

Темпоральные GNN и задача ETA

Зелинский Никита

План лекции

01. Graph attention

План лекции

01. Graph attention

02. Темпоральные GNN

План лекции

01. Graph attention

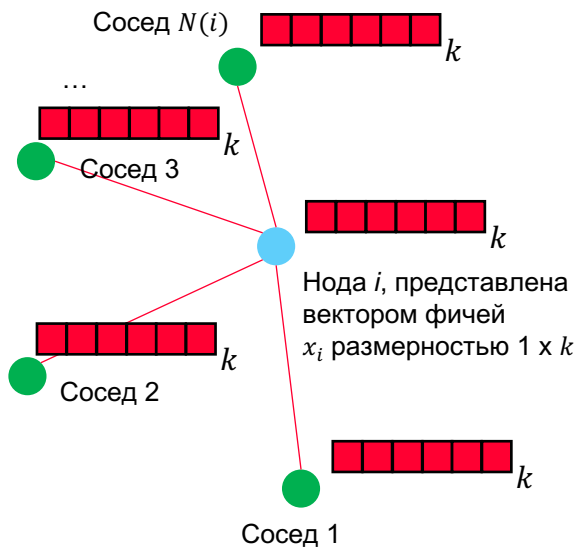
02. Темпоральные GNN

03. Задача ETA

Graph Attention

01

Графовая свертка GraphConv



https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GraphConv.html

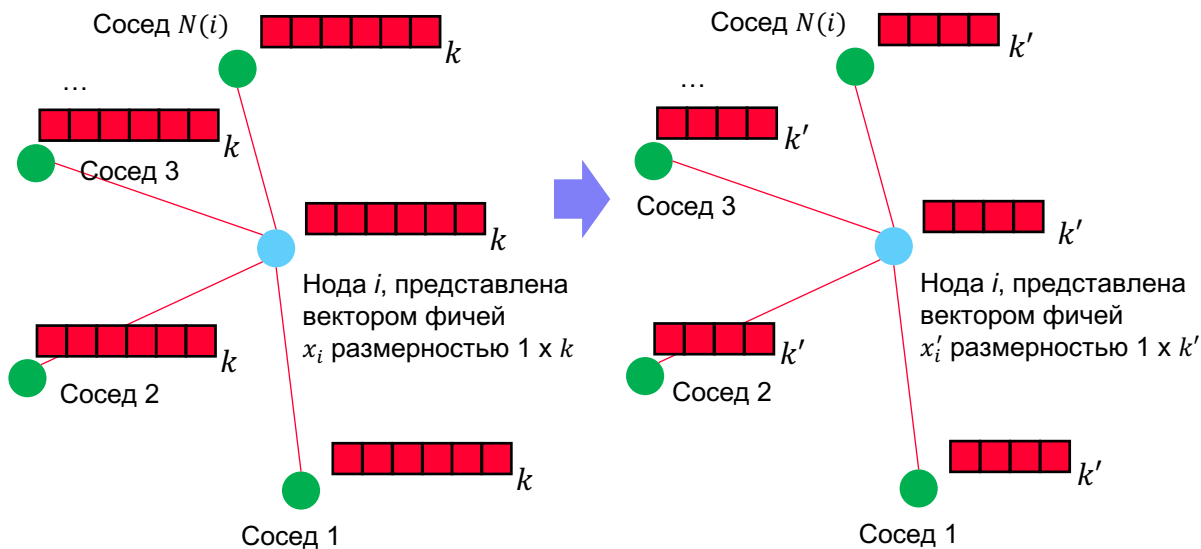
<https://arxiv.org/pdf/1704.01212.pdf>
<https://arxiv.org/pdf/1810.02244.pdf>
<https://arxiv.org/pdf/1709.05584.pdf>

Графовая свертка GraphConv



$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

W и H имеют размерность $k \times k'$:
 $(1 \times k) * (k \times k') = 1 \times k'$

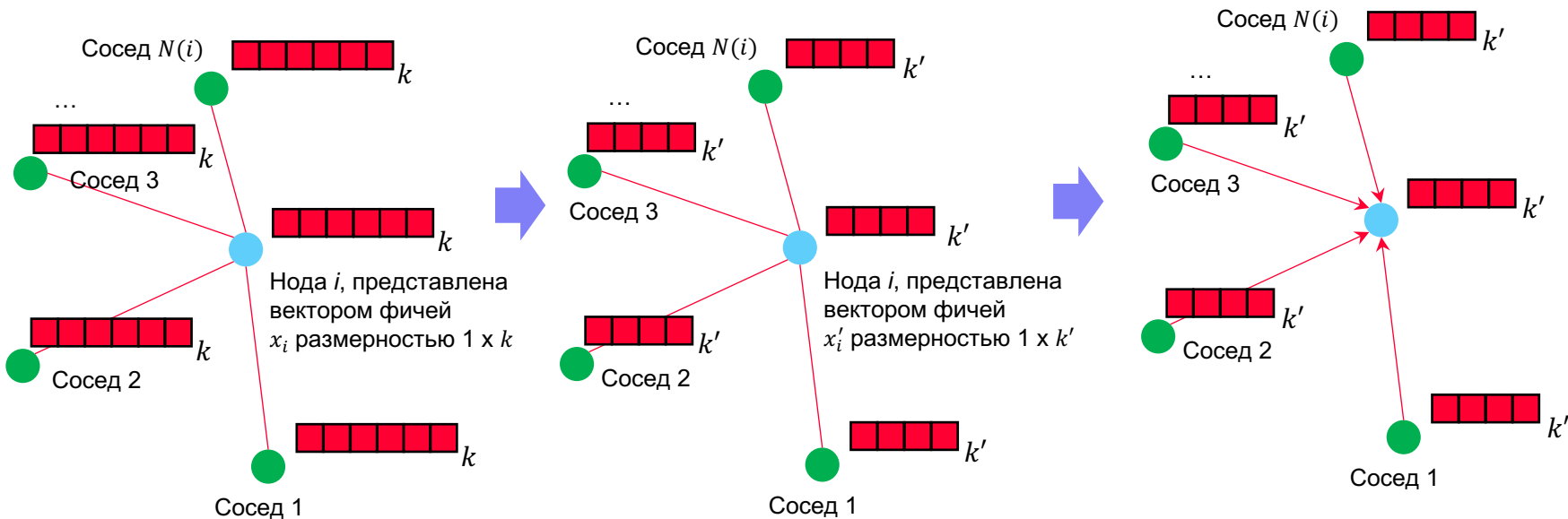


Графовая свертка GraphConv



$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

W и H имеют размерность $k \times k'$:
 $(1 \times k) * (k \times k') = 1 \times k'$

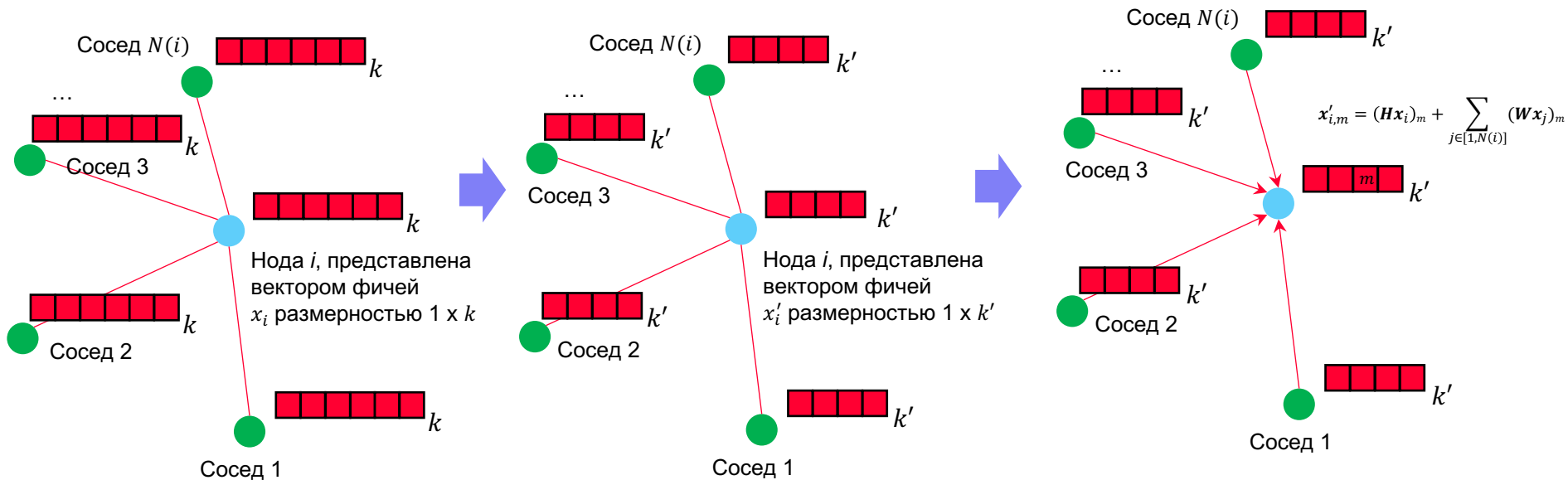


Графовая свертка GraphConv



$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

W и H имеют размерность $k \times k'$:
 $(1 \times k) * (k \times k') = 1 \times k'$

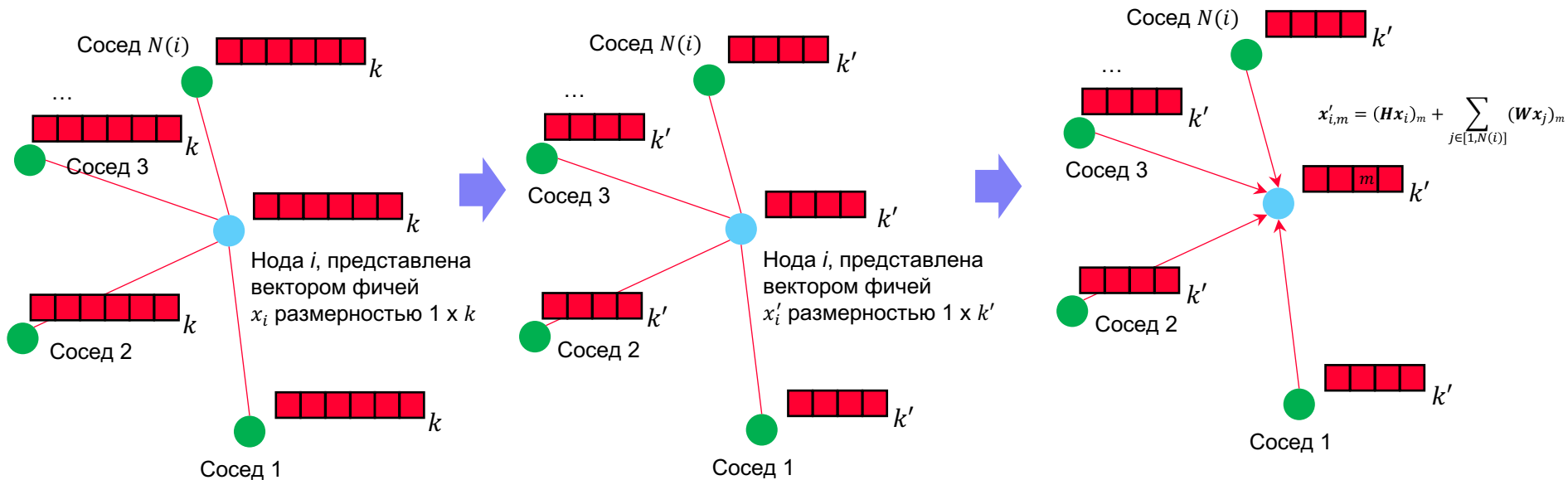


Графовая свертка GraphConv



$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

W и H имеют размерность $k \times k'$:
 $(1 \times k) * (k \times k') = 1 \times k'$



https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GraphConv.html

Вместо суммы – другие агрегирующие функции

<https://arxiv.org/pdf/1704.01212.pdf>

<https://arxiv.org/pdf/1810.02244.pdf>

<https://arxiv.org/pdf/1709.05584.pdf>

Разберемся в обозначениях

GraphConv

$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

GCNConv

$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} XW)$$

Разберемся в обозначениях

GraphConv

$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

GCNConv

$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} XW)$$

Чем отличаются A и \tilde{A} ?

$$\tilde{A} = A + I_N$$

Разберемся в обозначениях

GraphConv

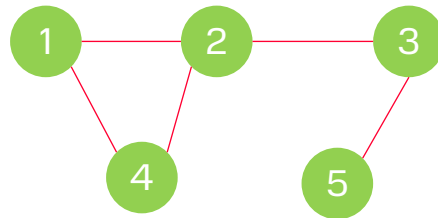
$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

Чем отличаются A и \tilde{A} ?

GCNConv

$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} XW)$$

$$\tilde{A} = A + I_N$$



0	1			
1	0			
		0		1
	1		0	
		1		0



1	1			
1	1			
		1		1
	1		1	
		1		1

Разберемся в обозначениях

GraphConv

$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

GCNConv

$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} XW)$$

Чем отличаются A и \tilde{A} ?

$$\tilde{A} = A + I_N$$

$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$$

Разберемся в обозначениях

GraphConv

$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

GCNConv

$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} XW)$$

Чем отличаются A и \tilde{A} ?

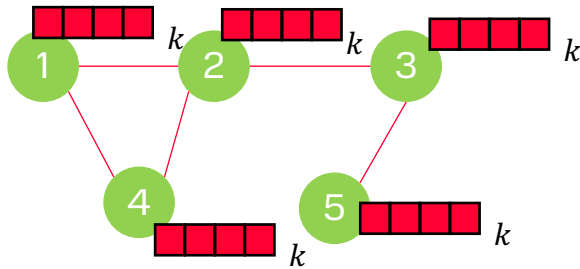
$$\tilde{A} = A + I_N$$

$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$$

Еще пишут так $\tilde{X} = GCN(A, X) = ReLU(\tilde{A}XW_0)$

Подразумевают $\tilde{A} = \tilde{D}^{-1/2} \hat{A} \tilde{D}^{-1/2}$ и $\hat{A} = A + I_N$

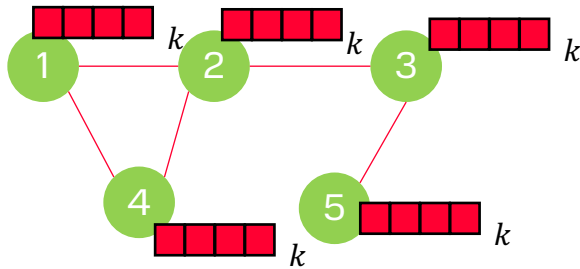
Графовая свертка GCNConv



\hat{A} (5x5)

1	1			
1	1		1	
		1		1
	1		1	
		1		1

Графовая свертка GCNConv



\hat{A} (5x5)

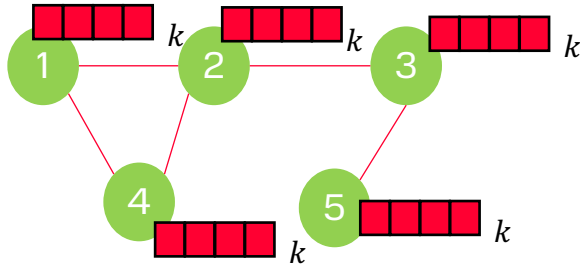
1	1			
1	1		1	
		1		1
	1		1	
		1		1

X (5x4)



Фичи вершин

Графовая свертка GCNConv



$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} X W)$$

\hat{A} (5x5)

1	1			
1	1		1	
		1		1
	1		1	
		1		1

X (5x4)



Фичи вершин

W (4x2)



Матрица весов –
обучаем

$ReLU()$



X' (5x2)



Матрица
эмбедингов вершин

Отличаются?

GraphConv

$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

GCNConv

$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} XW)$$

HO!

Отличаются?

GraphConv

$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} Wx_j$$

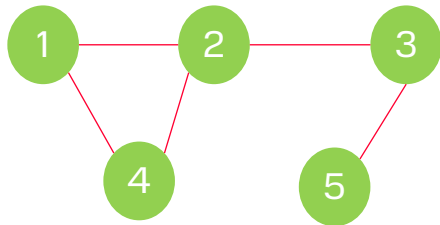
GCNConv

$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} XW)$$

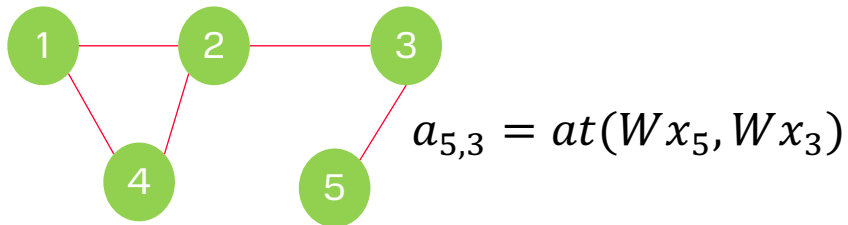
НО!

Работает для bipartитных графов!

Идея Graph Attention (не механизм)



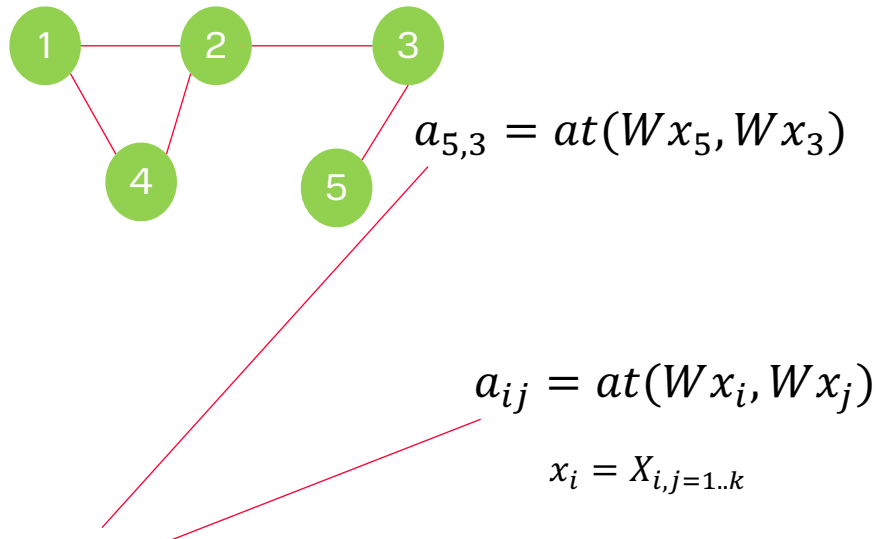
Идея Graph Attention (не механизм)



$$a_{ij} = at(Wx_i, Wx_j)$$

$$x_i = X_{i,j=1..k}$$

Идея Graph Attention (не механизм)



Коэффициент внимания

Graph Attention

$$a_{ij} = at(Wx_i, Wx_j)$$

1. Давайте считать a_{ij} только для вершин, имеющих общее ребро (показано на экспериментах что хватает 1 шага) – назовем это ***masked attention***

Graph Attention

$$a_{ij} = at(Wx_i, Wx_j)$$

1. Давайте считать a_{ij} только для вершин, имеющих общее ребро (показано на экспериментах что хватает 1 шага) – назовем это ***masked attention***
2. Давайте нормировать коэффициенты внимания

$$\alpha_{ij} = softmax_j (a_{ij}) = \frac{e^{a_{ij}}}{\sum_{k \in N_i} e^{a_{ik}}}$$

Graph Attention

$$a_{ij} = at(Wx_i, Wx_j)$$

1. Давайте считать a_{ij} только для вершин, имеющих общее ребро (показано на экспериментах что хватает 1 шага) – назовем это ***masked attention***
2. Давайте нормировать коэффициенты внимания

$$\alpha_{ij} = softmax_j(a_{ij}) = \frac{e^{a_{ij}}}{\sum_{k \in N_i} e^{a_{ik}}}$$

3. Давайте внимание сделаем линейным слоем с функцией активации LeakyReLU

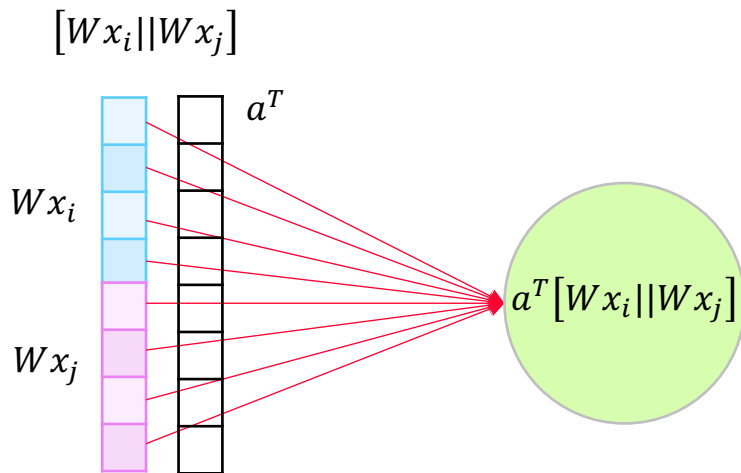
$$\alpha_{ij} = \frac{e^{LeakyReLU(a^T[Wx_i || Wx_j])}}{\sum_{k \in N_i} e^{LeakyReLU(a^T[Wx_i || Wx_k])}}$$

Graph Attention

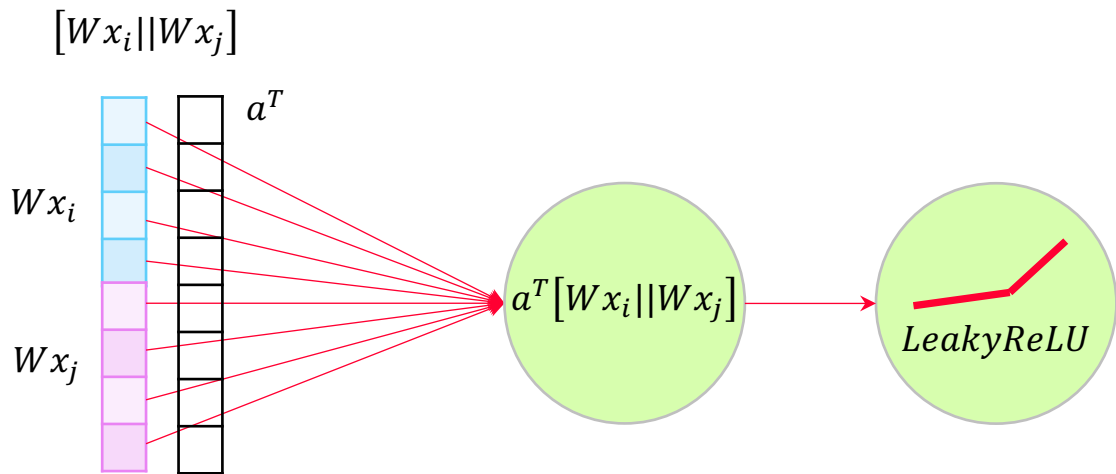
$$[Wx_i || Wx_j]$$



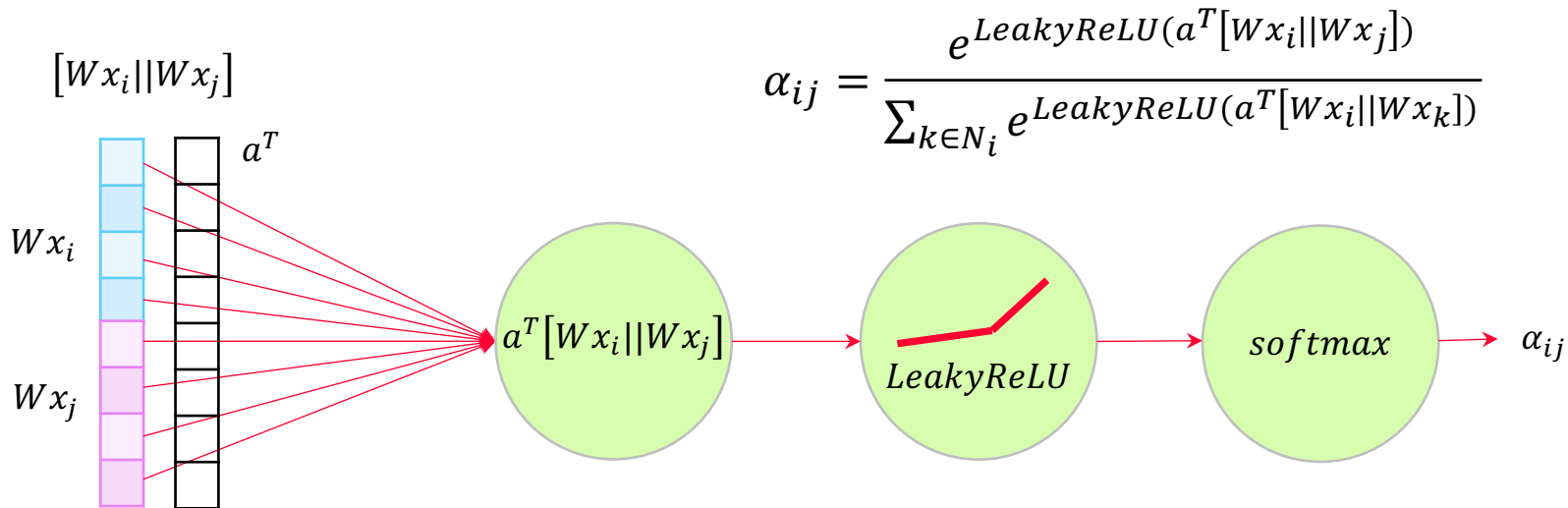
Graph Attention



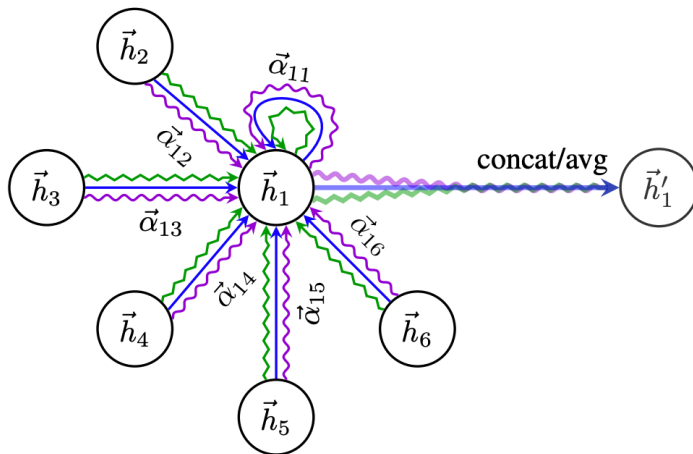
Graph Attention



Graph Attention



Multihead Graph Attention

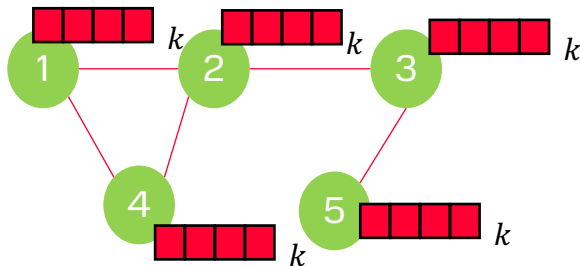


Graph attention networks (2018)

<https://arxiv.org/pdf/1710.10903.pdf>

Графовая свертка GATConv

$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} X W)$$



$$x'_i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} W x_j \right)$$

\hat{A} (5x5)

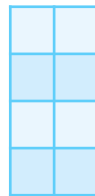
1	1			
1	1			
		1		1
	1		1	
		1		1

X (5x4)

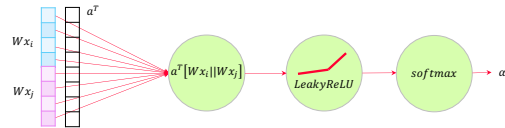


Фичи вершин

W (4x2)



Матрица весов –
обучаем



X' (5x2)



Матрица
эмбеддингов вершин

Реализация в PyG



The graph attentional operator from the “Graph Attention Networks” paper.

$$\mathbf{x}'_i = \alpha_{i,i} \Theta_s \mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \Theta_t \mathbf{x}_j,$$

where the attention coefficients $\alpha_{i,j}$ are computed as

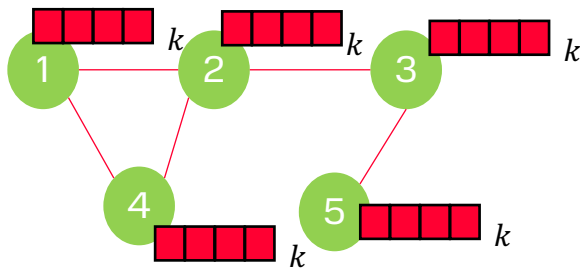
$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}_s^\top \Theta_s \mathbf{x}_i + \mathbf{a}_t^\top \Theta_t \mathbf{x}_j))}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(\text{LeakyReLU}(\mathbf{a}_s^\top \Theta_s \mathbf{x}_i + \mathbf{a}_t^\top \Theta_t \mathbf{x}_k))}.$$

If the graph has multi-dimensional edge features $\mathbf{e}_{i,j}$, the attention coefficients $\alpha_{i,j}$ are computed as

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}_s^\top \Theta_s \mathbf{x}_i + \mathbf{a}_t^\top \Theta_t \mathbf{x}_j + \mathbf{a}_e^\top \Theta_e \mathbf{e}_{i,j}))}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(\text{LeakyReLU}(\mathbf{a}_s^\top \Theta_s \mathbf{x}_i + \mathbf{a}_t^\top \Theta_t \mathbf{x}_k + \mathbf{a}_e^\top \Theta_e \mathbf{e}_{i,k}))}.$$

If the graph is not bipartite, $\Theta_s = \Theta_t$.

Какая проблема с GATConv?



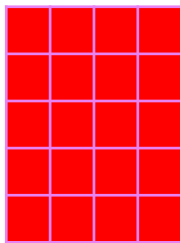
$$x'_i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} W x_j \right)$$

$$\alpha_{ij} = \frac{e^{\text{LeakyReLU}(a^T [W x_i || W x_j])}}{\sum_{k \in N_i} e^{\text{LeakyReLU}(a^T [W x_i || W x_k])}}$$

\hat{A} (5x5)

1	1			
1	1			
		1		1
	1		1	
		1		1

X (5x4)

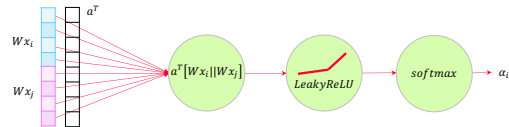


Фичи вершин

W (4x2)



Матрица весов –
обучаем

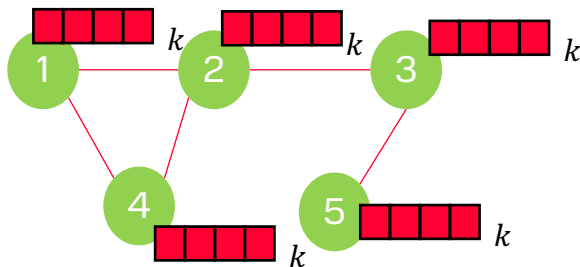


X' (5x2)



Матрица
эмбеддингов вершин

Какая проблема с GATConv?

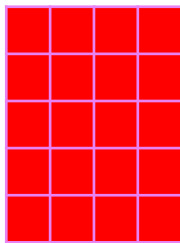


$$x'_i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} W x_j \right)$$

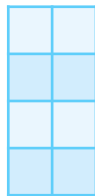
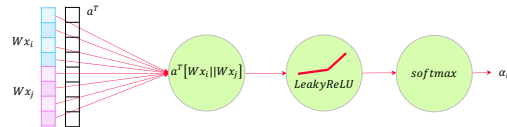
$$\alpha_{ij} = \frac{e^{\text{LeakyReLU}(a^T [W x_i || W x_j])}}{\sum_{k \in N_i} e^{\text{LeakyReLU}(a^T [W x_i || W x_k])}}$$

 \hat{A} (5x5)

1	1			
1	1			
		1		1
	1		1	
		1		1

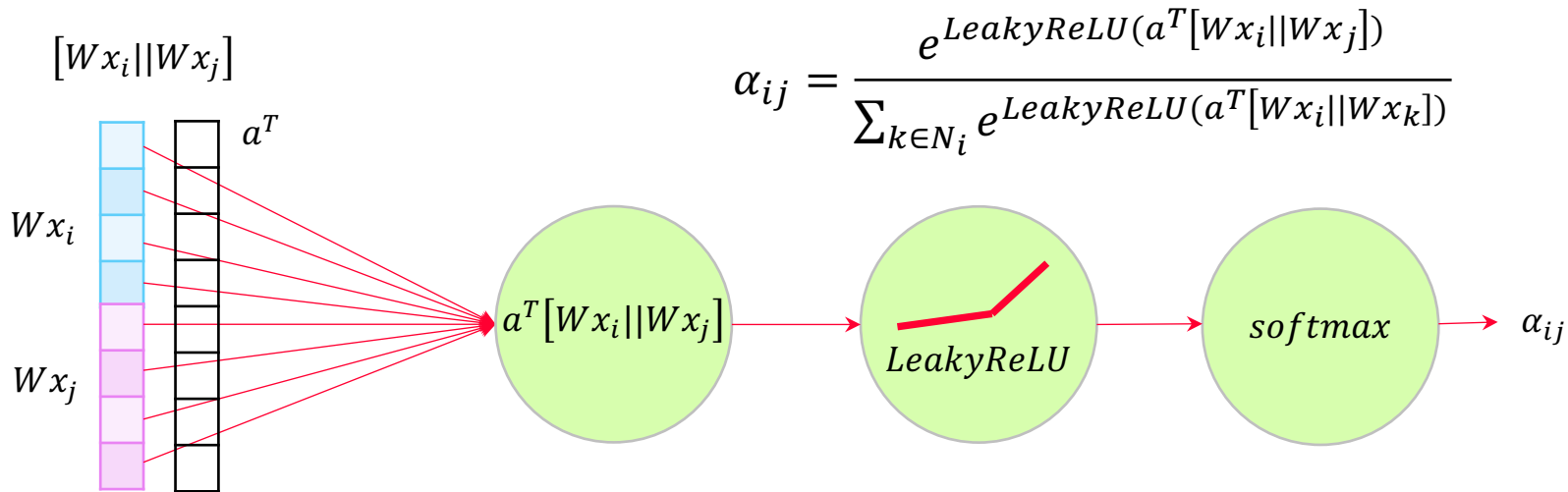
 X (5x4)


Фичи вершин

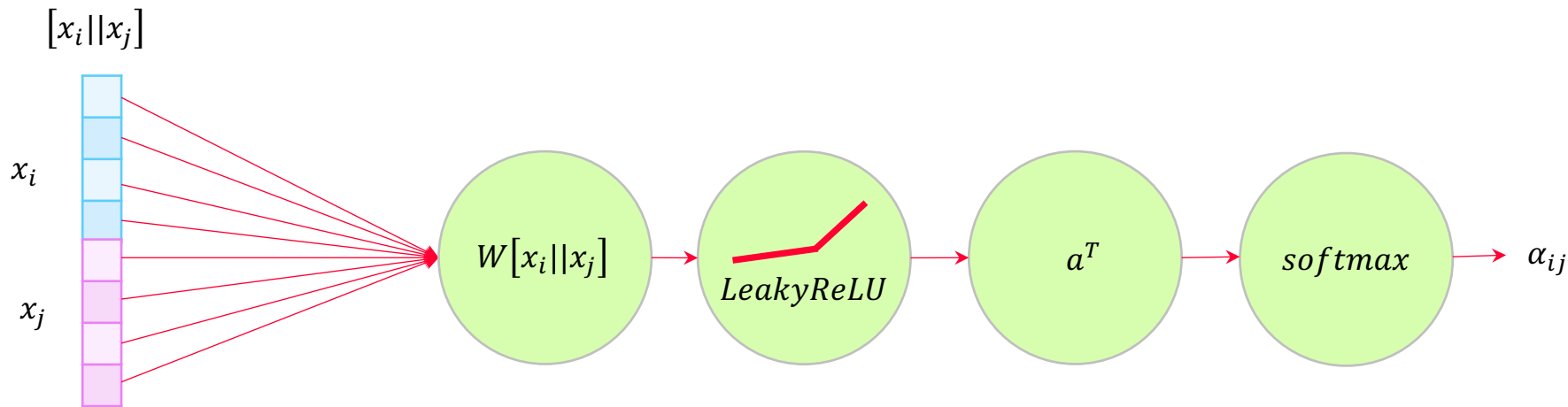
 W (4x2)

Матрица весов –
обучаем

 X' (5x2)

Матрица
эмбеддингов вершин

Graph Attention



Graph Attention 2



$$\alpha_{ij} = \frac{e^{a^T LeakyReLU(W[x_i || x_j])}}{\sum_{k \in N_i} e^{a^T LeakyReLU(W[x_i || x_k])}}$$

Сравните

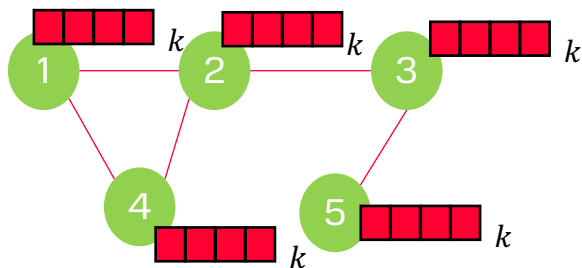
$$\alpha_{ij} = \frac{e^{\text{LeakyReLU}(a^T[Wx_i||Wx_j])}}{\sum_{k \in N_i} e^{\text{LeakyReLU}(a^T[Wx_i||Wx_k])}}$$

vs

$$\alpha_{ij} = \frac{e^{a^T \text{LeakyReLU}(W[x_i||x_j])}}{\sum_{k \in N_i} e^{a^T \text{LeakyReLU}(W[x_i||x_k])}}$$

GATConv2

conv.GATv2Conv



$$\alpha_{ij} = \frac{e^{\text{LeakyReLU}(a^T[Wx_i || Wx_j])}}{\sum_{k \in N_i} e^{\text{LeakyReLU}(a^T[Wx_i || Wx_k])}}$$

\hat{A} (5x5)

1	1			
1	1			
		1		1
		1		1
		1		1

X (5x4)

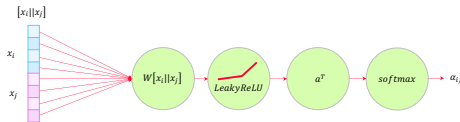
Фичи вершин

W (4x2)

Матрица весов –
обучаем

X' (5x2)

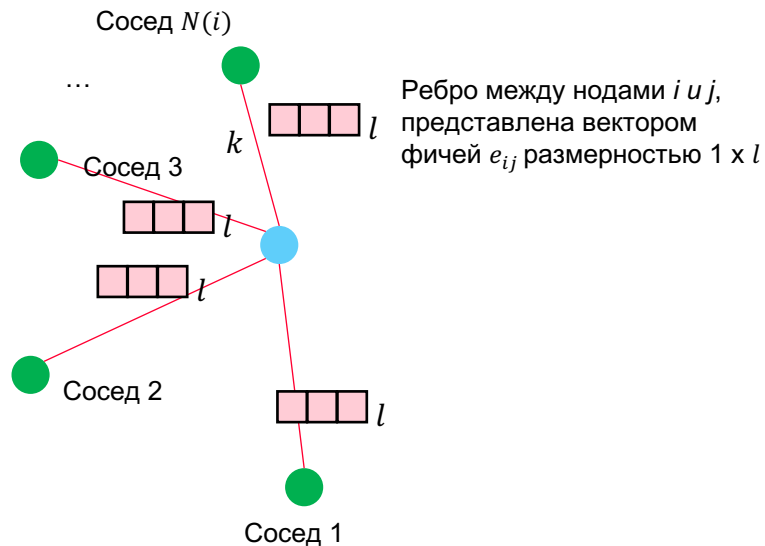
Матрица
эмбеддингов вершин



Графовая свертка NNConv



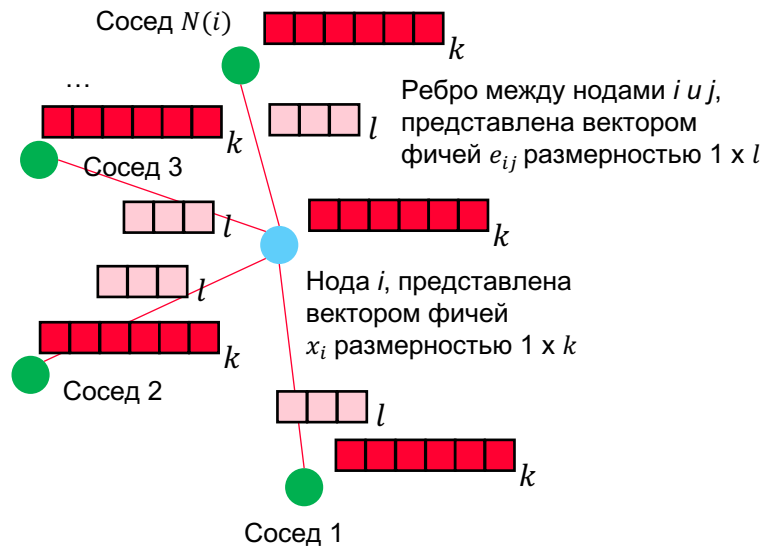
$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i + \sum_{j \in [1, N(i)]} \mathbf{x}_j \mathbf{NN}(e_{ij})$$



Графовая свертка NNConv



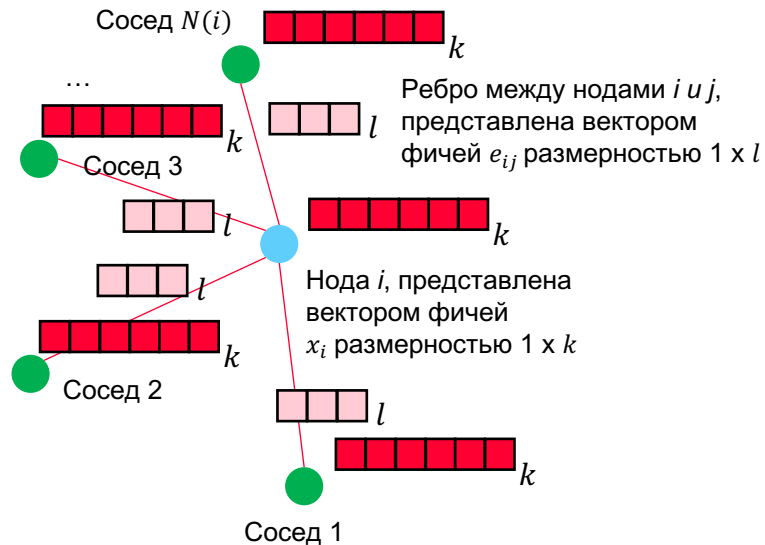
$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} x_j NN(e_{ij})$$



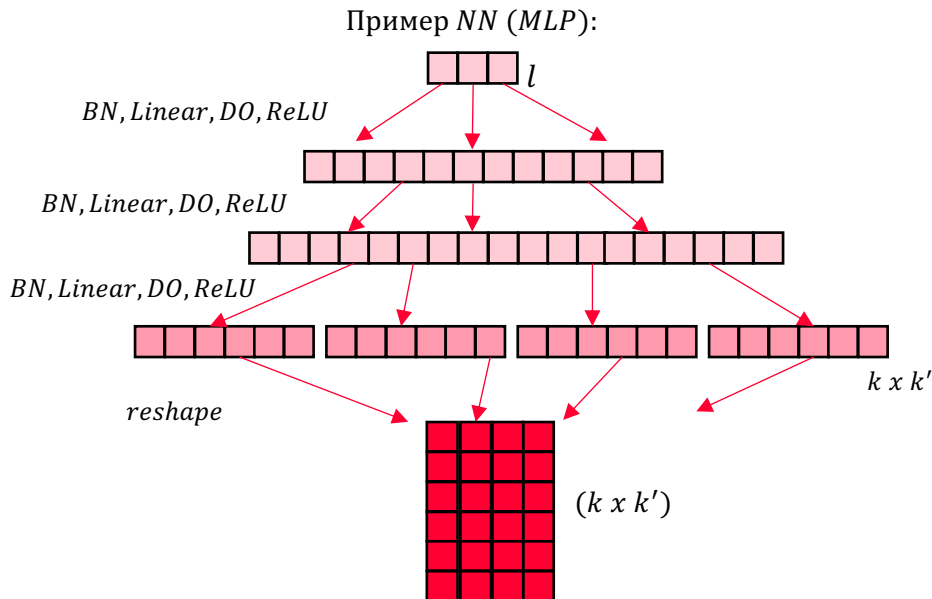
Графовая свертка NNConv



$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} x_j NN(e_{ij})$$



H и выходной слой NN имеют размерность $k \times k'$:
 $(1 \times k) * (k \times k') = 1 \times k'$
Входной слой NN имеет размерность l
(NN — часто MLP)

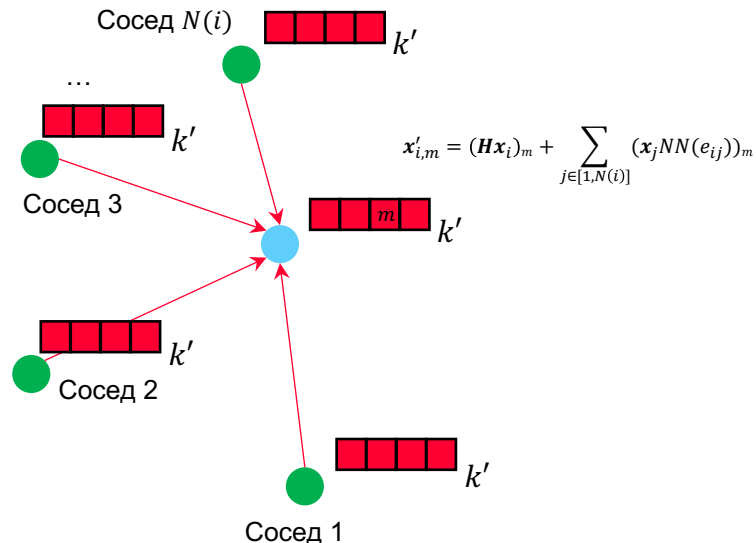
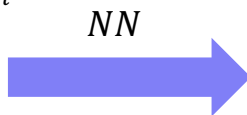
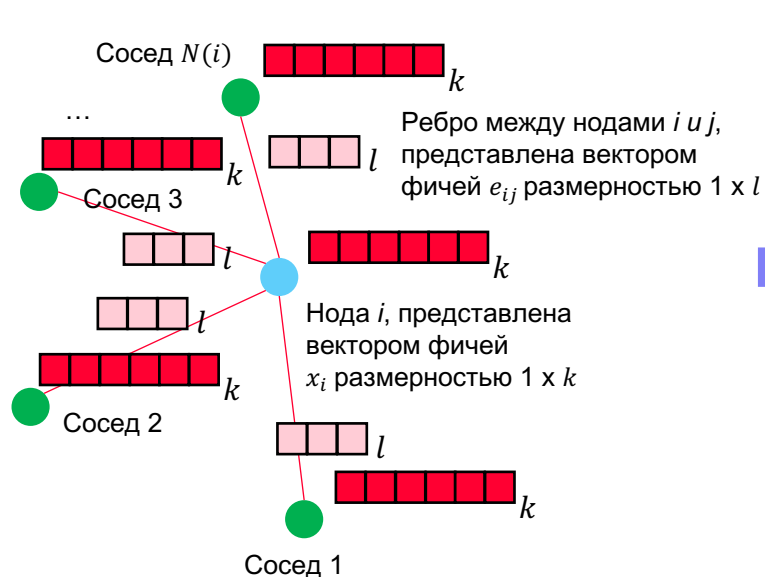


Графовая свертка NNConv



$$x'_i = Hx_i + \sum_{j \in [1, N(i)]} x_j NN(e_{ij})$$

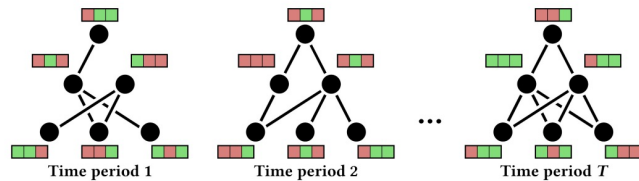
H и выходной слой NN имеют размерность $k \times k'$:
 $(1 \times k) * (k \times k') = 1 \times k'$
 Входной слой NN имеет размерность l
 (NN — часто MLP)



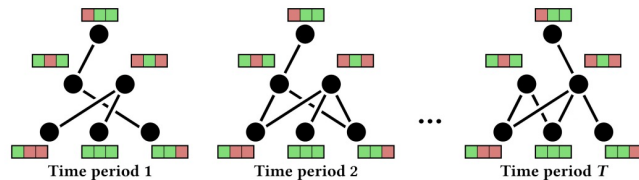
Темпоральные графы

02

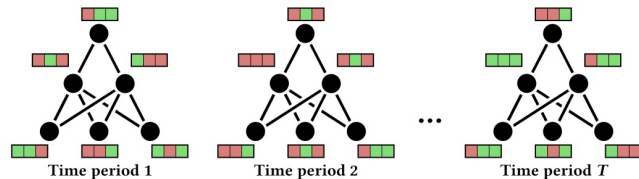
Типы темпоральных графов



(a) Dynamic graph with temporal signal.



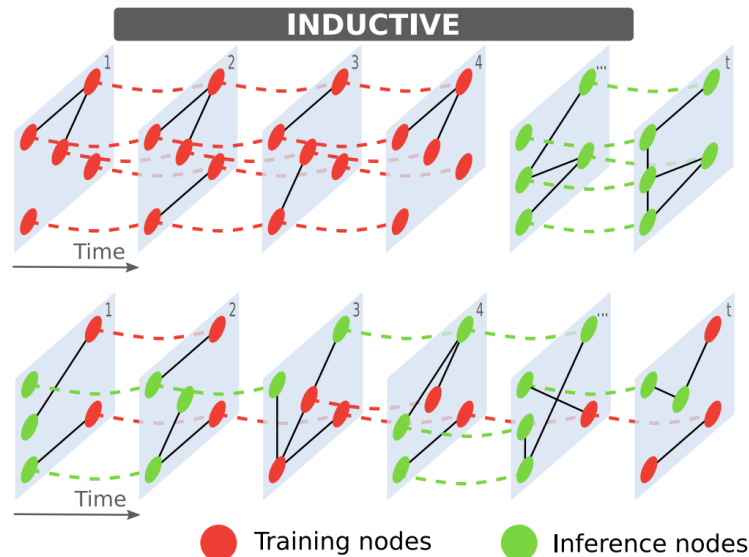
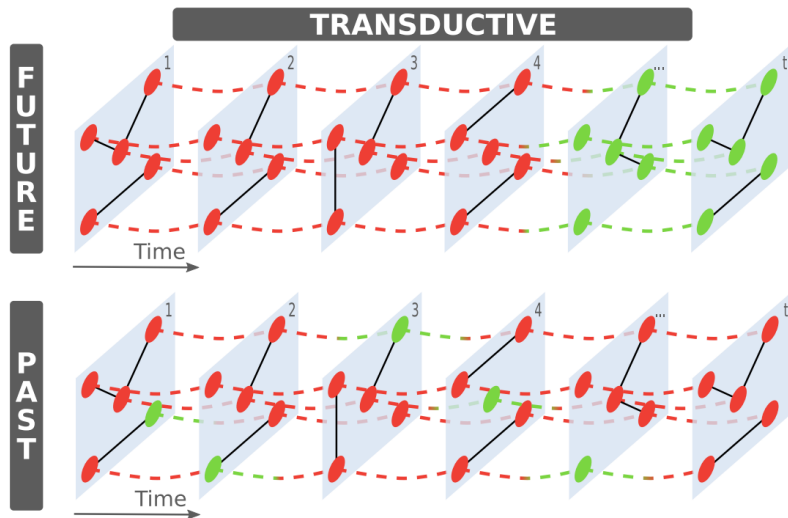
(b) Dynamic graph with static signal.



(c) Static graph with temporal signal.

Типы и примеры задач

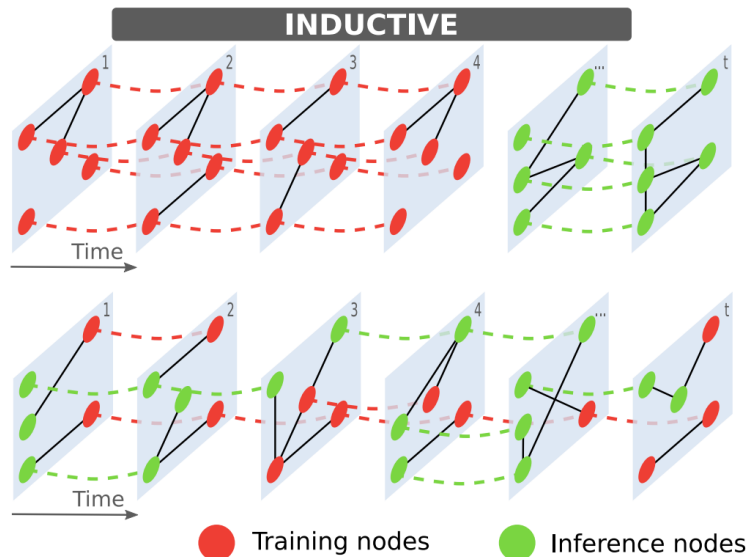
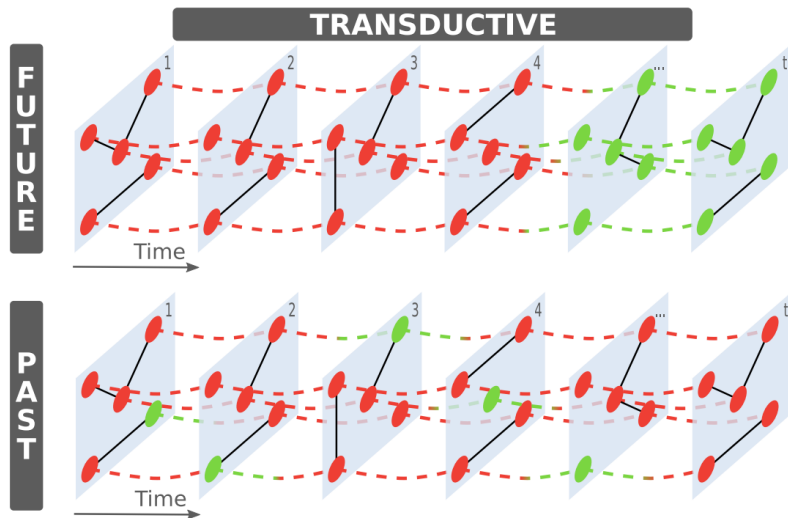
(FT) предикт класса ноды в будущем, *предсказание трафика*



Типы и примеры задач

(FT) предикт класса ноды в будущем, *предсказание траффика*

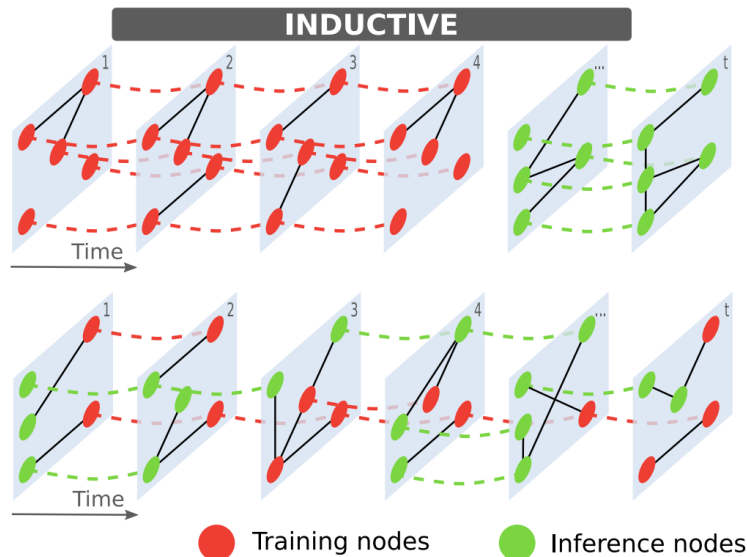
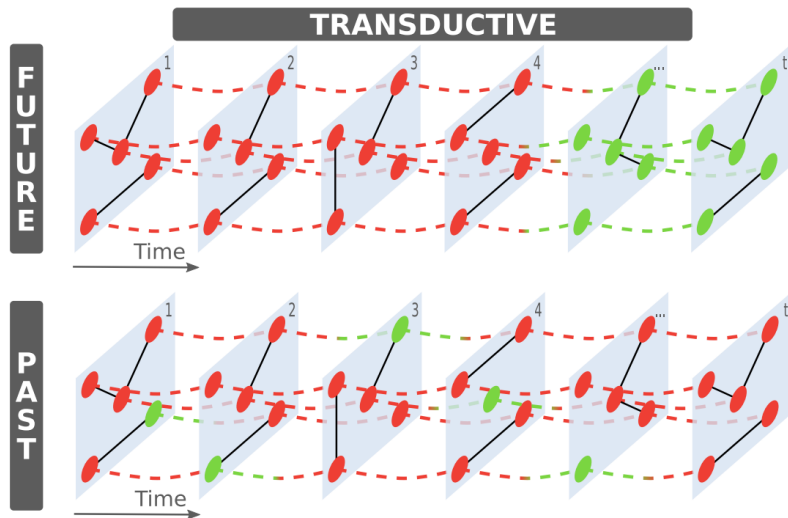
FI (future-inductive) – предикт класса **НОВОЙ** ноды (не было в обучении) в будущем



Типы и примеры задач

(FT) предикт класса ноды в будущем, *предсказание траффика*

FI (future-inductive) – предикт класса **НОВОЙ** ноды (не было в обучении) в будущем

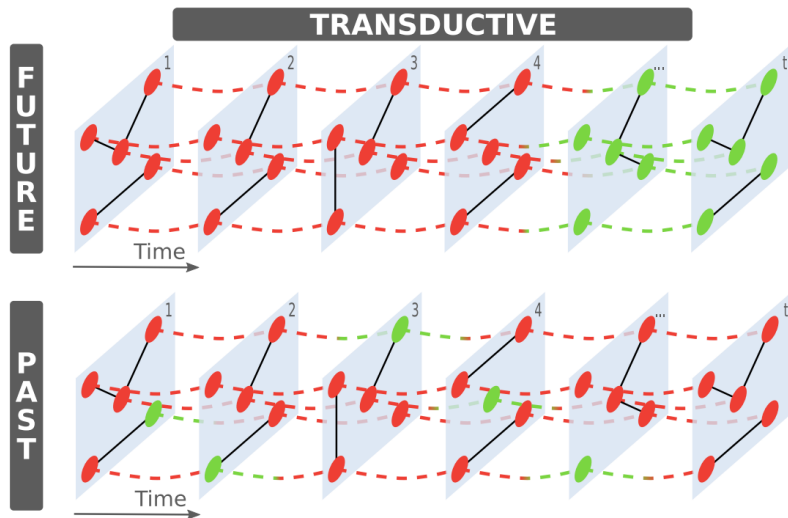


PT: Detecting critical stages during disease progression from gene expression profiles (Gao et al., 2022)

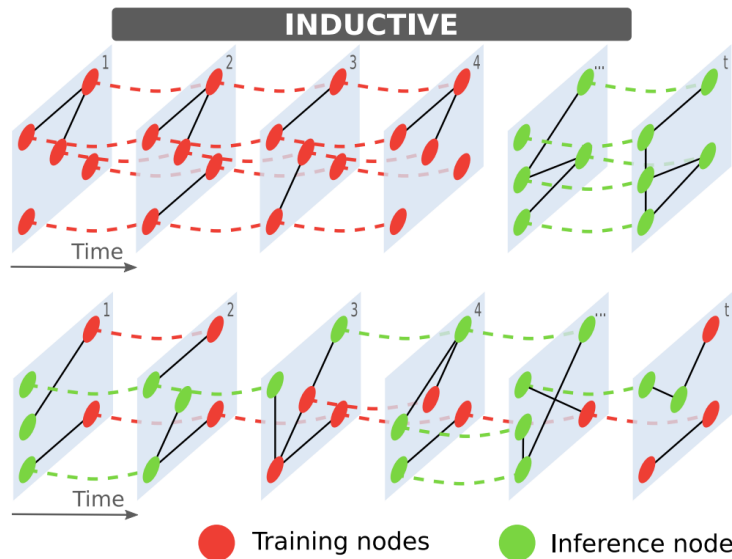
Типы и примеры задач

(FT) предикт класса ноды в будущем, *предсказание траффика*

FI (future-inductive) – предикт класса **НОВОЙ** ноды (не было в обучении) в будущем



PT: Detecting critical stages during disease progression from gene expression profiles (Gao et al., 2022)



PI (past-inductive), предикт класса ноды, которая была недоступна (маскирована) в трейне или предикт ребра между нодами, которых не было в обучении

Temporal Graph Neural Networks

- ▲ Attention
 ■ RNN/GRU
 ● Autoencoder

Snapshot-based

Model Evolution

EvolveGCN

Embedding Evolution

VGRNN

DySAT

DynGESN

Roland

SSGNN

Event-based

Temporal Embedding

TGAT

NAT

TGL

Temporal Neighborhood

APAN

DGNN

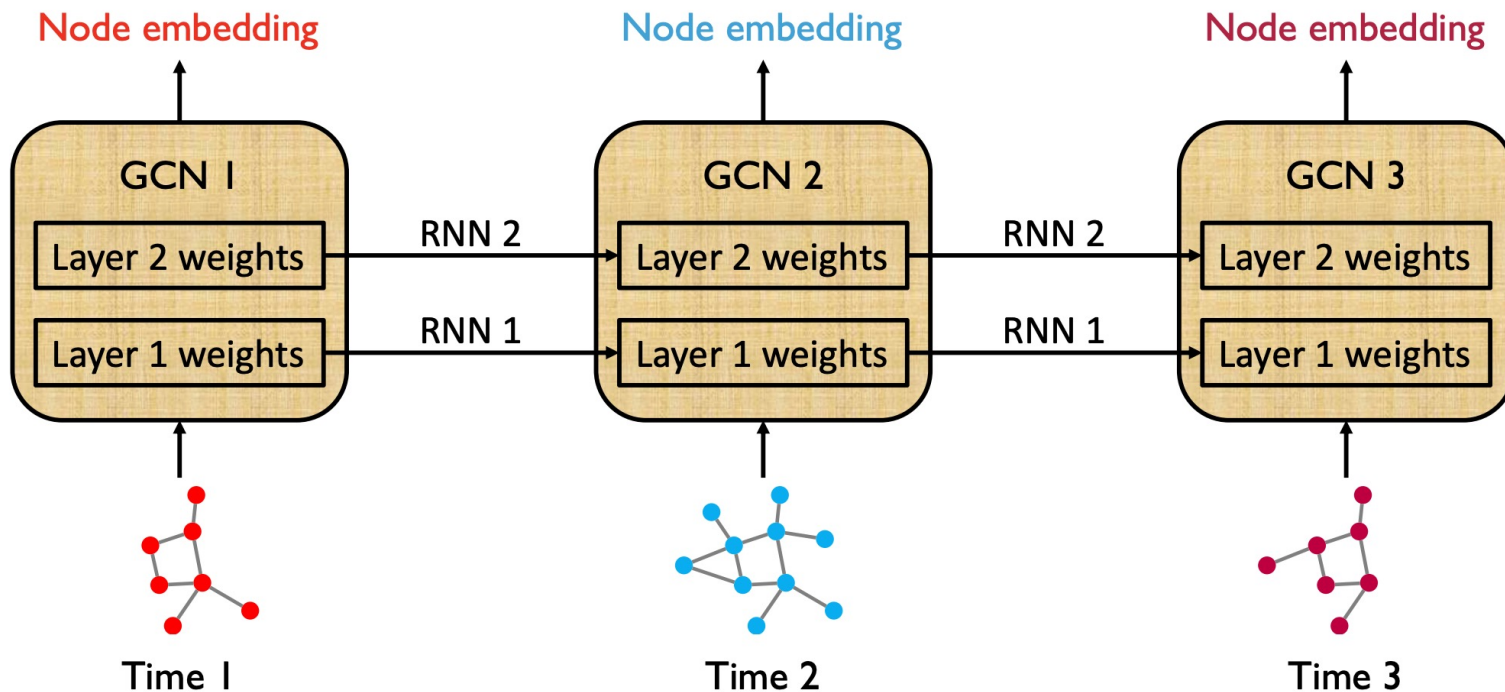
TGN

задача

f_{NC}	FT	X	X	X	X	X	FT, FI	X	FT	FT	PT	FT
f_{EC}	FT	X	X	X	X	X	X	X	X	FT	X	X
f_{GC}	X	X	X	PT	X	X	X	X	X	X	X	X
f_{LP}	FT	FI, FT, PT	PT, FT	X	FT	X	FT, FI	PI, FT	FT	FT	FT	FI, FT
f_{EP}	X	X	X	X	X	X	X	X	X	X	X	X
Clustering	X	X	X	X	X	X	X	X	X	X	X	X
LDE	X	X	X	X	X	X	X	X	X	X	X	X
Regression _(node)	X	X	X	FT	X	FT	X	X	X	X	X	X
Anomaly detect.	X	X	X	X	X	X	X	X	X	X	X	X

LDE == low-dimension embedding

EvolveGCN (2019)



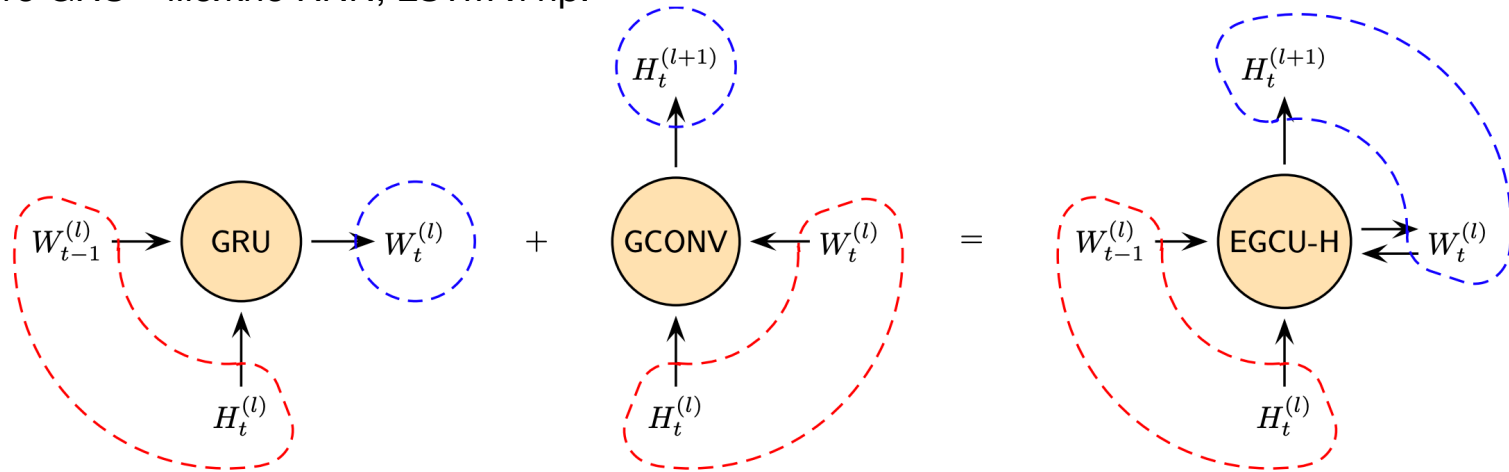
На вход RNN подавать не фичи / эмбединги вершин, а веса графовой свертки

EvolveGCN-H (2019)

$$H_t^{l+1} = \sigma(A_t H_t^l W_t^l)$$

$$W_t^l = GRU(H_t^l, W_{t-1}^l)$$

Вместо GRU – можно RNN, LSTM и пр.

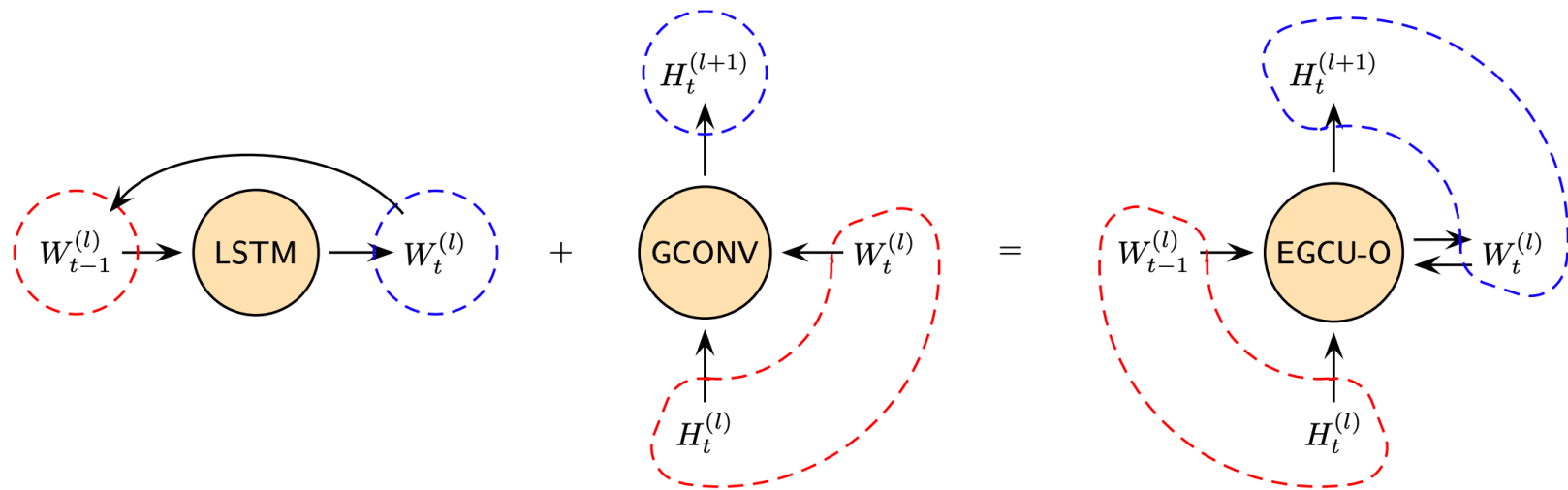


(a) EvolveGCN-H, where the GCN parameters are hidden states of a recurrent architecture that takes node embeddings as input.

EvolveGCN-O (2019)

$$H_t^{l+1} = \sigma(A_t H_t^l W_t^l)$$

$$W_t^l = LSTM(W_{t-1}^l)$$



(b) EvolveGCN-O, where the GCN parameters are input/outputs of a recurrent architecture.

В июне 2023 появился первый бенчмарк



<https://tgb.complexdatalab.com/>



<https://torch-spatiotemporal.readthedocs.io/en/latest/>

<https://github.com/TorchSpatiotemporal/tsl>

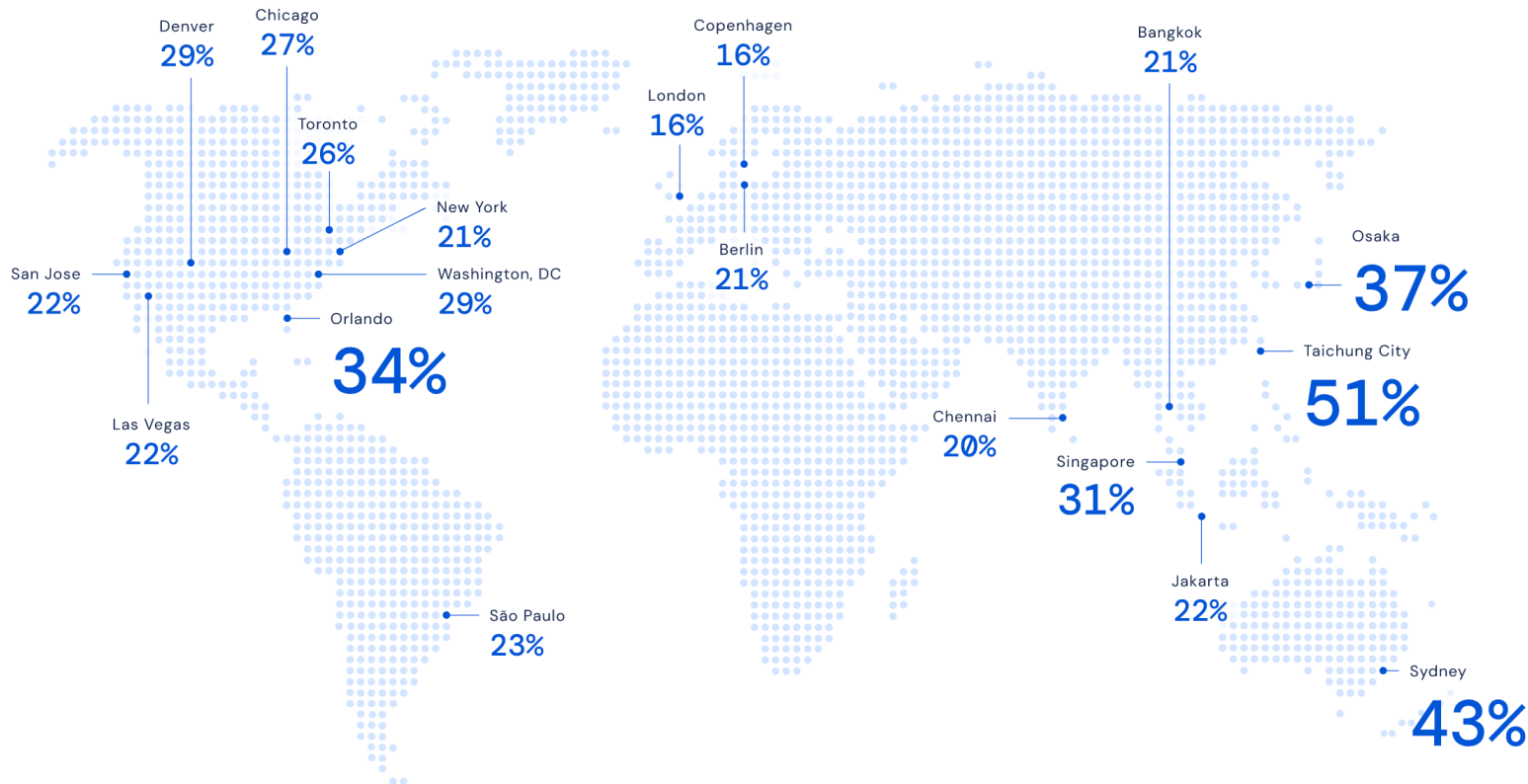


<https://www.youtube.com/watch?v=rUR3-YIHft0>

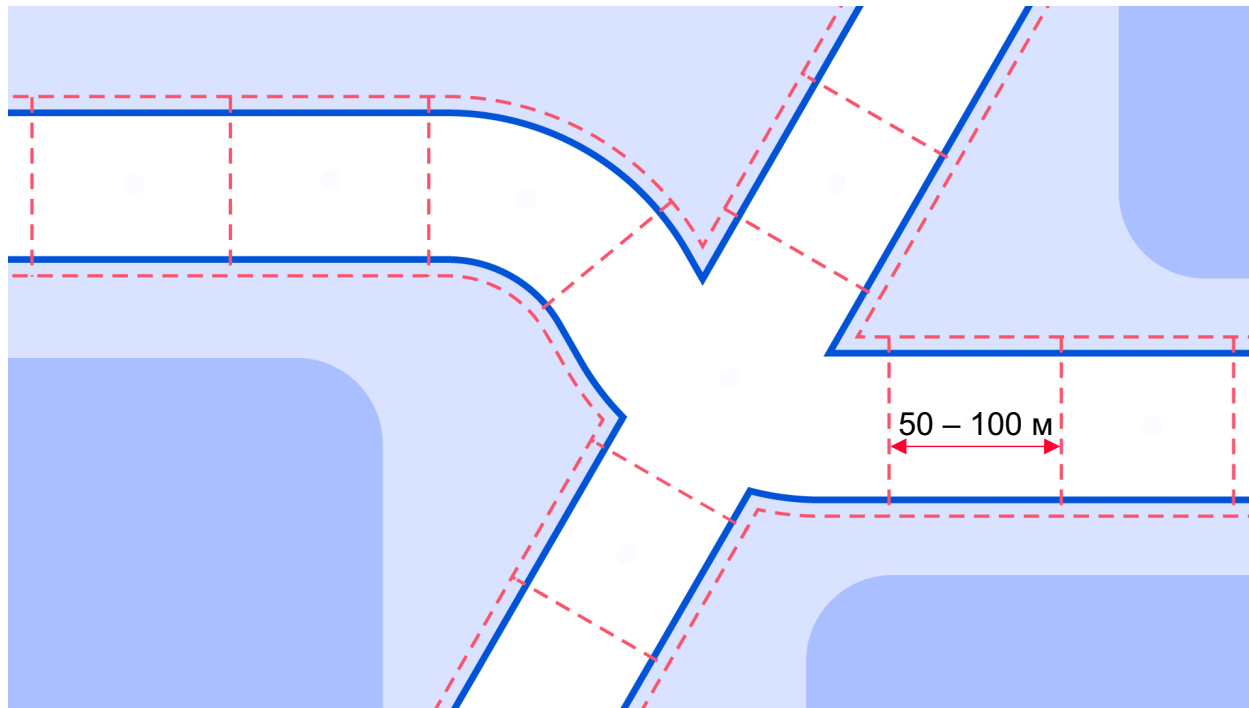
Задача ETA

03

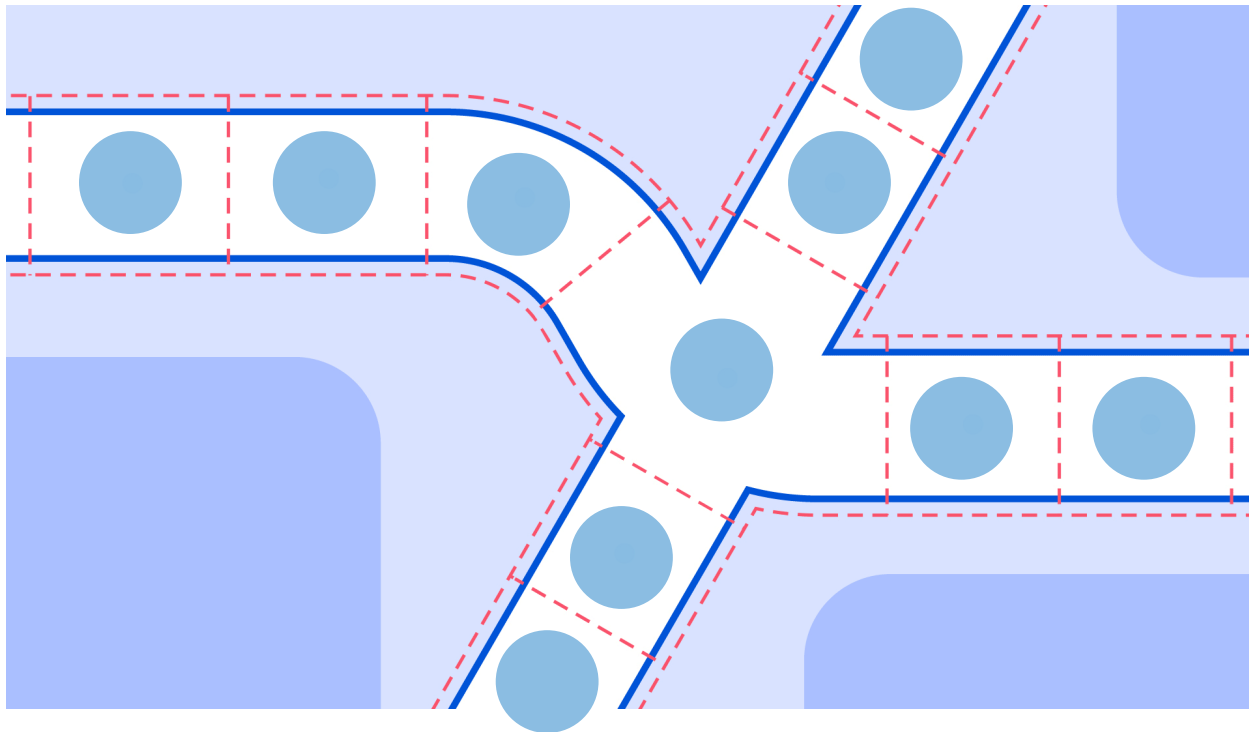
Эффект от GNN в задаче ETA в Google Maps



Дороги в графы



Дороги в графы



Фи́чи

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

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

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

Текущие и исторические скорости, длина сегмента и приоритет (напр.: хайвэй), время и пр.

Время проезда по сегмента и скорости. Плюс обучаемые эмбе́ддинги сегмента / суперсегмента

Схема работы

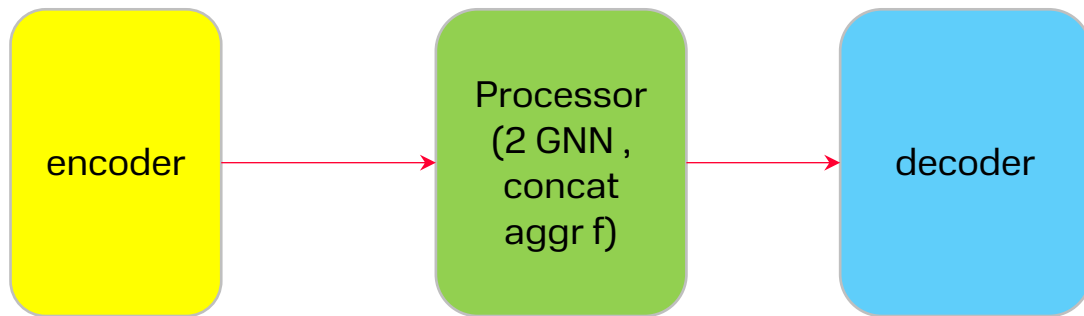
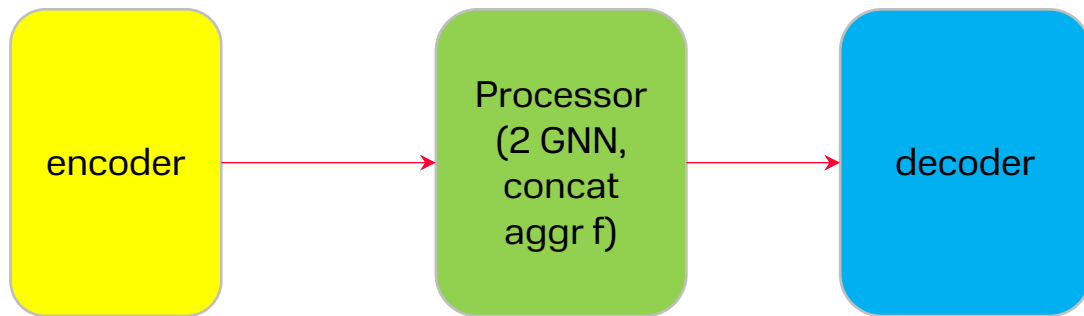
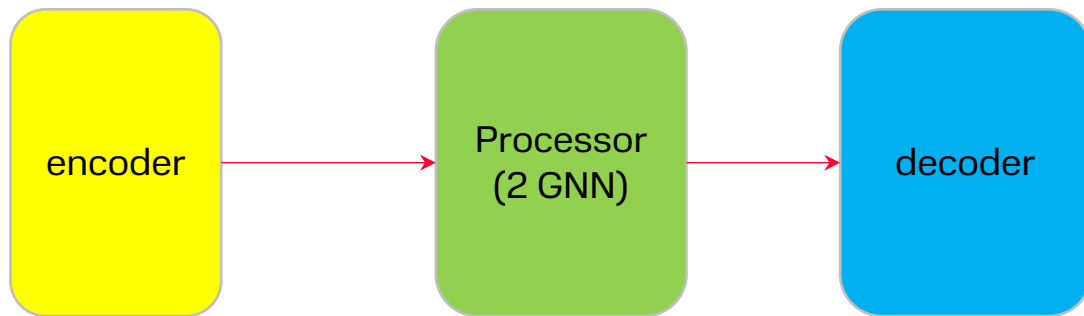


Схема работы



Не густо деталей. И где время?

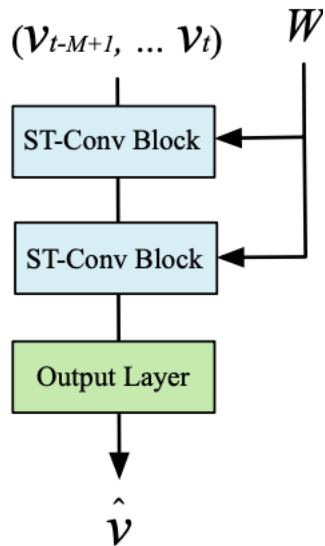
Схема работы



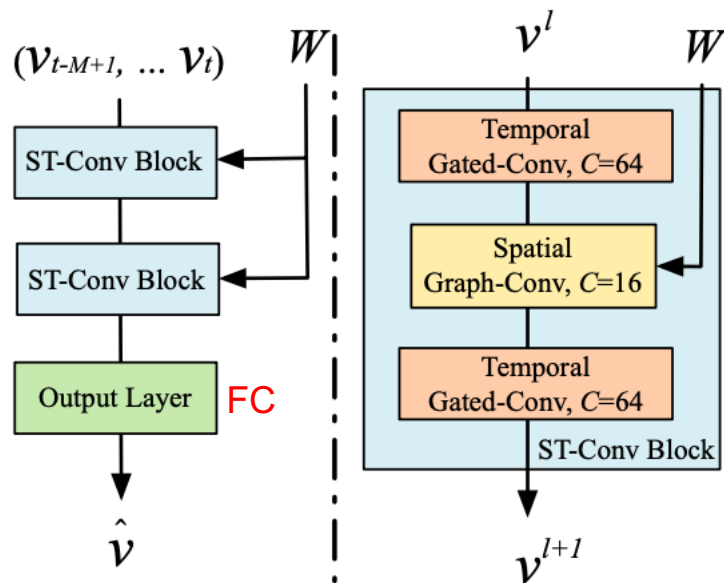
Не густо деталей. И где время?

5 GNN для пяти временных горизонтов

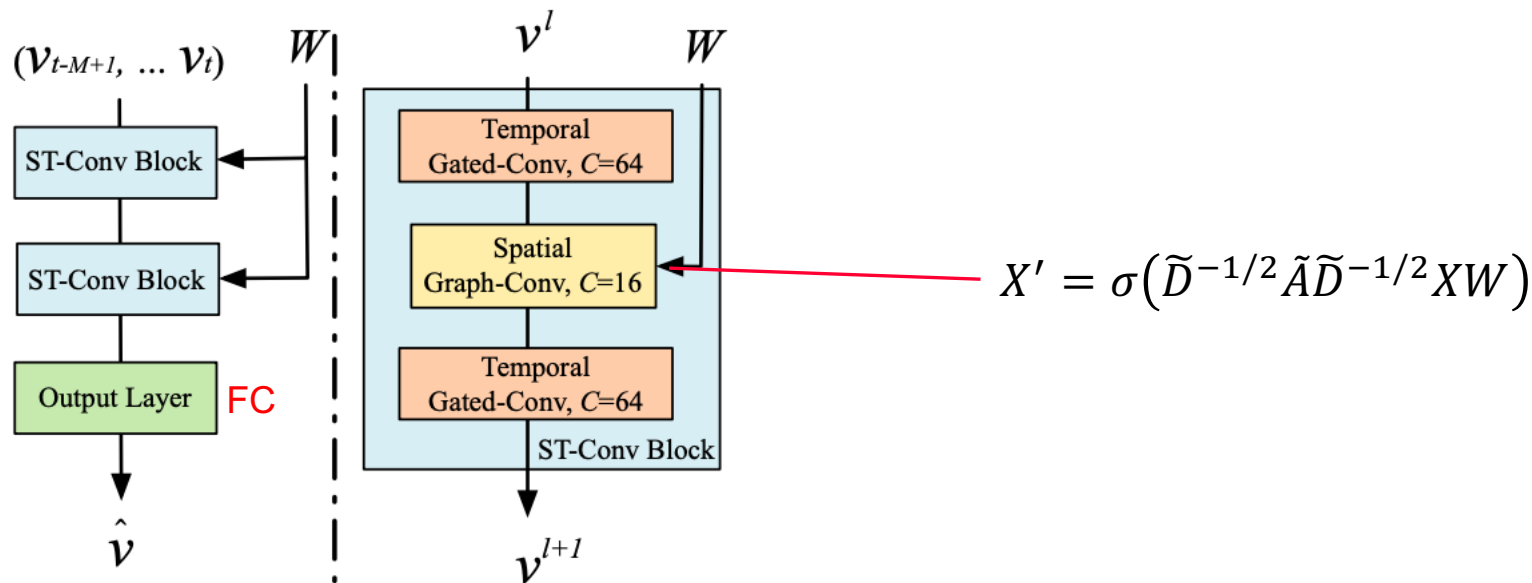
STGCN (2018)



STGCN (2018)

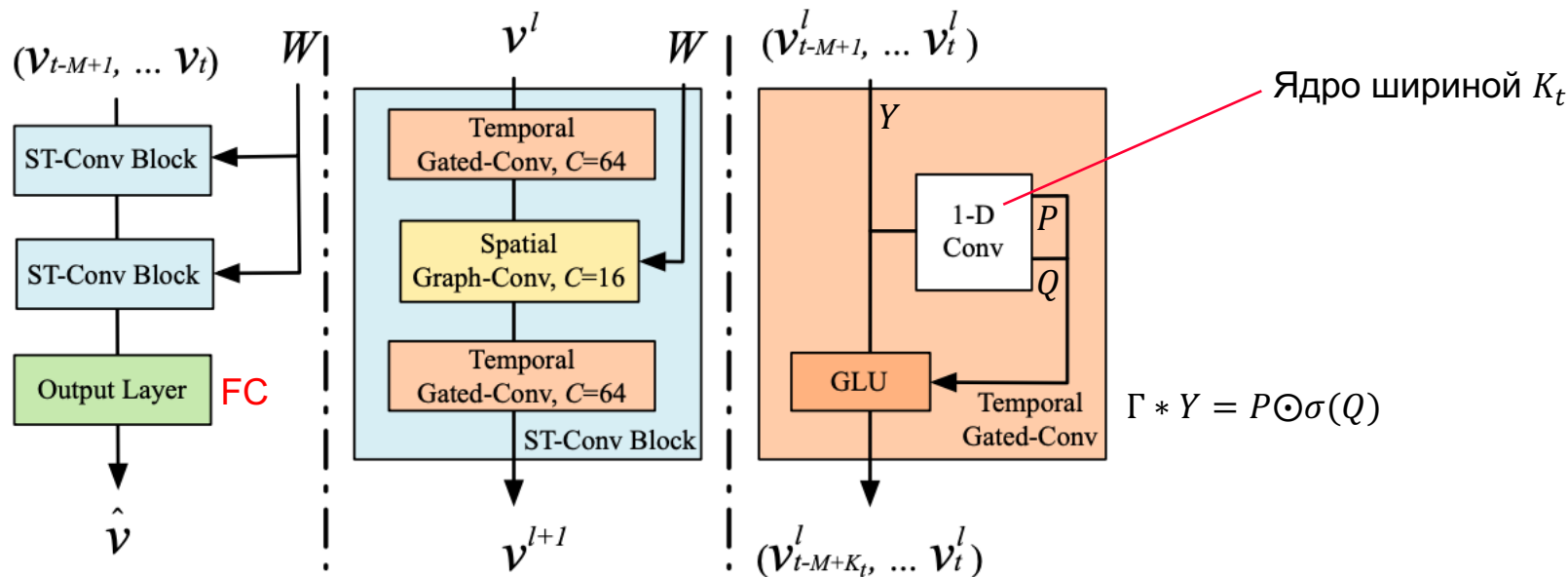


STGCN (2018)



$$X' = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} XW)$$

STGCN (2018)



GLU (борьба с затуханием градиентов)

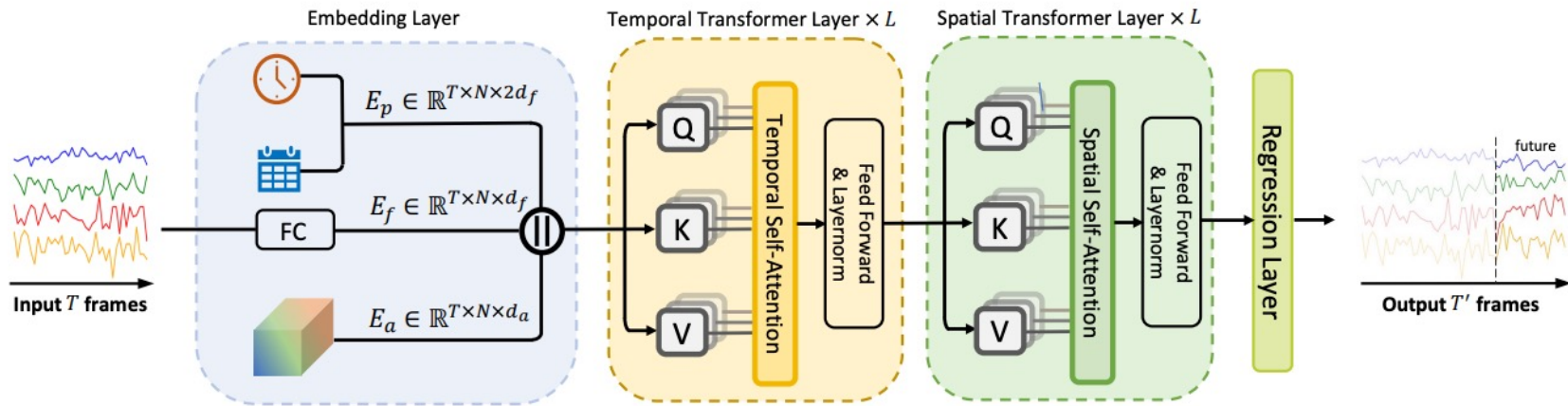
Градиент gate в LSTM

$$\begin{aligned}\nabla(\tanh(X) \odot \sigma(X)) &= \tanh'(X) \nabla X \odot \sigma(X) \\ &\quad + \sigma'(X) \nabla X \odot \tanh(X)\end{aligned}$$

Градиент gate в GLU

$$\nabla(X \odot \sigma(X)) = \nabla X \odot \sigma(X) + X \odot \sigma'(X) \nabla X$$

STAEformer



* Notably, we also apply layer normalization, residual connection and multi-head mechanism