
How to Find Your Friendly Neighborhood Graph

Attention Design With Self-Supervision

JungIn Kim

2022.04.10

Contents

- ❖ **Research Purpose**
- ❖ **Proposed Method**
 - **Graph Attention Network (GAT)**
 - **Self Supervised Graph Attention Network (SuperGAT)**
- ❖ **Experiments**
- ❖ **Conclusion**

Research Purpose

❖ How to Find Your Friendly Neighborhood Graph Attention Design With Self-Supervision

- 카이스트 대학교에서 연구하였고, 2022년 04월 10일 기준 55회 인용

HOW TO FIND YOUR FRIENDLY NEIGHBORHOOD: GRAPH ATTENTION DESIGN WITH SELF-SUPERVISION

Dongkwan Kim & Alice Oh

KAIST, Republic of Korea

`dongkwan.kim@kaist.ac.kr, alice.oh@kaist.edu`

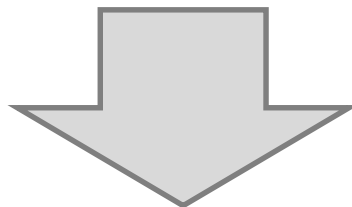
ABSTRACT

Attention mechanism in graph neural networks is designed to assign larger weights to important neighbor nodes for better representation. However, what graph attention learns is not understood well, particularly when graphs are noisy. In this paper, we propose a self-supervised graph attention network (SuperGAT), an improved graph attention model for noisy graphs. Specifically, we exploit two attention forms compatible with a self-supervised task to predict edges, whose presence and absence contain the inherent information about the importance of the relationships between nodes. By encoding edges, SuperGAT learns more expressive attention in distinguishing mislinked neighbors. We find two graph characteristics influence the effectiveness of attention forms and self-supervision: homophily and average degree. Thus, our recipe provides guidance on which attention design to use when those two graph characteristics are known. Our experiment on 17 real-world datasets demonstrates that our recipe generalizes across 15 datasets of them, and our models designed by recipe show improved performance over baselines.

Research Purpose

❖ How to Find Your Friendly Neighborhood Graph Attention Design With Self-Supervision

- GNN에서 Attention mechanism은 더 좋은 representation을 위해 사용되고 있음
 - ✓ 하지만, 그래프에 잡음(관련 없는 노드 간에 연결)이 있을 때 graph attention이 잘 학습되지 않음
- 해당 문제를 해결하기 위해 self-attention을 사용하는 Graph attention network(GAT) 등장
 - ✓ 그래프 attention이 실제로 배운 것이 무엇인지에 대한 이해가 부족
 - ✓ 여러 데이터 셋 전반에 걸쳐 GNN에 비해 성능 향상의 정도가 불일치



❖ Self Supervised Graph Attention Network (SuperGAT)

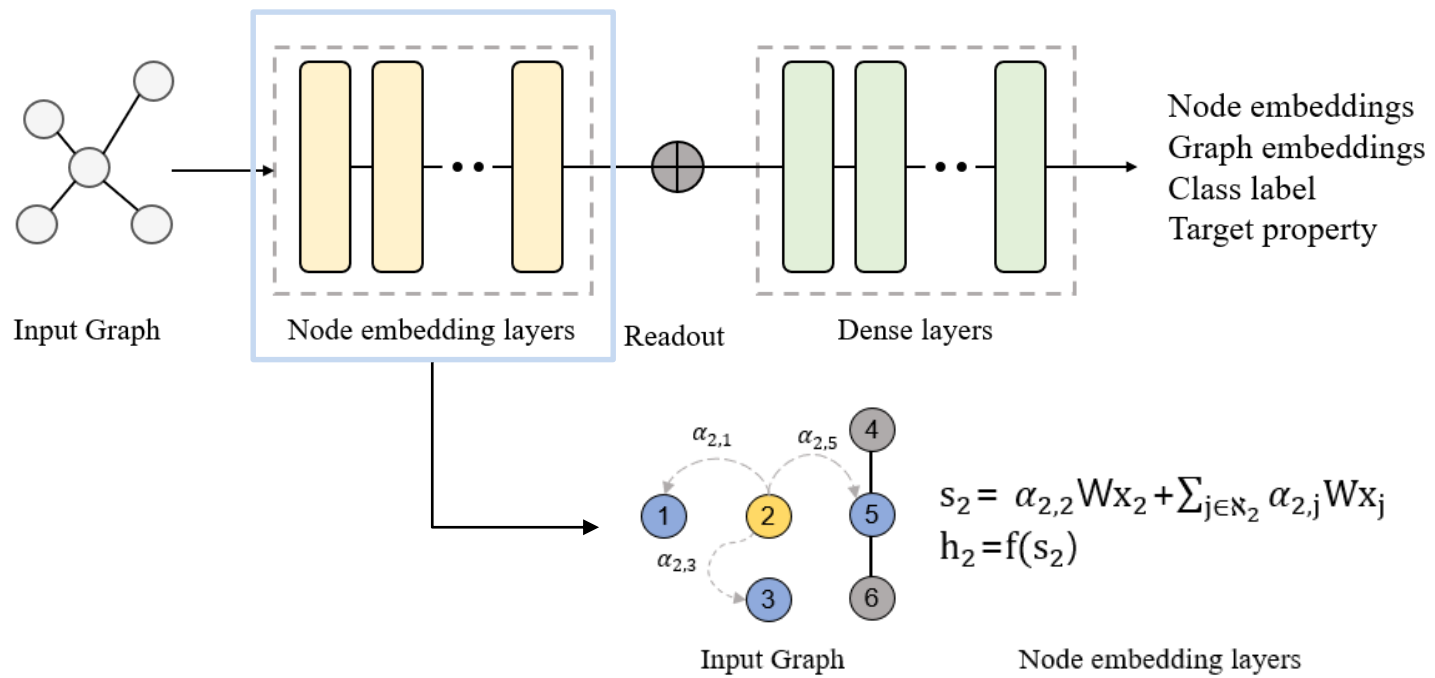
- GAT의 문제점을 개선하기 위해 self-supervised attention을 사용
 - ✓ 그래프의 중심 노드를 표현하기 위해 주변 노드의 중요성의 정도를 학습하고 평가하는데 사용
 - ✓ 노드 간에 edge의 존재 가능성을 예측하기 위해 attention value를 입력으로 사용하여 self-supervised task 수행

Proposed Method

Background

❖ Graph Attention Network (GAT) - Node embedding layer

- self-attention mechanism을 노드 embedding 과정에 적용한 인공신경망
- 입력 그래프에 대해 노드 embedding layer에서 각 노드에 대한 embedding을 생성



Proposed Method

Background

W : 학습 가능한 가중치 행렬

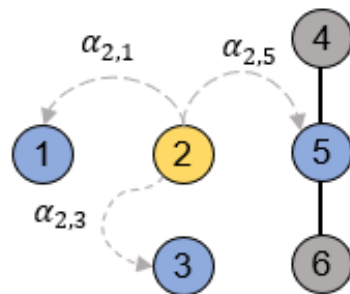
X_i : i 번 노드의 feature

$\alpha_{i,j}$: i 번 노드에서 j 번째 노드로의 attention 값

\mathcal{N}_i : i 번 노드와 edge로 연결된 인접 노드들의 index set

❖ Graph Attention Network (GAT) - Node embedding layer

- 2번 노드에 대한 attention 기반 node embedding s_2 생성
- 2번 노드와 인접한 1,3,5 노드에 대한 attention score ($\alpha_{2,1}, \alpha_{2,5}, \alpha_{2,3}$) 계산
- s_2 생성하기 위해 사전에 구한 attention score를 기반으로 가중합 수행
- 연산 과정을 통해 얻은 s_2 에 활성화 함수 f 를 적용하여 최종 node embedding(h_2) 생성



Input Graph

$$s_2 = \alpha_{2,2} W X_2 + \sum_{j \in \mathcal{N}_2} \alpha_{2,j} W X_j$$
$$h_2 = f(s_2)$$

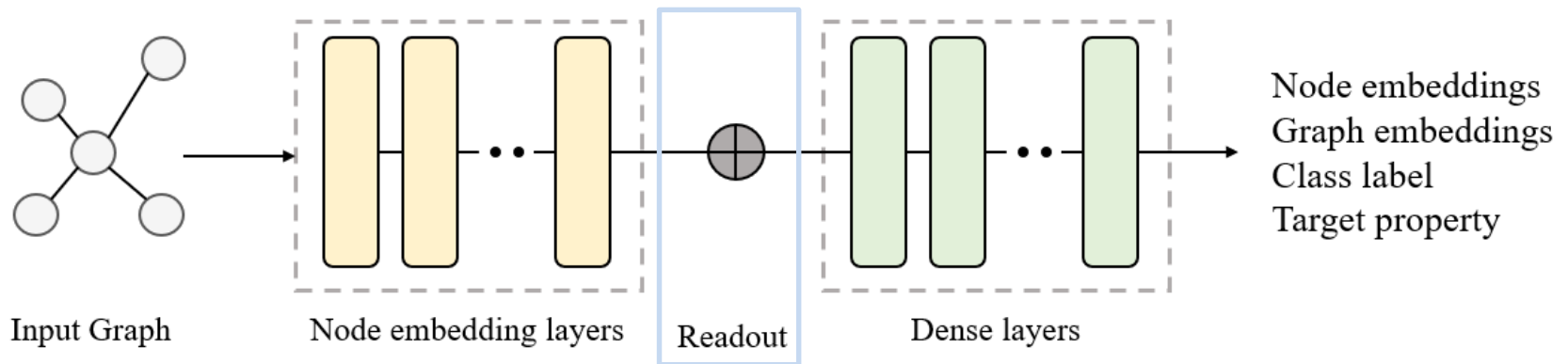
Node embedding layers

Proposed Method

Background

❖ Graph Attention Network (GAT) – Readout

- self-attention mechanism을 노드 embedding 과정에 적용한 인공신경망
- 그래프 단위의 출력을 만드는 경우(그래프 embedding)에만 존재하는 계층
 - ✓ 전체 노드에 대한 노드 embedding 값을 모두 더하거나 평균을 내는 방식을 사용

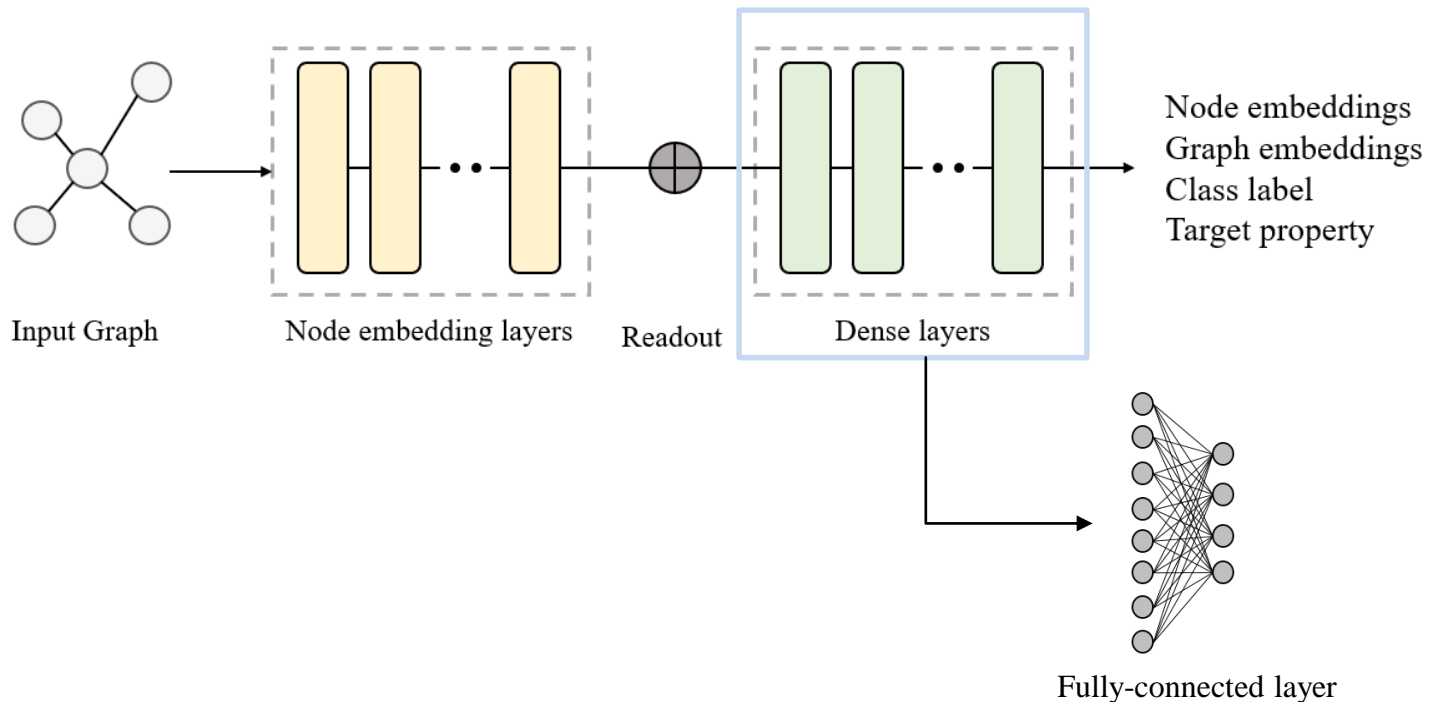


Proposed Method

Background

❖ Graph Attention Network (GAT) – Dense layer

- self-attention mechanism을 노드 embedding 과정에 적용한 인공지능망
- Fully-connected layer로 구현
 - ✓ 입력된 노드 embedding 행렬 또는 그래프 embedding 벡터에 대한 출력을 계산



Proposed Method

❖ Self Supervised Graph Attention Network(SuperGAT)

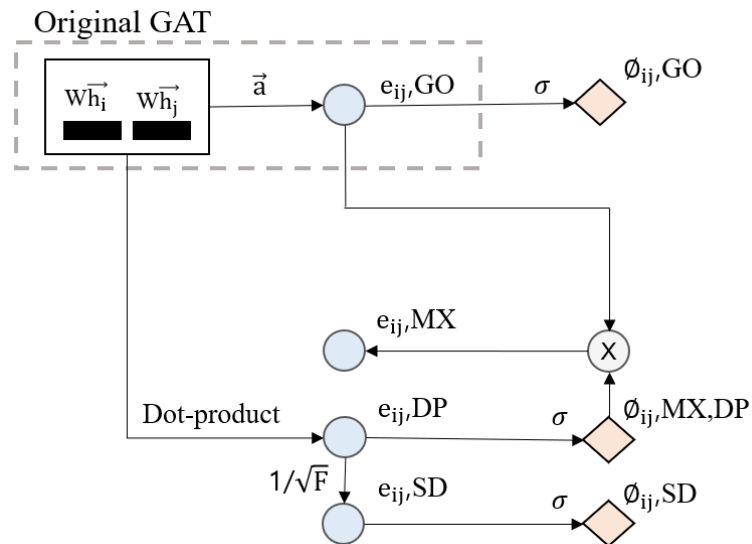
- edge의 존재 유무가 노드 간에 관계의 중요성에 대한 고유의 정보를 갖고 있다는 특성 고려
- attention value을 입력으로 받아 edge의 존재를 예측하는 self-supervised task를 수행
 - ✓ 기존의 link prediction task와 비슷함
 - ✓ 노드 i 와 j 사이에 edge가 존재하면 1, 존재하지 않으면 0으로 라벨
 - ✓ attention score를 입력으로 받아 self supervised task 진행
 - ✓ 해당 task를 수행함으로써 연결되어 있지 않은 이웃 노드를 구별하는데 더 효과적인 attention 학습 가능

Proposed Method

$e_{i,j}$: 활성화함수에 의해 정규화되지 않은 attention value
 $\phi_{i,j}$: 노드 i와 j 사이에 edge가 존재할 확률

❖ Self Supervised Graph Attention Network(SuperGAT)

- Original GAT(GO)와 Dot Product(DP) attention을 기반으로 한 4가지 attention을 사용
 - ✓ GO : Original GAT attention
 - ✓ DP : Dot Product attention
 - ✓ SD : Scaled Dot product attention
 - ✓ MX : Mixed GO and DP attention



$$e_{ij} \rightarrow \text{softmax}_j \rightarrow \alpha_{ij}$$

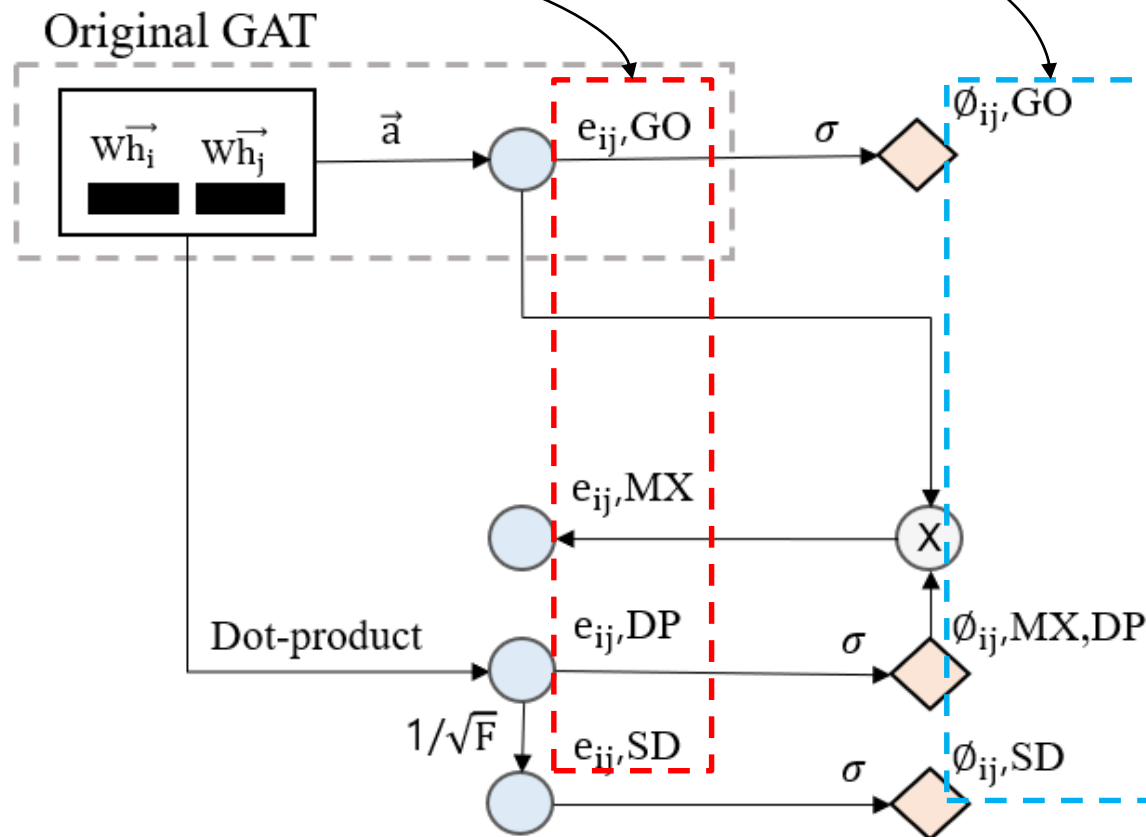
$$\phi_{ij} = P((j,i) \in E)$$

SuperGAT의 attention mechanism

Proposed Method

❖ Self Supervised Graph Attention Network(SuperGAT)

- attention score를 입력으로 받아 노드 간에 edge가 존재할 확률을 예측 → self supervised task



Experiment

DP vs GO

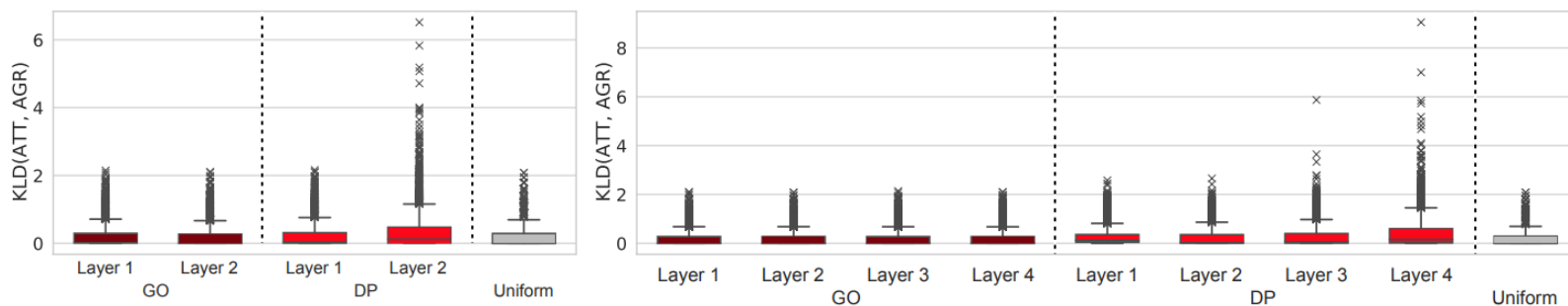
❖ Experiment 1 – Does graph attention learn label-agreement?

- 어떤 attention(GO,DP)이 두 노드 간에 라벨을 잘 구분할 수 있는지에 대한 실험
- 정규화 된 attention score(α_k), label agreement distribution(ℓ_k) \rightarrow KL divergence
- 중심 노드와 그 이웃 노드 간에 label-agreement를 잘 capture했다면 $KLD(\alpha_k, \ell_k)$ 작아짐
- GO attention⁰ | DP attention보다 label agreement를 더 잘 학습함

$$\alpha_k = [\alpha_{kk}, \alpha_{k1}, \dots, \alpha_{kJ}]$$

$$\ell_k = [\ell_{kk}, \ell_{k1}, \dots, \ell_{kJ}]$$

$$KLD(\alpha_k, \ell_k) = \sum_{j \in \mathbb{N}_k \cup \{k\}} \alpha_{kj} \log(\alpha_{kj} / \ell_{kj})$$

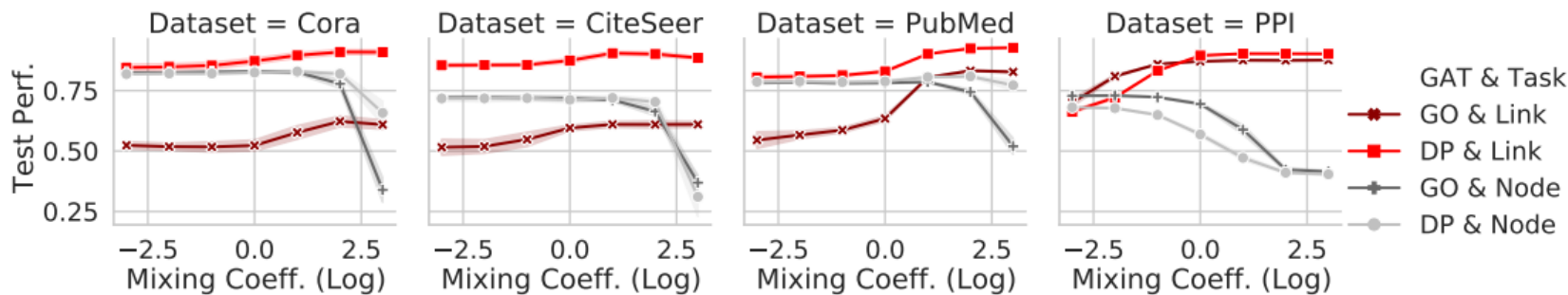


Experiment

DP vs GO

❖ Experiment 2 – Is graph attention predictive for edge presence?

- DP와 GO 중 어떤 attention 기법이 edge의 정보를 더 잘 표현하는지에 대한 실험
 - ✓ Edge의 존재 유무를 예측하면서 학습된 attention value의 가중치가 적합하게 학습되었는지 확인
 - ✓ Link prediction & Node classification 성능 비교 진행
- DP attention⁰ | DP attention보다 edge의 존재 유무를 더 잘 예측

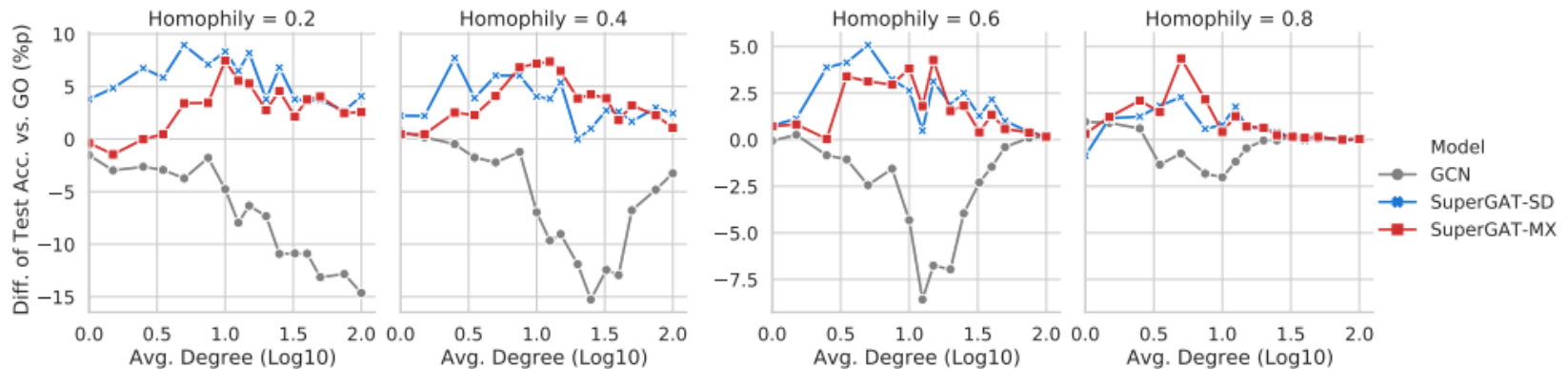


Experiment

DP vs GO

❖ Experiment 3 – Which graph attention should we use for given graphs?

- 그래프의 성질 중 homophily와 Average degree에 따라 node classification의 성능이 달라지는지 실험
 - ✓ Homophily: 유사한 노드가 다른 노드보다 서로 연결될 가능성이 높은 성질
 - ✓ Degree: 노드에 연결된 edge의 수
 - ✓ Average degree: 모든 노드의 degree의 합을 총 노드 수로 나눈 값
- GCN, SuperGAT-SD, SuperGAT-MX 등 여러 모델을 사용하여 accuracy 비교
- 실험 결과, homophily와 average degree에 따라 성능이 달라짐
 - ✓ 두 성질에 따라 모델 선정 및 attention 선정이 필요함



Conclusion

❖ Conclusion

- GNN에서 attention mechanism의 도입으로 노드 간에 더 좋은 representation을 학습하려함
- 하지만, 그래프 자체가 noisy하다면 graph attention이 학습한 것이 잘 이해가 되지 않음
- 본 논문에서는 이러한 문제를 개선하기 위해 self supervised graph attention network (SuperGAT) 제안
- 실험을 통해 DP와 GO 각각의 기법을 이용해 label-agreement와 edge 존재 유무 예측을 동시에 하는 것이 어렵다는 것을 확인
 - ✓ 두 가지 task에 균형을 맞춘 다양한 graph attention 기법을 제안해야 한다고 생각됨
 - ✓ 또한, 입력 그래프의 average degree와 homophily를 조절하면서 attention 기법을 제안해야 한다고 생각됨

Thank You