

# 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

山口 寛人<sup>†</sup> 前川 政司<sup>†</sup> 佐々木 勇和<sup>†</sup> 鬼塚 真<sup>†</sup>

<sup>†</sup> 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: †{yamaguchi.hiroto,maekawa.seiji,sasaki,onizuka}@ist.osaka-u.ac.jp

**あらまし** 多くのアプリケーションにおいてユーザの行動が連続時間グラフによって表されており、将来の行動を予測するような時系列リンク予測が注目されている。既存手法では、データセット毎にモデルを学習するように設計されている。しかし、一つのデータセットから得られる教師データ（リンク）は少なく、既存手法ではモデルの汎化性能が上がらないという問題がある。したがって、本稿では複数データセットに共通するリンク発生の傾向を捉えた連続時間グラフのリンク予測手法を提案する。提案手法では、複数データセットに共通するリンク発生の構造的傾向と時間的傾向を捉えるためにリンク履歴シーケンスを提案する。リンク履歴シーケンスでは、リンク発生直前のノードの行動とその行動時間間隔を表現する。このリンク履歴シーケンスから Transformer encoder を用いて次のリンク発生を学習することで、ノードの行動傾向とその行動時間間隔の傾向を捉える。実験では、5つのデータセットで既存手法を上回る精度を達成した。

**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

多くのアプリケーションにおける行動が時系列グラフによって表されている。例えば、Twitter 上のユーザーをノード、リツイートをクリックとするリツイートネットワークが存在する。そこで、グラフから将来のリンク発生つまり将来のノードの行動を予測する時系列リンク予測の需要が高まっている。実際に、ツイートの情報拡散を予測する手段としてリツイートグラフを用いて将来のリツイートを予測する研究が行われている [1]。このようなことから、近年では時系列リンク予測手法の研究が盛んに行われている [2–9]。

既存手法では、連続時間グラフを用いた深層学習手法が数多く提案され、精度向上を達成している [5–9]。連続時間グラフは、リンクの発生時刻を連続値で表現することが可能なグラフである。既存手法ではこの連続時間グラフを活用することで、詳細なリンク発生の時系列を考慮したリンク予測を実現している。特に、連続時間グラフを活用した最新のリンク予測手法 [7, 8] では、リンク発生時刻直前の node embedding から点過程アプローチを用いてリンク発生確率を学習することで精度向上を達成している。

しかし、既存手法ではモデルの学習に利用できる教師データ（リンク）が少なく訓練データ内でリンクが未観測のノードペア間で初めて発生するリンク（unobserved link）の予測精度が上がらない。これは、既存手法がデータセット毎にモデルを学習するように設計されていることが原因となっている。一つのデータセットから深層学習モデルを訓練するために十分な教師データを収集することは困難であるため、モデルの汎化性能を十分に向上させることができない。したがって、既存手法では unobserved link を高精度に予測することができない。

自然言語処理の分野では、このように一つのデータセットか

**表 1:** 本稿の実験で使用する 5 つのデータセットにおける unobserved link の予測精度。評価指標には Hits@10 を用いており、値が大きいほど高精度であることを示している。表中の太字は最高精度であることを表す。

methods	DyRep	Proposed
ia-enron-employees	0.08 ± 0.00	<b>0.36 ± 0.08</b>
ia-radoslow-email	0.01 ± 0.00	<b>0.36 ± 0.09</b>
ia-contact	0.00 ± 0.00	<b>0.28 ± 0.06</b>
ia-contacts-hypertext09	0.05 ± 0.00	<b>0.43 ± 0.07</b>
rt-pol	<b>0.10 ± 0.00</b>	0.08 ± 0.01

らでは訓練データが不足するため、傾向が類似する複数のデータセットを用いてモデルを訓練することで汎化性能を向上させる技術が普及している [10]。本研究では、このアイデアに着想を得て、グラフデータにおいても複数データセット間でのノードの行動には共通の傾向があると仮定し、複数データセットから共通のリンク発生傾向を学習することでリンク予測精度の向上を目指す。上記の仮説を検証するために表 1 において、最先端の既存手法（DyRep [8]）と複数データセットで訓練した提案手法のリンク予測精度の比較を示す。この表から、実際に複数データセットでモデルを訓練することで、unobserved link に対する予測精度が大幅に向上しており、モデルの汎化性能が向上していることが確認できる。

本研究では、以下の二つの技術的な問いを解決することにより、複数データセットに共通するリンク発生傾向を捉える時系列リンク予測手法を提案する。(1) どのようにして複数データセットのリンク発生を一つのモデルで学習するか。(2) 複数グラフで共通のリンクの発生傾向を学習するためにどのような特徴を捉えるべきか。我々は上記の技術的な問いに対してそれぞれ以下のアプローチで解決する。(1) 複数データセットのリンク発生を一つのモデルで学習するために、複数データセットで共通の入力データ構造となるリンク履歴シーケンスを提案する。(2) 複数データセットで共通するリンクの発生傾向として、

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、情報の伝播傾向の3つのノードの行動傾向とその行動時間間隔の傾向を捉える。上記二つのアプローチを実現する手段として、提案手法ではリンク発生直前のノードの行動とその行動時間間隔を表すリンク履歴シーケンスを作成し、このシーケンスから将来のリンク発生を学習することで、複数データセットに共通するリンク発生傾向を捉える。以上の提案手法において、実験では時系列リンク予測で広く用いられる5つのデータセットで既存手法を上回る精度を達成した。

## 2 事前知識

### 2.1 問題定義

本研究では、時刻  $t$  までの連続時間グラフ  $G(t)$  とリンク予測時刻  $T \geq t$ 、予測対象のソース・ターゲットノードペアを入力として、時刻  $T$  における予測対象のソースノードからターゲットノードへのリンク発生確率を予測する。時刻  $t$  の連続時間グラフは  $G(t) = (\mathcal{V}, \mathcal{E}^t)$  で表され、 $\mathcal{V}$  はノードセット、 $\mathcal{E}^t$  はタイムスタンプが付与されたリンクセットである。 $e_i = (v_i^{src}, v_i^{tgt}, t_i) \in \mathcal{E}^t$  はグラフ中のタイムスタンプ付きリンクを表し、 $v_i^{src}$  はソースノード、 $v_i^{tgt}$  はターゲットノード、 $t_i$  は  $t_i < t$  を満たすリンク発生時刻である。 $\mathcal{E}^t$  はリンクのタイムスタンプによってソートされている。

### 2.2 Transformer

Transformer [11] は BERT [10] や GPT [12], XLnet [13] などのベースとなっている手法であり、自然言語処理の分野で幅広く利用されている手法である。Transformer は Encoder-Decoder モデルであり、encoder ではベクトルシーケンスからそれらの潜在表現を出力し、decoder では encoder の潜在表現から目的シーケンスを出力する。Encoder および decoder では、シーケンスに対して attention を適応することで各要素間の依存関係の強さを捉えた潜在表現及び目的シーケンスの取得が可能である。ただし、attention のみでは入力シーケンスの順序を考慮できないため、要素の順序を表す情報をシーケンスに付与する必要がある。そこで、一般的にはシーケンス中の各要素の位置情報に基づく positional encoding を入力シーケンスに加算することで各要素の順序関係を表現している。

自然言語処理の分野では、複数のデータセットを用いて Transformer encoder を訓練することで汎用的な文脈理解を実現する BERT [10] が提案され、幅広いタスクに利用されている。BERT では、複数データセット内の文章を word embedding シーケンスとして Transformer encoder に入力することで一般的な単語間の依存関係を捉えることに成功している。そこで、連続時間グラフのリンク予測においては複数データセット内のリンク履歴シーケンスを Transformer encoder に入力することで、複数データセットで共通するリンク発生の依存関係を捉えることが可能になると考えられる。

### 2.3 時系列グラフのリンク予測手法

近年では、連続時間グラフを用いたリンク予測手法が数多く

提案されている [5-9]。連続時間グラフは、リンクの発生時刻を連続値で表現することが可能なグラフである。したがって、既存手法ではこの連続時間グラフを活用することで、詳細なリンク発生の時系列を捉えたリンク予測を実現している。特に、最新の連続時間グラフを活用したリンク予測手法 [7,8] では、リンク発生を構造的傾向と時間的傾向の二つの側面から学習することで精度向上を達成している。具体的には、入力された連続時間グラフから各ノードの構造的な特徴を反映した node embedding を作成することでリンク発生の構造的傾向を学習し、その node embedding からリンク発生確率の算出の際に点過程アプローチを導入することで、リンク発生の時間的傾向を学習している。しかし、既存手法では学習に利用できる教師データ（リンク）が少なく、unobserved link を高精度に予測することができないという問題が存在する。これは、既存手法がデータセット毎にモデルを学習するように設計されていることに起因する。1つのデータセットから深層学習モデルを訓練するために十分な教師データを取得することは困難であるため、既存手法はモデルの汎化性能を十分に向上させることができない。そこで、複数データセットを用いてモデルを学習可能な時系列リンク予測手法を設計することにより、このような訓練データの不足が解消され、時系列リンク予測精度が向上すると考えられる。

## 3 提案手法

既存手法では汎化性能を向上させることが困難であるという問題点を解決するために、複数データセットから共通のリンク発生傾向を捉える時系列リンク予測モデルを提案する。

### 3.1 提案手法のアプローチ

提案手法は複数データセットで共通するリンク発生（ノードの行動）の構造的傾向と時間的傾向を捉えることで、時系列リンク予測の精度向上を実現する。具体的には、リンク発生の構造的傾向としてノードの行動傾向を捉え、リンク発生の時間的傾向としてノードの行動時間間隔の傾向を捉える。手法の全体像を図1に示す。図1では、連続時間グラフ  $G(t_0)$  から時刻  $t_{10}$  にノード a から b にリンクが発生する確率を予測する例を示している。提案手法は、リンク履歴シーケンス作成ステップと推論ステップの二つのステップで構成される。複数データセットのリンク発生を一つのモデルで学習可能にするために、リンク履歴シーケンス作成ステップでは複数データセットで共通の入力データ構造となるリンク履歴シーケンスを作成する。リンク履歴シーケンスでは (1) 直近のノードの行動と (2) その行動時間間隔を表現することで、複数データセットに共通するノードの行動傾向とノードの行動時間間隔の傾向を学習可能にする。

(1) ノードの行動を表現するために、予測対象ノードペア毎に直近  $n$  件のリンク履歴シーケンスを作成する。これにより、推論ステップにおいて直近のノードの行動に起因するリンク発生を捉えることを可能にする。具体的には、ソースノードの行動履歴とターゲットノードの被行動履歴、ターゲットノードへ

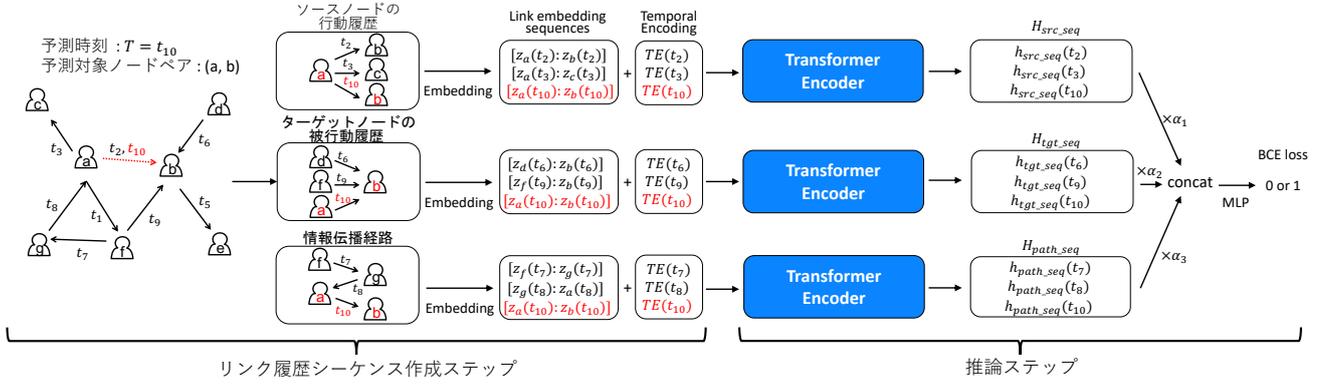


図 1: 提案手法の概要図。リンク履歴シーケンス作成ステップでは、予測対象ノードペアごとにソースノードの行動履歴シーケンスとターゲットノードの被行動履歴シーケンス、直近の情報伝播経路シーケンスの 3 つのリンク履歴シーケンスを作成する。推論ステップでは、作成した 3 つのシーケンスをそれぞれ異なる Transformer encoder に入力し、ソースノードの行動傾向とターゲットノードの被行動傾向、情報の伝播傾向を捉える。

の直近の情報伝播経路の 3 つのリンク履歴シーケンスを作成することで、多様なノードの行動傾向を捉えることを可能にする。最初の 2 つのソースノードの行動履歴やターゲットノードの被行動履歴では各ノードのリツイート傾向や被リツイート傾向のようなノードの行動・被行動傾向を捉えることが可能であり、3 つ目の情報伝播経路ではインフルエンサーから一般ユーザーへの情報伝播のような情報の伝播傾向を捉えることが可能である。

(2) ノードの行動時間間隔を表現するために、上記の 3 つのシーケンスに対してそれぞれのリンク発生時刻に基づく temporal encoding を行う。この temporal encoding では、予測時刻とシーケンス中の各リンク発生時刻との差分に基づいた encoding を行うことで、推論ステップでリンク発生時間間隔傾向を捉えることを可能にする。

次に推論ステップでは、初めに上記で作成した 3 種類のシーケンスをそれぞれ異なる Transformer encoder に入力することで、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た潜在表現を取得する。さらに、上記 3 つの行動傾向の重要度を学習するために、これら 3 つの潜在表現に対してそれぞれの潜在表現の重要度を表す学習パラメータ  $\alpha$  を用いて重み付き結合した特徴量を作成する。この特徴量から MLP を用いて最終的なリンク発生を学習することで、異なるリンク発生傾向の重要度を捉えたリンク予測を実現する。

## 3.2 提案手法アルゴリズム

### 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 3 つのリンク履歴シーケンス（ソースノードの行動履歴シーケンス、ターゲットノードの被行動履歴シーケンス、ターゲットノードへの情報伝播経路シーケンス）の作成手順について説明する。これらのシーケンス作成手順は a) リンク発生履歴の抽出と b) link embedding の作成、c) Temporal encoding の 3 つの手順で構成される。

#### a) リンク発生履歴の抽出

一つ目のソースノードの行動履歴シーケンスを作成するために、ソースノードの行動履歴を抽出する。本稿では、ソース

### 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**  $S_{src}$

### Algorithm 2 Extracting action history of target node

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

**Ensure:** node action sequence  $S_{tgt}$

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

ノードの行動履歴は連続時間グラフ  $\mathcal{G}(t)$  における（予測対象リンクを含む）直近  $l_{src}$  件のソースノードの out-degree リンクと定義する。図 1 の例では、図中の上段において  $l_{src} = 3$  とし、予測対象のソースノード a の直近 3 件の out-degree リンクを抽出している。具体的なソースノードの行動履歴の取得

---

**Algorithm 3** Extracting information propagation path

---

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

**Ensure:** source node action sequence  $S_{path}$

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

---

アルゴリズム Algorithm 1 に示す. 初めに, モデルがリンク予測対象のノードペアを判別可能にするために, 予測対象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). 以降では, リンク発生履歴を時刻  $t$  から過去へと探索し, 各履歴のソースノード  $v_i^{src}$  が予測対象のソースノード  $v^{src}$  と等しい場合, ソースノードの行動履歴  $S_{src}$  に追加する. これを  $S_{src}$  の長さが  $l_{src}$  と等しくなるか全てのリンク発生履歴を探索し終えるまで行う (2–9 行目). 最後に,  $S_{src}$  内のリンクを実際のリンク発生順にするために,  $S_{src}$  の要素をリンク発生時刻で昇順に並べ替えを行う (10 行目).

次に, ターゲットノードの被行動履歴シーケンスを作成するために, ターゲットノードの被行動履歴を抽出する. 本稿では, ソースノードの行動履歴を連続時間グラフ  $\mathcal{G}(t)$  における (予測対象リンクを含む) 直近  $l_{tgt}$  件のターゲットノードの in-degree リンクと定義する. 図 1 の例では, 図中の中段において  $l_{tgt} = 3$  とし, 予測対象のターゲットノード b の直近 3 件の in-degree リンクを抽出している. 具体的なターゲットノードの被行動履歴の取得アルゴリズムを Algorithm 2 に示す. ターゲットノードの被行動履歴の取得アルゴリズムはソースノードの行動履歴抽出アルゴリズムとほぼ同じアルゴリズムである. 差分は, リンク発生履歴を時刻  $t$  から過去へと探索し, 各履歴のターゲットノード  $v_i^{tgt}$  が予測対象のターゲットノード  $v^{tgt}$  と等しい場合, ターゲットノードの被行動履歴  $S_{tgt}$  に追加するという点のみである (3–5 行目).

最後に, 情報伝播経路シーケンスを作成するために, ターゲットノードへの情報伝播経路を抽出する. ターゲットノードへの情報伝播経路は, 連続時間グラフ  $\mathcal{G}(t)$  のターゲットノードから最新の in-degree リンクをウォークした長さ  $l_{path}$  の経路と定義する. 図 1 の例では,  $l_{path} = 3$  とし, 図中の下段において予測対象のターゲットノード b から最新の in-degree リンクをウォークした長さ 3 の経路を抽出している. ターゲットノードへの情報伝播経路の取得アルゴリズム Algorithm 3 に示す. 初めに, モデルがリンク予測対象のノードペアを判別可

能にするために, 予測対象リンクを情報伝播経路  $S_{path}$  に追加する (1 行目). 次に, 時刻  $t$  から過去へとリンク発生履歴を探索し, 最も直近にターゲットノード  $v_i^{tgt}$  が予測対象のソースノード  $v^{src}$  と等しくなっているリンクを情報伝播経路  $S_{path}$  に追加する. 以降では, 追加したリンクのソースノードがターゲットノードになっているリンク発生履歴を  $S_{path}$  に追加する. これを  $S_{path}$  の長さが  $l_{path}$  と等しくなるか全てのリンク発生履歴を探索し終えるまで行う (2–9 行目). 最後に  $S_{path}$  内のリンクを実際のリンク発生順にするために,  $S_{path}$  の要素をリンク発生時刻で昇順に並べ替える (10 行目).

b) link embedding の作成

上記で作成したリンク発生履歴をモデルに入力可能な link embedding のシーケンスに置き換える. link embedding は事前学習した既存の node embedding 手法で得られるリンクの両端のノードペアの embedding を結合したものである. 提案手法では, 既存の node embedding 手法として連続時間グラフの node embedding 手法の中で最先端の手法である TDGNN [6] と node2vec [14] を組み合わせることにより, より正確なノードの構造的特徴とその時系列変化を捉えた node embedding を取得する. TDGNN では任意の時刻の任意のノードの embedding を取得可能なモデルである. TDGNN は入力時刻以前の連続時間グラフを用いて, より直近にリンクが発生した隣接ノードの embedding を重く伝播することで, 直近にリンクが発生したノードの特徴をより強く反映した node embedding を得ることが可能である. この手法を用いることで, 提案手法においてもノード特徴の時系列変化を表現した link embedding を取得可能になる. さらに, TDGNN の初期値として node2vec [14] で得られる node embedding を用いることで, グラフの構造情報をより正確に反映した node embedding を取得している<sup>1</sup>.

c) Temporal encoding

リンク履歴シーケンスにノードの行動時間間隔の情報を埋め込むために, Temporal Encoding を提案する. 自然言語処理で用いられている Positional Encoding [11] に着想を得て, 上記で作成した link embedding に対してリンク予測時刻とシーケンス中のリンク発生時刻の差に基づく Temporal Encoding を加算する. 本稿では, 上記の条件を満たす簡単な encoding 方法として以下の式 (1) を用いる.

$$TE(t_k) = (T - t_k)W \quad (1)$$

ここで,  $t_k$  はリンク履歴シーケンス中の  $k$  番目のリンク発生時刻を表す.  $W \in \mathbb{R}^{1 \times d}$  は学習パラメータであり,  $d$  は link embedding の次元数を表す. ただし,  $W$  は各シーケンスの種類毎に  $W_{src}, W_{tgt}, W_{path}$  に置き換えられる.

---

1: 提案手法では, node2vec のパラメータである random walk bias を近傍ノード間で類似した embedding となると報告されている  $p = 1.0, q = 0.5$  として実装している.

### 3.2.2 推論ステップ

推論ステップでは、リンク履歴シーケンス作成ステップで作成した3種類のシーケンスから Transformer encoder を用いてリンクの発生傾向を学習する。3種類のシーケンスをそれぞれ異なる Transformer encoder に入力することで、各 Transformer encoder はそれぞれのノードの行動傾向を捉えた潜在表現  $H_{src\_seq}$ ,  $H_{tgt\_seq}$ ,  $H_{path\_seq}$  を算出する。これらの潜在表現を重み  $\alpha_i$  ( $i = 1, 2, 3$ ) で重み付け結合した特徴量を MLP に入力し、リンク発生確率を算出する。モデルの学習では、訓練データの positive link と同数の negative sample link を用いて Binary Cross Entropy loss による学習を行う。

## 4 実験

本実験では、以下の七つの質問に答えるような実験を行う。

**Q1:** 既存手法と比較して予測精度が向上しているか？

**Q2:** 複数データセットで学習することで精度が向上するか？

**Q3:** 提案手法で作成する3つのリンク履歴シーケンスがそれぞれ精度に貢献しているか？

**Q4:** 提案手法で導入した Temporal Encoding は時系列リンク予測精度の向上に貢献しているか？

**Q5:** 提案手法で導入した各行動傾向の重要度を学習するパラメータ  $\alpha_i$  は時系列リンク予測精度の向上に貢献しているか？

**Q6:** 作成するリンク履歴シーケンスの長さが精度にどれほど影響を与えるか？

**Q7:** 複数データセットで学習した提案手法を各データセットで finetuning することで予測精度は向上するか？

### 4.1 実験設定

本実験では、ある時刻にあるソースノードとリンクが発生するターゲットノードを予測する。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  $v_i^{src}$  とその他の全ノードとのリンク発生確率を予測する。評価指標には、既存手法の評価で広く用いられている Mean Rank と Hits@10 を用いる [15, 16]。Mean Rank (MR) は各テストデータで予測した全ノードとのリンク発生確率から正解のターゲットノード  $v_i^{tgt}$  のランキングを取得し、全テストデータで平均値を取ったものである。したがって、MR の値は小さいほど良い精度であることを示す。Hits@10 は上記と同様の方法で取得した正解データのランキングが上位 10 位以内に入っているものの割合を示す。したがって、Hits@10 の値は大きいほど良い精度であることを示す。以降の実験結果では 5 回の精度評価を行なった平均値と標準偏差を示す。

### 4.2 データセット

本実験では、以下の5つの連続時間グラフを用いる。ia-enron-employees [17] と ia-radoslow-email [18] は E-mail の送受信を表した E-mail ネットワークであり、ia-contact [17] と ia-contacts-hypertext09 [17] は人の接触を表す接触ネットワークである。rt-pol [19] は Twitter のリツイート関係を表した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 2 に示す。

表 2: データセットの統計情報。Span と Granularity はそれぞれデータセットの観測期間 (日数) とタイムスタンプの取得間隔を示す。

dataset	# nodes	# links	Span (days)	Granularity
ia-enron-employees	151	50.5K	1137.55	day
ia-radoslow-email	167	89.2K	271.19	sec
ia-contact	274	28.2K	3.97	20 sec
ia-contacts-hypertext09	113	20.8K	2.46	20 sec
rt-pol	4036	60.2K	48.78	day

### 4.3 比較手法

実験では、提案手法と連続時間グラフを用いた最先端のリンク予測手法を含む 7 個の既存手法との比較を行う。以下に実験で用いる提案手法と既存手法の概要を示す。node2vec [14]<sup>2</sup> と GraphSAGE [20]<sup>3</sup> は静的グラフの node embedding 手法である。DySAT [4]<sup>4</sup> は離散時間グラフを活用する最新のリンク予測手法である。TDGNN [6]<sup>5</sup> と JODIE [9]<sup>6</sup>, TGN [21]<sup>7</sup> は連続時間グラフの node embedding 手法である。Dyrep [8]<sup>8</sup> は連続時間グラフを用いた時系列リンク予測手法である。Prop\_AllData と Prop\_SingleData はそれぞれ提案手法を使用する全データセットの訓練データで訓練したモデルと提案手法をテストするデータセットの訓練データのみで訓練したモデルを表す。

### 4.4 実験結果

#### 4.4.1 精度評価 (Q1)

提案手法と既存手法との精度比較の結果を表 3 に示す。全データセットを用いて訓練した提案手法 (Prop\_AllData) は、ほとんどの結果において既存手法と同等かそれ以上の精度を達成している。特に ia-contact の Hits@10 においては最新の手法 (TGN) と比較して 2 倍以上の精度向上を達成している。rt-pol においても最新の手法と比較して MR が約 90 向上しており、大幅な精度向上が確認できる。これら二つのデータセットは表 2 に示すように 1 ノードあたりのリンク数が少なく、既存手法ではグラフサイズに対して学習に利用できる教師データ数が少ないため精度が低い結果となっている。

#### 4.4.2 Ablation Study (Q2 - Q5)

a) 複数データセットによるモデル学習の有効性の検証 (Q2) 複数データセットでモデルを訓練することで精度が向上することを確認するために、表 3 の Prop\_AllData と Prop\_SingleData を比較する。この結果から、ほとんどの結果において複数データセットで学習したモデルが高精度となっていることが確認できる。したがって、提案手法は複数データを用いて訓練することで、より高精度なリンク予測を実現できる

2: <https://github.com/eliorc/node2vec>

3: [https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch\\_geometric.nn.models.GraphSAGE](https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch_geometric.nn.models.GraphSAGE)

4: [https://github.com/FeiGSSS/DySAT\\_pytorch](https://github.com/FeiGSSS/DySAT_pytorch)

5: <https://github.com/Leo-Q-316/TDGNN>

6: <https://github.com/srijankr/jodie>

7: <https://github.com/twitter-research/tgn>

8: <https://github.com/uoguelph-mlrg/LDG>

表 3: 精度評価. 太字の結果は最高精度であることを表し, 下線の結果は 2 番目に良い結果であることを示す.

datasets	ia-enron-employees		ia-radoslow-email		ia-contact		ia-contacts-hypertext09		rt-pol	
	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10
node2vec	52.33 ± 5.22	0.14 ± 0.00	17.52 ± 0.86	0.48 ± 0.03	<u>19.46 ± 7.03</u>	0.23 ± 0.27	45.20 ± 3.40	0.07 ± 0.02	418.47 ± 121.39	0.04 ± 0.03
GraphSAGE	65.91 ± 5.29	0.08 ± 0.01	50.61 ± 4.95	0.12 ± 0.04	56.05 ± 8.93	0.25 ± 0.02	49.49 ± 8.00	0.12 ± 0.05	1709.50 ± 398.29	0.00 ± 0.00
DySAT	43.73 ± 1.79	0.21 ± 0.00	63.03 ± 2.57	0.10 ± 0.01	94.11 ± 24.92	0.05 ± 0.04	53.44 ± 7.79	0.16 ± 0.08	1544.30 ± 151.32	0.03 ± 0.01
TDGNN	45.44 ± 0.34	0.14 ± 0.03	32.70 ± 0.02	0.27 ± 0.00	41.35 ± 6.73	0.19 ± 0.00	31.61 ± 0.47	0.25 ± 0.00	1099.72 ± 263.78	0.03 ± 0.00
JODIE	25.32 ± 0.97	<b>0.66 ± 0.00</b>	<u>15.22 ± 0.10</u>	<u>0.68 ± 0.00</u>	49.05 ± 1.91	0.27 ± 0.00	15.17 ± 1.73	<b>0.76 ± 0.01</b>	635.51 ± 61.37	<u>0.18 ± 0.00</u>
TGN	<b>18.00 ± 0.62</b>	0.59 ± 0.01	17.20 ± 0.21	0.60 ± 0.01	23.10 ± 0.56	0.40 ± 0.00	<b>11.18 ± 1.51</b>	0.73 ± 0.05	515.41 ± 14.82	0.16 ± 0.01
DyRep	23.12 ± 0.42	0.56 ± 0.00	17.42 ± 0.84	0.55 ± 0.00	33.79 ± 0.12	0.25 ± 0.00	19.00 ± 1.44	0.59 ± 0.00	767.44 ± 20.03	<b>0.23 ± 0.00</b>
Prop_AllData	<u>18.32 ± 2.18</u>	<u>0.64 ± 0.03</u>	<b>9.73 ± 2.94</b>	<b>0.74 ± 0.11</b>	<b>14.91 ± 2.51</b>	<b>0.88 ± 0.02</b>	<u>14.99 ± 1.43</u>	<u>0.75 ± 0.03</u>	<b>329.00 ± 18.77</b>	0.09 ± 0.02
Prop_SingleData	19.49 ± 1.29	0.55 ± 0.01	24.97 ± 1.61	0.26 ± 0.04	23.06 ± 6.50	<u>0.68 ± 0.23</u>	15.81 ± 0.92	0.72 ± 0.02	<u>341.51 ± 16.86</u>	0.10 ± 0.01

表 4: 各シーケンスを削除した場合の予測精度. Prop w/o src\_seq, Prop w/o tgt\_seq, Prop w/o path\_seq はそれぞれソースノードの行動履歴シーケンスを用いないモデル, ターゲットノードの被行動履歴シーケンスを用いないモデル, 情報伝播経路シーケンスを用いないモデルを表している. 太字の結果は最高精度であることを表し, 下線の結果は 2 番目に良い結果であることを示す.

methods	Prop_AllData		Prop w/o src_seq		Prop w/o tgt_seq		Prop w/o path_seq	
	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10
ia-enron-employees	<b>18.32 ± 2.18</b>	<b>0.64 ± 0.03</b>	20.72 ± 1.70	0.59 ± 0.07	21.42 ± 0.89	0.54 ± 0.01	<u>18.87 ± 1.25</u>	<u>0.62 ± 0.03</u>
ia-radoslow-email	<u>9.73 ± 2.94</u>	<b>0.74 ± 0.11</b>	12.10 ± 4.70	0.66 ± 0.15	19.79 ± 0.22	0.41 ± 0.01	<b>9.19 ± 3.46</b>	<b>0.74 ± 0.15</b>
ia-contact	<u>14.91 ± 2.51</u>	<u>0.88 ± 0.02</u>	15.89 ± 3.04	0.84 ± 0.05	38.40 ± 0.57	0.19 ± 0.01	<b>14.79 ± 1.00</b>	<b>0.89 ± 0.01</b>
ia-contacts-hypertext09	<b>14.99 ± 1.43</b>	<b>0.75 ± 0.03</b>	16.80 ± 1.69	0.73 ± 0.10	24.24 ± 1.92	0.51 ± 0.02	<u>16.04 ± 1.43</u>	<u>0.74 ± 0.02</u>
rt-pol	<b>329.00 ± 18.77</b>	<b>0.09 ± 0.02</b>	<u>349.06 ± 37.63</u>	0.08 ± 0.02	786.24 ± 6.43	<b>0.09 ± 0.00</b>	365.83 ± 85.18	<b>0.09 ± 0.03</b>

表 5: 提案手法における Temporal Encoding の有効性の検証. Prop w/o TE は提案手法で導入している Temporal Encoding を行わなかった場合の予測結果を示している. 太字の結果は最高精度であることを表す.

methods	Prop_AllData		Prop w/o TE	
	MR	Hits@10	MR	Hits@10
ia-enron-employees	<b>18.32</b>	<b>0.64</b>	32.44	0.39
ia-radoslow-email	<b>9.73</b>	<b>0.74</b>	25.05	0.32
ia-contact	<b>14.91</b>	<b>0.88</b>	39.56	0.17
ia-contacts-hypertext09	<b>14.99</b>	<b>0.75</b>	25.10	0.47
rt-pol	<b>329.00</b>	0.09	543.92	<b>0.10</b>

ことを示している.

#### b) 各リンク履歴シーケンスの有効性の検証 (Q3)

次に, 提案手法で導入した 3 つのシーケンス (ソースノードの行動履歴シーケンス, ターゲットノードの被行動履歴シーケンス, ソースノードへの情報伝播経路シーケンス) についてそれぞれの有効性を確認する. 提案手法から各シーケンスを削除した場合の精度を表 4 に示す. この表から, 提案手法の 3 つのシーケンス全てを用いた場合の精度が最も高く, 各シーケンスが精度向上に貢献していることが確認できる. また, シーケンスを削除したモデルの中では Prop w/o tgt\_seq が最も低い精度である. これは, リンク発生が直前のターゲットノードの被リンクと強い依存関係があることを示している.

#### c) Temporal Encoding の有効性の検証 (Q4)

提案手法ではノードの行動時間間隔をリンク履歴シーケンスに埋め込むために, Temporal Encoding を導入している. 本実験では, この Temporal Encoding の有効性を確認する. 表 5 では提案手法の時系列リンク予測精度と提案手法で Temporal Encoding を行わなかった場合の時系列リンク予測精度を比較する. この表から, ほとんどの精度比較において Temporal Encoding を行っている Prop\_AllData の方が高いリンク予測精度を達成していることが確認できる. したがって, Temporal Encoding は時系列リンク予測の精度向上に貢献しており, 将来のノードの行動は過去の行動の時間間隔に依存していること

が確認できる.

表 6: 提案手法における行動傾向の重要度学習パラメータ  $\alpha_i$  の有効性の検証. Prop w/o alpha は提案手法で導入している行動傾向の重要度学習パラメータ  $\alpha_i$  を全て 1 にした場合の予測結果を示している. 太字の結果は最高精度であることを表す.

methods	Prop_AllData		Prop w/o alpha	
	MR	Hits@10	MR	Hits@10
ia-enron-employees	<b>18.32</b>	<b>0.64</b>	26.58	0.52
ia-radoslow-email	<b>9.73</b>	<b>0.74</b>	20.61	0.40
ia-contact	<b>14.91</b>	<b>0.88</b>	38.42	0.19
ia-contacts-hypertext09	<b>14.99</b>	<b>0.75</b>	23.43	0.52
rt-pol	<b>329.00</b>	<b>0.09</b>	934.37	<b>0.09</b>

#### d) 行動傾向の重要度パラメータ $\alpha_i$ の有効性の検証 (Q5)

提案手法では, 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, 情報伝播傾向の 3 つの行動傾向の重要度を学習するパラメータ  $\alpha_i$  を導入している. 本実験では, この  $\alpha_i$  の有効性を確認する. 表 6 では提案手法の時系列リンク予測精度と提案手法の行動傾向の重要度学習パラメータ  $\alpha_i$  を全て 1 とした場合の時系列リンク予測精度を比較している. この表から, 全てのデータセットにおいて行動傾向の重要度学習パラメータ  $\alpha_i$  を学習している Prop\_AllData が Prop w/o alpha 以上のリンク予測精度を達成している. この結果から, 行動傾向の重要度を学習することの有効性が確認できる.

#### 4.4.3 Parameter Sensitivity (Q6)

本実験では, 提案手法で作成するリンク履歴シーケンスの長さが予測精度に与える影響を確認する. シーケンス長を 1 ~ 5 の整数値に設定した場合の精度を図 2 に示す. この図から, 提案手法はシーケンス長を 2 か 3 とした場合の精度が最も高いことが確認できる. このことから, より直前に発生したリンクが次に発生するリンクと強い依存関係があることが確認できる. 一方で, シーケンス長 4 以上の場合では精度が低下しているため, リンク発生直前の 3 件以上前のリンクは次のリンク発生との依存関係が小さいと考えられる.

表 7: finetuning を行なった提案手法による精度比較. Prop\_Finetune\_all, Prop\_Finetune\_alpha\_mlp, Prop\_Finetune\_mlp はそれぞれ Prop\_AllData の全ての学習パラメータを再学習したモデル, 各行動傾向の重要度と最終層の MLP のみを再学習したモデル, 最終層の MLP のみを再学習したモデルの予測精度を表している. 太字の結果は最高精度であることを表し, 下線の結果は 2 番目に良い結果であることを示す.

datasets	ia-enron-employees		ia-radoslow-email		ia-contact		ia-contacts-hypertext09		rt-pol	
metrics	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10
Prop_AllData	<b>18.32</b>	<b>0.64</b>	<b>9.73</b>	<b>0.74</b>	<u>14.91</u>	<b>0.88</b>	<b>14.99</b>	<b>0.75</b>	329.00	0.09
Prop_SingleData	<u>19.49</u>	0.55	24.97	0.26	23.06	0.68	<u>15.81</u>	<u>0.72</u>	341.51	<u>0.10</u>
Prop_Finetune_all	30.29	0.48	25.08	0.24	35.72	0.37	17.90	0.61	<b>213.91</b>	0.09
Prop_Finetune_alpha_mlp	22.84	0.60	20.18	0.37	<b>14.29</b>	<u>0.84</u>	17.55	0.69	226.70	<b>0.18</b>
Prop_Finetune_mlp	21.35	<u>0.62</u>	<u>18.65</u>	<u>0.41</u>	17.38	0.77	17.23	0.69	<u>224.24</u>	<b>0.18</b>

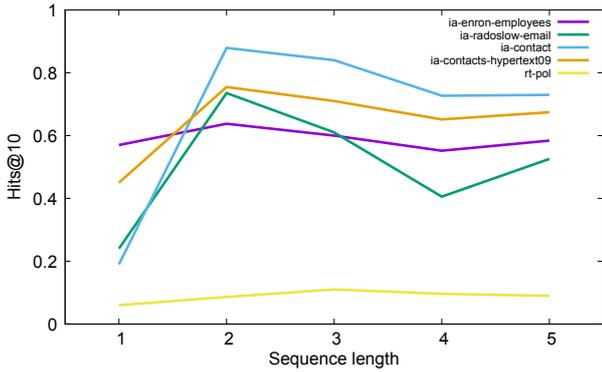


図 2: シーケンス長を変化させた場合の予測精度 (Hits@10)

#### 4.4.4 提案手法の finetuning (Q7)

提案手法を複数データセットで学習した後, テストに用いるデータセットでモデルを finetuning することで予測精度が向上するかを検証する. 本実験では, Prop\_Finetune\_all, Prop\_Finetune\_alpha\_mlp, Prop\_Finetune\_mlp の 3 つの finetuning 方法を用いて提案手法の再学習を行う. 一つ目の Prop\_Finetune\_all は一般的な finetuning 方法に倣ってモデルの全パラメータを再学習したモデルである. 複数データセットで事前学習することで複数データセットに共通するリンク発生傾向を学習した後, テストに用いるデータセットのみを用いて finetuning することで, データセット特有のリンク発生傾向を学習することを期待している. 二つ目の Prop\_Finetune\_alpha\_mlp はノードの行動傾向の重要度を学習するパラメータ  $\alpha_i$  ( $i = 1, 2, 3$ ) と最終層の MLP のみを再学習したモデルである. このモデルでは, ノードの行動傾向は複数データセットで異なる傾向を持たず, ノードの行動傾向の重要度はデータセット毎に異なるという仮説に基づいて再学習を行うモデルである.  $\alpha_i$  と最終層の MLP のみを再学習することで, Transformer encoder の汎化性能を維持しつつ, データセット毎にノードの行動傾向の重要度のみを再学習する. 三つ目の Prop\_Finetune\_mlp は最終層の MLP のみを再学習するモデルである. このモデルでは, ノードの行動傾向とそれらの重要度は異なるデータセットにおいても同様であるという仮説に基づいて再学習を行うモデルである. 最終層の MLP のみを再学習することで, Transformer encoder の汎化性能と複数データセットで共通するノードの行動傾向の重要度を維持しつつ, データセット毎にリンク発生の有無の分類性能のみを再学習する.

Prop\_AllData と Prop\_SingleData の予測精度と上記 3 つの再学習方法でモデルを再学習した予測精度を表 7 に示す. この表から, rt-pol を除いて Prop\_AllData の予測精度が最も優れていることが確認できる. これは, finetuning することにより事前学習モデルの汎化性能が失われ, finetuning に用いた訓練データに過適合してしまっていると考えられる. 一方で, rt-pol においては finetuning を行うことにより精度が向上している. これは, 表 2 に示すように rt-pol のみ他のデータセットと比較してノード数に対するリンク数が少なく, 他のデータセットとは異なるリンク発生傾向を持っていることが原因であると考えられる.

また, Prop\_Finetune\_all はほとんどの結果においてその他の finetuning 方法より低い予測精度になっている. このことから Transformer encoder を再学習することは予測精度の低下につながっていることが確認できる. これは, Transformer encoder を一つのデータセットで再学習することにより, Transformer encoder の汎化性能が低下してしまったことが原因であると考えられる.

## 5 関連研究

本章では, 時系列グラフのリンク予測の関連研究を紹介する. 時系列グラフのリンク予測手法は離散時間グラフを用いる手法 [2-4, 22] と連続時間グラフを用いる手法 [5-9] の二つに分類できる. 離散時間グラフでは, 一定期間毎に取得するグラフのスナップショットによってグラフ構造の時系列変化を表現する. 離散時間グラフを用いる代表的な手法として DynGEM [2] や Dyn2AERNN [3], DySAT [4], EvolveGCN [22] が提案されている. これらの手法では, 各スナップショットの node embedding を算出し, この node embedding を用いてリンク発生確率を学習する. 具体的には, GCN [23] や GAT [24] などの static なグラフで成功を収めている手法と LSTM や self-attention などの時系列手法を組み合わせることにより, ノード特徴の時系列変化を捉えた node embedding を学習している. しかし, このような離散時間グラフを用いる手法では各スナップショット内でのリンク発生順序を考慮できないという問題が存在する.

この問題を解決する手法として, 連続時間グラフを用いた手法が提案されている [5-9]. 連続時間グラフでは 2.1 節で示すように, リンク発生時刻を連続値で表現可能であるため詳

細なリンク発生順序を考慮したリンク予測が可能である。連続時間グラフを用いた手法は embedding ベースの手法と点過程 [25] を用いる手法に分類できる。embedding ベースの手法である TDGNN [6] や JODIE [9] では、任意の時刻の node embedding を取得可能であるため、各時刻のノード特徴を反映した node embedding からリンク予測を行うことが可能である。点過程を用いる手法である DeepCoevolve [7] や DyRep [8] では、リンク発生毎に更新される最新の node embedding からリンク発生強度（微小時間内にリンクが発生する回数の期待値）を学習する。点過程では、強度を用いて任意の時刻のリンク発生確率を算出することが可能である。この点過程を用いた手法は二部グラフや知識グラフのリンク予測でも成功を収めており、近年注目を集めている [15, 26]。

## 6 終わりに

本稿では、複数データセットから共通のリンク発生傾向を学習するリンク予測モデルを提案した。提案手法では、複数データセットで共通の入力データ構造となるリンク履歴シーケンスを作成し、これをモデルの入力とすることで複数データセットのリンク発生を一つのモデルで学習可能にした。また、リンク履歴シーケンスで過去のノードの行動とその行動時間間隔を表現することで、ノードの行動傾向と行動時間間隔を捉えることを可能にした。以上の工夫により、提案手法は既存手法と比較して高精度なリンク予測精度を達成した。今後の課題としては、より多くのデータセットを用いてモデルを訓練することによりさらなる精度向上を図ることが挙げられる。

## 謝 辞

本研究は JSPS 科研費 JP20H00583 の助成を受けたものです。

## 文 献

- [1] Qi Zhang, Yeyun Gong, Jindou Wu, Haoran Huang, and Xuanjing Huang. Retweet prediction with attention-based deep neural network. In *Proceedings of CIKM*, 2016.
- [2] Palash Goyal, Nitin Kamra, Xinran He, and Yan Liu. Dyn-gem: Deep embedding method for dynamic graphs. *arXiv*, 2018.
- [3] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems*, 2020.
- [4] 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.
- [5] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen Ahmed, Eunye Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In *Proceedings of WWW*, 2018.
- [6] Liang Qu, Huaisheng Zhu, Qiqi Duan, and Yuhui Shi. Link prediction via temporal dependent graph neural network. In *Proceedings of WWW*, 2020.
- [7] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. Deep coevolutionary network: Embedding user and item features for recommendation. *arXiv*, 2017.
- [8] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal,

- and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *Proceedings of ICLR*, 2019.
- [9] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of KDD*, 2019.
  - [10] 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.
  - [11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NeurIPS*, 2017.
  - [12] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
  - [13] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of NeurIPS*, 2019.
  - [14] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of KDD*, p. 855–864. Association for Computing Machinery, 2016.
  - [15] 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.
  - [16] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupard. Relational representation learning for dynamic (knowledge) graphs: A survey. *ArXiv*, 2019.
  - [17] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of AAAI*, 2015.
  - [18] 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.
  - [19] M.D. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, A. Flammini, and F. Menczer. Political polarization on twitter. In *Proceedings of ICWSM*, 2011.
  - [20] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of NeurIPS*, 2017.
  - [21] 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.
  - [22] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E Leiserson. Evolvegn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of AAAI*, 2020.
  - [23] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.
  - [24] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *ICLR*, 2018.
  - [25] O. Aalen, O. Borgan, and H. Gjessing. *Survival and Event History Analysis: A Process Point of View*. Statistics for Biology and Health. Springer New York, 2008.
  - [26] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of ICML*, p. 3462–3471, 2017.