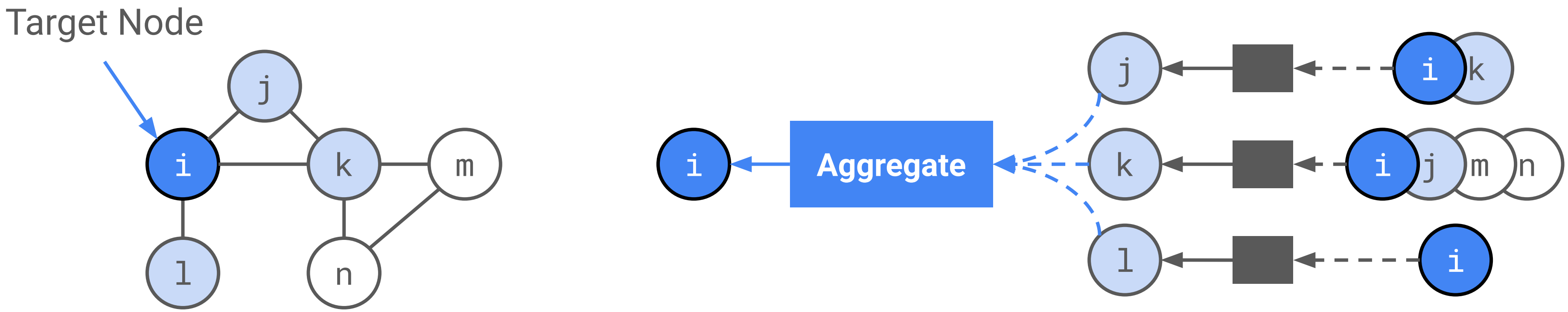
How to Find Your Friendly Neighborhood: Graph Attention Design with Self-Supervision

Dongkwan Kim and Alice Oh
KAIST

Learning on Graphs and Geometry Reading Group
23rd November 2021

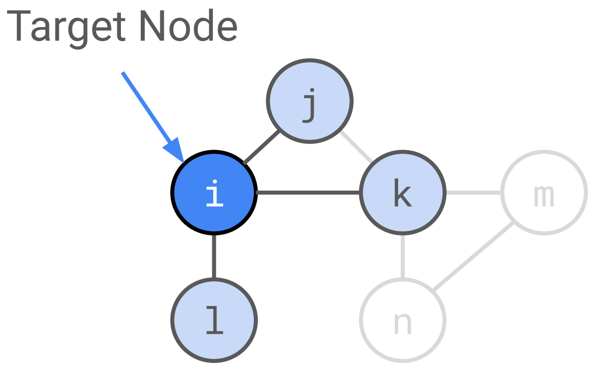
Preliminary: Graph Neural Networks

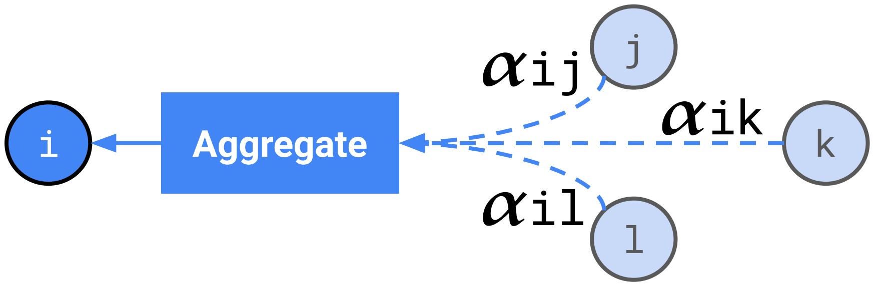


To generate an embedding for node `i`, the GNN aggregates embeddings of `i`'s local neighborhoods (i.e., `j`, `k`, and `l`)

Inspired by Hamilton et al. "Representation Learning on Graphs: Methods and Applications", IEEE Data Engineering Bulletin. 2017.

Preliminary: Graph Attention Networks

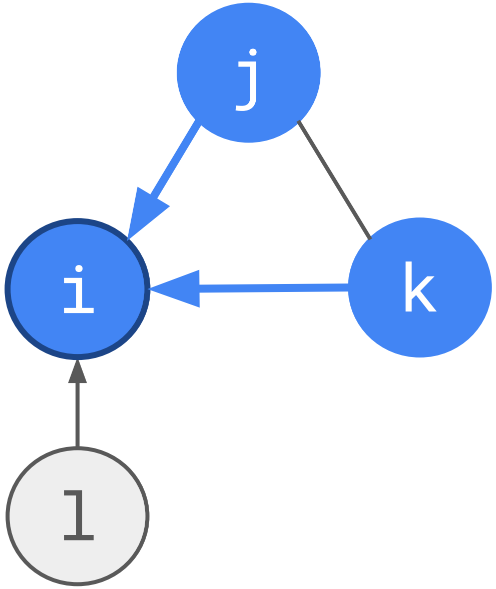




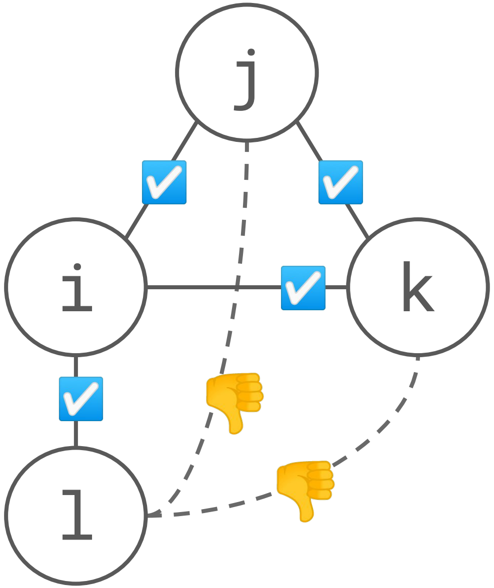
$$\alpha_{ij} = \mathrm{softmax}_{\mathcal{N}_i}\left(f(\vec{h}_i, \vec{h}_j)\right)$$

Graph Attention Networks (GATs) implicitly assign different importances (by attention) to neighbors in aggregating them

Attention over edges in graph attention networks learns the relational importance between nodes



Presence & absence of edges explicitly represent information about the importance of relations



Self-Supervision

Presence & absence of edges explicitly represent information about the importance of relations

Attention over edges in graph attention networks learns the relational importance between nodes

How nodes make friends with each other

How to find the node's friendly neighborhoods

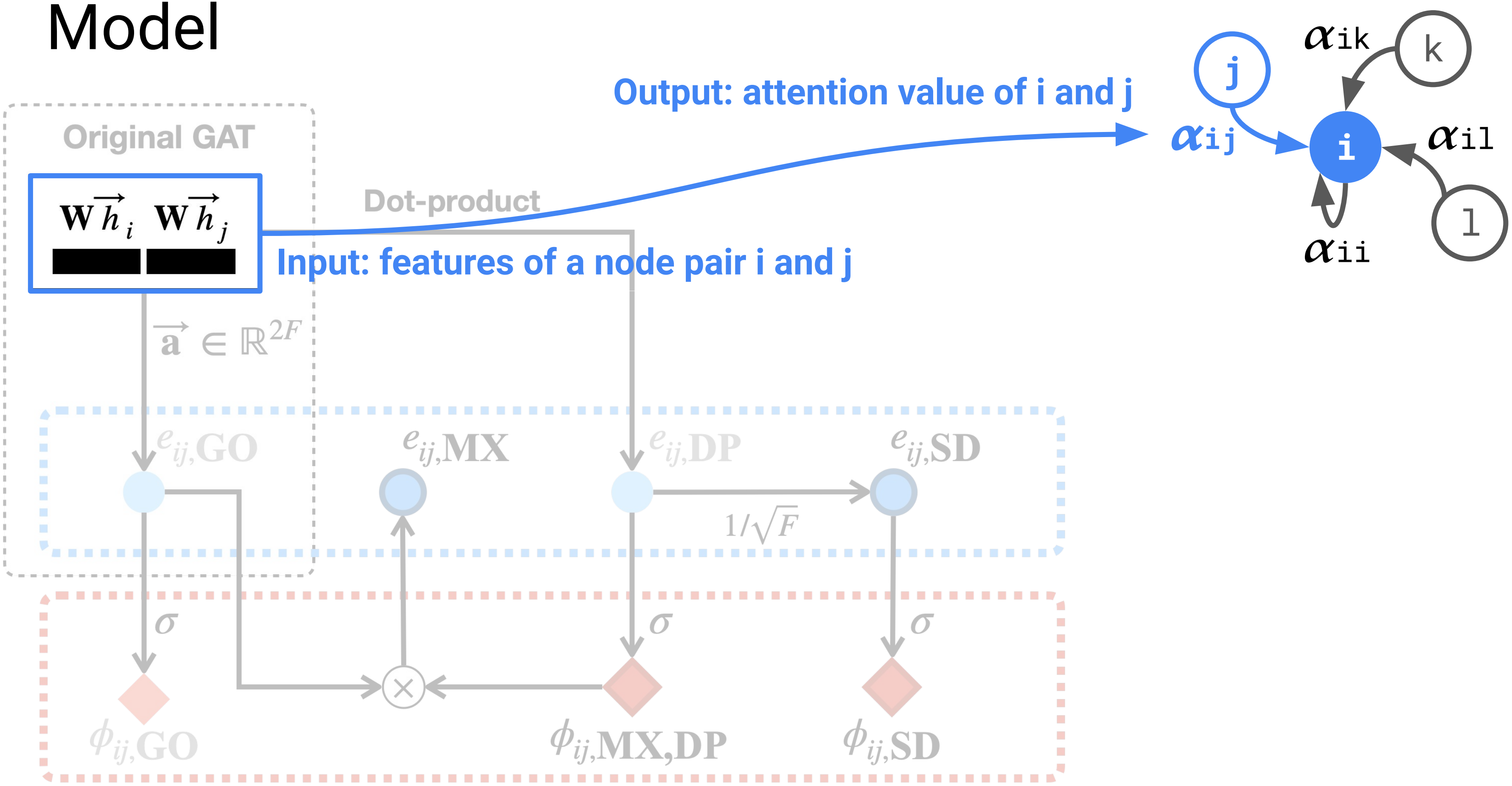
Contribution

1. Present models with self-supervised graph attention using edge information: SuperGAT

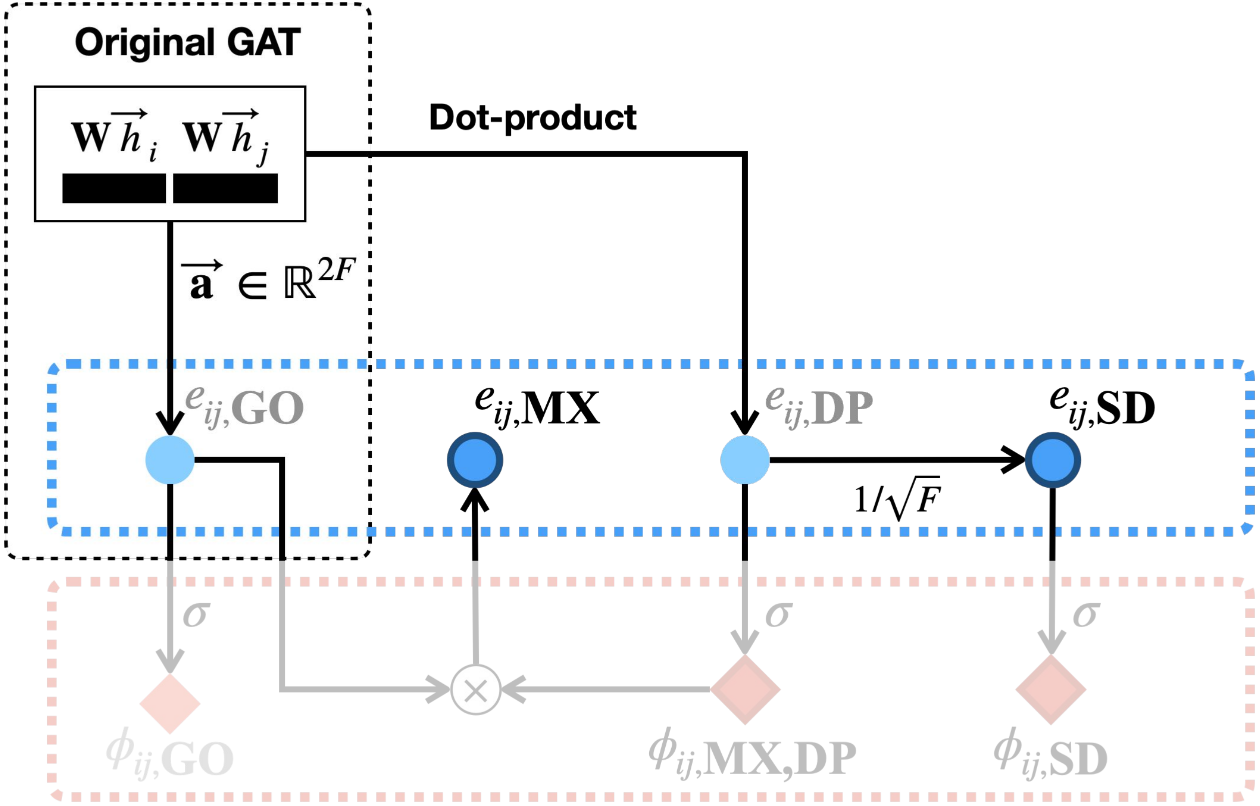
2. Analyze GAT's original (GO) and Dot-product (DP) attention: GO is better than DP in label-agreement, but DP is better than GO in link prediction

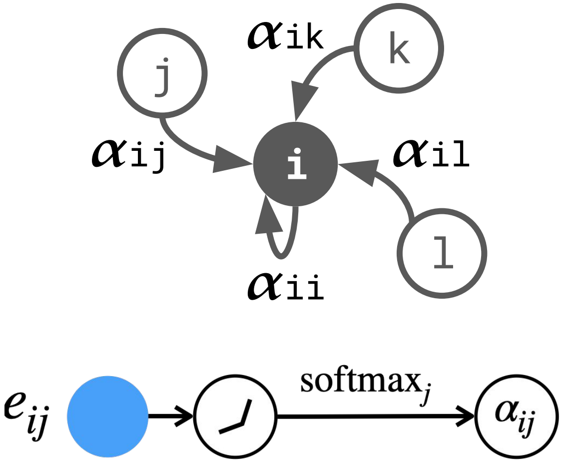
3. Propose recipes to design graph attention concerning homophily and average degree and confirm its validity

Model


Output: attention value of i and j
Input: features of a node pair i and j
Original GAT
$\mathbf{W}\vec{h}_i$ $\mathbf{W}\vec{h}_j$
Dot-product
$\vec{\mathbf{a}} \in \mathbb{R}^{2F}$
$e_{ij,\mathbf{GO}}$
$e_{ij,\mathbf{MX}}$
$e_{ij,\mathbf{DP}}$
$e_{ij,\mathbf{SD}}$
$1/\sqrt{F}$
σ
$\phi_{ij,\mathbf{GO}}$
$\phi_{ij,\mathbf{MX,DP}}$
$\phi_{ij,\mathbf{SD}}$
j
k
i
l
α_{ij}
α_{ik}
α_{il}
α_{ii}

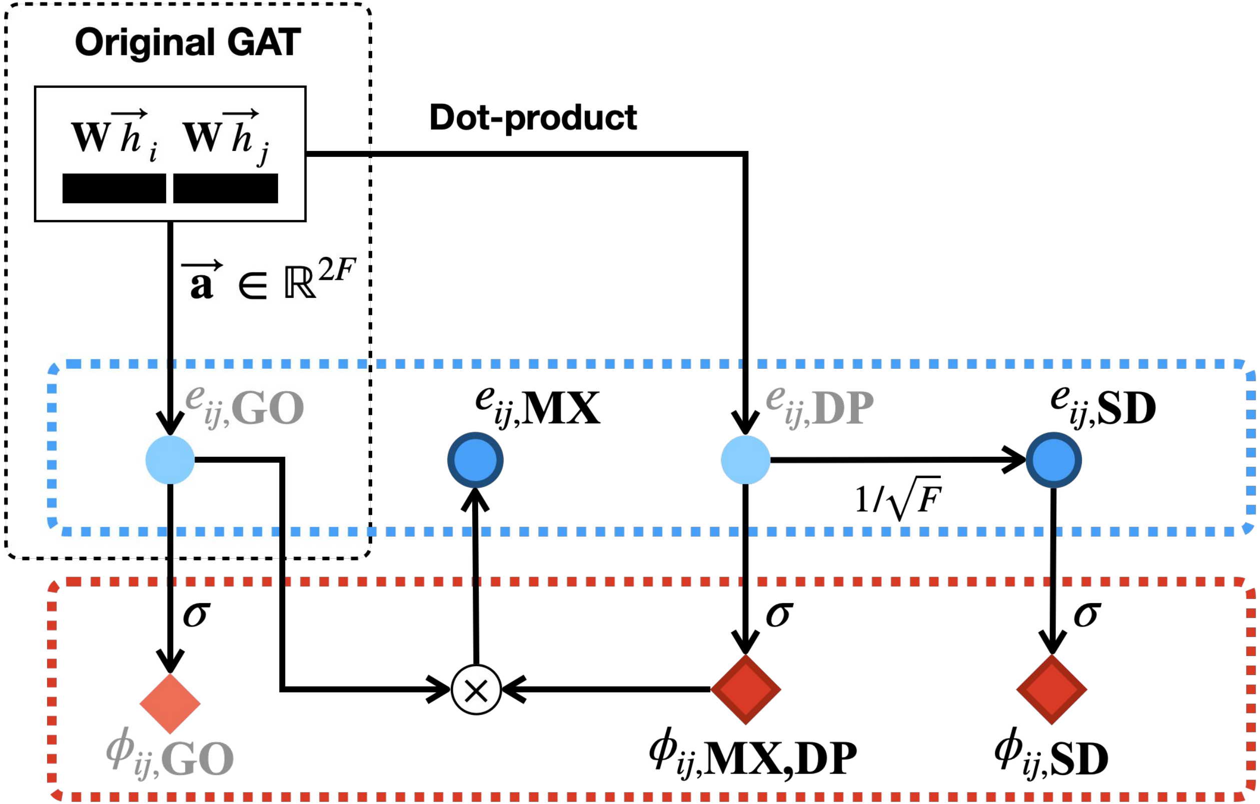
Model





Unnormalized attention before softmax

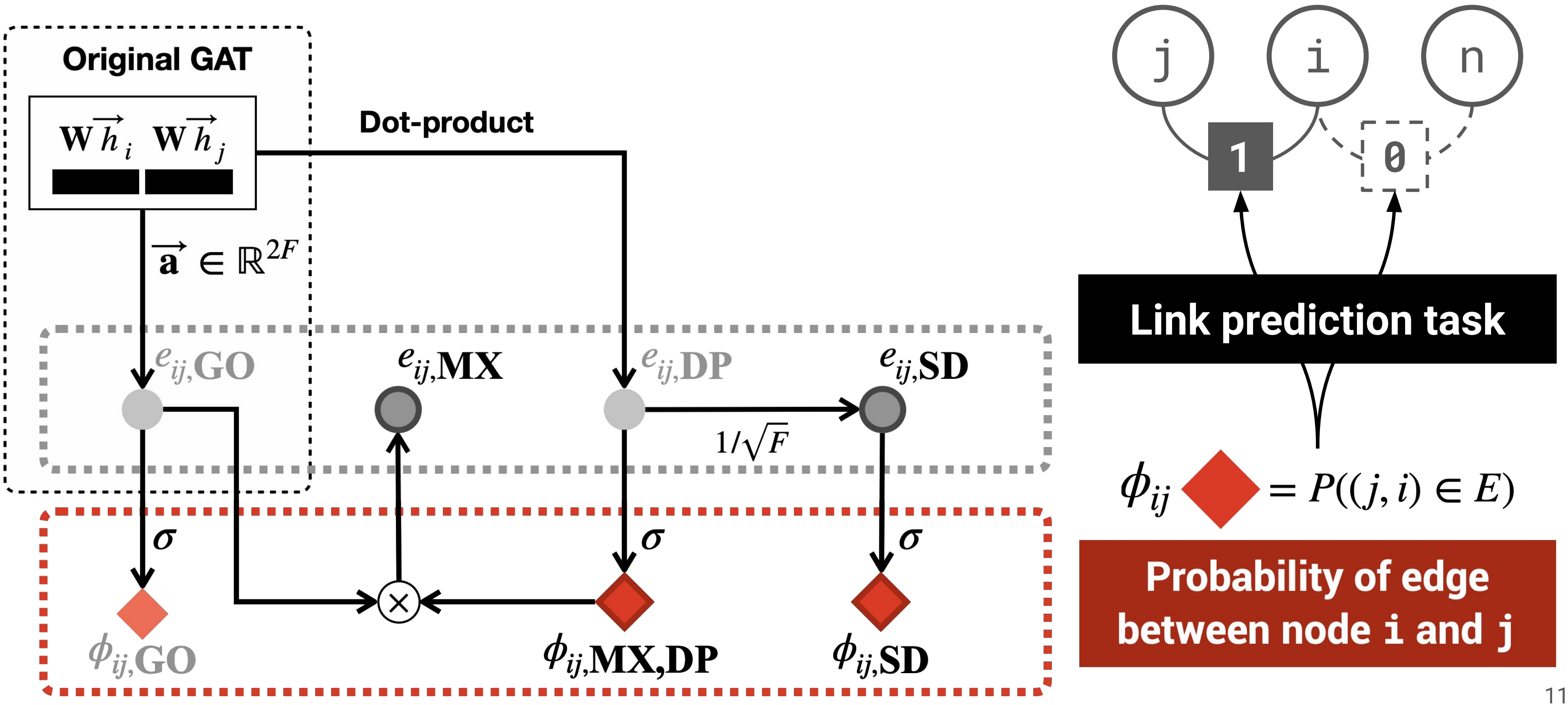
Model



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

Probability of edge between node i and j

Proposed Self-Supervised Task



Proposed Self-Supervised Task

Training Loss: $\mathcal{L}_V + \lambda_E \cdot \sum_{l=1}^{L} \mathcal{L}_E^l$

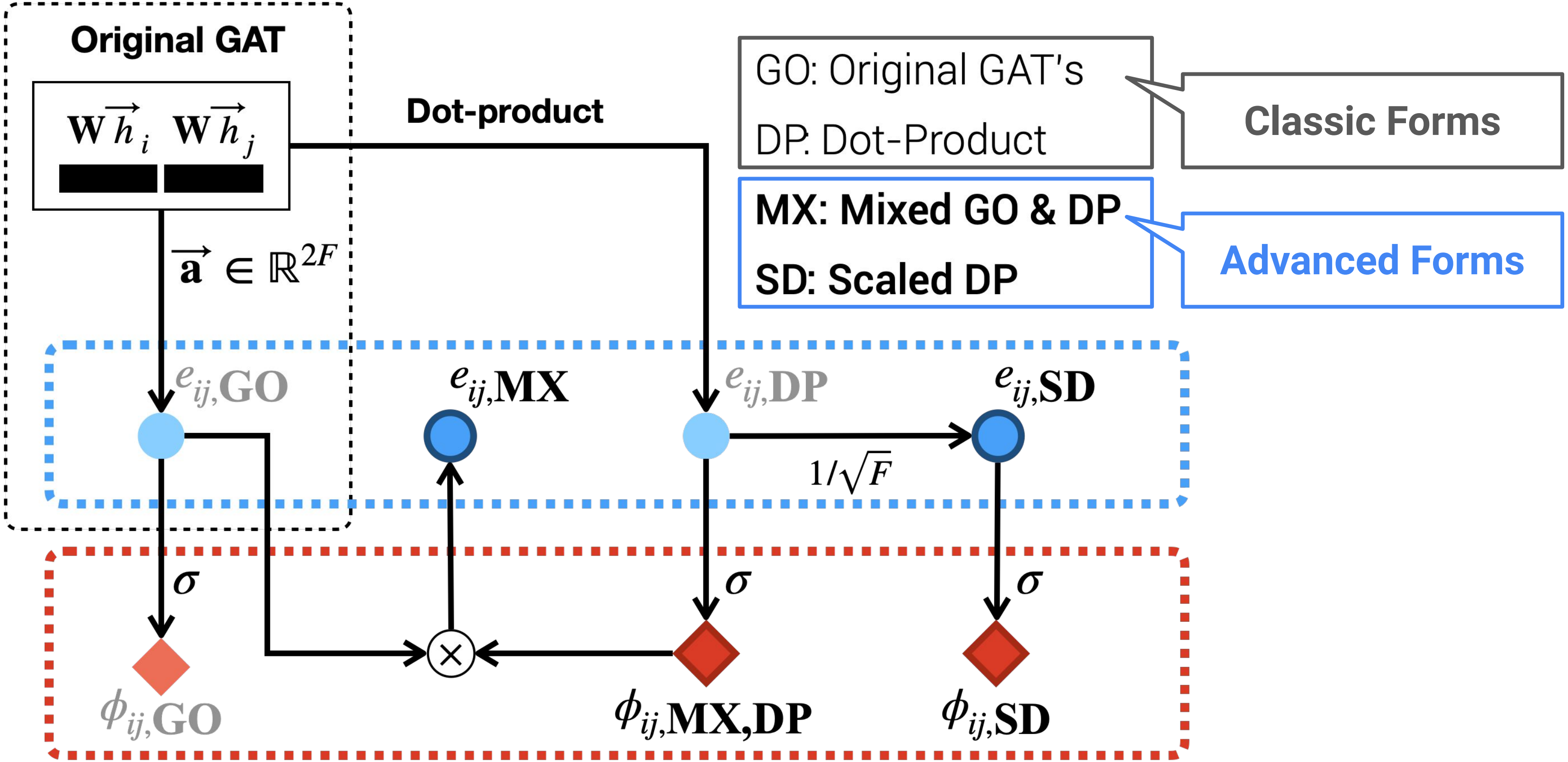
$$\mathcal{L}_V = \text{CrossEntropy}(\forall i : \overrightarrow{h}_i^L, \text{label}_i),$$

$$\mathcal{L}_E^l = -\sum_{(j,i)\in E\cup E^-} \mathbf{1}_{(j,i)=0} \cdot \log(1-\phi_{ij}) + \mathbf{1}_{(j,i)=1} \cdot \log \phi_{ij},$$

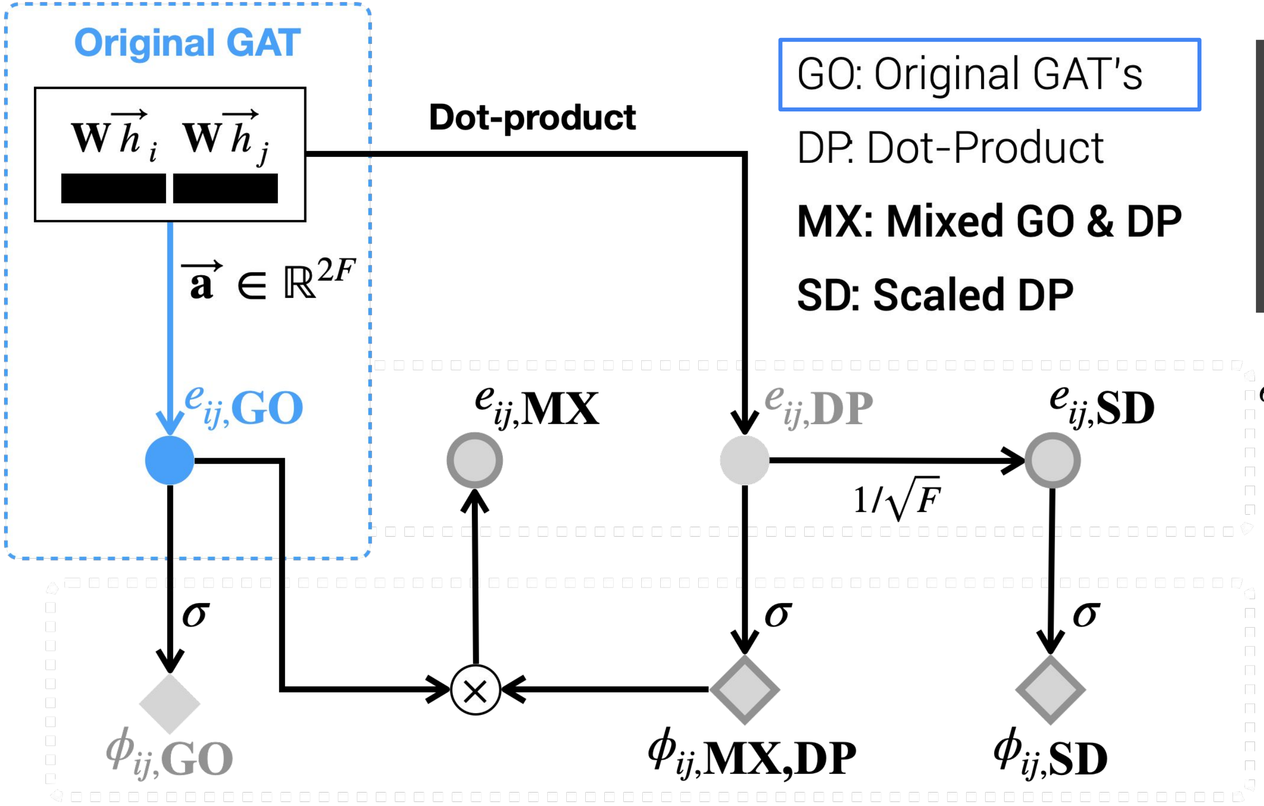
where E^- are negative samples drawn from $(V \times V)\backslash E$

Our proposed self-supervised task is the link prediction with attention, and can be optimized with the binary cross-entropy on edge labels

Model


Original GAT
$\mathbf{W}\vec{h}_i$ $\mathbf{W}\vec{h}_j$
$\vec{\mathbf{a}} \in \mathbb{R}^{2F}$
Dot-product
GO: Original GAT's
DP: Dot-Product
Classic Forms
MX: Mixed GO & DP
SD: Scaled DP
Advanced Forms
$e_{ij,\mathbf{GO}}$
$e_{ij,\mathbf{MX}}$
$e_{ij,\mathbf{DP}}$
$e_{ij,\mathbf{SD}}$
$1/\sqrt{F}$
σ
σ
σ
$\phi_{ij,\mathbf{GO}}$
$\phi_{ij,\mathbf{MX,DP}}$
$\phi_{ij,\mathbf{SD}}$

Model

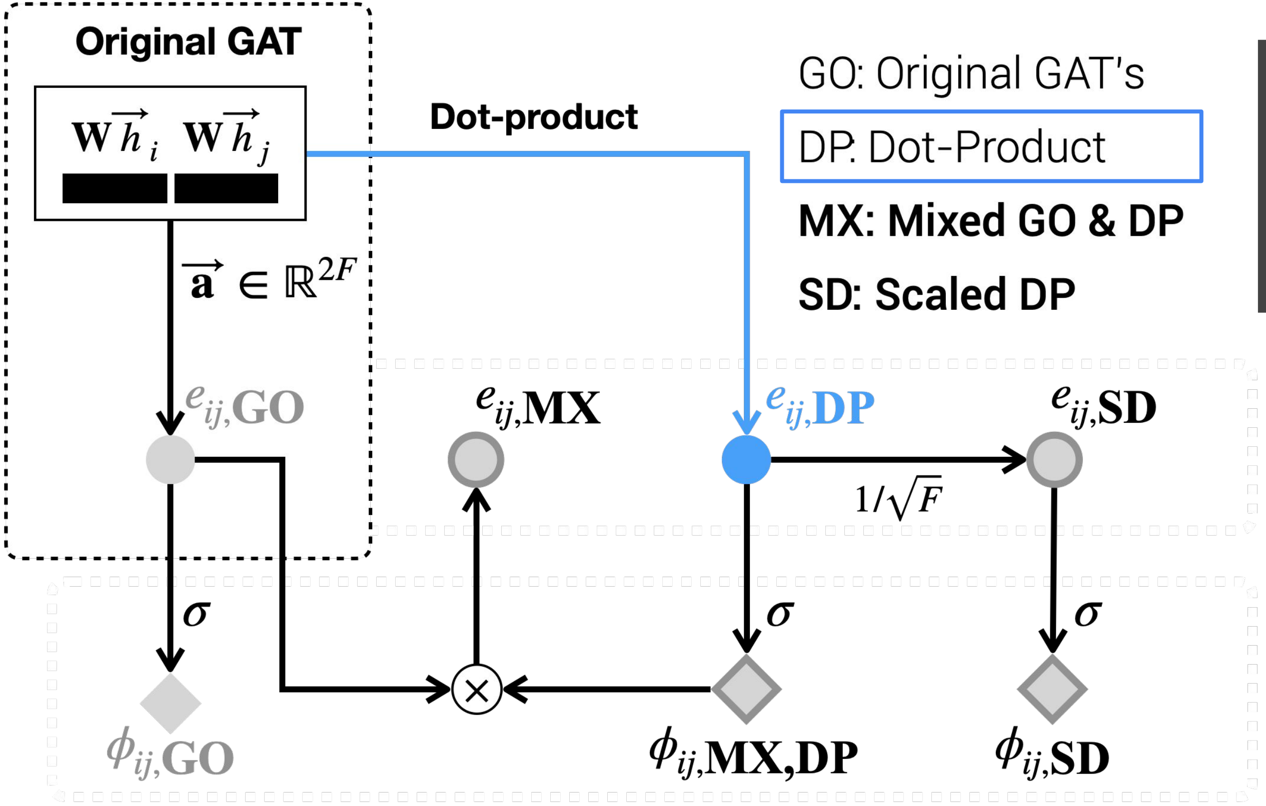


Veličković et al., ICLR 2018

The original GAT (GO) computes the coefficients by a single-layer feed-forward network

$$e_{ij,\mathrm{GO}}^{l+1} = (\boldsymbol{a}^{l+1})^\top \left[\boldsymbol{W}^{l+1}\boldsymbol{h}_i^l \| \boldsymbol{W}^{l+1}\boldsymbol{h}_j^l\right]$$

Model

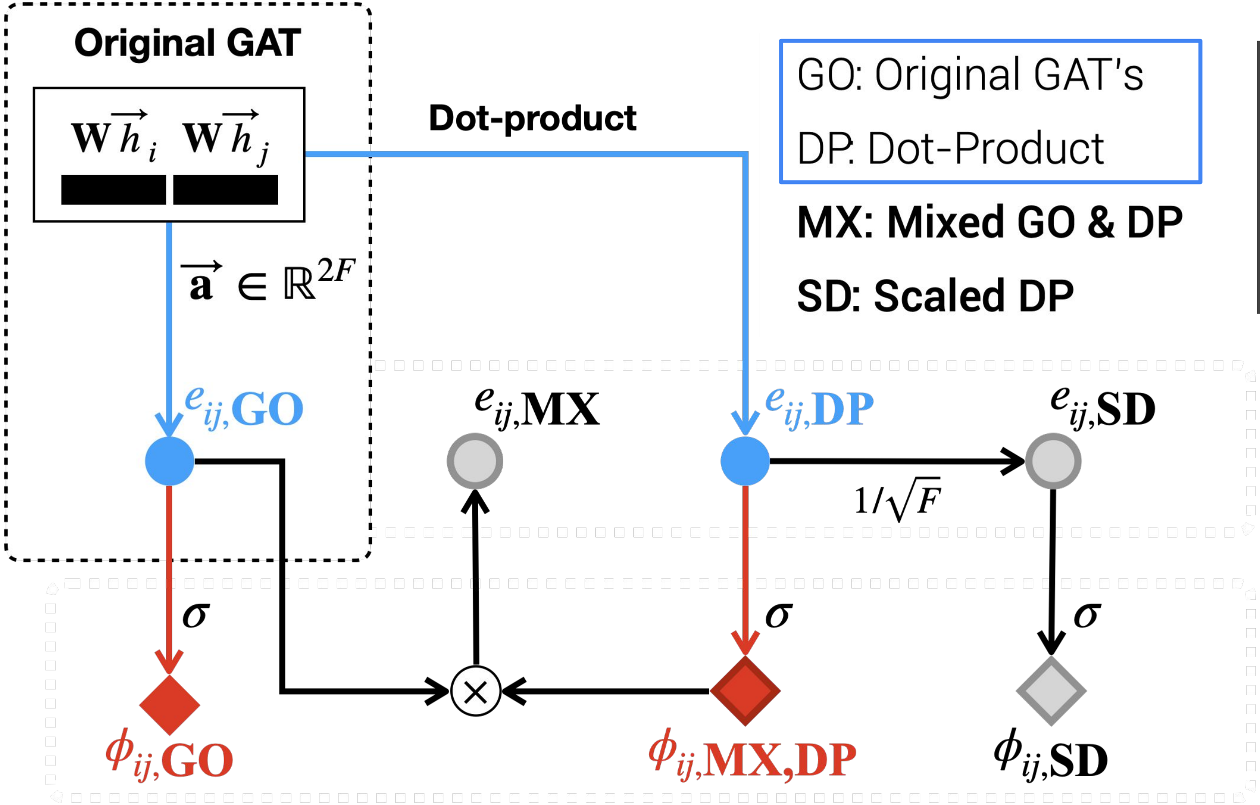


Similar to DeepWalk, LINE, Node2Vec

The dot-product (DP) computes the coefficients by dot-product of two node vectors

$$e_{ij,\mathrm{DP}}^{l+1} = (\boldsymbol{W}^{l+1}\boldsymbol{h}_i^l)^\top \cdot \boldsymbol{W}^{l+1}\boldsymbol{h}_j^l$$

Model



Probability of edges is defined by the sigmoid of the unnormalized attention of GO and DP

$$\phi_{ij,\mathrm{GO}} = \sigma(e_{ij,\mathrm{GO}})$$

$$\phi_{ij,\mathrm{DP}} = \sigma(e_{ij,\mathrm{DP}})$$

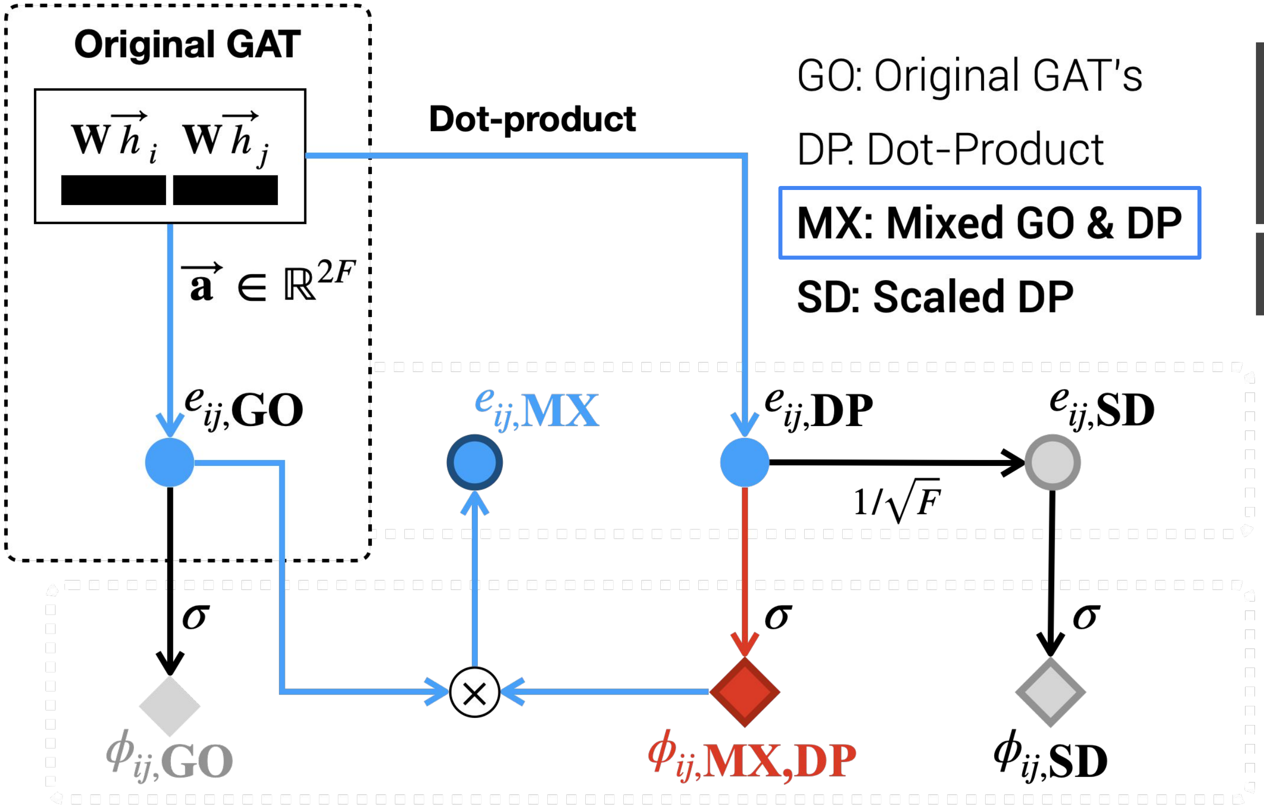
Model

GO: Original GAT's

DP: Dot-Product

MX: Mixed GO & DP

SD: Scaled DP



Motivated by GRU (Cho et al., 2014)

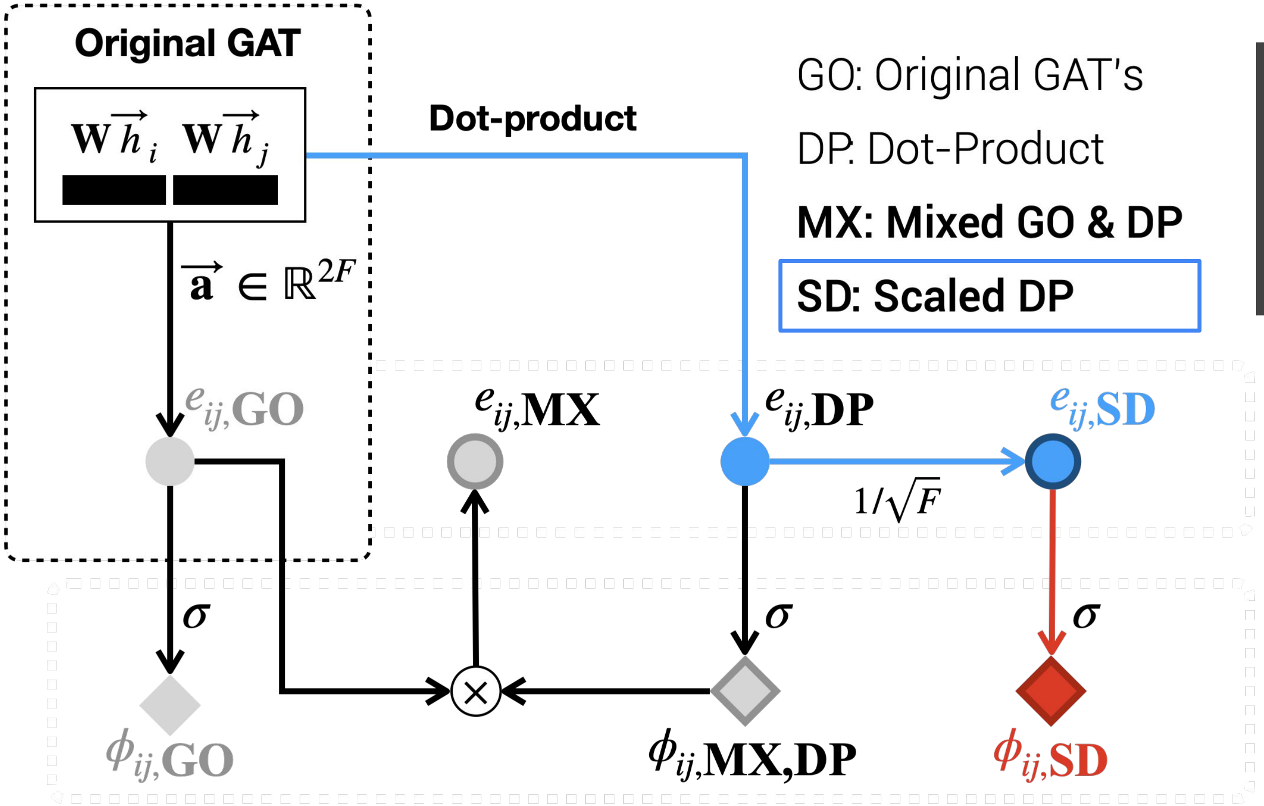
MX attention e is the multiplication of $e_{ij,\mathrm{GO}}$ and $\phi_{ij,\mathrm{DP}}$

$\phi_{ij,\mathrm{MX}}$ equals to $\phi_{ij,\mathrm{DP}}$

$$e_{ij,\mathrm{MX}} = e_{ij,\mathrm{GO}} \cdot \sigma(e_{ij,\mathrm{DP}})$$

$$\phi_{ij,\mathrm{MX}} = \sigma(e_{ij,\mathrm{DP}})$$

Model



Similar to Transformer
(Vaswani et al., NeurIPS 2017)

SD attention is the dot-product scaled by the number of features

$$e_{ij,\mathrm{SD}} = e_{ij,\mathrm{DP}}/\sqrt{F}$$

$$\phi_{ij,\mathrm{SD}} = \sigma(e_{ij,\mathrm{SD}})$$

Contribution

1. Present models with self-supervised attention using edge information: SuperGAT
2. Analyze GAT's original (GO) and Dot-product (DP) attention: GO is better than DP in label-agreement, but DP is better than GO in link prediction
3. Propose recipes to design graph attention concerning homophily and average degree and confirm its validity

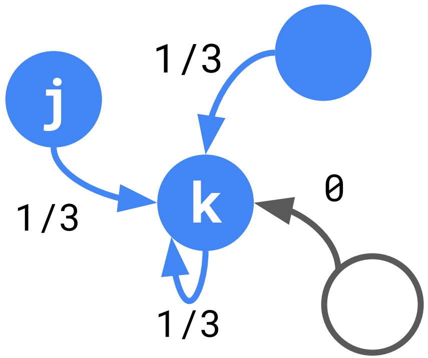
RQ 1. Does Graph Attention Learn Label-Agreement?

_DP_ learns label-agreement worse than _GO_

RQ 1. Does Graph Attention Learn Label-Agreement?

DP learns label-agreement worse than *GO*

Label-agreement is an ideal attention where weights are only given to neighbor nodes with the same label of the center node

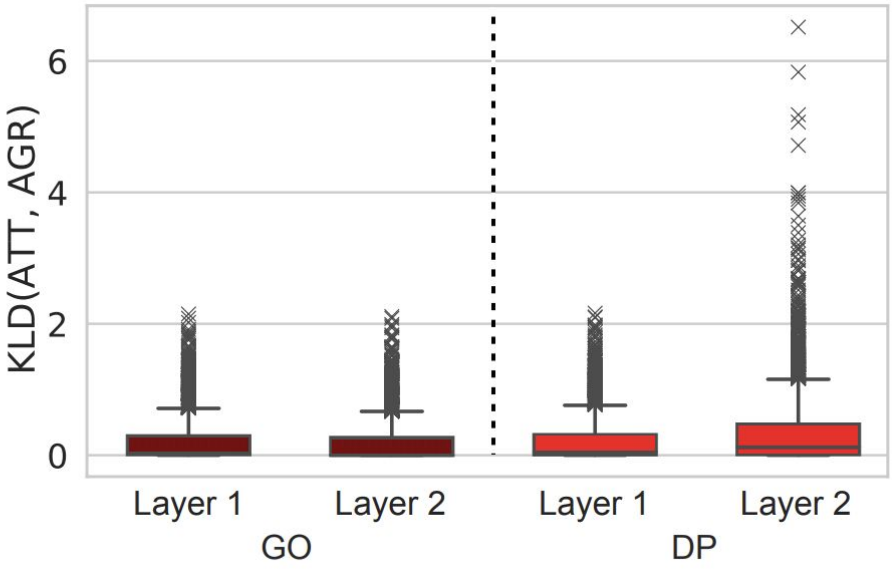


$$\ell_{kj} = \hat{\ell}_{kj} / \sum_s \hat{\ell}_{ks},$$

$$\hat{\ell}_{kj} = 1 \text{ (if } k \text{ and } j \text{ have the same label) or } 0 \text{ (otherwise)}$$

RQ 1. Does Graph Attention Learn Label-Agreement?

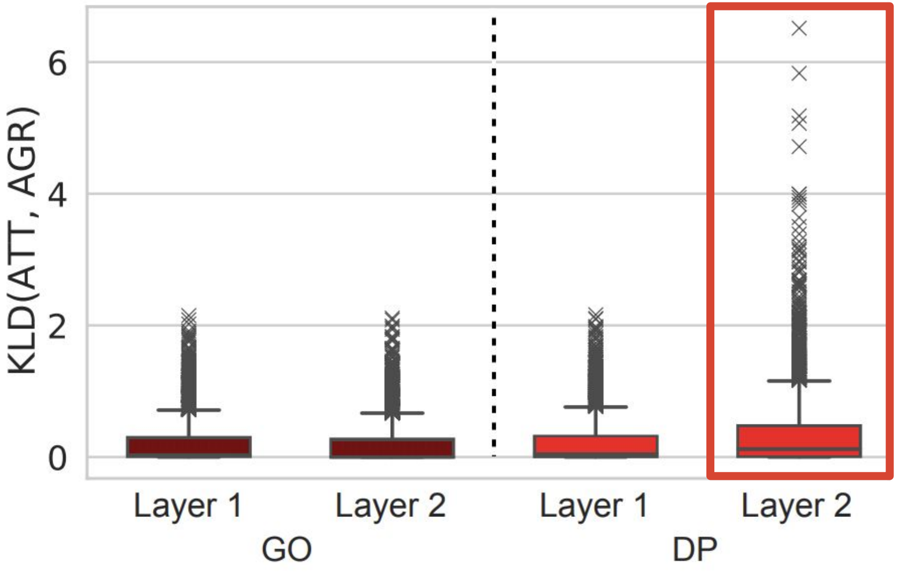
DP learns label-agreement worse than *GO*



$$\mathrm{KLD}(\boldsymbol{\alpha}_k, \boldsymbol{\ell}_k) = \sum_{j \in \mathbb{N}_k \cup \{k\}} \alpha_{kj} \log(\alpha_{kj} / \ell_{kj})$$

RQ 1. Does Graph Attention Learn Label-Agreement?

DP learns label-agreement worse than *GO*



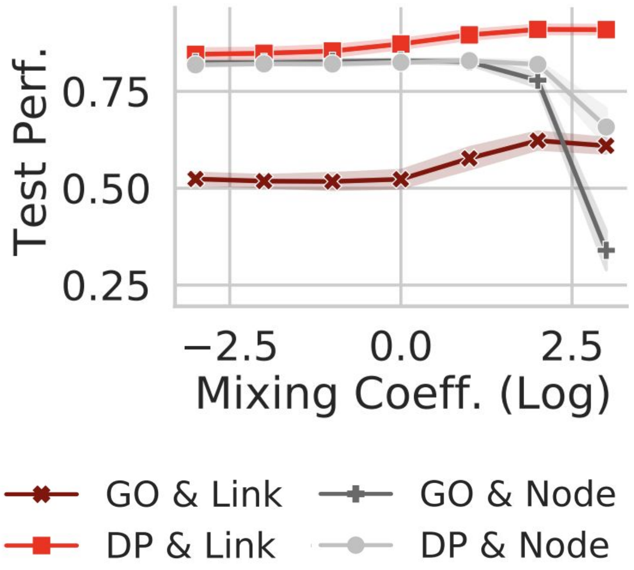
DP attention has a larger KL divergence between label-agreement and the learned attention distribution

RQ 2. Is Graph Attention Predictive for Edge Presence?

_GO_ predicts edge presence worse than _DP_

RQ 2. Is Graph Attention Predictive for Edge Presence?

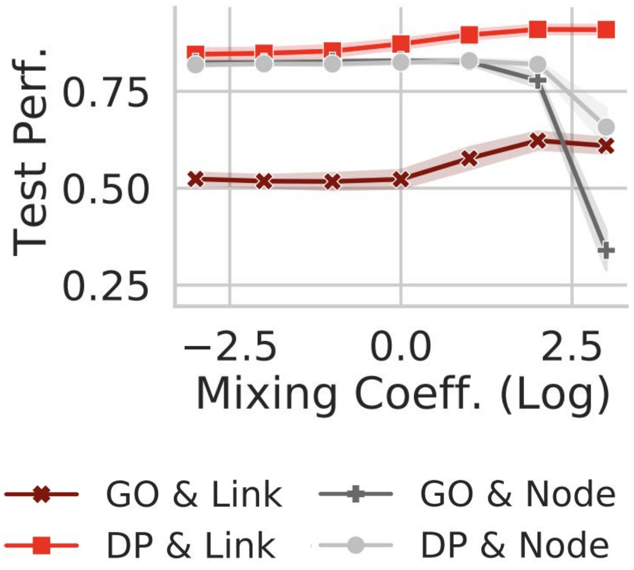
GO **predicts edge presence worse than** ***DP***



GO attention underperforms DP attention for the link prediction task

RQ 2. Is Graph Attention Predictive for Edge Presence?

GO predicts edge presence worse than *DP*



GO attention underperforms DP attention for the link prediction task

Node classification performance decreases when we give too much self-supervision to GO and DP attention

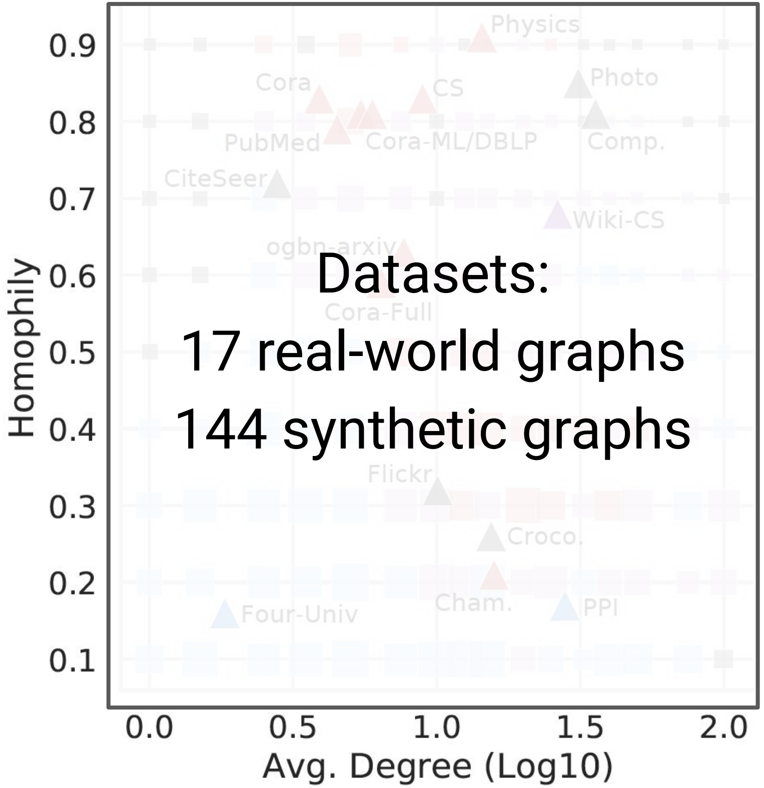
RQ 1&2. How Proper Are Classic Attentions for Self-Supervision?

GO & DP* are not proper for encoding self-supervision, we need more advanced versions: *MX & SD

Contribution

1. Present models with self-supervised attention using edge information: SuperGAT
2. Analyze GAT's original (GO) and Dot-product (DP) attention: GO is better than DP in label-agreement, but DP is better than GO in link prediction
3. Propose recipes to design graph attention concerning homophily and average degree and confirm its validity

RQ 3&4. What graph attention design should we use?



Datasets:

17 real-world graphs

144 synthetic graphs

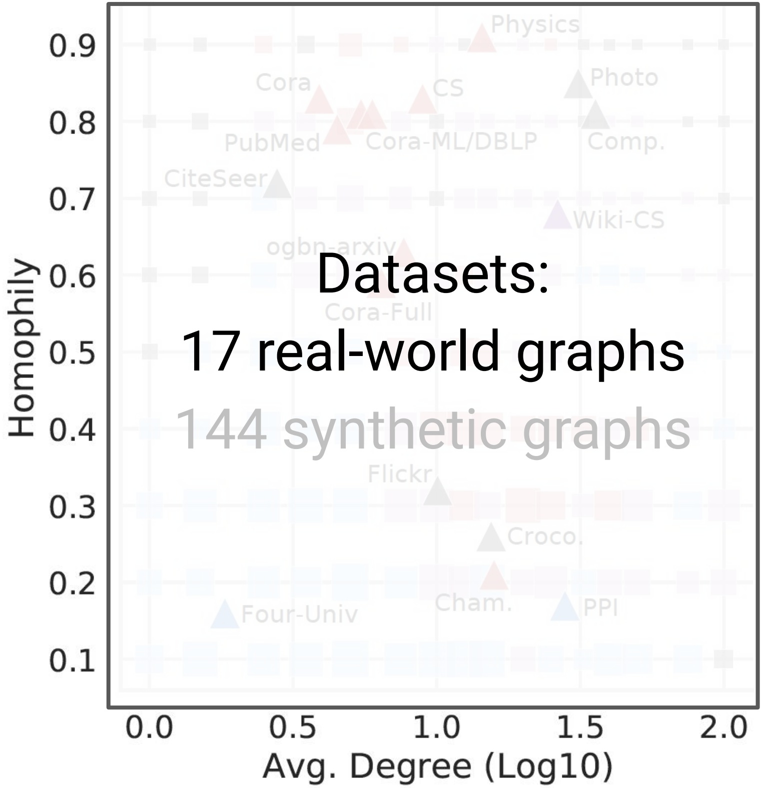
	Synthetic	Real-world
MX is the best	■	▲
SD is the best	■	▲
No significant difference* between MX and SD	■	▲
No significant difference* between all models	■	▲

*significance: p-value ≤ .05

Four models:

GCN, GAT, SuperGAT MX & SD

RQ 3&4. What graph attention design should we use?



Datasets:

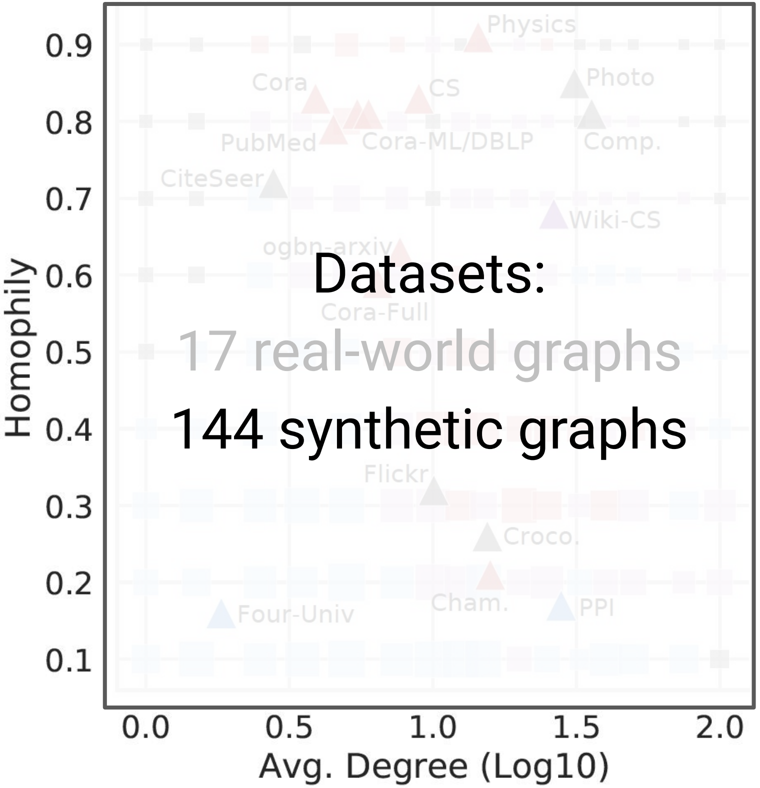
17 real-world graphs

144 synthetic graphs

Table 4: Average degree and homophily of real-world graphs.

Dataset	Degree	Homophily
Four-Univ	1.83 ± 1.71	0.16
PPI	28.0 ± 39.26	0.17
Chameleon	15.85 ± 18.20	0.21
Crocodile	15.48 ± 15.97	0.26
Flickr	10.08 ± 31.75	0.32
Cora-Full	6.41 ± 8.79	0.59
ogbn-arxiv	7.68 ± 9.05	0.63
Wiki-CS	26.40 ± 36.04	0.68
CiteSeer	2.78 ± 3.39	0.72
PubMed	4.50 ± 7.43	0.79
Cora-ML	5.45 ± 8.24	0.81
DBLP	5.97 ± 9.35	0.81
Computers	35.76 ± 70.31	0.81
Cora	3.90 ± 5.23	0.83
CS	8.93 ± 9.11	0.83
Photo	31.13 ± 47.27	0.85
Physics	14.38 ± 15.57	0.91

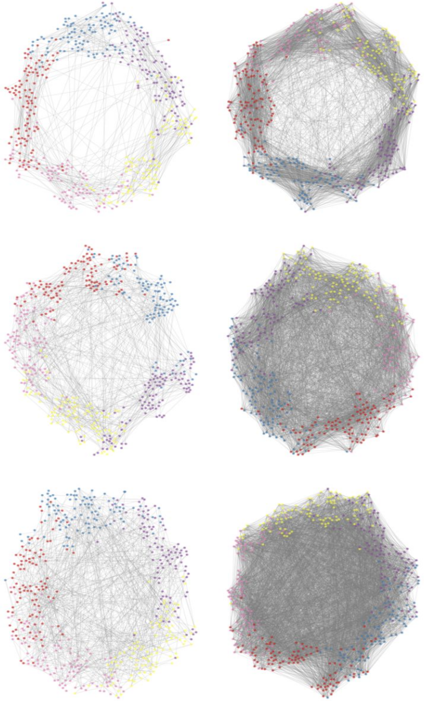
RQ 3&4. What graph attention design should we use?



Datasets:

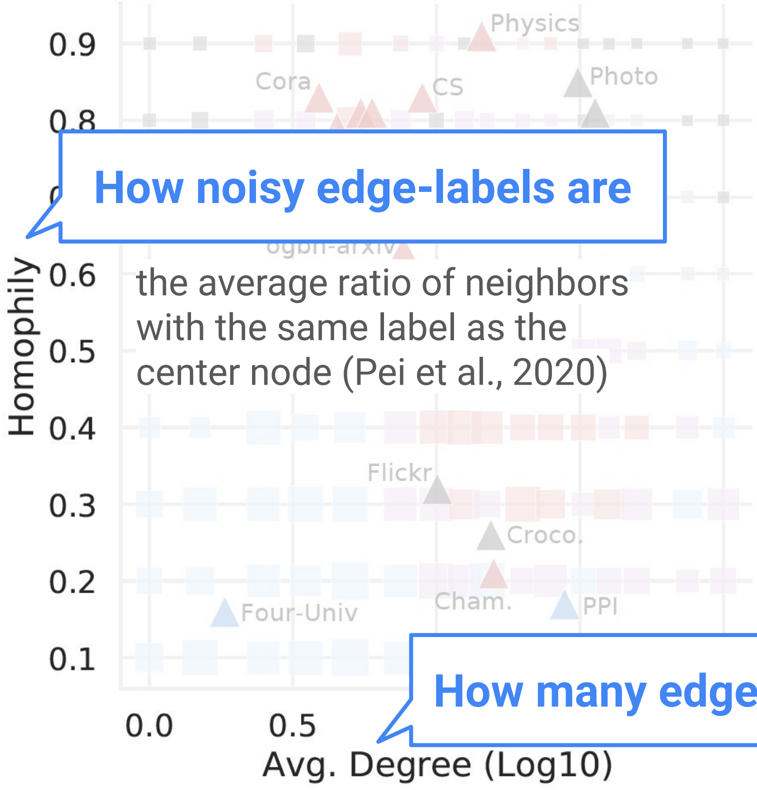
17 real-world graphs

144 synthetic graphs



Random Partition Graphs: If the nodes have the same class labels, they are connected with p_{in}, and otherwise, they are connected with p_{out}

RQ 3&4. What graph attention design should we use?



How noisy edge-labels are

the average ratio of neighbors with the same label as the center node (Pei et al., 2020)

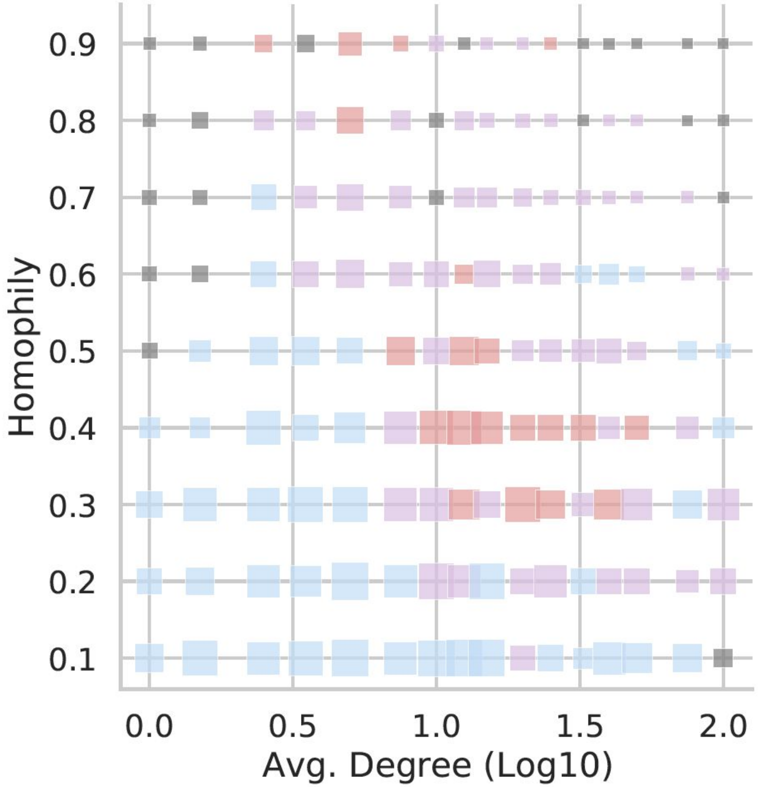
How many edge-labels exist

Best-performed attention depends on *homophily & average degree*

	Synthetic	Real-world
MX is the best	■	▲
SD is the best	■	▲
No significant difference* between MX and SD	■	▲
No significant difference* ...en all models	■	▲

...nce: p-value ≤ .05

RQ 3&4. What graph attention design should we use?



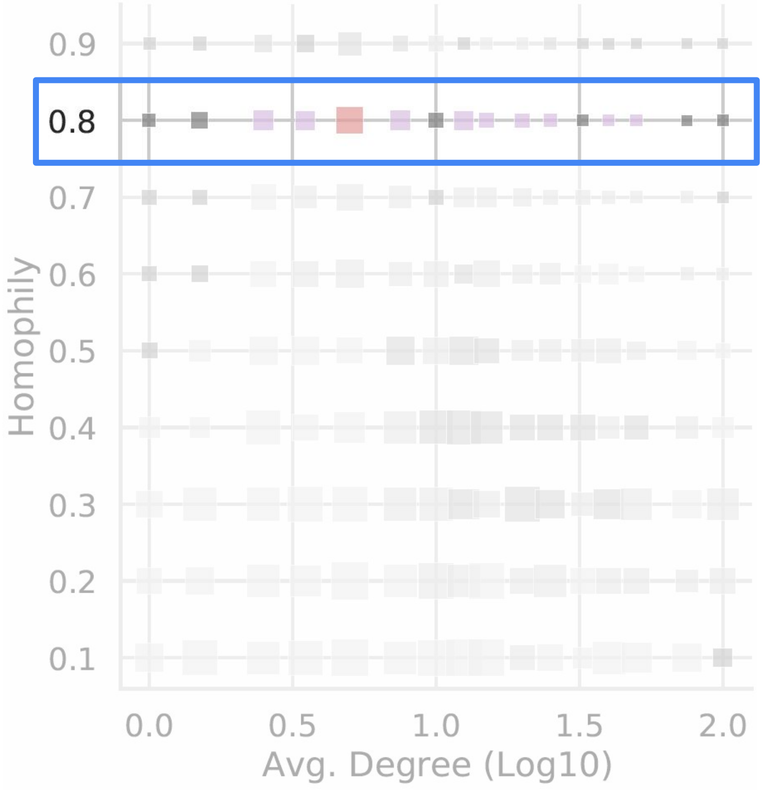
Best-performed attention depends on *homophily & average degree*

	Synthetic	Real-world
MX is the best		
SD is the best		
No significant difference* between MX and SD		
No significant difference* between all models		

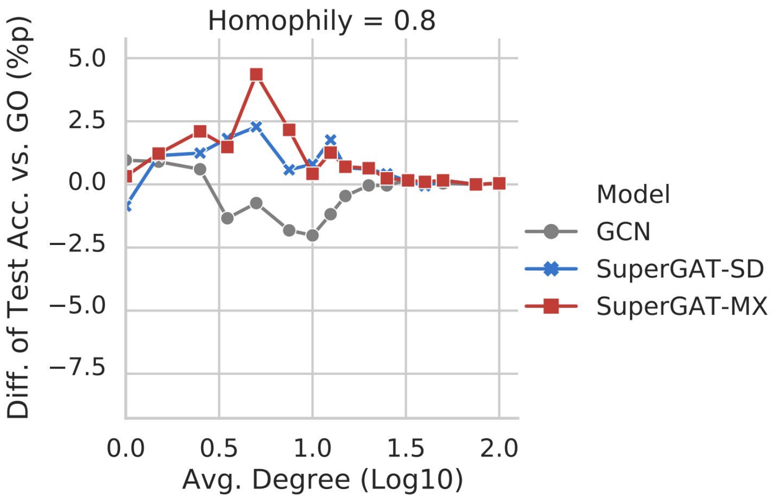
**significance: p-value ≤ .05*

Contribution 3 Propose & confirm recipes to design attention concerning homophily and average degree

RQ 3&4. What graph attention design should we use?

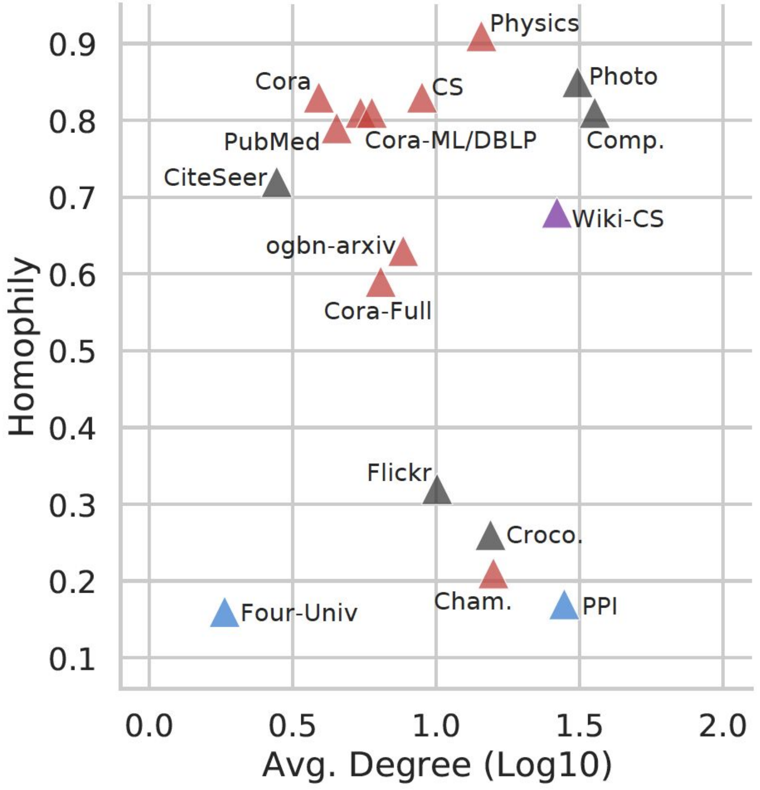


Best-performed attention depends on *homophily & average degree*



Contribution 3 Propose & confirm recipes to design attention concerning homophily and average degree

RQ 3&4. What graph attention design should we use?

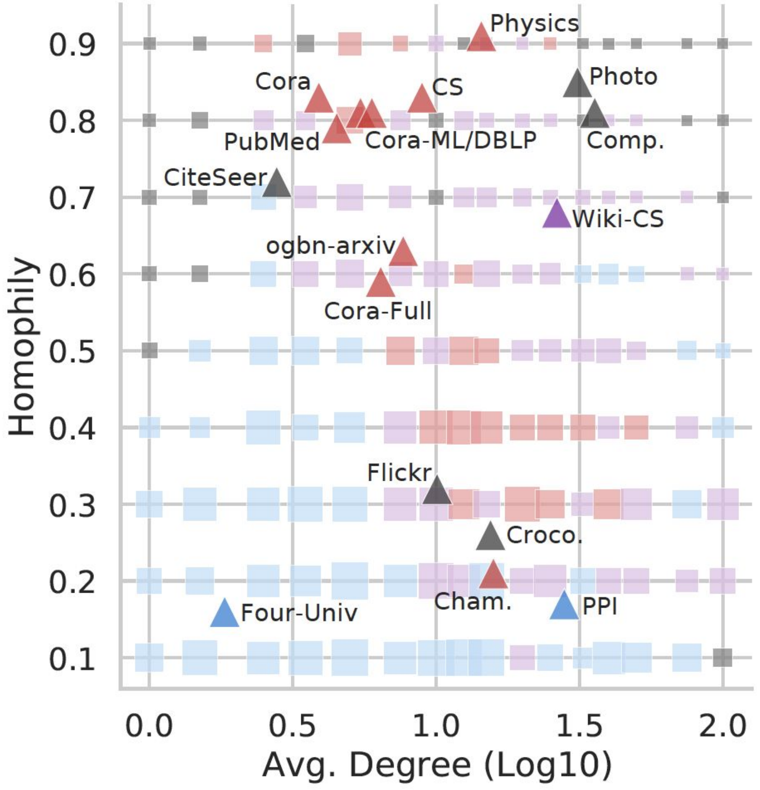


Best-performed attention depends on *homophily & average degree*

	Synthetic	Real-world
MX is the best	■	▲
SD is the best	■	▲
No significant difference* between MX and SD	■	▲
No significant difference* between all models	■	▲

**significance: p-value ≤ .05*

RQ 3&4. What graph attention design should we use?



Best-performed attention depends on *homophily & average degree*

	Synthetic	Real-world
MX is the best	■ (red)	▲ (red)
SD is the best	■ (blue)	▲ (blue)
No significant difference* between MX and SD	■ (purple)	▲ (purple)
No significant difference* between all models	■ (gray)	▲ (black)

**significance: p-value ≤ .05*

Summary

1. Present models with self-supervised graph attention using edge information: SuperGAT

2. Analyze GAT's original (GO) and Dot-product (DP) attention: GO is better than DP in label-agreement, but DP is better than GO in link prediction

3. Propose recipes to design graph attention concerning homophily and average degree and confirm its validity

✉️ dongkwan.kim@kaist.ac.kr
🏠 https://dongkwan-kim.github.io
🔗 https://openreview.net/forum?id=Wi5KUNlqWty
💬 LoGaG slack @Dongkwan Kim