぞれの構造的情報を捉えることを可能にする．得られた 3 種のリンク履歴シーケンスそれぞれについて，リンクの発生時間間隔に基づいた temporal encoding を行うことにより時間的情報を捉えることを可能にする．

(2) 学習・推論ステップでは予測対象ノードペア間にリンクが発生する確率を用いて学習・推論する．具体的には Transformer と Multilayer perceptron (MLP) を用いる．TLIM の 1 つ目のステップで作成された 3 種類のリンク履歴シーケンスをそれぞれ異なる Transformer に入力する．それによって得られた潜在表現にそれぞれのリンク履歴シーケンスの重要度を表す学習可能パラメータ α を乗算し，重みをつけた特徴量を作成する．得られた特徴量から MLP を用いてリンクの発生確率を算出することによりリンクの発生予測を実現している．

3.2 TLIM アルゴリズム

3.2.1 リンク履歴シーケンス作成ステップ

ここでは上述したリンク履歴シーケンスの作成手順についてソースノードの行動履歴シーケンスを例として取り上げ説明する [1]．シーケンス作成手順は a) リンク発生履歴の取得と b) link embedding の作成， c) Temporal encoding の 3 つの手順で構成される．

a) リンク発生履歴の取得

予測ノードペアのソースノードの行動傾向を把握するために，ソースノードの行動履歴シーケンスを作成する．TLIM ではソースノード v^{src} の行動履歴 S^{src} を，$\mathcal{G}(t)$ における v^{src} から発生した直近 l_{src} 件のリンクとして定義する．図 1 の例は， $l_{src} = 3$ のものであり，直近 3 件の履歴を取得していることを示している．

具体的なアルゴリズム 1 に示す．初めに，予測対象ターゲットノードとの予測時刻のリンクをソースノードの行動履歴 S^{src} に追加する (1 行目)．続いて，発生したリンクの集合 $\mathcal{E}^t$ を時系列順で新しいものから順番に遡り，ソースノードの行動履歴 S^{src} に含まれるリンク数が l_{src} に達するか， $\mathcal{E}^t$ を遡りきるまでソースノードが v^{src} であるリンクを S^{src} に追加する (2-9 行目)．最後に S^{src} 内のリンクを発生時刻で昇順にソートする (10 行目)．

Algorithm 1 Extracting action history of source node

Require: prediction time T, node pair (v^{src}, v^{tgt}), CTDN $\mathcal{G}(t)$, sequence length l_{src},
Ensure: node action sequence S_{src}

1: $S_{src} \leftarrow \{(v^{src}, v^{tgt}, T)\}$
2: **for** $i = |\mathcal{E}^t| - 1$ to 0 **do**
3: **if** $v_i^{src} == v^{src}$ **then**
4: add e_i to S_{src}
5: **end if**
6: **if** $|S_{src}| == l_{src}$ **then**
7: break
8: **end if**
9: **end for**
10: Sort S_{src} by link appearance time in ascending order
11: **return** S_{src}

b) link embedding シーケンスの作成

取得したリンク発生履歴をモデルに入力する形式にするため，履歴を構成する各リンクを link embedding に変換する．ここで，link embedding とは，既存手法を用いて node embedding を事前学習しておき，リンクの発生時刻における両端のノードペアの node embedding を連結したものである．

c) Temporal encoding

ノードの行動時間間隔を捉えるために，得られた link embedding シーケンスに Temporal encoding を施す．現実世界において，特定の周期でリンク発生が起きることがある．具体的には，会社の従業員が特定の曜日に特定の上司に連絡するといったものである．このような時間間隔を踏まえてリンク予測

1：ターゲットノードの被行動履歴，ターゲットノードへの情報伝搬経路シーケンスの作成についても同様である．

を行うことで，精度を高めることができると考えられる．そのため，ノードの行動間隔を捉えるために TLIM では Temporal encoding を施す (詳細は文献 [22] を参照のこと).

3.2.2 学習・推論ステップ

本ステップでは時刻 t において予測ノードペア間にリンクが発生する確率を計算する．リンク履歴シーケンスを入力し，Transformer のエンコーダ層から得られた潜在表現をもとに，MLP を利用してリンクの発生確率を算出する．初めに Transformer [19] のエンコーダ層にリンク履歴シーケンスを入力する．それにより，リンク履歴シーケンスからノード周辺の構造的/時間的情報を捉えた潜在表現 h_{src_seq}，h_{tgt_seq}，h_{path_seq} を獲得する．その際に，3 種のリンク履歴シーケンスの処理にはそれぞれ異なる Transformer を使用する．得られた 3 つの潜在表現の各要素に，それぞれのリンク履歴シーケンスの重要さを表すパラメータ α_{src}，α_{tgt}，α_{path} を乗算したものを連結する．それを MLP に入力することでリンクの発生確率を算出する．モデルの学習には実際に学習データに存在する positive link とそれと同数のターゲットノードをデータセットに存在する全ノード中からランダムに選ばれたノードに置き換えた negative link を対象に Binary Cross Entropy (BCE) を損失関数としてモデルを学習する．

4 実　　験

本章では本研究で行った実験について述べる．主な目的は TLIM の未検証な部分である複数データセットの使用について分析を深めることと，Fine-Tuning を行っても精度が向上しない理由について分析を行うことである．実験の目的は以下の問いに答えることである．

Q1: データセット数を増やすことで精度を改善できるか

Q2: 複数データセットの選び方は精度に影響するか

Q3: Fine-Tuning を行っても精度が向上しないのはなぜか

4.1 実験データセット

ここで実験に使用するデータセットの説明とその詳細な数値を以下に示す．ia-enron-employees [23] と ia-radoslow-email [24] は E-mail の送受信を表した E-mail ネットワークであり，ia-contact [23] と ia-contacts-hypertext09 [23] は人の接触を表す接触ネットワークである．rt-pol [25] は Twitter でのリツイート関係を表したネットワークである．fb-forum [23] はオンラインコミュニティ上のユーザのアクションネットワークである．soc-sign-bitcoinalpha [23] は bitcoin 取引プラットフォームにおける信頼ネットワークである．soc-wiki-elec [23] は Wikipedia 上の管理者選挙の投票ネットワークである．それぞれのデータセットの統計情報を表 1 に示す.

dataset	# nodes	# links	Span (days)
ia-enron-employees	151	50.5K	1137.55
ia-radoslow-email	167	89.2K	271.19
ia-contact	274	28.2K	3.97
ia-contacts_hypertext09	113	20.8K	2.46
rt-pol	4036	60.2K	48.78
fb-forum	899	33.7K	164.49
soc-bitcoin-alpha	3.7K	24.1K	1901.00
soc-wiki-elec	7.1K	107K	1378.34

表 1: データセットの統計情報．Span はそれぞれデータセットの観測期間（日数）を示す.

4.2 実 験 詳 細

4.2.1 実 験 設 定

本実験では，将来の時刻 T においてターゲットノードペア間にリンクが発生する確率を予測し，予測精度を評価する．具体的には，テストデータの各リンク発生履歴 $(v_i^{src}, v_i^{tgt}, t_i)$ ごとに時刻 t_i においてノード v_i^{src} と全ノードとのリンク発生確率を予測する．その後，特に断りがない場合 Hits@10， Mean Rank (MR) を用いて予測精度を評価する．Hits@10 はテストデータ内のすべてのリンクの中で，正解のターゲットノードとの確率が全ノード中上位 10 件に入っているリンクの割合である．どれほど正確に正解ターゲットノードを当てることができるか示す指標であり，数値が高いほど手法の予測精度が高い．MR はテストデータ内のリンクについて予測する際に，正解ターゲットノードが平均で何位に位置するか表すものである．こちらは数値が引くいほど予測精度が高い．実験において，すべてのデータセットについて，最も過去のデータから全体の 7 割を学習に，残りの 3 割をテスト (予測精度の評価) に用いた．負例のターゲットノードは，データセットに存在する全ノードからランダムに 1 つを選択している．また，すべての学習に使用するデータセットでリンク履歴シーケンスを作成した後に，シャッフルして Transformer に入力する．

4.3 実 験 結 果

データセット数を増やすことで精度を改善できるか (Q1)

ここでは，複数データセットの使用について分析をするために，学習に使用するデータセットの個数を増やしていき，予測精度の変化を分析する．学習に使用するデータセットは 4.1 節において示した順番で上から追加していく．この順番は TLIM [15] の論文内で示された順に上から 5 つのデータセットを並べ，新たに追加した 3 つのデータセットを辞書順に並べて追加している．ただし，テストに使用したテスト用のデータセットに対応する学習データセットは，学習にも使用する．例えば ia-radoslow-email をテストとした場合は， ia-radoslow-email → hyper09 → employees → … と 1 つずつ学習用データセットに追加していく．以下の表 2 に結果を示す．また，学習に使用したデータセット数と Hits@10 の関係を表すグラフを図 2 に示す．図 2 より，すべてのデータセットで，単一データセットで学習を行った場合に比べて，すべてのデータセットで学習を行った場合のほうが予測精度が高くなっている．これは，複数データセットを用いることにより，データセットを横断してリンクの発生傾向を学習できているためと考えられる．このことから，今後の方針としてさらにデータセットを追加すること

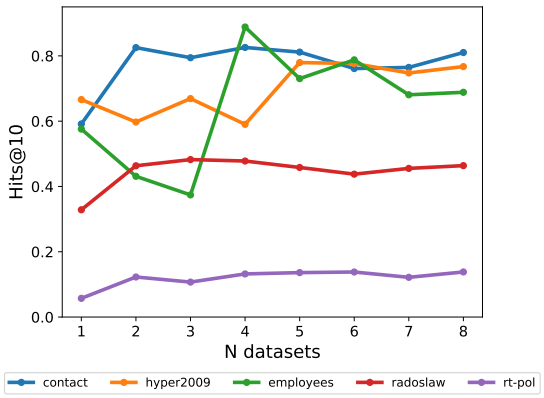


図 2: 学習に使用するデータセット数による予測精度．5 つのデータセットに対して精度評価を行っている．横軸が学習に使用するデータセット，縦軸が Hits@10 である．

で予測精度を向上することが挙げられる．また，データセット数の増加に伴い，単調に予測精度が向上しているわけではない．これは，用いるデータセットの組み合わせによって予測精度が変化することを示している．そのため，実験 2 においてデータセット間の組み合わせの良し悪しを確かめる実験を行う．

複数データセットの選び方は精度に影響するか (Q2)

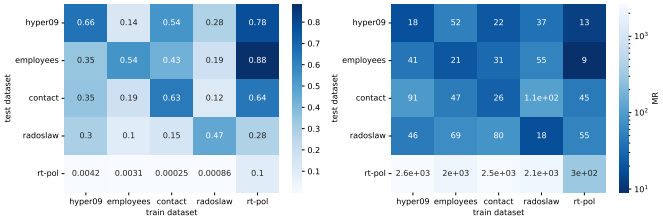
実験 1 の結果から用いるデータセットの組み合わせがモデルの予測精度に影響を与えることが示唆された．そこで，実際にデータセット間にはどのような関連性があるのか分析する．具体的には，学習とテストにそれぞれ単一のデータセットを使用してモデルの学習と予測精度の評価を行った．2 種類の実験結果から分析を行う．1 つ目は本研究で用いる評価指標の Hits@10，MR による結果であり，2 つ目はモデルが実際に学習できているか確認するために Area Under the ROC Curve (AUC)，Accuracy (ACC) によって評価した結果である．

a） 評価指標に Hits@10， MR を用いた結果

実験結果を図 3 に示す．各行が同じデータセットをテスト，各列が同じデータセットを学習に使用していることを示す．一般には，学習とテストで同一のデータセットを用いる場合に精度が最大になると考えられる．なぜなら，異なるデータセット間よりも同一データセット間のほうが傾向が類似しているため，モデルがテストデータの予測に活用できる傾向を学習しやすいからである．しかし，実験結果を見ると，学習とテストのデータセットが一致しない場合の予測精度が最大のデータセットが存在している (例えば Hits@10 においては contact， hyper09, employees の 3 つをテストに使用した場合) ．これは，他のデータセットからもテストデータでのリンク予測に有意な傾向を捉えられているためと考えられる．

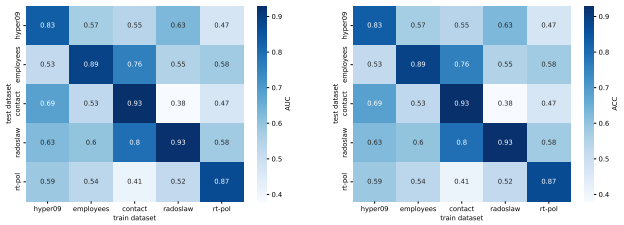
b） 評価指標に AUC， ACC を用いた結果

Hits@10, MR を用いた場合と同じ実験条件下で AUC， ACC を用いて予測精度の評価を行った．実験結果を図 4 に示す．図の 1 行目には学習に用いたデータセット，1 列目にはテストに用いたデータセットである．AUC， ACC のどちらでも，学習とテストにおいて同じデータセットが同じ場合が値が最大に



(a) Hits@10 を用いた結果　　**(b)** MR を用いた結果

図 3: 両図ともに各行が同じデータセットをテスト，各列が同じデータセットを学習に使用していることを示す．bitcoin， wiki-elec の実験結果は今後実験予定．



(a) AUC を用いた結果　　**(b)** ACC を用いた結果

図 4: 両図ともに各行が同じデータセットをテスト，各列が同じデータセットを学習に使用していることを示す．bitcoin， wiki-elec の実験結果は今後実験予定．

なっていることがわかる．このことから，テストに用いるデータセットの学習用のデータを使用して学習することが最も高い予測精度の実現に寄与していると考えられる．

また，多くのデータセットにおいて学習に異なるデータセットを用いた場合は AUC， ACC ともに 0.5 付近の値を取っている．これは AUC， ACC の意味から考えるとほとんど何も学習することができていないことを意味する．そのため，複数のデータセットを用いて学習を行ってもモデルは何も学習できていない可能性が示唆される．しかし，単一のデータセットを用いて TLIM を学習させた場合よりも，複数のデータセットを用いて TLIM を学習させたほうが予測精度が高い．これには 2 つの原因が考えられる．1 つ目は実験 2 で評価指標に Hits@10, MR を用いた結果において，試行回数が少ないために誤差による影響が大きいことである．これは，十分に回数をこなすことでより正確な値を算出するできる可能性がある．2 つ目は単一のデータセット毎では捉えられている傾向があまりないが，複数のデータセットから学習することで予測精度の向上を実現していることである．これらの原因についての追加の実験・分析は今後の課題とする．

Fine-Tuning を行っても精度が向上しないのはなぜか (Q3)

TLIM の提案論文 [15] ではモデルの Fine-Tuning を行っても予測精度が改善しないという結果が報告されている．これは，TLIM が最後に入力するデータセットに過剰適合していることが原因と考えられる．そこで，モデルへのデータセットの入力順がどのように精度に影響するか分析する．TLIM ではすべての学習に使用するデータセットでリンク履歴シーケン

datasets	contact		hyper09		employees		radoslaw		rt-pol	
metrics	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10
1	28.82	0.59	15.07	0.67	19.59	0.58	22.94	0.33	491.25	0.06
2	18.64	**0.83**	18.10	0.60	27.26	0.43	17.70	0.46	267.09	0.12
3	16.40	0.79	14.47	0.67	22.58	0.37	**16.74**	**0.48**	328.54	0.11
4	**14.98**	**0.83**	16.79	0.59	**6.34**	**0.89**	18.02	**0.48**	279.04	0.13
5	17.54	0.81	9.56	**0.78**	13.97	0.73	17.75	0.46	266.91	**0.14**
6	16.77	0.76	**8.84**	**0.78**	11.58	0.79	18.59	0.44	244.48	**0.14**
7	16.72	0.76	10.83	0.75	20.99	0.68	18.68	0.46	292.71	0.12
8	15.83	0.81	11.27	0.77	20.68	0.69	19.42	0.46	**239.75**	**0.14**

表 2: 学習に使用したデータセット数ごとの Hits@10． 最左列が学習に使用したデータセット数あり，最上段が精度評価を行ったデータセットである．各データセットについて最高精度の場合を太字で表示している．

スを作成した後に，データセット関係なくシャッフルした後に Transformer に入力している．そこで，テストに使用するデータセットの学習用データから作成されたシーケンスを最後に Transformer に入力した場合の予測精度を分析する．

Hits@10 を用いて評価した結果を表 3 に示す．表 3 においてすべてのデータセットにおいてシャッフルしてからモデルに入力した場合 (shuffle) が精度が高い．これは，シーケンスをシャッフルして入力するため，特定のデータセットに過適合することなく汎化性能を獲得したため，テストデータでも高い予測を行うことができたと考えられる．一方，シャッフルせずに入力すると，モデルは最後に入力されたデータに過適合を起こしたため，テストデータにおいて予測がうまく行えなかったと考えられる．以上より，TLIM で Fine-Tuning を行った際にも，モデルが入力データに過適合を起こるために，予測精度が改善しないと考えられる．そのため，TLIM では Fine-Tuning による予測精度の改善を行うには，TLIM の過適合を防ぐことが求められる．

5 TLIM の課題とその解決

5.1 TLIM の抱える課題

TLIM は高い予測精度を実現しているが，課題が残されている．主要な課題は図 5 中の点線で表される通り，入力シーケンス長の増加に伴い，予測精度が低下していることが挙げられる．

5.2 課題の解決

本節では，入力シーケンスに Positional Encoding (PE) を施すことで，TLIM の主要な課題であるシーケンス長の増加に伴う予測精度低下が解決されたことを示す．

TLIM では Transformer が入力シーケンスの各要素の順番を利用することができていない．そのため，シーケンス中の予測に有用な要素に重きをおいて扱うことが難しくなり，予測精度が低下してしまうと考えられる．具体的には，直近でリンクが発生したノードの情報がリンク予測に対し有効だとしても，Transformer はどこからどこまでが各ノードに由来するか区別できないために，シーケンス中のどこが直近でリンクが発生したノードに由来するかわからないということが起きる．

そこで，Transformer の原論文 [19] の通り入力シーケンスに Positional Encoding (PE) を施すことにより，Transformer が入力シーケンスの要素順を利用することを可能にする．d_{model} 次元のリンク履歴シーケンスについての PE は以下の式で表され，シーケンスの各要素に加算することで PE を施している．

$$PE(pos, 2i) = \sin(pos/10000^{2i/d_{model}}), \quad (1)$$

$$PE(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}}), \quad (2)$$

pos はシーケンス内での各要素の位置を表し，$2i$， $2i+1$ はシーケンス中の何番目の次元かを表す．本研究では 2 つの node embedding を連結させてリンク履歴シーケンスを作成しているので，$pos = 1, 2$ となる．

ここで，PE の有効性を確認するために予測精度の確認を行う．シーケンス長を変化させた際の予測精度を表 4 に示す．また，シーケンス長の変化させた際の Hits@10 を図 5 に示す．図 5 より PE を施した場合はシーケンス長の増加に伴う予測精度の低下が防げている．これは，PE により Transformer が入力シーケンス各要素の順番を活用することができたために，シーケンス長が増加してもリンク予測に有効な要素に重きを置くことができ，予測精度の低下を防ぐことができたと考えられる．また，contact のシーケンス長が 2 の場合を除いたすべての場合において 既存の TLIM を上回る予測精度を達成していることがわかる．

以上より，PE により TLIM の主要な課題であるシーケンス長の増加に伴う予測精度の低下を解決し，予測精度を改善することができた．

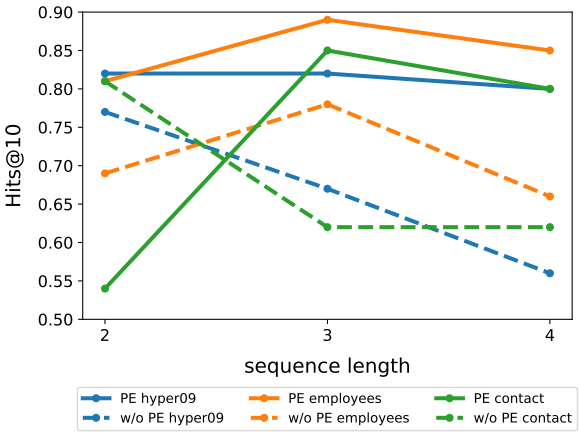


図 5: シーケンス長を変化させた場合の Hits@10．実線で表されるのが PE ありの場合，点線で表されるのが PE なしの場合．PE によりシーケンス長の増加に伴う予測精度の低下を防いでいることがわかる．残りのデータセットについては実験中．

	contact		hyper09		employees		radoslaw		rt-pol	
metrics	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10
w/o shuffle	29.15	0.63	24.04	0.44	28.01	0.59	32.28	0.31	315.38	0.095
shuffle	**15.83**	**0.81**	**11.27**	**0.77**	**20.68**	**0.69**	**19.42**	**0.46**	**239.75**	**0.14**

表 3: モデルに入力するシーケンス順による Hits@10．w/o shuffle がテストに用いるデータセットの学習データを最後にモデルに入力した場合，shuffle は元々の TLIM である．各データセットにおいて予測精度が高いほうを太字表示している．

	contact		hyper09		employees	
metrics	MR	Hits@10	MR	Hits@10	MR	Hits@10
w/o PE_2	**15.83**	**0.81**	11.27	0.77	20.68	0.69
PE_2	25.64	0.54	**8.76**	**0.82**	**8.88**	**0.81**
w/o PE_3	22.29	0.62	14.34	0.67	8.58	0.78
PE_3	**17.83**	**0.85**	**10.50**	**0.82**	**5.66**	**0.89**
w/o PE_4	22.15	0.62	18.75	0.56	14.56	0.66
PE_4	**16.47**	**0.80**	**11.44**	**0.80**	**7.37**	**0.85**

表 4: PE の有無とシーケンス長ごとの予測精度評価．w/o PE_n は PE 無しでシーケンス長が n の場合であり，PE_n は PE 有りでシーケンス長が n の場合を表す．各シーケンス長において予測精度が高いほうを太字で表示．残りのデータセットについては実験中．

6 結　　論

最先端手法である TLIM は複数データセットからリンクの発生傾向を学習することで最先端の予測精度を達成したが，複数データセットの使用について未検証な部分が残されていたり，入力シーケンス長の増加に伴い予測精度が低下する課題が残されていた．本稿では，TLIM を分析対象として未検証部分の分析と課題の解決による予測精度の改善を行った．

未検証部分の検証では以下の 3 つの知見が得られた．(1) 学習に使用するデータセットの個数を増やすことで予測精度の改善が見られる．(2) データセットには組み合わせの良し悪しがあり，それにより予測精度が変化する．(3) TLIM は過適合が起きやすいため Fine-Tuning を行っても精度が向上しなかった．そのために，すべてのデータセットを用いてデータセット関係なくシャッフルしてシーケンスを入力するほうが汎用性を獲得したために予測精度が高い．

TLIM の主要な課題であるシーケンス長の増加に伴い予測精度が低下していることに関しては，Transformer が入力シーケンスの要素順を活用できていないことが原因と考えた．そこで，入力シーケンスに Positional Encoding (PE) を施すことにより，シーケンスの各要素に位置情報を付与し Transformer にシーケンスの要素順を捉えさせた．予測精度の確認を行い，シーケンス長が増加しても予測精度が低下しなかった．これは Positional Encoding を行ったことで Transformer が要素順を活用できたために，シーケンス中の予測精度向上に重要な要素を捉えることができたためと考えられる．

6.1 今後の課題

今後の方針としては以下の 4 つが挙げられる．(1) 学習に使用するデータセット数を増やし，さらなる予測精度向上を目指す．(2) 予測精度の向上のためによりよいデータセットの組み合わせを選択するようモデルを拡張すること．(3) モデルの過適合を防ぐことで，さらなる汎化性能の向上を目指す．(4) より連続時間グラフに適した PE を施すことにより，さらなる予測精度の向上を目指す．これらの解決によりさらなる予測精度の向上や，連続時間グラフ分野における Fine-Tuning が可能になり，モデルを 1 から学習する必要がなくなるために，訓練時間の短縮や計算コストの軽減が可能であると考えている．

謝　　辞

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

文　　献

[1] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of NeurIPS*, 2017.

[2] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[3] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 135–144, 2017.

[4] Palash Goyal, Nitin Kamra, Xinran He, and Yan Liu. Dyngem: Deep embedding method for dynamic graphs. *arXiv*, 2018.

[5] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems*, 2020.

[6] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of ICDM*, 2020.

[7] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. *In Proceedings of WWW*, 2018.

[8] Liang Qu, Huaisheng Zhu, Qiqi Duan, and Yuhui Shi. Link prediction via temporal dependent graph neural network. In *Proceedings of WWW*, 2020.

[9] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. Deep coevolutionary network: Embedding user and item features for recommendation. *arXiv*, 2017.

[10] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over

dynamic graphs. In *Proceedings of ICLR*, 2019.

[11] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of KDD*, 2019.

[12] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *International Conference on Learning Representations*, 2021.

[13] Ming Jin, Yuan-Fang Li, and Shirui Pan. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs. In *Advances in Neural Information Processing Systems*, 2022.

[14] 中西宏和, 前川政司, 佐々木勇和, 鬼塚真. 動的グラフを対象としたリンク予測手法の動向調査. In *DEIM*, 2023.

[15] 山口寛人. 複数データセットを用いた汎用的な時系列リンク予測モデル. 大阪大学大学院情報科学研究科修士論文 (未刊行), 2022.

[16] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *Proceeding of the ICML*, 2020.

[17] Xiaofu Chang, Xuqin Liu, Jianfeng Wen, Shuang Li, Yanming Fang, Le Song, and Yuan Qi. Continuous-time dynamic graph learning via neural interaction processes. In *Proceedings of CIKM*, 2020.

[18] Sergey Ivanov and Evgeny Burnaev. Anonymous walk embeddings. In *International conference on machine learning*, pp. 2186–2195. PMLR, 2018.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NeurIPS*, 2017.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.

[21] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[22] 佐々木勇和山口寛人, 鬼塚真. 複数データセットに共通する傾向を捉えた連続時間グラフのリンク予測. 2022.

[23] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of AAAI*, 2015.

[24] Radosław Michalski, Sebastian Palus, and Przemysław Kazienko. Matching organizational structure and social network extracted from email communication. In *Lecture Notes in Business Information Processing*, 2011.

[25] M.D. Conover, J. Ratkiewicz, M. Francisco, B. Gonç alves, A. Flammini, and F. Menczer. Political polarization on twitter. In *Proceedings of ICWSM*, 2011.