Do Transformers Really Perform Bad for Graph Representation?

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, Tie-Yan Liu

NeurIPS 2021

0. Index

01. Introduction

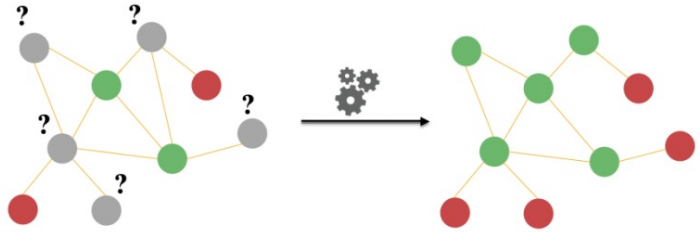
01. Introduction

Do **Transformers** Really Perform Bad for **Graph Representation**?

01. Introduction

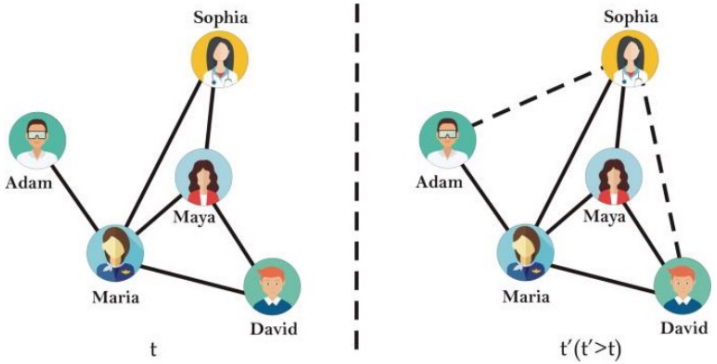
Graph Network Task

Node Classification



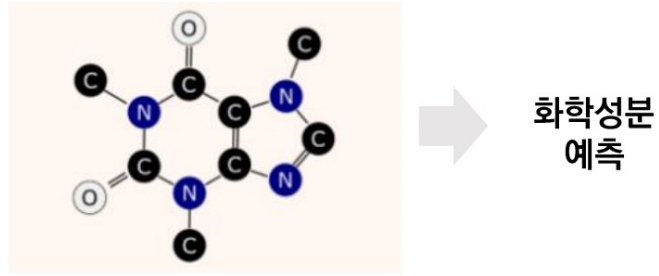
- Node 의 label 을 예측하는 task
- 예 : 논문의 카테고리 예측
- Node Representation 중요
- X : Node → Y : Node label

Link Prediction



- Node 간 Missing Edge 를 예측하는 task
- 예 : Social network 의 친구추천
- Node Representation 중요
- X : Node pair → Y : edge

Graph Classification

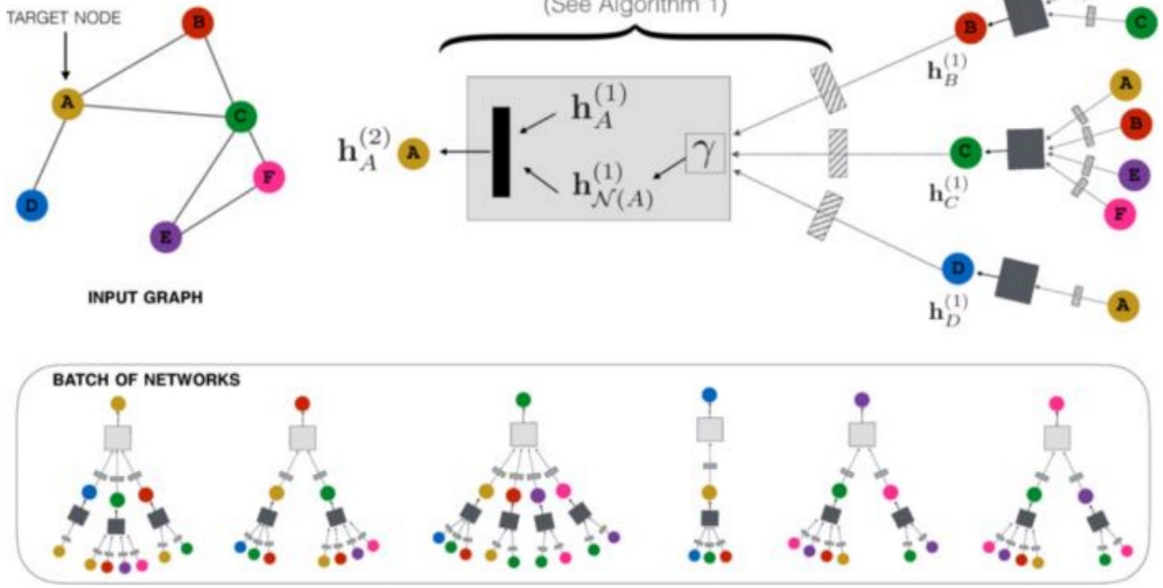


- Graph 의 class 를 분류하는 task
- 예 : 분자구조의 화학속성 예측
- Graph Representation 중요
- X : Nodes, edges → Y : graph label

01. Introduction

Graph Network Task

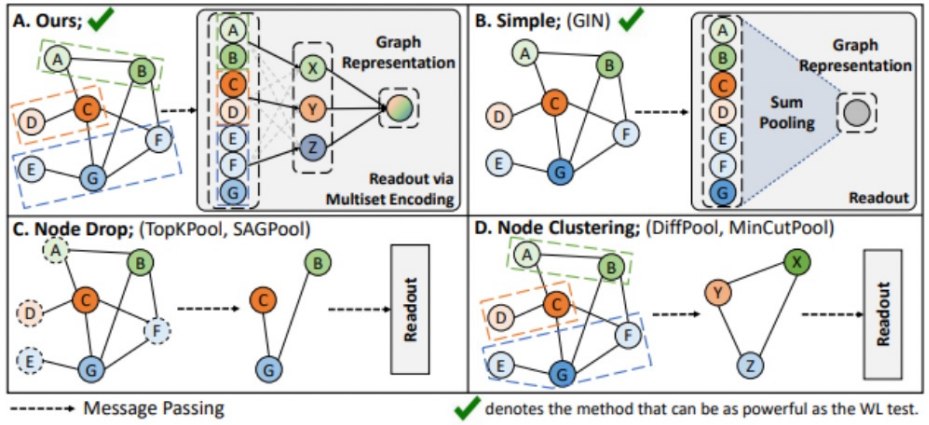
Node Classification



Link Prediction

GNN 이 매우 적합한 모델

Graph Classification



Node 값을 다시 통합하는 과정 필요

1) Sum pooling, Mean pooling
2) 대표 Node 를 추출
3) FC layer 를 통해 학습

01. Introduction

Graph Network Task

Transformer

- Graph를 나타내는 모든 node와 edge를 하나의 Context로 표현

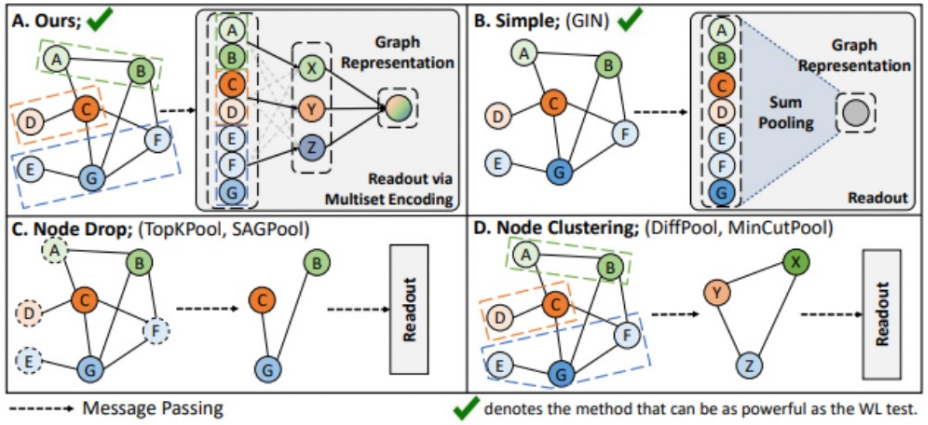
+

- 그래프의 구조적인 특징을 반영할 수 있는 정보 사용
  - ✓ 그래프의 구조적 특징이란?
    1. 그래프의 노드는 순서가 없다 (절대적인 위치 정보가 없다)
    2. Edge를 통한 연결 정보만 있을 뿐, 거리는 없다

Graphormer

- ✓ Transformer 구조에 그래프의 구조적인 특징을 반영할 수 있게 함
- ✓ 기존 GNN은 node 단위 학습이 이루어지고, 1-hop의 정보를 Layer를 쌓으면서 Multi-hop 정보를 받을 수 있음 (Local)
- ✓ Multi-hop 정보를 Self-attention을 통해 한번에 학습

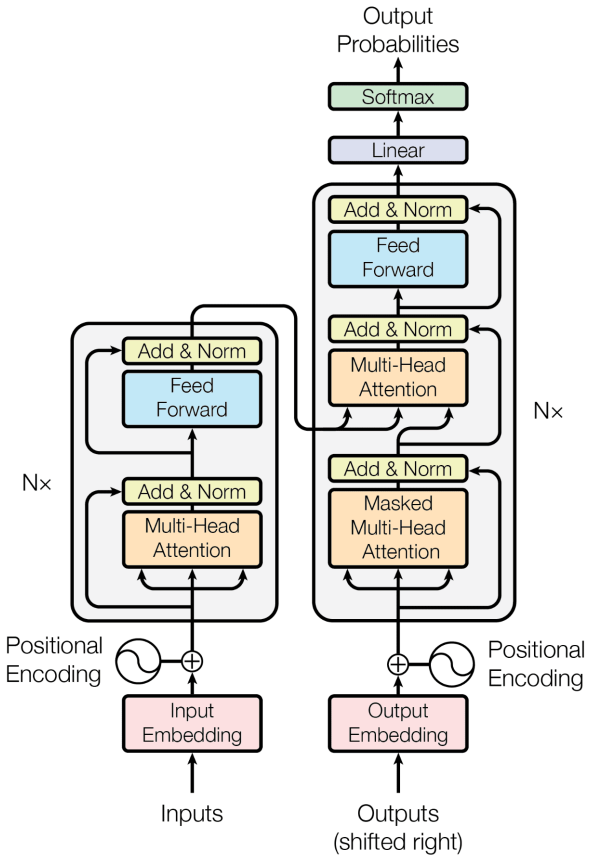
Graph Classification



그래프 자체를 표현할 수 있는 모델을 만들자!

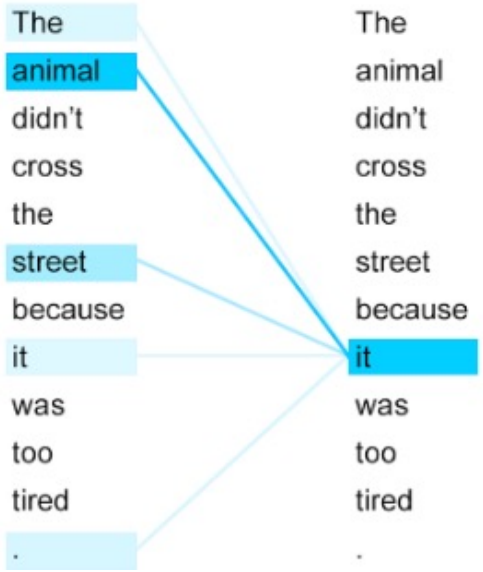
01. Introduction

Self attention



They love a song by **taylor**, a **singer** from their hometown.

Q: who is the singer?



01. Introduction

Self attention

(유사한 정도의 Softmax)

Q: 귤

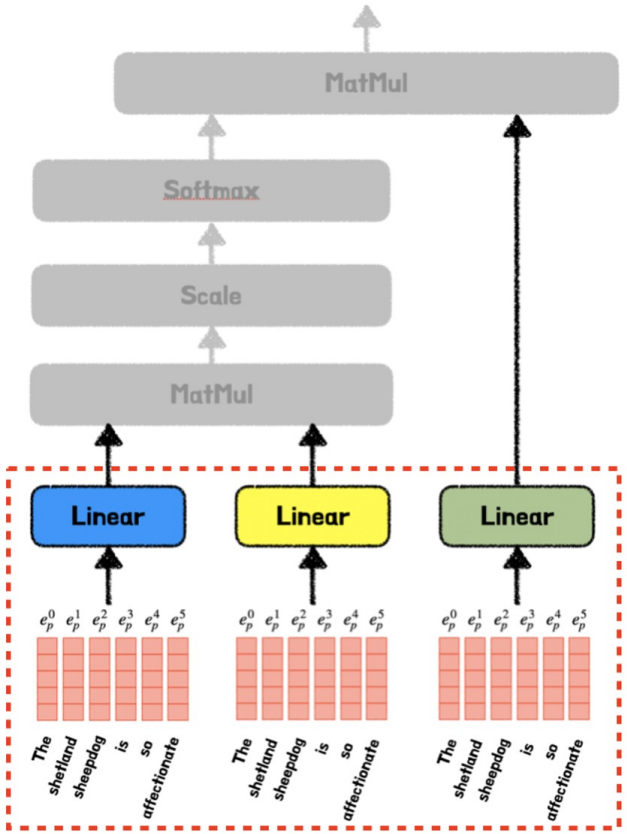
Key (K)	Similarity Sim(K, Q)	Value (V)	Sim(K, Q) * V
레몬	0.35	새콤한맛	0.35 * 새콤한맛
오렌지	0.64	달콤한맛	0.64 * 달콤한맛
아보카도	0.01	크레파스맛	0.01 * 크레파스맛

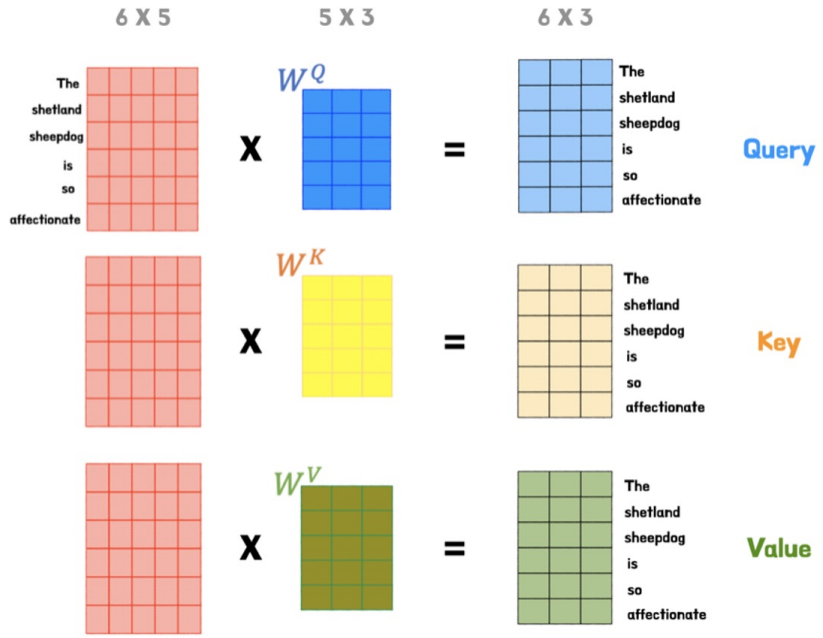
Attention Score

-> 새콤달콤한맛!

01. Introduction

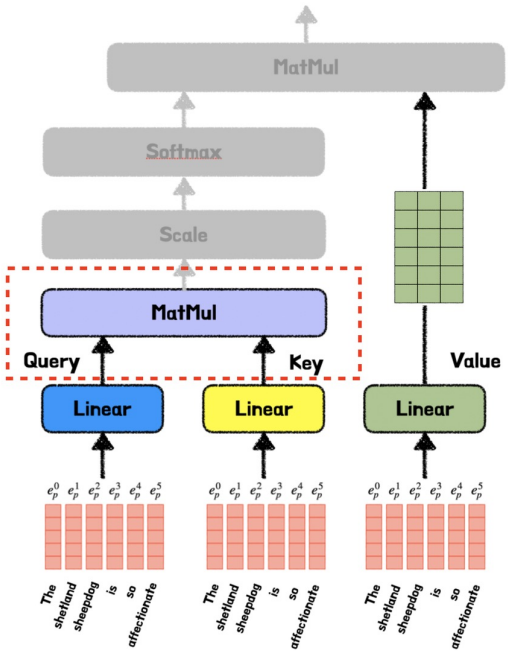
Self attention

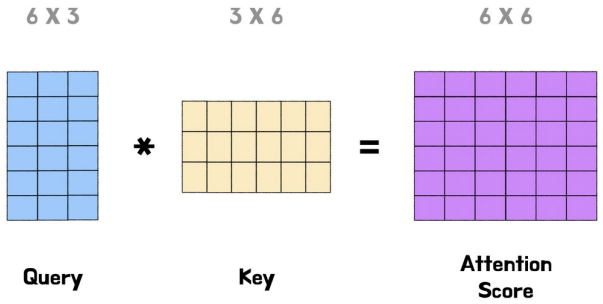


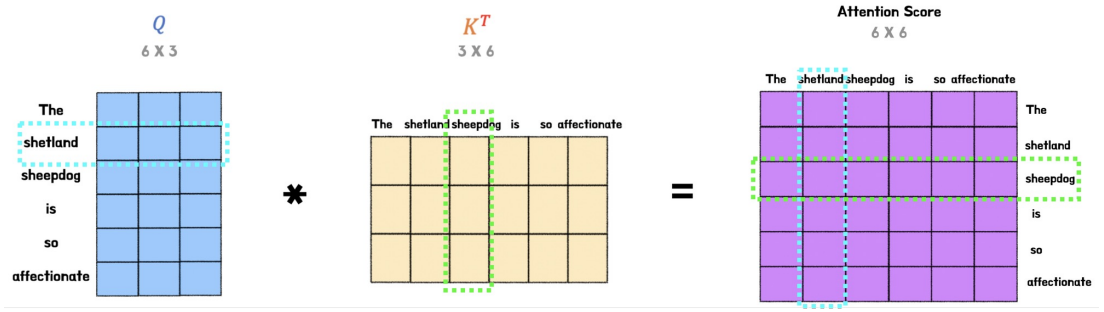


01. Introduction

Self attention

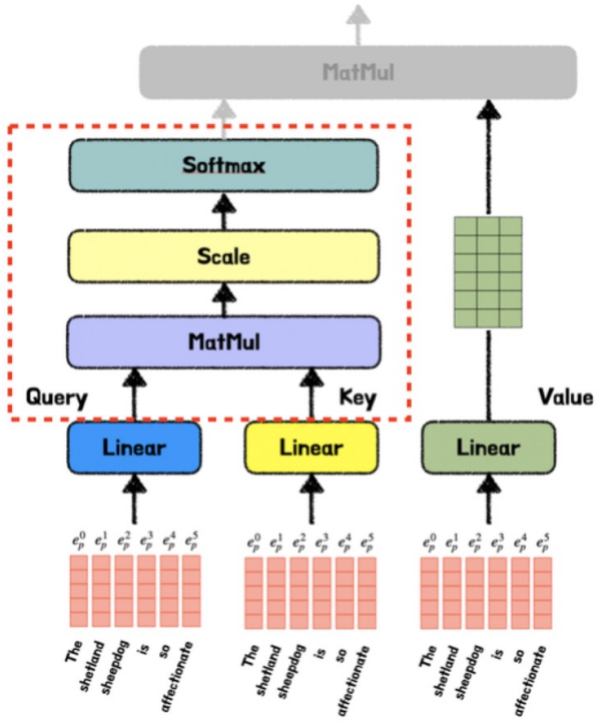






01. Introduction

Self attention



Attention Score

6 X 6

Softmax

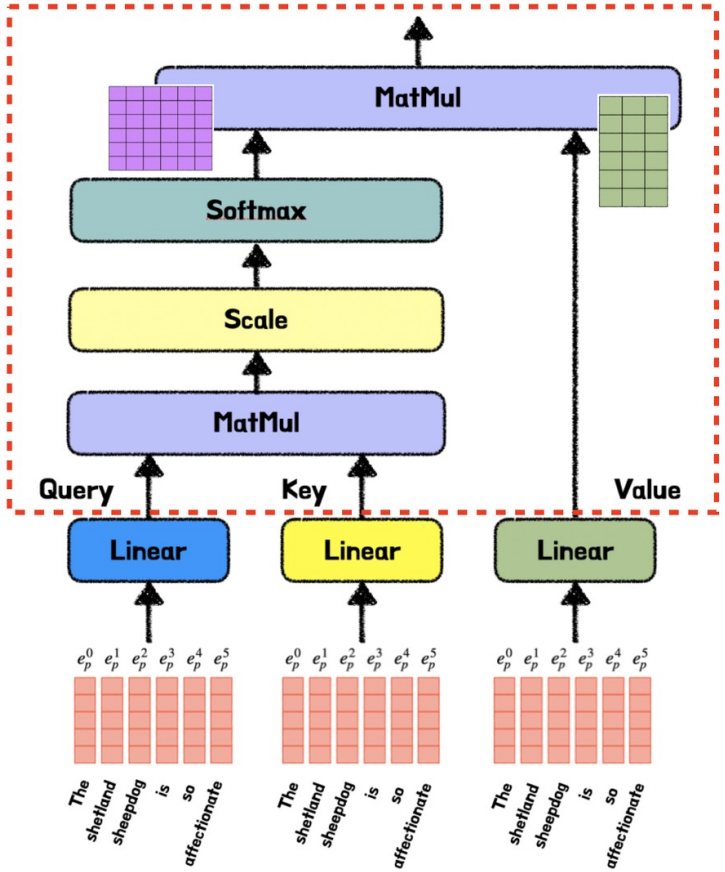
	The	shetland	sheepdog	is	so	affectionate
The	37.14	24.9	31.84	10.2	6.12	8.16
shetland	22.45	39.18	36.33	7.35	11.02	14.69
sheepdog	12.65	35.92	37.96	15.10	18.78	35.51
is	8.16	28.98	15.51	35.92	9.39	19.59
so	7.35	19.18	14.29	31.43	35.51	37.14
affectionate	21.22	15.92	27.25	28.57	36.33	39.59

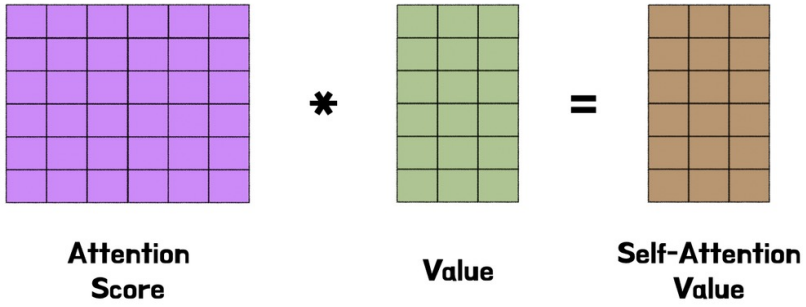
$$softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right)$$

Dimension of key vector

01. Introduction

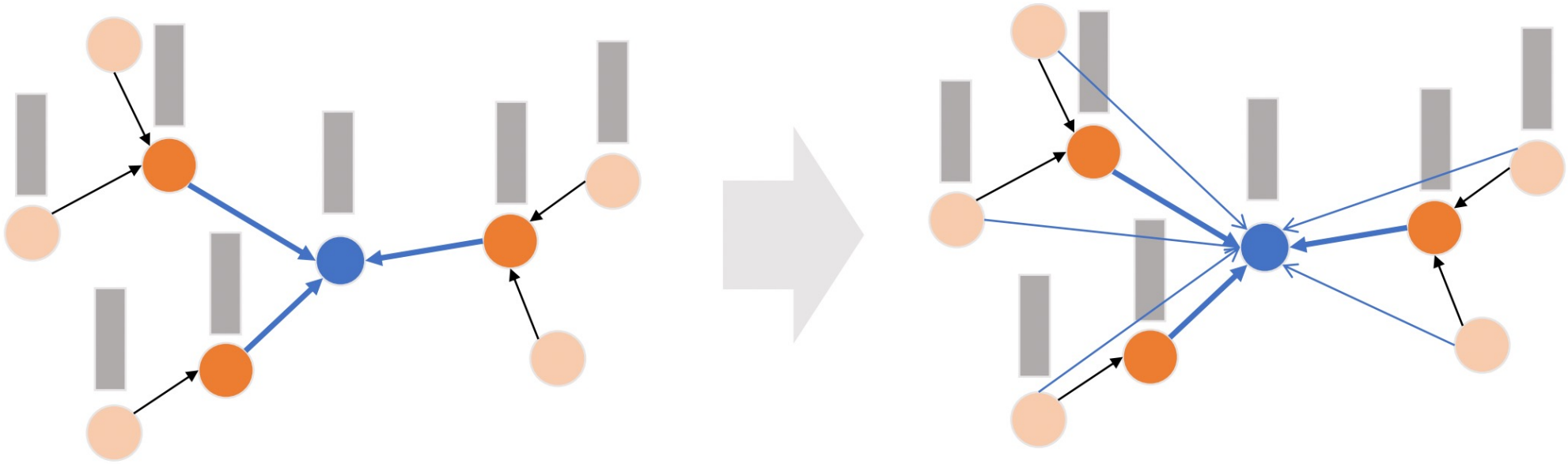
Self attention





01. Introduction

Graph Network Task

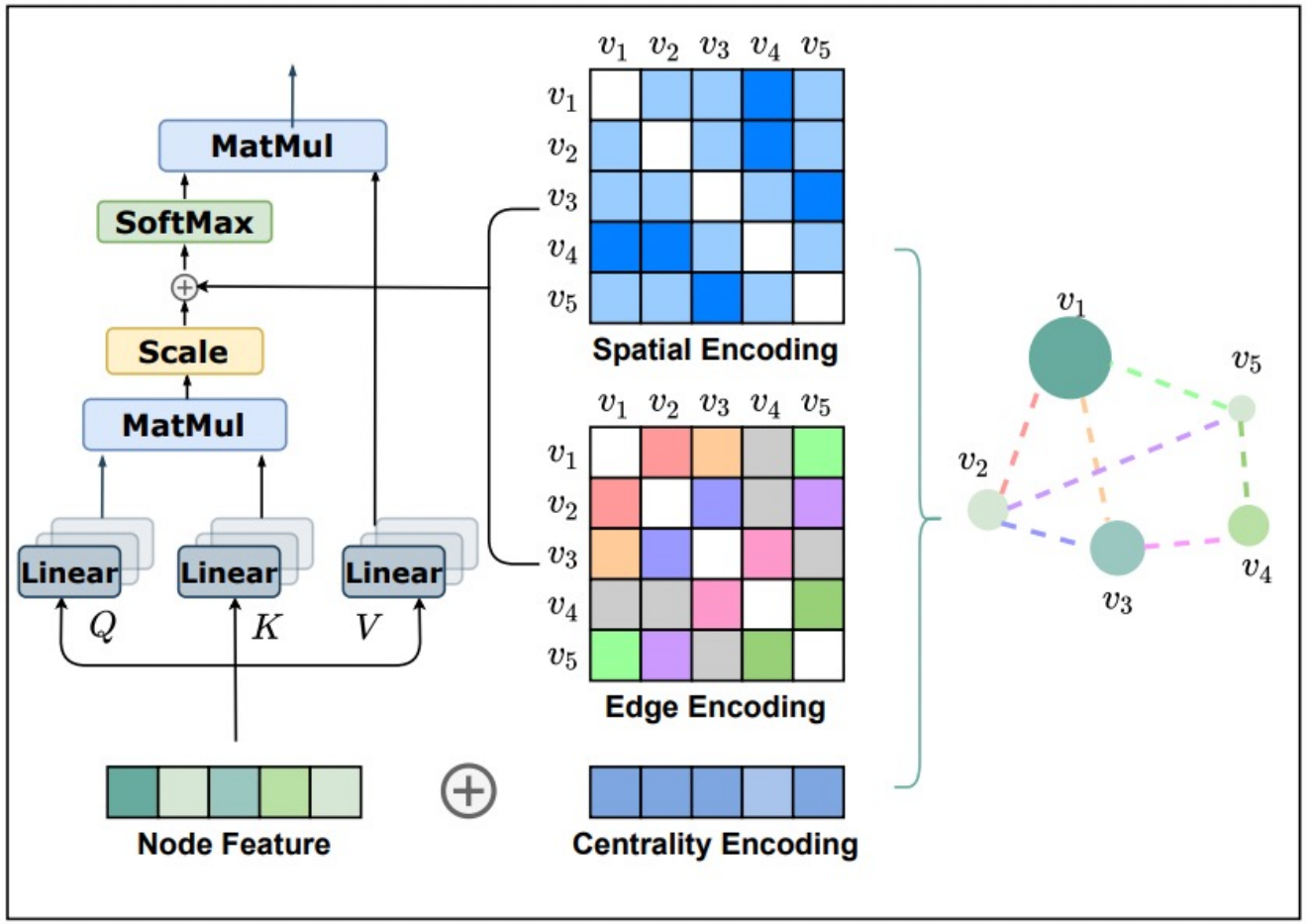


Local

global

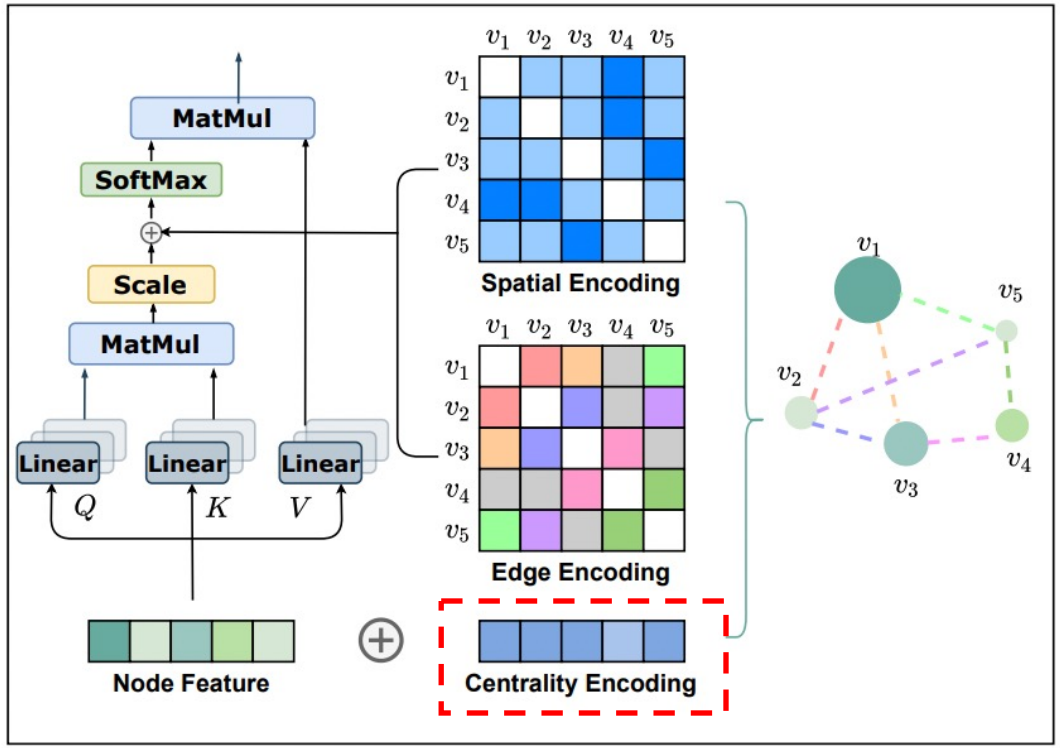
02. Method

02. Method


MatMul
SoftMax
Scale
MatMul
Linear
Linear
Linear
Q
K
V
Node Feature
Spatial Encoding
Edge Encoding
Centrality Encoding
v_1 v_2 v_3 v_4 v_5

02. Method

Centrality Encoding

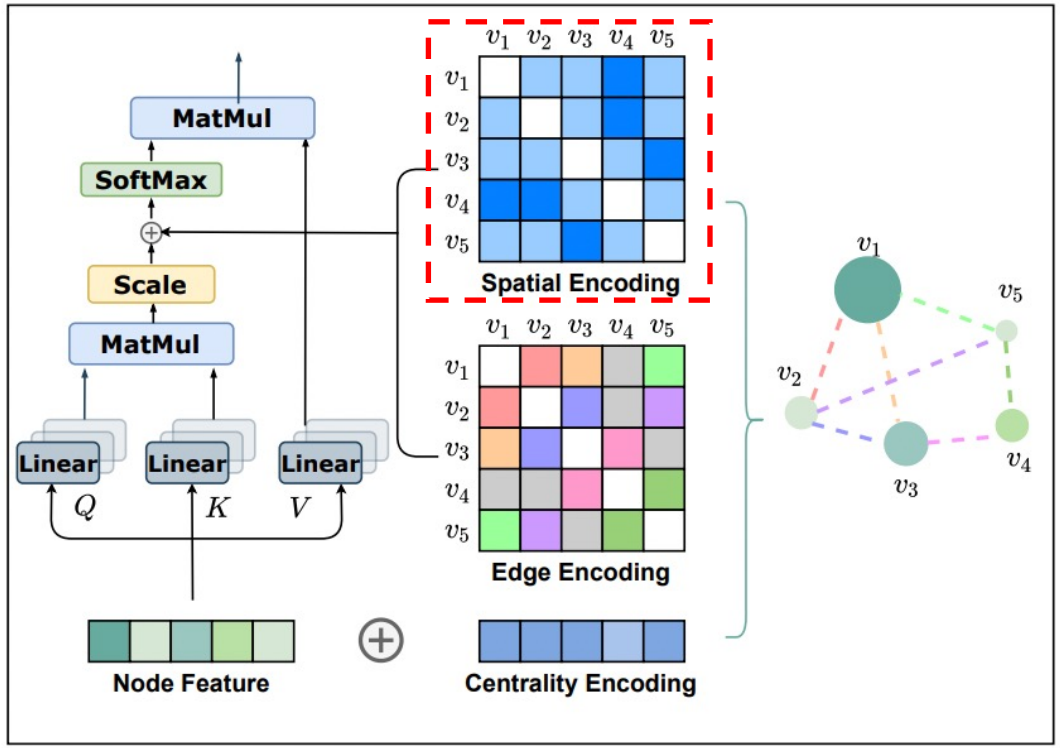


의미적 유사도와 node의 중요도를 고려한 Attention!

- ✓ Centrality : 그래프 내 중심성을 나타내는 지표
  - **degree**, betweenness, closeness, page rank, etc
- ✓ 그래프에는 허브 node가 존재함
- ✓ 기존의 Self-attention은 Centrality 정보를 충분히 담지 못함
- ✓ Centrality Encoding을 degree 값을 사용한 벡터로 나타낸다. (Learnable)
  - 만약 directed graph일 경우, in-degree emb와 out-degree emb 따로 사용하여, 두개의 합을 Centrality로 정의

02. Method

Spatial Encoding

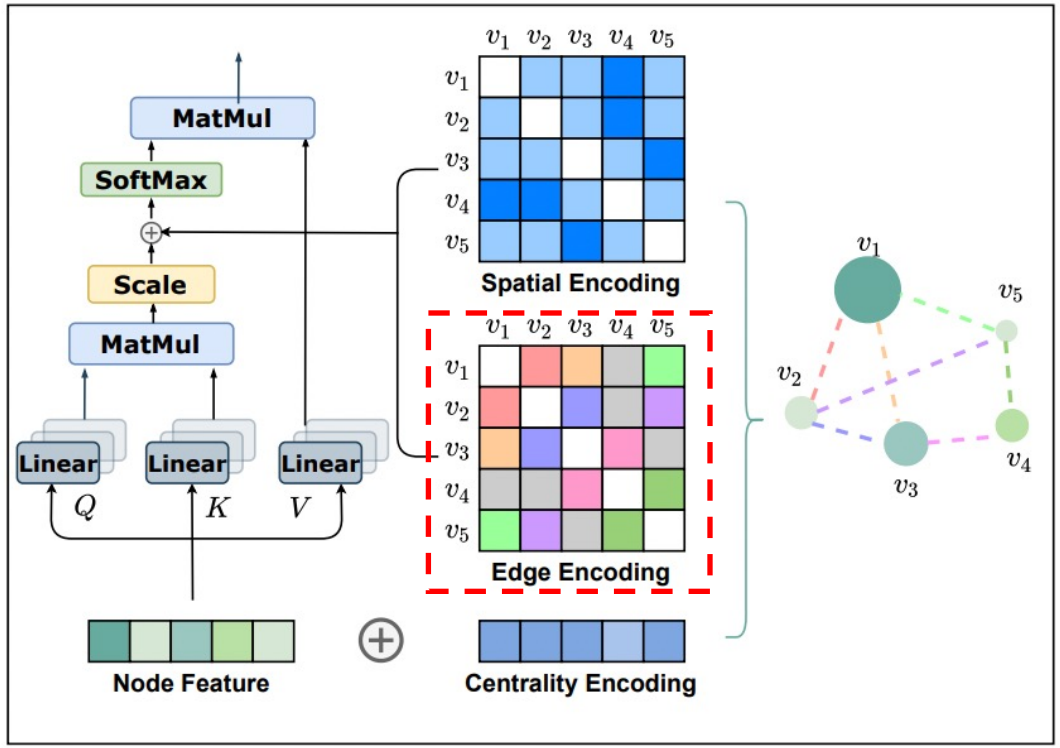


그래프의 구조적 정보 학습

- ✓ 고차원에 표현되는 node의 위치 정보를 나타내야 함
- ✓ Edge를 통해 연결된 정보를 기반으로, node간 거리를 측정 (Relative positional encoding)
- ✓ 두 node간 최단거리 계산하는 함수 정의 (Shortest Path)
  - 최단거리 구할때는 Floyd-Warshall 알고리즘 사용
  - 만약 두 node가 연결되어있지 않을 시에는 -1
- ✓ 위 함수에서 구한 값으로 인덱싱되는 scalar b를 두 node간 attention score의 bias 값으로 이용함 (Learnable)

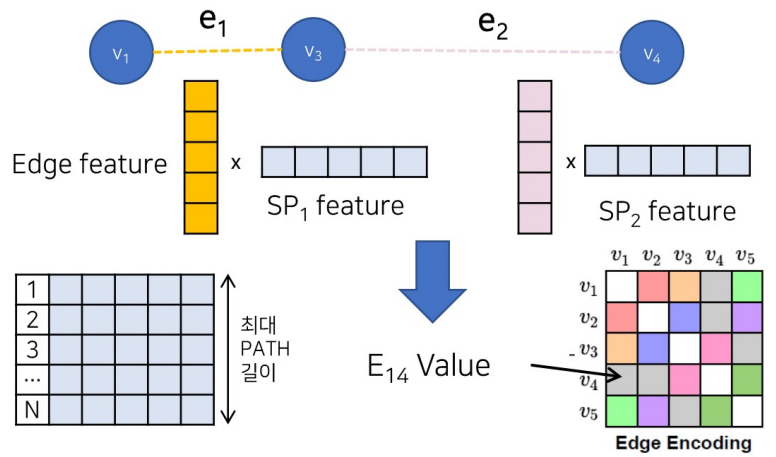
02. Method

Edge Encoding



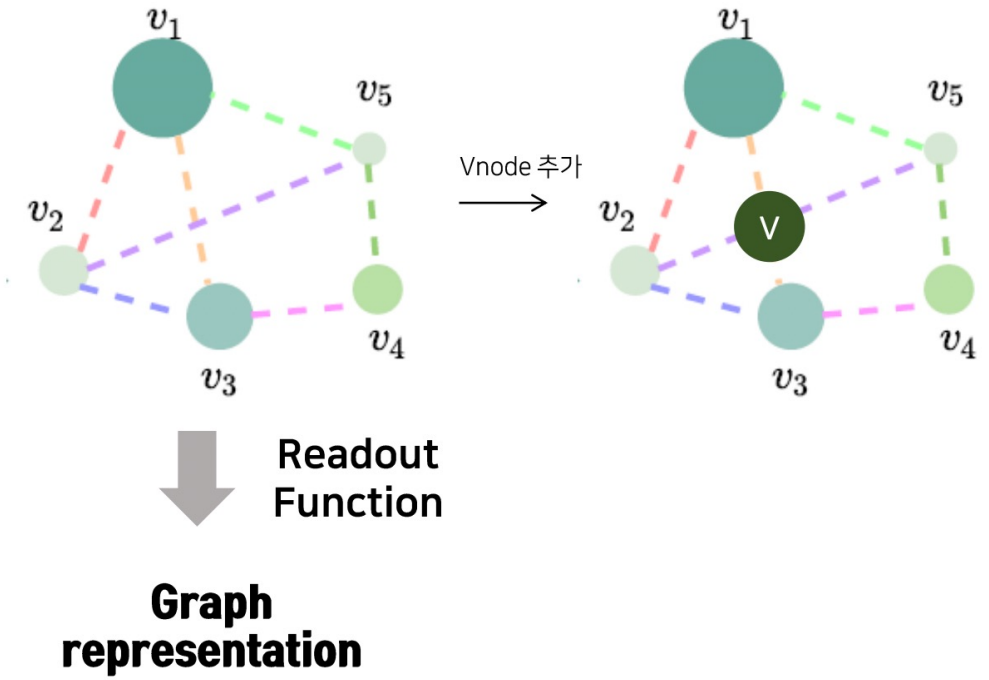
유의미한 정보를 가진 Edge Encoding

- ✓ Edge Embedding 값을 통해 node 간의 상관관계 표현
- ✓ 최단거리를 구성하는 Edge의 평균을 통해 두 node의 관계 표현



02. Method

Special Node



- ✓ 전체 node와 연결되는 가상의 node 생성
- ✓ Bert에서 [CLS] Token과 유사한 역할 수행
- ✓ [Vnode] 는 일반 node처럼 앞선 과정들을 똑같이 수행함 (단, 실존하는 허브 node와는 구별됨)
  -> Spatial Encoding에서 다른 값으로 처리함
- ✓ 학습 시, [Vnode] 값을 통해 Graph Classification을 진행
- ✓ [Vnode] 는 전체 그래프에 대한 정보를 담음
- ✓ self-attention 없이 [Vnode] 사용 시, Over Smoothing 문제 발생

03. Evaluation

03. Evaluation

- OGB Large-Scale Challenge
  - OGB-LSC Dataset : 화학결합 데이터셋 -> Graph regression Task
  - VN : [Vnode] 사용 여부 -> 기존 모델에 사용 시 성능 향상이 일어남

method	#param.	train MAE	validate MAE
GCN [26]	2.0M	0.1318	0.1691 (0.1684*)
GIN [50]	3.8M	0.1203	0.1537 (0.1536*)
GCN-VN [26, 15]	4.9M	0.1225	0.1485 (0.1510*)
GIN-VN [50, 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
$\text{Graphormer}_{\text{SMALL}}$	12.5M	0.0778	0.1264
Graphormer	47.1M	**0.0582**	**0.1234**

03. Evaluation

- Graph representation (Pre-train)
  - Pre-train ZINC Dataset -> OGB-LSC Dataset Inference
  - Transformer 기반 방법론이 해당 Task에서 비교적 Transfer 성능이 좋은 것을 알 수 있음

Table 4: Results on ZINC.

method	#param.	test MAE
GIN [50]	509,549	0.526±0.051
GraphSage [18]	505,341	0.398±0.002
GAT [47]	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
$\text{Graphormer}_{\text{SLIM}}$	489,321	**0.122**±0.006

03. Evaluation

- Ablation Study
  - 논문에서 제안된 방법론이 의미 있는 결과를 보였음을 알 수 있다.

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

04. Conclusion

04. Conclusion

- Graph Classification Task에서 Graph 를 표현하기 위해 Transformer 방법을 적용함
- Graph Representation Task를 수행할 때, 기존에 GNN 기반 방식에서 사용하던 Readout 과정을 생략
- Transformer에 그래프의 구조적 정보를 Encoding 하기 위해, 3가지 방법을 적용시킴

Thank You
감사합니다