

AIの最前線、AIはこう使う！？～業界でのことを聞いてみた～



# AIとグラフ

## グラフ深層学習の現在と今後と応用

大阪大学 大学院情報科学研究科  
佐々木勇和

SS研 ICTフォーラム, 富士通汐留本社, 2023 Aug. 7th

# 佐々木 勇和 Yuya Sasaki



助教 @ 大阪大学 2016/04~  
さきがけ「信頼されるAI」領域 2021/10~

## 研究テーマ

グラフデータ分析と管理

時空間・モバイルデータ分析と管理

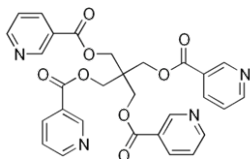
情報処理技術の異分野適用

## 研究プロジェクト

- ・ グラフデータの説明可能なバイアスに関する基盤技術の創出
- ・ 機械学習を活用した自己最適化グラフデータベース管理システムの開発と検証
- ・ 化学: 分子グラフ機械学習を用いた有機太陽電池と有機トランジスタの学理融合と新規材料創製
- ・ 医学: 人工知能技術を活用した尿路結石の発症予測モデルの構築
- ・ 哲学: 人間を対象とする人工知能技術とアフターマティブアクション
- ・ 都市工学: Society5.0における社会課題解決に向けた利用者誘引型低遅延MaaS基盤

# 多様なデータがグラフで表現される

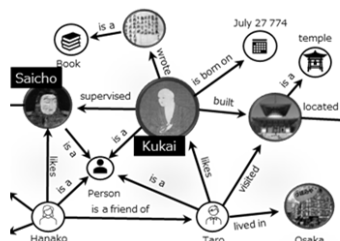
分子



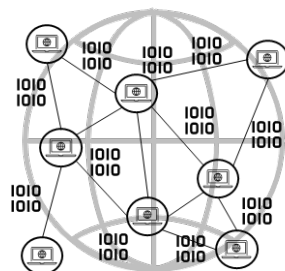
道路網



知識



ウェブ



神経ネットワーク

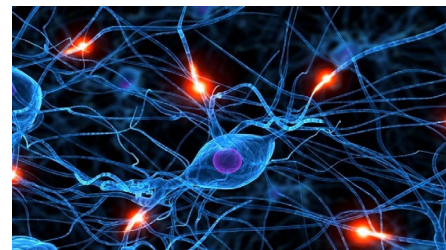
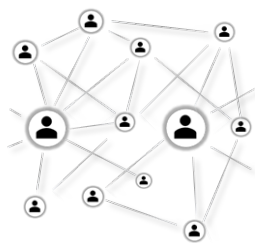


image credit [[viso.ai](https://www.viso.ai)]

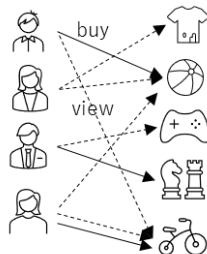
通信網



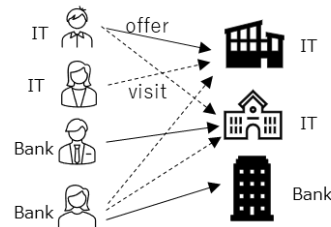
人間関係



購買履歴



ジョブマッチング

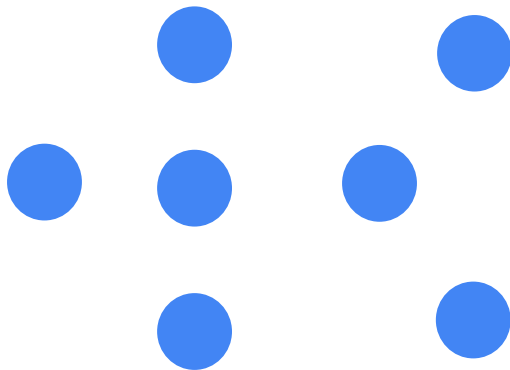


引用グラフ



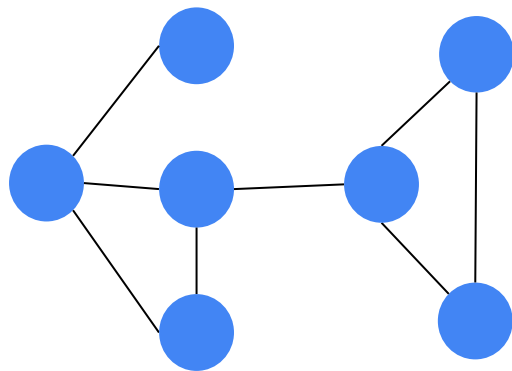
グラフは様々なデータの**一般表現**。

具体的には、データポイント間の関係や相互依存を表現できる



# グラフは様々なデータの**一般表現**。

具体的には、データポイント間の関係や相互依存を表現できる

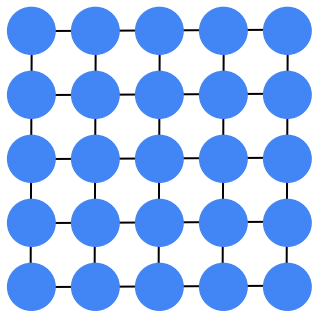


○ : Node/頂点  
— : Edge/辺

**Graph/グラフ**

# 今日では、強力な機械学習ツールがある

画像



テキスト/スピーチ

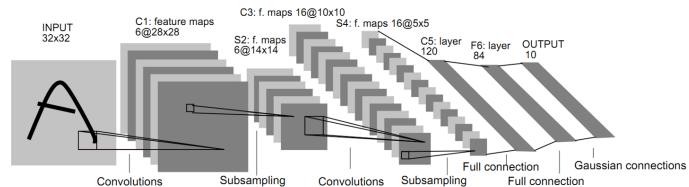
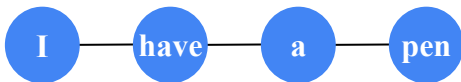
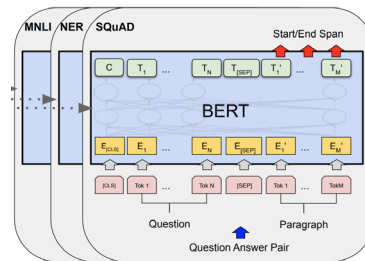


image credit [\[LeCun et al.\]](#)

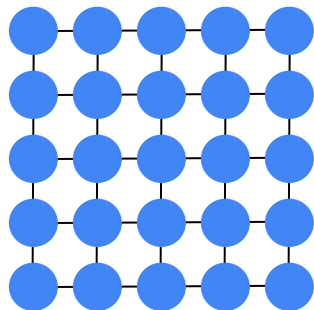


Fine-Tuning

image credit [\[Devlin et al.\]](#)

# 今日では、強力な機械学習ツールがある

画像



テキスト/スピーチ

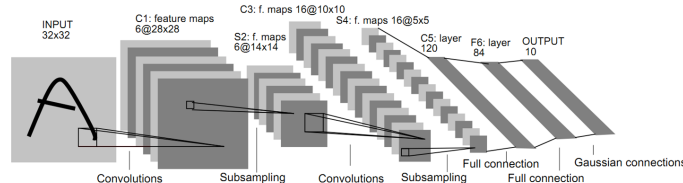
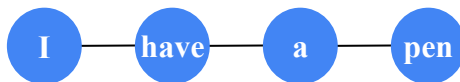
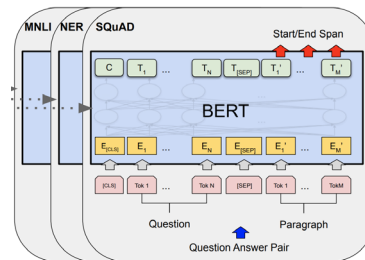


image credit [\[LeCun et al.\]](#)

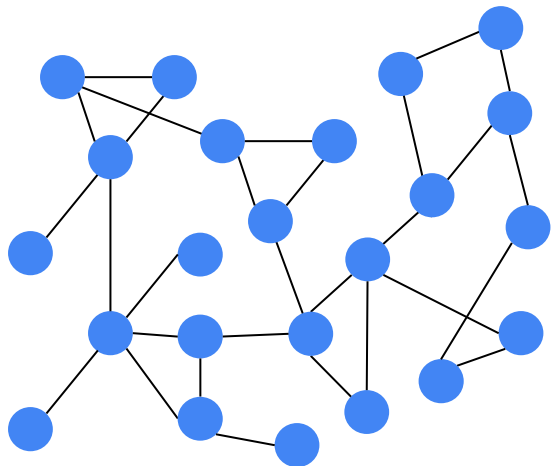


Fine-Tuning

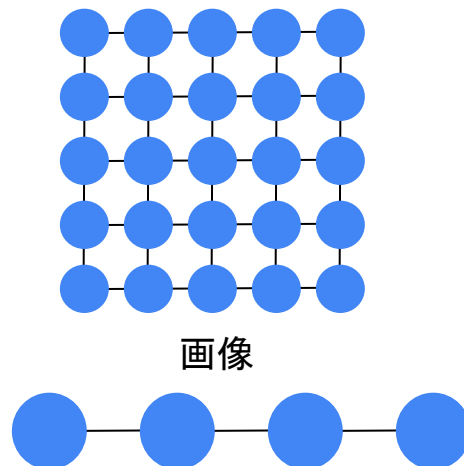
image credit [\[Devlin et al.\]](#)

これらの深層学習ツールボックスは、  
単純なグリッドかシーケンスのために設計されている

# しかし、すべてがグリッドやシーケンスで表現できるわけではない



グラフ/ネットワーク



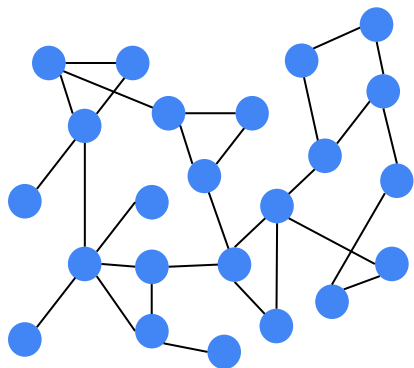
画像

テキスト/スピーチ

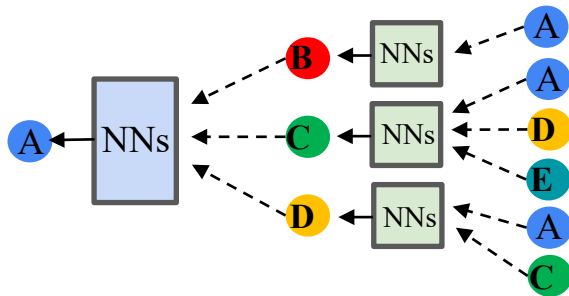
- 任意のサイズや複雑な構造を取れる
- 決まったノードの順番がない
- 時系列の変化や、マルチモーダルな特徴量がある場合もしばしば



より広い一般的なデータ（グラフ）に適用できる  
深層学習を開発することが大きな課題になっている



グラフ/ネットワーク



グラフ深層学習

**深層学習の未開拓エリアとして、研究者が殺到中**

# 機械学習分野のホットトピック

## NeurIPS 2020 Top keywords

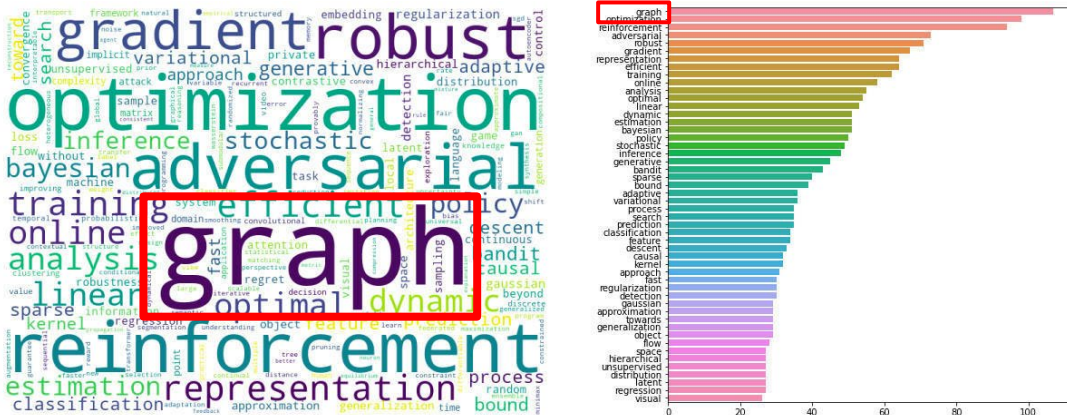


image credit [\[MACNICA\]](#)

## VLDB2022

- グラフ関連セッション 9件(44件中) = 20%

## ICLR 2021 Top keywords

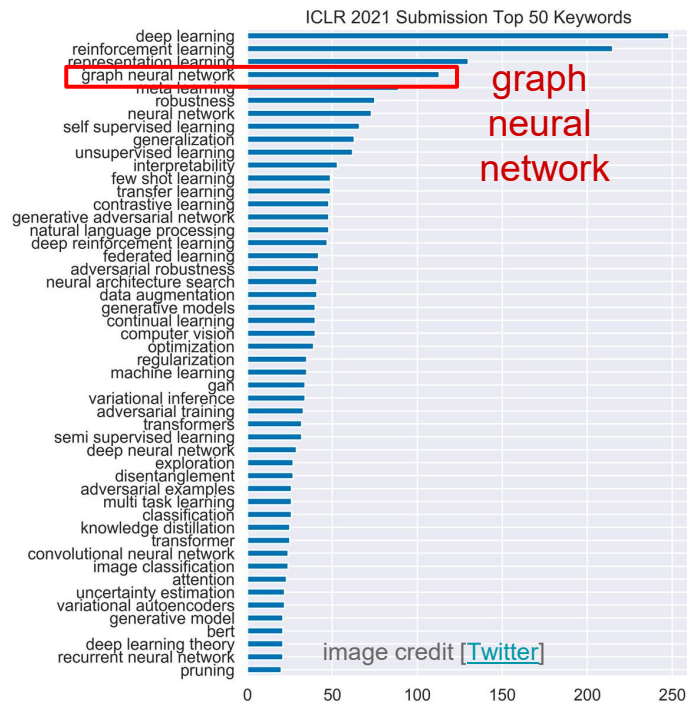


image credit [\[Twitter\]](#)

## 本日の目的：

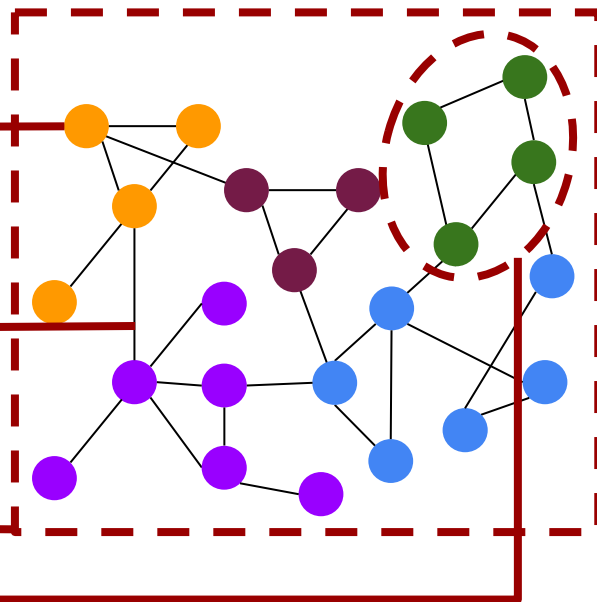
グラフ分析がどんどん熱い分野になってきていることを共有する

グラフ深層学習の気持ちを伝える

グラフ分析の応用例を伝える

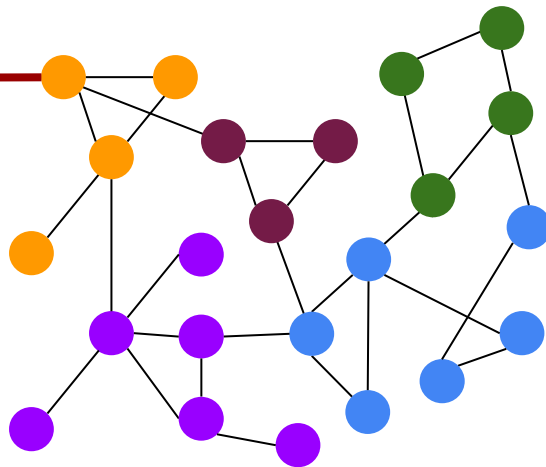
# グラフ分野の主要なタスク

- ノードレベル ←
- リンクレベル ←
- グラフレベル  
(部分グラフレベル) ←



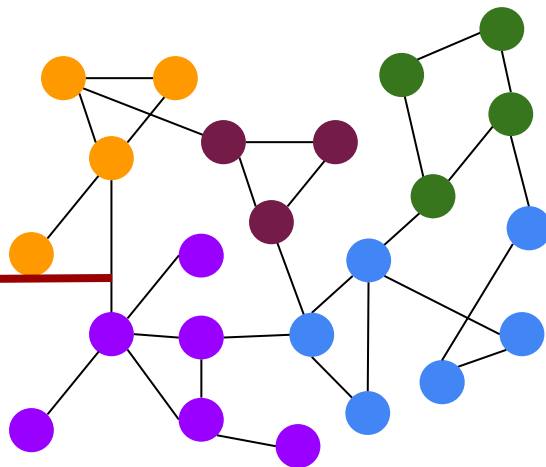
# グラフ分野の主要なタスク

- ノードレベル  
例：SNSユーザの分類  
論文のトピック分類



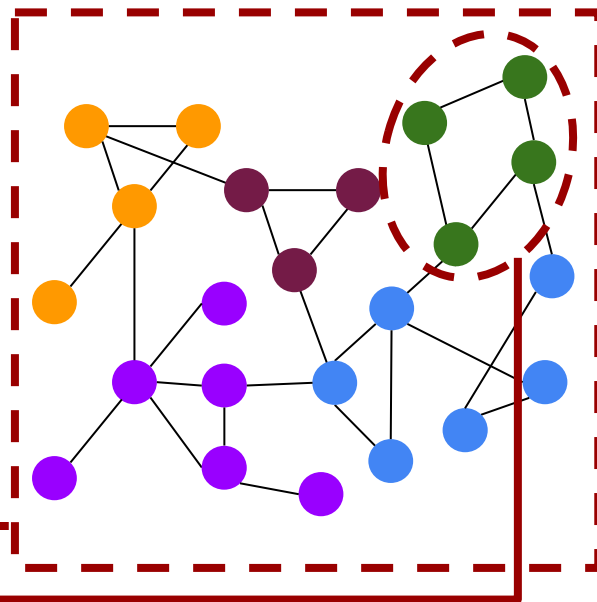
# グラフ分野の主要なタスク

- ノードレベル  
例：SNSユーザの分類  
論文のトピック分類
- リンクレベル ←  
例：フォロワー予測  
商品推薦

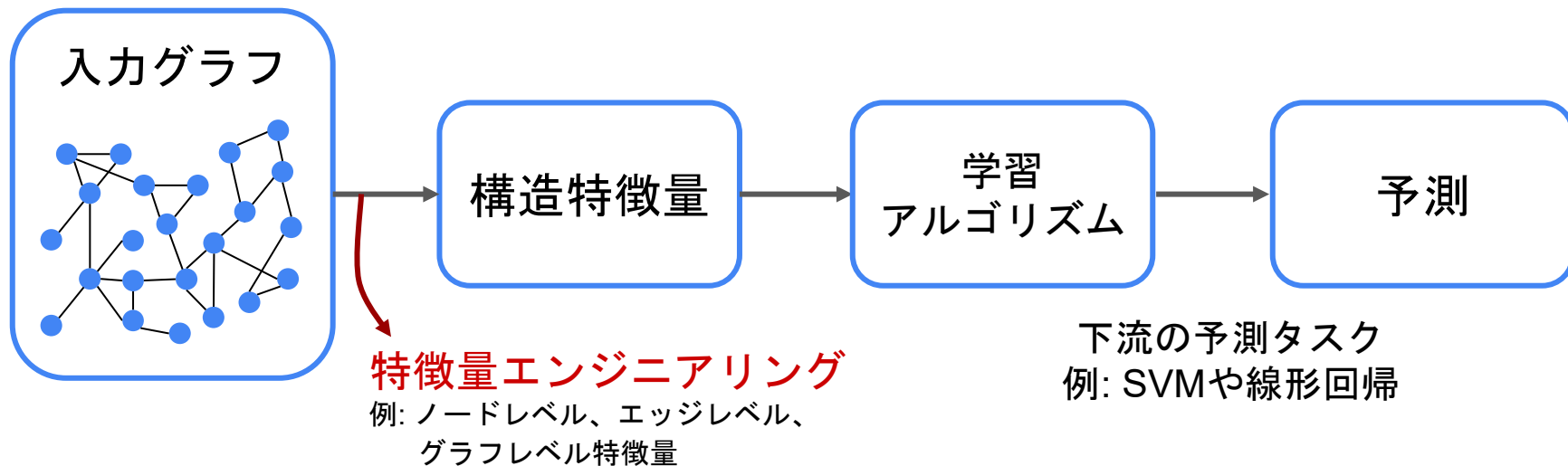


# グラフ分野の主要なタスク

- ノードレベル  
例：SNSユーザの分類  
論文のトピック分類
- リンクレベル  
例：フォロワー予測  
商品推薦
- グラフレベル  
(部分グラフレベル)  
例：分子の特性予測  
問合せ処理

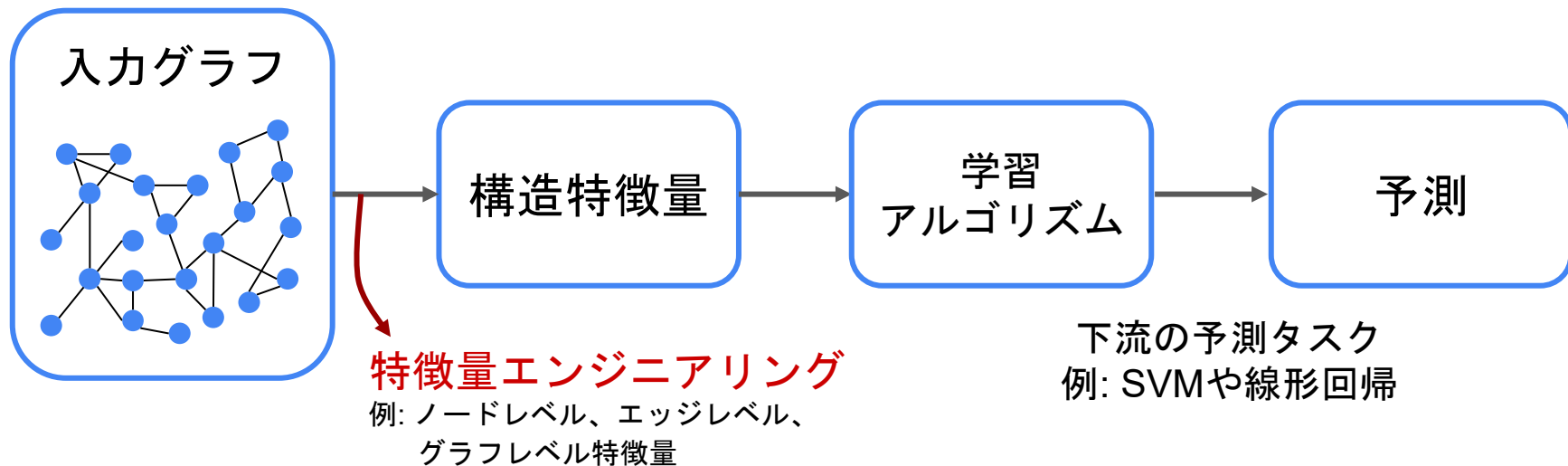


# 伝統的なグラフ機械学習手法のアプローチ



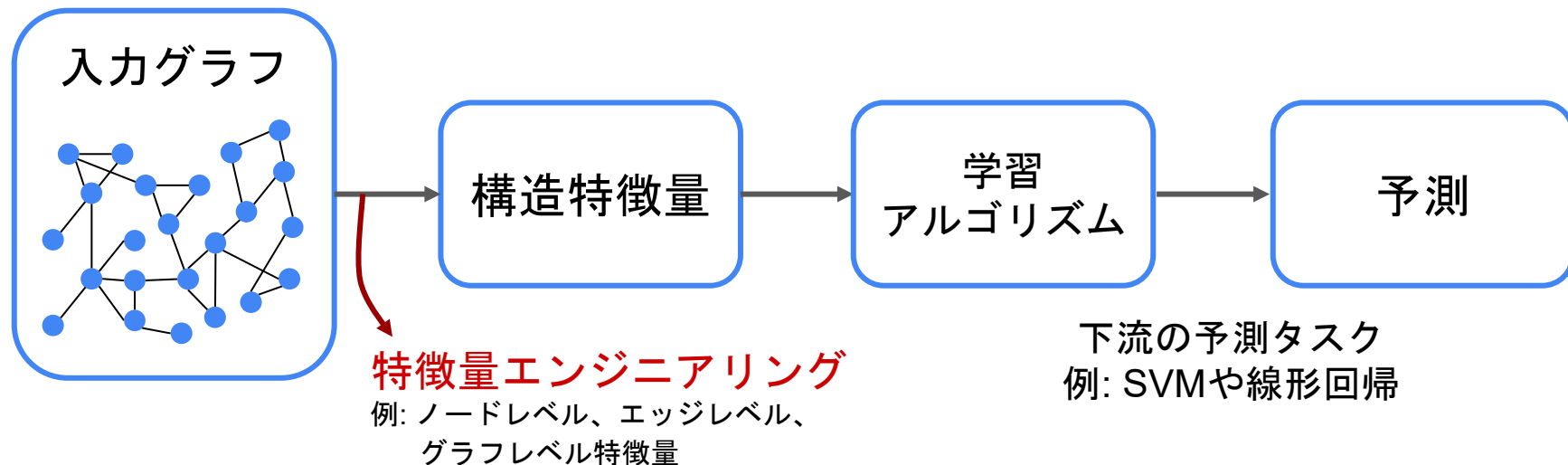


# 伝統的なグラフ機械学習手法のアプローチ



タスクごとに特徴量の設計を行う必要がある

# 伝統的なグラフ機械学習手法のアプローチ

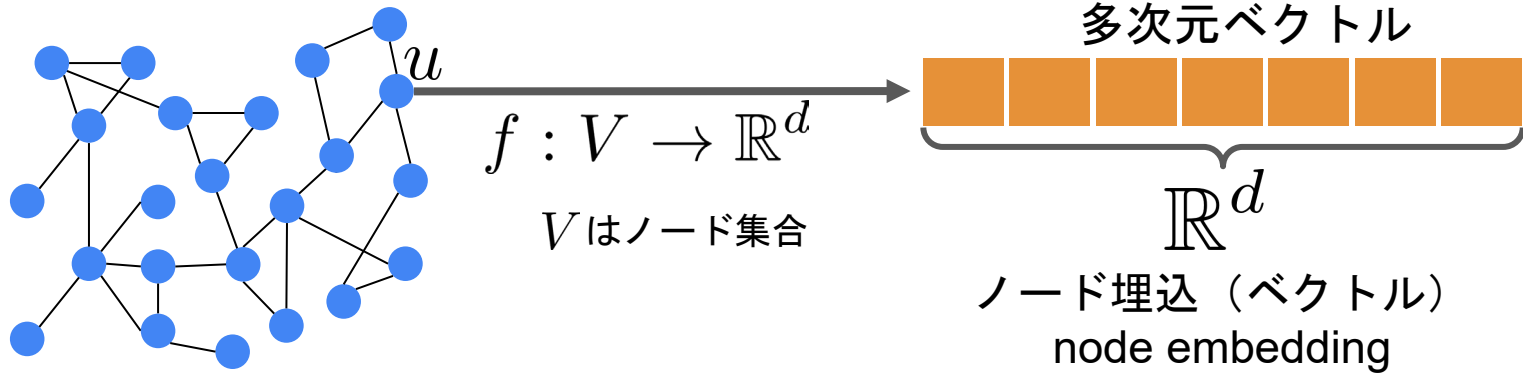


タスクごとに特徴量の設計を行う必要がある

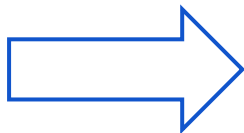
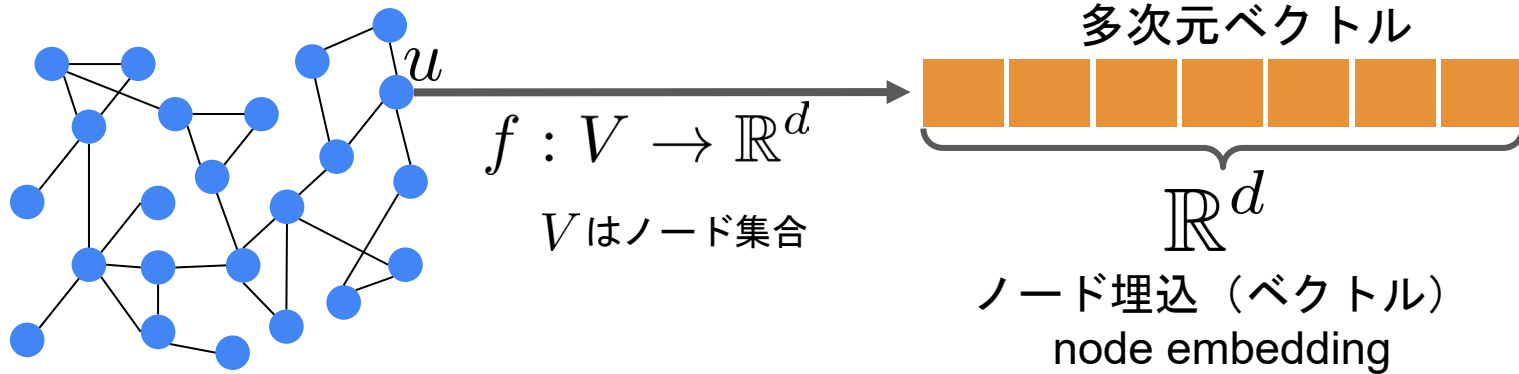


多様なタスクで使える特徴量があればなあ...

# グラフ表現学習 1/2



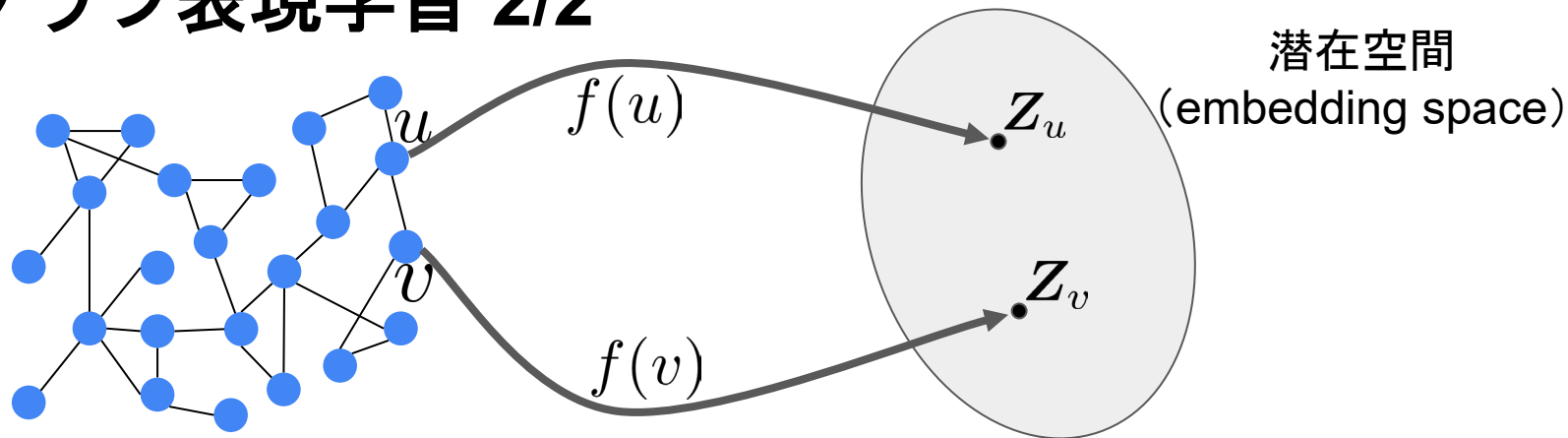
# グラフ表現学習 1/2



## 下流タスク

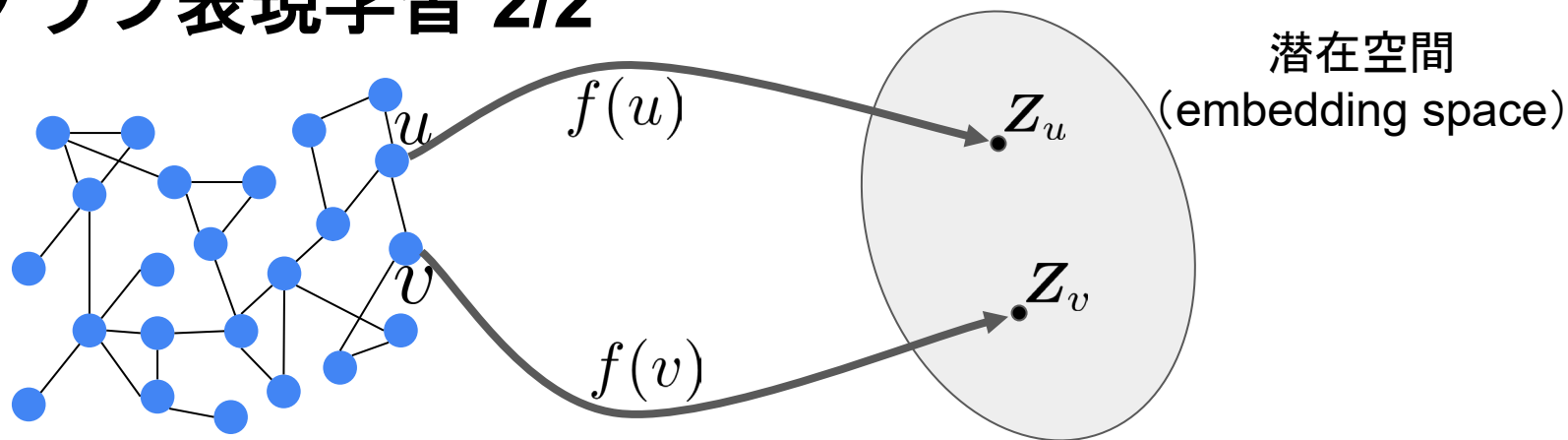
- ノード分類
- リンク予測
- グラフ分類
- 異常ノード検知
- クラスタリング

## グラフ表現学習 2/2



潜在空間に埋込むことで、  
他のデータと同じように扱うことができる！

## グラフ表現学習 2/2

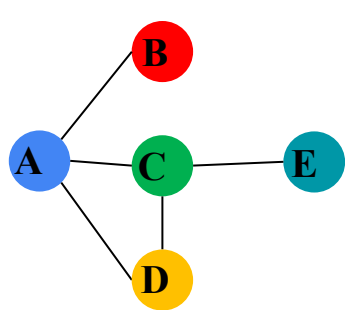


潜在空間に埋込むことで、  
他のデータと同じように扱うことができる！



どうすれば、うまくグラフを埋込めるだろうか

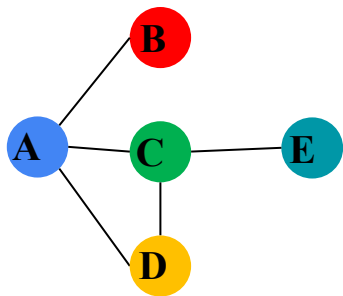
# 埋込の単純なアプローチ



	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	0	0
C	1	0	0	1	1
D	1	0	1	0	0
E	0	0	1	0	0

隣接行列

# 埋込の単純なアプローチ

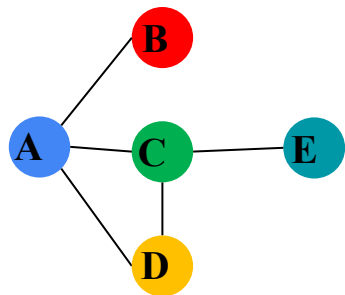


	A	B	C	D	E	ノード 特徴量
A	0	1	1	1	0	0 1
B	1	0	0	0	0	0 0
C	1	0	0	1	1	1 1
D	1	0	1	0	0	1 0
E	0	0	1	0	0	0 0

隣接行列

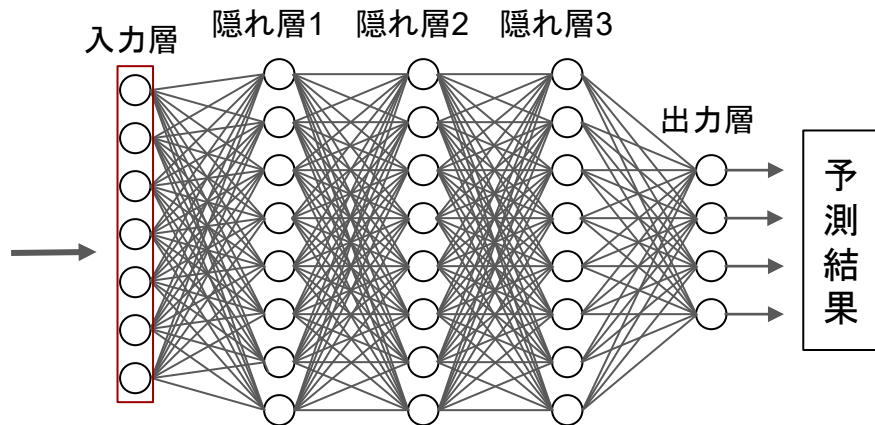


# 埋込の単純なアプローチ

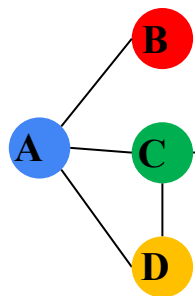


	A	B	C	D	E	ノード 特徴量
A	0	1	1	1	0	0 1
B	1	0	0	0	0	0 0
C	1	0	0	1	1	1 1
D	1	0	1	0	0	1 0
E	0	0	1	0	0	0 0

隣接行列

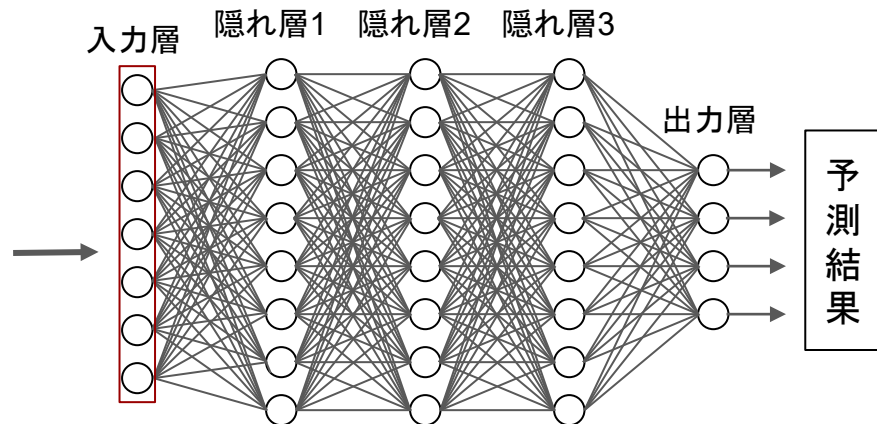


# 埋込の単純なアプローチ



	A	B	C	D	E	ノード 特徴量
A	0	1	1	1	0	0 1
B	1	0	0	0	0	0 0
C	1	0	0	1	1	1 1
D	1	0	1	0	0	1 0
E	0	0	1	0	0	0 0

隣接行列



## 課題

- ノード数のパラメータが必要
- ノードの順番によって結果が変化



# 畳込みニューラルネット（CNN）のアイデアが良いのでは？

## 画像データでの畳込み

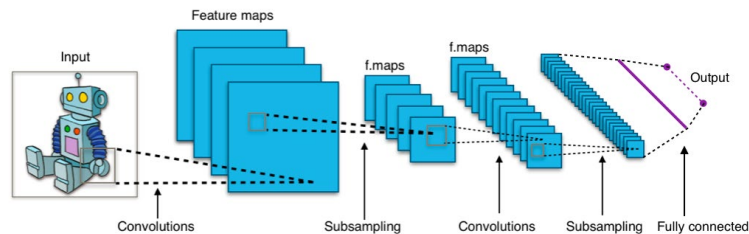
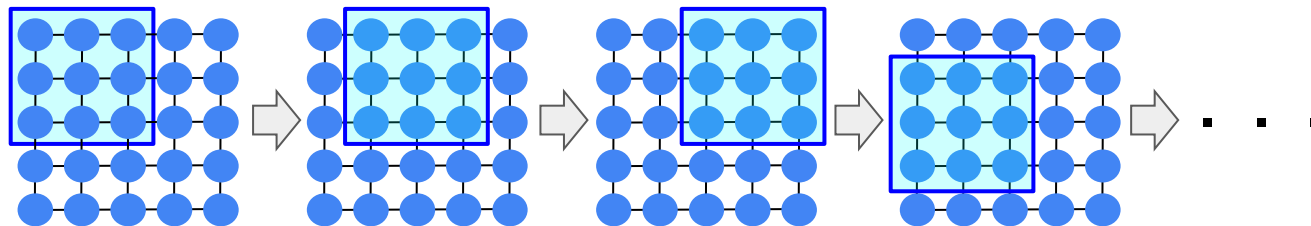


image credit [\[Wikipedia\]](#)

# 畳込みニューラルネット（CNN）のアイデアが良いのでは？

## 画像データでの畳込み

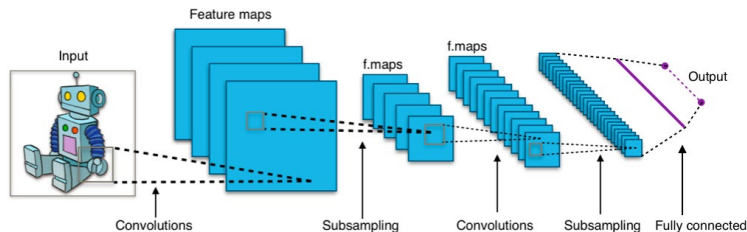
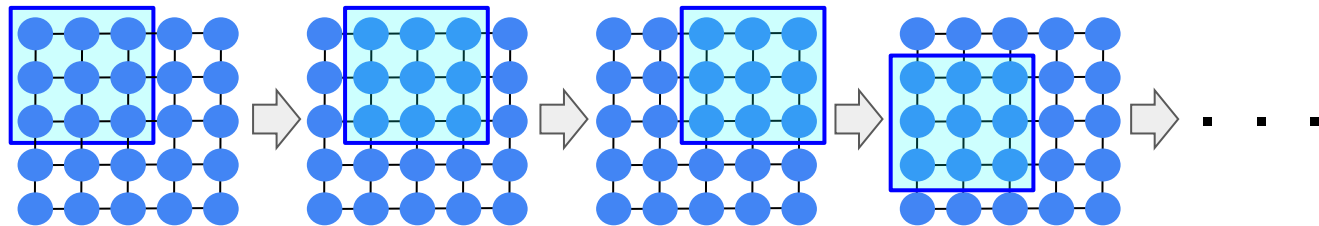


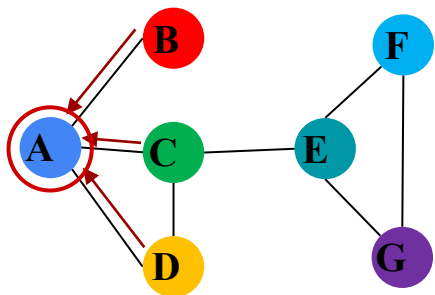
image credit [\[Wikipedia\]](#)



単純なグリッド向けのCNNをグラフに  
一般化できないだろうか？

# グラフ畳込みは、グラフ深層学習の常識を変えた

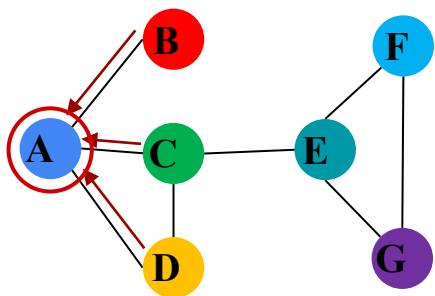
🤔 グラフ上での近傍を使って、畳込みできないか



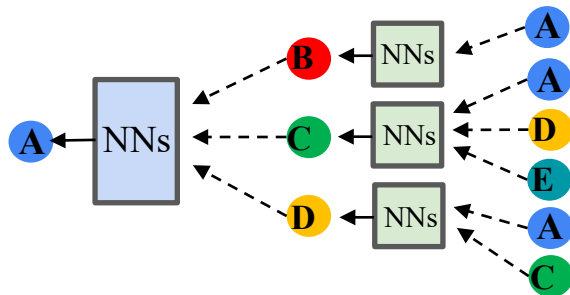
# グラフ畳込みは、グラフ深層学習の常識を変えた



グラフ上での近傍を使って、畳込みできないか



ノードAの  
計算グラフ



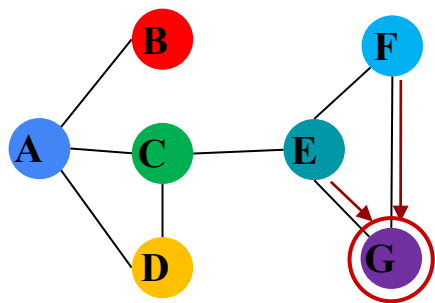
ノードの初期値は  
入力属性を使うか  
ランダム値を割当

ノード埋込は近傍の情報を保持することで  
下流タスクの高精度化に貢献！

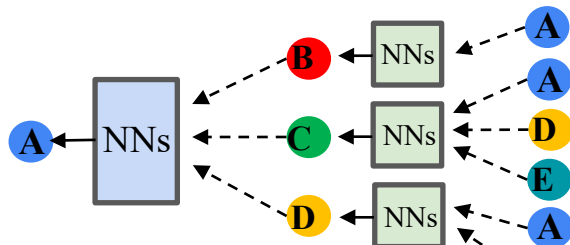
# グラフ畳込みは、グラフ深層学習の常識を変えた



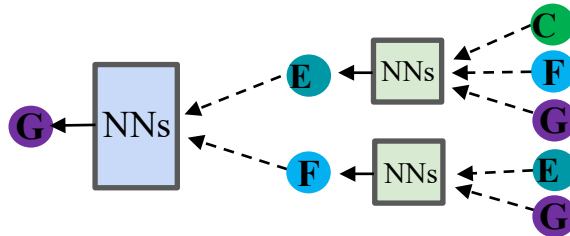
グラフ上での近傍を使って、畳込みできないか



ノードAの  
計算グラフ



ノードGの  
計算グラフ



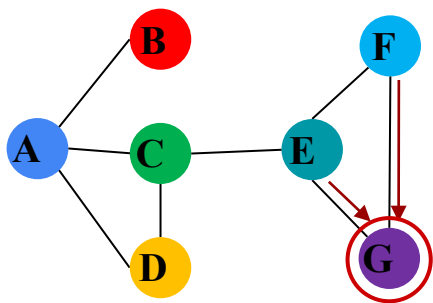
ノードの初期値は  
入力属性を使うか  
ランダム値を割当

ノード埋込は近傍の情報を保持することで  
下流タスクの高精度化に貢献！

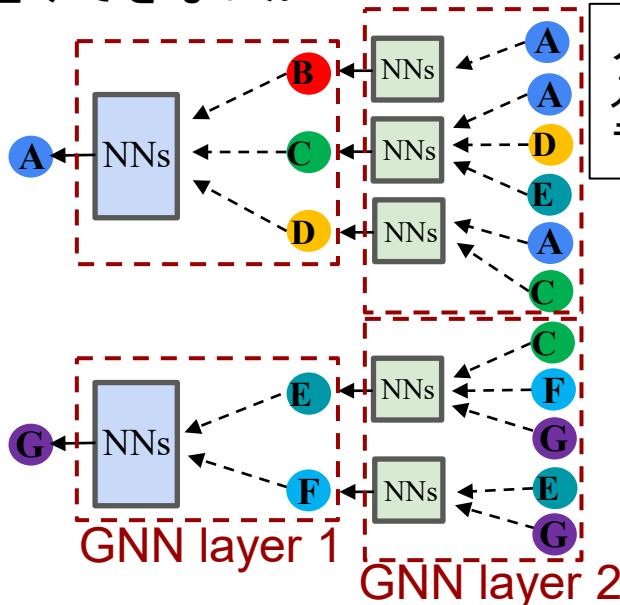
# グラフ畳込みは、グラフ深層学習の常識を変えた



グラフ上での近傍を使って、畳込みできないか



ノードAの  
計算グラフ



ノードの初期値は  
入力属性を使うか  
ランダム値を割当

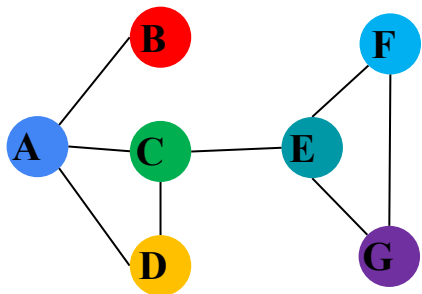
ノード埋込は近傍の情報を保持することで  
下流タスクの高精度化に貢献！



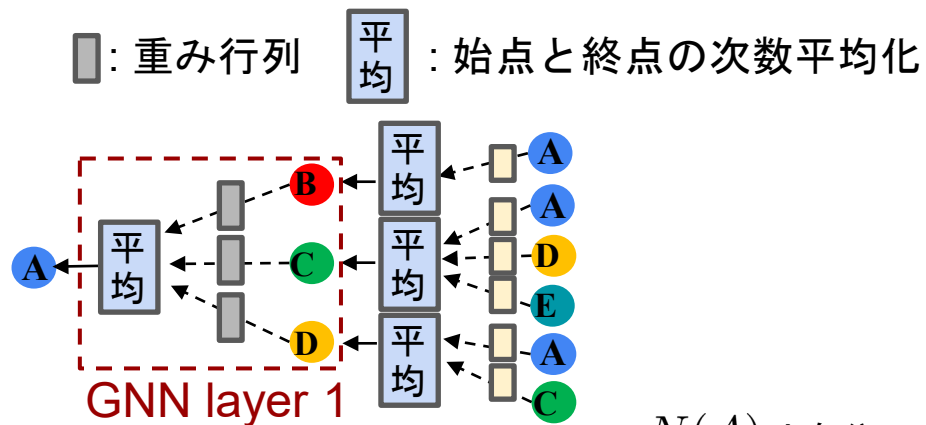
# グラフ畳込み: GCN (Graph Convolutional Networks) [1]

GCNは最もベーシックなGNNであり、  
二つのプロセスからなる：

1. 伝播 (propagation/message)
2. 集約 (aggregation)



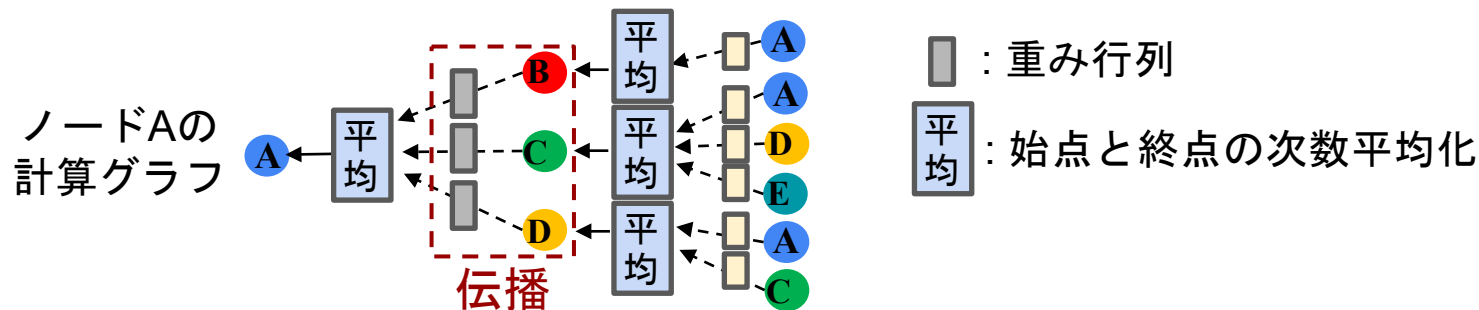
ノードAの  
計算グラフ



$$h_A^{(l)} = \sigma \left( \mathbf{W}^{(l)} \sum_{u \in N(A)} \frac{h_u^{(l-1)}}{\sqrt{d_A} \sqrt{d_u}} \right)$$

$N(A)$ は自分  
(この場合A)を  
含めた隣接集合  
 $d_A$ はAの次数

# グラフ畳込み: GCN (Graph Convolutional Networks) [1]

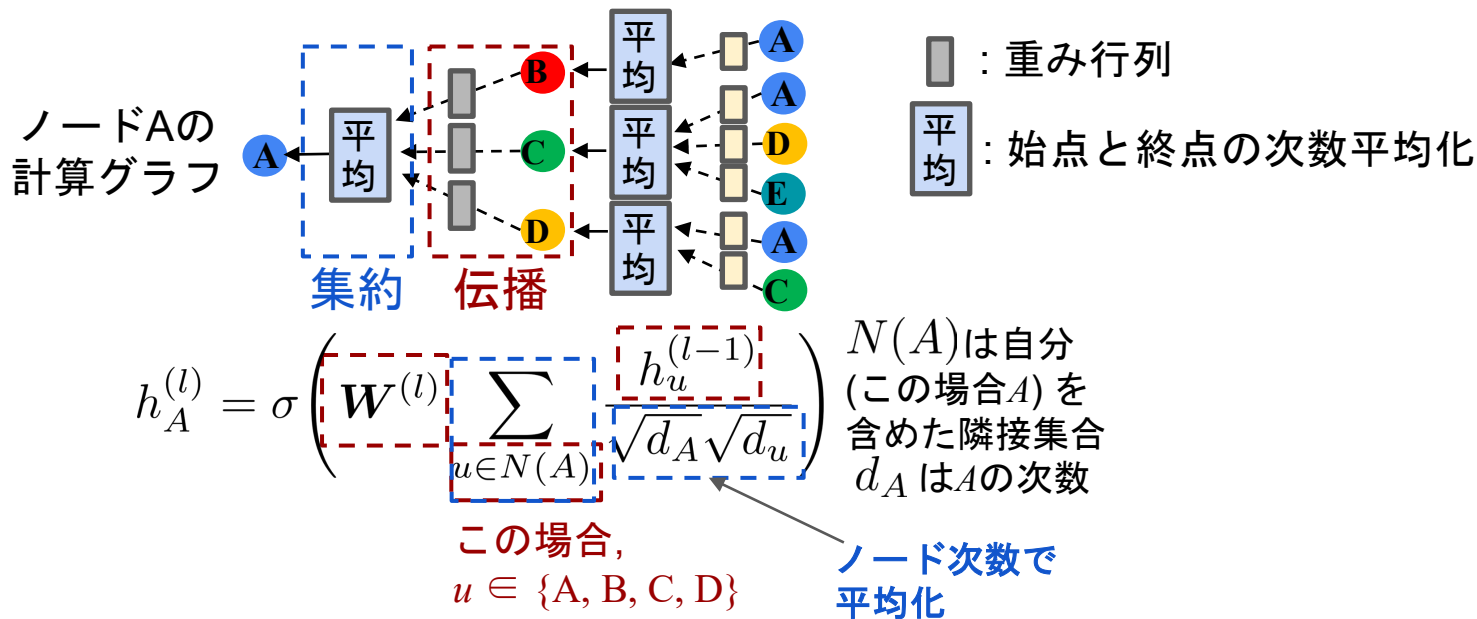


$$h_A^{(l)} = \sigma \left( \mathbf{W}^{(l)} \sum_{u \in N(A)} \frac{h_u^{(l-1)}}{\sqrt{d_A} \sqrt{d_u}} \right)$$

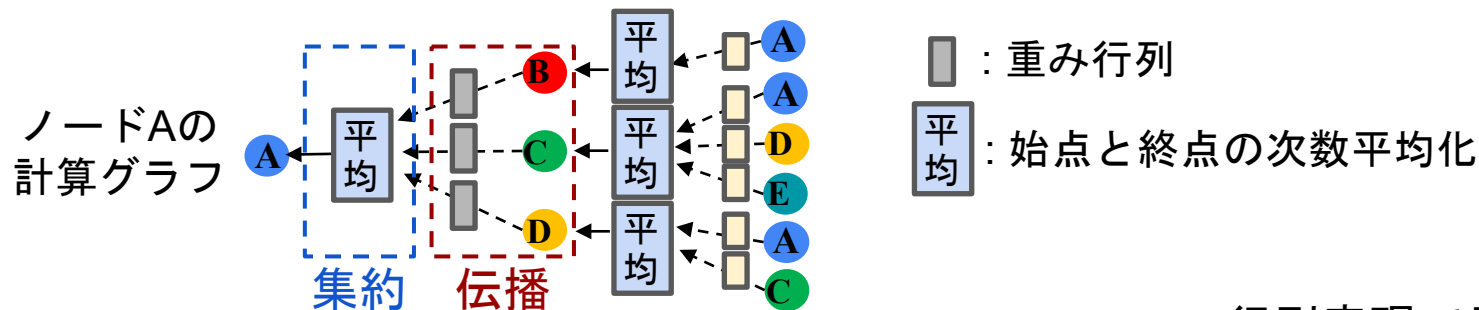
この場合,  
 $u \in \{A, B, C, D\}$

$N(A)$ は自分  
 (この場合A)を  
 含めた隣接集合  
 $d_A$ はAの次数

# グラフ畳込み: GCN (Graph Convolutional Networks) [1]



# グラフ畳込み: GCN (Graph Convolutional Networks) [1]



$$h_A^{(l)} = \sigma \left( \mathbf{W}^{(l)} \sum_{u \in N(A)} \frac{h_u^{(l-1)}}{\sqrt{d_A} \sqrt{d_u}} \right)$$

$N(A)$ は自分(この場合A)を含めた隣接集合  
 $d_A$ はAの次数

この場合,  
 $u \in \{A, B, C, D\}$

ノード次数で平均化

行列表現で書くと:

$$\mathbf{H}^{(l)} = [h_1^{(l)}, \dots, h_n^{(l)}]^T$$

$$\mathbf{H}^{(l)} = \sigma(\mathbf{S}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)})$$

$\mathbf{S}$ は次数正規化した隣接行列

# グラフ畳込みの拡張手法

## 精度の向上



グラフ畳込みの表現力が低く、異なるグラフを区別できない場合がある

### 二つの派閥

- homophily仮定：隣接するノードは似ている
- non-homophily仮定：グラフによっては隣接するノードは似ていない

## 効率性/スケーラビリティの向上



グラフが大きくなると、学習のたびグラフ全体を畳込むのは非効率

# homophilous/non-homophilous

- 2018年まではグラフにhomophilyを仮定  
例：SNSのコミュニティ推定  
文書のトピック予測
- 2019年からhomophilyを持たない  
グラフへの拡張が流行  
例：SNSでの性別推定  
文書の年代予測

homophilous グラフの例



image credit [techjournal]

[New!] non-homophilous グラフの例

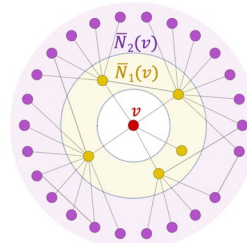


image credit [2]

# homophilous/non-homophilous グラフの比較

homophilous グラフ

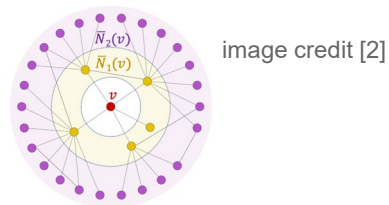


近傍は同じクラスの可能性**高**

畳込みによって同じクラスに  
属するノードが似た  
embeddingを持つ

GCNで**高**精度

non-homophilous グラフ



近傍は同じクラスの可能性**低**

畳込みすると異なるクラスに  
属するノードとembeddingが  
似てしまう

GCNは**低**精度

homophilous/non-homophilous グラフの両方で動作する手法の研究が盛ん

# グラフ畳込みの拡張手法

## 精度の向上



グラフ畳込みの表現力が低く、異なるグラフを区別できない場合がある

### 二つの派閥

- homophily仮定：隣接するノードは似ている
- non-homophily仮定：グラフによっては隣接するノードは似ていない

## 効率性/スケーラビリティの向上



グラフが大きくなると、学習のたびグラフ全体を畳込むのは非効率

### 二つの派閥

- サンプリングベース
- 事前計算ベース



# サンプリングベースと事前計算ベースの比較

## サンプリングベース

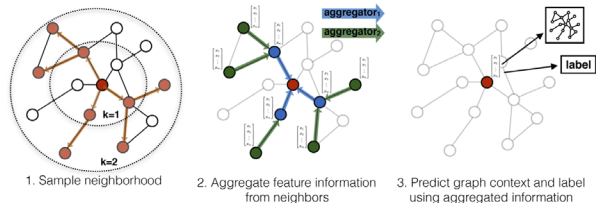


image credit [5]

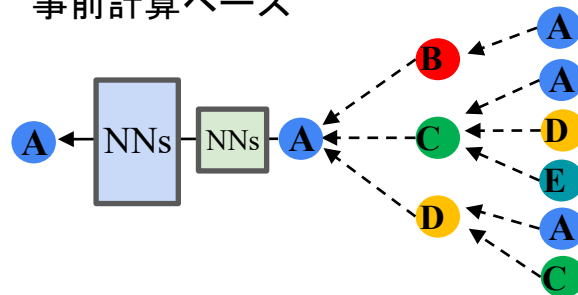
事前計算：無し

学習のメモリ消費量：小

学習時間：大

予測精度：高

## 事前計算ベース



事前計算の時間・メモリ消費量：大

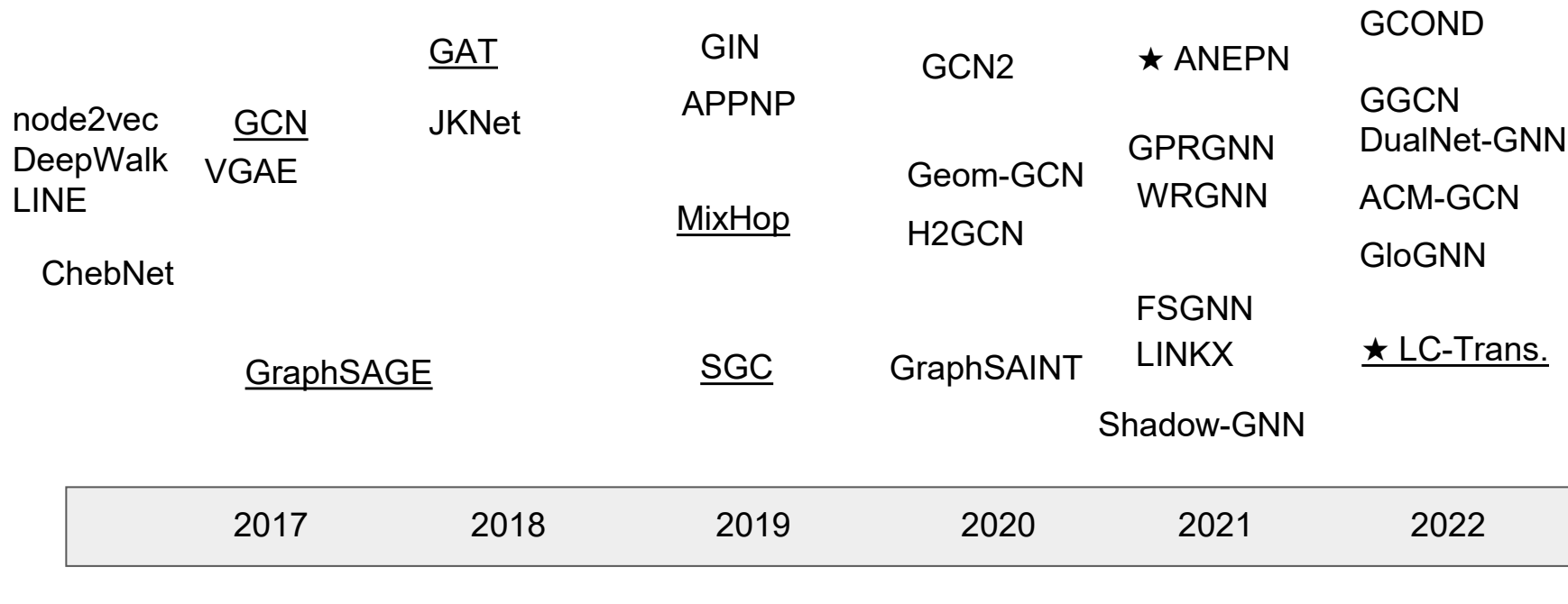
学習のメモリ消費量：小

学習時間：小

予測精度：高

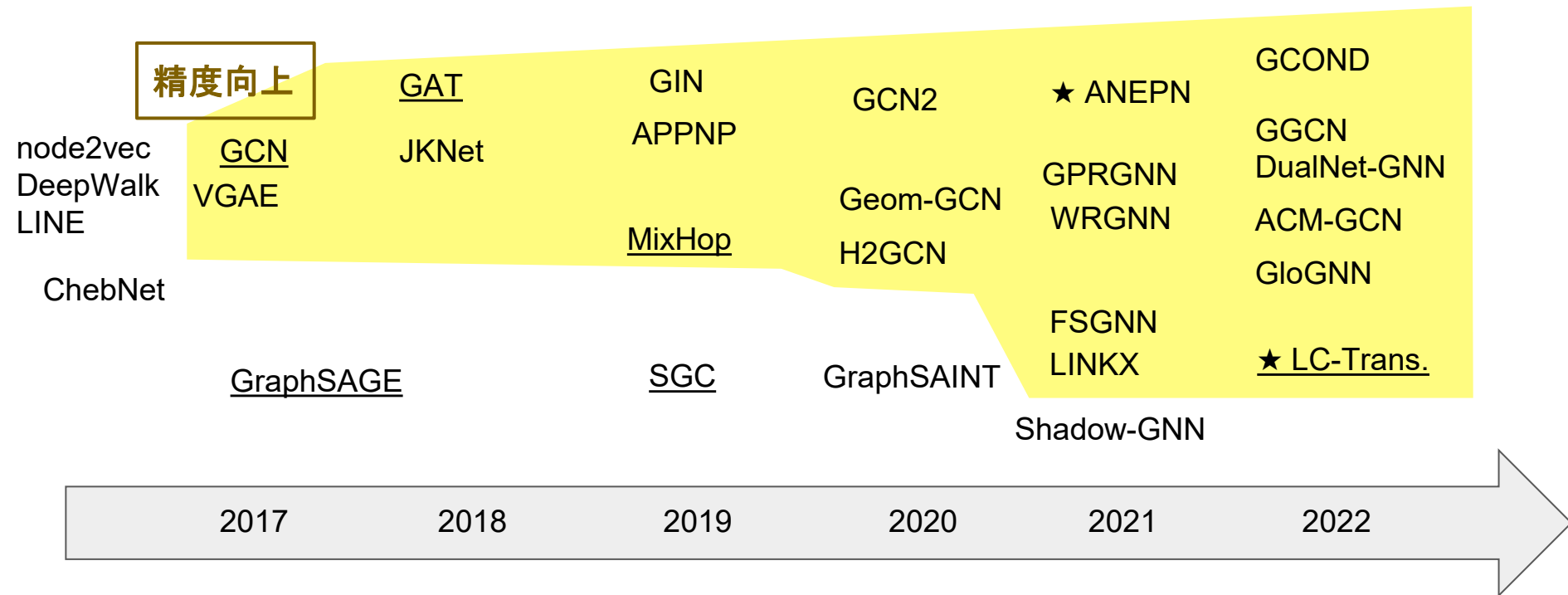
どちらも盛んに研究されていて、トップ会議で新しい手法が提案されている

# グラフ分析 & グラフ深層学習の大まかな流れ



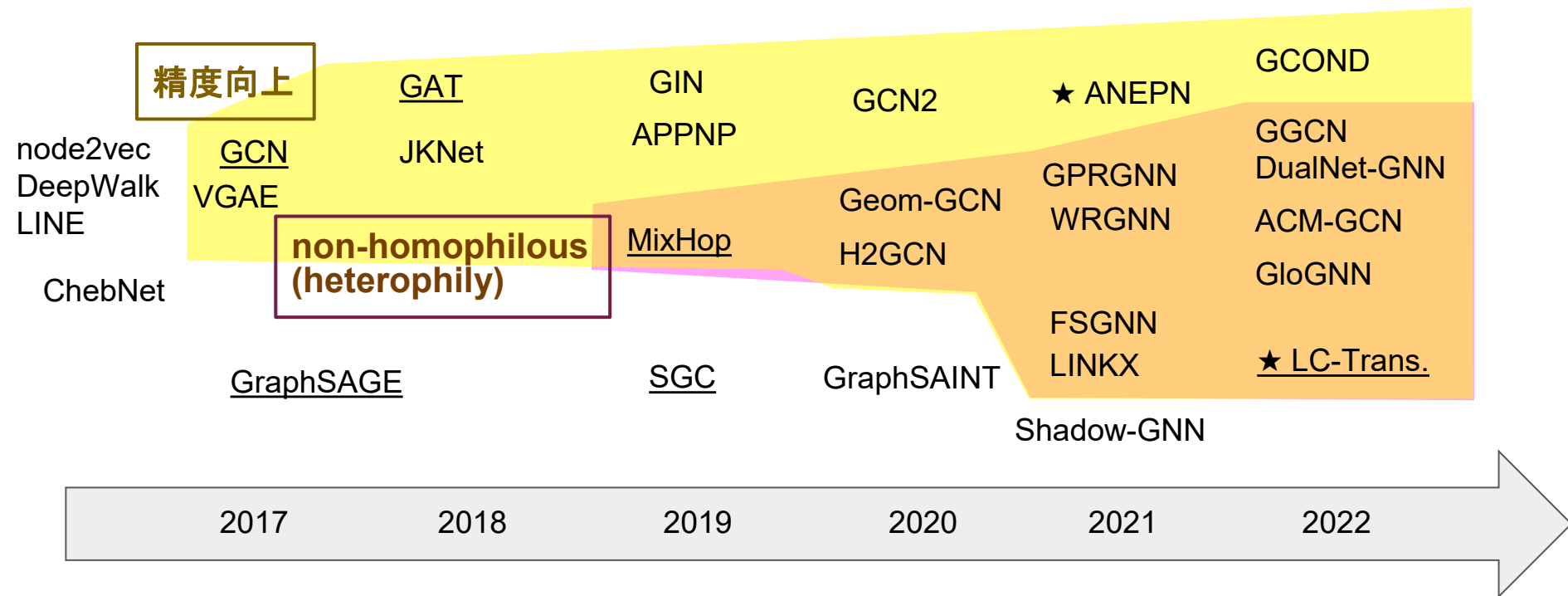
※referenceは最後にまとめてあります。★は我々の研究グループが提案した手法。

# グラフ分析 & グラフ深層学習の大まかな流れ



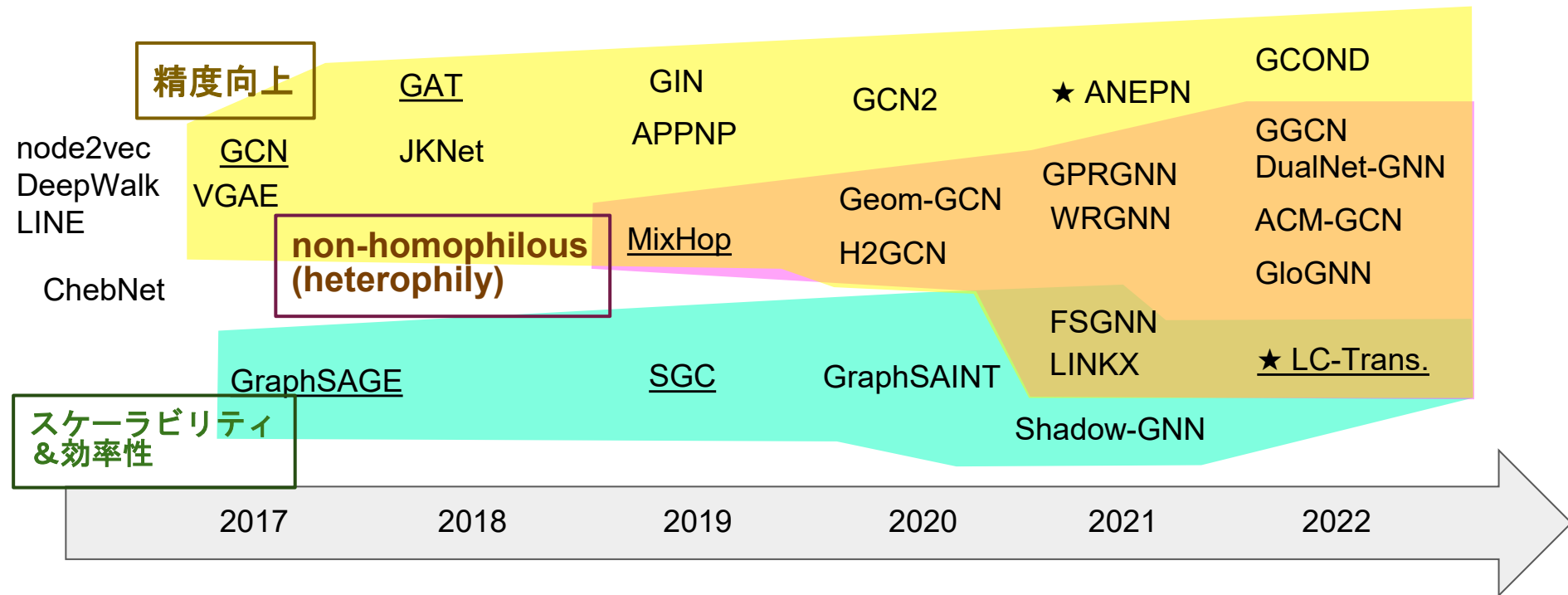
※referenceは最後にまとめてあります。★は我々の研究グループが提案した手法。

# グラフ分析 & グラフ深層学習の大まかな流れ



※referenceは最後にまとめてあります。★は我々の研究グループが提案した手法。

# グラフ分析 & グラフ深層学習の大まかな流れ



※referenceは最後にまとめてあります。★は我々の研究グループが提案した手法。


## グラフ深層学習のこれまで

- 精度改善
  - トップ会議でも0.1%の改善を競う状況 😞
- スケーラビリティ
  - エッジ数が10億を超えても動作可能な手法が提案

単純なグラフ（一つのノード/エッジタイプ）での手法の  
改善は**頭打ち**になってきている印象

公平な評価のために、いくつかの大規模データは提案されている [8]  
しかし、様々な思想を持った大量の手法が提案されていて、カオスな状況

## グラフ深層学習のこれまで

- 精度改善
  - トップ会議でも0.1%の改善を競う状況 
- スケーラビリティ
  - エッジ数が10億を超えても動作可能な手法が提案


単純なグラフ（一つのノード/エッジタイプ）での手法の  
改善は**頭打ち**になってきている印象

公平な評価のために、いくつかの大規模データは提案されている [8]  
しかし、様々な思想を持った大量の手法が提案されていて、カオスな状況。。。。

 誰か色々な実験設定で、手法をベンチマークしてくれないだろうか

## 我々の研究 [9]: 大量の既存研究をベンチマークしておこう

目的：多様な実験設定で、既存GNNsの利点/欠点を調べる

着想：うまく人工グラフを作れば、有益な比較実験ができないか？ 

手段：20以上の多様なグラフと17種類のGNNsを使って、実験しよう！



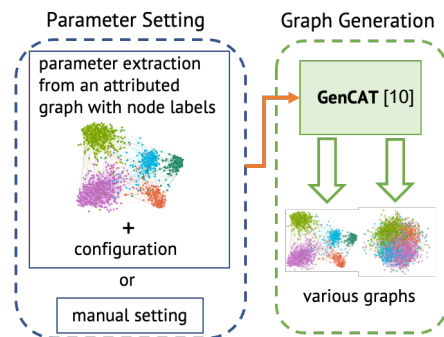
# 我々の研究 [9]: 大量の既存研究をベンチマークしておこう

目的：多様な実験設定で、既存GNNsの利点/欠点を調べる

着想：うまく人工グラフを作れば、有益な比較実験ができないか？ 🤔

手段：20以上の多様なグラフと17種類のGNNsを使って、実験しよう！

所望の人工グラフを作成



[9] “Beyond Real-world Benchmark Datasets: An Empirical Study of Node Classification with GNNs”, Maekawa et al., NeurIPS Datasets & Benchmarks 2022.

[10] “GenCAT: Generating Attributed Graphs with Controlled Relationships between Classes, Attributes, and Topology.”, Maekawa et al., Information Systems 2023.

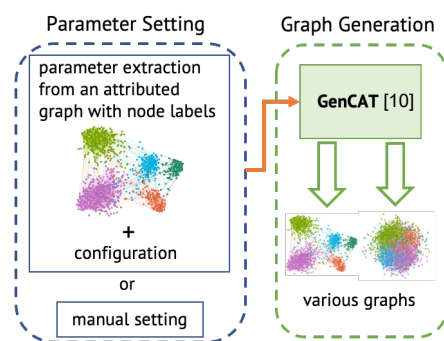
# 我々の研究 [9]: 大量の既存研究をベンチマークしておこう

目的：多様な実験設定で、既存GNNsの利点/欠点を調べる

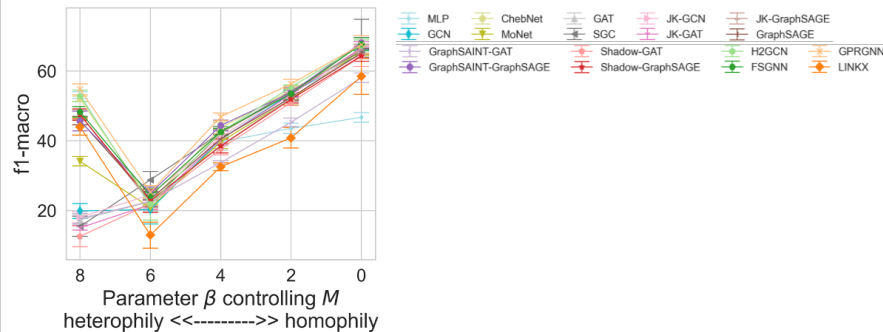
着想：うまく人工グラフを作れば、有益な比較実験ができないか？ 🤔

手段：20以上の多様なグラフと17種類のGNNsを使って、実験しよう！

所望の人工グラフを作成



特性がわかっているグラフでGNNsを評価！



- 実験を通して、既存研究の問題点と改善余地を指摘
- 人工グラフの作成と17種類のGNNsが動作する評価フレームワークを公開

[9] “Beyond Real-world Benchmark Datasets: An Empirical Study of Node Classification with GNNs”, Maekawa et al., NeurIPS Datasets & Benchmarks 2022.

[10] “GenCAT: Generating Attributed Graphs with Controlled Relationships between Classes, Attributes, and Topology.”, Maekawa et al., Information Systems 2023.

## 最近のGNNの動向

- homophilous/non-homophilous

- スケーラビリティ

↑ ここまでで説明済み

- 
- 小規模教師データ

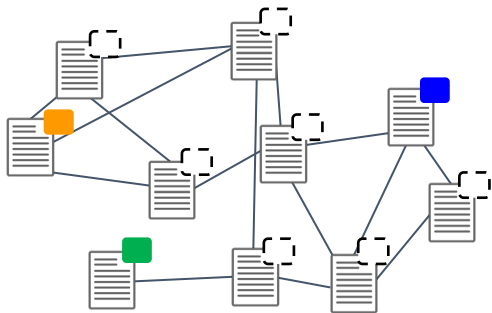
↓ この章で説明

- Neural architecture search

- 時系列グラフ

## 小規模教師データ

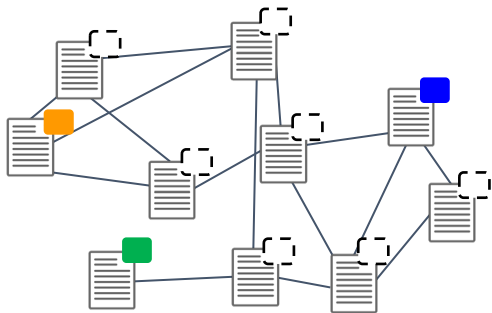
各クラスが少数しかラベルを持っていない場合、  
モデルは教師データに過学習してしまう



ラベルが少ない...

## 小規模教師データ

各クラスが少数しかラベルを持っていない場合、  
モデルは教師データに過学習してしまう



ラベルが少ない...

方針1： 畳込みの伝播回数を増やして、できるかぎり多くの学習データを embedding に使えるようにしよう！ [★11]

方針2： まずラベルなしデータで self-supervised learning をしてから、分類器を学習しよう！（contrastive learning が大流行中）

# Neural Architecture Search (NAS)



GNNの種類が多すぎてどれを選べばよいかわからない。

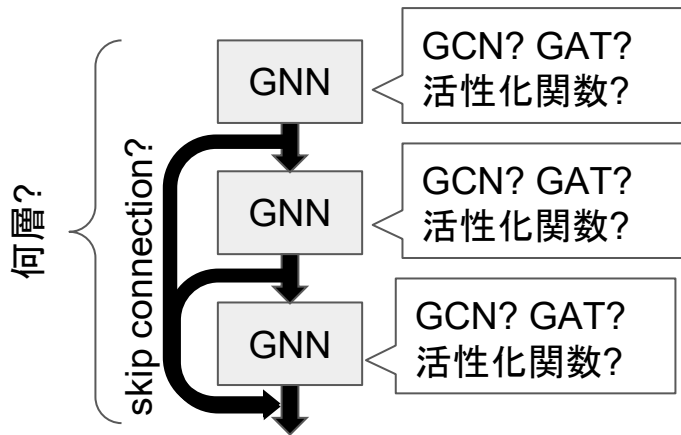


簡単なモデルを組み合わせるだけではだめなんやろか？

# Neural Architecture Search (NAS)

- 🙄 GNNの種類が多すぎてどれを選べばよいかわからない。
- 🤔 簡単なモデルを組み合わせるだけではだめなんやろか？

モデル組合せの例



研究1：最適/汎用的なモデルの探索空間を考える。

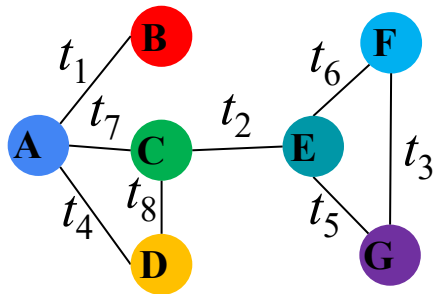
研究2：膨大なモデルの中から良さそうなモデルを優先的に探索する。

強化学習, 遺伝的アルゴリズム, etc

# 時系列グラフ

多くの現象は時系列の特徴を持っている

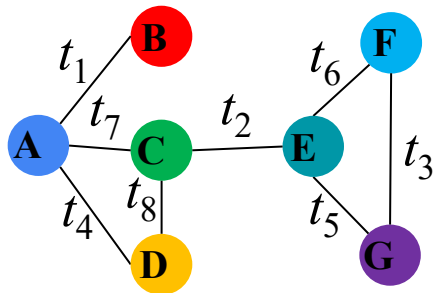
ex. 友達申請、メール、引用、神経回路の発火





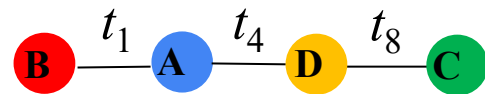
# 時系列グラフ

多くの現象は時系列の特徴を持っている  
 ex. 友達申請、メール、引用、神経回路の発火



方針1：グラフ畳込みを時系列に拡張しよう！

方針2：時系列による遷移情報を活用しよう！

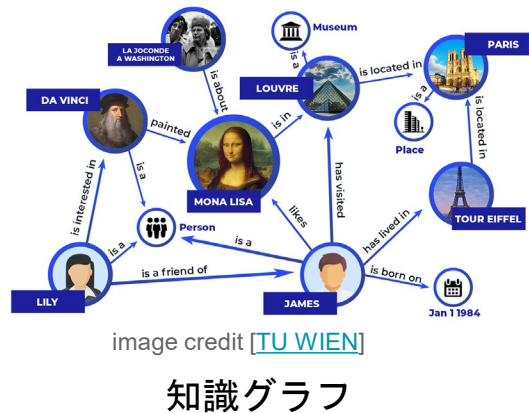
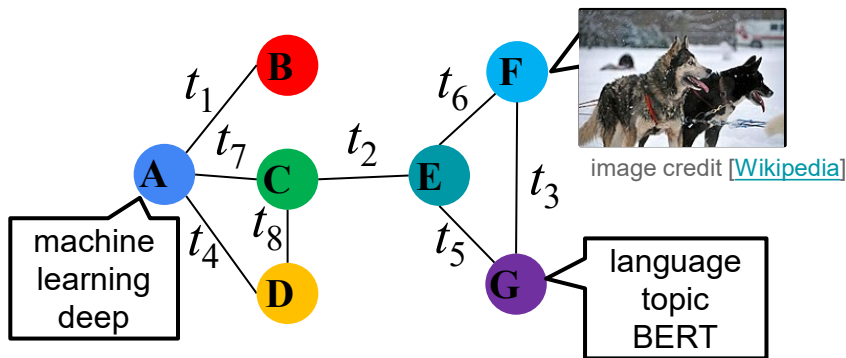


時系列遷移の例

## (発表者が思う) グラフ深層学習の展望

- マルチモーダル設定への適用
- 知識グラフでの分析
- 汎用事前学習（転移学習）モデル
- 生データから適切にグラフを作成

# マルチモーダル設定・知識グラフでの分析



シンプルなグラフだけでなく、リッチな情報を扱うことで実用的なタスクを解けるようになっていく！（はず）

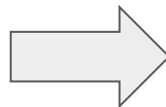
# 汎用事前学習（転移学習）モデル

BERTのような...

- 様々なタスクにfine-tuneで対応可能
- 様々なデータドメインを考慮

実現できれば

- 計算時間やリソースを節約
- 小規模な学習データでも高精度が期待



実応用につながる！

実現への課題

- 様々なタスクに利用できる特徴量の抽出方法は自明ではない
- データセットを横断した共通的なノード属性はない

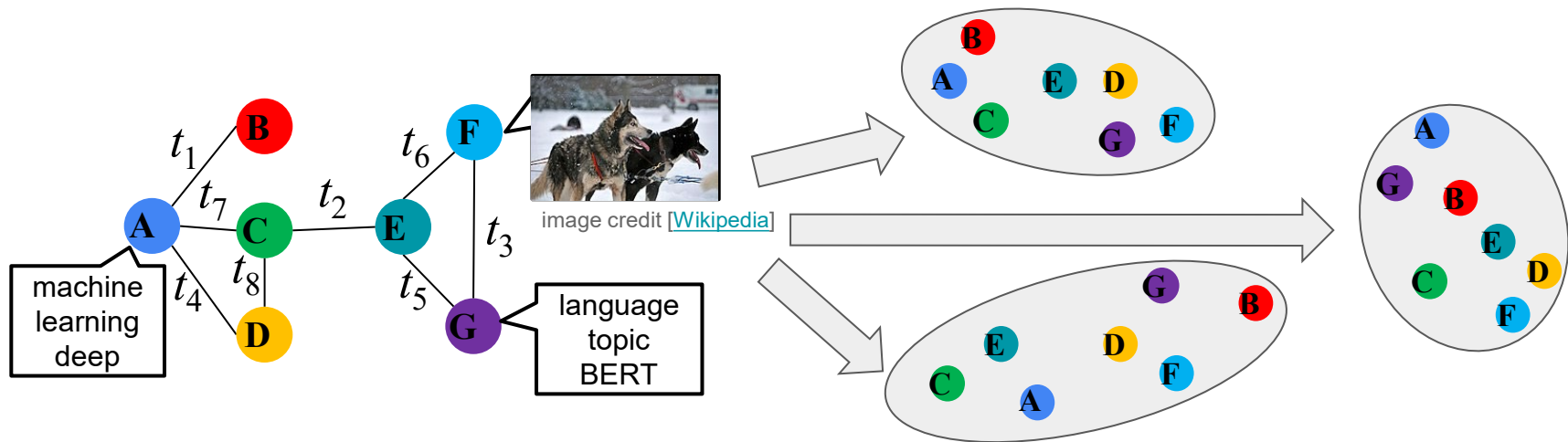


# 汎用事前学習（転移学習）モデル

## 展望

様々な側面で区別可能なembeddingを作成

- 各embedding spaceで、ノードは区別可能
  - ノード区別可能性を失わない埋込方法が複数必要
- 下流タスクではfine-tuneによって、必要な次元を選択



# 生データから適切にグラフを作成

グラフは多くの事象を正確にモデリングできる一方で、その自由度のために統一されたフォーマットにすることが困難

例えば、知識グラフでの関係は能動態と受動態で方向が逆転する

時系列情報の時間の単位も、データセットによって異なる  
(年, 月, 週, 日, 時, 秒)

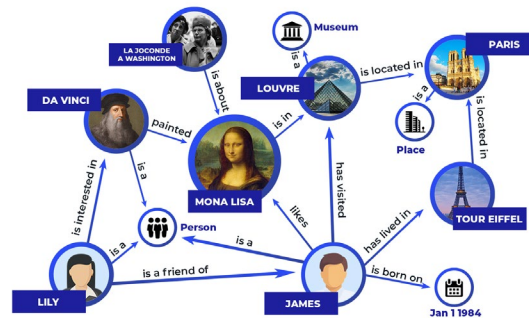


image credit [TU WIEN](#)

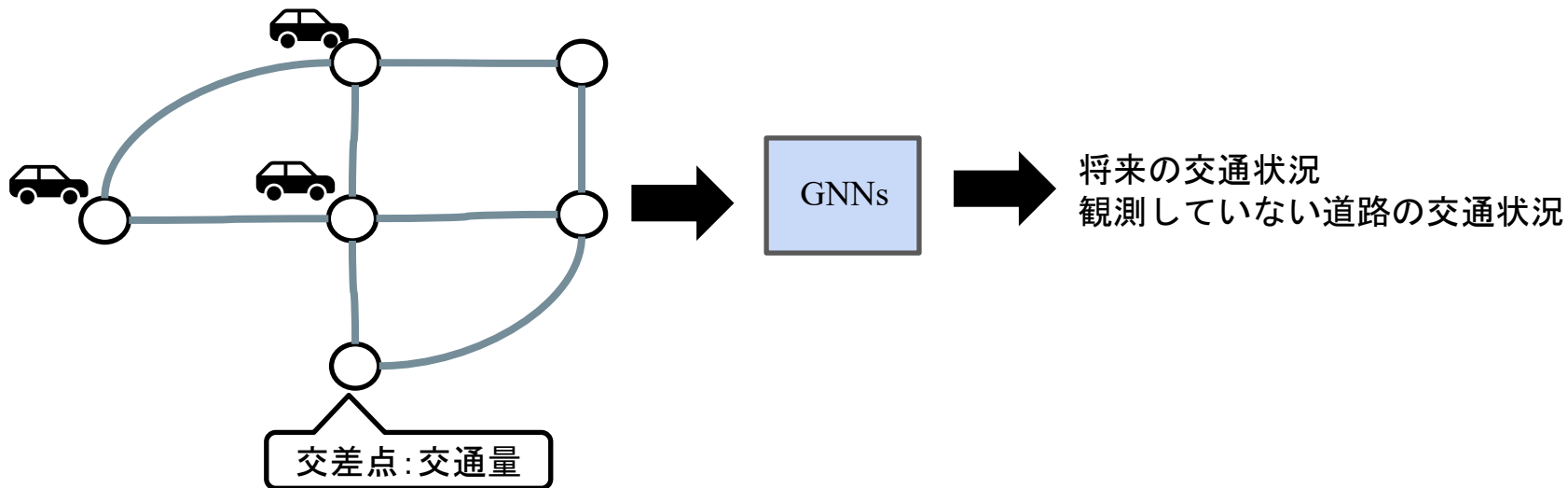
汎用モデルの実現には、生データに正規化/標準化を施し、統一的に利用できるようにする必要がある

## 応用分野

- 都市交通・都市計画
- 化学・創薬・物質科学
- ニュースやアイテム推薦

## 都市交通・都市計画

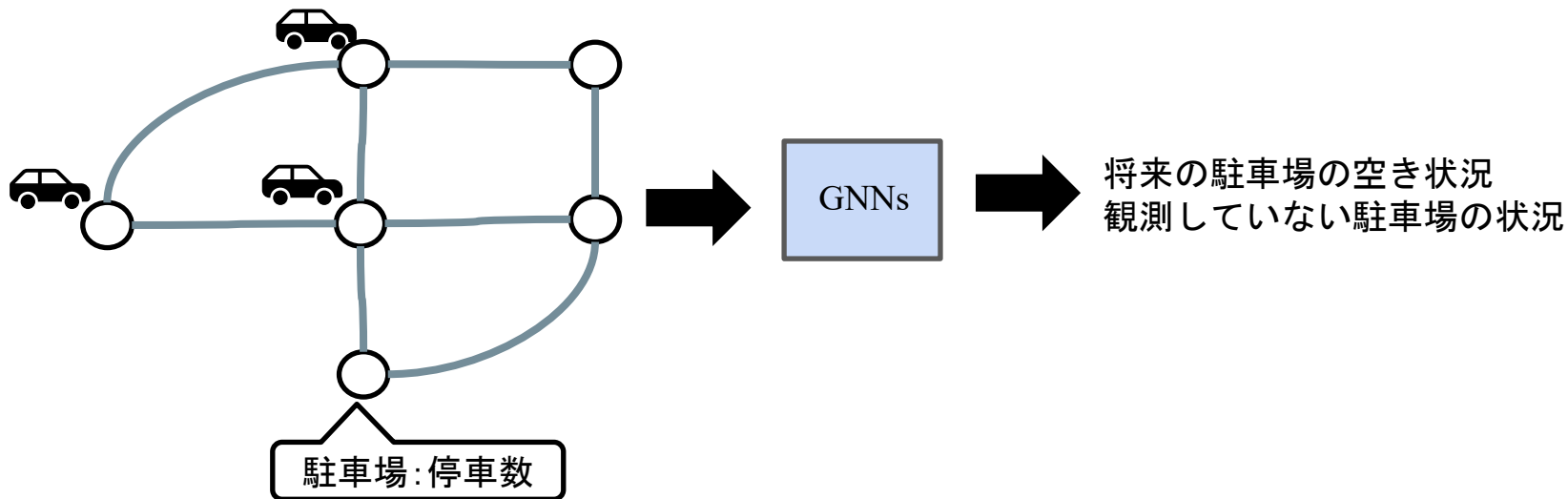
- 都市内のセンサ間の関連性をグラフで表現
  - 道路ネットワークやセンサ間の距離で枝を決定





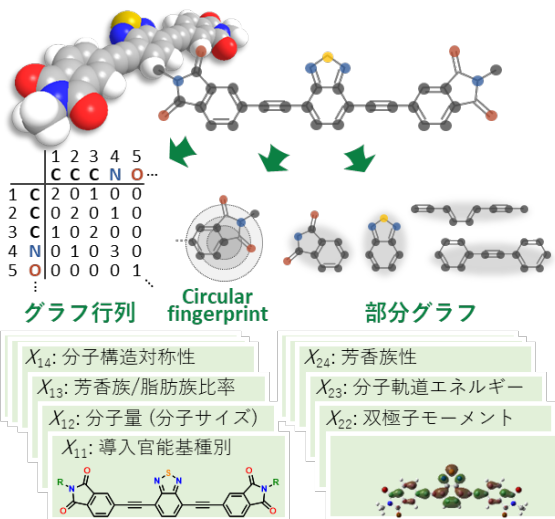
## 都市交通・都市計画

- 都市内のセンサ間の関連性をグラフで表現
  - 道路ネットワークやセンサ間の距離で枝を決定

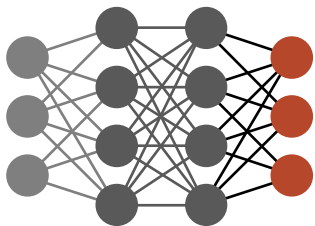


# 化学・創薬・物質科学

- 分子構造をグラフ化し，有用な物性をもつ物質を効率的に探索

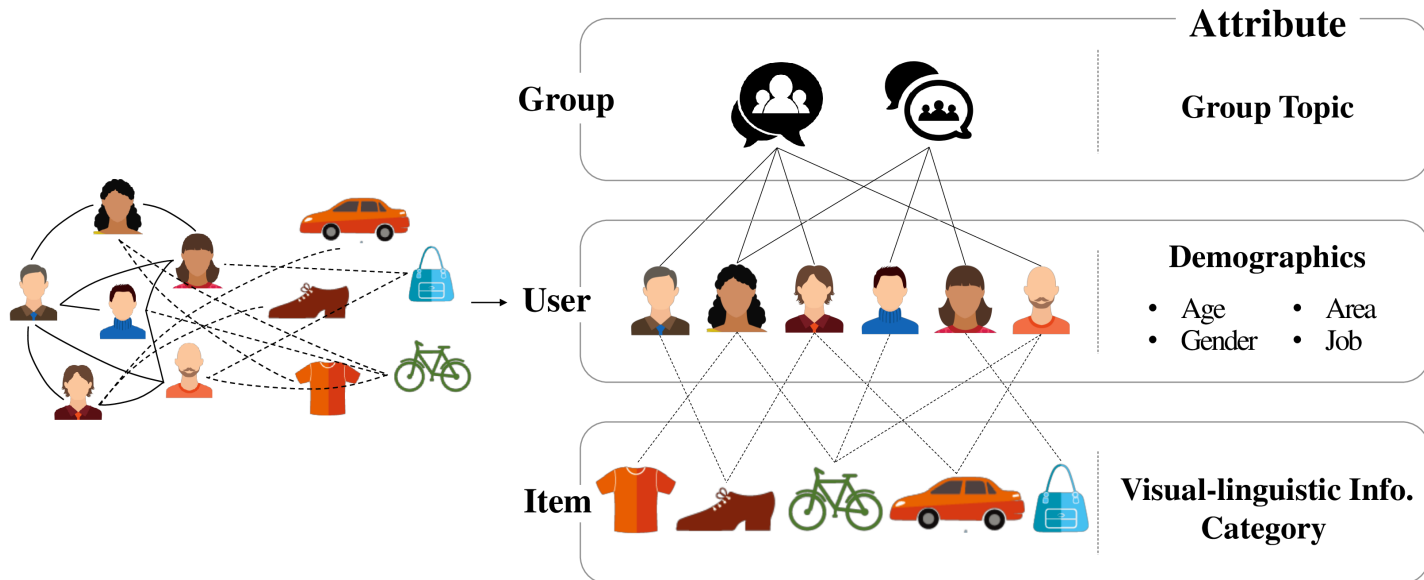


分子設計・知識抽出



# ニュースやアイテム推薦

- ユーザとニュース/アイテムの好み・購買などをグラフ化
- リンク予測により好きなもの・次に購入するものなどを推薦



# 皆さん、どうGNNを勉強しますか？

論文を読む？論文だけでは実装に不十分なことが多い！

Githubのコードを動かす？理論がわからない！

ウェブ記事をあさる？誤情報が結構あります...

グラフ道場

GCN

GAT

SGC

表記



## はじめに

グラフ道場は、[グラフニューラルネットワーク](#)の基礎を実践的に学び、自身で実装し、発展モデルを設計する手助けすることを目指しています。

グラフ道場で学べること

# GNN学習用Webサイト グラフ道場をオープン!

- [GNNの歴史や発展の流れ](#)
- [Deep Graph Library](#)や[Pytorch geometric](#)を活用したモデル構築（そのうち追加するかもしれません）

グラフ道場はオープンソース・プロジェクトは、[クリエイティブ・コモンズ 表示 - 非営利 - 改変禁止 4.0 国際 \(CC BY-NC-ND 4.0\)](#)（プログラム部分以外）および[MITライセンス](#)（プログラム部分）で公開されているオープンソース・プロジェクトです。不具合報告はGitHubの[issues](#)までお願いします。

次へ  
GCN >

# グラフ道場のポイント

数式と  
イメージ図で  
理論を説明

Colabで  
実装も公開

日本語!  
(そのうち  
英語化  
するかも)

## GCNの理論

GCNは下記の式で表すことができます。

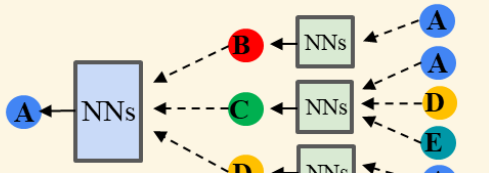
$$\mathbf{H}^{(l+1)} = \sigma(\mathbf{D}'^{-\frac{1}{2}} \mathbf{S}' \mathbf{D}'^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)})$$

ここで、 $\mathbf{H}^{(0)} = \mathbf{X}$ になります。

あるひとつの節点 $v$ の特徴量で考えると下記になります。

$$H_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}_v} \frac{1}{\sqrt{d_u d_v}} H_u \mathbf{W}^{(l)}\right)$$

GCNのイメージは周りにEmbeddingを集めてNNを通してEmbeddingを生成して、またのGCNは2層から構成されていて、経験的にも2層が最適になることが多いです。



GAT.ipynb ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ すべての変更を保存しました

+ コード + テキスト

```

super(GraphAttentionLayer, self).__init__()
self.dropout = dropout
self.in_features = in_features
self.out_features = out_features
self.alpha = alpha
self.final = final

self.W = nn.Parameter(torch.empty(size=(in_features, out_features)))
nn.init.xavier_uniform_(self.W.data, gain=1.414) # Xavierの方法により初期化
self.a = nn.Parameter(torch.empty(size=(2*out_features, 1)))
nn.init.xavier_uniform_(self.a.data, gain=1.414) # Xavierの方法により初期化

self.leakyrelu = nn.LeakyReLU(self.alpha) # Attention計算用のLeakyReLU

def forward(self, h, adj): #hはNNへの入力行列
    Wh = torch.mm(h, self.W)
    e = self._prepare_attentional_mechanism_input(Wh)

    zero_vec = -9e15*torch.ones_like(e)
    attention = torch.where(adj > 0, e, zero_vec) # 枝がないところは attention = 0
    attention = F.softmax(attention, dim=1) # dim=1は行単位でのsoftmax
  
```

## まとめ

グラフ分析がどんどん熱い分野になってきていることを共有する

グラフ深層学習の気持ちを伝える

グラフ分析の応用例を伝える

# 年表のReferences

- GCN: “Semi-Supervised Classification with Graph Convolutional Networks”, Thomas N. Kipf, Max Welling, ICLR 2017.
- VGAE: “Variational Graph Auto-Encoders”, Thomas N. Kipf, Max Welling, NeurIPS 2016.
- GraphSAGE: “Inductive representation learning on large graphs.”, Hamilton et al., NeurIPS 2017.
- GAT: “Graph Attention Networks”, Veličković et al., ICLR 2018.
- JKNet: “Representation Learning on Graphs with Jumping Knowledge Networks”, Xu et al., ICML 2018.
- GIN: “How Powerful are Graph Neural Networks?”, Xu et al., NeurIPS 2019.
- APPNP: “Predict then Propagate: Graph Neural Networks meet Personalized PageRank.”, Gasteiger et al., ICLR 2019.
- SGC: “Simplifying Graph Convolutional Networks”, Wu et al., ICML 2019.
- MixHop: “MixHop: Higher-Order Graph Convolution Architectures via Sparsified Neighborhood Mixing”, Abu-El-Haija et al., ICML 2019.
- GraphSAINT: “GraphSAINT: Graph Sampling based Inductive Learning Method”, Zeng et al., ICLR 2020.
- GCN2: “Simple and Deep Graph Convolutional Networks”, Chen et al., ICML 2020.
- Geom-GCN: “Geom-GCN: Geometric Graph Convolutional Networks”, Pei et al., ICLR 2020.
- H2GCN: “Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs”, Zhu et al., NeurIPS 2020.
- ANEPN: “Adaptive Node Embedding Propagation for Semi-Supervised Classification.”, Ogawa et al., ECML PKDD 2021.
- FSGNN: “Improving Graph Neural Networks with Simple Architecture Design.”, Maurya et al., arxiv 2021.
- LINKX: “Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods.”, Lim et al., NeurIPS 2021.
- GPRGNN: “Adaptive Universal Generalized PageRank Graph Neural Network”, Chien et al., ICLR 2021.
- WRGNN: “Breaking the Limit of Graph Neural Networks by Improving the Assortativity of Graphs with Local Mixing Patterns.”, Suresh et al., KDD 2021.
- ShaDow-GNN: “Decoupling the Depth and Scope of Graph Neural Networks.”, Zeng et al., NeurIPS 2021.
- GCOND: “Condensing Graphs via One-Step Gradient Matching.”, Jin et al., KDD 2022.
- LC Trans.: “GNN Transformation Framework for Improving Efficiency and Scalability”, Maekawa et al., ECML PKDD 2022.
- GGCN: “Two Sides of the Same Coin: Heterophily and Oversmoothing in Graph Convolutional Neural Networks”, Yan et al., ICDM 2022.
- DualNet-GNN: “Not All Neighbors are Friendly: Learning to Choose Hop Features to Improve Node Classification”, Maurya et al., CIKM 2022.
- ACM-GCN: “Revisiting Heterophily For Graph Neural Networks”, Luan et al., NeurIPS 2022.
- GloGNN: “Finding Global Homophily in Graph Neural Networks When Meeting Heterophily”, Li et al., ICML 2022.