

Лекция 14. Графовые нейронные сети

Храбров Кузьма

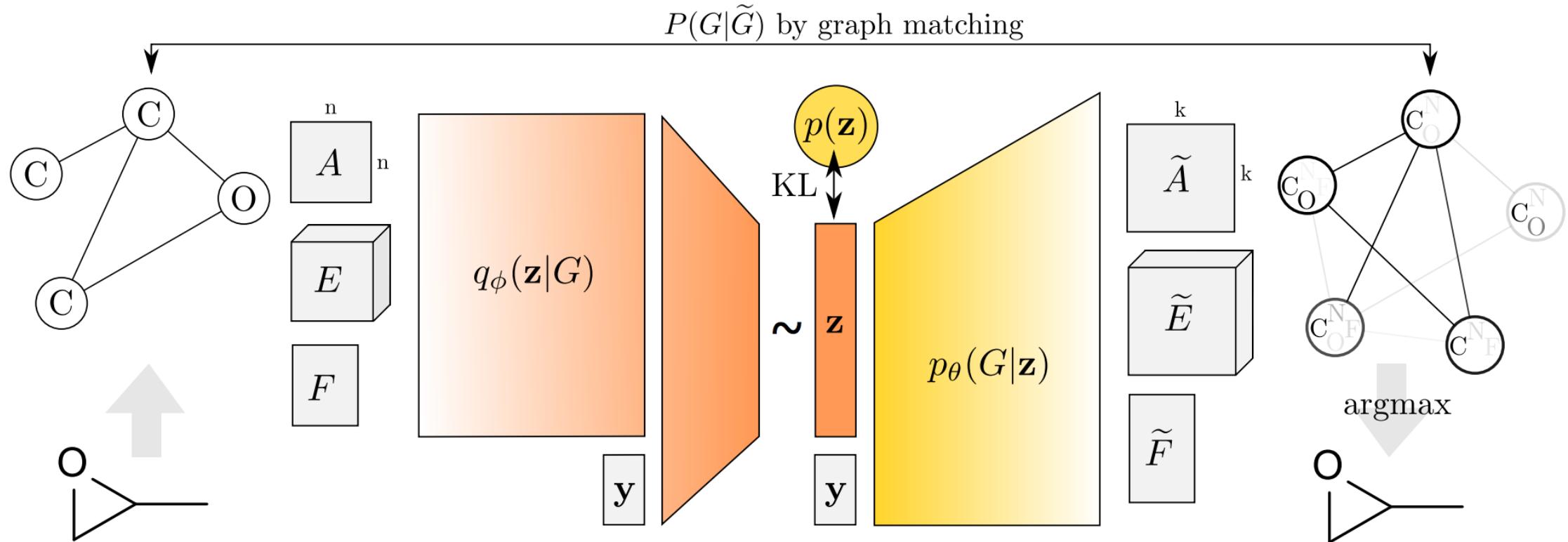


Содержание

- 01 Граф как объект
- 02 Message passing
- 03 Graph convolution
- 04 Graph attention
- 05 Задачи и проблемы



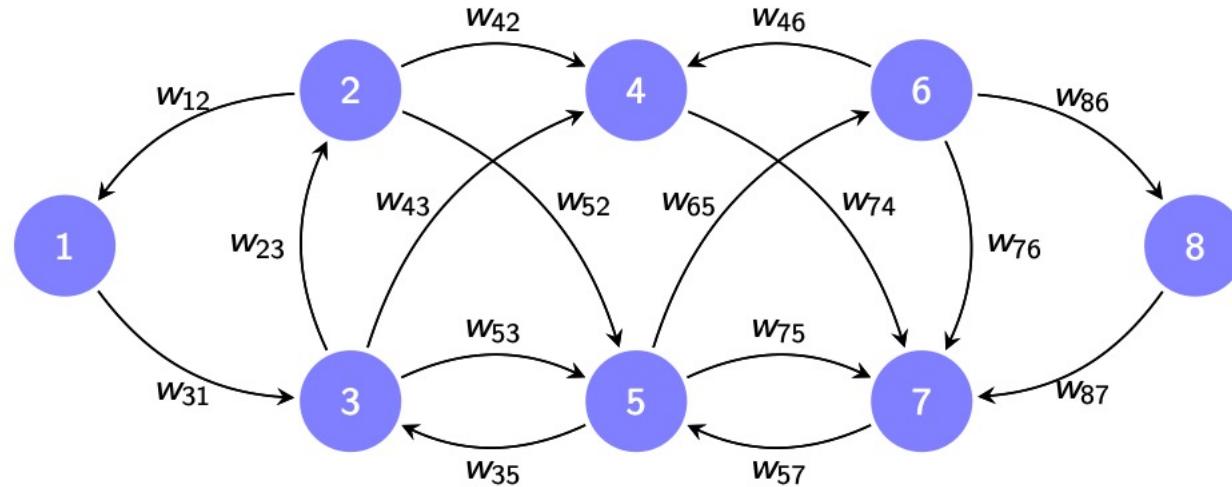
Графы в машинном обучении



Граф можно представить в виде матрицы смежности, матрицы признакового описания вершин и тензора с признаками ребер.

Графы в машинном обучении

Граф это тройка (V, E, W) , где V – множество вершин, E – множество ребер (упорядоченных пар вершин), а W – матрица весов.



Примеры графов и задач

ДАННЫЕ

Изображения, тексты

Социальные сети

Графы знаний

Молекулы

3D изображения / видео



| | Graphs | are | all | around | us |
|--------|--------|-----|-----|--------|----|
| Graphs | | | | | |

ЗАДАЧИ

Предсказание на уровне вершин
(например, node classification)

Предсказание на уровне пар вершин
(например, предсказание недостающих
ребер)

Предсказание на уровне графа
(например, классификация графов)

Генерация / Кластеризация

Примеры графов и задач

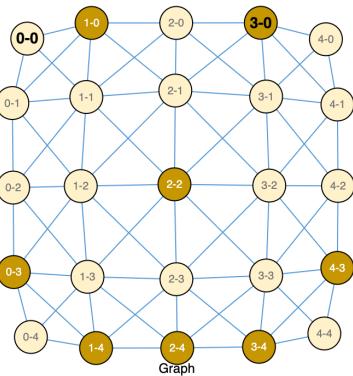
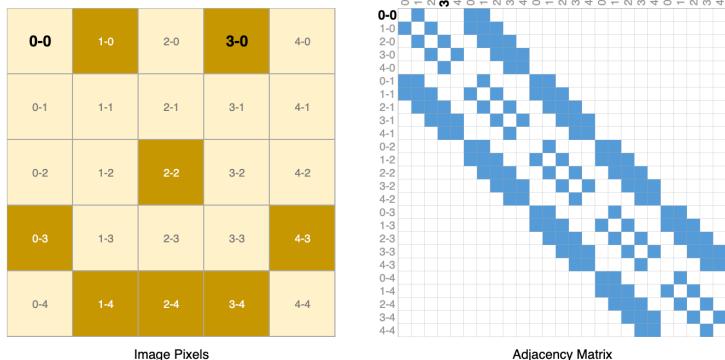
ДАННЫЕ

Изображения, тексты

Социальные сети

Графы знаний

Молекулы



ЗАДАЧИ

Предсказание на уровне вершин (например, node classification)

Предсказание на уровне пар вершин
(например, предсказание недостающих
ребер)

Предсказание на уровне графа
(например, классификация графов)

Генерация / Кластеризация

Примеры графов и задач

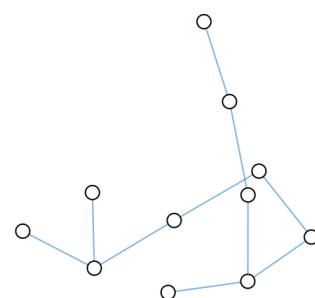
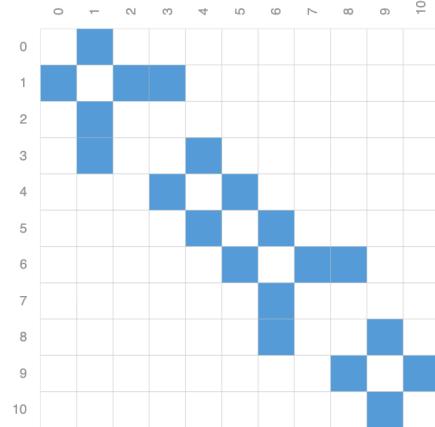
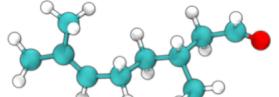
ДАННЫЕ

Изображения, тексты

Социальные сети

Графы знаний

Молекулы



ЗАДАЧИ

Предсказание на уровне вершин
(например, node classification)

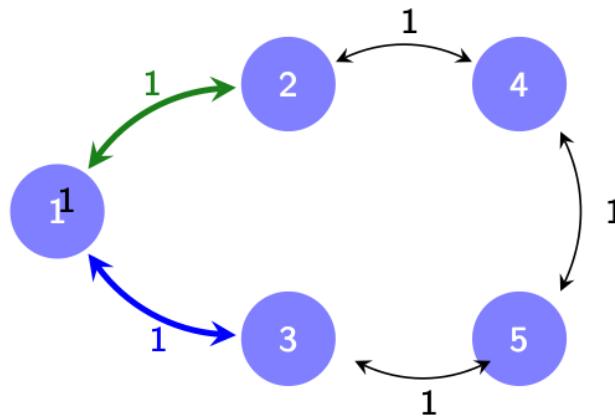
Предсказание на уровне пар вершин
(например, предсказание недостающих ребер)

Предсказание на уровне графа
(например, классификация графов)

Генерация / Кластеризация

Графы в машинном обучении

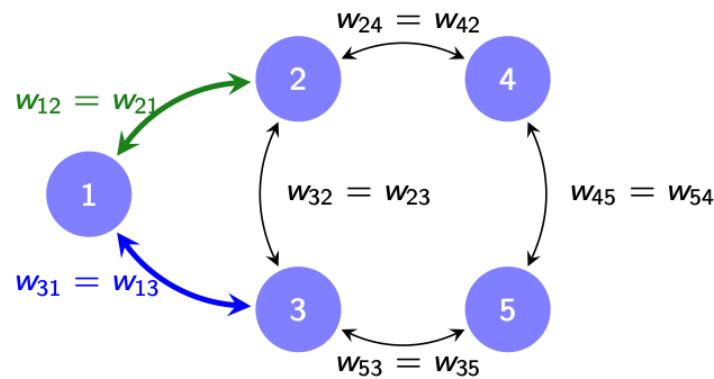
Матрица смежности $A_{ij} = 1$, если $(i, j) \in E$ и 0 иначе



$$\mathbf{A} = \begin{bmatrix} 0 & \textcolor{green}{1} & \textcolor{blue}{1} & 0 & 0 \\ \textcolor{blue}{1} & 0 & 1 & 1 & 0 \\ \textcolor{blue}{1} & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Графы в машинном обучении

Матрица смежности $A_{ij} = W_{ij}$, если $(i, j) \in E$ и 0 иначе

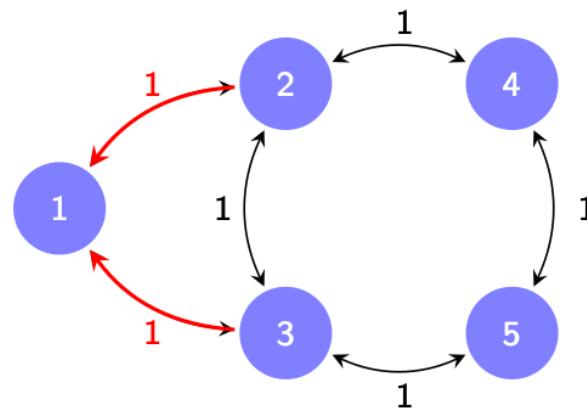


$$\mathbf{A} = \begin{bmatrix} 0 & w_{12} & w_{13} & 0 & 0 \\ w_{21} & 0 & w_{23} & w_{24} & 0 \\ w_{31} & w_{32} & 0 & 0 & w_{35} \\ 0 & w_{42} & 0 & 0 & w_{45} \\ 0 & 0 & w_{53} & w_{54} & 0 \end{bmatrix}.$$

Графы в машинном обучении

Степень вершины i : $d_i = \sum_{j \in n(i)} w_{ij} = \sum_{j: (i,j) \in \mathcal{E}} w_{ij}$

Матрица степеней – диагональная матрица из d_i



$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Графы в машинном обучении

Матрица Лапласа (Лапласиан): $L := D - A$

$$\nabla^2 f(x, y) = f_{xx}(x, y) + f_{yy}(x, y)$$

Графы в машинном обучении

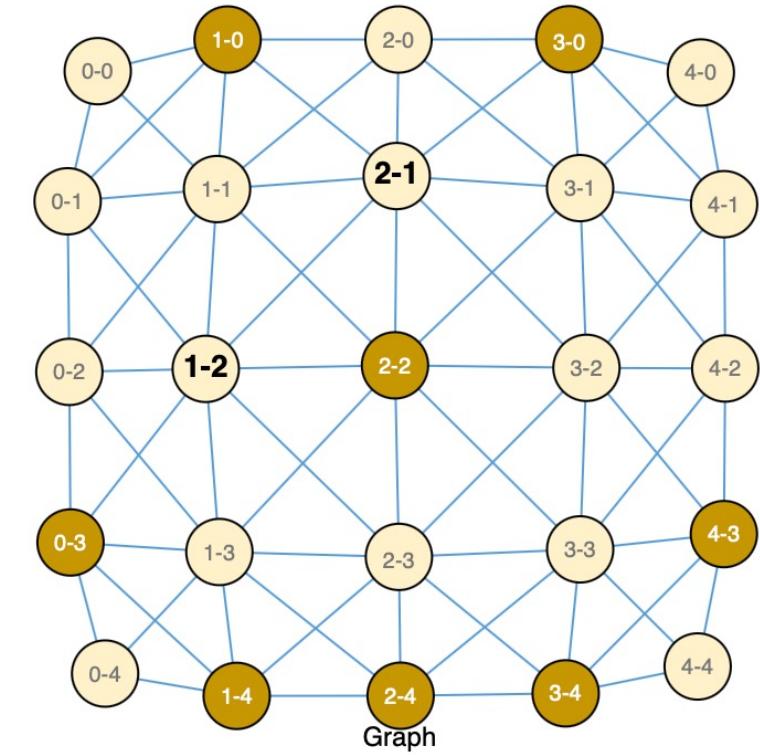
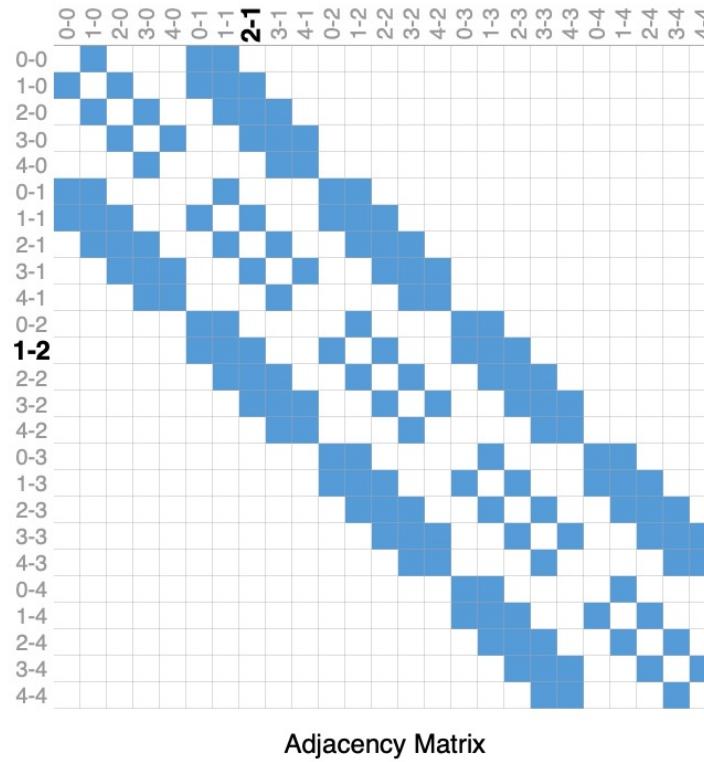
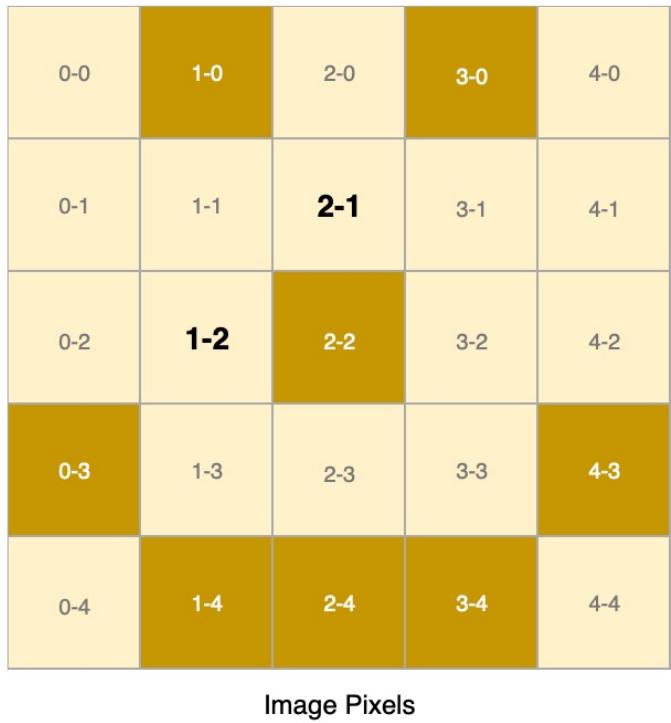
Матрица смежности A_{ij}

Введем матрицу признаков вершин графа и обозначим ее за x

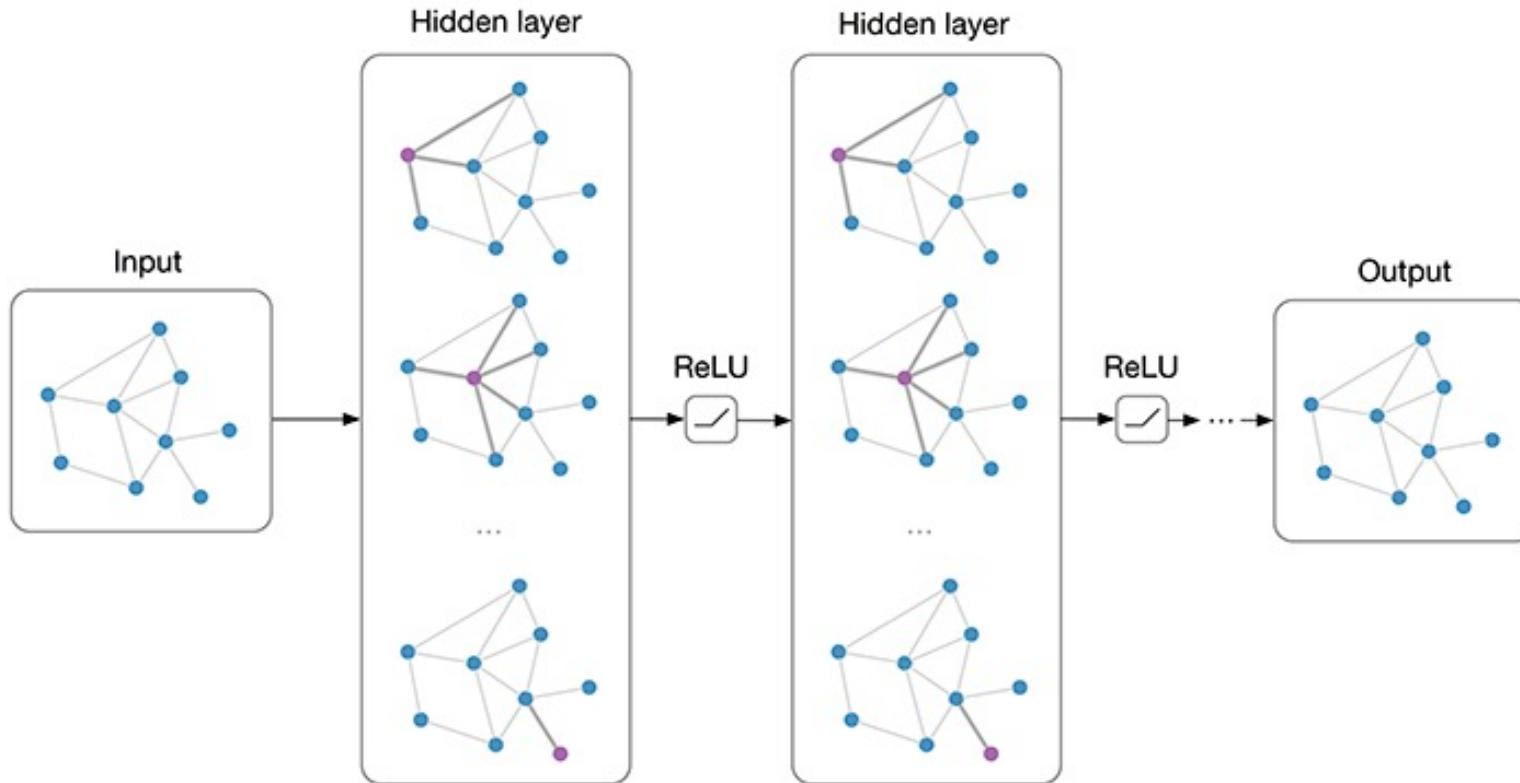
Посмотрим как разные степени A действуют на x

$$z = h_0 A^0 x + h_1 A^1 x + h_2 A^2 x + \dots = \sum_{k=0}^N h_k A^k x$$

Свертки на графах



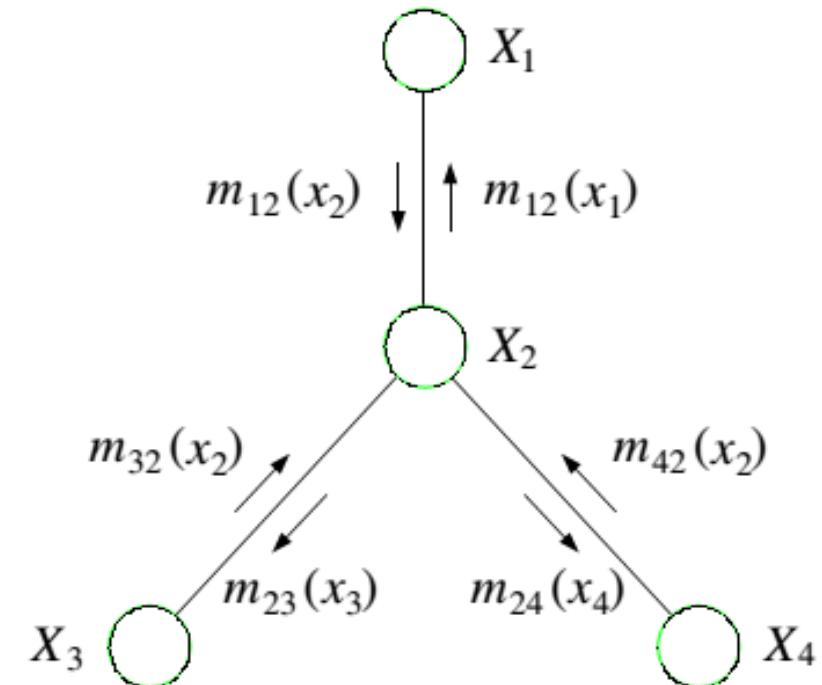
Свертки на графах



$$H^{(l+1)} = f(H^{(l)}, A) = \sigma(AH^{(l)}W)$$

Передача сообщений (Message passing)

Каждый шаг все вершины отправляют сообщения своим соседям, после чего обновляют свое состояние в соответствии с полученными новыми сообщениями.



Передача сообщений (Message passing)

По сути, сообщение — это состояние вершины и свойства ребра по которому сообщение передается. А состояния обновляются с помощью нейронной сети принимающей на вход сообщения и текущее состояние вершины.

$$h_v^{(k)} = \text{UPDATE}^{(k)} \left(h_v^{(k-1)}, \underbrace{\text{AGG}^{(k)}(\{h_u^{(k-1)}\}_{u \in N(v)})}_{=m_{N(v)}^{(k)}} \right),$$

AGG — функция агрегации
(например, сумма)

UPDATE — функция обновления
состояния вершины

Передача сообщений (Message passing).

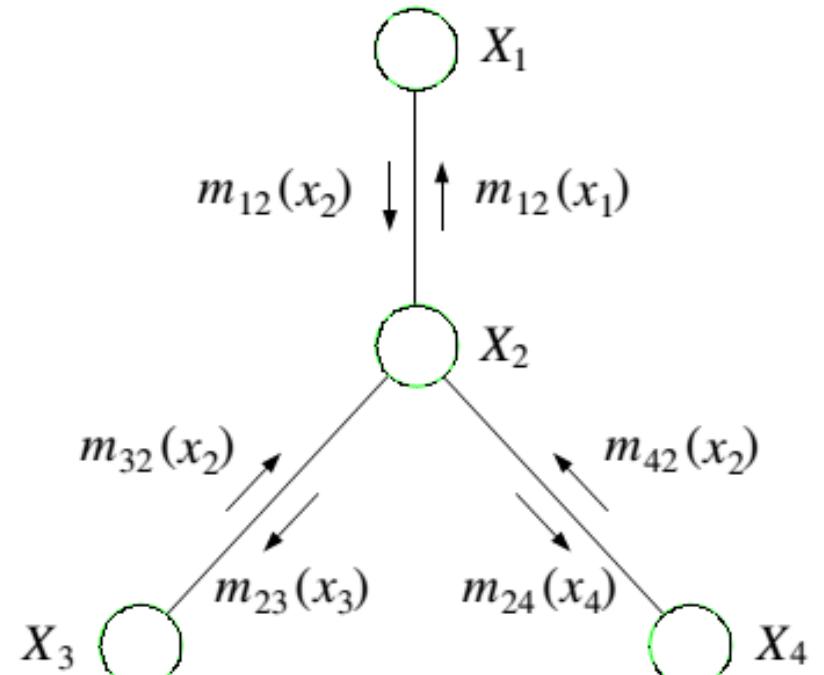
Варианты.

$$h_v^{(k)} = \sigma \left(W_0^{(k)} h_v^{(k-1)} + W_1^{(k)} \sum_{u \in N(v)} h_u^{(k-1)} + b^{(k)} \right),$$

$$\text{AGG}(\{h_u^{(k-1)}\}_{u \in N(v)}) = \max[\{W_{\text{pool}} h_u^{(k-1)} + b\}_{u \in N(v)}].$$

$$h_A^{(1)} = f \left(W^{(1)} \times \text{RNN} \left[h_C^{(0)}, h_E^{(0)} \right] + B^{(1)} \times h_A^{(0)} \right)$$

$$m_{N(v)}^{(k)} = f_\theta \left(\sum_{u \in N(v)} g_\phi(h_u^{(k-1)}) \right)$$



Передача сообщений (Message passing)

По сути, сообщение — это состояние вершины и свойства ребра по которому сообщение передается. А состояния обновляются с помощью нейронной сети принимающей на вход сообщения и текущее состояние вершины.

$$h_v^{(k)} = \text{UPDATE}^{(k)} \left(h_v^{(k-1)}, \underbrace{\text{AGG}^{(k)}(\{h_u^{(k-1)}\}_{u \in N(v)})}_{=m_{N(v)}^{(k)}} \right),$$
$$h_v^{(k)} = \sigma \left(W_0^{(k)} h_v^{(k-1)} + W_1^{(k)} \sum_{u \in N(v)} h_u^{(k-1)} + b^{(k)} \right),$$

Graph SAGE

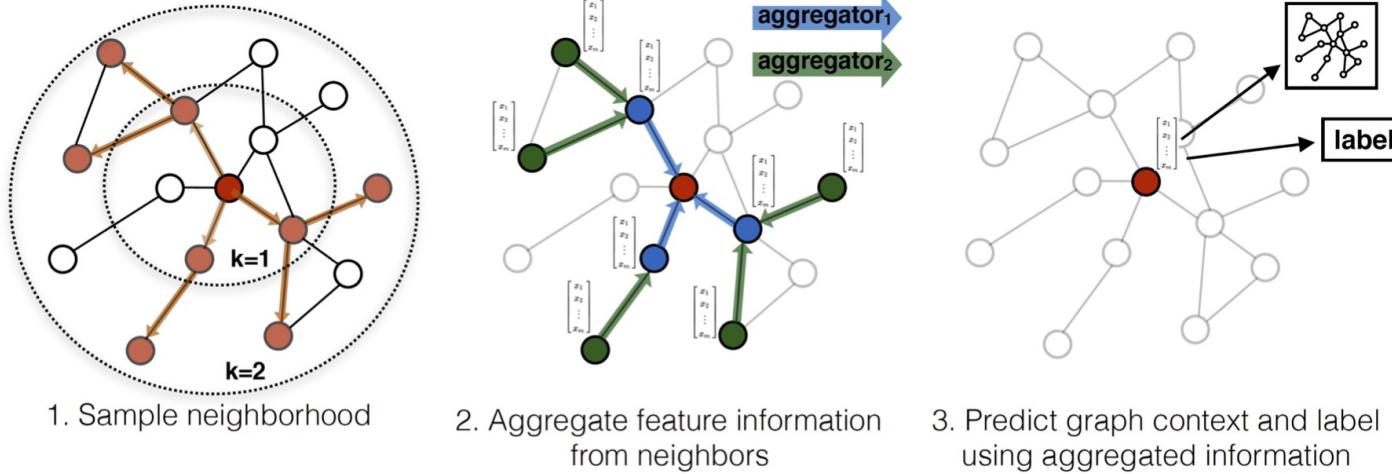
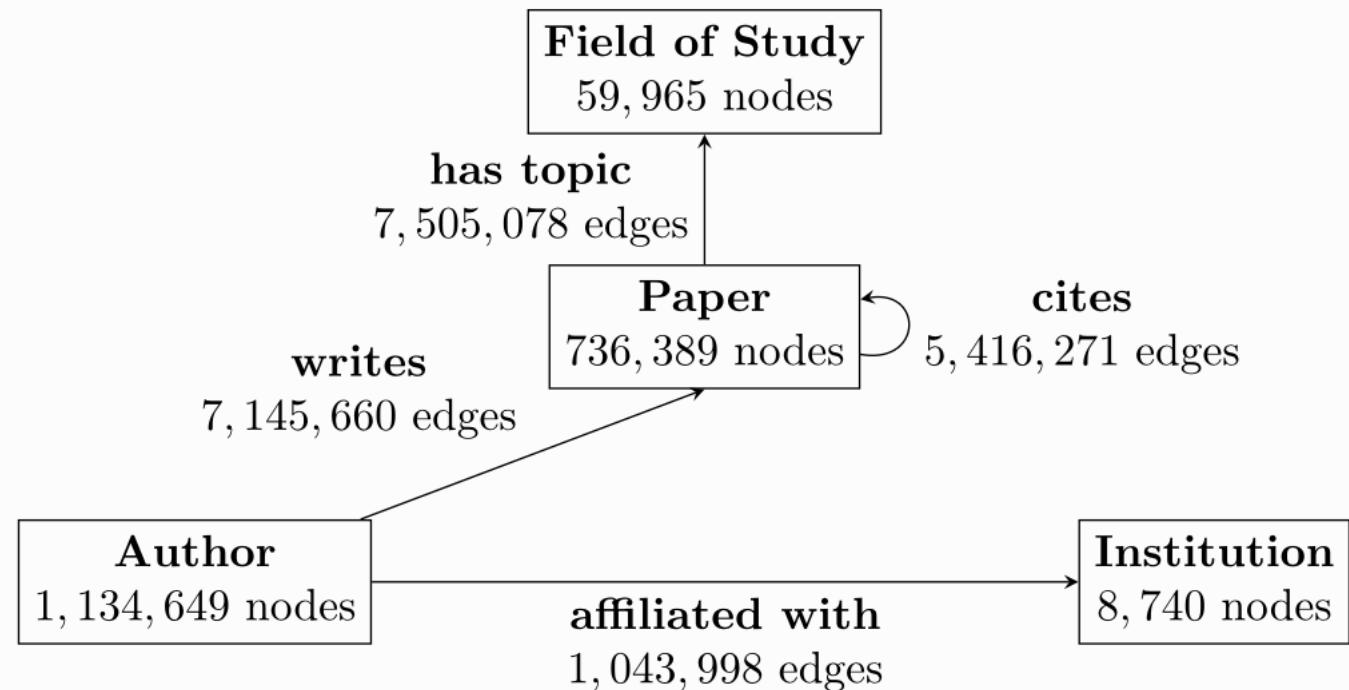


Figure 1: Visual illustration of the GraphSAGE sample and aggregate approach.

$$h_v^{(k)} = \sigma \left(W^{(k)} \left[\text{AGG}(\{h_u^{(k-1)}\}_{u \in N(v)}), h_v^{(k-1)} \right] \right),$$

Неоднородные графы

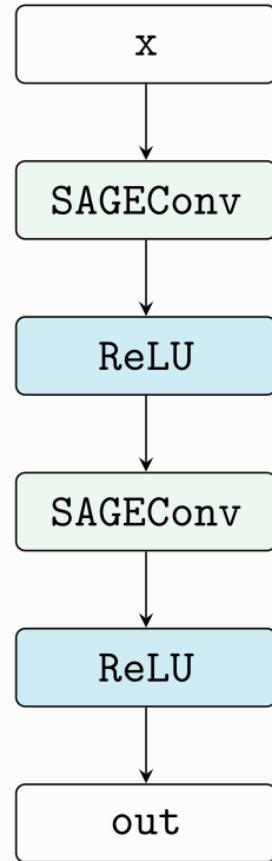
Dataset ogbn-mag



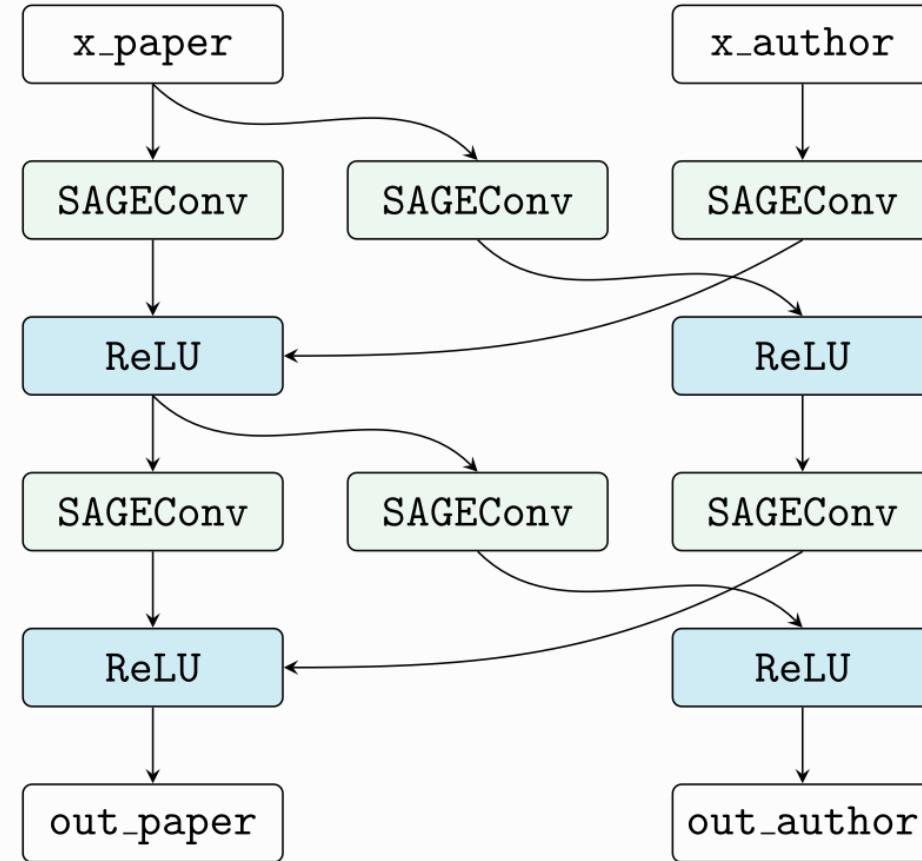
<https://ogb.stanford.edu/docs/nodeprop/>

Неоднородные графы

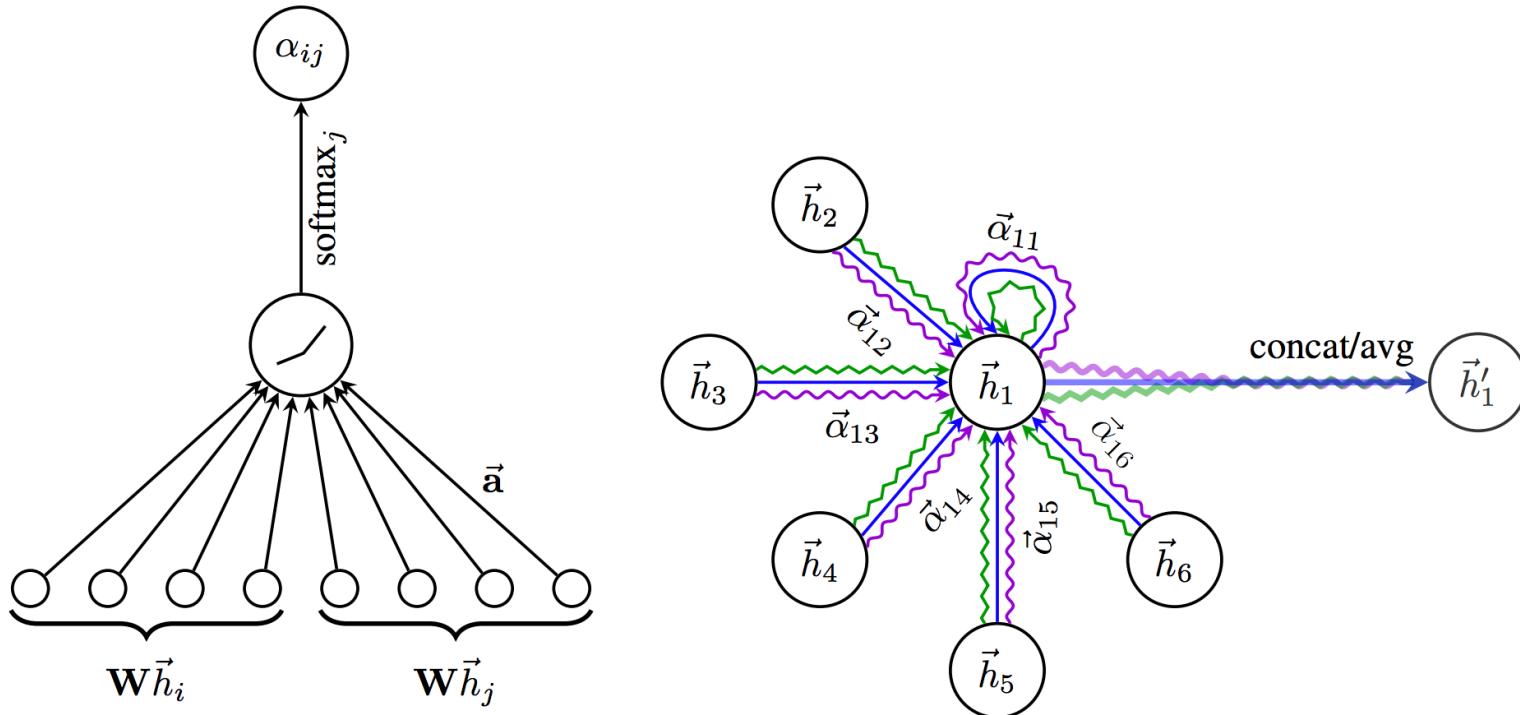
Homogeneous Model



Heterogeneous Model



Graph attention networks

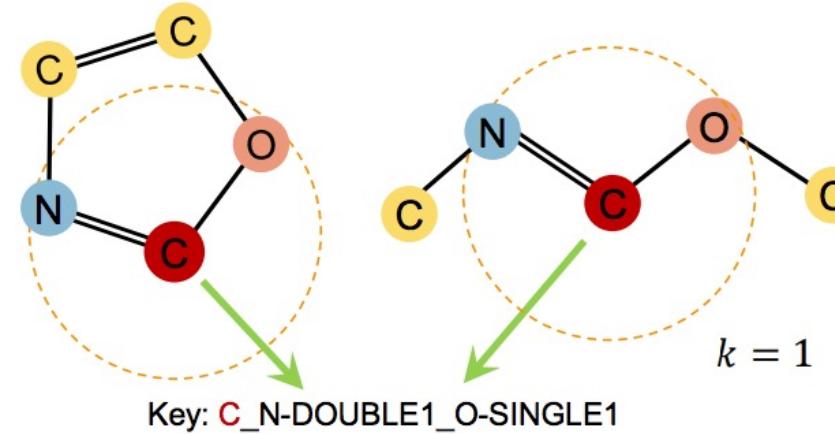


$$m_{N(v)}^{(k)} = \sum_{u \in N(v)} \alpha_{u,v}^{(k-1)} h_u^{(k-1)},$$

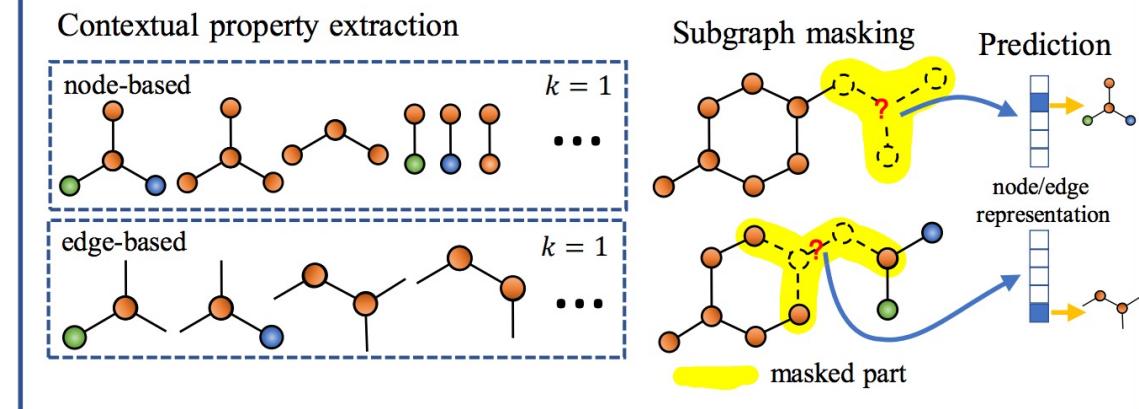
$$\alpha_{u,v}^{(k-1)} = \frac{\exp(f(a^\top [Wh_u^{(k-1)}, Wh_u^{(k-1)}]))}{\sum_{s \in N(v)} \exp(f(a^\top [Wh_s^{(k-1)}, Wh_u^{(k-1)}])))}.$$

GROVER: Transformer on graphs

Как и в обычном трансформере, в GROVER используется self-supervised подход. Все возможные локальные подграфы из тренировочного множества получают идентификатор, после чего, для обучения, случайным образом скрывается часть графа и задача модели предсказать что было скрыто глядя на молекулу целиком.

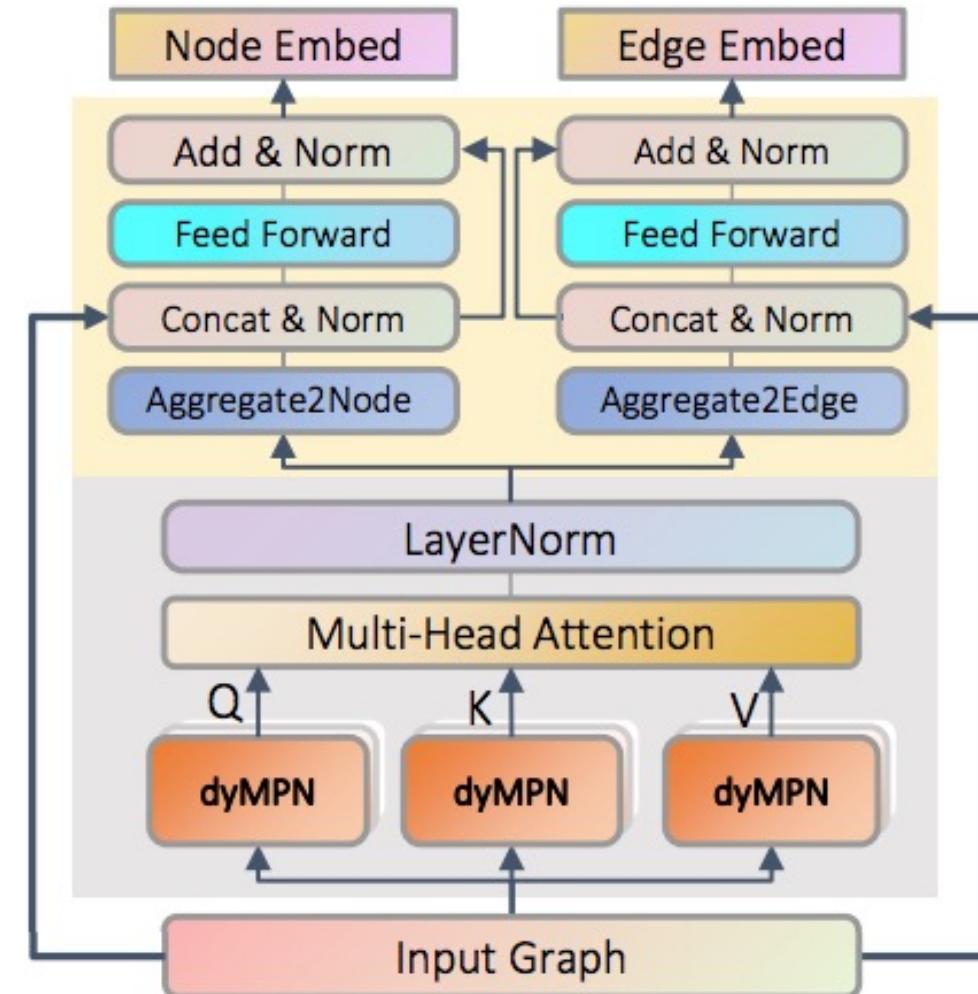


Contextual property prediction (node/edge level task)



GROVER: Transformer on graphs

Архитектура модели повторяет архитектуру трансформера с поправкой на тип данных — граф. Отдельные сверточные блоки соответствуют блокам Q, K, V самовнимания в трансформере. А на выходе сети, каждой вершине и каждому ребру приписывается «эмбеддинг».



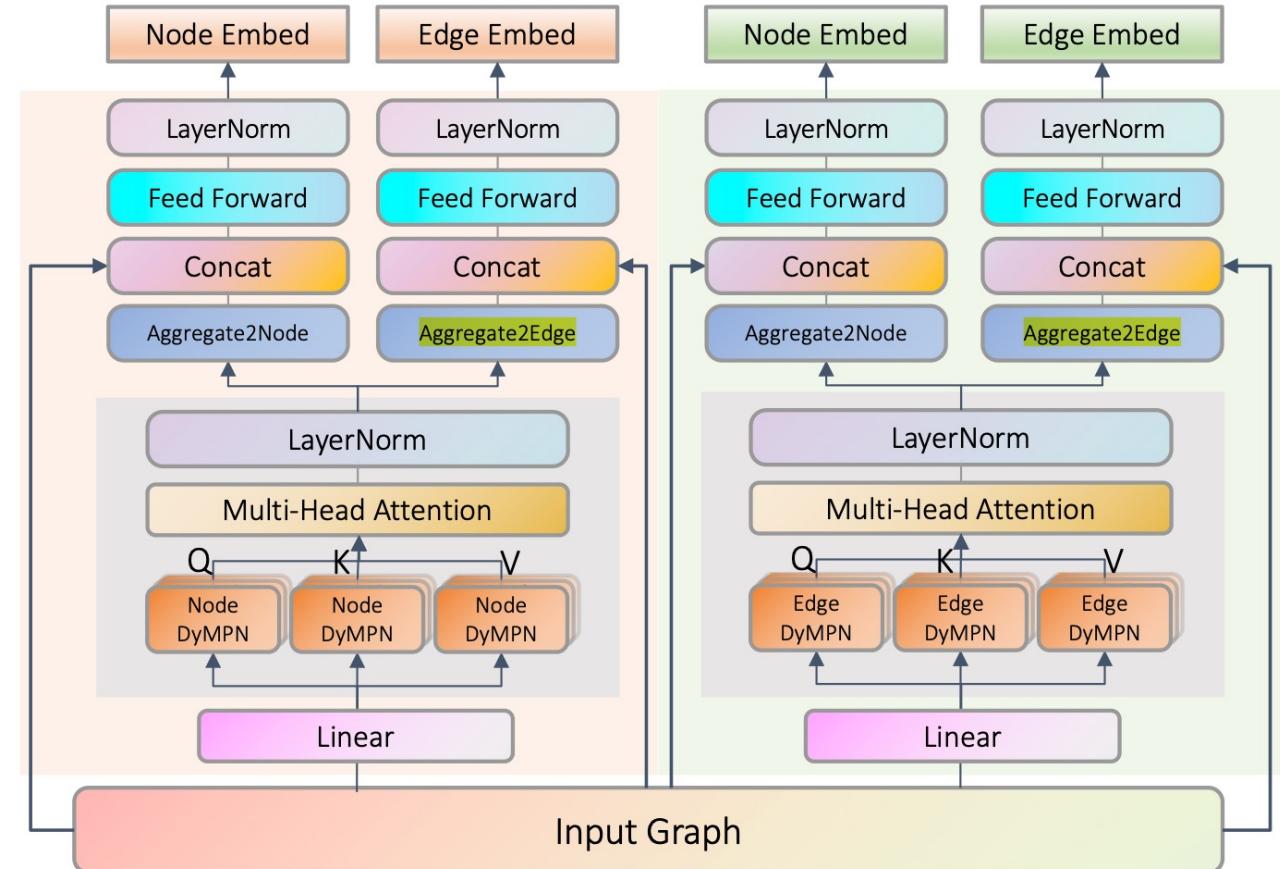
GROVER: Transformer on graphs

$$\mathbf{m}_v^{\text{node-embedding-from-node-states}} = \sum_{u \in \mathcal{N}_v} \bar{\mathbf{h}}_u$$

$$\mathbf{m}_{vw}^{\text{edge-embedding-from-node-states}} = \sum_{u \in \mathcal{N}_v \setminus w} \bar{\mathbf{h}}_u.$$

$$\mathbf{m}_v^{\text{node-embedding-from-edge-states}} = \sum_{u \in \mathcal{N}_v} \bar{\mathbf{h}}_{uv},$$

$$\mathbf{m}_{vw}^{\text{edge-embedding-from-edge-states}} = \sum_{u \in \mathcal{N}_v \setminus w} \bar{\mathbf{h}}_{uv}$$

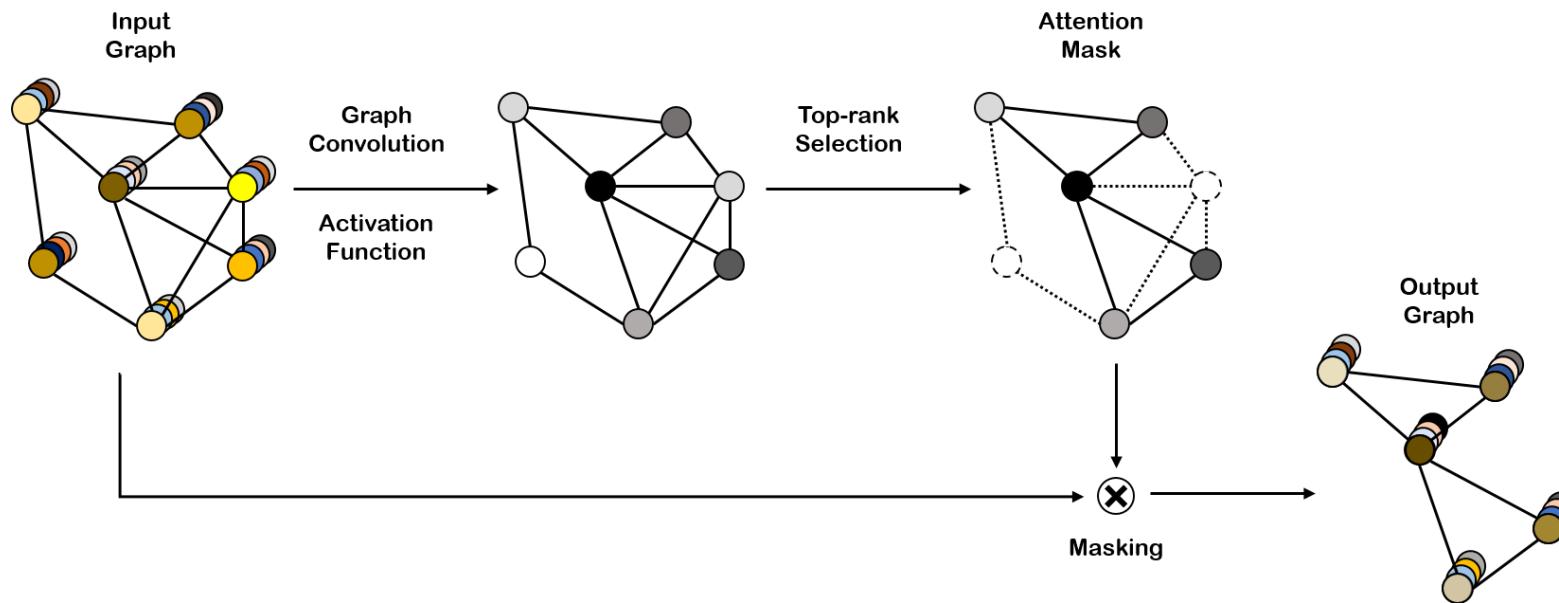


Представление графа

Агрегация представлений вершин/ребер

$$z_G = \frac{1}{f(|V|)} \sum_{u \in V} z_u, \quad z_G = \left[\frac{1}{f(|V|)} \sum_{u \in V} z_u, \max \{z_u\}_{u \in V} \right].$$

Последовательное упрощение структуры графа



Проблемы GNN

Чрезмерное сглаживание
(over-smoothing): представления
соседних вершин становятся мало
отличимы после нескольких
сверточных слоев.

КАК С НИМ БОРОТЬСЯ?

использовать меньше слоёв
для агрегации и больше
для «обработки признаков»

делать прокидывание слоёв
или конкатенацию состояний
с предыдущих слоёв

использовать архитектуры, в которых
есть эффект памяти (например GRU)

использовать аугментацию, например
DropEdge (удаление случайных рёбер)

Can GCNs Go as Deep as CNNs?

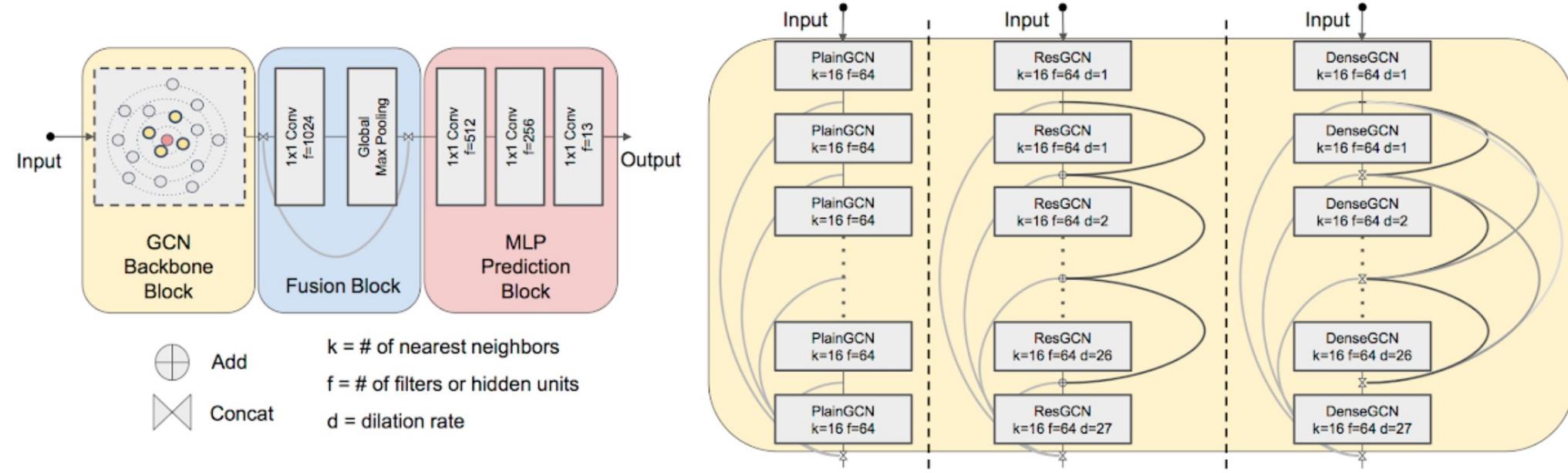


Figure 1. Our GCNs Network architecture for point clouds semantic segmentation. Left: Our framework consists of three blocks (one GCN Backbone Block, one Fusion Block and one MLP Prediction Block). Right: We mainly study three types of GCN Backbone Blocks i.e. PlainGCN, ResGCN and DenseGCN. There are two kinds of GCN skip connections vertex-wise additions and vertex-wise concatenations. k is the number of nearest neighbors in GCN layers. f is the number of the filters or hidden units. d is the dilation rate.

What can GNNs Do?

| | | |
|----------------------------|---|--|
| Graph Mining | Graph Matching Graph Clustering | (Riba et al., 2018; Li et al., 2019b) (Zhang et al., 2019c; Ying et al., 2018b; Tsitsulin et al., 2020) |
| Physics | Physical Systems Modeling | (Battaglia et al., 2016; Sukhbaatar Fergus et al., 2016; Watters et al., 2017; Hoshen, 2017; Kipf et al., 2018; Sanchez et al., 2018) |
| Chemistry | Molecular Fingerprints Chemical Reaction Prediction | (Duvenaud et al., 2015; Kearnes et al., 2016) Do et al. (2019) |
| Biology | Protein Interface Prediction Side Effects Prediction Disease Classification | Fout et al. (2017) Zitnik et al. (2018) Rhee et al. (2018) |
| Knowledge Graph | KB Completion KG Alignment | (Hamaguchi et al., 2017; Schlichtkrull et al., 2018; Shang et al., 2019) (Wang et al., 2018b; Zhang et al., 2019d; Xu et al., 2019c) |
| Generation | Graph Generation | (Shchur et al., 2018b; Nowak et al., 2018; Ma et al., 2018; You et al., 2018a, 2018b; De Cao and Kipf, 2018; Li et al., 2018d; Shi et al., 2020; Liu et al., 2019; Grover et al., 2019) |
| Combinatorial Optimization | Combinatorial Optimization | (Khalil et al., 2017; Nowak et al., 2018; Li et al., 2018e; Kool et al., 2019; Bello et al., 2017; Vinyals et al., 2015b; Sutton and Barto, 2018; Dai et al., 2016; Gasse et al., 2019; Zheng et al., 2020a; Selsam et al., 2019; Sato et al., 2019) |
| Traffic Network | Traffic State Prediction | (Cui et al., 2018b; Yu et al., 2018; Zheng et al., 2020b; Guo et al., 2019) |
| Recommendation Systems | User-item Interaction Prediction Social Recommendation | (van den Berg et al., 2017; Ying et al., 2018a) (Wu et al., 2019c; Fan et al., 2019) |

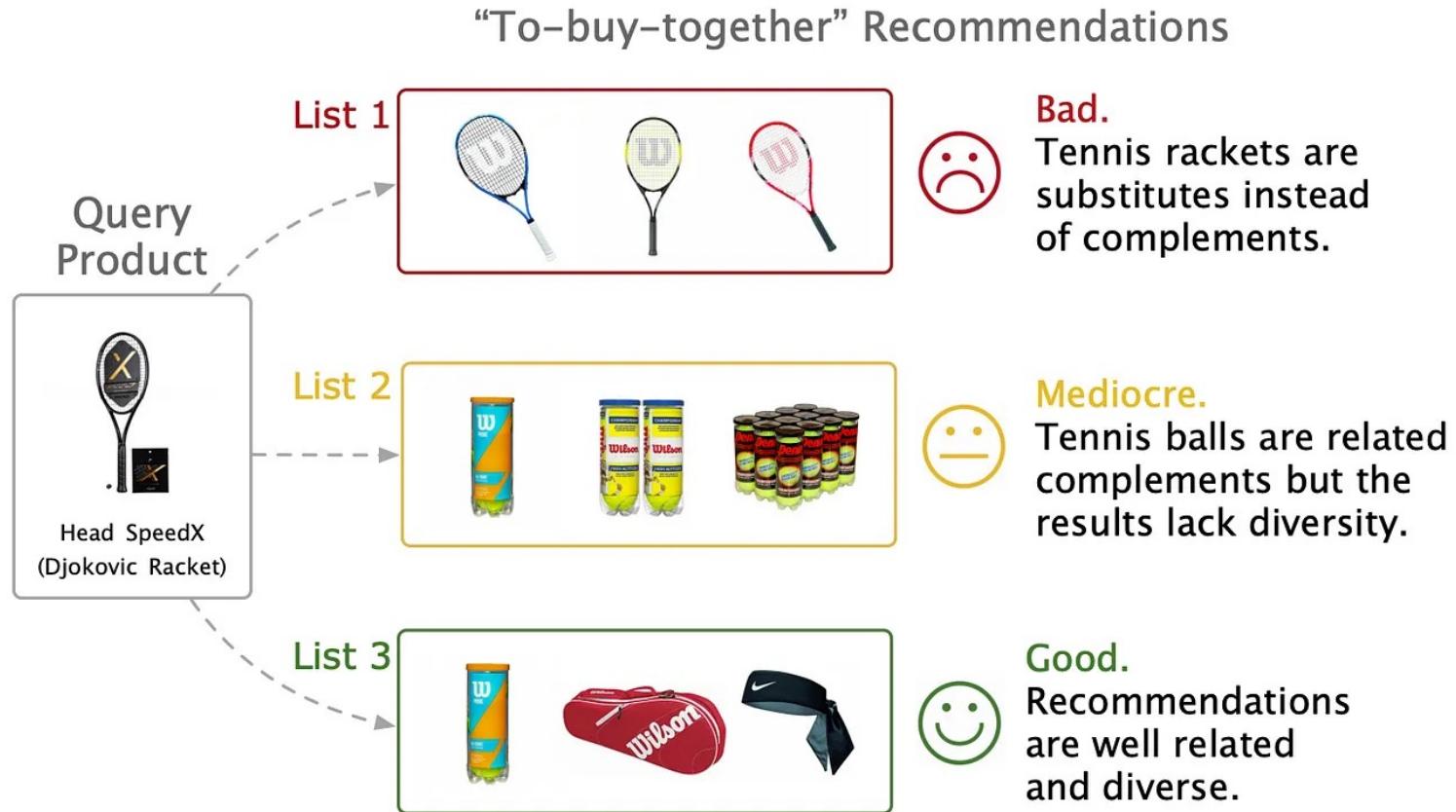
<https://arxiv.org/pdf/1812.08434.pdf>

What can GNNs Do?

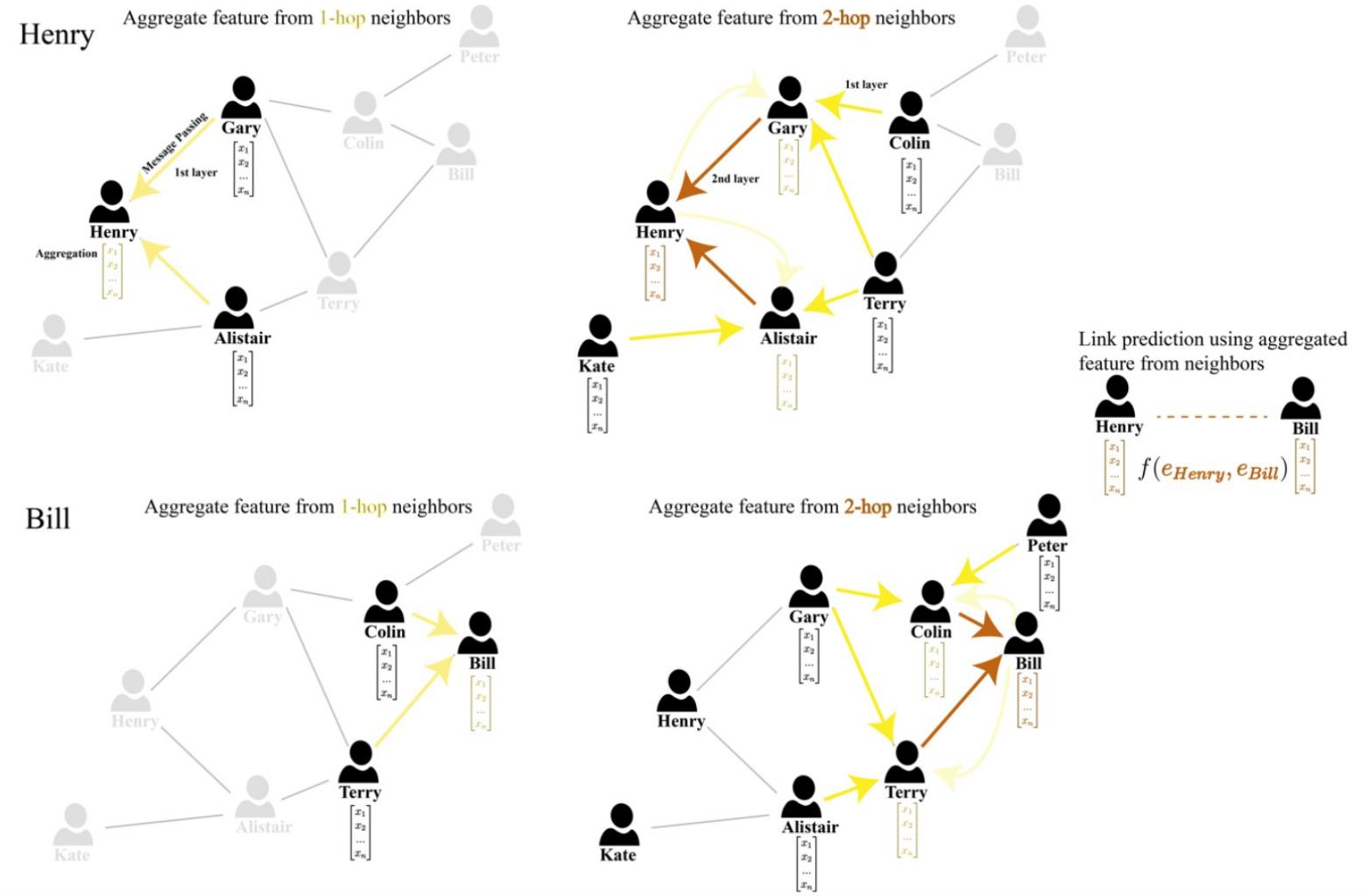
| | | |
|------------------------|--|---|
| Others (Structural) | Stock Market Software Defined Networks AMR Graph to Text | (Matsunaga et al., 2019; Yang et al., 2019; Chen et al., 2018c; Li et al., 2020; Kim et al., 2019) Rusek et al. (2019) (Song et al., 2018a; Beck et al., 2018) |
| Text | Text Classification Sequence Labeling Neural Machine Translation Relation Extraction Event Extraction Fact Verification Question Answering Relational Reasoning | (Peng et al., 2018; Yao et al., 2019; Zhang et al., 2018d; Tai et al., 2015) (Zhang et al., 2018d; Marcheggiani and Titov, 2017) (Bastings et al., 2017; Marcheggiani et al., 2018; Beck et al., 2018) (Miwa and Bansal, 2016; Peng et al., 2017; Song et al., 2018b; Zhang et al., 2018f) (Nguyen and Grishman, 2018; Liu et al., 2018) (Zhou et al., 2019; Liu et al., 2020; Zhong et al., 2020) (Song et al., 2018c; De Cao et al., 2019; Qiu et al., 2019; Tu et al., 2019; Ding et al., 2019) (Santoro et al., 2017; Palm et al., 2018; Battaglia et al., 2016) |
| Image | Social Relationship Understanding Image Classification Visual Question Answering Object Detection Interaction Detection Region Classification Semantic Segmentation | Wang et al. (2018c) (Garcia and Bruna, 2018; Wang et al., 2018d; Lee et al., 2018b; Kampffmeyer et al., 2019; Marino et al., 2017) (Teney et al., 2017; Wang et al., 2018c; Narasimhan et al., 2018) (Hu et al., 2018; Gu et al., 2018) (Qi et al., 2018; Jain et al., 2016) Chen et al. (2018d) (Liang et al., 2016, 2017; Landrieu and Simonovsky, 2018; Wang et al., 2018e; Qi et al., 2017b) |
| Other (Non-structural) | Program Verification | (Allamanis et al., 2018; Li et al., 2016) |

<https://arxiv.org/pdf/1812.08434.pdf>

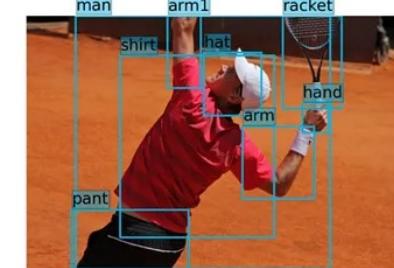
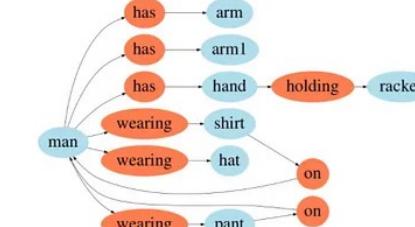
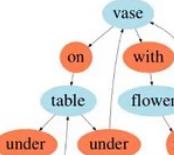
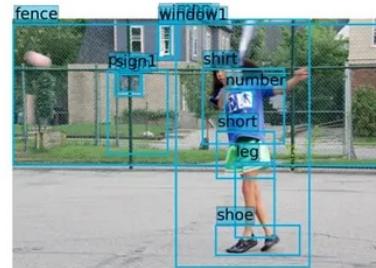
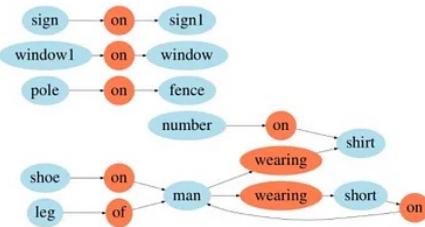
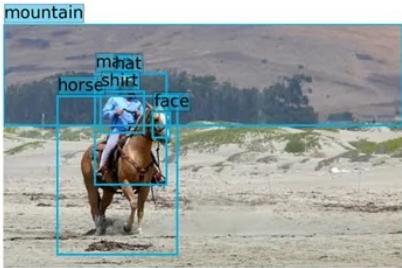
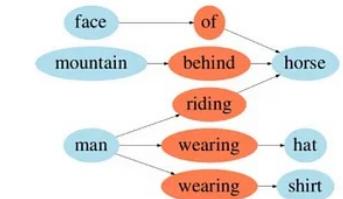
Recommendation systems



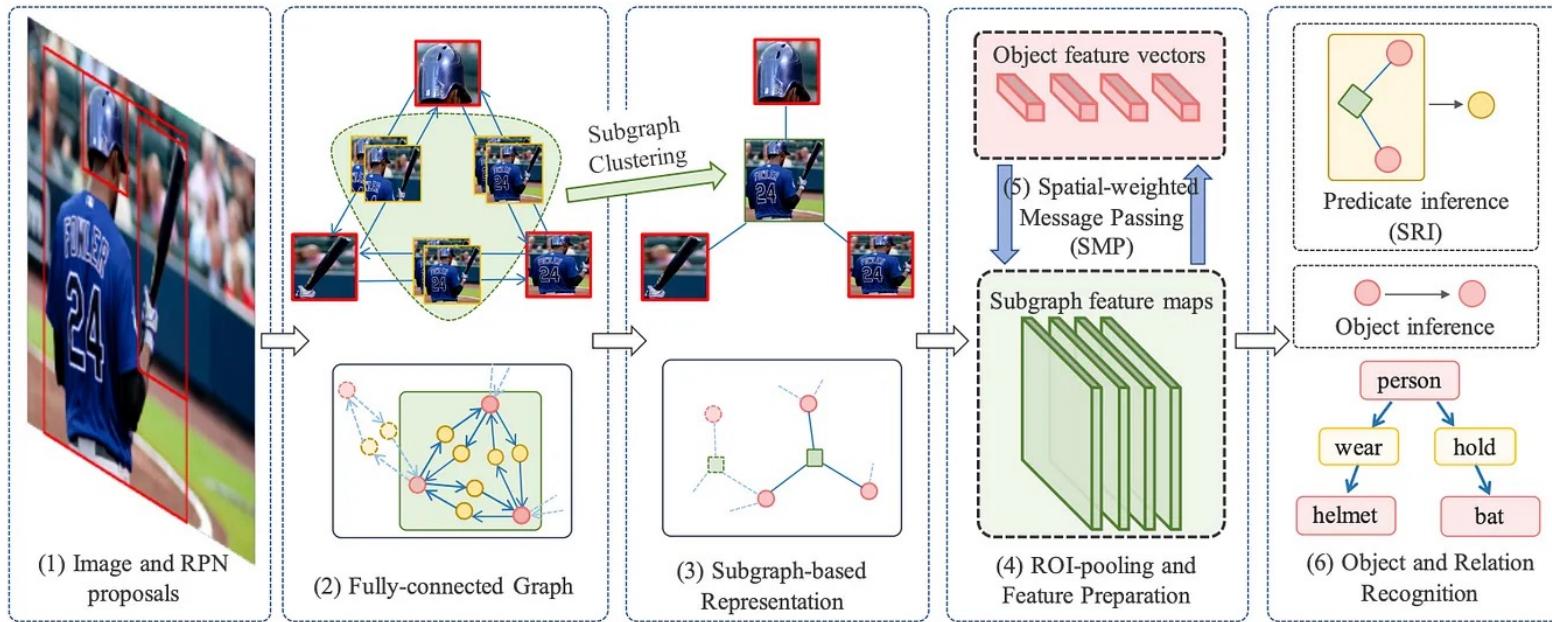
Recommendation systems



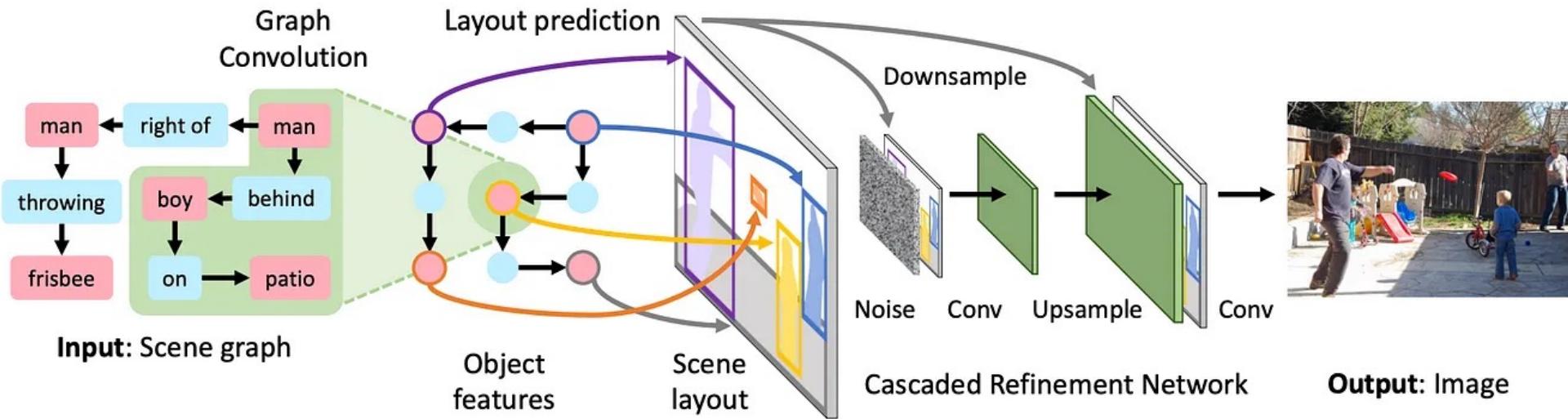
Scene graph generation



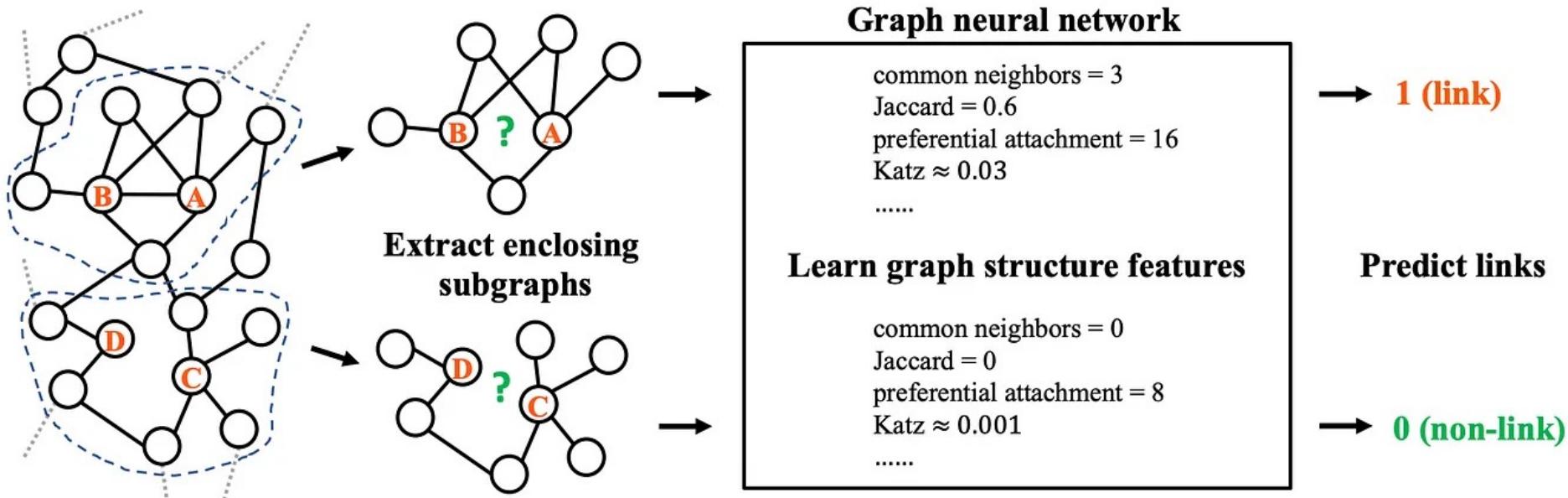
Scene graph generation



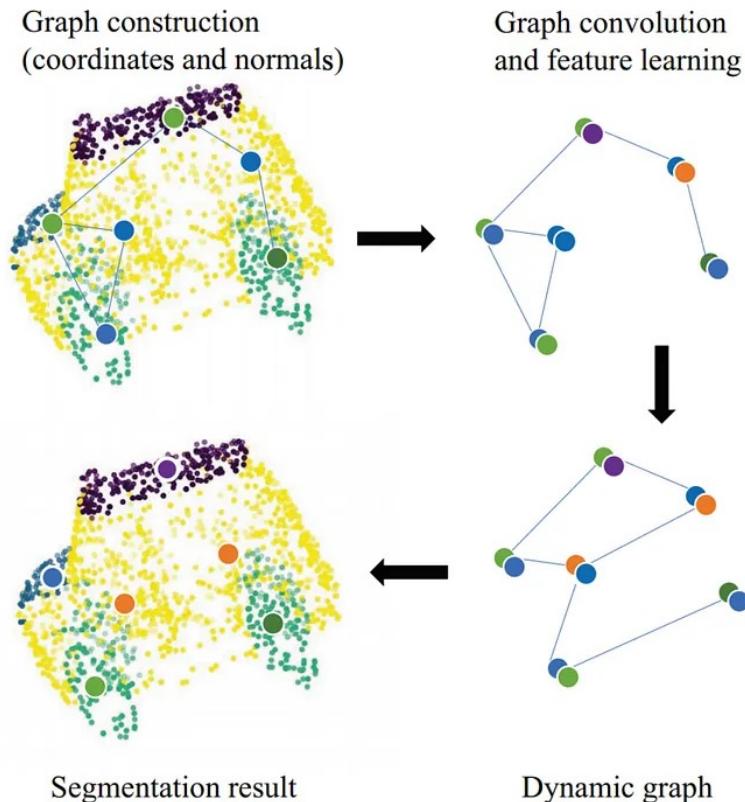
Scene graph generation



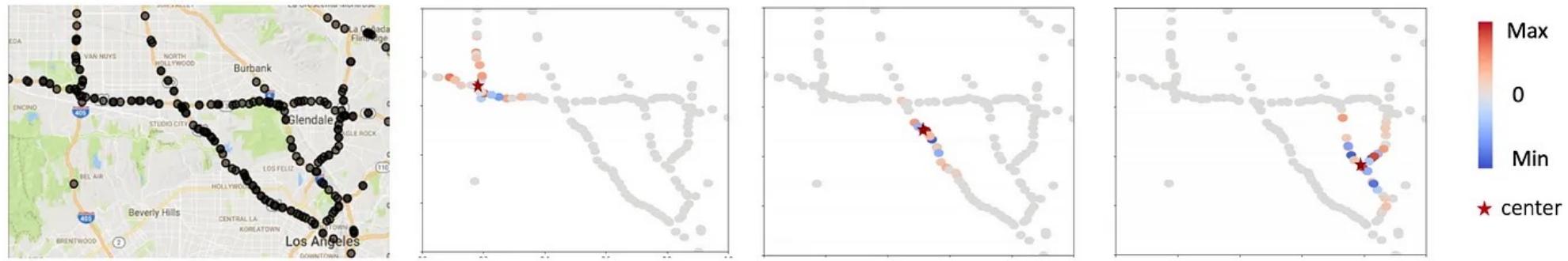
Link prediction



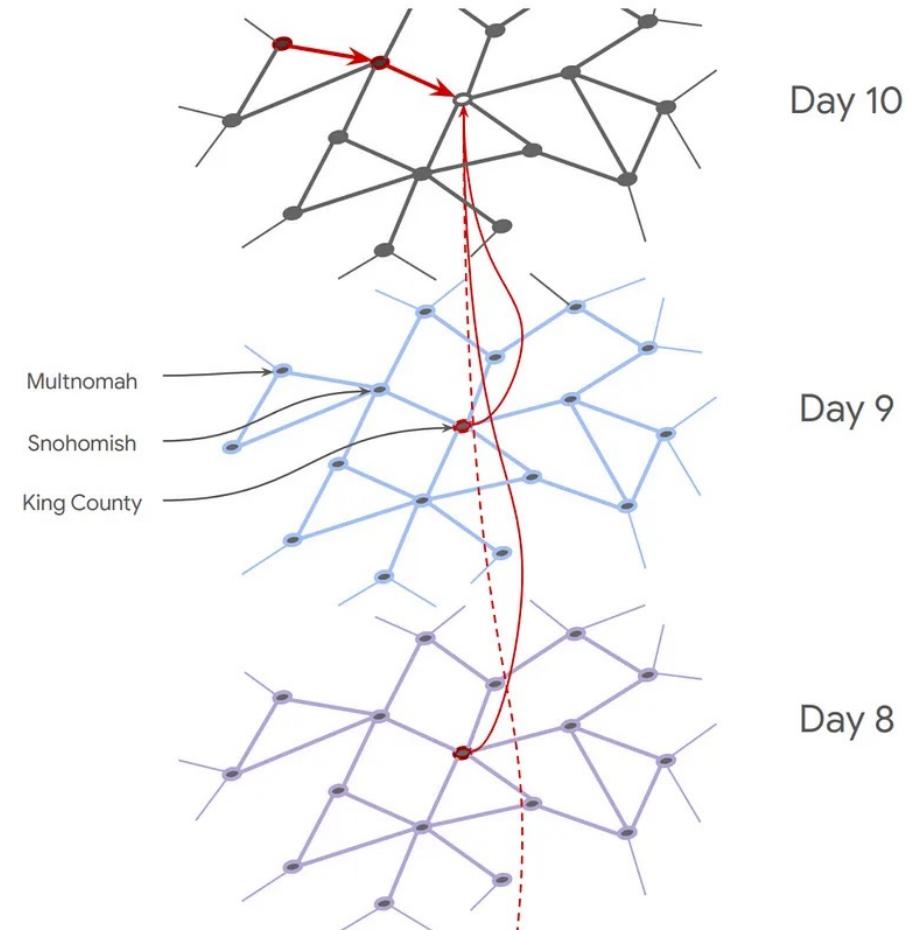
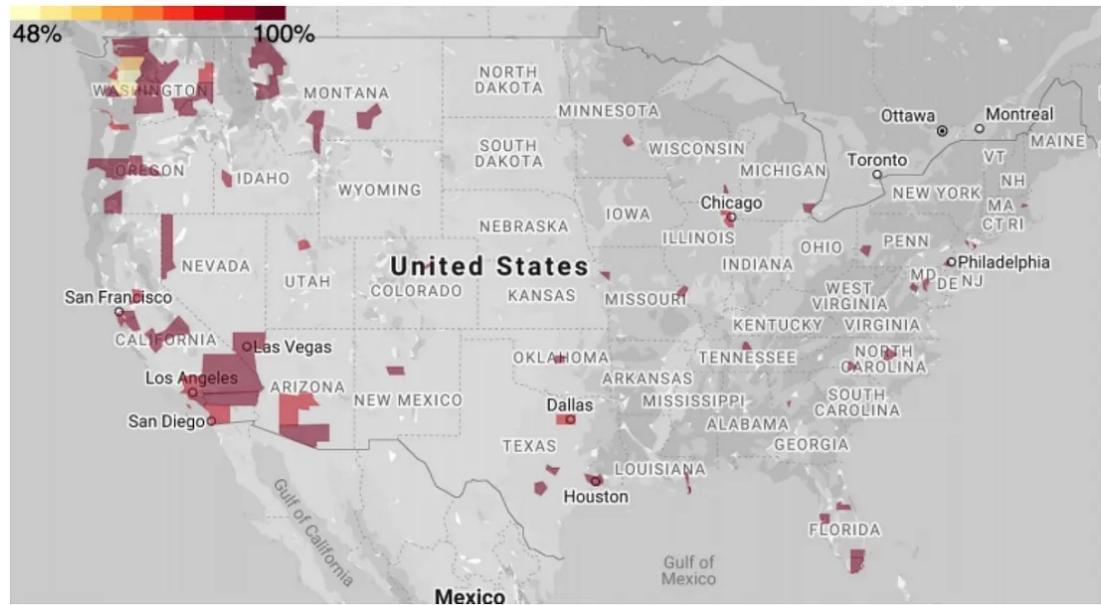
Point clouds



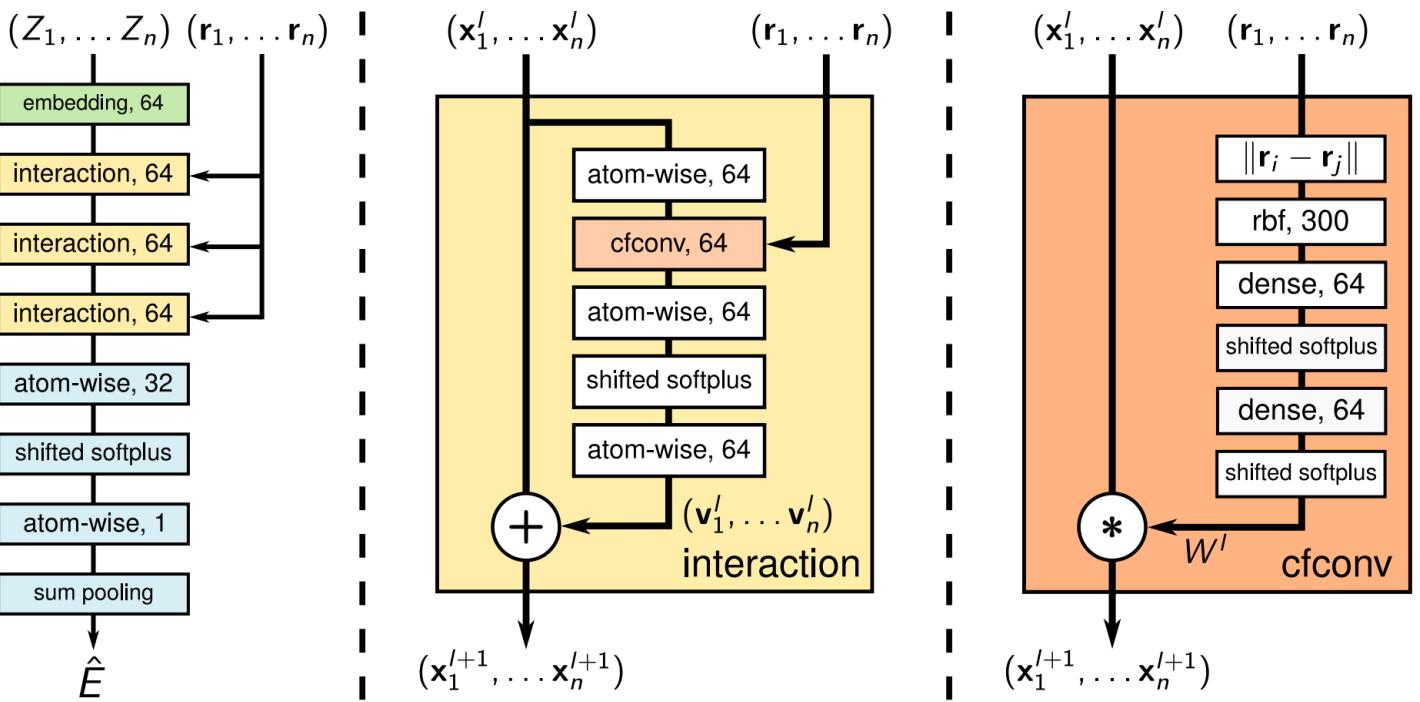
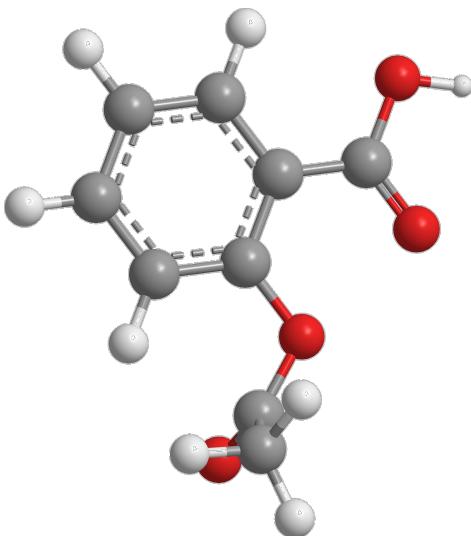
Traffic forecast



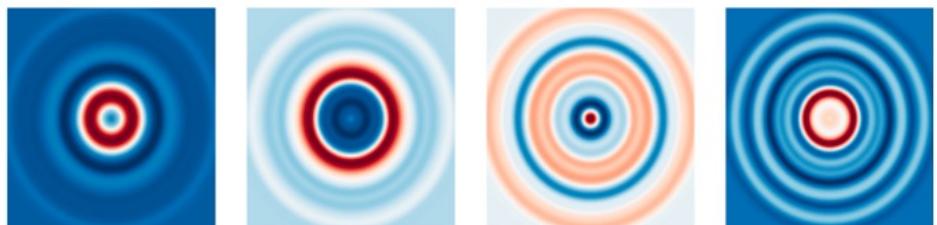
Modeling viruses spread



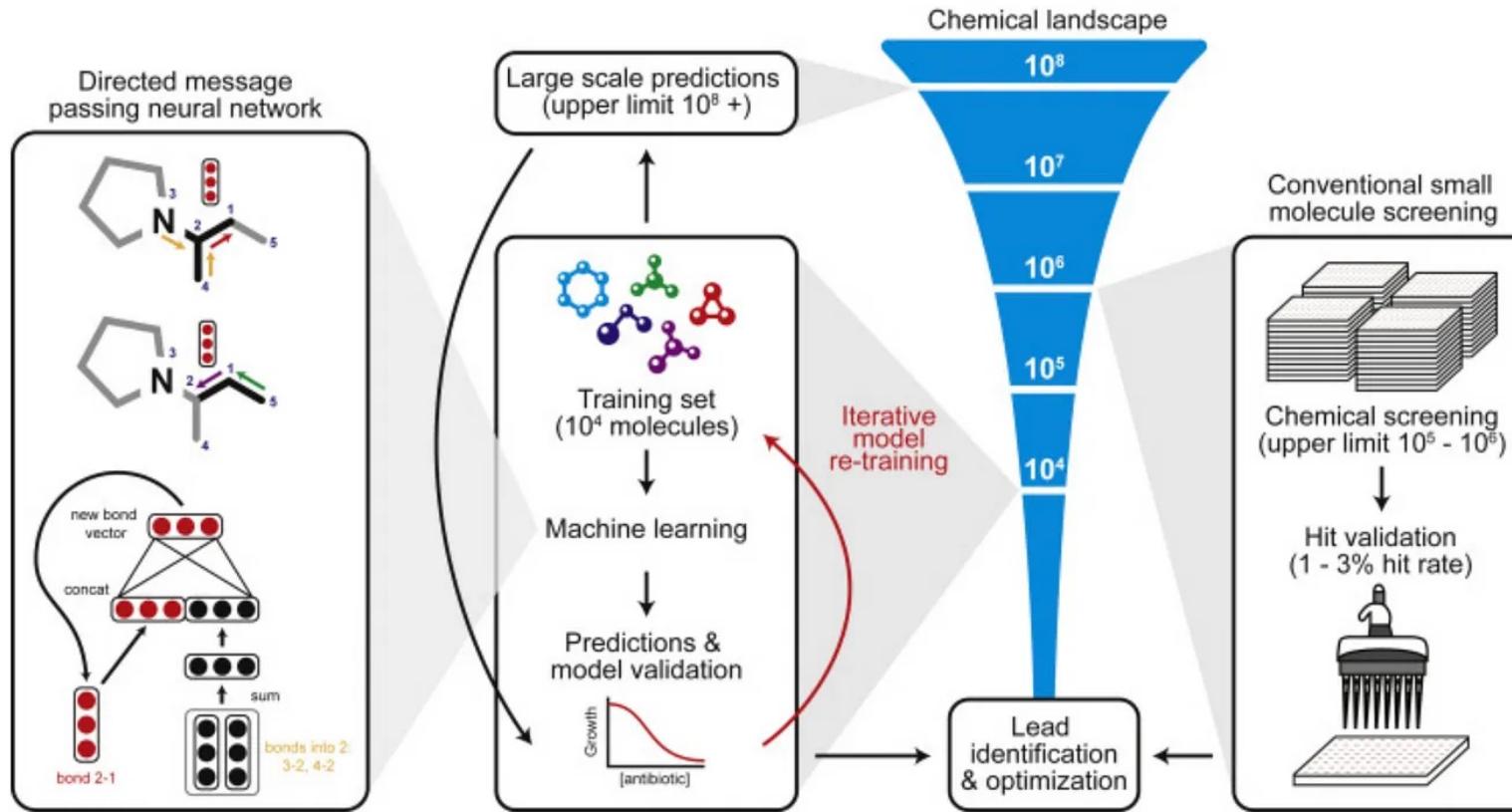
Quantum chemistry



$$\mathbf{x}_i^{l+1} = (X^l * W^l)_i = \sum_{j=0}^{n_{\text{atoms}}} \mathbf{x}_j^l \circ W^l(\mathbf{r}_j - \mathbf{r}_i),$$

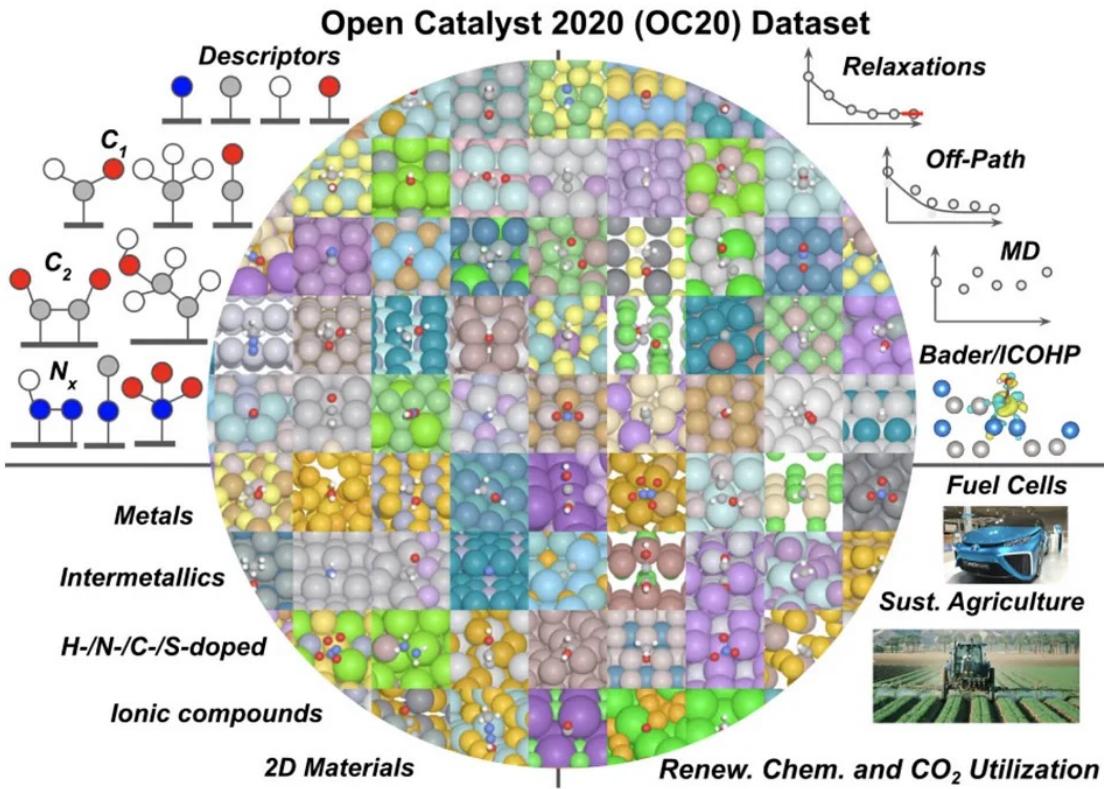


Drug discovery and materials design



<https://arxiv.org/pdf/1812.08434.pdf>

Drug discovery and materials design



Ссылки на статьи и ресурсы

Justin Gilmer, Samuel S. Schoenholz, Oriol Vinyals et al. [Neural Message Passing for Quantum Chemistry](#)

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, [The Graph Neural Network Model,](#)

Si Zhang et al. [Graph convolutional networks: a comprehensive review](#)

[Блог Александра Дьяконова](#)

[A Gentle Introduction to Graph Neural Networks](#)

Yu Rong et al. [Self-Supervised Graph Transformer on Large-Scale Molecular Data](#)

Guohao Li et al. [DeepGCNs: Can GCNs Go as Deep as CNNs?](#)

William L. Hamilton [Graph Representation Learning](#)



Спасибо
за внимание!