

# Transformer but for Graphs

# Основной вопрос

Можем ли заставить это

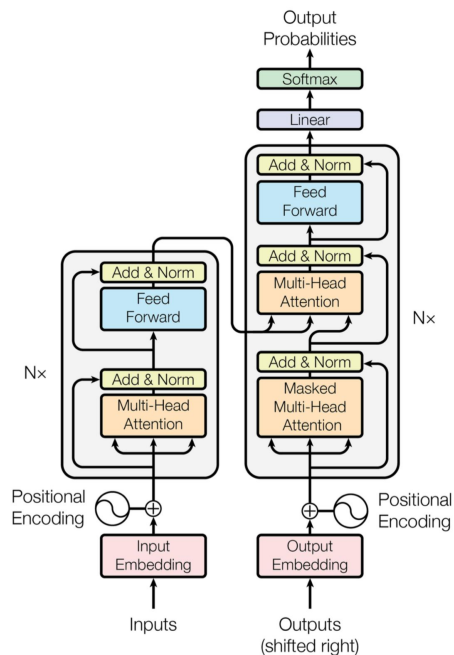
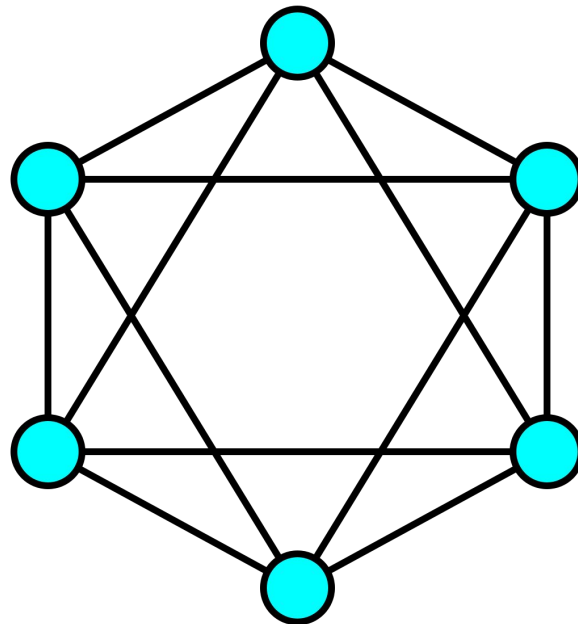


Figure 1: The Transformer - model architecture.

Можем ли заставить на этом



# Основной вопрос

Можем ли мы

Можем ли заставить

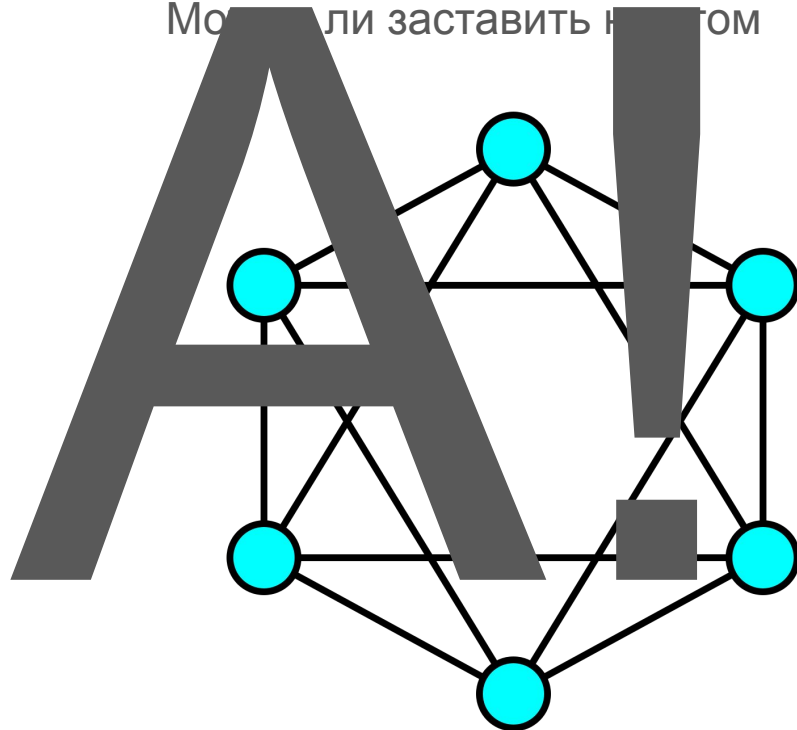
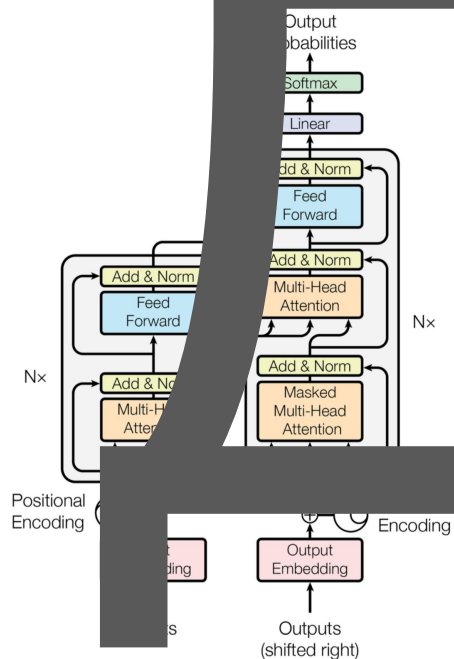


Figure 1: The Transformer - model architecture.

# Но перед этим немного про то как вообще работают с графами

AGGREGATE COMBINE состоит из 2х сложных шагов

$$a_i^{(l)} = \text{AGGREGATE}^{(l)} \left( \left\{ h_j^{(l-1)} : j \in \mathcal{N}(v_i) \right\} \right), \quad h_i^{(l)} = \text{COMBINE}^{(l)} \left( h_i^{(l-1)}, a_i^{(l)} \right), \quad (1)$$

Для графовой задачи добавляем доп шаг:

$$h_G = \text{READOUT} \left( \left\{ h_i^{(L)} \mid v_i \in G \right\} \right).$$

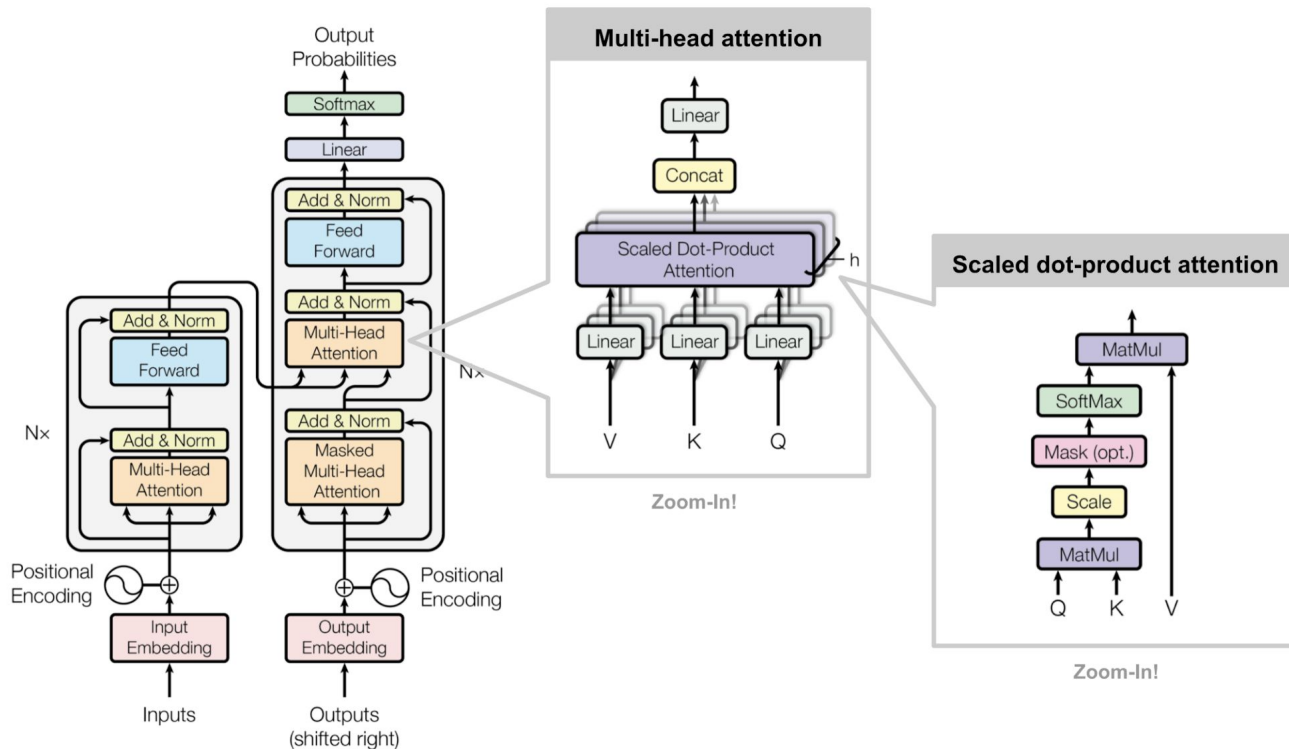
# Но перед этим немного про то как вообще работают трансформеры

Работать в  
основном будем  
с Attention:

$$Q = HW_Q, \quad K = HW_K, \quad V = HW_V,$$
$$A = \frac{QK^\top}{\sqrt{d_K}}, \quad \text{Attn}(H) = \text{softmax}(A) V,$$

# Но перед этим немного про то как вообще работают трансформеры

Работать в основном будем с Attention:

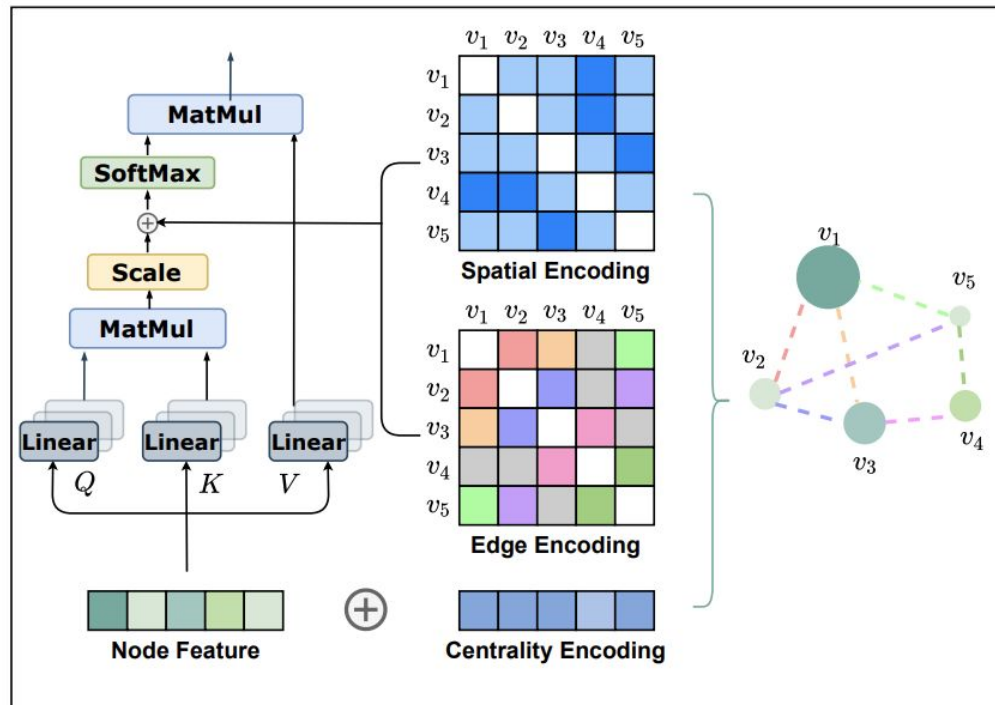


# Обратно к трансформерам: Как мы этого добьемся?

Добавляем ряд  
модификаций в  
self-attention

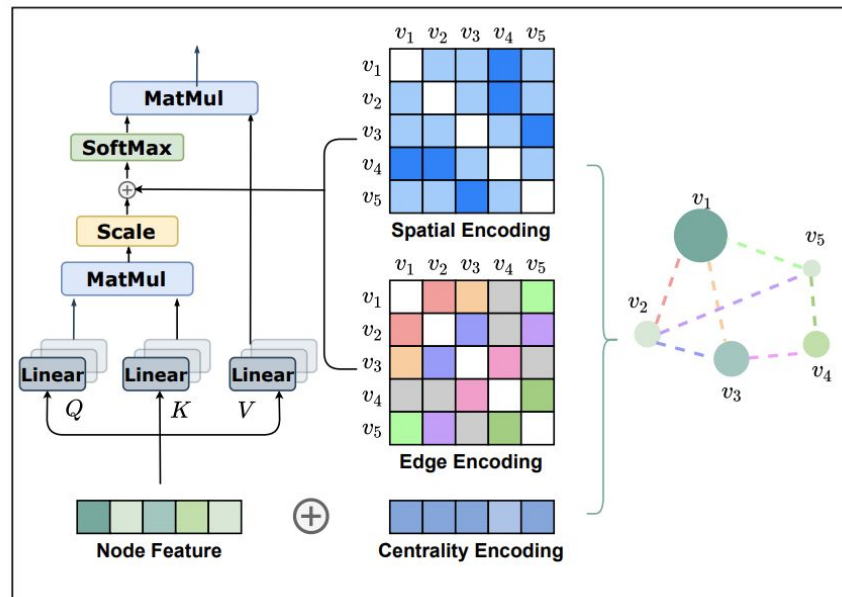
3 основные компоненты:

- Centrality Encoding
- Spatial Encoding
- Edge Encoding



# Centrality Encoding

$$h_i^{(0)} = x_i + z_{\text{deg}^-(v_i)}^- + z_{\text{deg}^+(v_i)}^+,$$





# Spatial Encoding

$$A_{ij} = \frac{(h_i W_Q)(h_j W_K)^T}{\sqrt{d}} + b_{\phi(v_i, v_j)},$$

# Edge Encoding in the Attention

$$A_{ij} = \frac{(h_i W_Q)(h_j W_K)^T}{\sqrt{d}} + b_{\phi(v_i, v_j)} + c_{ij}, \text{ where } c_{ij} = \frac{1}{N} \sum_{n=1}^N x_{e_n} (w_n^E)^T,$$

# Graphormer Layer

Дальше стандартный трансформер:

$$h'^{(l)} = \text{MHA}(\text{LN}(h^{(l-1)})) + h^{(l-1)}$$

$$h^{(l)} = \text{FFN}(\text{LN}(h'^{(l)})) + h'^{(l)}$$

# Special Node

Добавляем специальную [VNode], связанную со всеми вершинами в графе.

Для нее отдельные  $\phi$  ([VNode],  $x_i$ ). Так что выносим обучаемые параметры отдельно

Полезно для агрегации всей информации на графе

# OGB Large-Scale Challenge

method	#param.	train MAE	validate MAE
GCN [26]	2.0M	0.1318	0.1691 (0.1684*)
GIN [54]	3.8M	0.1203	0.1537 (0.1536*)
GCN-VN [26, 15]	4.9M	0.1225	0.1485 (0.1510*)
GIN-VN [54, 15]	6.7M	0.1150	0.1395 (0.1396*)
GINE-VN [5, 15]	13.2M	0.1248	0.1430
DeeperGCN-VN [30, 15]	25.5M	0.1059	0.1398
GT [13]	0.6M	0.0944	0.1400
GT-Wide [13]	83.2M	0.0955	0.1408
Graphormer <sub>SMALL</sub>	12.5M	0.0778	0.1264
Graphormer	47.1M	<b>0.0582</b>	<b>0.1234</b>

Эксперименты на PCQM4M-LSC

# GNN выносим ногами вперед

Table 2: Results on MolPCBA.

method	#param.	AP (%)
DeeperGCN-VN+FLAG [30]	5.6M	28.42±0.43
DGN [2]	6.7M	28.85±0.30
GINE-VN [5]	6.1M	29.17±0.15
PHC-GNN [29]	1.7M	29.47±0.26
GINE-APPNP [5]	6.1M	29.79±0.30
GIN-VN[54] (fine-tune)	3.4M	29.02±0.17
Graphormer-FLAG	119.5M	<b>31.39±0.32</b>

Table 3: Results on MolHIV.

method	#param.	AUC (%)
GCN-GraphNorm [5, 8]	526K	78.83±1.00
PNA [10]	326K	79.05±1.32
PHC-GNN [29]	111K	79.34±1.16
DeeperGCN-FLAG [30]	532K	79.42±1.20
DGN [2]	114K	79.70±0.97
GIN-VN[54] (fine-tune)	3.3M	77.80±1.82
Graphormer-FLAG	47.0M	<b>80.51±0.53</b>

Table 4: Results on ZINC.

method	#param.	test MAE
GIN [54]	509,549	0.526±0.051
GraphSage [18]	505,341	0.398±0.002
GAT [50]	531,345	0.384±0.007
GCN [26]	505,079	0.367±0.011
GatedGCN-PE [4]	505,011	0.214±0.006
MPNN (sum) [15]	480,805	0.145±0.007
PNA [10]	387,155	0.142±0.010
GT [13]	588,929	0.226±0.014
SAN [28]	508,577	0.139±0.006
Graphormer <sub>SLIM</sub>	489,321	<b>0.122±0.006</b>

# Ablation

Node Relation Encoding		Centrality	Edge Encoding			valid MAE
Laplacian PE[13]	Spatial		via node	via Aggr	via attn bias(Eq.7)	
-	-	-	-	-	-	0.2276
✓	-	-	-	-	-	0.1483
-	✓	-	-	-	-	0.1427
-	✓	✓	-	-	-	0.1396
-	✓	✓	✓	-	-	0.1328
-	✓	✓	-	✓	-	0.1327
-	✓	✓	-	-	✓	0.1304

# Transformers can be used on graphs!

И сюда их справились запихнуть...