ICEPP
The University of Tokyo

グラフニューラルネットワークを用いた事象選別

解析技術 + ILC懇談会 (第一回勉強会) @ ONLINE

2021 / 7 / 20

東京大学素粒子物理国際研究センター (ICEPP)

齊藤真彦

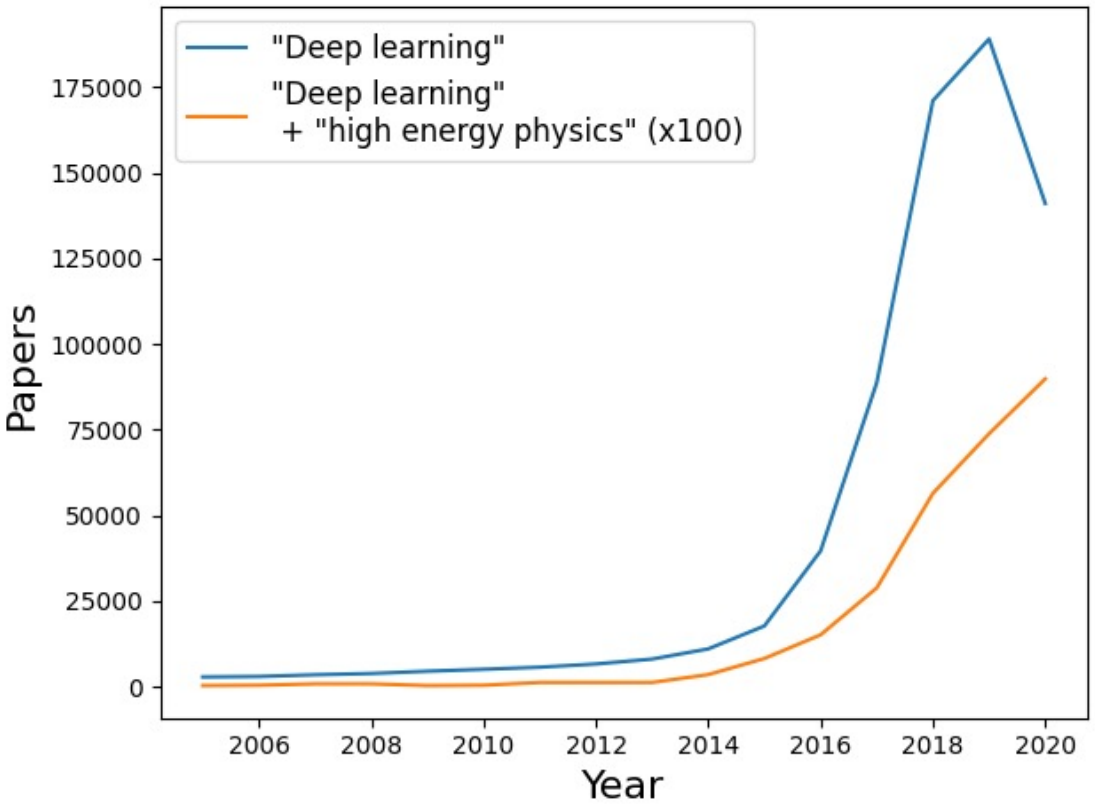
Outline

- グラフニューラルネットワークとは何か？
  - 代表的な深層学習モデルとの対比
  - 具体的な計算方法
- グラフニューラルネットワークを用いたイベント分類
- グラフニューラルネットワークを使った他の研究例

HEPと深層学習

- 深層学習のブームは継続中
- 高エネルギー分野においても応用例が増加し続けている
  - HEPML-LivingReview: ~ 500 papers
- 様々なタイプの深層学習モデルが提案されている。

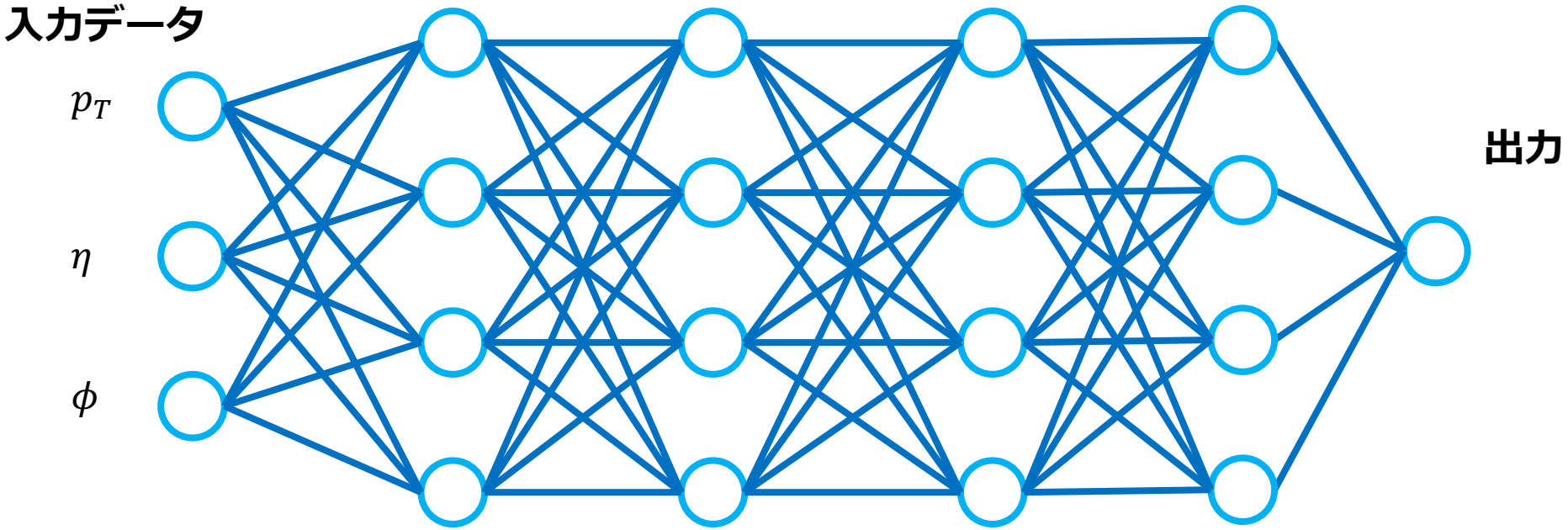
Google Scholarでのキーワード検索結果



代表的な深層学習モデル

Multi-layer perceptron (MLP)

- Fully Connected (FC), Deep neural network (DNN), Dense とも呼ばれる

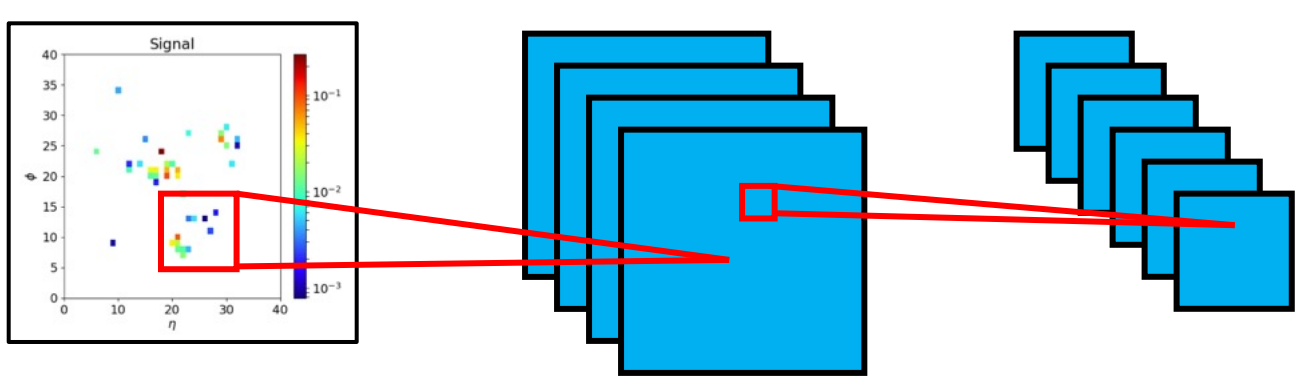


- 最も基本的な深層学習モデル
- 全てのノードを結合させている
  - 強力な表現(近似)能力
  - 非常に多くのパラメータが必要
- 高レベルの特徴量を入力とすることもあるが、より低次元な入力でも良い性能がでる

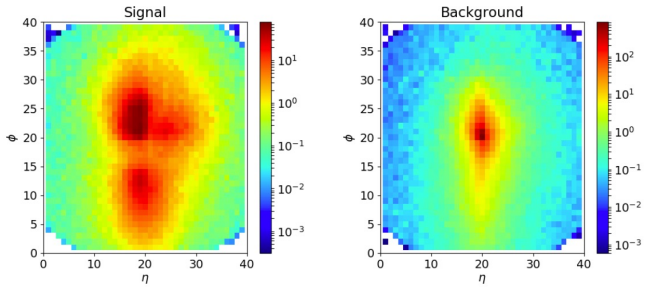
代表的な深層学習モデル

Convolutional neural network (CNN)

入力データ



Top-tagging
(SciPost Phys, 7, 014 (2019))

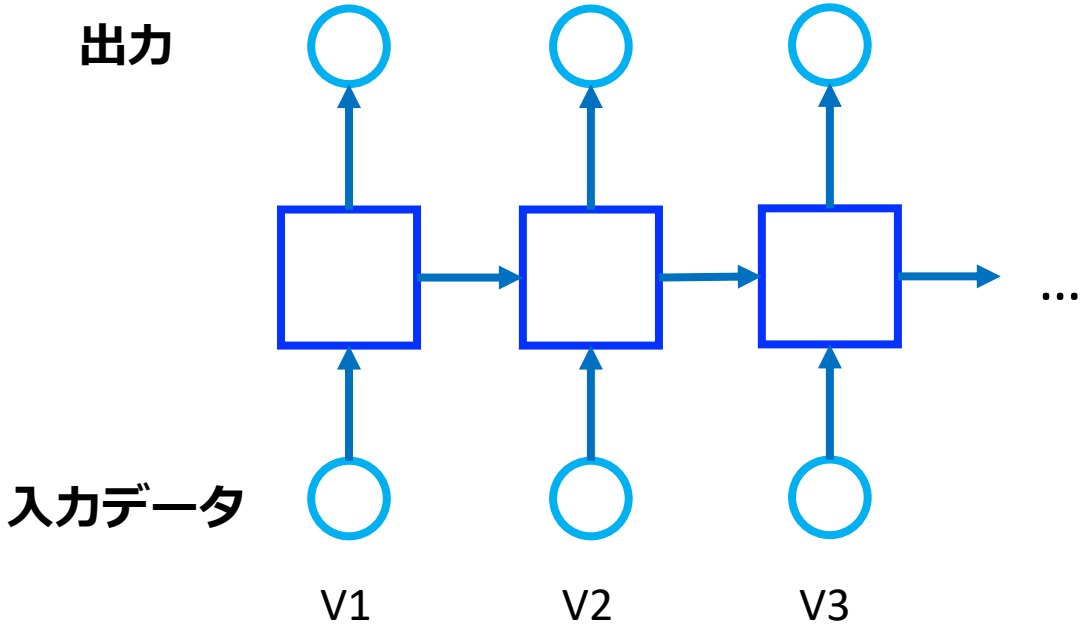


- 画像認識で用いられる深層学習モデル
- 小さいフィルターで畳み込み積分をすることで
  - 平行移動に対して頑健
  - 近隣ピクセル同士の関係を効率良く学習
- 例: カロリメータでのエネルギー分布を画像としてCNNで処理

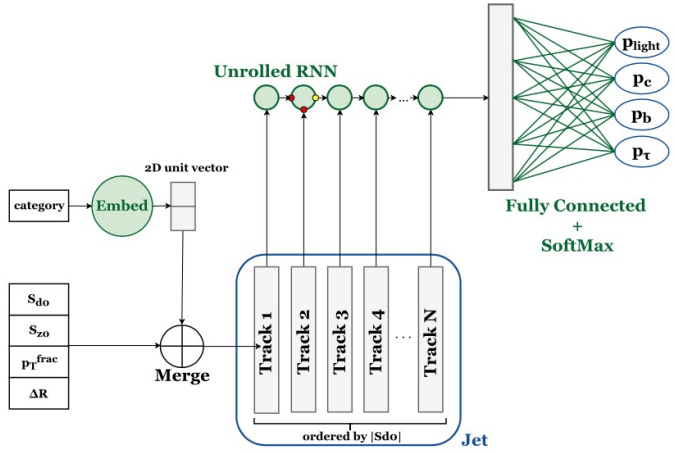
代表的な深層学習モデル

Recurrent neural network (RNN)

- LSTM, GRU が有名



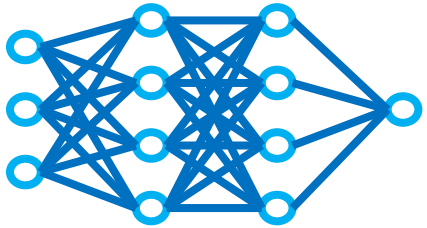
RNN b-tagging
(ATL-PHYS-PUB-2017-003)



- 時系列データや自然言語処理でよく用いられる
- 出力を再帰的に扱うことにより、
  - 系列データ・可変な入力を扱える
- 例: ジェット中のトラックなど、入力変数の数が変化するものに適用

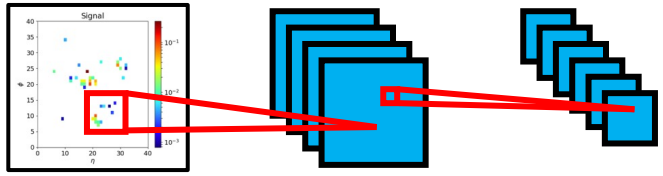
代表的な深層学習モデル

Multilayer Perceptron(MLP)



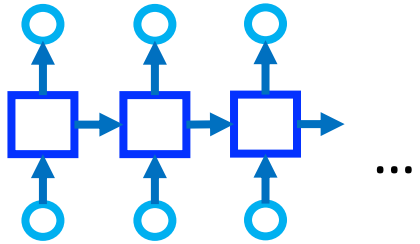
- １次元データ
- 全変数の相関

Convolutional NN (CNN)



- グリッド状データ
- 局所的な相関
- 位置の異なるデータに同じ操作

Recurrent NN(RNN)



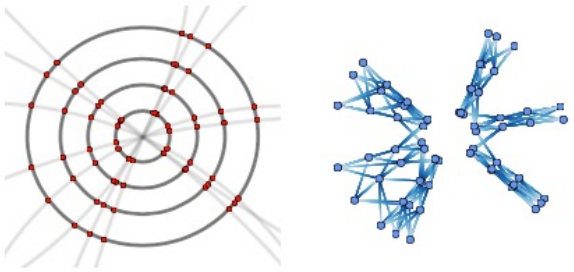
- 系列データ
- 時間的な相関
- 時間軸の異なるデータに同じ操作

Q: どのような深層学習モデルを使うべきか？
A: 深層学習モデルの構造は、対象とするタスクの構造(ドメイン知識)を反映させるのが良い
  - モデルパラメータの削減、過学習の抑制

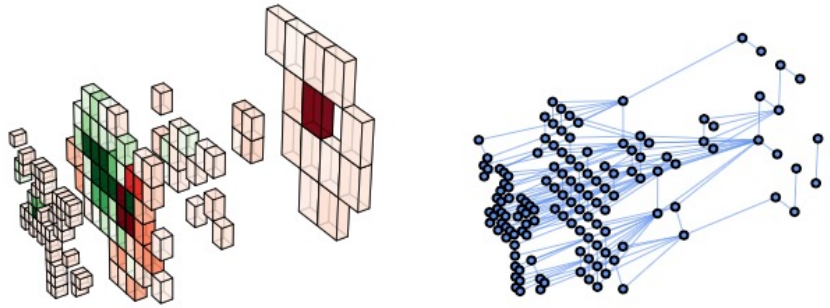
- HEPで扱う使うデータの構造は、**グラフ** と相性が良いことがある

HEPにおけるグラフ構造

[2007.13681]



(a) Tracking



(b) Calorimeter clustering

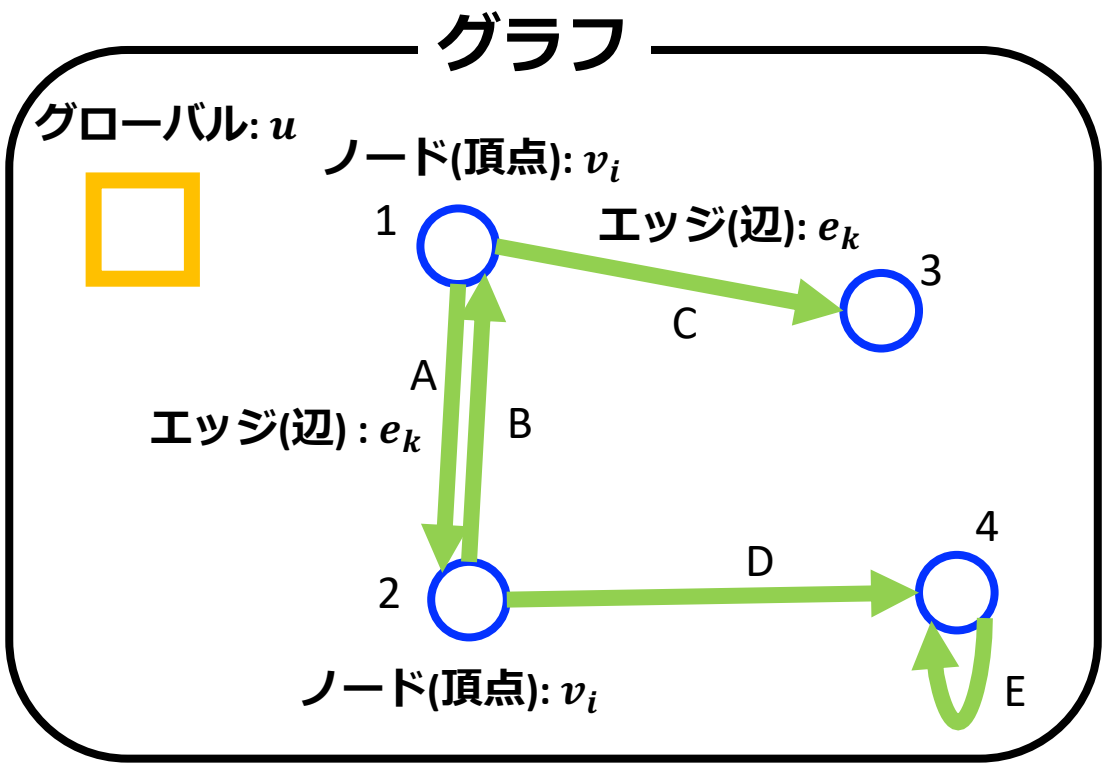
(c) Event classification

(d) Jet classification

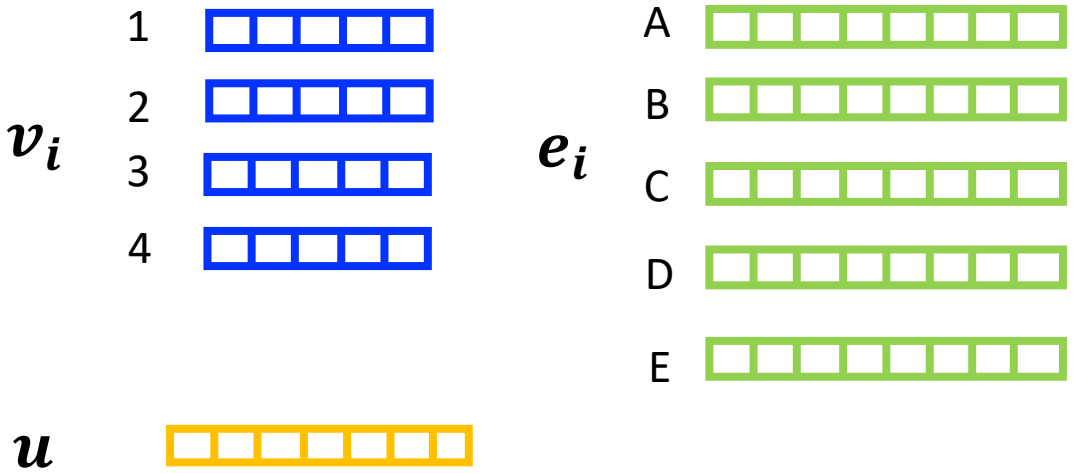
- グラフとして扱うと良いデータ構造の性質
  - 扱う対象の数が変化する
  - 扱う対象に順序がない(集合)
  - モノ同士の関係性が定義できる
- グラフを入力として扱う深層学習モデルとして、**グラフニューラルネットワーク(GNN)** がある

グラフネットワーク概要

1806.01261

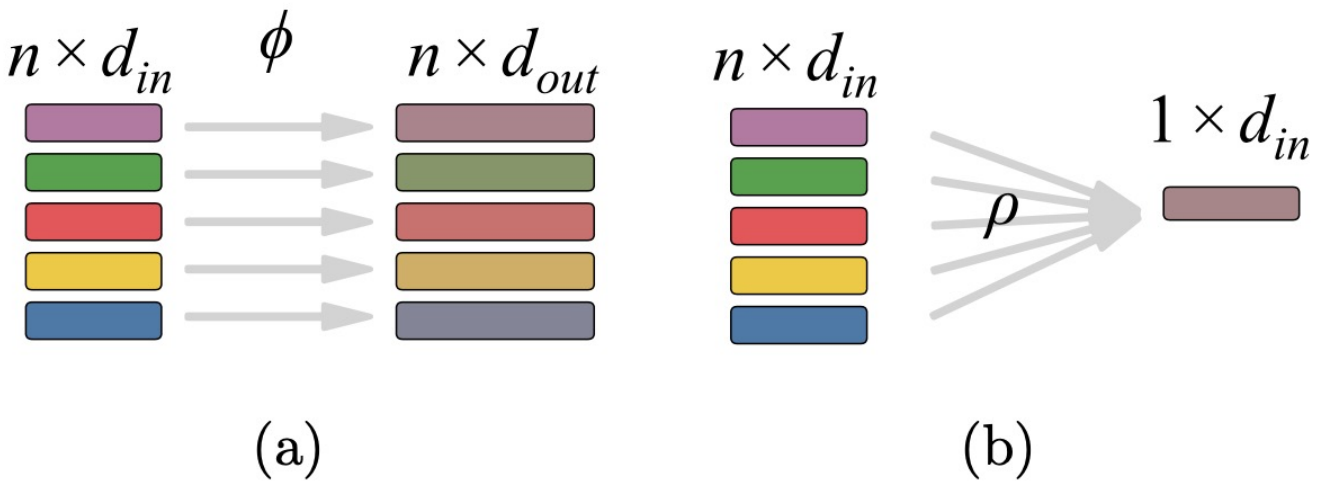


ノード・エッジごとの特徴量



- グラフはノード(頂点)とエッジ(辺)から構成される
- それぞれのノード・エッジは特徴量(attributes)を持つ
- グラフ(ノード・エッジ)の特徴量を逐次的にアップデートすることで出力値を計算

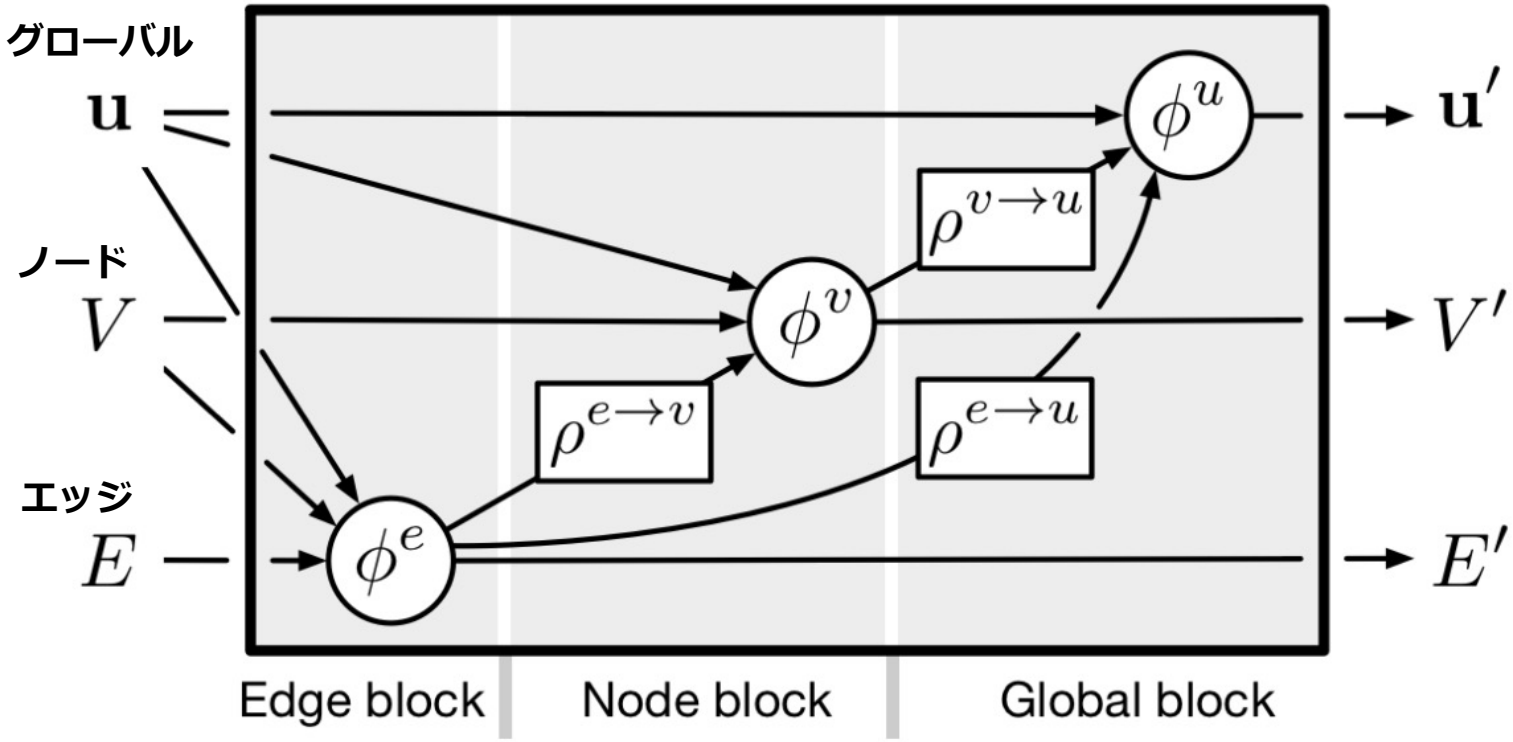
グラフネットワーク具体的な計算方法 1806.01261 2007.13681



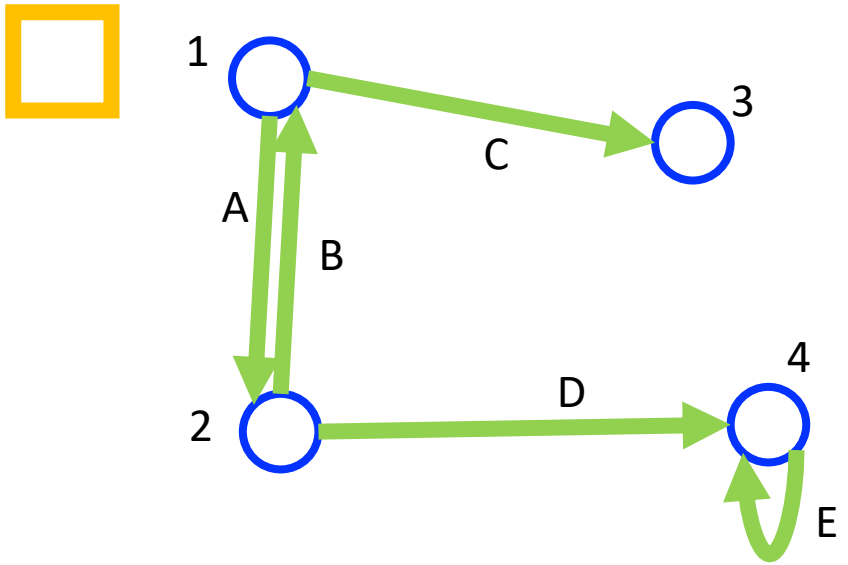
- Update function(ϕ):
  - ノード・エッジごとにattributeをアップデートする
  - 更新の関数としてはNN (MLP) がよく用いられる

- Aggregation function(ρ):
  - 隣接している or グラフ上のすべてのノード・エッジのattributeを集約する
  - 可変入力に対応できる関数である必要がある。
  - 例: sum, mean, max/min

グラフネットワーク具体的な計算方法 1806.01261

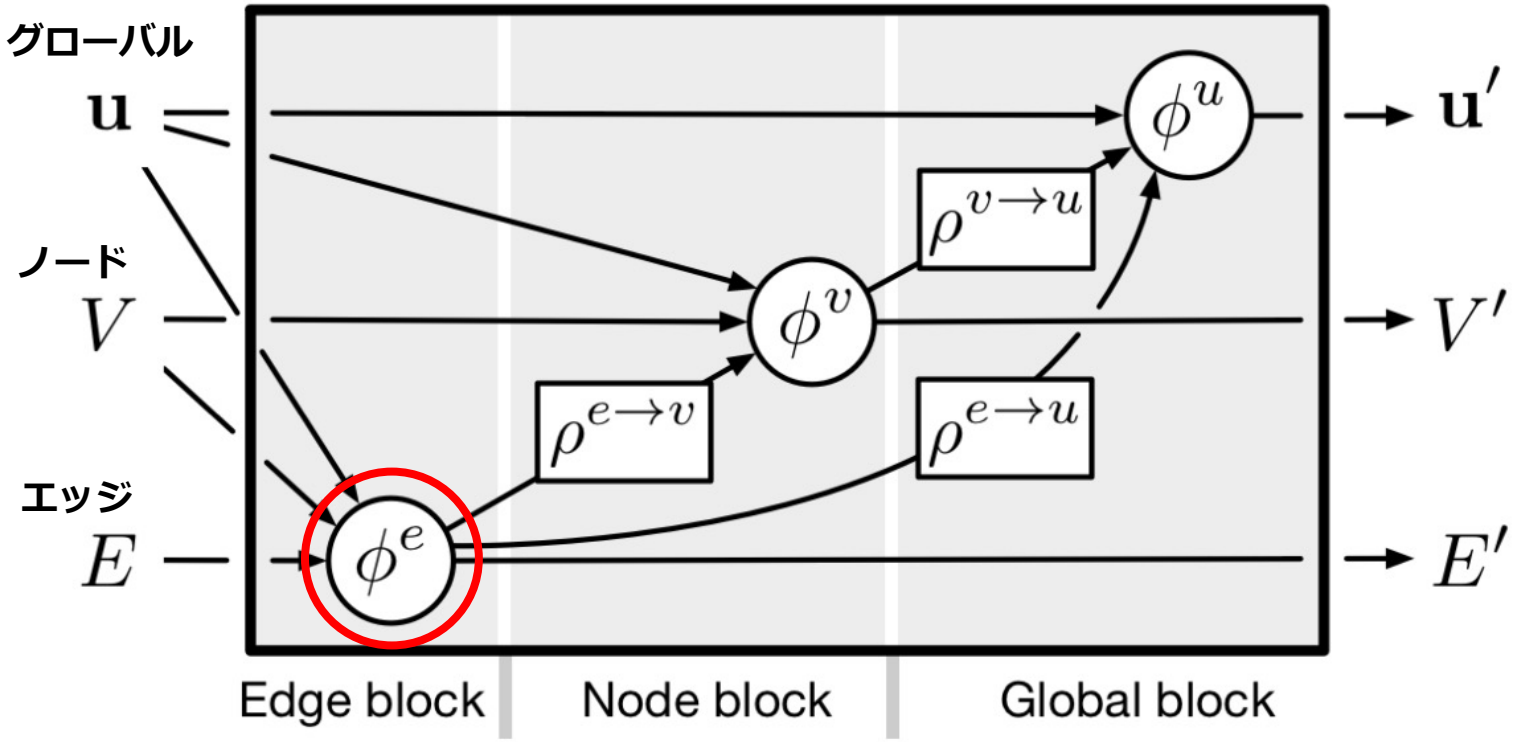


グラフの例

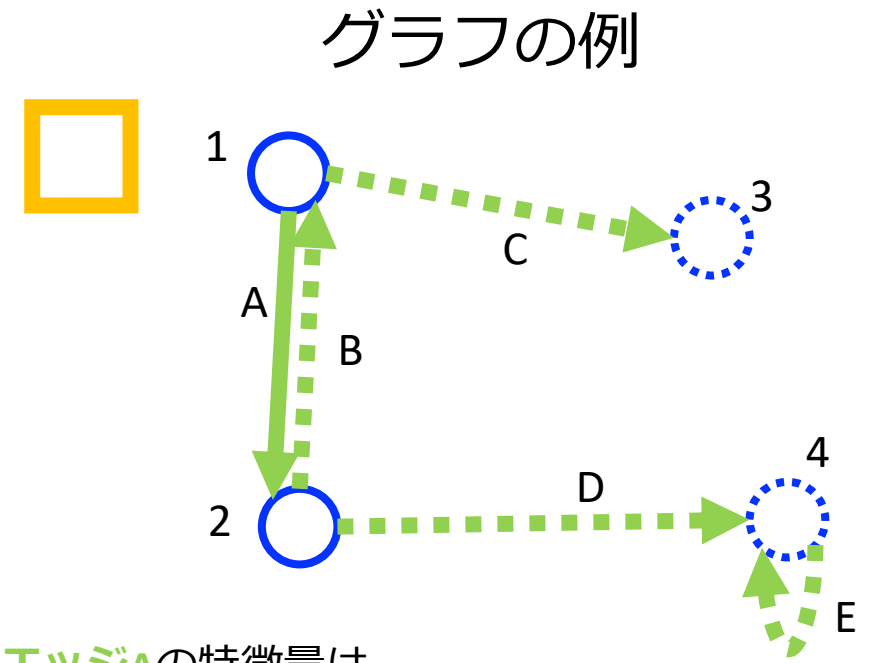


グラフネットワーク具体的な計算方法 1806.01261

1. エッジの更新



$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$$



エッジAの特徴量は

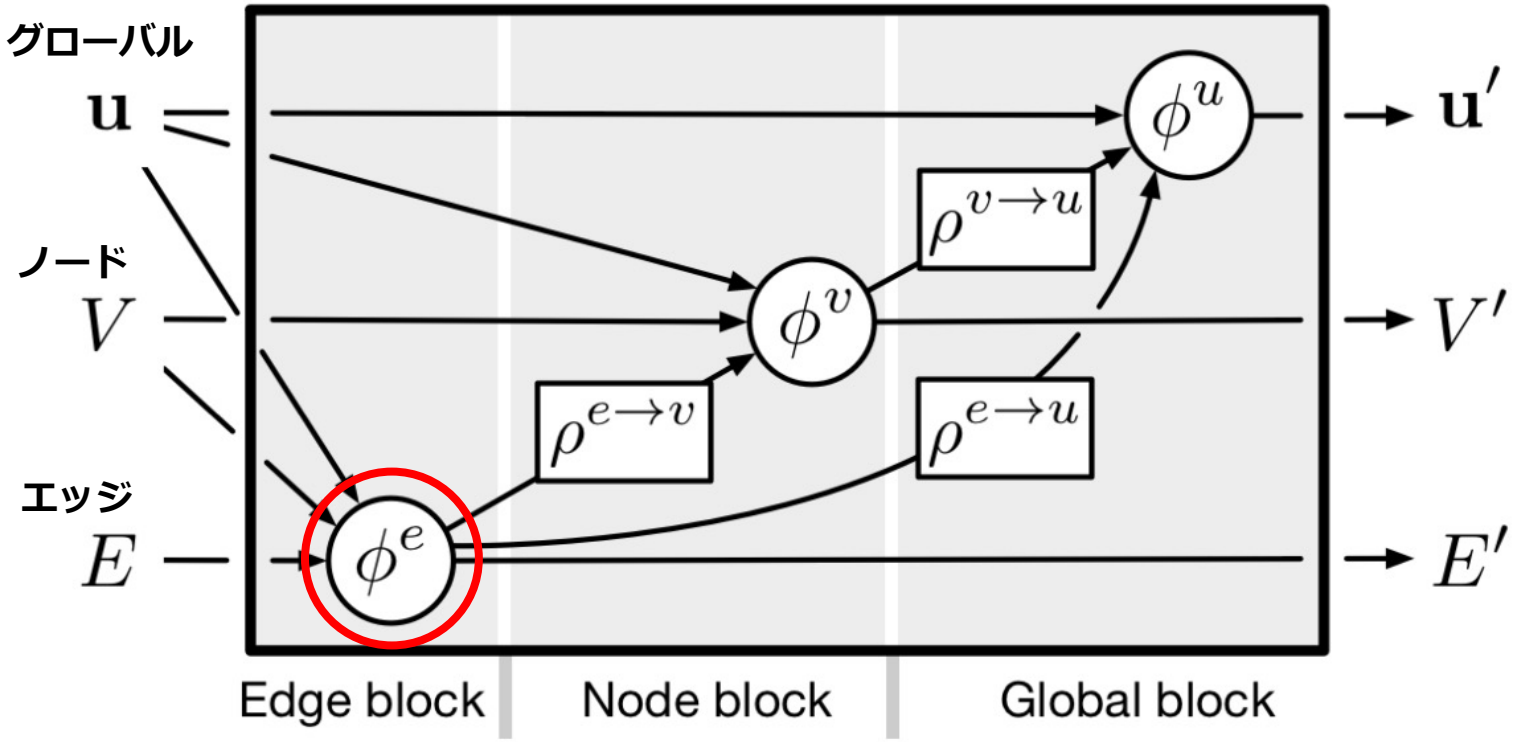
- エッジAの特徴量
- ノード1の特徴量
- ノード2の特徴量
- グローバル特徴量

を用いて更新される

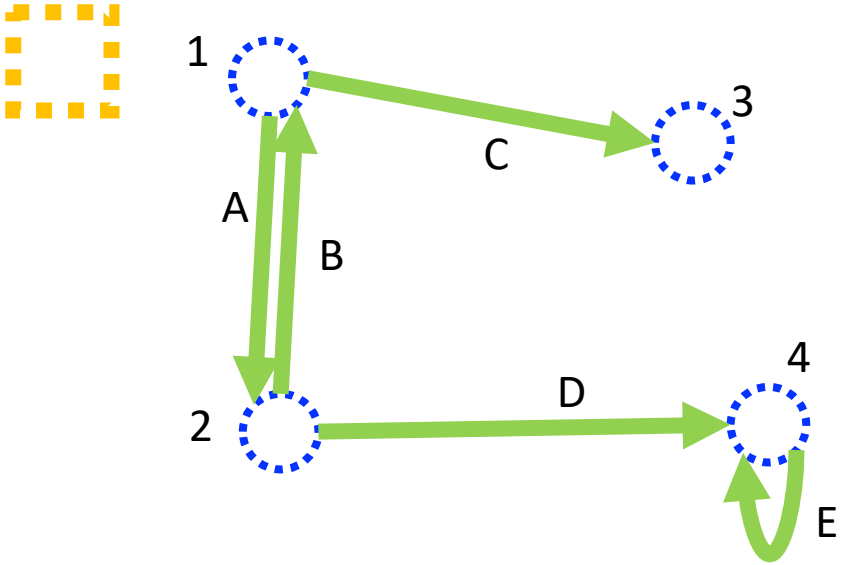
例: $e'_A = NN(Concatenate([e_A, v_1, v_2, u]))$

グラフネットワーク具体的な計算方法 1806.01261

1. エッジの更新



グラフの例



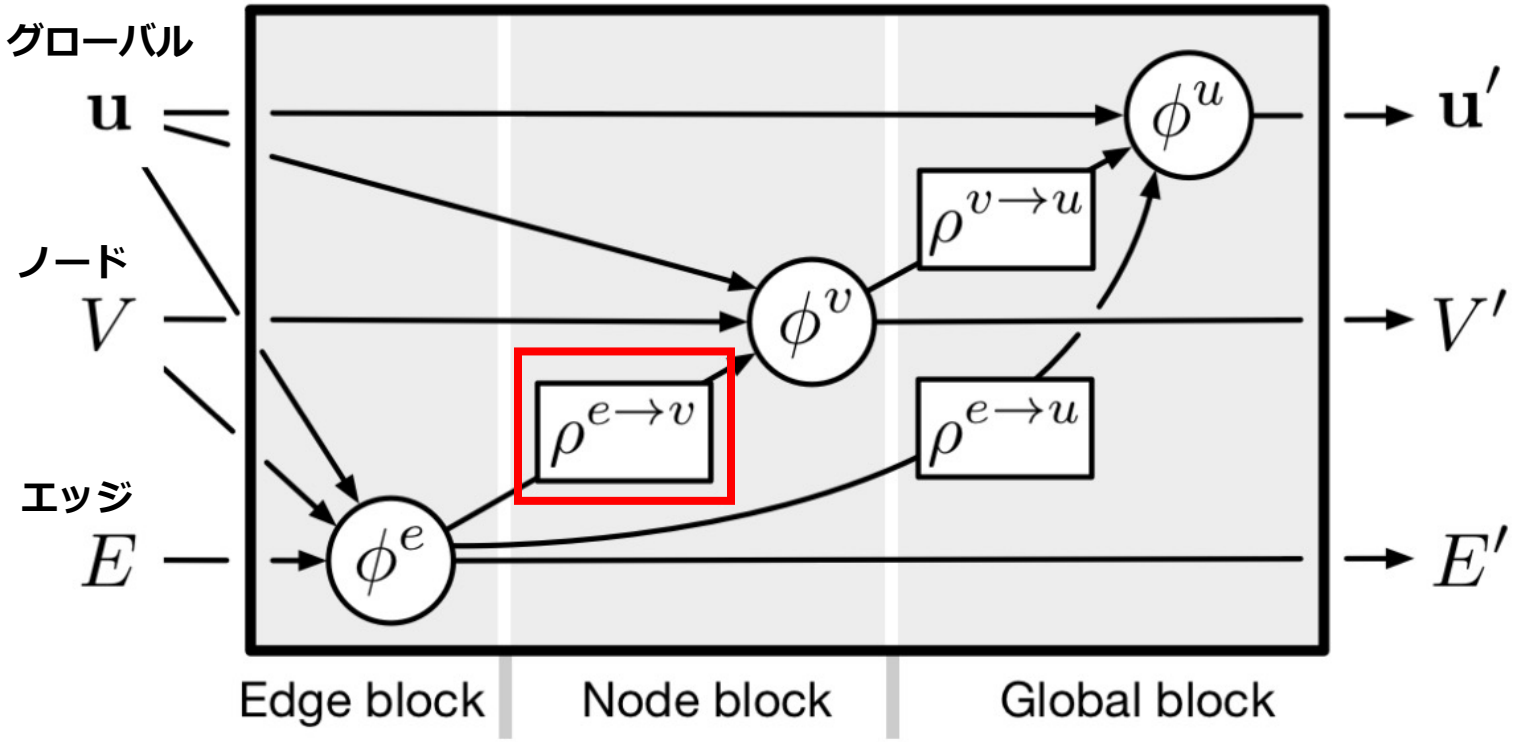
$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$$

すべてのエッジは同様に更新される

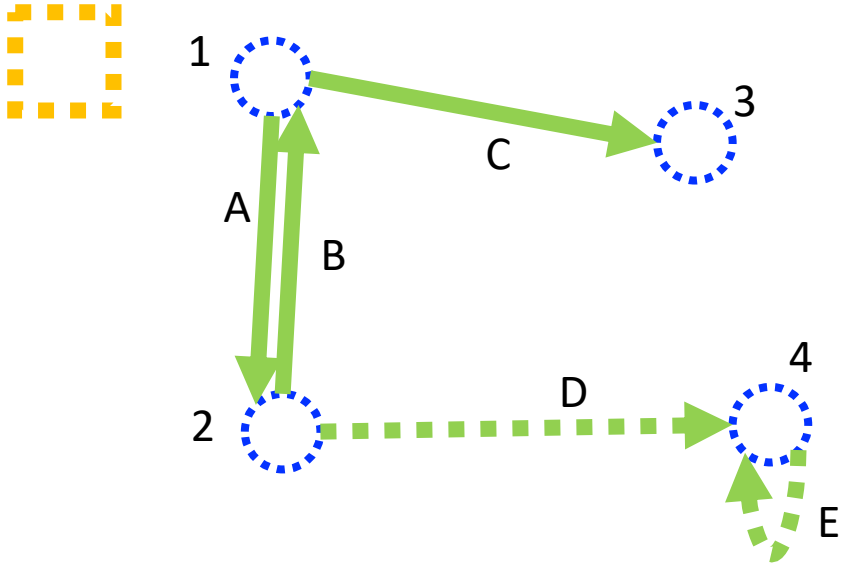
例: $e'_k = NN(Concatenate([e_k, v_i, v_j, u]))$

グラフネットワーク具体的な計算方法　1806.01261

2. ノードの更新



グラフの例



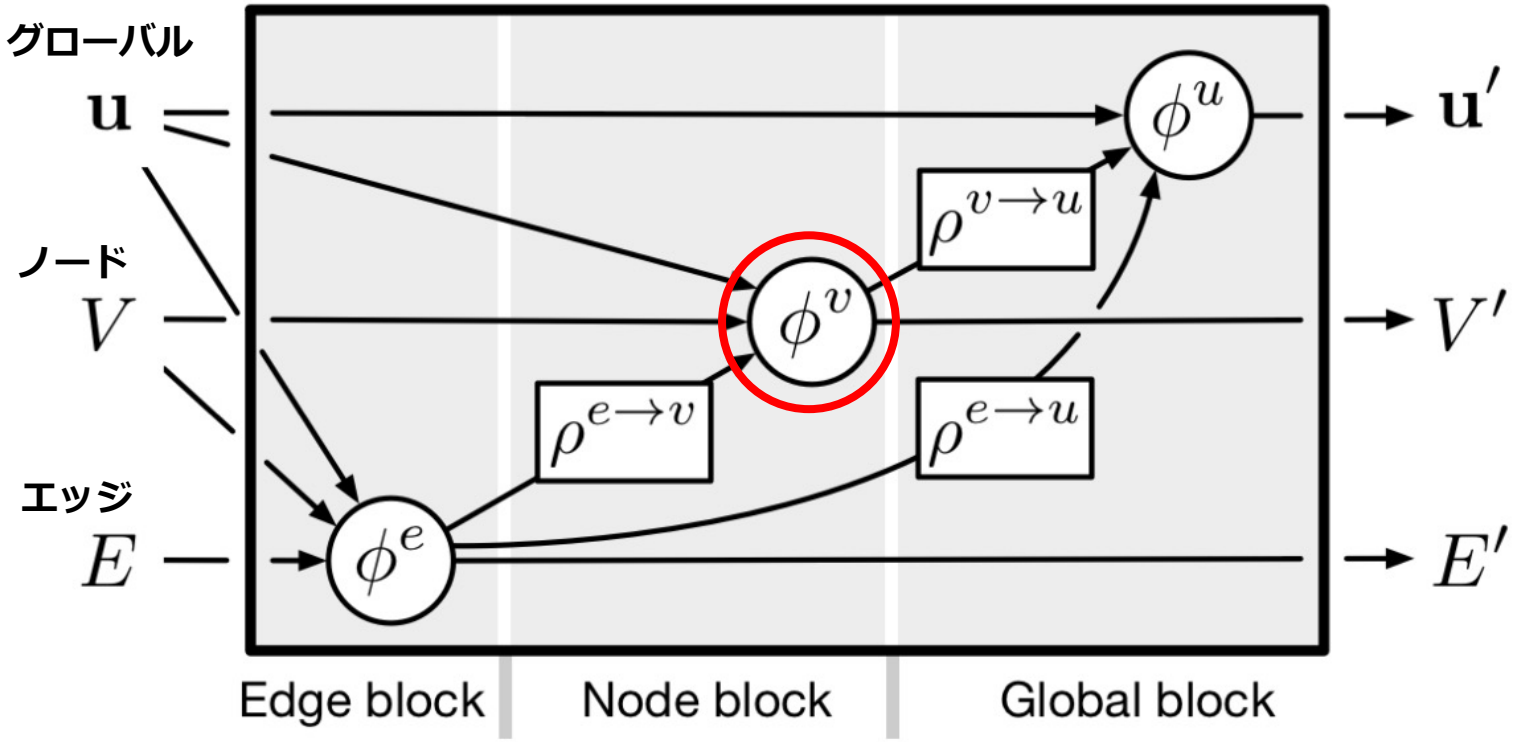
ノード1に隣接したエッジの特徴量を集約する

例: $\bar{e}'_1 = Sum([e_A, e_B, e_C])$

$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right) \qquad \bar{\mathbf{e}}'_i = \rho^{e\to v}\left(E'_i\right)$$

グラフネットワーク具体的な計算方法 1806.01261

2. ノードの更新

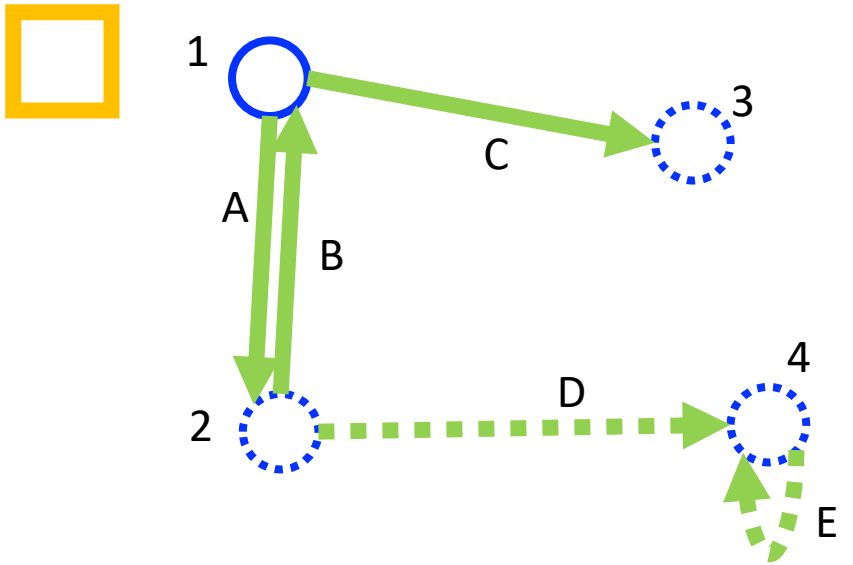


$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$$

$$\mathbf{v}'_i = \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right)$$

$$\bar{\mathbf{e}}'_i = \rho^{e\to v}\left(E'_i\right)$$

グラフの例



ノード1の特徴量は

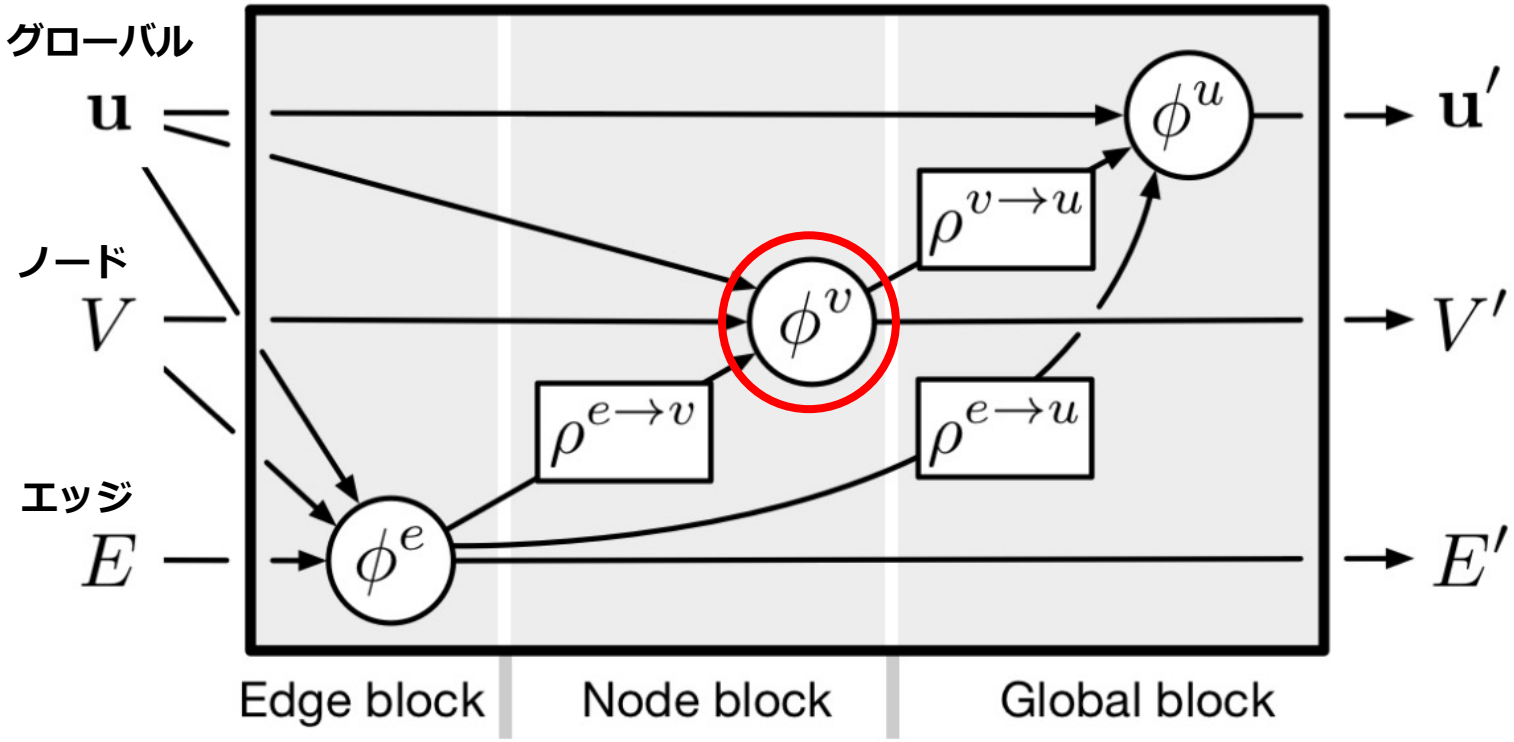
- ノード1の特徴量
- 集約したエッジの特徴量
- グローバル特徴量

を用いて更新される

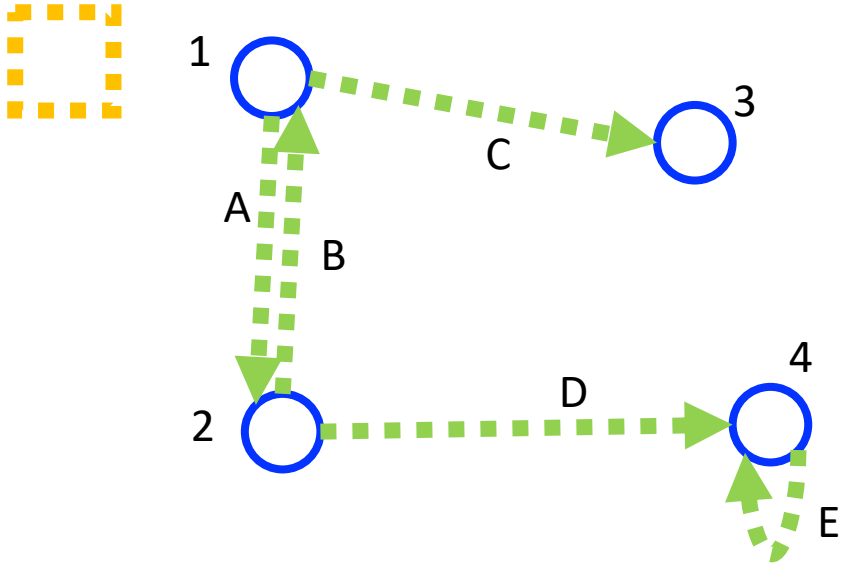
例: $v'_1 = NN(Concatenate([v_1, \bar{e}'_1, u]))$

グラフネットワーク具体的な計算方法 1806.01261

2. ノードの更新



グラフの例



$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$$

$$\mathbf{v}'_i = \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right)$$

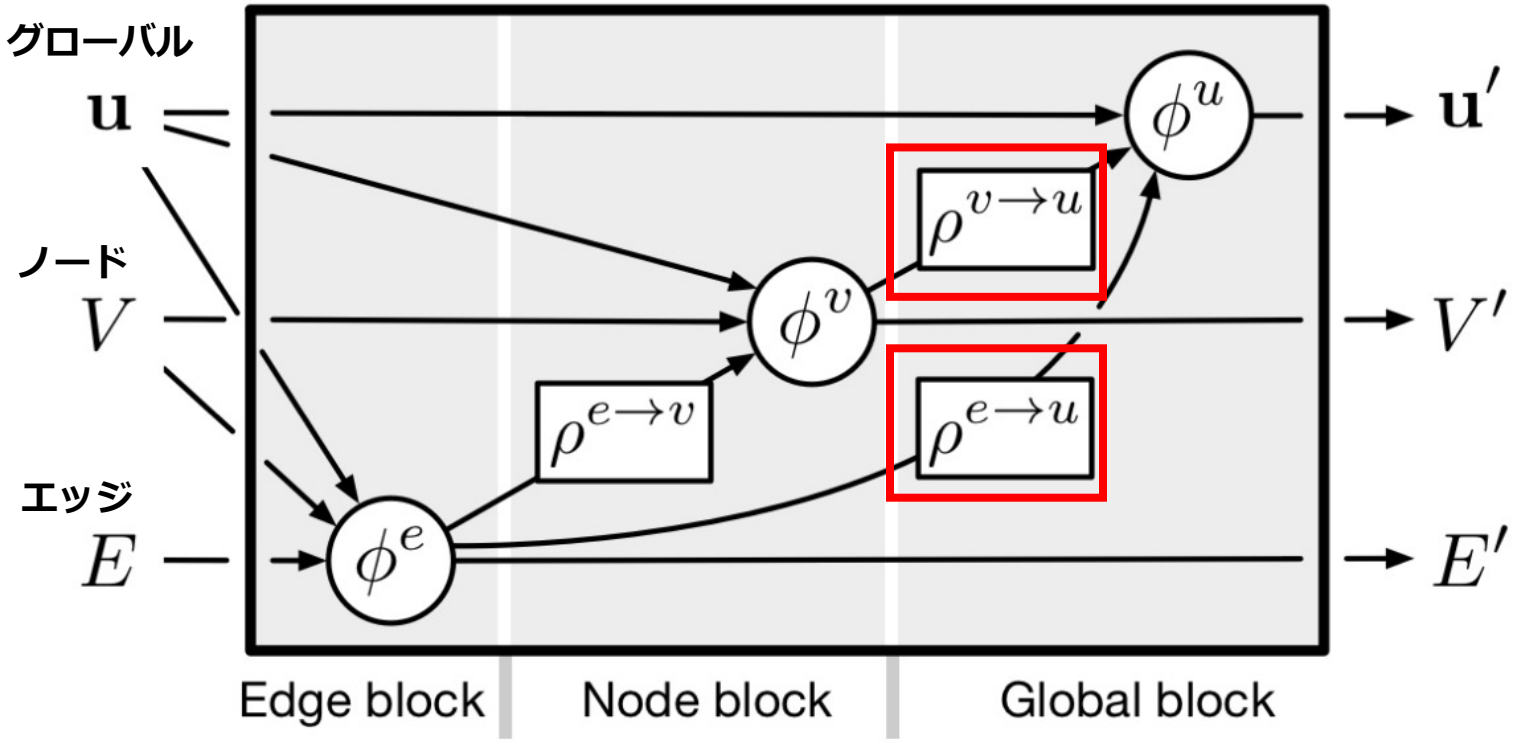
$$\bar{\mathbf{e}}'_i = \rho^{e\to v}\left(E'_i\right)$$

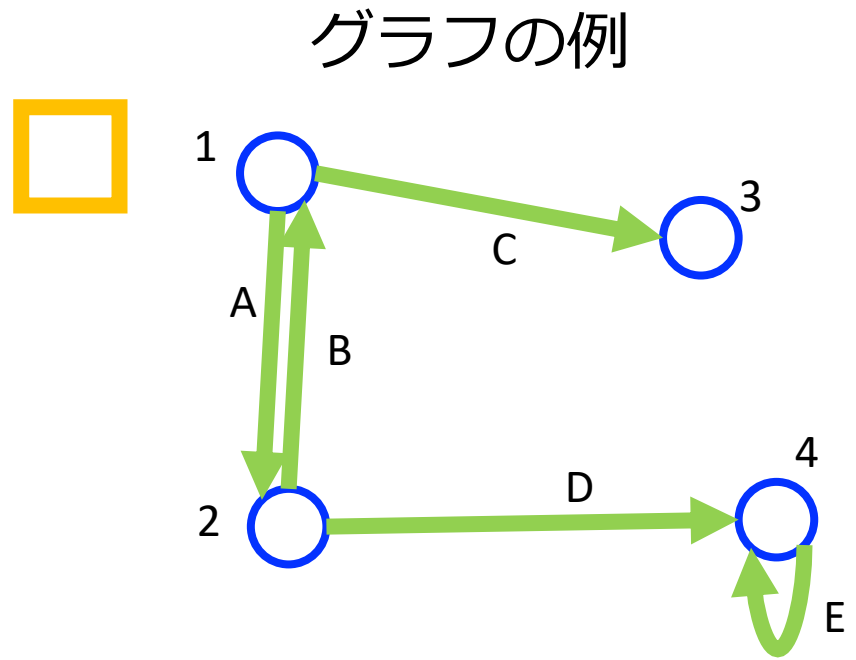
すべての**ノード**は同様に更新される

例: $v'_1 = NN(Concatenate([v_1, \bar{e}'_1, u]))$

グラフネットワーク具体的な計算方法 1806.01261

3. グローバルの更新





全てのノード・エッジの特徴量を集約する

$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$$
$$\mathbf{v}'_i = \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right)$$

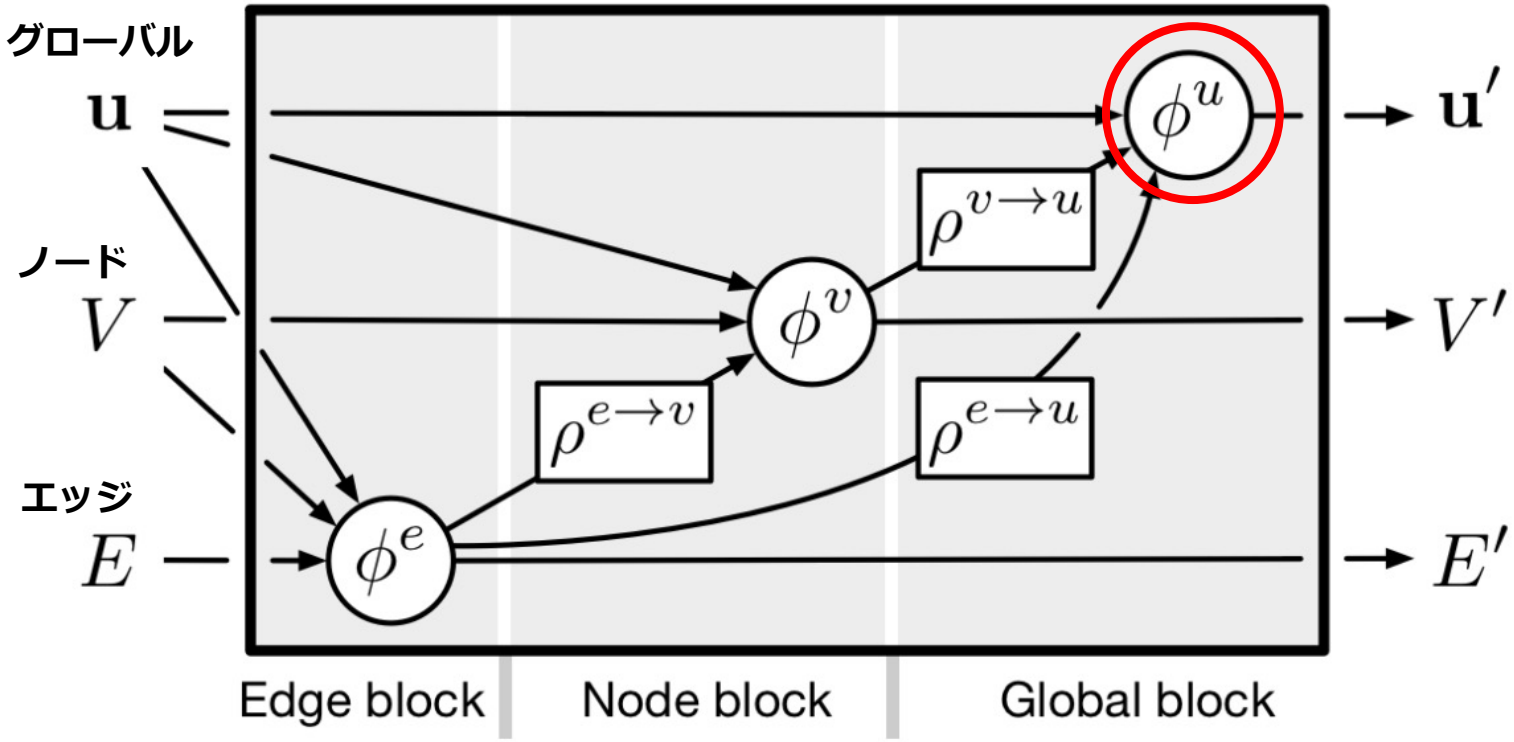
$$\bar{\mathbf{e}}'_i = \rho^{e\to v}\left(E'_i\right)$$
$$\bar{\mathbf{e}}' = \rho^{e\to u}\left(E'\right)$$
$$\bar{\mathbf{v}}' = \rho^{v\to u}\left(V'\right)$$

例: $\bar{e}' = Sum([e_A, e_B, e_C, e_D, e_E])$

例: $\bar{v}' = Sum([v_1, v_2, v_3, v_4])$

グラフネットワーク具体的な計算方法 1806.01261

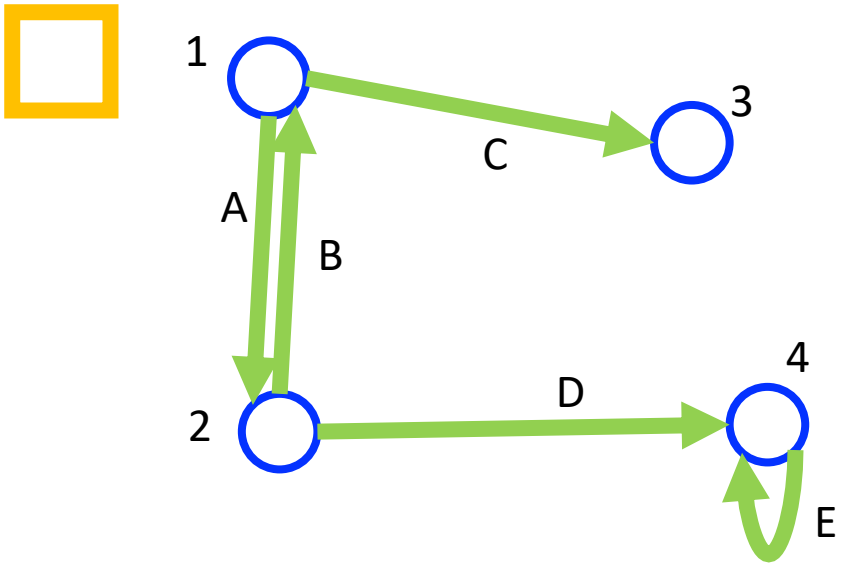
3. グローバルの更新



$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$$
$$\mathbf{v}'_i = \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right)$$
$$\mathbf{u}' = \phi^u\left(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}\right)$$

$$\bar{\mathbf{e}}'_i = \rho^{e\to v}\left(E'_i\right)$$
$$\bar{\mathbf{e}}' = \rho^{e\to u}\left(E'\right)$$
$$\bar{\mathbf{v}}' = \rho^{v\to u}\left(V'\right)$$

グラフの例

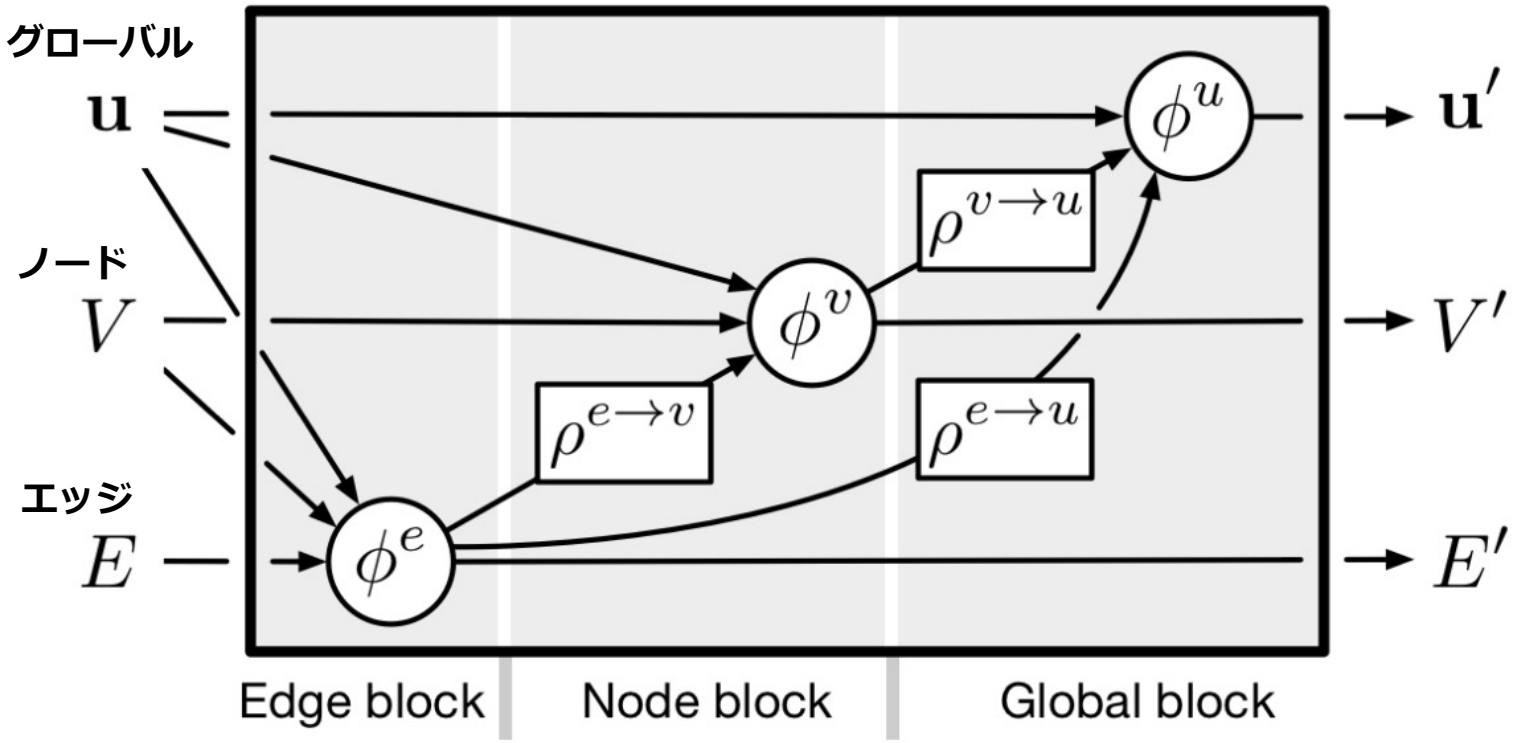


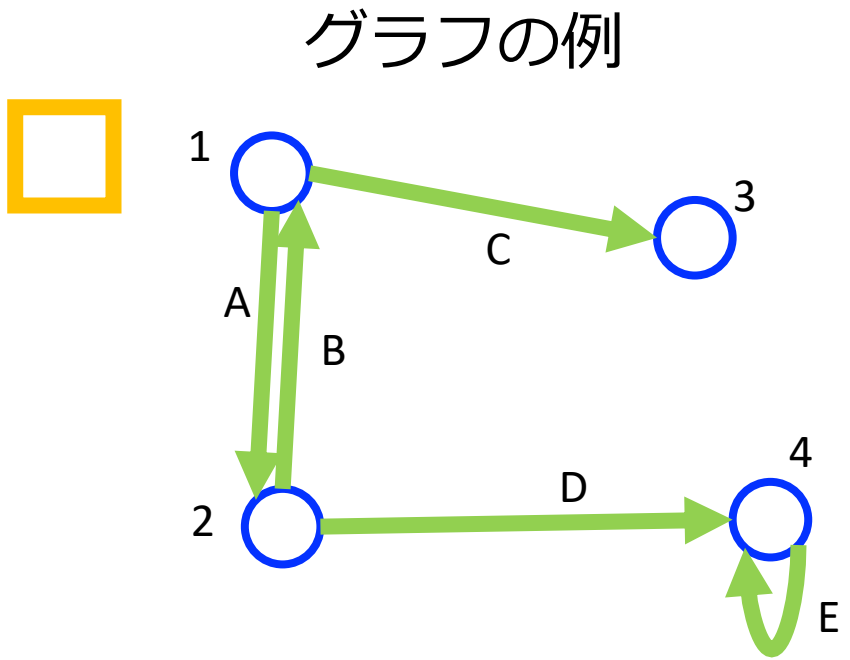
グローバルの特徴量は
- 集約したノードの特徴量
- 集約したエッジの特徴量
- グローバル特徴量

を用いて更新される

例: $u' = NN(Concatenate([\bar{v}', \bar{e}', u]))$

グラフネットワーク具体的な計算方法 1806.01261





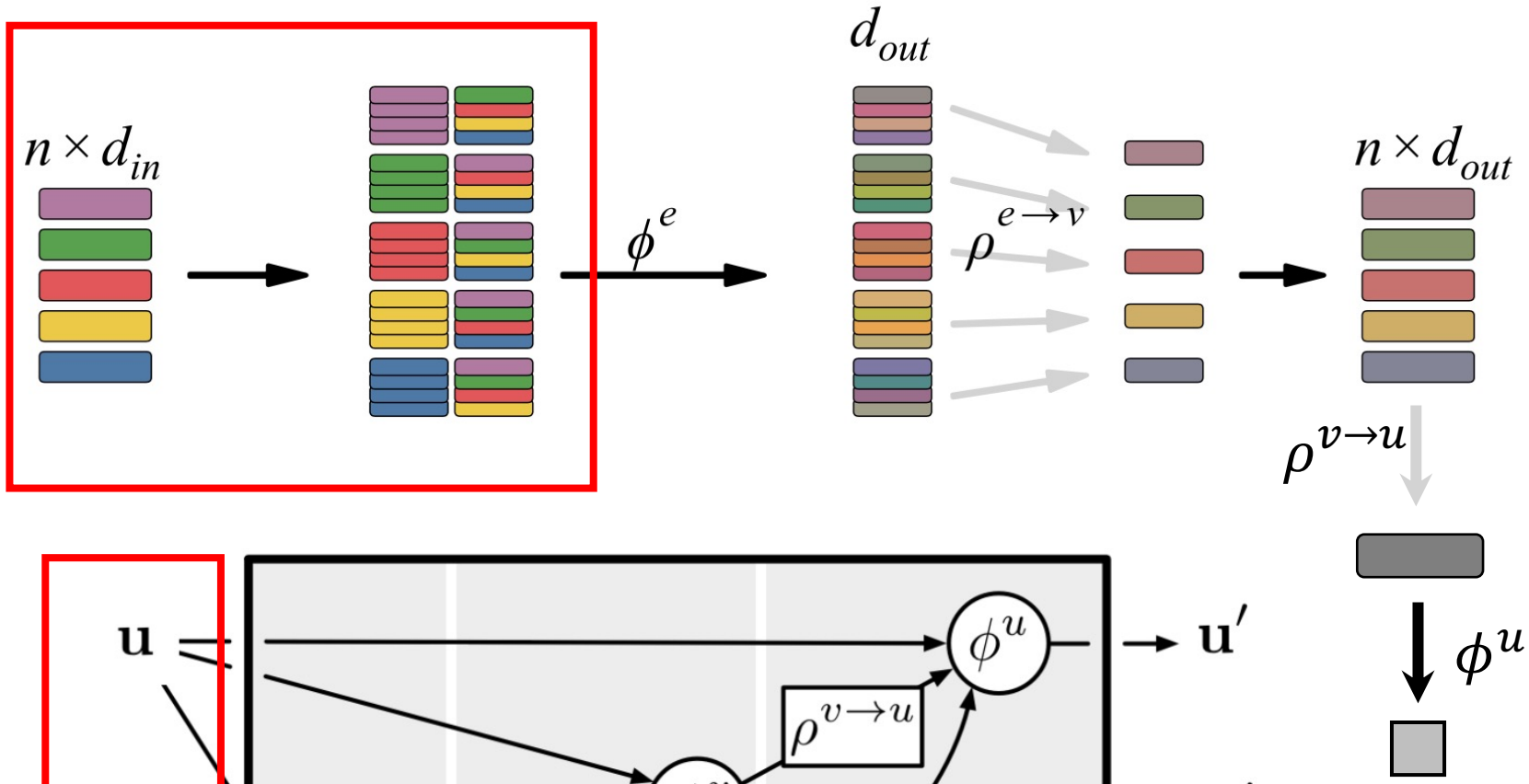
この操作を何度も繰り返す
(繰り返さないこともある)

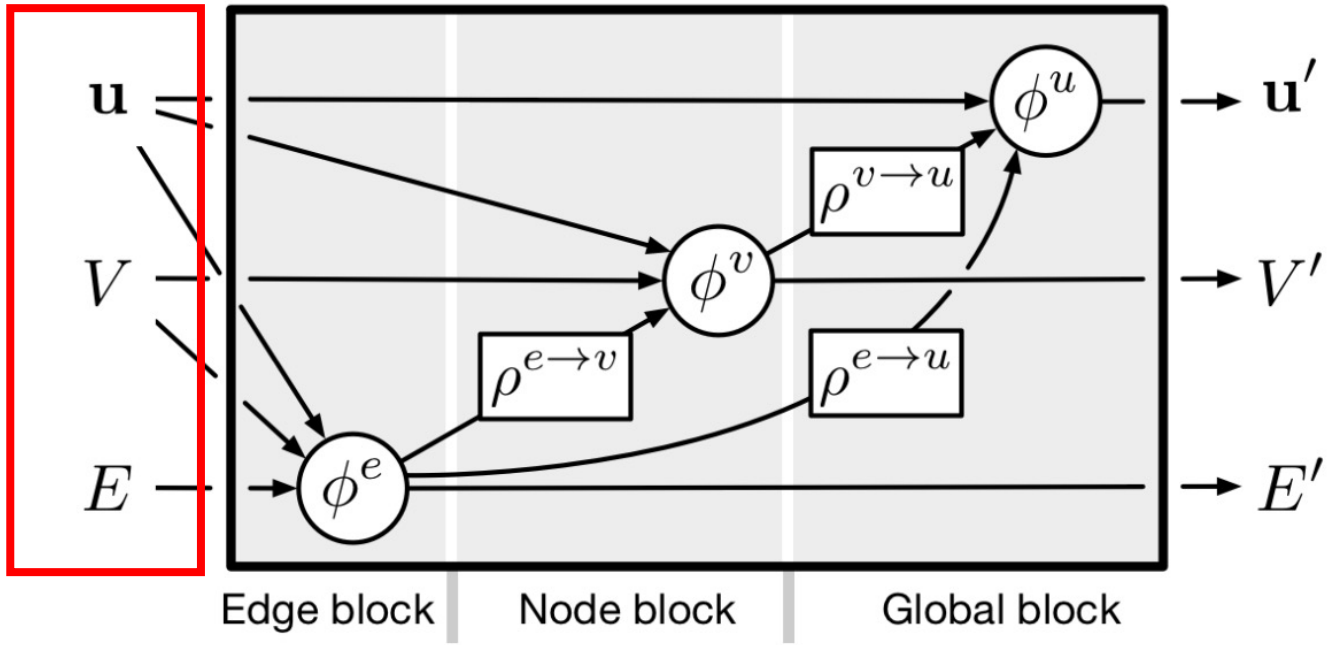
$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right) \qquad \bar{\mathbf{e}}'_i = \rho^{e\to v}\left(E'_i\right)$$

$$\mathbf{v}'_i = \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right) \qquad \bar{\mathbf{e}}' = \rho^{e\to u}\left(E'\right)$$

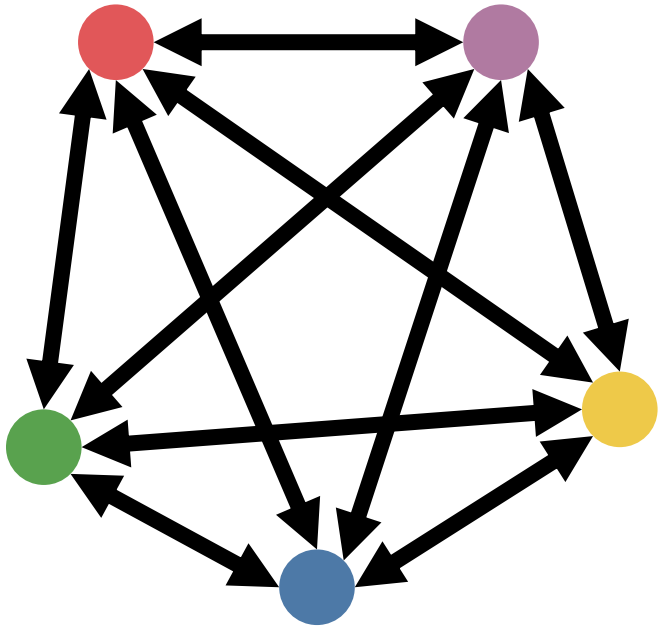
$$\mathbf{u}' = \phi^u\left(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}\right) \qquad \bar{\mathbf{v}}' = \rho^{v\to u}\left(V'\right)$$

グラフネットワーク具体的な計算方法 1806.01261 2007.13681

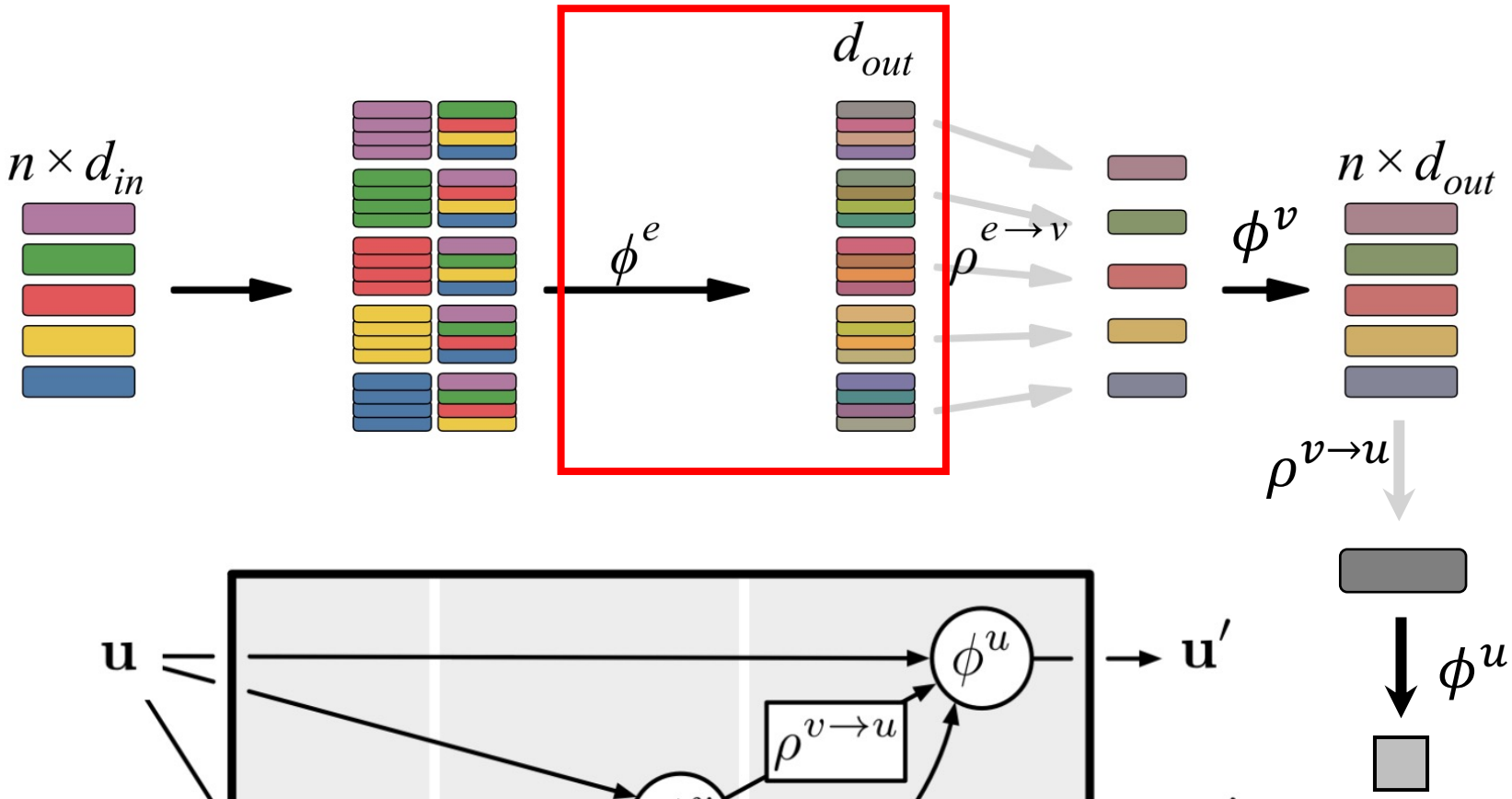


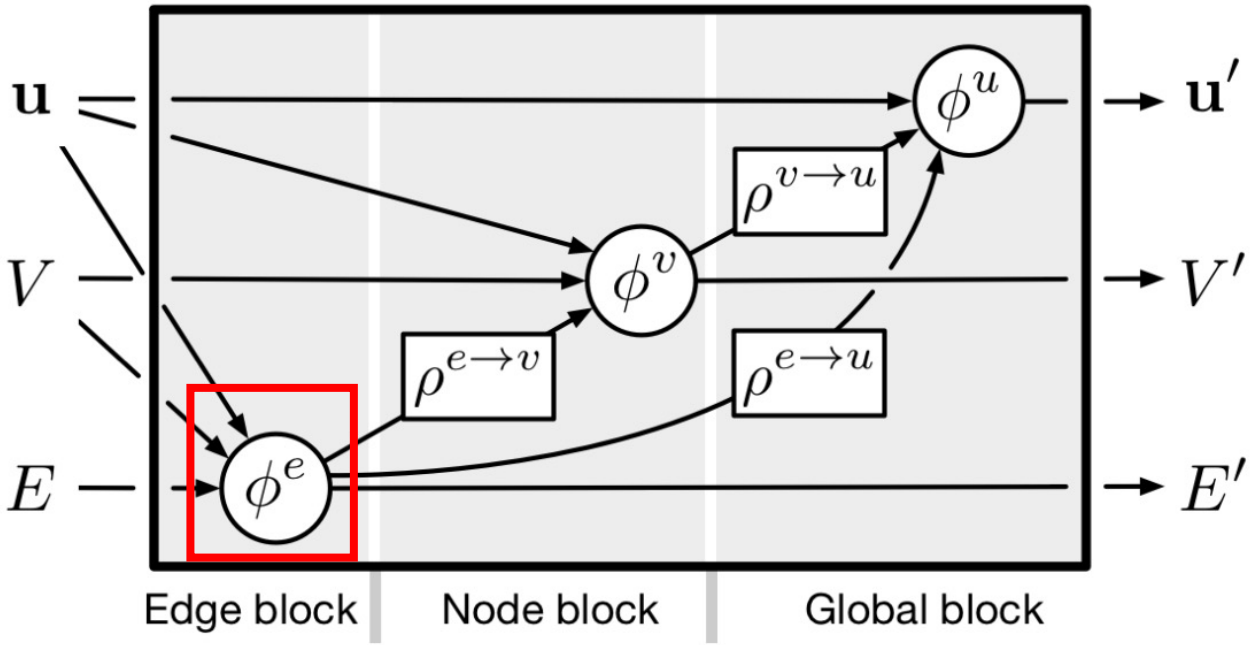


グラフの例

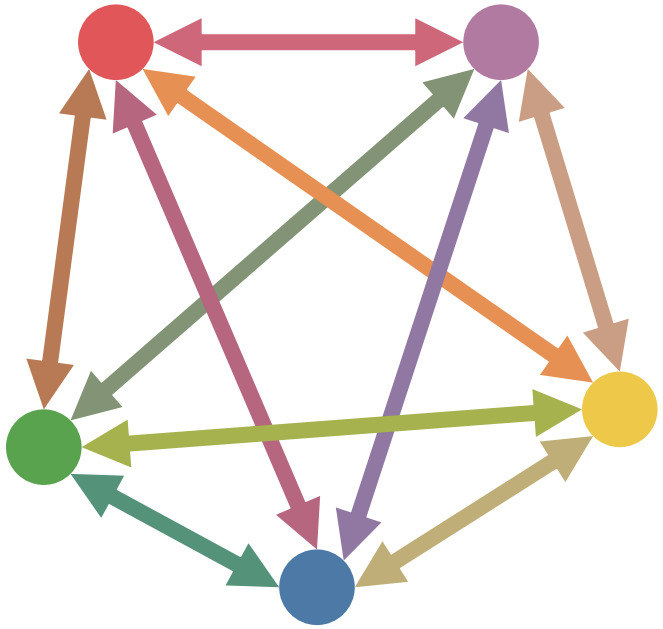


グラフネットワーク具体的な計算方法 1806.01261 2007.13681

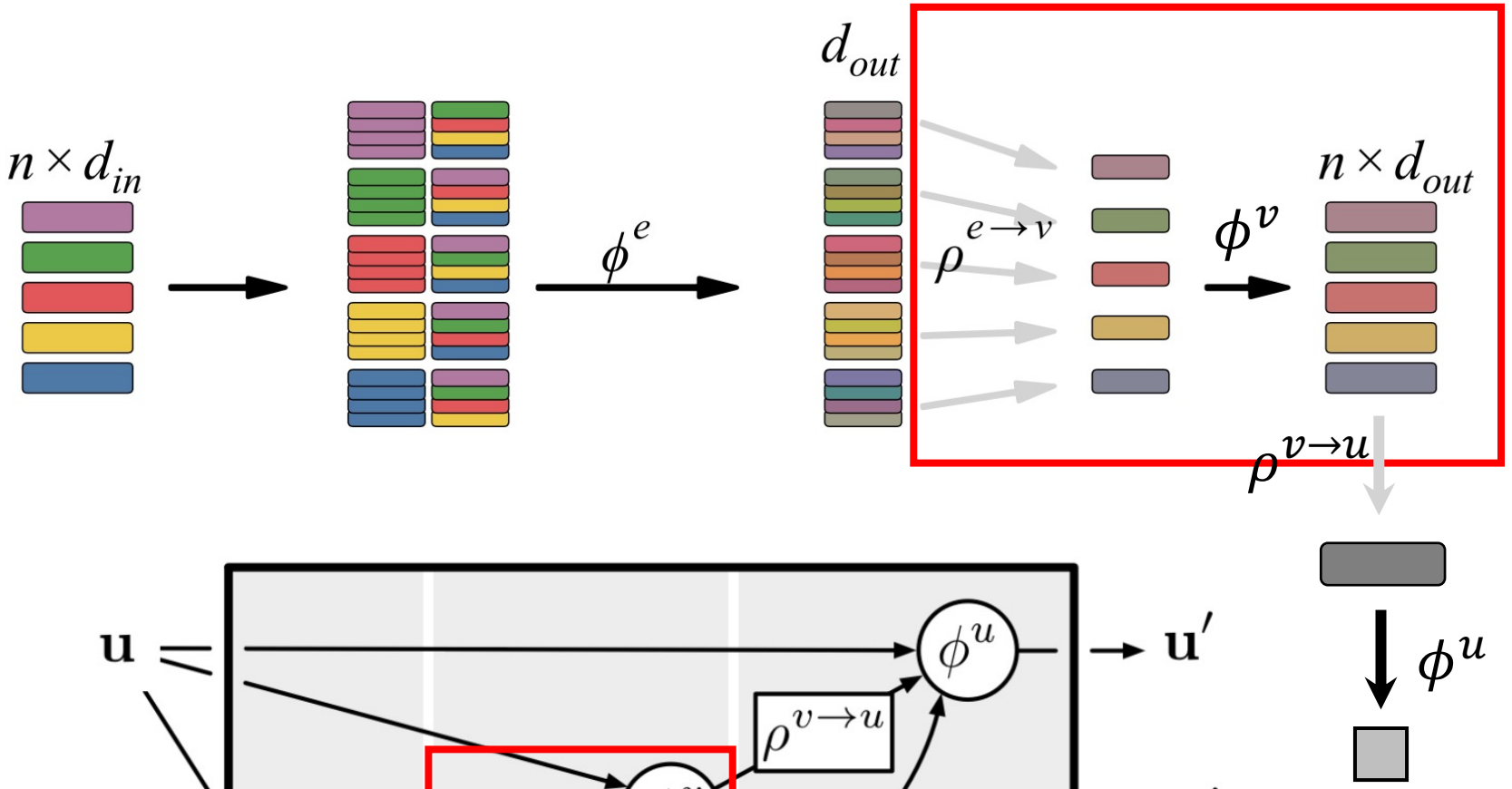


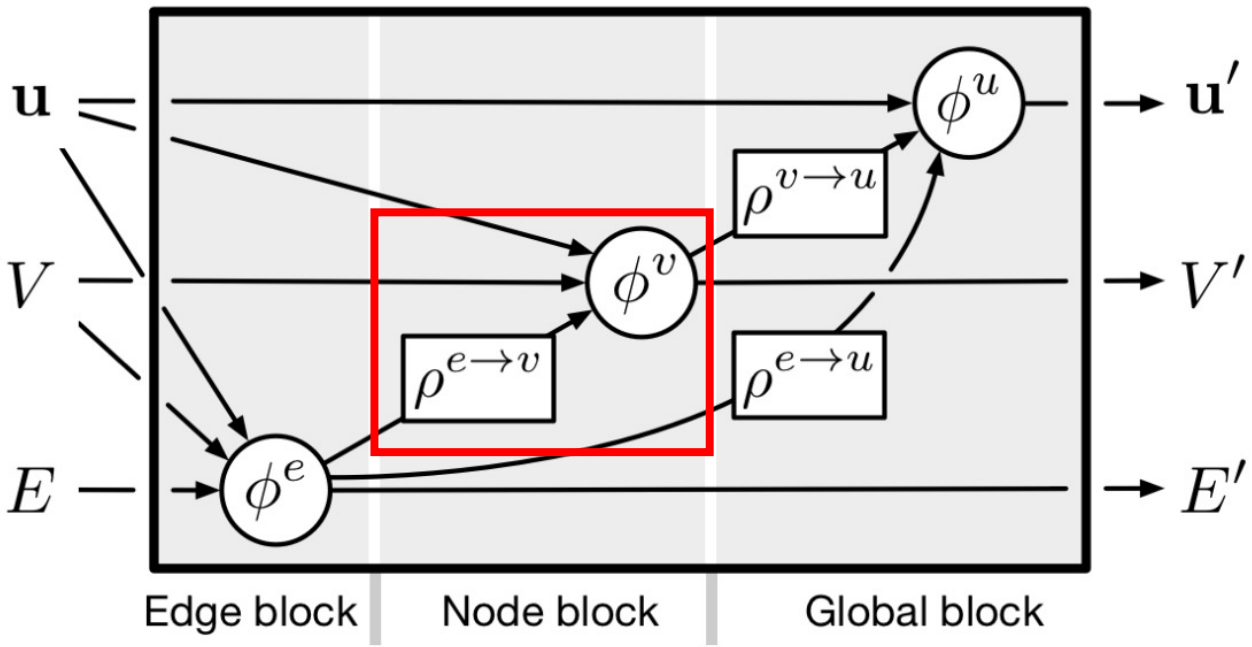


グラフの例

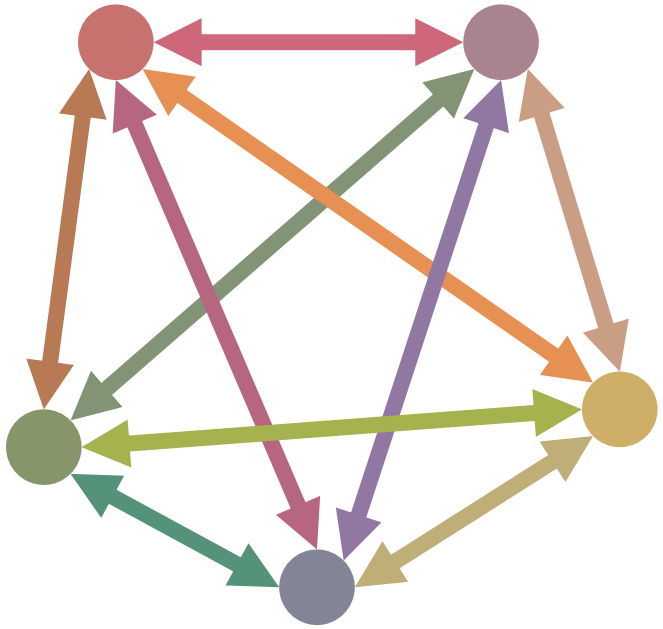


グラフネットワーク具体的な計算方法 1806.01261 2007.13681

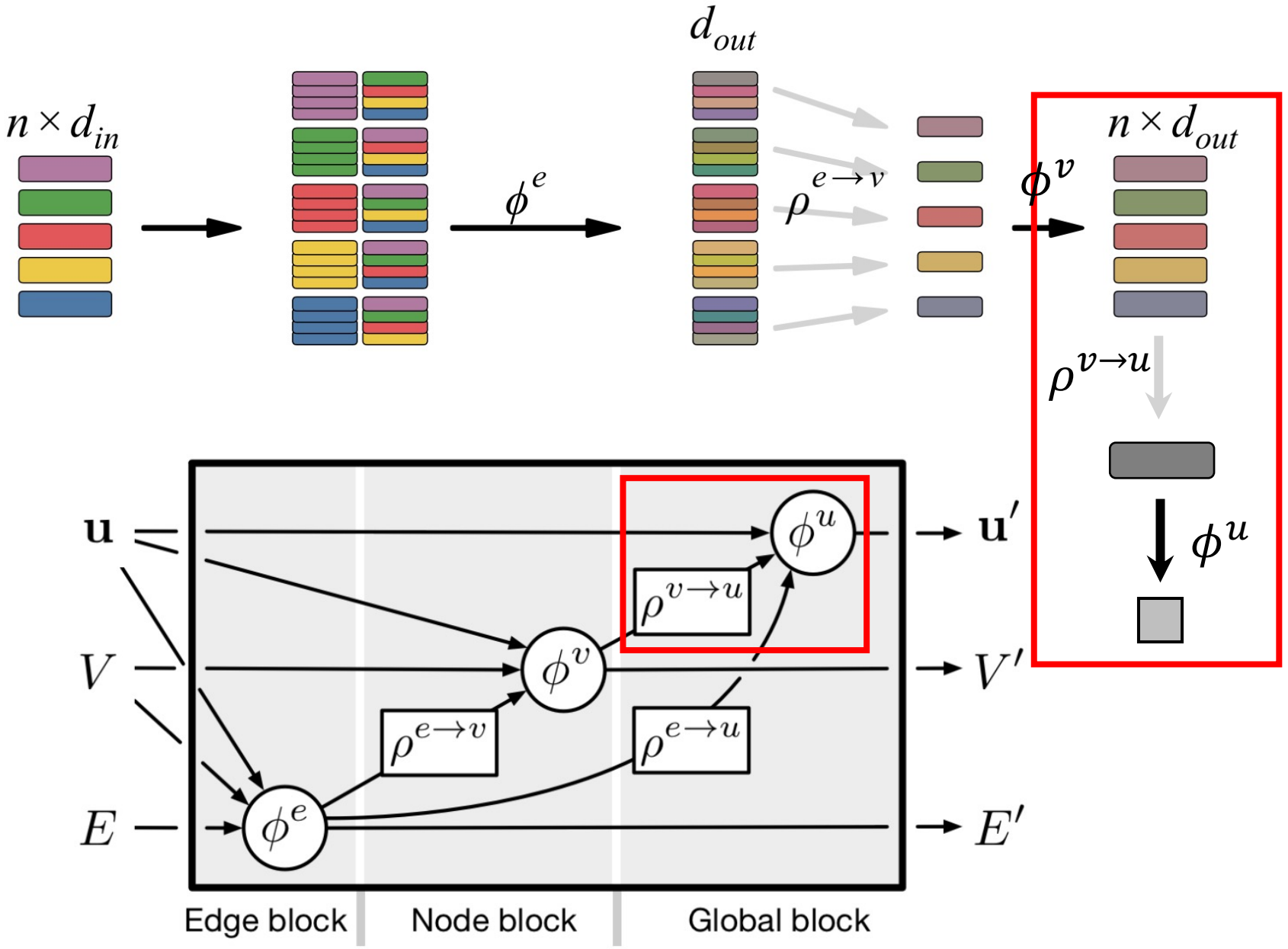




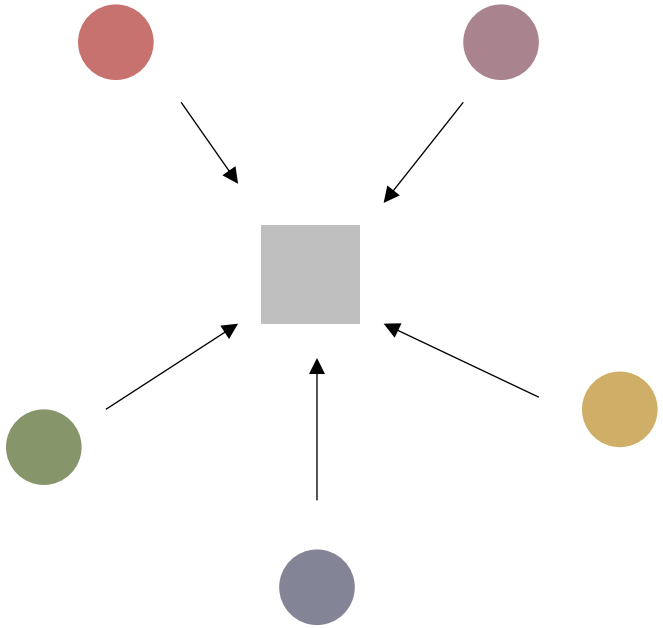
グラフの例



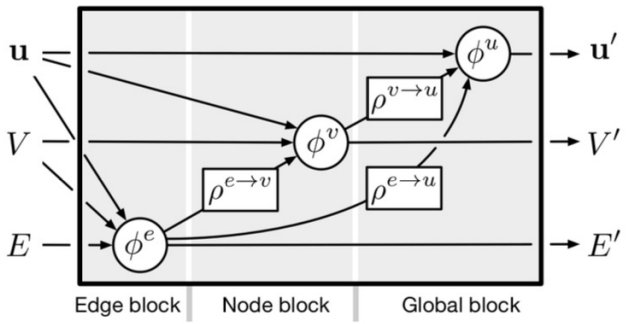
グラフネットワーク具体的な計算方法 1806.01261 2007.13681



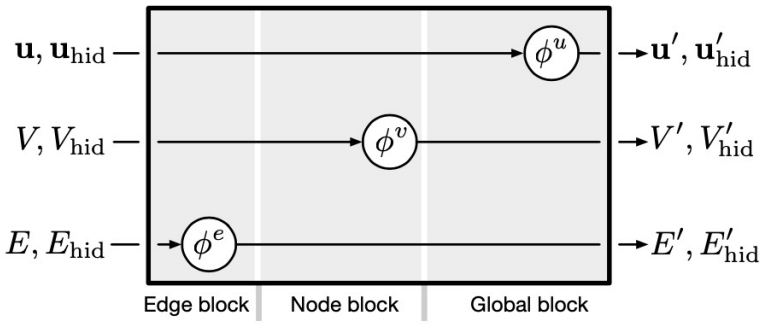
グラフの例



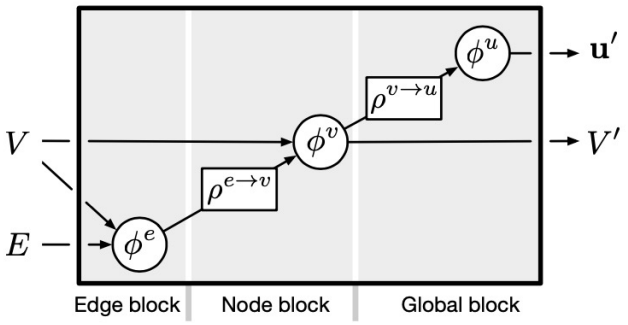
グラフネットワークの分類 1806.01261



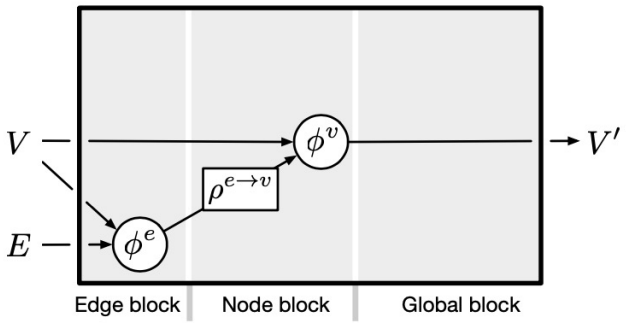
(a) Full GN block



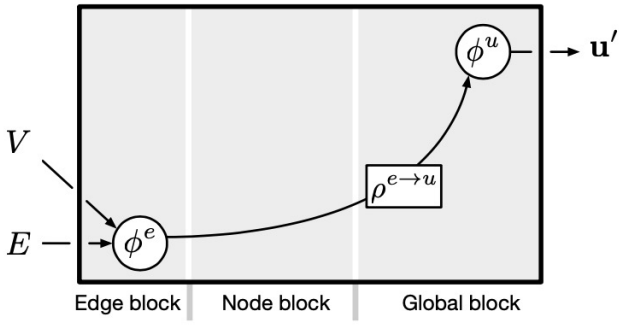
(b) Independent recurrent block



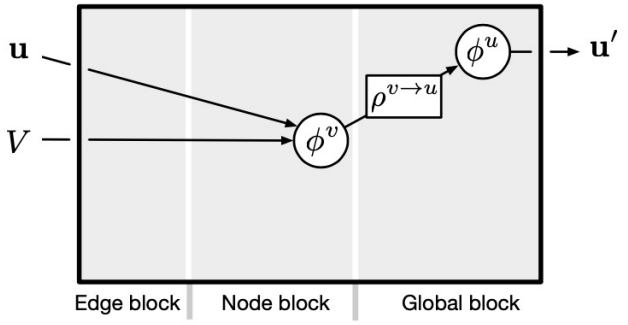
(c) Message-passing neural network



(d) Non-local neural network

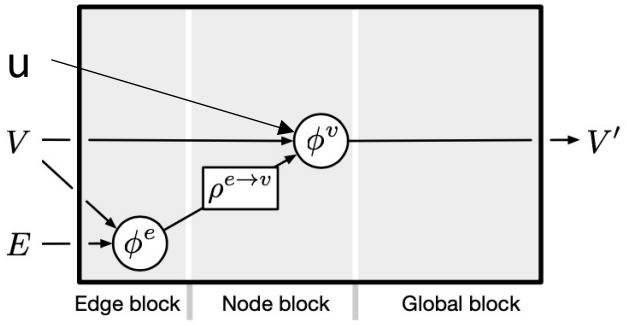


(e) Relation network



(f) Deep set

- 歴史的な経緯で色々な名前が付いている。
- 特に(HEPで)よく用いられるのは
  - Message-passing NN (MPNN)
  - Deep Set
- 実際のところ、境界は曖昧



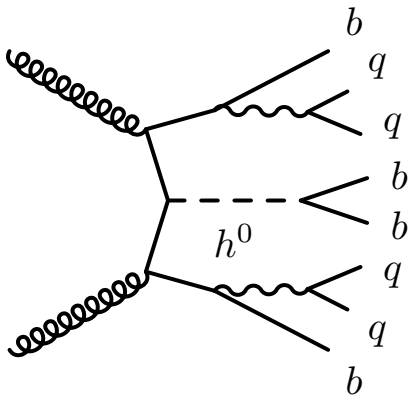
Interaction Network

グラフネットワークを用いた LHCシミュレーション・データでの イベント分類

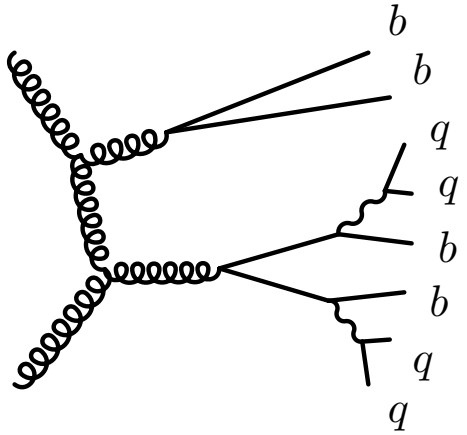
問題設定

信号事象と背景事象

信号事象
ttH(→bb)



背景事象
ttbb



- 信号事象(ttH(→bb))と背景事象(ttbb)の分類問題
- レプトンに崩壊しないチャンネルに着目
  - 4 つのb-クォーク + 4 つのライトクォーク
  - 8個のジェットが観測される
- 粒子数が多く、親粒子の再構成が難しい

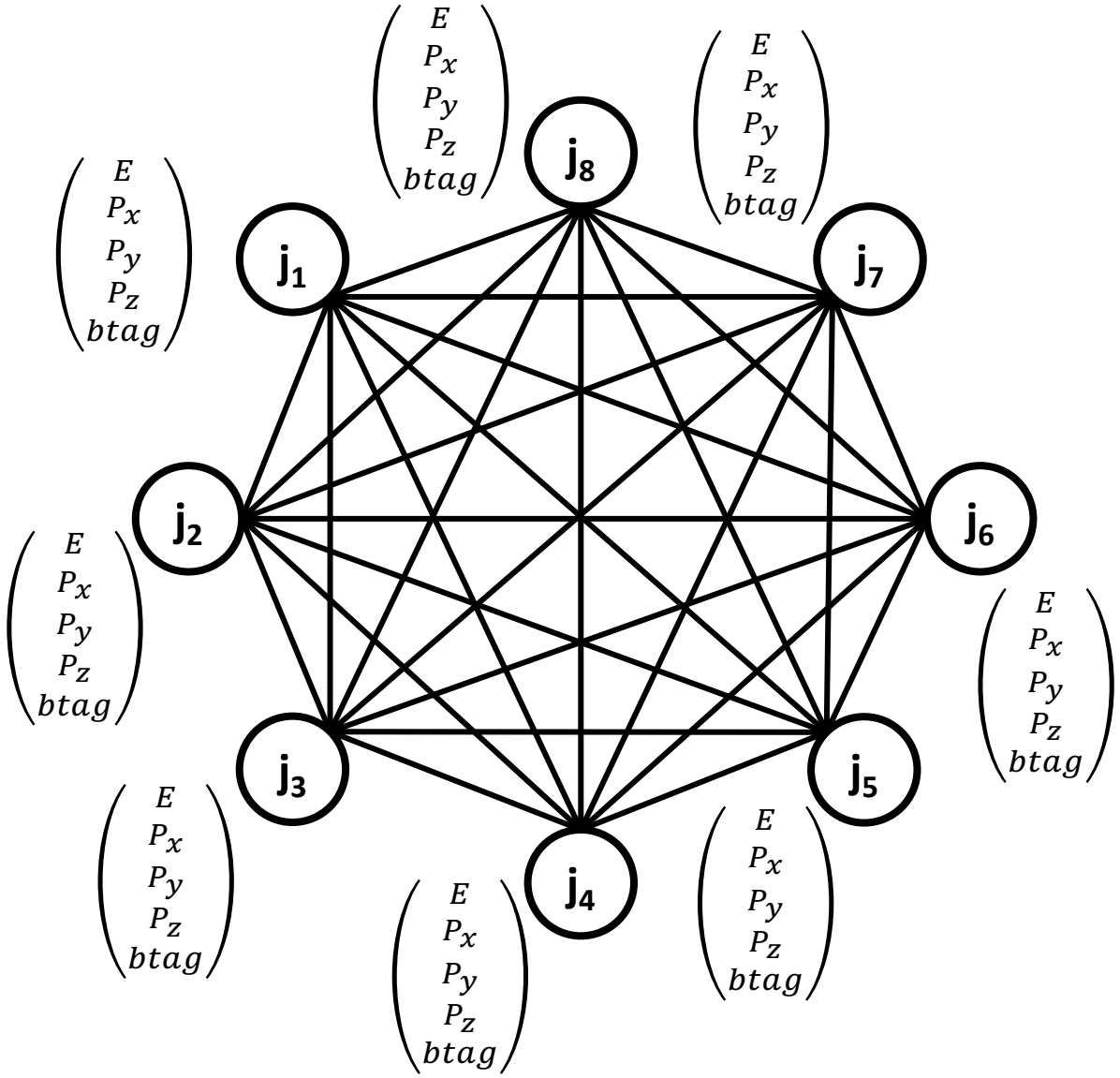
データセット

- シミュレーション・データセットを使用
  - MadGraph5 + Pythia8 + Delphes
  - ATLAS 検出器 のレスポンスを再現
- プロセスごとに100万事象を作成
  - 80 % : 10 % : 10 %の割合で
    Training/Validation/Test 用のサンプルに分割

入力変数

- それぞれの粒子の４元運動量(p_x, p_y, p_z, E) + b-tag
- 8 (粒子数) x 5 (変数の数/粒子) = 40 変数

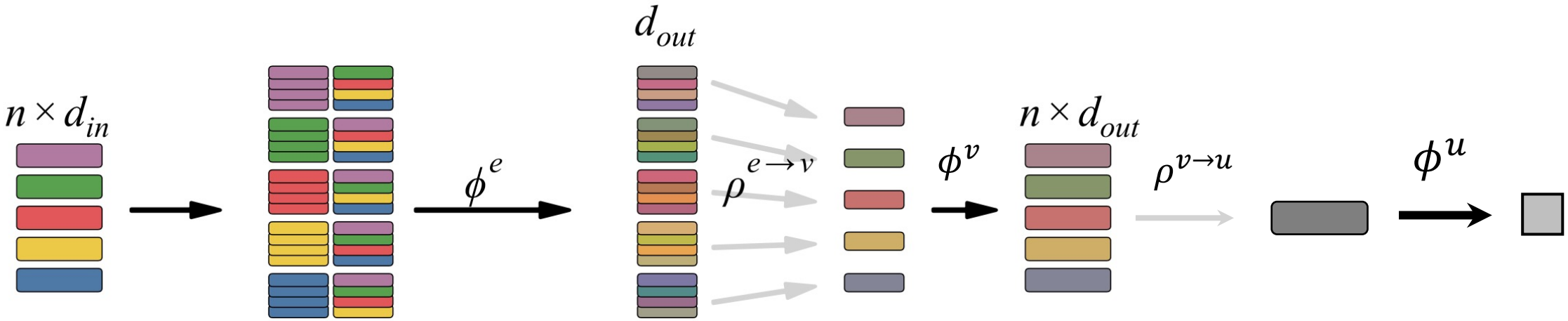
入力に使うグラフ



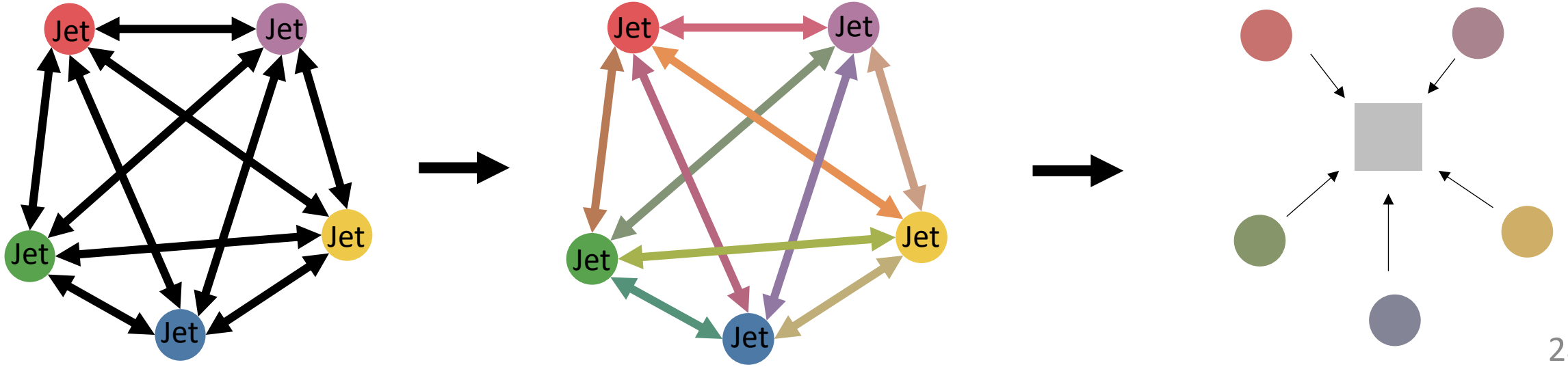
- ８個のジェットと、それぞれに対して5個の特徴量が入力値

- 各ジェットはノードとして割り当てる
  - ノード特徴量: (E, p_x, p_y, p_z, b-tag) ($\mathbb{R}^5$)
- 各ノードは他の全てのノードと接続する(全結合)
  - エッジ特徴量: $\mathbb{R}^N$
    - 2粒子間の関係を表す

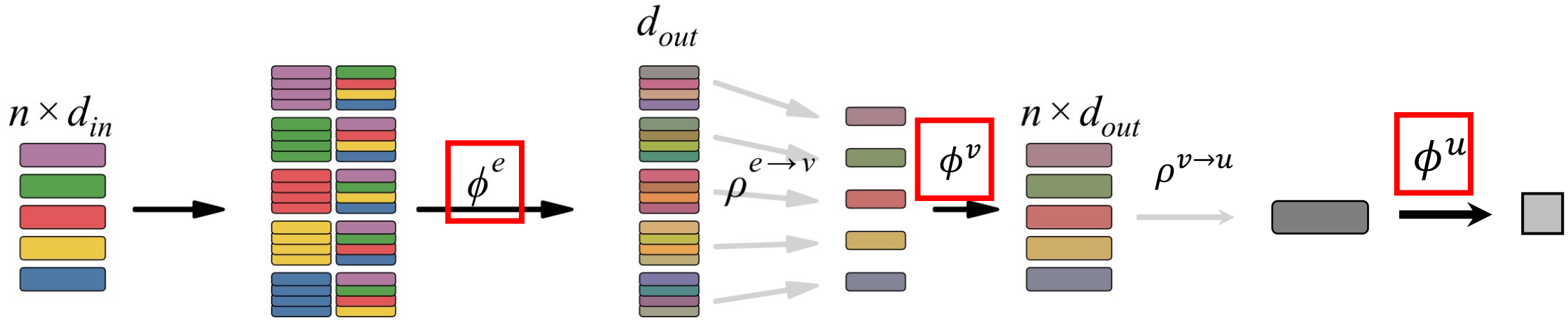
グラフネットワーク具体的な計算方法 1806.01261 2007.13681



ジェット　　ジェット間の関係　　イベント



グラフネットワーク具体的な計算方法 1806.01261 2007.13681



ジェット　　ジェット間の関係　　イベント

3つの MLP module (ϕ) を学習させる

各 MLP の構成

- ノード数が256の隠れ層を2層
- 活性化関数: ReLU (最後の出力のみsigmoid 関数)
- 全ての層に Batch normalization を適用

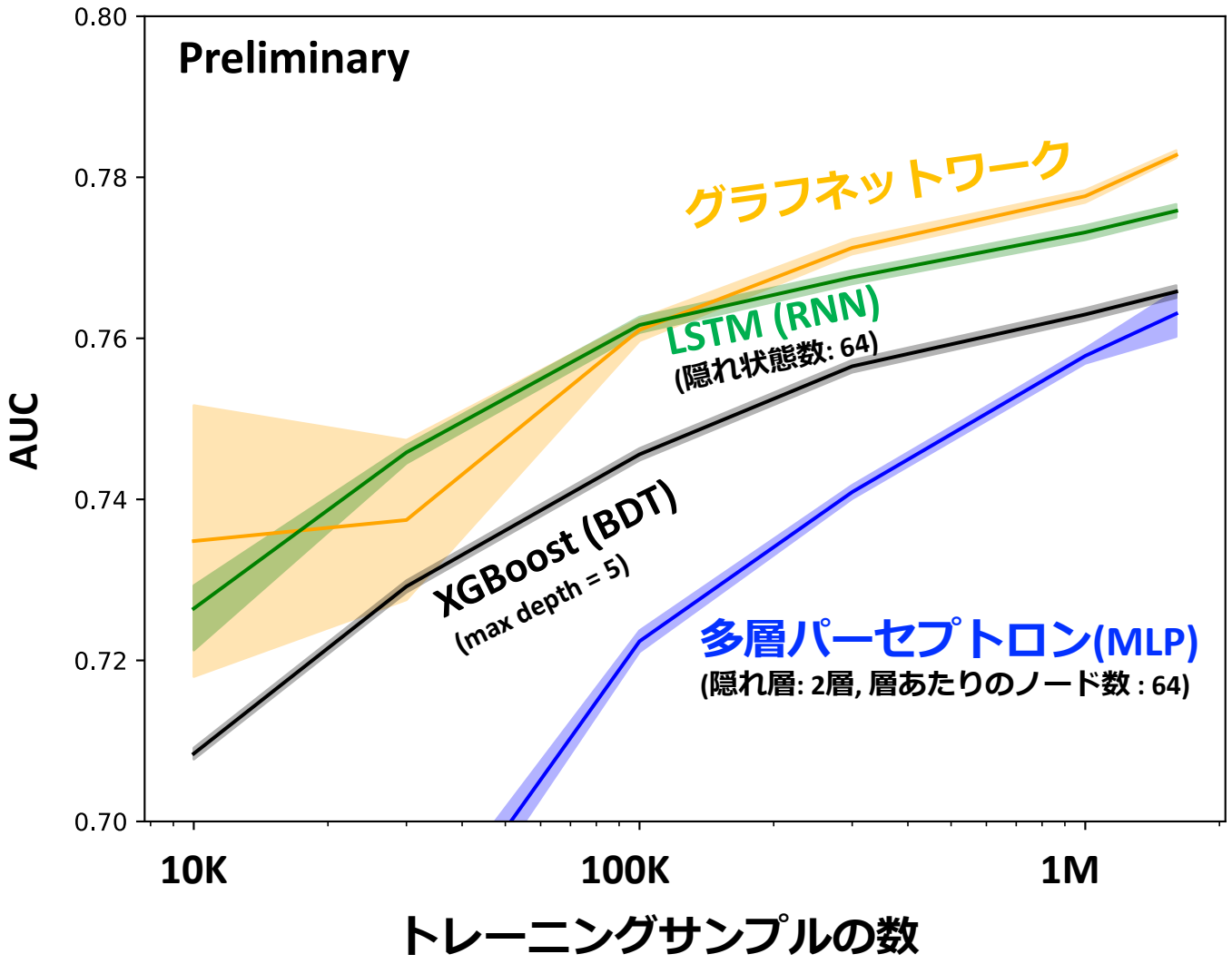
- ノード・エッジの特徴量のサイズや MLPのレイヤー構成等はAUCが最大になるように最適化

比較モデル

- 3つの機械学習モデルと比較
  1. Multilayer perceptron (MLP)
     - 変数40変数を1列に並べたものを入力とする
     - 隠れ層の数・ノード数を変えて最適化
  2. Long short-term memory (LSTM)
     - 粒子(5変数, 最大8個)を横運動量(p_T)順に並べてモデルに入力
     - 内部状態数を変えて最適化
  3. Boosted decision tree (BDT)
     - 変数40変数を1列に並べたものを入力とする
     - 木の深さを変えて最適化
     - XGBoost を使用

- それぞれのモデルについてAUCが最大になるようにハイパーパラメータを最適化。最も良い性能のもの同士でモデルを比較した

結果



- MLP, RNN, BDT と比較して、グラフネットワークは良い性能

- サンプル数が多いところで最良の性能
- サンプルの数が少ないときにもMLPと比較して良い性能
  - 入力変数の構造を深層学習モデルに自然に取り込むことで、過学習を抑えた性能向上ができている

GNNのHEPでの応用例

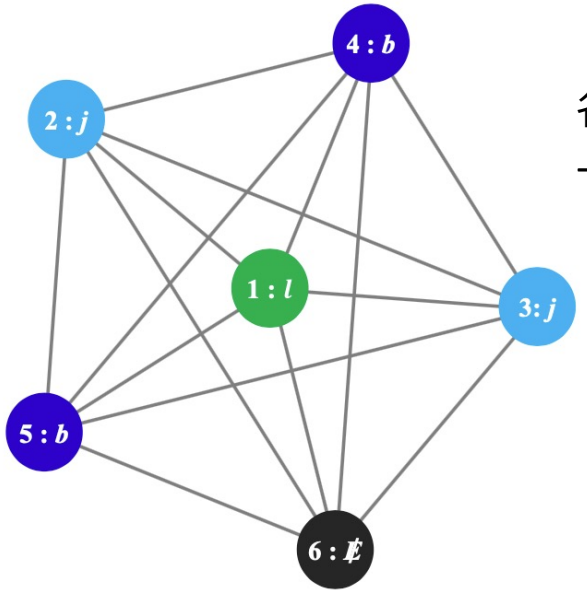
- ここでは、イベント分類で使われている例を紹介します
- 2007.13681 に多くの他の研究例があります

他のグラフネットワークをイベント分類に用いた研究例

グラフネットワークを用いたスカラートップ解析

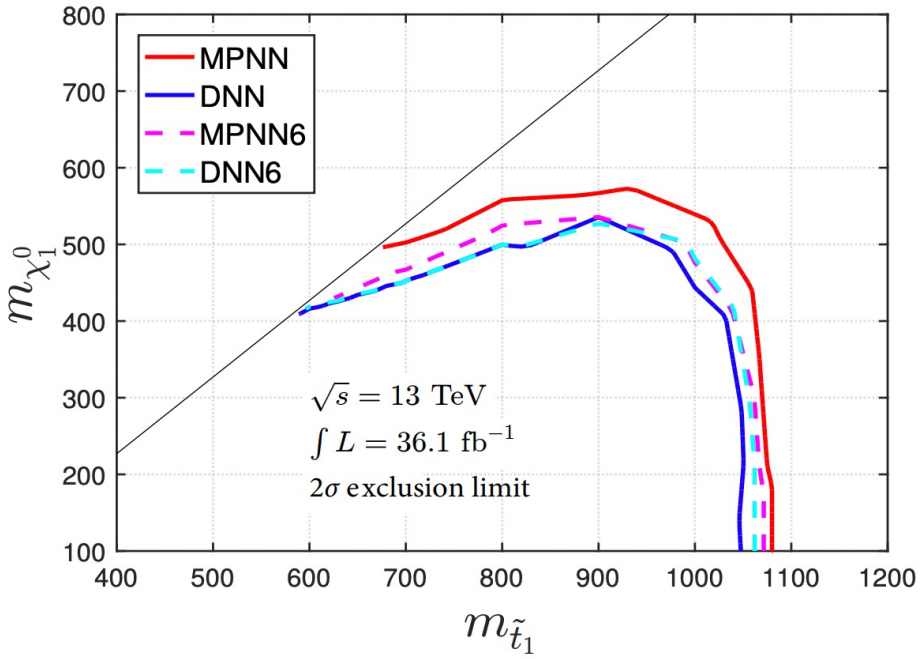
JHEP08(2019)055

$$\tilde{t}_1\tilde{t}_1^* \to t\bar{t}\,\tilde{\chi}_1^0\tilde{\chi}_1^0 \to 2b + 2j + \ell + E_T^{miss}$$



各粒子がノードに対応
- 特徴量: 粒子の種類, pT, E, 質量

GNN vs DNN (MLP)



Message Passing Neural Network (MPNN)

- 隣接ノードの値と粒子間の距離(ΔR)を用いてノードの値を更新する
- 更新を繰り返すことで、グラフ全体に情報を行き渡らせる
- 最後は、ノードの値の平均値を出すことで出力とする

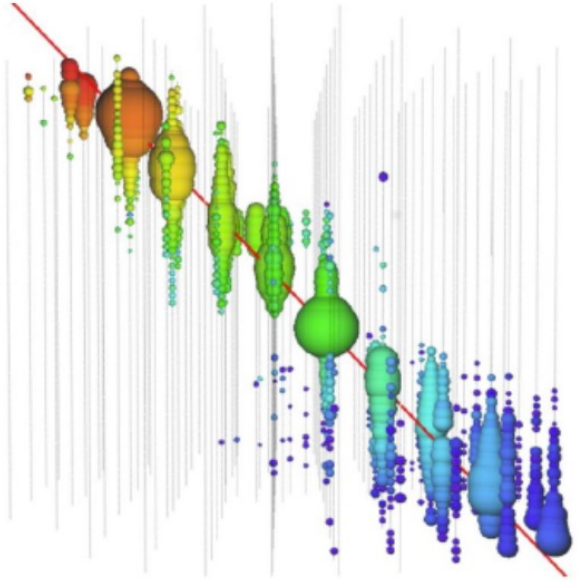
Graph Networkを使うことで、MLPよりも良い性能

他のグラフネットワークをイベント分類に用いた研究例

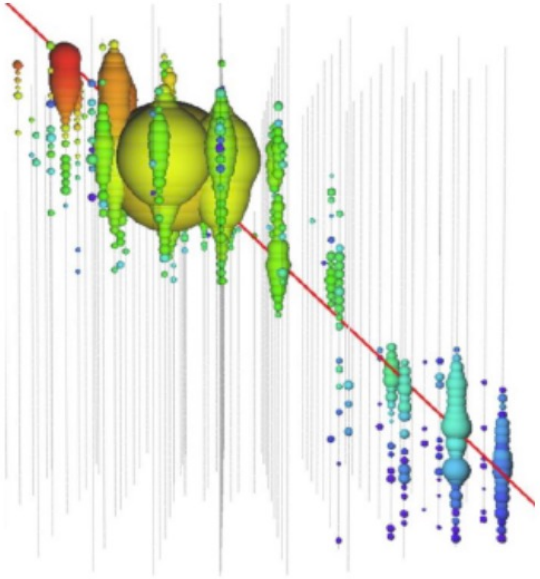
グラフネットワークを用いたIceCubeイベント解析

1809.06166

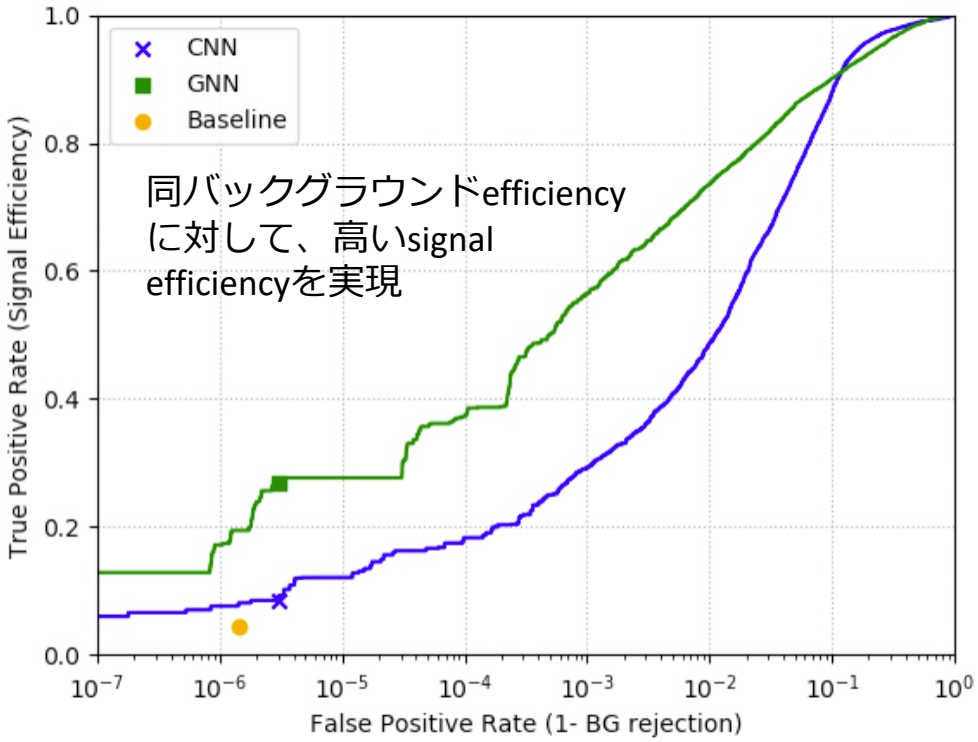
Background:
muon from cosmic shower



Signal:
muon from neutrino



GNN vs CNN



- センサー間の距離を基にグラフを定義
- 各センサーが各ノードに対応。特徴量はチャージ量など。
- 反応があるセンサーだけをノードにすることで、モデルサイズが削減
- グラフを用いることで、イレギュラーなジオメトリに対応

Graph Networkを使うことで、
CNNよりも良い性能

まとめ

- タスクの構造・**ドメイン知識** をモデルに組み込むことが、深層学習の性能向上に必要
- 扱う対象が**グラフ**である問題は多い
  - グラフネットワークはグラフを入力に扱う深層学習モデル

- **グラフネットワーク**を物理解析タスクに適用した
  - グラフネットワークの利点: 入力変数の構造化、可変の粒子数に対応、順序付けが不要
  - グラフネットワークの性能はMLPやBDTと比較して良い

- 実際に物理結果に使った例は(おそらく)まだないが、研究は多く進められている
  - これからの実用化に期待

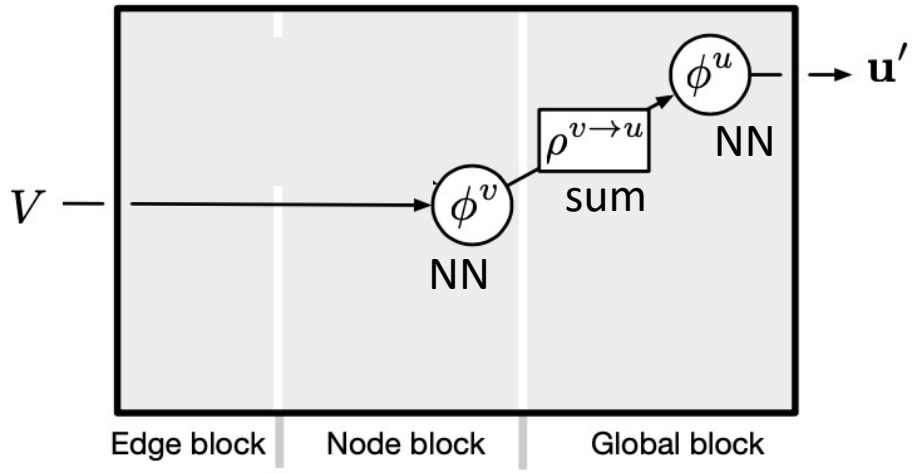
Backup

グラフネットワークのHEPでの応用例

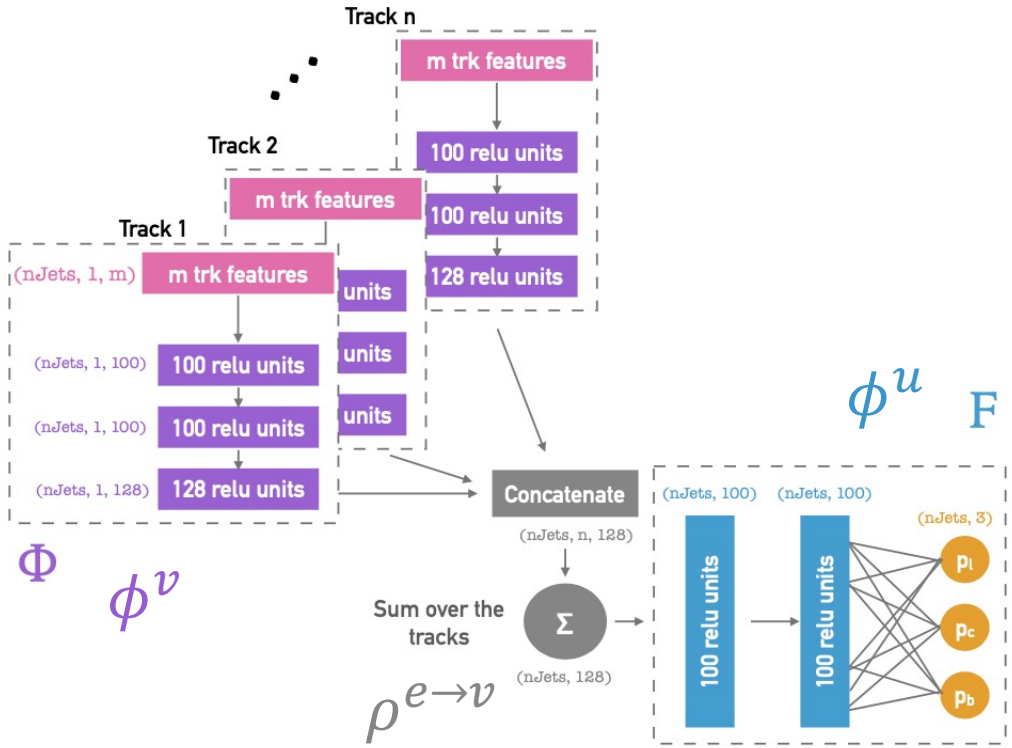
Deep Impact Parameter Sets (DIPS): [ATL-PHYS-PUB-2020-014]

- ジェット中のトラックを使ってb-tagging

		対象	特徴量 (attribute)
入力	ノード	トラック	ヒット数, IP, 4-vector 等
出力	グローバル	ラベル	b/c/l jet である確率



(f) Deep set



$$O\left(\{p_1,\ldots,p_n\}\right) = F\left(\sum_{i=1}^{n}\Phi\left(p_i\right)\right),$$

NN で(トラックごとの)特徴量の作成

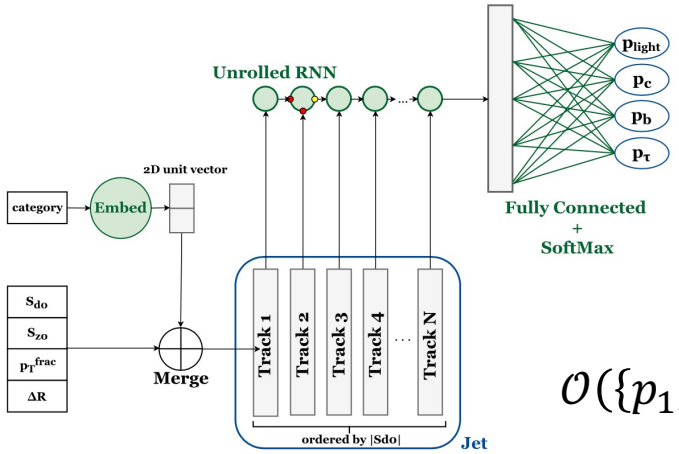
Aggregation

Global (フレーバーのラベル)のupdate

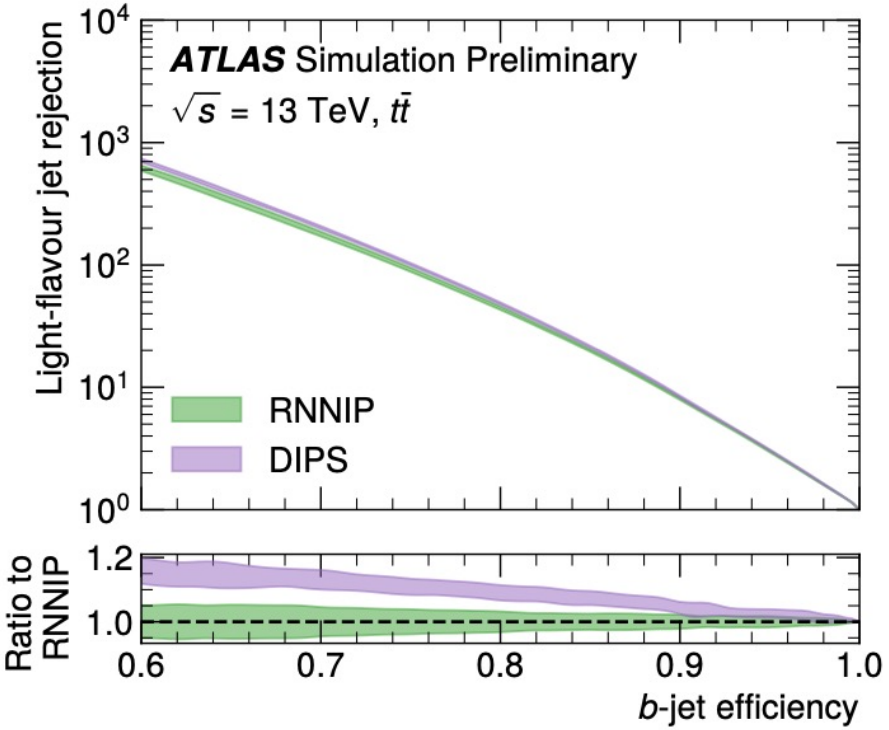
グラフネットワークのHEPでの応用例

Deep Impact Parameter Sets (DIPS): ATL-PHYS-PUB-2020-014

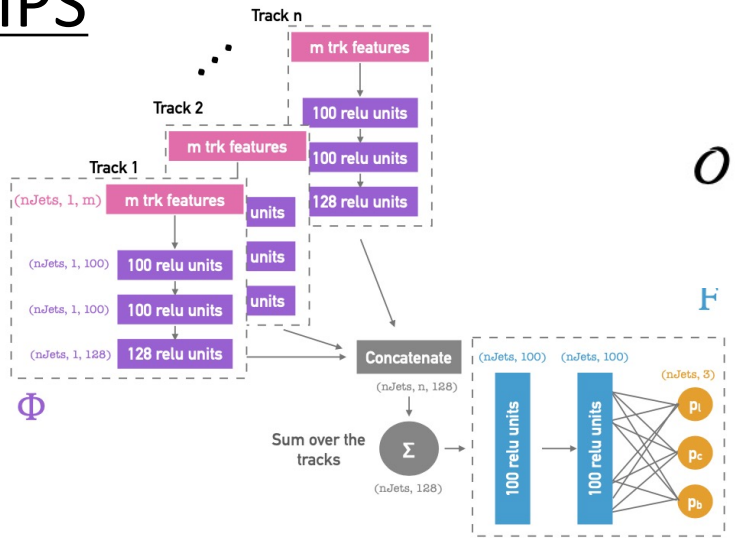
RNNIP



$$\mathcal{O}(\{p_1, \ldots, p_n\}) = F(\mathrm{LSTM}(\{p_1, \ldots, p_n\}))$$



DIPS



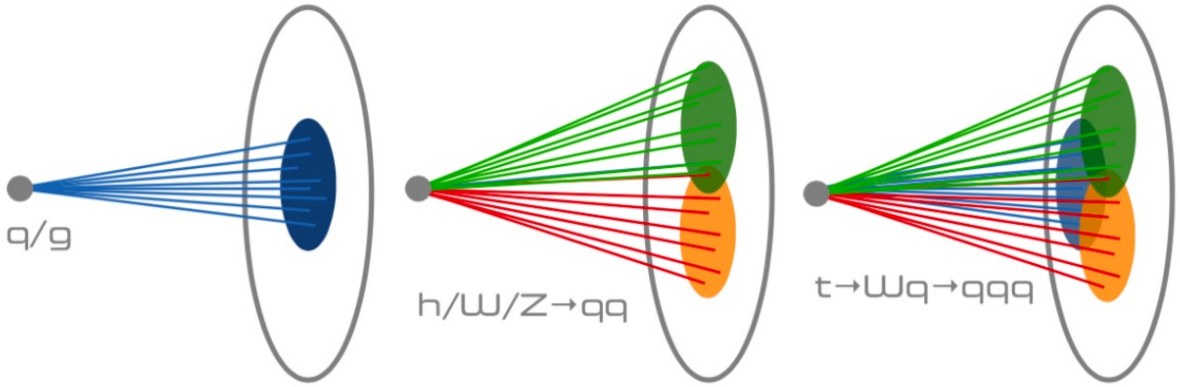
$$O\left(\{p_1, \ldots, p_n\}\right) = F\left(\sum_{i=1}^{n} \Phi\left(p_i\right)\right),$$

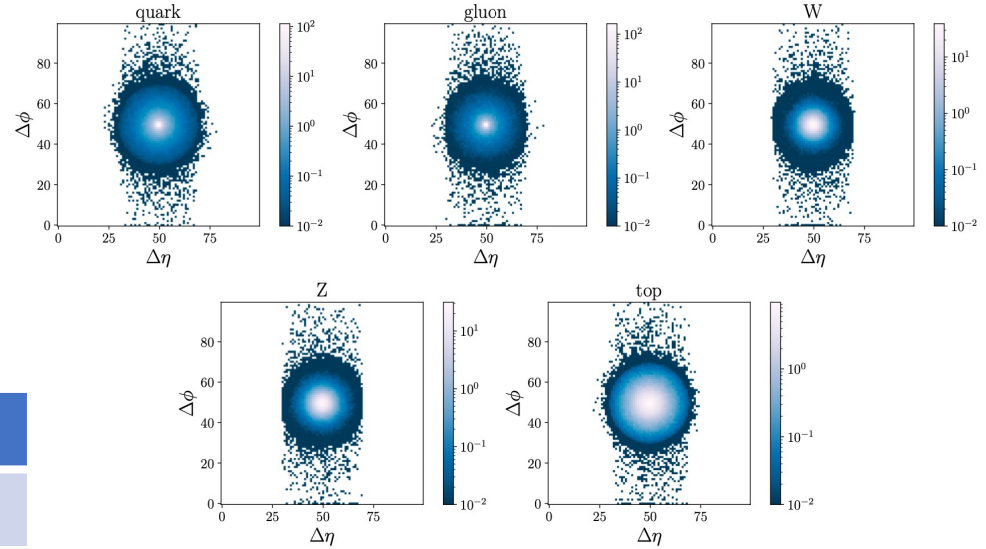
グラフネットワークのHEPでの応用例

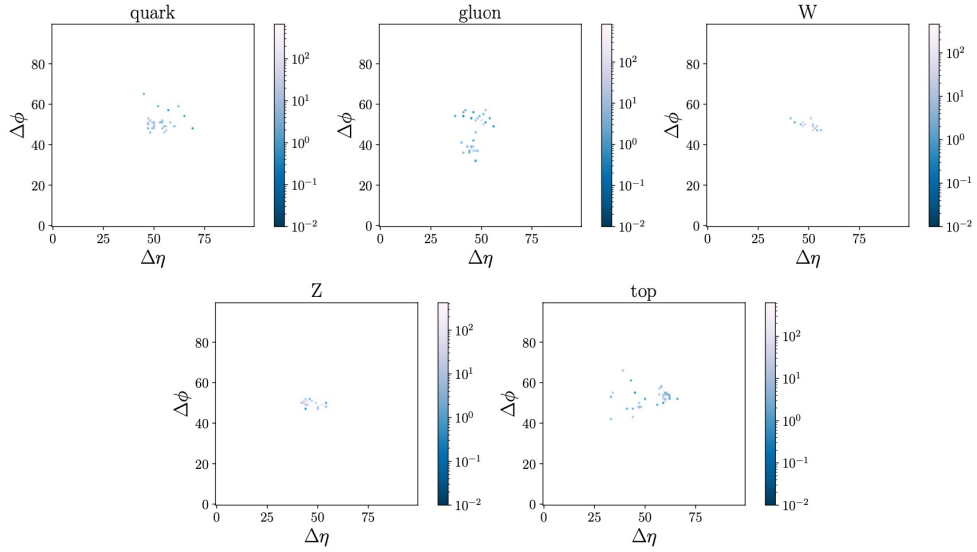
Interaction Network for jet tagging (JEDI-NET): [1908.05318]

- Large-R Jet (R=0.8)中の粒子(最大150個)を使ってjet tagging

		対象	特徴量 (attribute)
入力	ノード	粒子	4-vector, $\Delta\eta, \Delta\phi$, rel pT, E 等
出力	グローバル	ラベル	q/g/W/Z/t である確率

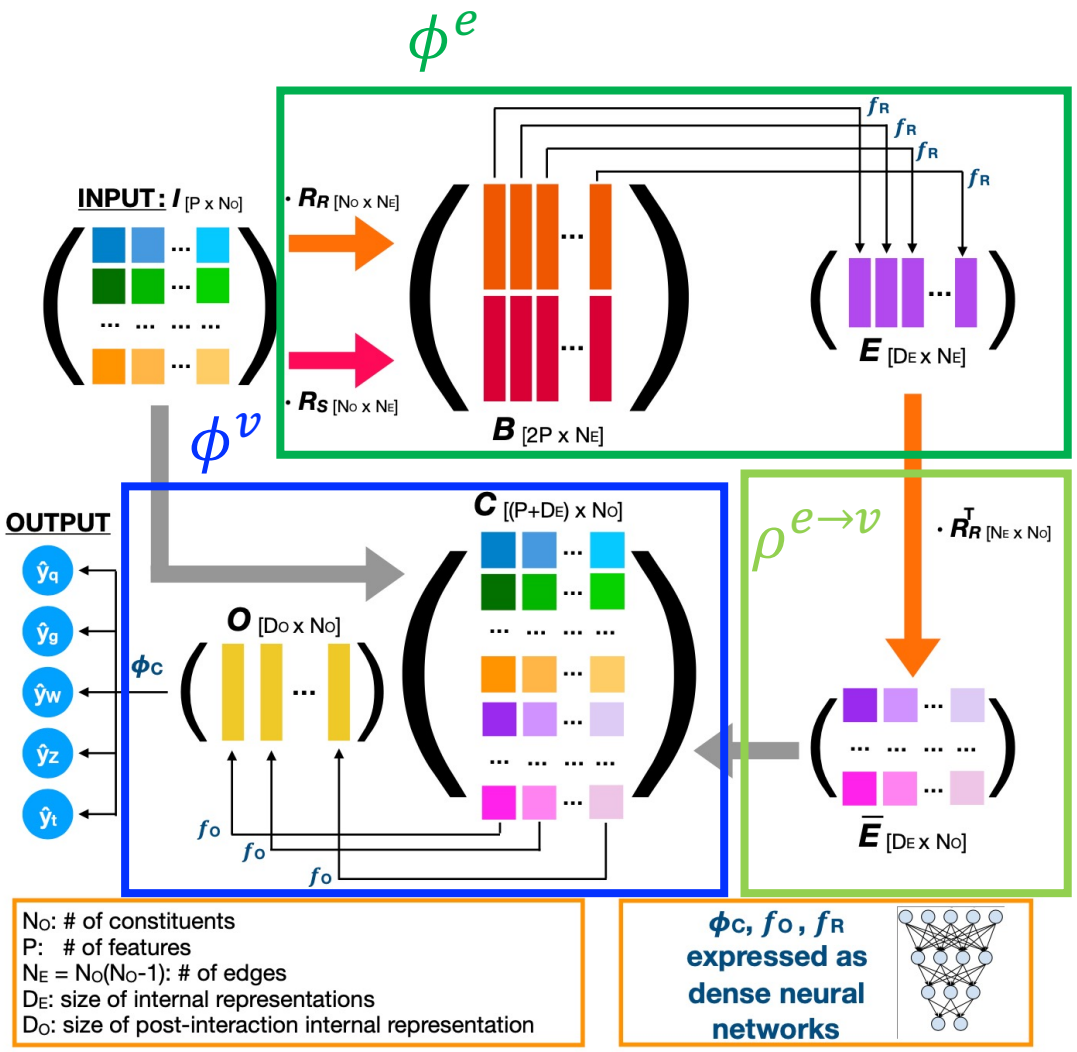


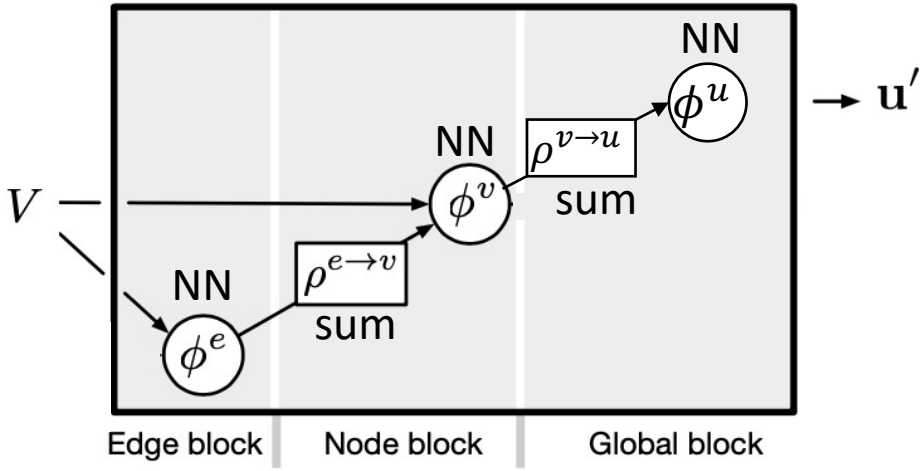




グラフネットワークのHEPでの応用例

Interaction Network for jet tagging (JEDI-NET): [1908.05318]





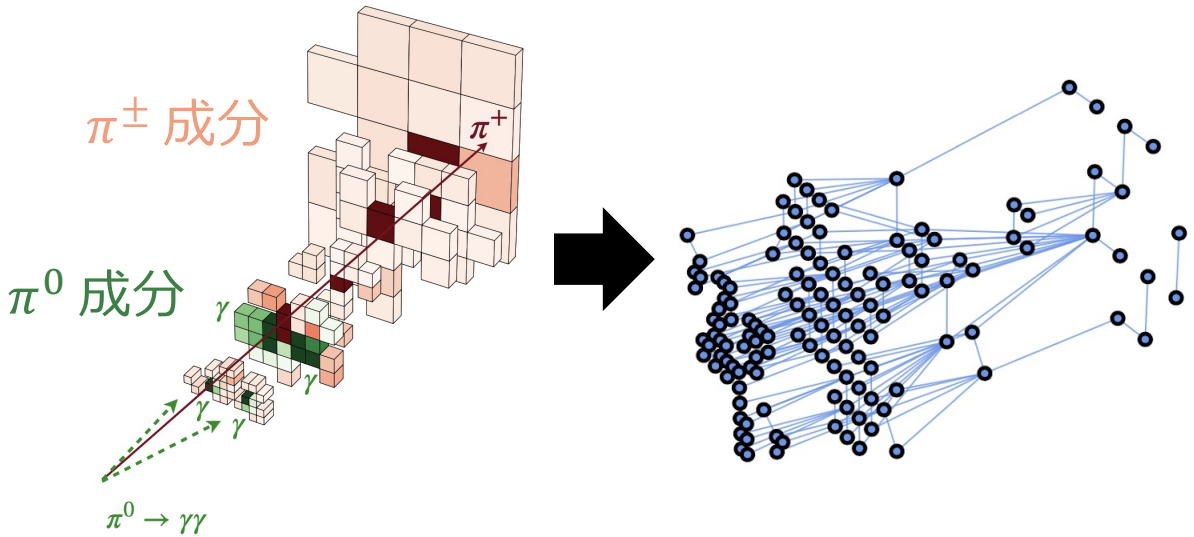
Jet category	DNN	GRU	CNN	JEDI-net	JEDI-net with $\sum O$
			TPR for FPR=10%		
gluon	0.830 ± 0.002	0.740 ± 0.014	0.700 ± 0.008	0.878 ± 0.001	$\mathbf{0.879 \pm 0.001}$
light quarks	0.715 ± 0.002	0.746 ± 0.011	0.740 ± 0.003	$\mathbf{0.822 \pm 0.001}$	0.818 ± 0.001
W boson	0.855 ± 0.001	0.812 ± 0.035	0.760 ± 0.005	$\mathbf{0.938 \pm 0.001}$	0.927 ± 0.001
Z boson	0.833 ± 0.002	0.753 ± 0.036	0.721 ± 0.006	$\mathbf{0.910 \pm 0.001}$	0.903 ± 0.001
top quark	0.917 ± 0.001	0.867 ± 0.006	0.889 ± 0.001	0.930 ± 0.001	$\mathbf{0.931 \pm 0.001}$
			TPR for FPR=1%		
gluon	0.420 ± 0.002	0.273 ± 0.018	0.257 ± 0.005	$\mathbf{0.485 \pm 0.001}$	0.482 ± 0.001
light quarks	0.178 ± 0.002	0.220 ± 0.037	0.254 ± 0.007	$\mathbf{0.302 \pm 0.001}$	0.301 ± 0.001
W boson	0.656 ± 0.002	0.249 ± 0.057	0.232 ± 0.006	$\mathbf{0.704 \pm 0.001}$	0.658 ± 0.001
Z boson	0.715 ± 0.001	0.386 ± 0.060	0.291 ± 0.005	$\mathbf{0.769 \pm 0.001}$	0.729 ± 0.001
top quark	$\mathbf{0.651 \pm 0.003}$	0.426 ± 0.020	0.504 ± 0.005	0.633 ± 0.001	0.632 ± 0.001

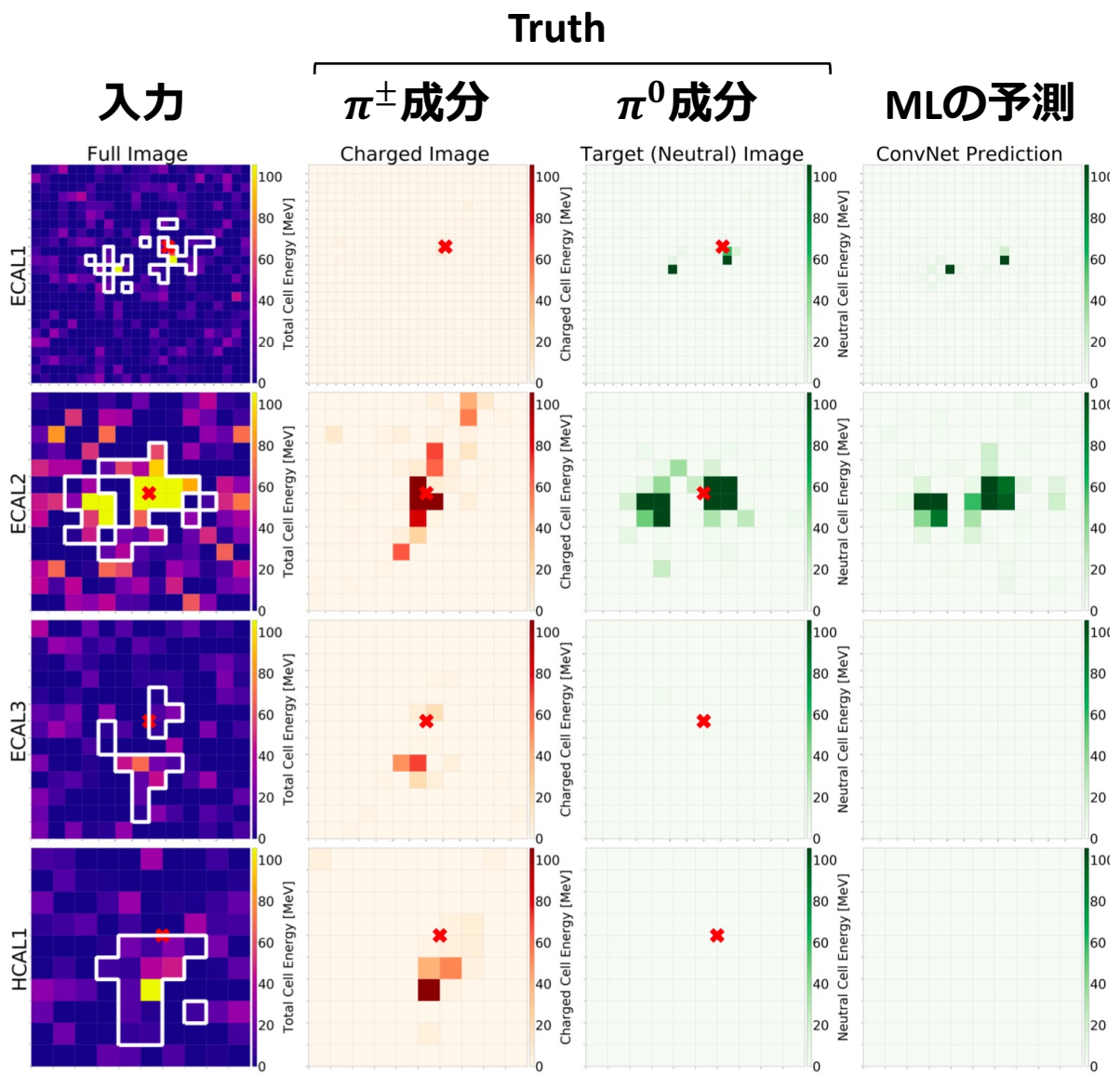
グラフネットワークのHEPでの応用例

EdgeConv for particle flow: 2003.08863, 2003.11603
(EdgeConv: MPNNの一種(PointCloud, ParticleNet))

- トラック・カロリメータクラスターを使って
  ジェット中の $\pi^{\pm}$ ・π^{0} の分離

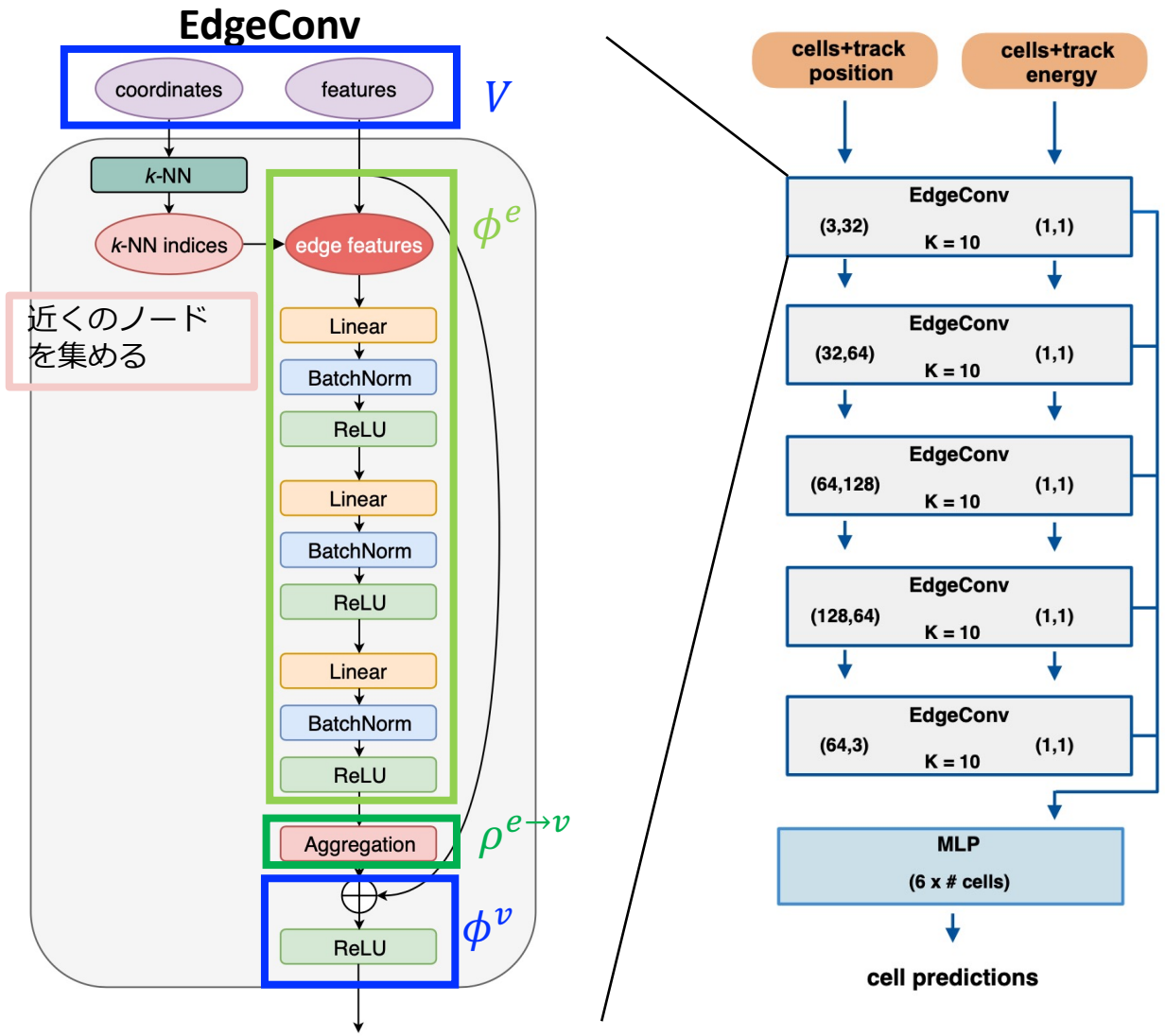
		対象	特徴量 (attribute)
入力	ノード	トラック・ Calo cluster	位置・エネルギー
出力	ノード	同上	π^0由来のエネルギー

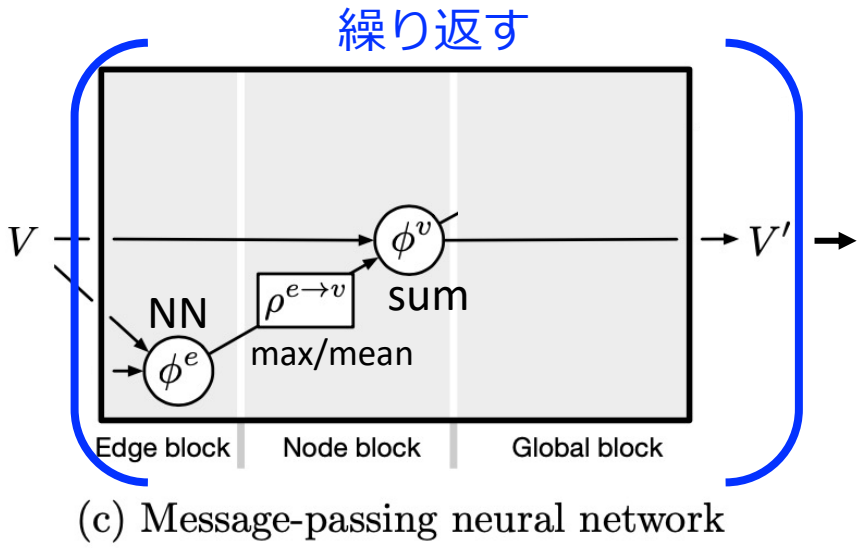




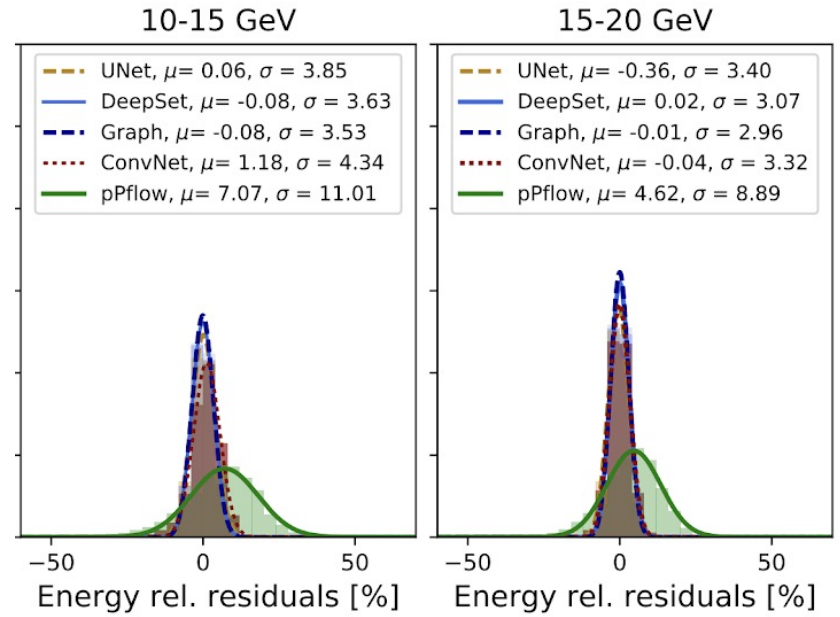
グラフネットワークのHEPでの応用例

EdgeConv for particle flow: 2003.08863, 2003.11603
(EdgeConv: MPNNの一種(PointCloud, ParticleNet))





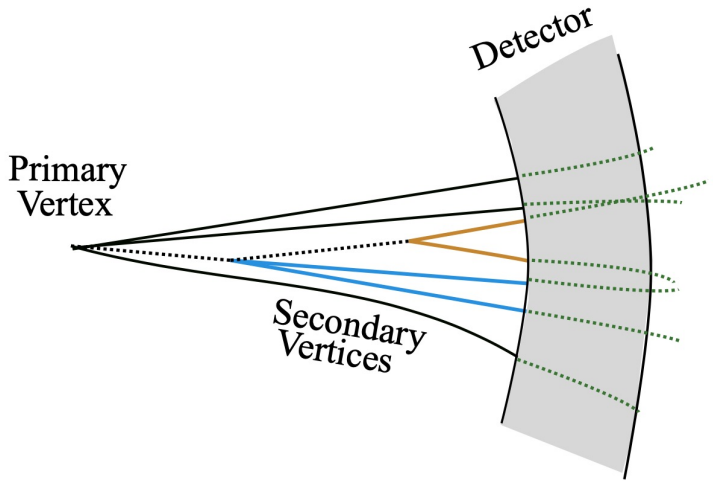
(c) Message-passing neural network

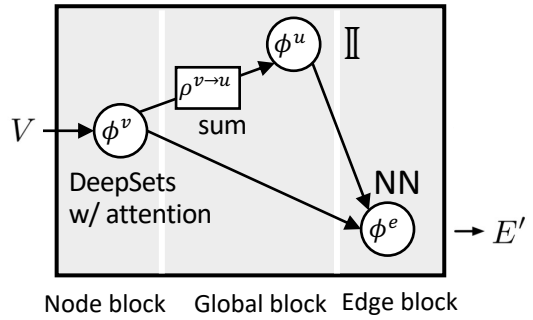


GNN > parametric model

グラフネットワークのHEPでの応用例

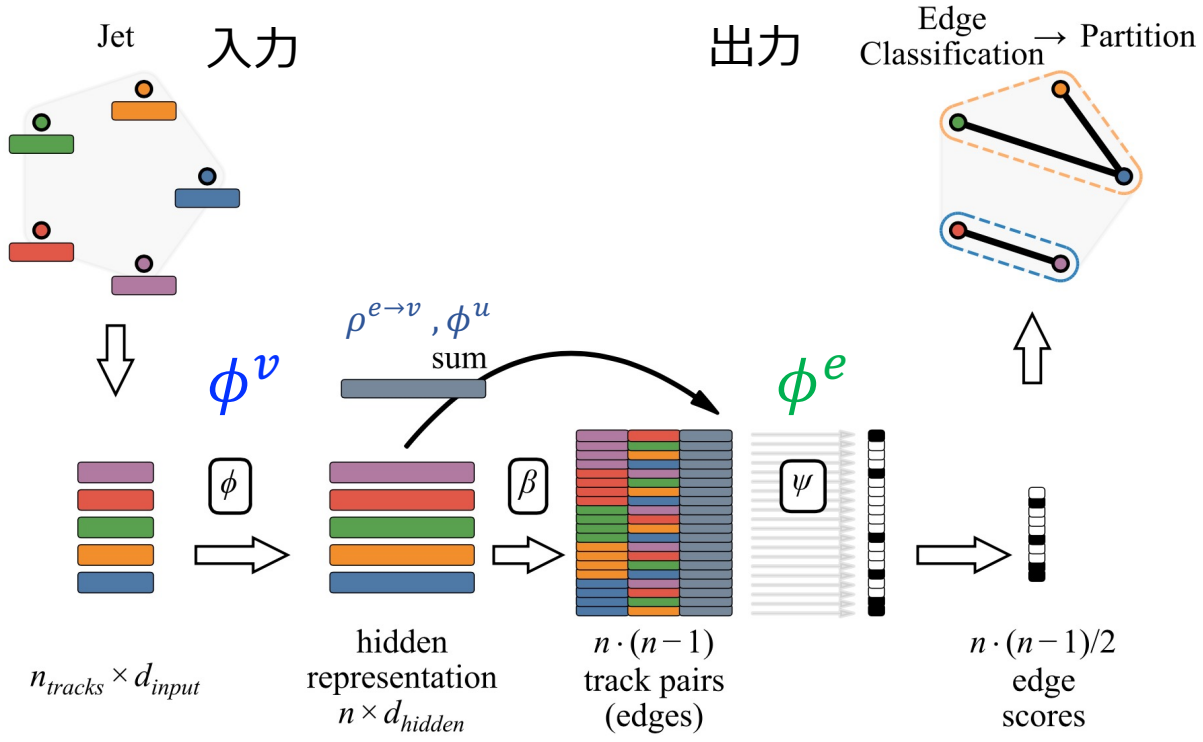
Set-to-graph for secondary vertex finding: [2008.02831]





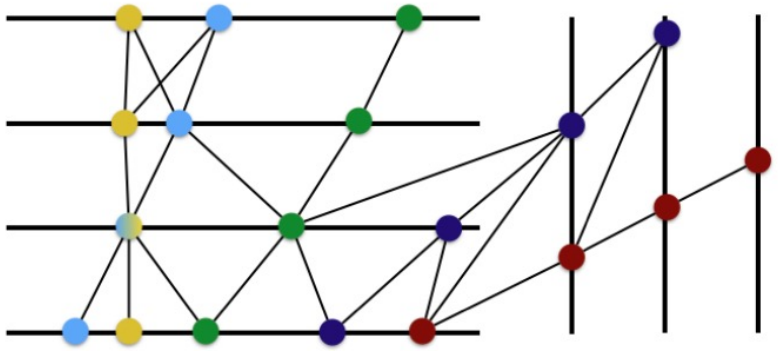
- ジェット中のトラックを
  primary vertex, secondary vertexに分類

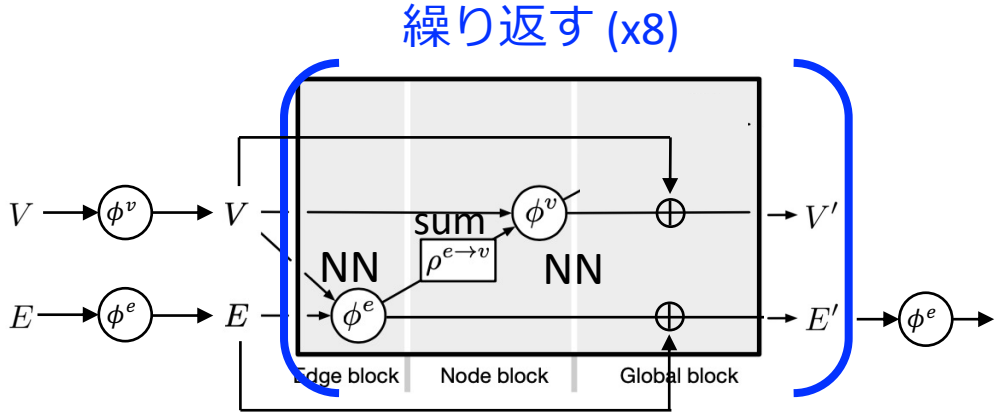
		対象	特徴量 (attribute)
入力	ノード	トラック	pT, eta, phi, d0, z0, q
出力	エッジ	トラック対の関係	同じvertex由来である確率

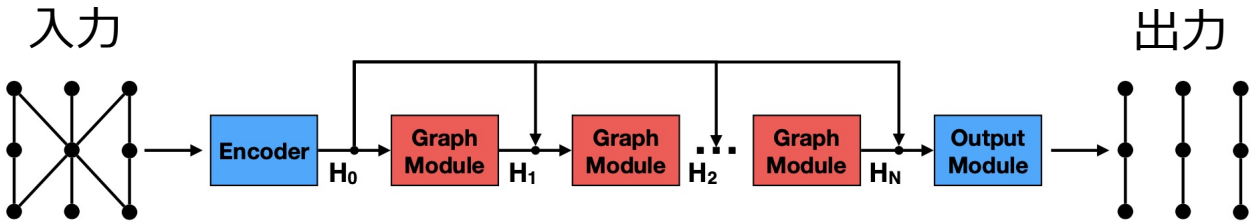


グラフネットワークのHEPでの応用例

GNN for Tracking: [2003.11603]

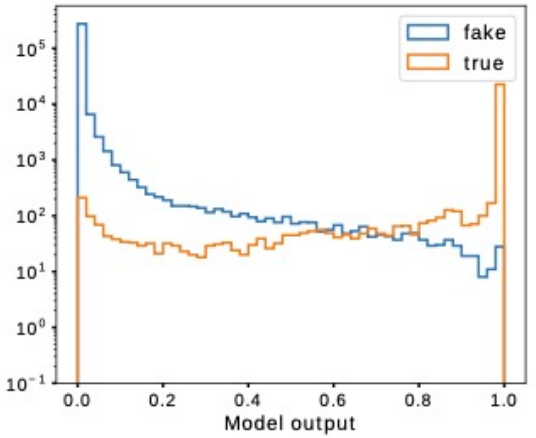






- ヒット(3次元の点)からトラックを再構成する

		対象	特徴量 (attribute)
入力	ノード	ヒット	位置(x, y, z)
	エッジ	ヒット間関係	$\Delta\eta, \Delta\phi$
出力	エッジ	同上	同じ粒子由来である確率



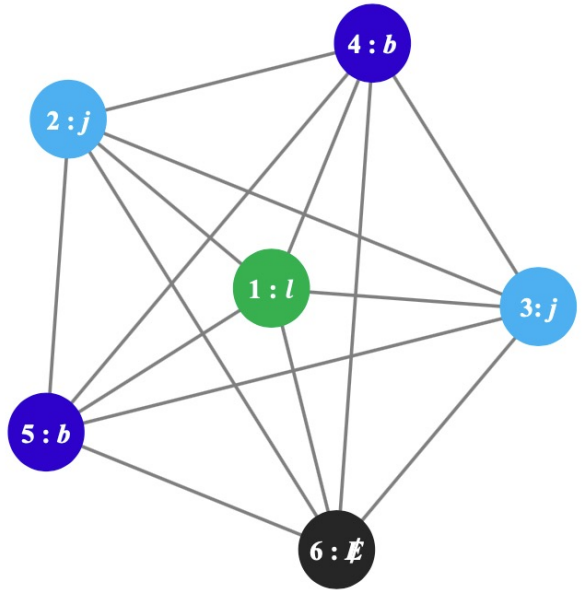
グラフネットワークのHEPでの応用例

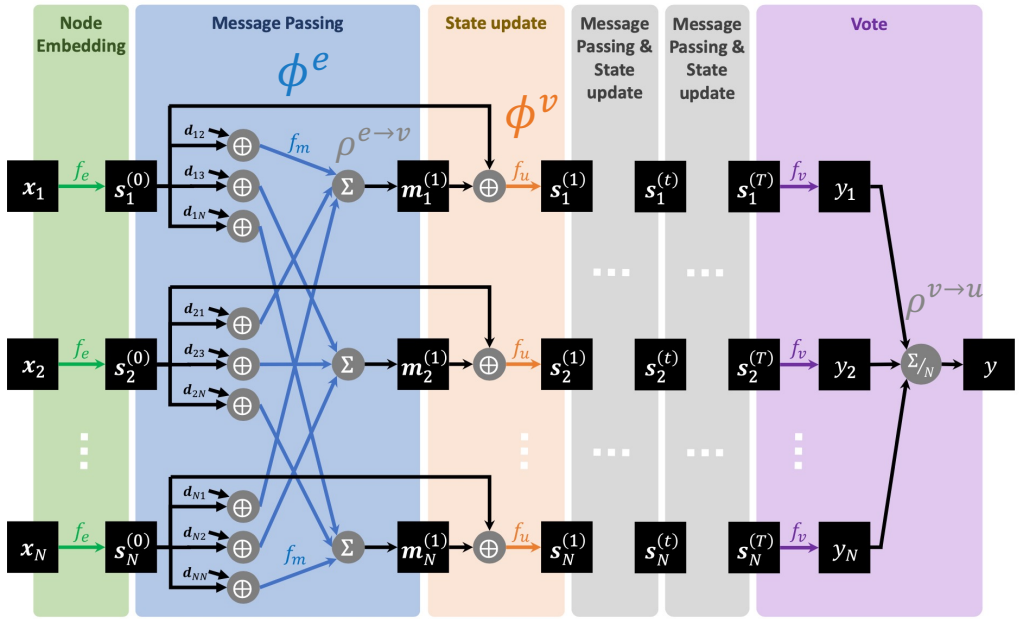
$\tilde{t}_1\tilde{t}_1^* \to t\bar{t}\,\tilde{\chi}_1^0\tilde{\chi}_1^0 \to 2b + 2j + \ell + E_T^{miss}$

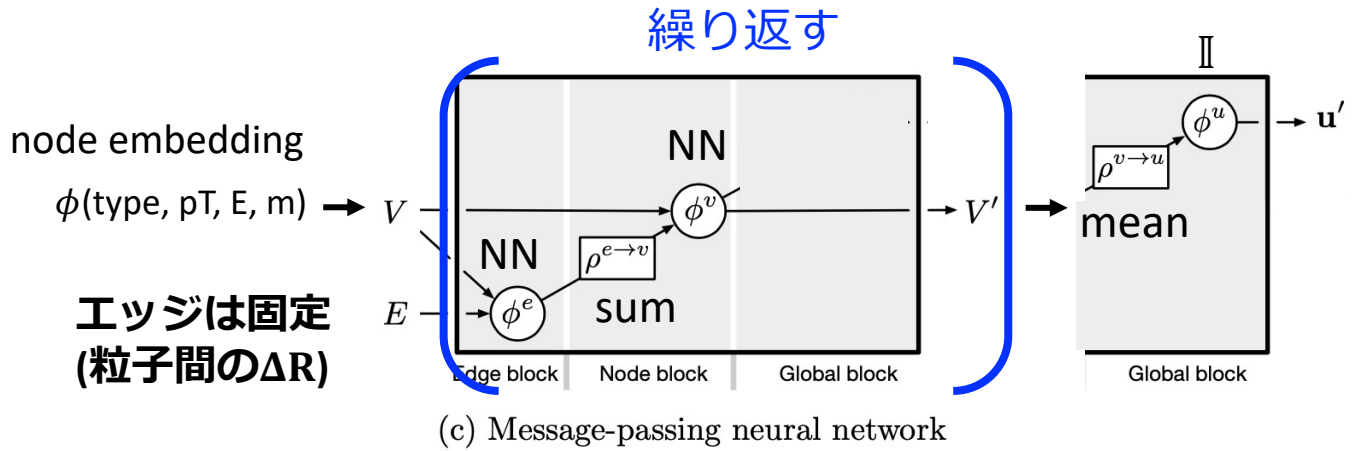
Message Passing Neural Network for stop search: JHEP08(2019)055

- Supersymmetric top 探索

		対象	特徴量 (attribute)
入力	ノード	粒子	粒子種類(jet, lepton, MET), 4-vector
	エッジ	粒子間関係	ΔR
出力	グローバル	ラベル	シグナルである確率







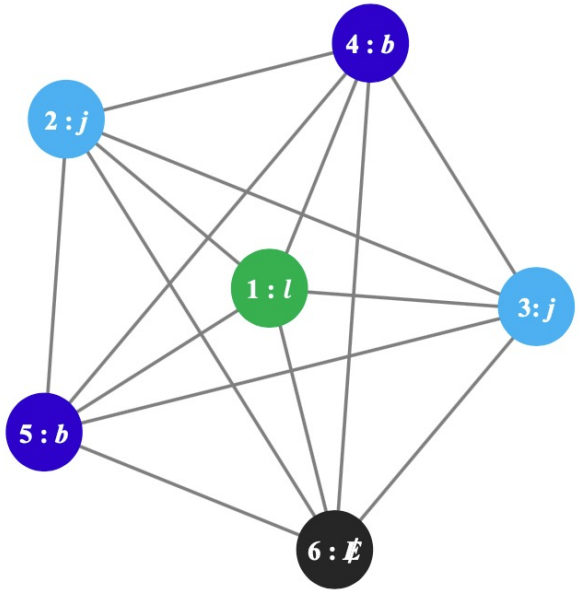
エッジは固定 (粒子間のΔR)

(c) Message-passing neural network

グラフネットワークのHEPでの応用例

$\tilde{t}_1\tilde{t}_1^* \rightarrow t\bar{t}\,\tilde{\chi}_1^0\tilde{\chi}_1^0 \rightarrow 2b + 2j + \ell + E_T^{miss}$

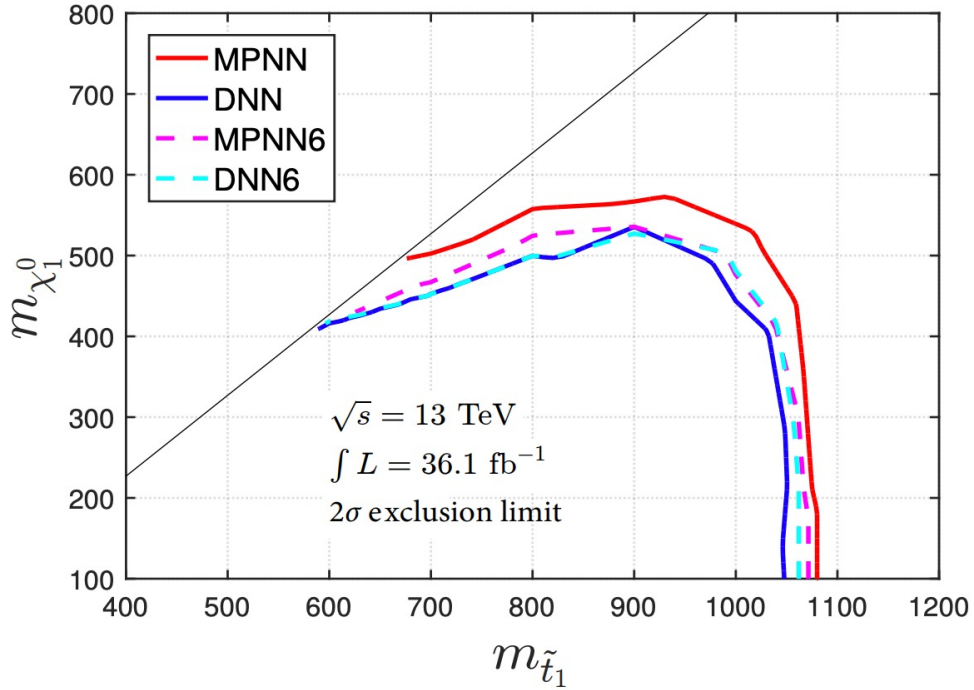
Message Passing Neural Network for stop search: JHEP08(2019)055



- Supersymmetric top 探索

- 入力:
  - ノード: 粒子(Jet, lepton, MET, etc.). 粒子タイプや4-vectorが特徴量
  - エッジ: 粒子間のΔR
- 出力:
  - グローバル: イベントがシグナルである確率

GNN vs DNN (MLP)



GNN > DNN

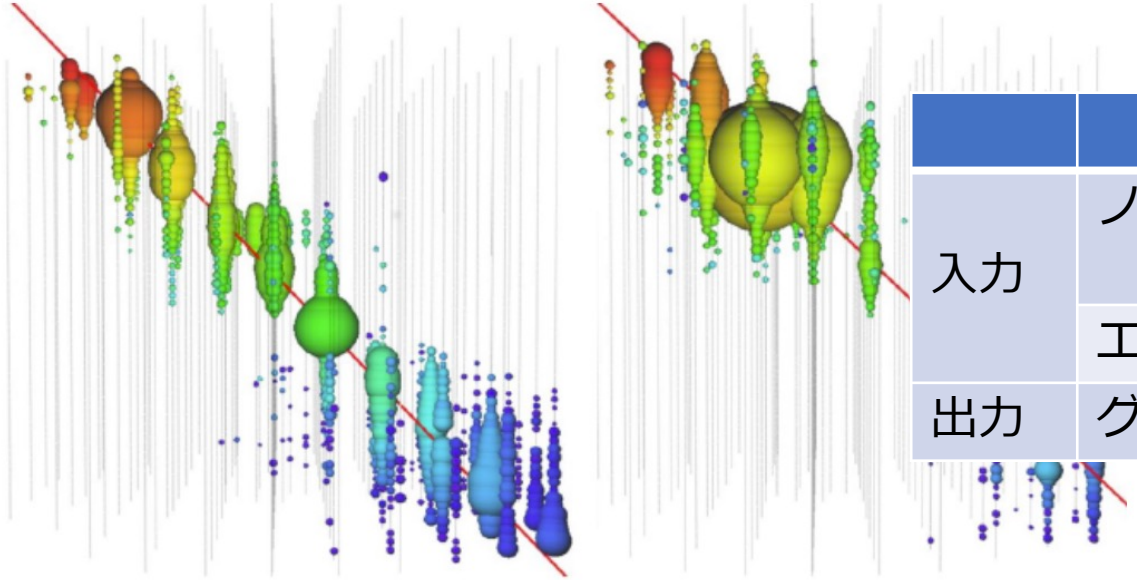
他のグラフネットワークをイベント分類に用いた研究例

グラフネットワークを用いたIceCubeイベント解析

1809.06166

Background:
muon from cosmic shower

Signal:
muon from neutrino

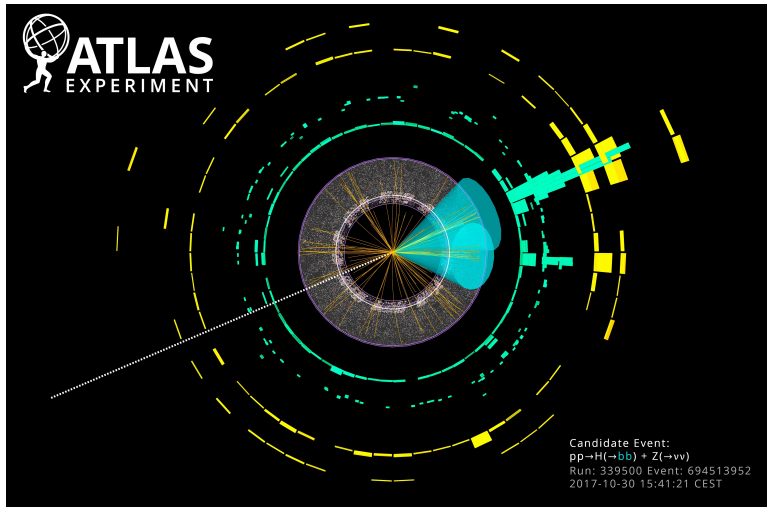


		対象	特徴量 (attribute)
入力	ノード	センサー	位置、パルスの電荷量、パルス到達時間
	エッジ	センサー間距離	３次元距離を基にした量
出力	グローバル	ラベル	シグナルである確率

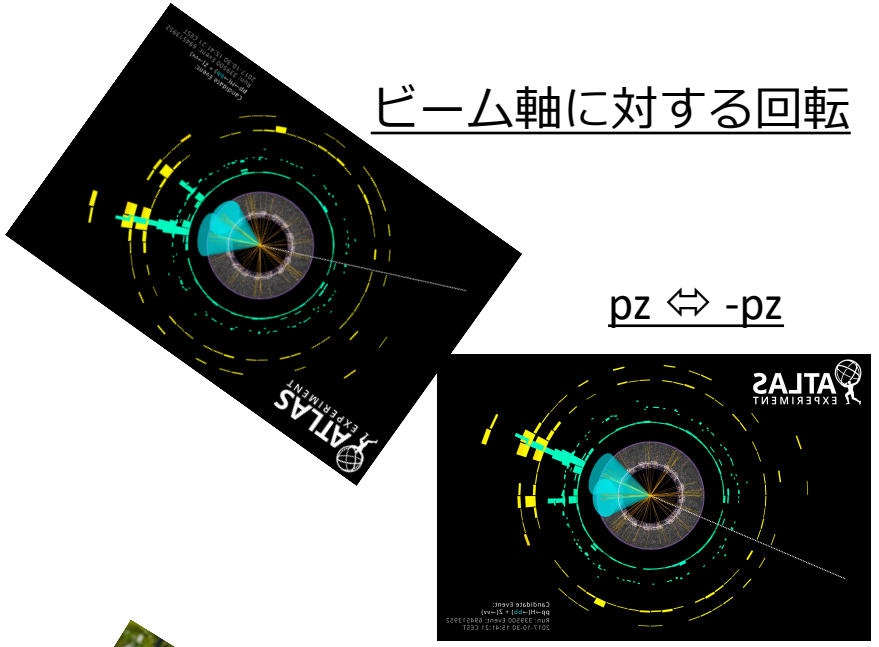
- センサー間の距離を基にグラフを定義
- 各センサーが各ノードに対応。特徴量はチャージ量など。
- 反応があるセンサーだけをノードにすることで、モデルサイズが削減
- グラフを用いることで、イレギュラーなジオメトリに対応

Data Augmentation

- ドメイン知識:
  - 物理法則や検出器は空間的な回転に対して不変である



同じ確率で起こる



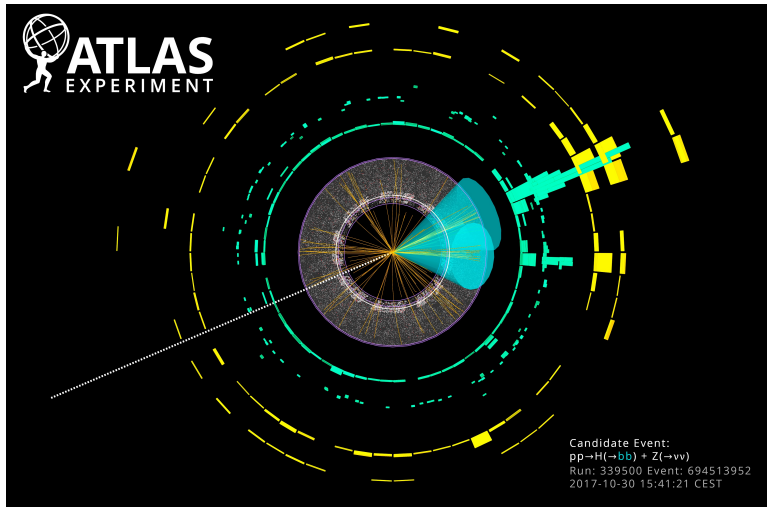
ビーム軸に対する回転

pz ⇔ -pz

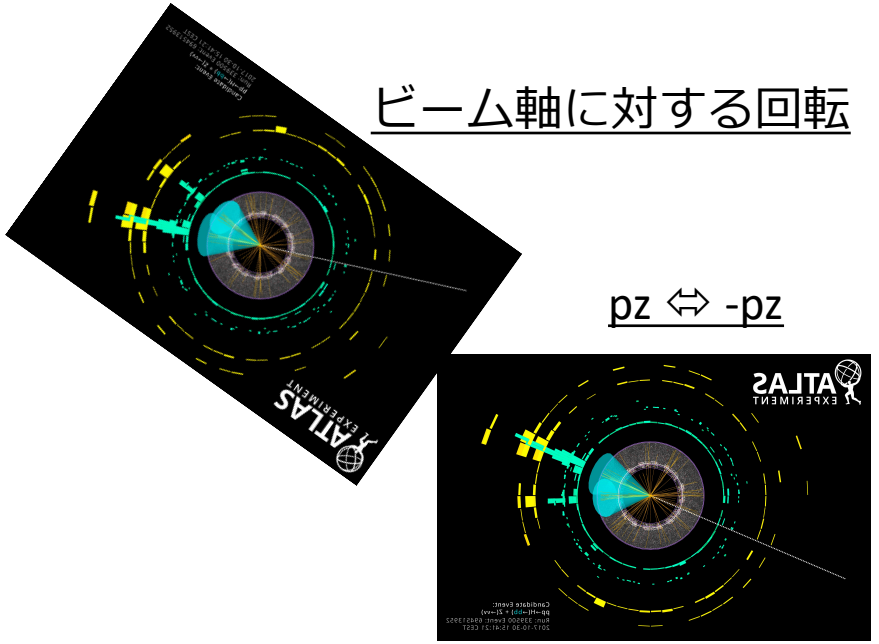
人間は画像を回転させても対象を見分けることができる

Data Augmentation

- ドメイン知識:
  - 物理法則や検出器は空間的な回転に対して不変である



同じ確率で起こる



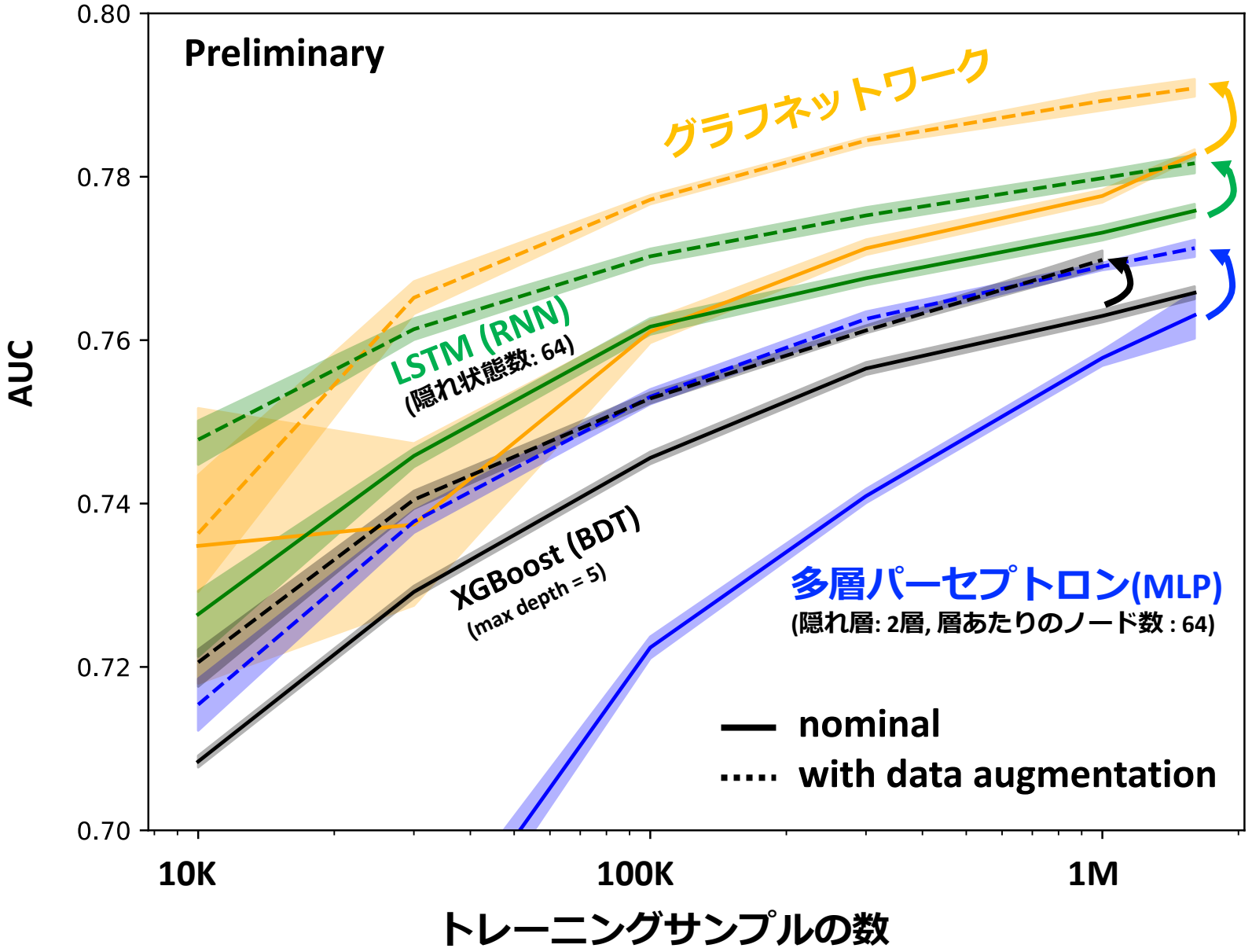
ビーム軸に対する回転

pz ⇔ -pz

画像認識分野におけるデータの水増し手法と同様に、各イベントの運動量を対称性に従ってランダムに変換し、モデルのトレーニングに使う

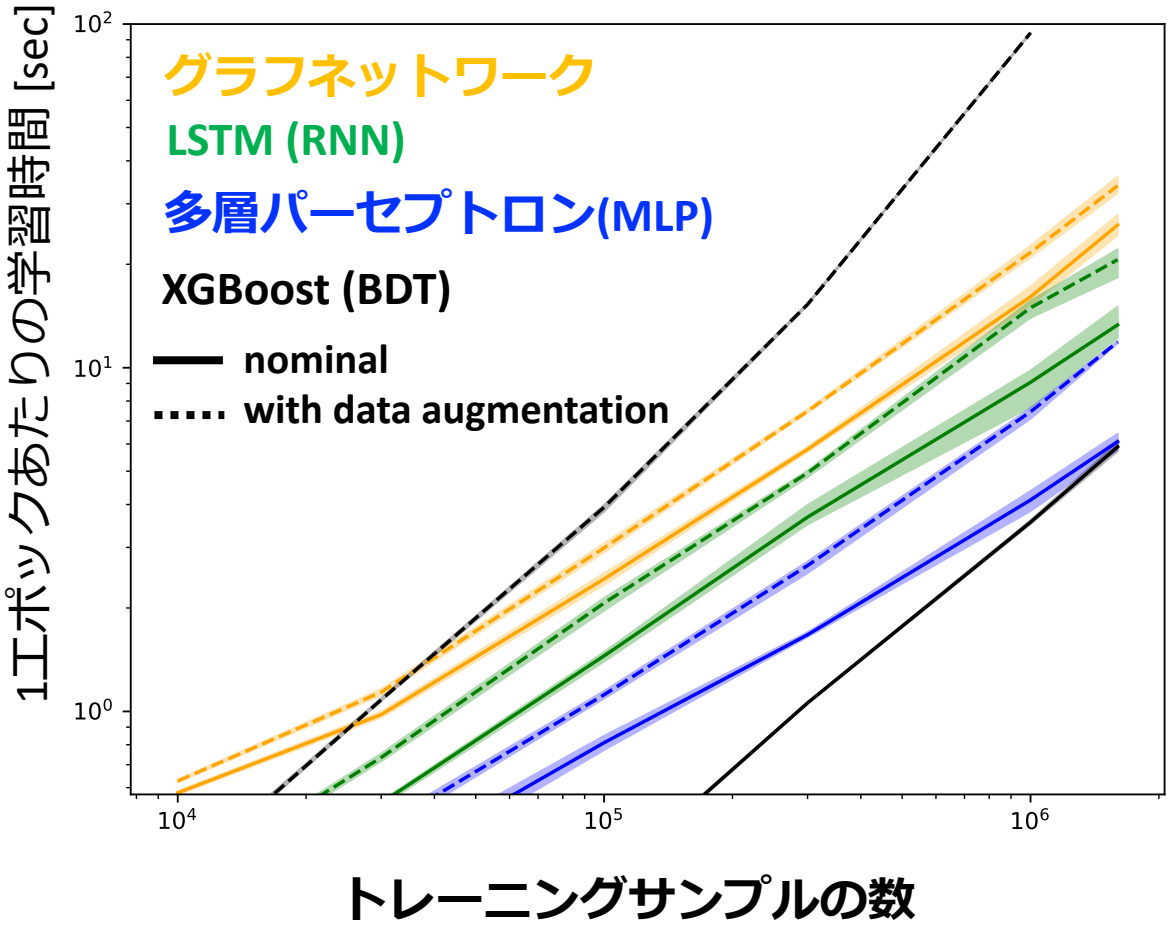
$$\begin{pmatrix} E \\ P_x \\ P_y \\ P_z \end{pmatrix} \rightarrow \begin{pmatrix} E \\ P_x\cos\theta - P_y\sin\theta \\ P_x\sin\theta + P_y\cos\theta \\ P_z \end{pmatrix} \qquad \begin{pmatrix} E \\ P_x \\ P_y \\ P_z \end{pmatrix} \rightarrow \begin{pmatrix} E \\ \pm P_x \\ \pm P_y \\ \pm P_z \end{pmatrix}$$

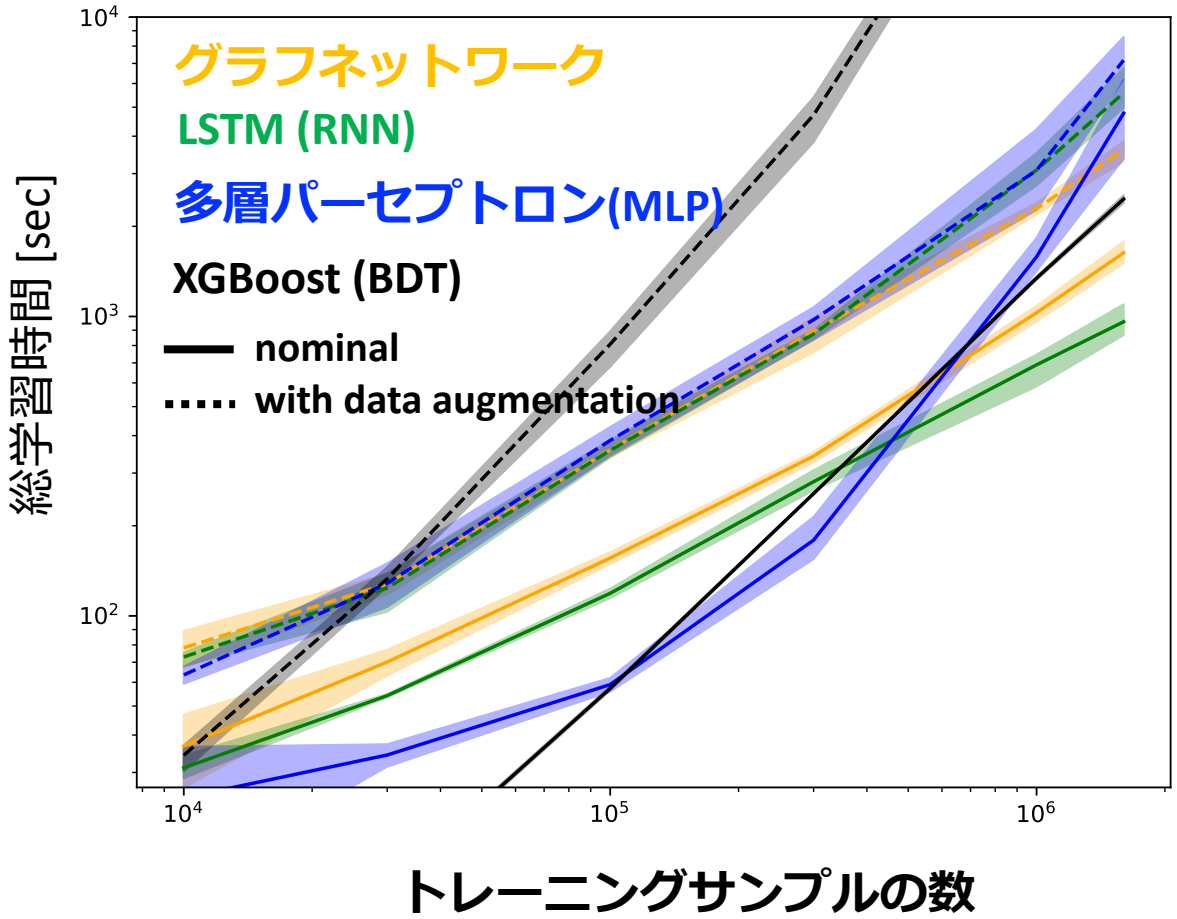
結果 with data augmentation



- データの水増しは**物理解析においても非常に有効**
  - データ量が実効的に増加
  - 過学習を抑制
  - トレニーニングサンプル数が少ないときのMLPで顕著

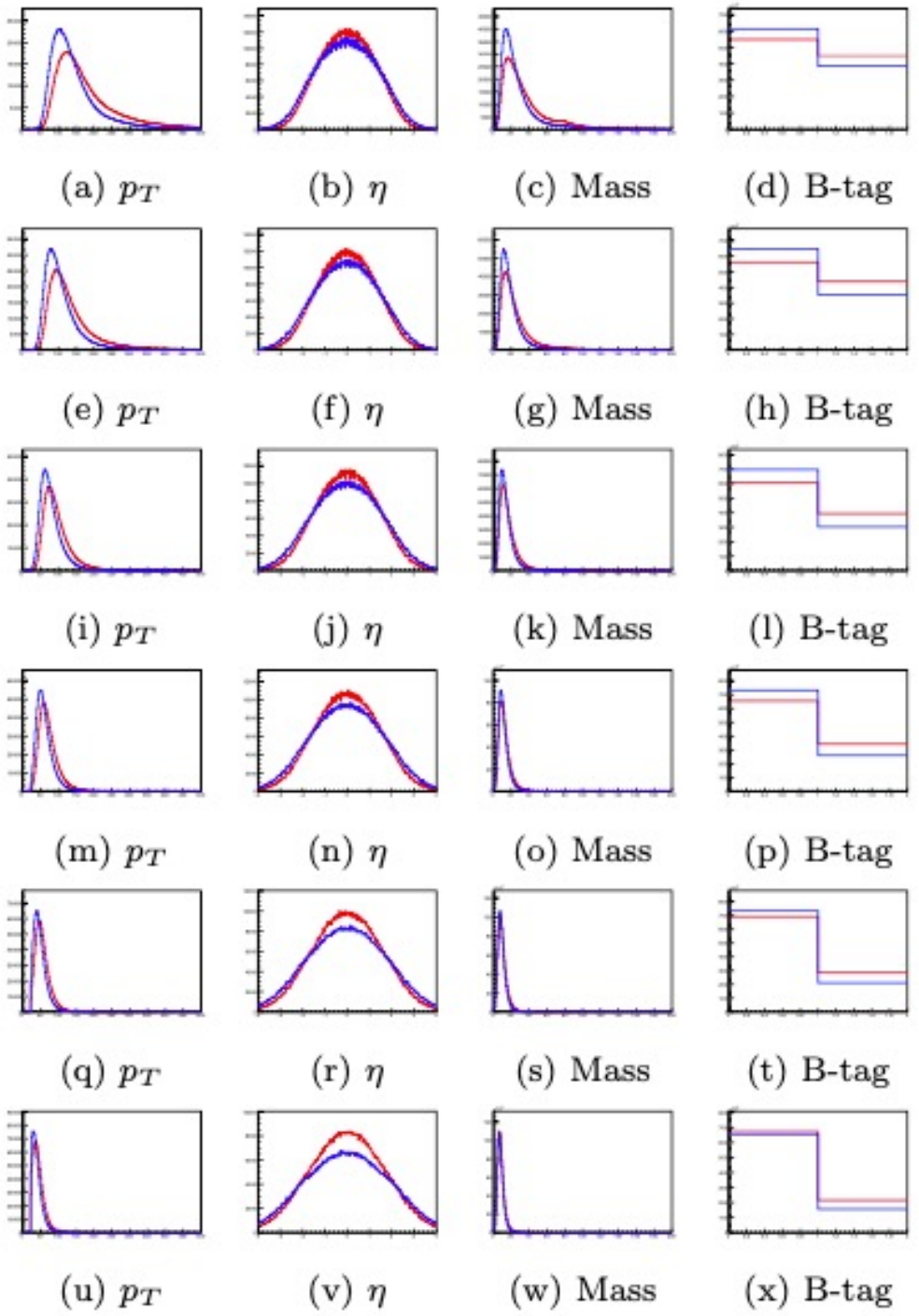
各モデルの学習時間





- 学習にはTesla V100 を使用

入力変数の分布


(a) p_T
(b) η
(c) Mass
(d) B-tag
(e) p_T
(f) η
(g) Mass
(h) B-tag
(i) p_T
(j) η
(k) Mass
(l) B-tag
(m) p_T
(n) η
(o) Mass
(p) B-tag
(q) p_T
(r) η
(s) Mass
(t) B-tag
(u) p_T
(v) η
(w) Mass
(x) B-tag