Representation Learning on Graphs

Chanyoung Park

Assistant Professor
Department of Industrial & Systems Engineering
KAIST
cy.park@kaist.ac.kr

This talk

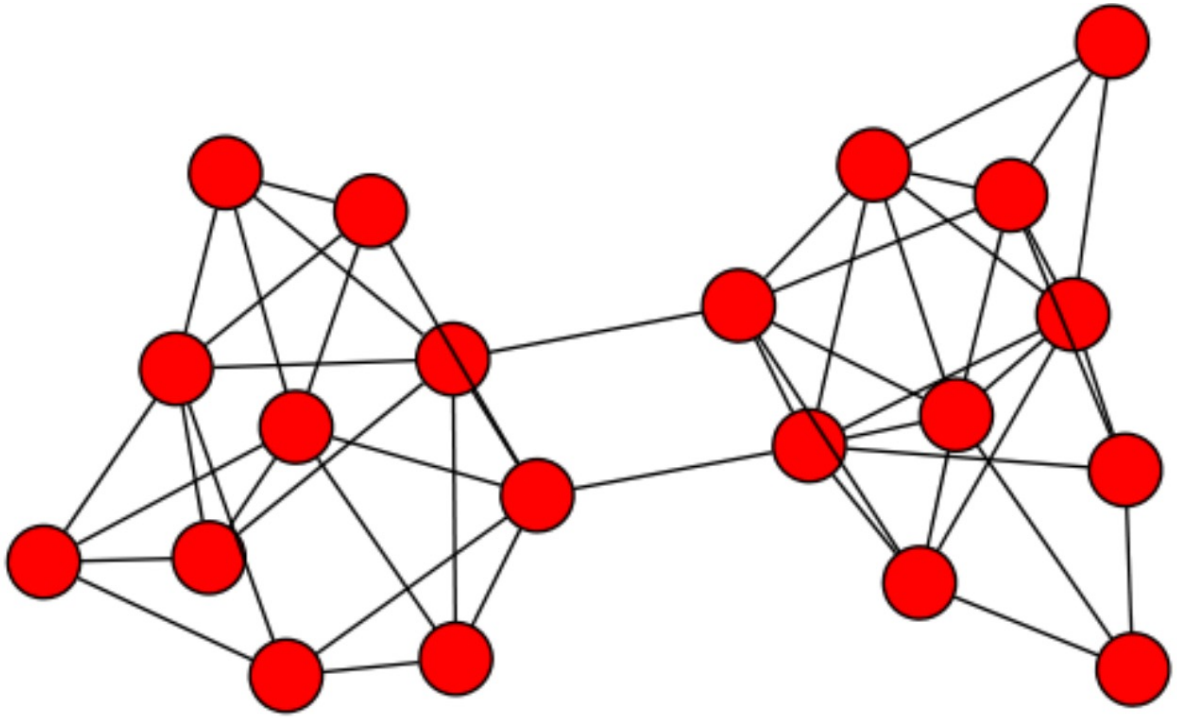
- Overview
- Self-supervised learning on Graphs
  - [AAAI'22] Augmentation-Free Self-Supervised Learning on Graphs
  - [CIKM'22] Relational Self-Supervised Learning on Graphs

This talk

- Overview
- Self-supervised learning on Graphs
  - [AAAI'22] Augmentation-Free Self-Supervised Learning on Graphs
  - [CIKM'22] Relational Self-Supervised Learning on Graphs

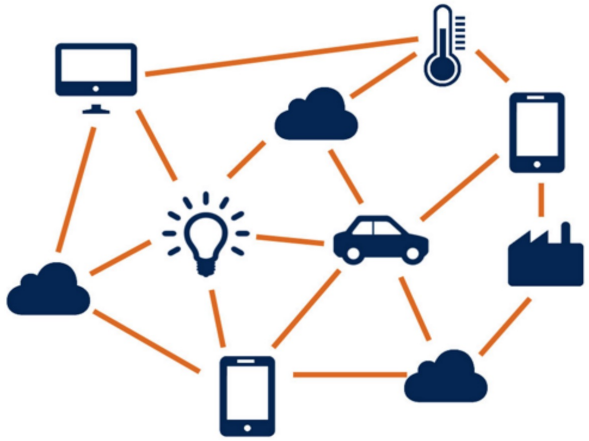
Graph (Network)

- A general description of data and their relations

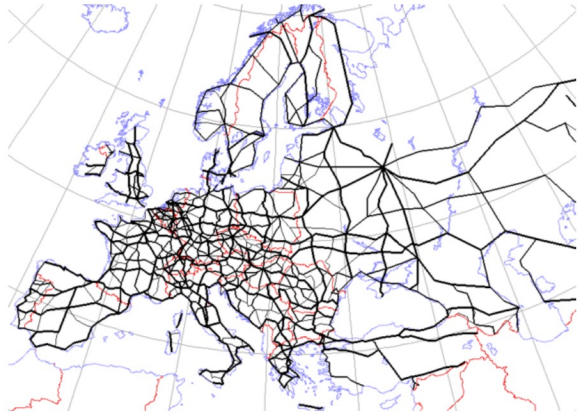


Various Real-World Graphs

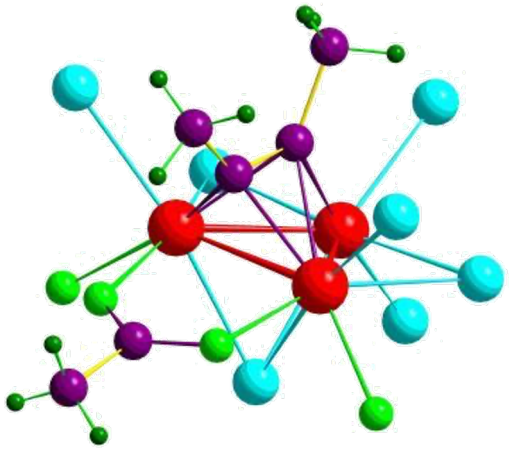
Social graph



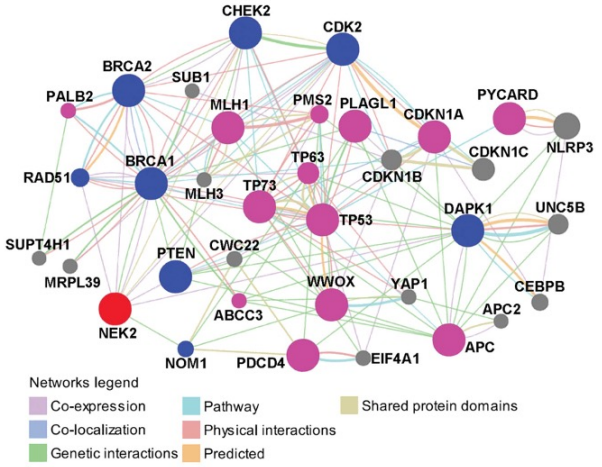
Internet-of-Things



Transportation



Molecular graph



Gene network

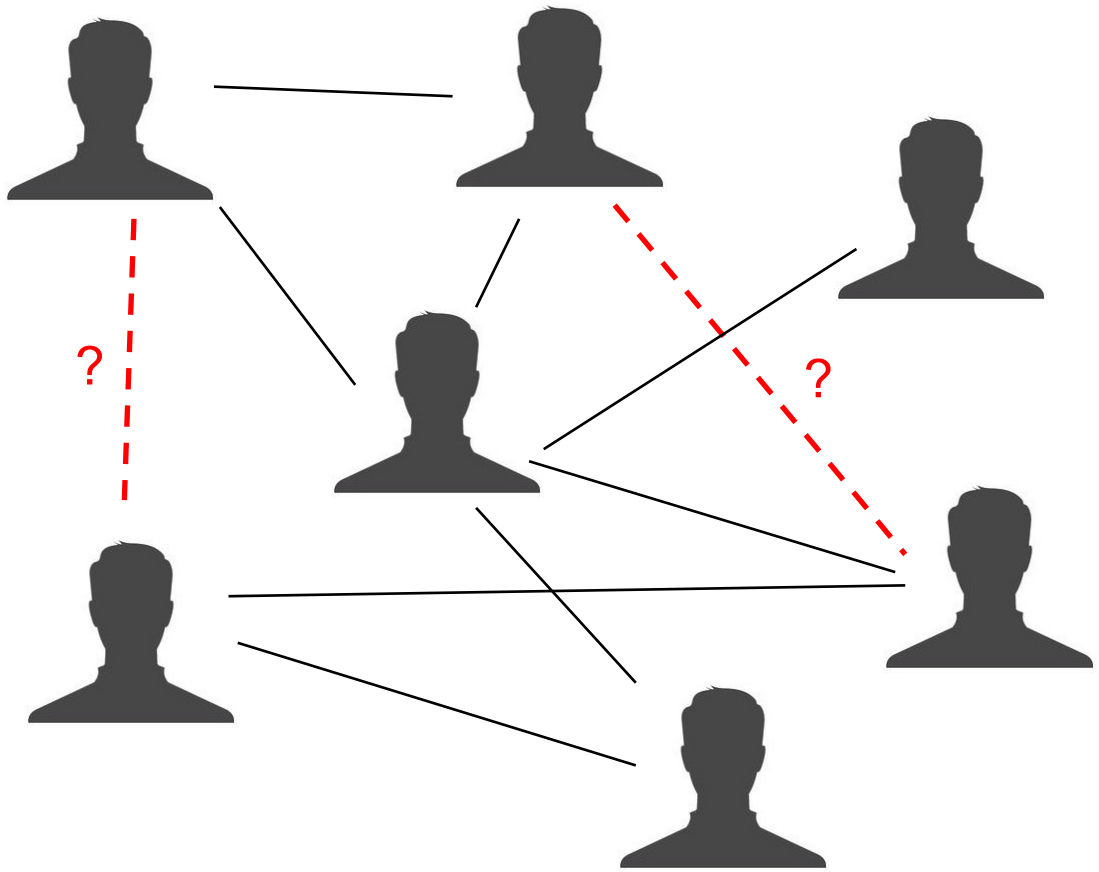
Web graph

(Figure credit) Web

Machine Learning on Graphs

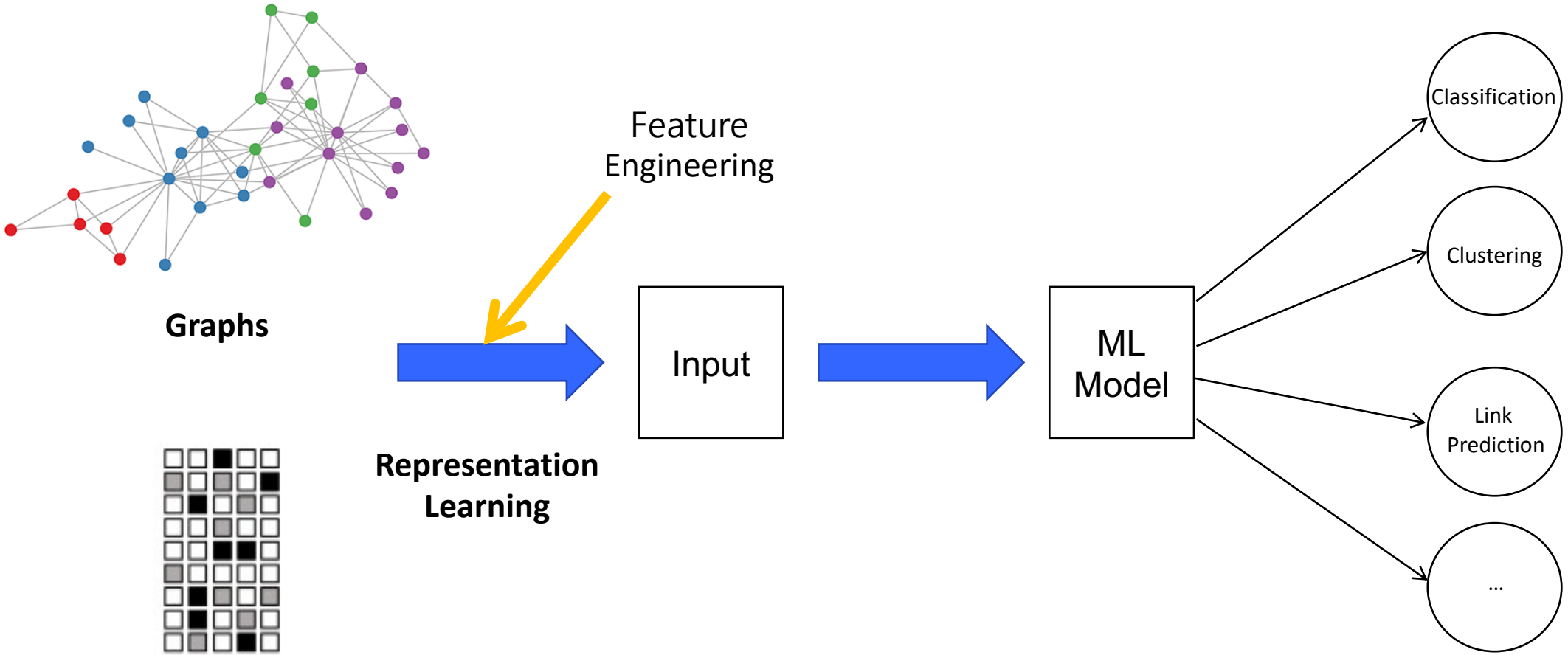
Classical ML tasks in graphs:

- Node classification
  - Predict the type of a given node
- Link prediction
  - Predict whether two nodes are linked
- Community detection
  - Identify densely linked clusters of nodes
- Network similarity
  - How similar are two (sub)networks

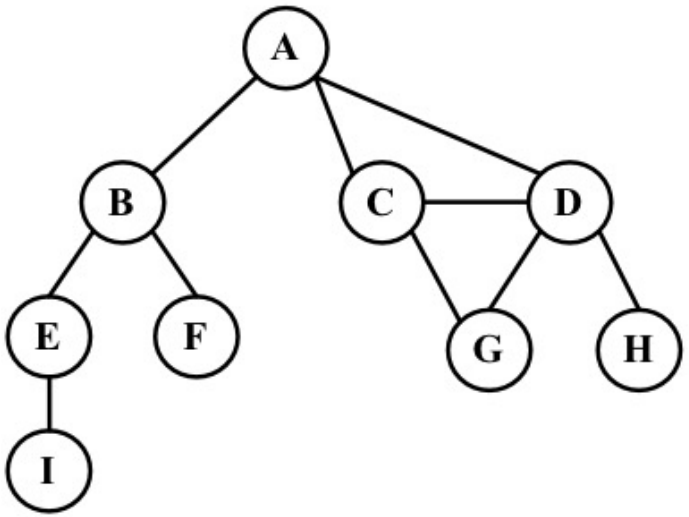


Link Prediction
(Friend Recommendation)

Machine Learning on Graphs



Traditional Graph Representation



	A	B	C	D	E	F	G	H	I
A	0	1	1	1	0	0	0	0	0
B	1	0	0	0	1	1	0	0	0
C	1	0	0	1	0	0	1	0	0
D	1	0	1	0	0	0	1	1	0
E	0	1	0	0	0	0	0	0	1
F	0	1	0	0	0	0	0	0	0
G	0	0	1	1	0	0	0	0	0
H	0	0	0	1	0	0	0	0	0
I	0	0	0	0	1	0	0	0	0

Adjacency matrix

Problems

- Suffer from data sparsity
- Suffer from high dimensionality
- High complexity for computation
- Does not represent “semantics”
- …

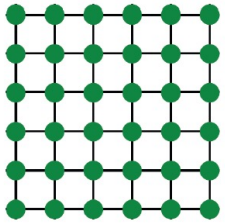
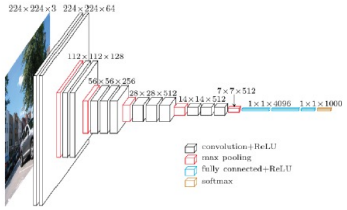
How to effectively and efficiently represent graphs is the key!

→ Deep learning-based approach?

(Figure credit) https://www.oreilly.com/library/view/learning-javascript-data/9781788623872/8a7d3187-7c57-418c-a426-3aceab96f47f.xhtml

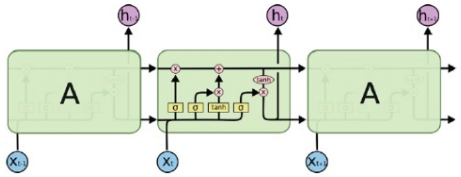
Challenges of Graph Representation Learning

- Existing deep neural networks are designed for data with regular-structure (grid or sequence)
  - CNNs for fixed-size images/grids …

  - RNNs for text/sequences …

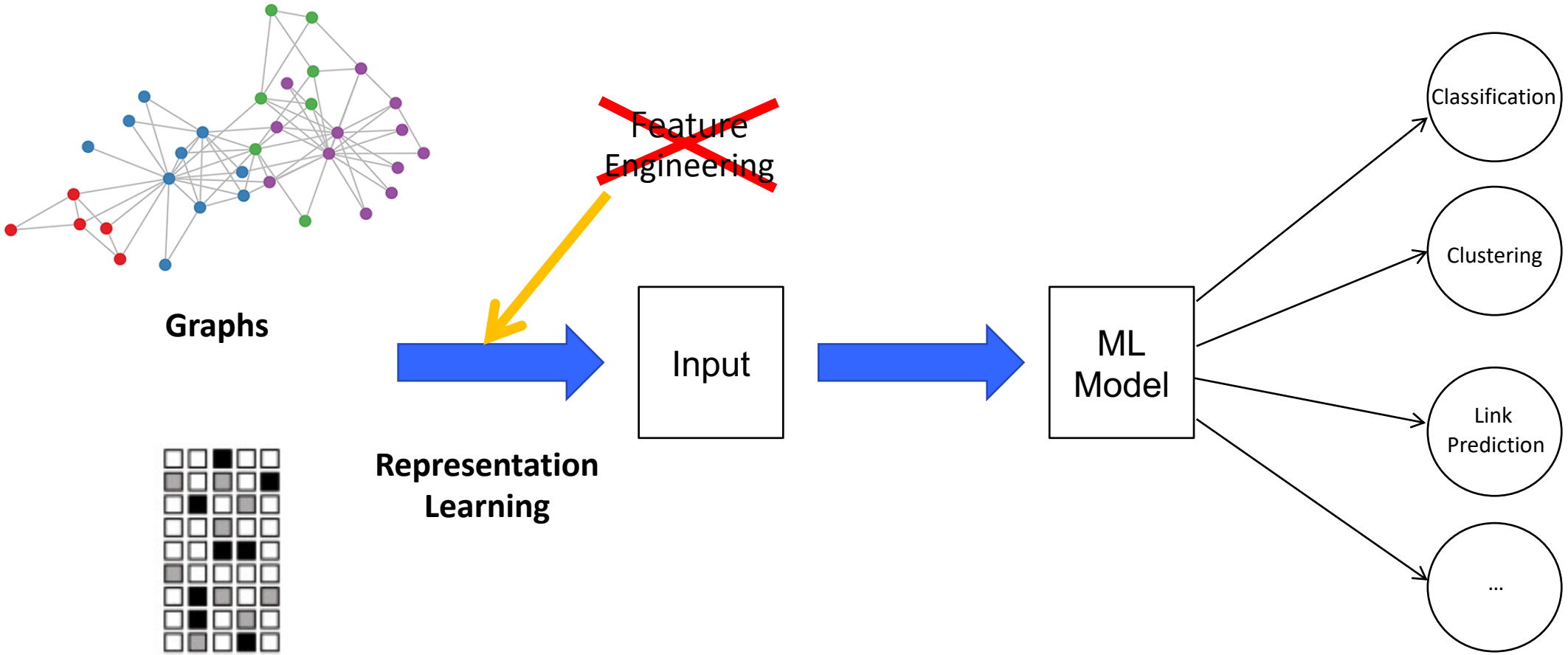




- **Graphs are very complex**
  - Arbitrary structures (no spatial locality like grids / no fixed orderings)
  - Heterogeneous: Directed/undirected, binary/weighted/typed, multimodal features
  - Large-scale: More than millions of nodes and billions of edges

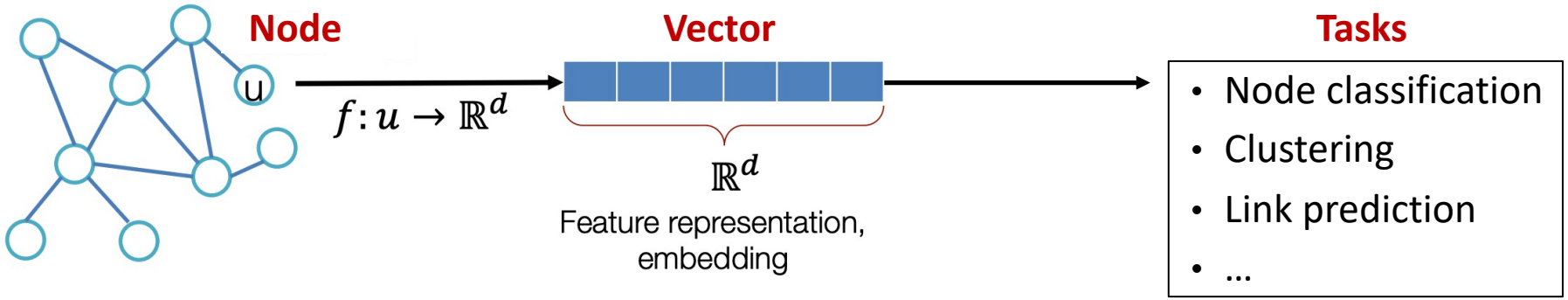
(Figure credit) Tutorial on Graph Representation Learning, AAAI 2019

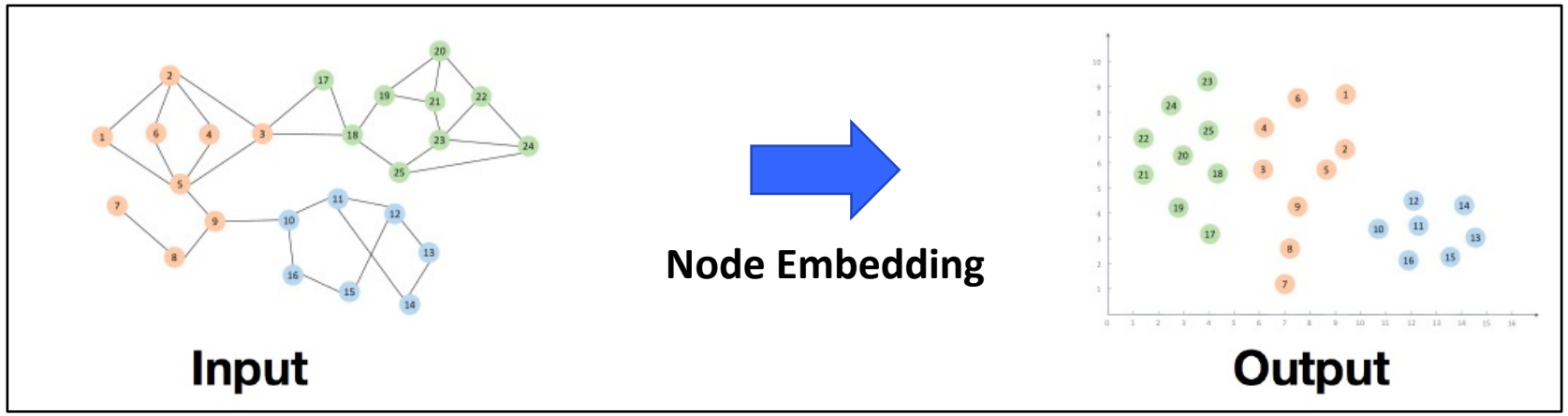
Machine Learning on Graphs



Graph Representation Learning

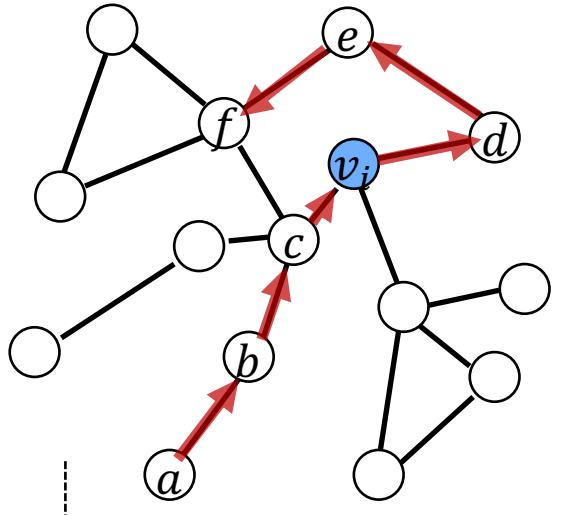
- **Goal**: Encode nodes so that **similarity in the embedding space** approximates **similarity in the original network**
- **Similar nodes in a network have similar vector representations**





Deepwalk

- Deepwalk converts a graph into a collection of node sequences through random walk
- Treat random walks on networks as sentences
- Distributional hypothesis
  - **Word embedding**: Words in similar contexts have similar meanings
  - **Node embedding**: Nodes in similar structural contexts are similar



Deepwalk

$$\mathcal{L}_{DW}(\theta) = \sum_{o \in O} \log\ p(o|\theta) = \sum_{o \in O} \log\ p\big((\mathcal{N}(v_i), v_i)|\theta\big)$$
$$= \sum_{o \in O} \sum_{v_j \in \mathcal{N}(v_i)} \log\ p(v_j|v_i),$$

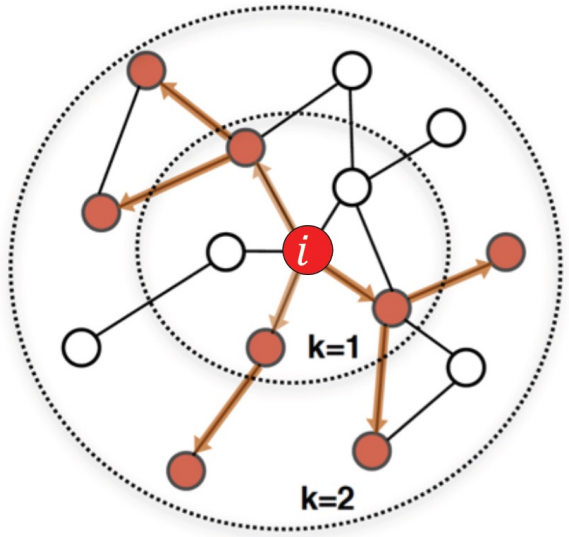
- $\mathcal{O}$: The set of all observations obtained from random walks
- $o = (N(v_i), v_i) \in \mathcal{O}$
  - Center node v_i
  - Neighboring nodes $N(v_i)$

Example seq	$a \to b \to c \to v_i \to d \to e \to f$
Window size=2	$a \to b \to c \to v_i \to d \to e \to f$
Center node	v_i
Neighborhood	$N(v_i) = b, c, d, e$
Observation o	$o = (N(v_i), v_i) = (\{b, c, d, e\}, v_i)$

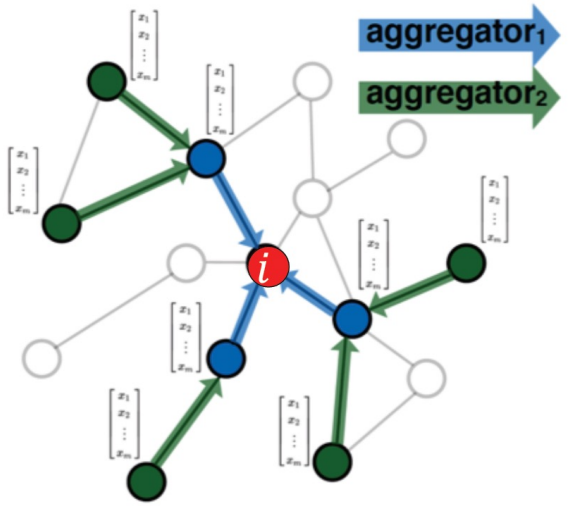
Deepwalk: Online learning of social representations, KDD2014

Graph Convolutional Network (GCN)

- **Idea**: Node's neighborhood defines a computation graph
  - Messages contain **relational information** + **attribute information**



Determine node computation graph



Propagate messages and transform information

Learn how to propagate information across the graph to compute node features

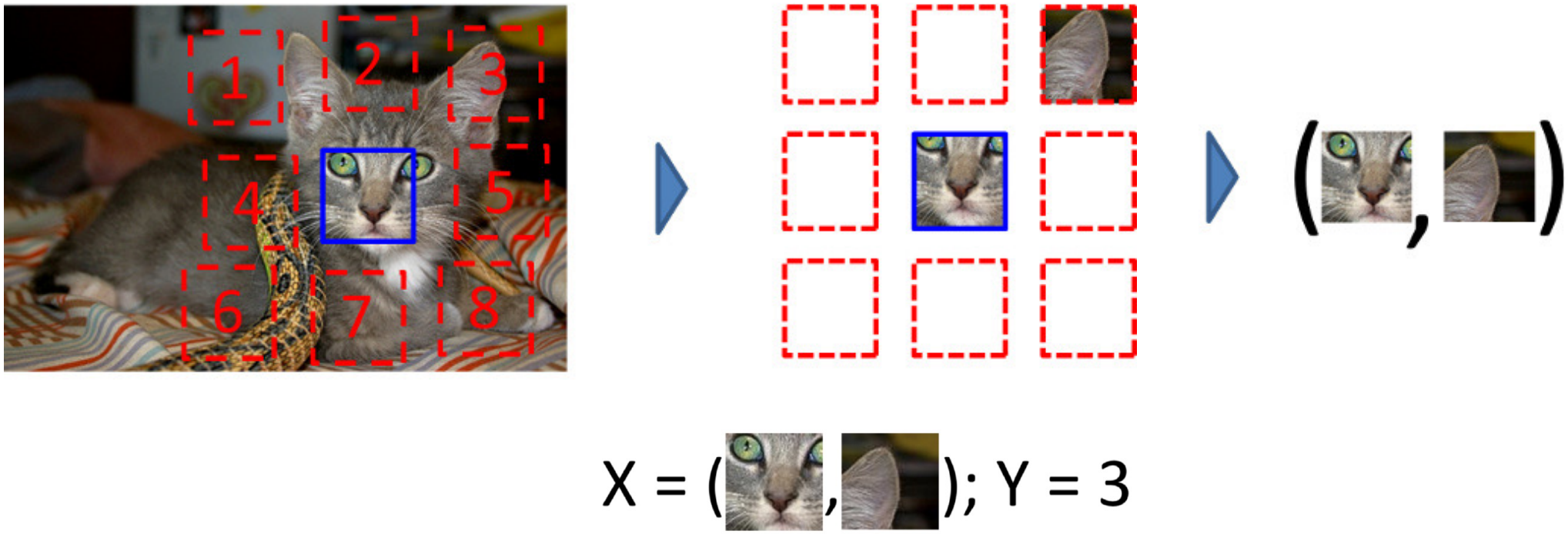
Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." ICLR 2017

This talk

- Overview
- Self-supervised learning on Graphs
  - [AAAI’22] Augmentation-Free Self-Supervised Learning on Graphs
  - [CIKM’22] Relational Self-Supervised Learning on Graphs

What is self-supervised learning?

- A form of unsupervised learning where the data provides the supervision
- In general, withhold some part of the data, and task the network with predicting it
- An example of **pretext task**: **Relative positioning**
  - Train network to predict relative position of two regions in the same image



(Figure credit) Unsupervised visual representation learning by context prediction, ICCV2015

What is self-supervised learning?

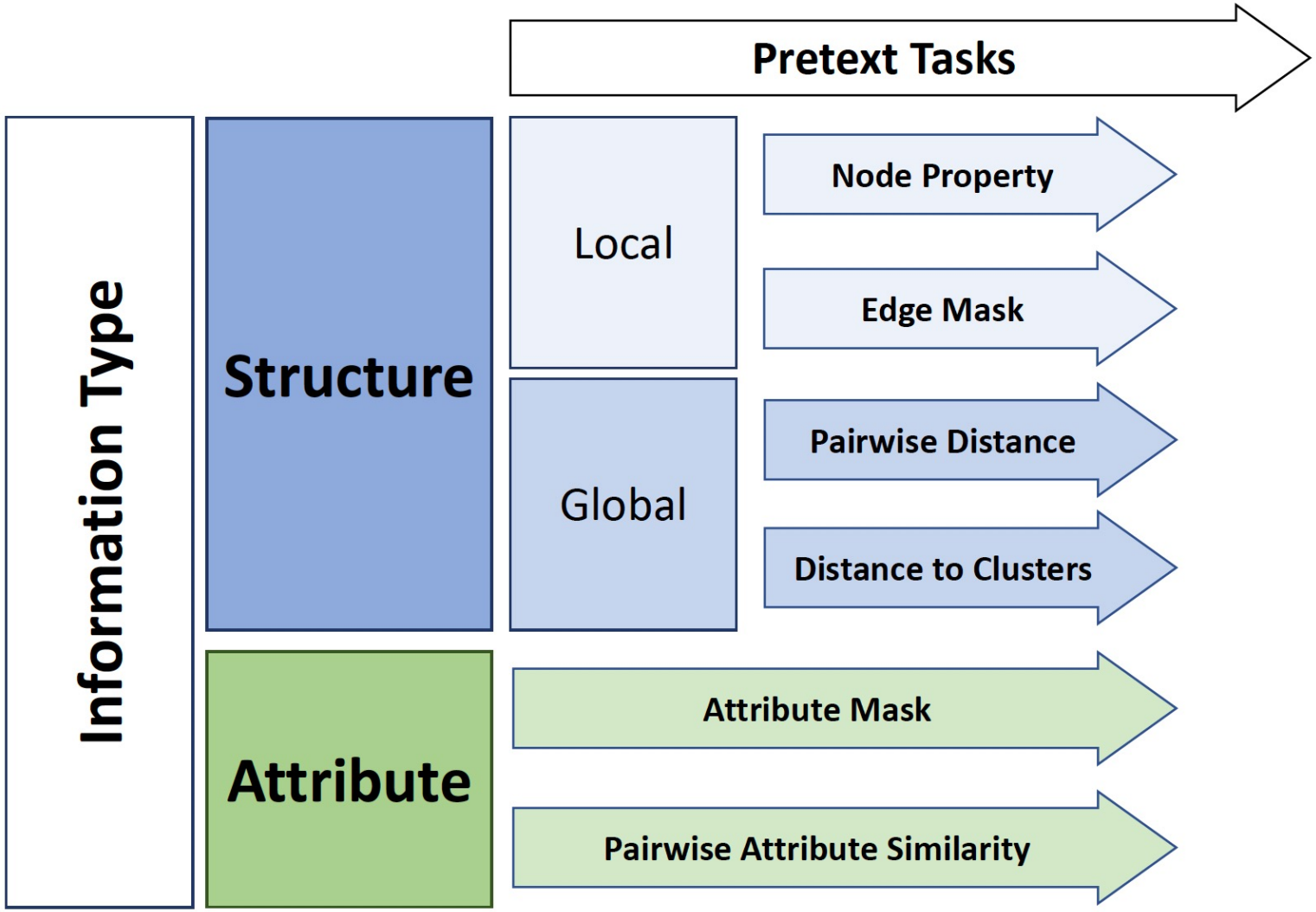
- Pretext task: **Jigsaw puzzle**

- Pretext task : **Colorization**

- Pretext task : **Rotation**
  - Which one has the correct rotation?

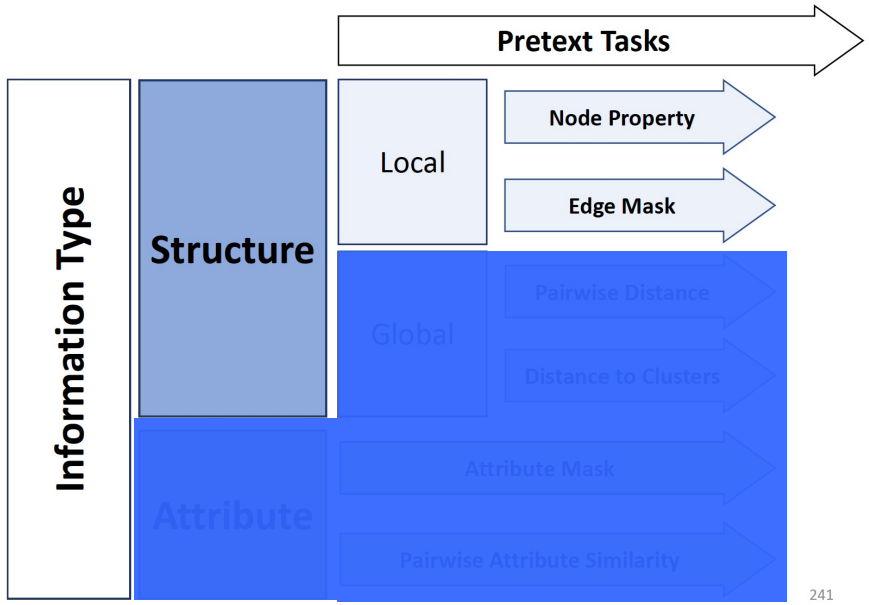
(Figure credit) Self-Supervised Learning, Andrew Zisserman

Examples of Pretext tasks on graphs



(Figure credit) Tutorial in Graph Neural Networks: Models and Applications, AAAI2021

Local Structure-based Pretext Task



- Node property
  - **Goal**: To predict the property for each node in the graph such as their *degree*, *local node importance*, and *local clustering coefficient*.

Predicted degree of node v_i

$$\mathcal{L}_{self}(\theta', \mathbf{A}, \mathbf{X}, \mathcal{D}_U) = \frac{1}{|\mathcal{D}_U|} \sum_{v_i \in \mathcal{D}_U} (f_{\theta'}(\mathcal{G})_{v_i} - d_i)^2$$

Degree of node v_i

- Edge mask
  - **Goal**: To predict *whether or not there exists a link between a given node pair*

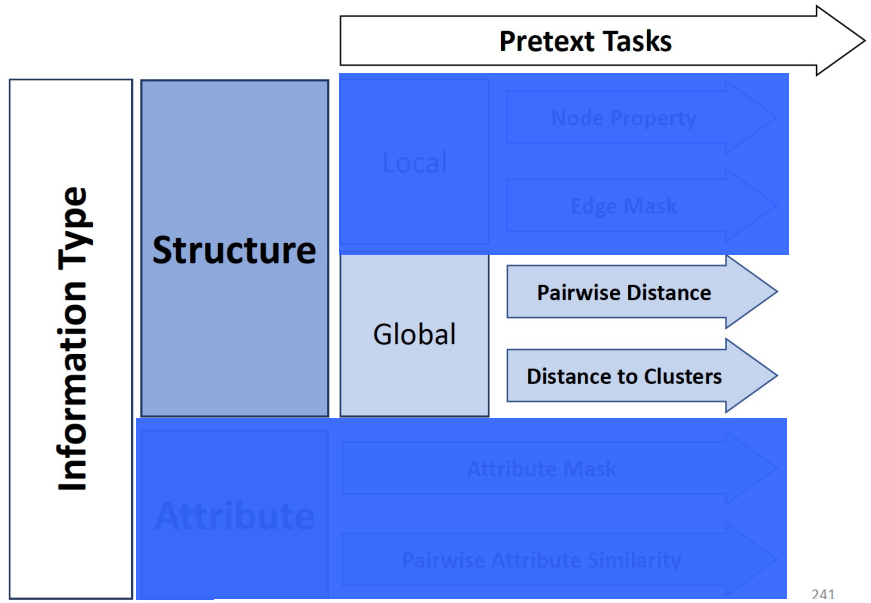
Cross-entropy loss

$$\mathcal{L}_{self}(\theta', \mathbf{A}, \mathbf{X}, \mathcal{D}_U) = \frac{1}{|\mathcal{M}_e|} \sum_{(v_i, v_j) \in \mathcal{M}_e} \ell\left(f_w(|f_{\theta'}(\mathcal{G})_{v_i} - f_{\theta'}(\mathcal{G})_{v_j}|), 1\right) + \frac{1}{|\overline{\mathcal{M}}_e|} \sum_{(v_i, v_j) \in \overline{\mathcal{M}}_e} \ell\left(f_w(|f_{\theta'}(\mathcal{G})_{v_i} - f_{\theta'}(\mathcal{G})_{v_j}|), 0\right)$$

Connected edges

Not connected edges

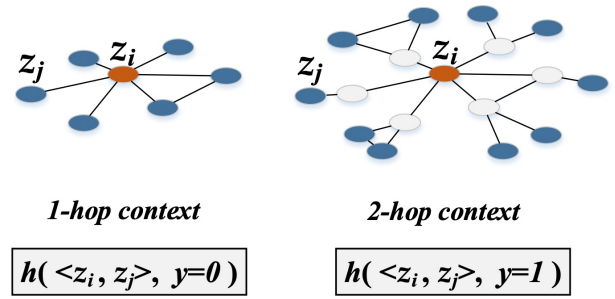
Global Structure-based Pretext Task



- Pairwise distance
  - **Goal**: To predict the distance between different node pair.

$$\mathcal{L}_{self}(\theta', \mathbf{A}, \mathbf{X}, \mathcal{D}_U) = \frac{1}{|\mathcal{S}|} \sum_{(v_i, v_j) \in \mathcal{S}} \ell\left(f_w(|f_{\theta'}(\mathcal{G})_{v_i} - f_{\theta'}(\mathcal{G})_{v_j}|), C_{p_{ij}}\right)$$

Pairwise distance between node v_i and v_j



- Distance2Clusters
  - **Goal**: To predict the distance from the unlabeled nodes to predefined graph clusters
  - Step 1: Apply graph clustering to get k clusters $\{C_1, C_2, \ldots, C_k\}$
  - Step 2: In each cluster C_j, assume the node with the highest degree as the center node

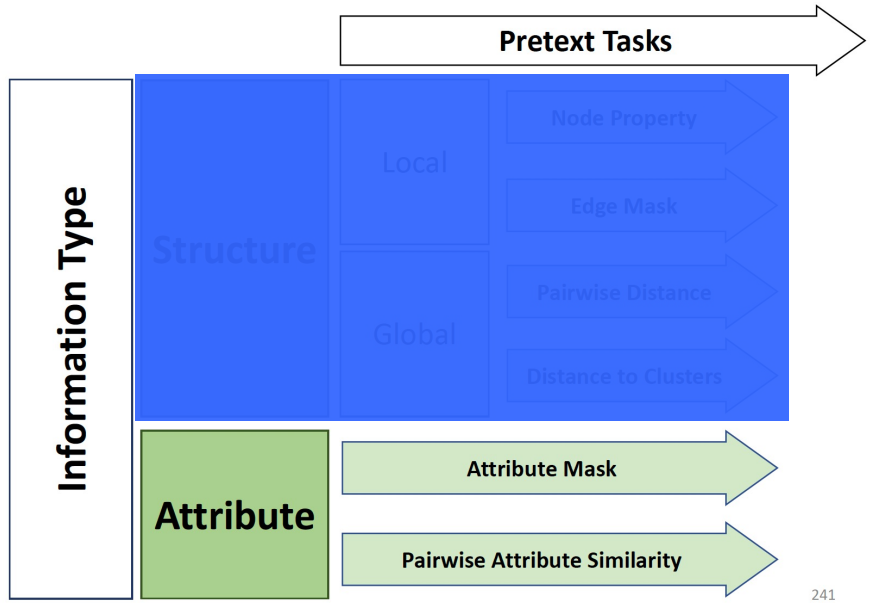
$$\mathcal{L}_{self}(\theta', \mathbf{A}, \mathbf{X}, \mathcal{D}_U) = \frac{1}{|\mathcal{D}_U|} \sum_{v_i \in \mathcal{D}_U} \|f_{\theta'}(\mathcal{G})_{v_i} - \mathbf{d}_i\|^2$$

$$\boldsymbol{d_i} = [d_{i1}, d_{i2}, \ldots, d_{ik}]$$

Distance from node v_i to cluster c_2

Self-Supervised Graph Representation Learning via Global Context Prediction, arxiv2020
Self-supervised Learning on Graphs: Deep Insights and New Directions, arxiv2020

Attribute-based Pretext Task



- Attribute mask
  - **Goal**: To predict the masked attribute
  - Apply PCA to reduce the dimensionality of features

$$\mathcal{L}_{self}(\theta', \mathbf{A}, \mathbf{X}, \mathcal{D}_U) = \frac{1}{|\mathcal{M}_a|} \sum_{v_i \in \mathcal{M}_a} \| f_{\theta'}(\mathcal{G})_{v_i} - \mathbf{x}_i \|^2$$

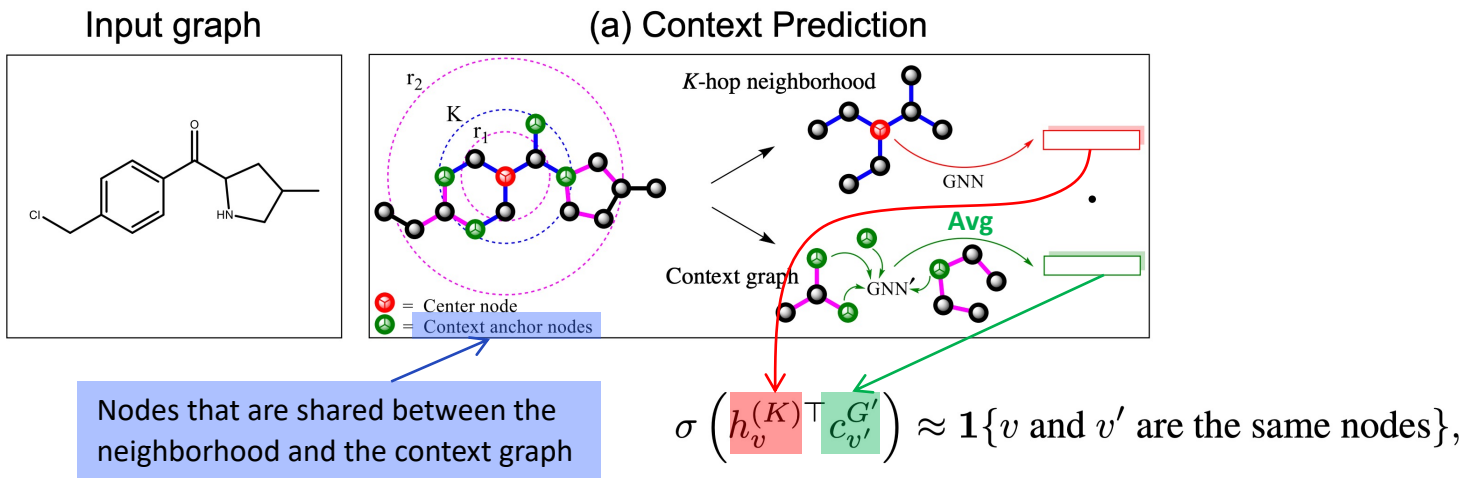
Feature of node v_i

- Pairwise attribute similarity
  - **Goal**: To predict the similarity of pairwise node features

$$\mathcal{L}_{self}(\theta', \mathbf{A}, \mathbf{X}, \mathcal{D}_U) = \frac{1}{|\mathcal{T}|} \sum_{(v_i, v_j) \in \mathcal{T}} \| f_w(|f_{\theta'}(\mathcal{G})_{v_i} - f_{\theta'}(\mathcal{G})_{v_j}|) - s_{ij} \|^2$$

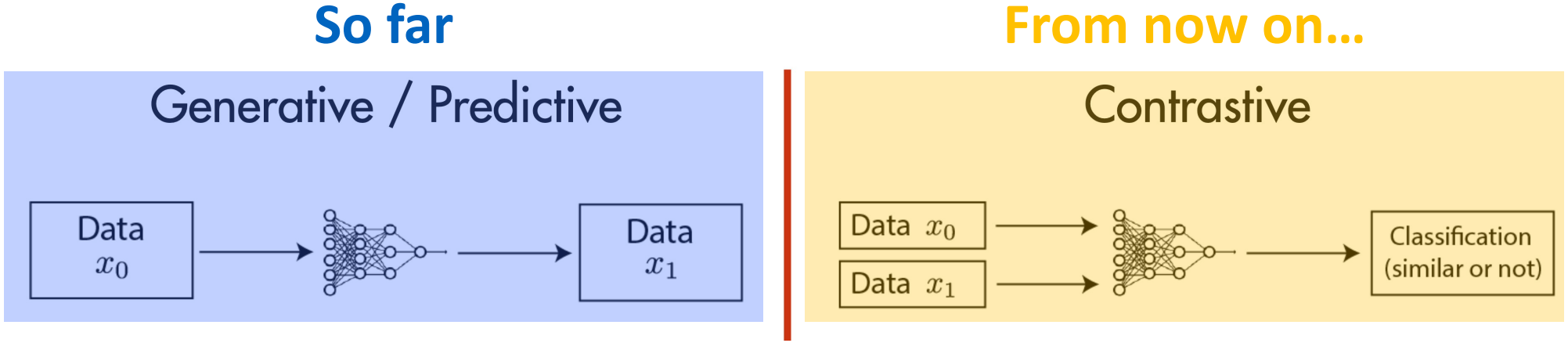
Context prediction

- **Pretext task:** Context prediction



	Chemistry			Biology		
	Non-pre-trained	Pre-trained	Gain	Non-pre-trained	Pre-trained	Gain
GIN	67.0	**74.2**	**+7.2**	64.8 ± 1.0	**74.2 ± 1.5**	**+9.4**
GCN	**68.9**	72.2	+3.4	63.2 ± 1.0	70.9 ± 1.7	+7.7
GraphSAGE	68.3	70.3	+2.0	65.7 ± 1.2	68.5 ± 1.5	+2.8
GAT	66.8	60.3	-6.5	**68.2 ± 1.1**	67.8 ± 3.6	-0.4

Taxonomy of Self-Supervised Learning

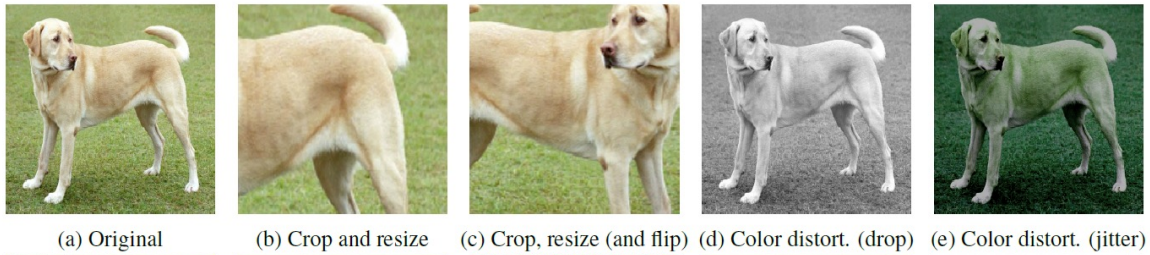


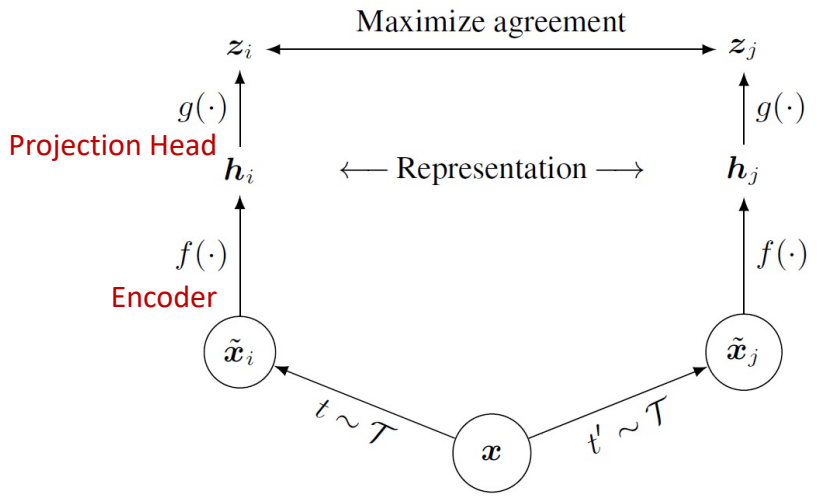
- **Contrastive learning**
  - **Given:** $X = \{x, \ x^+, \ x_1^-, \ldots, x_{N-1}^-\}$; Similarity function $s(\cdot)$ (e.g., cosine similarity)
  - **Goal:** $s\big(f(x), f(x^+)\big) > s(f(x), f(x^-))$
  - **Contrastive/InfoNCE Loss**

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{\exp\left(s\big(f(x), f(x^+)\big)\right)}{\exp\left(s\big(f(x), f(x^+)\big)\right) + \sum_{j=1}^{N-1} \exp\left(s\left(f(x), f\left(x_j^-\right)\right)\right)} \right]$$

(Figure credit) https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-learning.html

The Contrastive Learning Paradigm





- Algorithm
  - 1) Sample mini batch of N examples
  - 2) Create $2N$ data points via Data Augmentation
  - 3) Given a positive pair, treat other $2(N-1)$ points as negative examples
  - → **Instance Discrimination!**

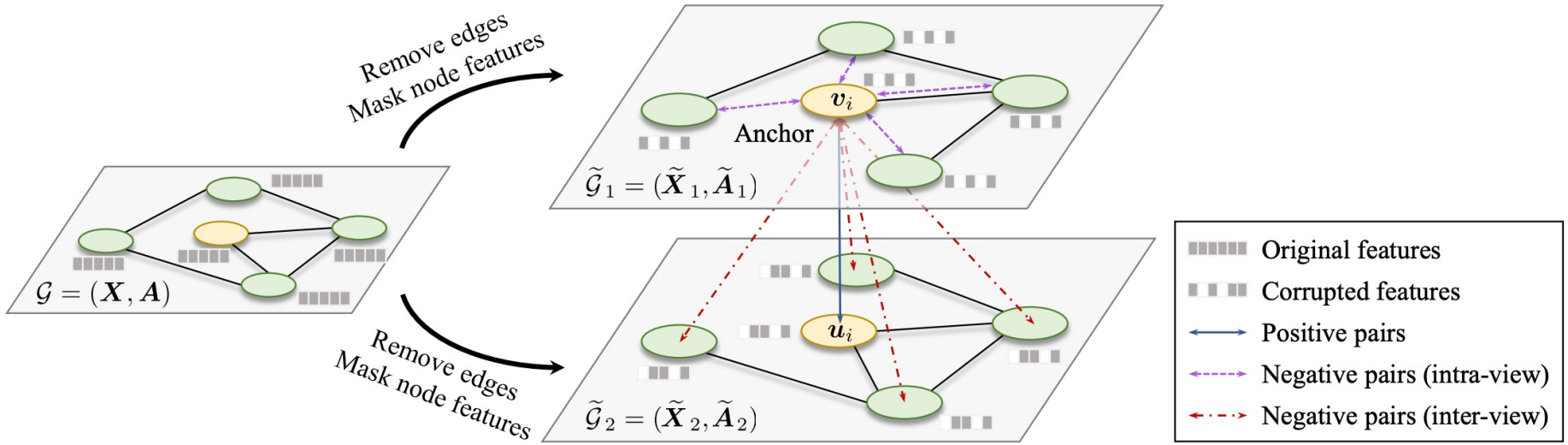
$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Reduce: Dist. between representations of different augmented views of the same image (Positive)
Increase: Dist. between representations of augmented views from different images (Negative)

A simple framework for contrastive learning of visual representations, ICML 2020

Deep Graph Contrastive Representation Learning (GRACE)

- **Pull** the representation of the same node in the two augmented graphs
- **Push** apart representations of every other node



$$\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) = \log \frac{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}{\underbrace{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}_{\text{the positive pair}} + \underbrace{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_k)/\tau}}_{\text{inter-view negative pairs}} + \underbrace{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} e^{\theta(\boldsymbol{u}_i, \boldsymbol{u}_k)/\tau}}_{\text{intra-view negative pairs}}},$$

Shortcomings of Contrastive Methods

- 1) Requires negative samples → **Sampling bias**
  - Treat different image as negative even if they share the semantics
- 2) Requires careful augmentation

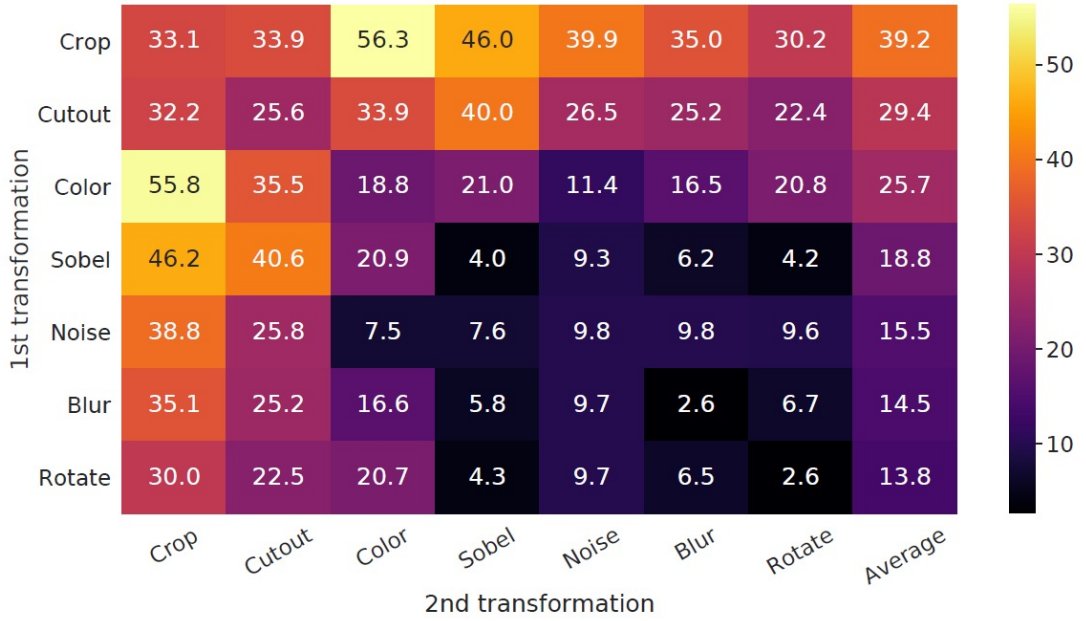
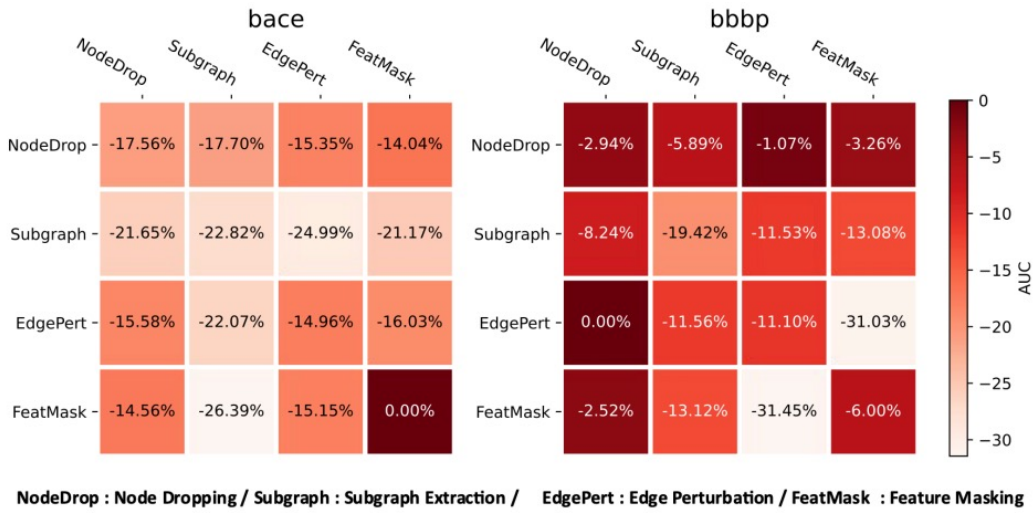


Image classification



Graph classification

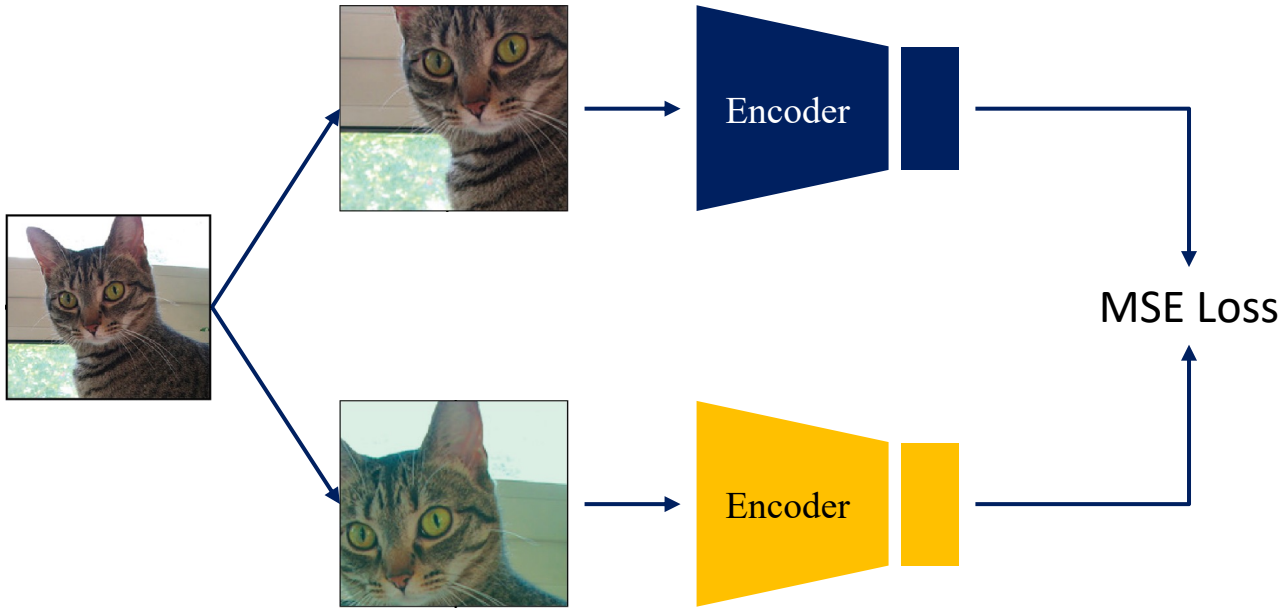
A simple framework for contrastive learning of visual representations, ICML 2020

MoCL: Data-driven Molecular Fingerprint via Knowledge-aware Contrastive Learning from Molecular Graph, KDD 2020

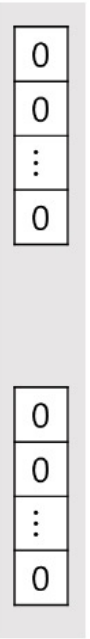
Can We Remove Negative Sampling?

- **Can we train cross-view prediction framework without negative samples?**

- **Problem:** Predicting directly in representation space can lead to collapsed representation
  - Contrastive methods circumvents this by reformulating the prediction problem as discrimination task (Pos ↔ Neg)



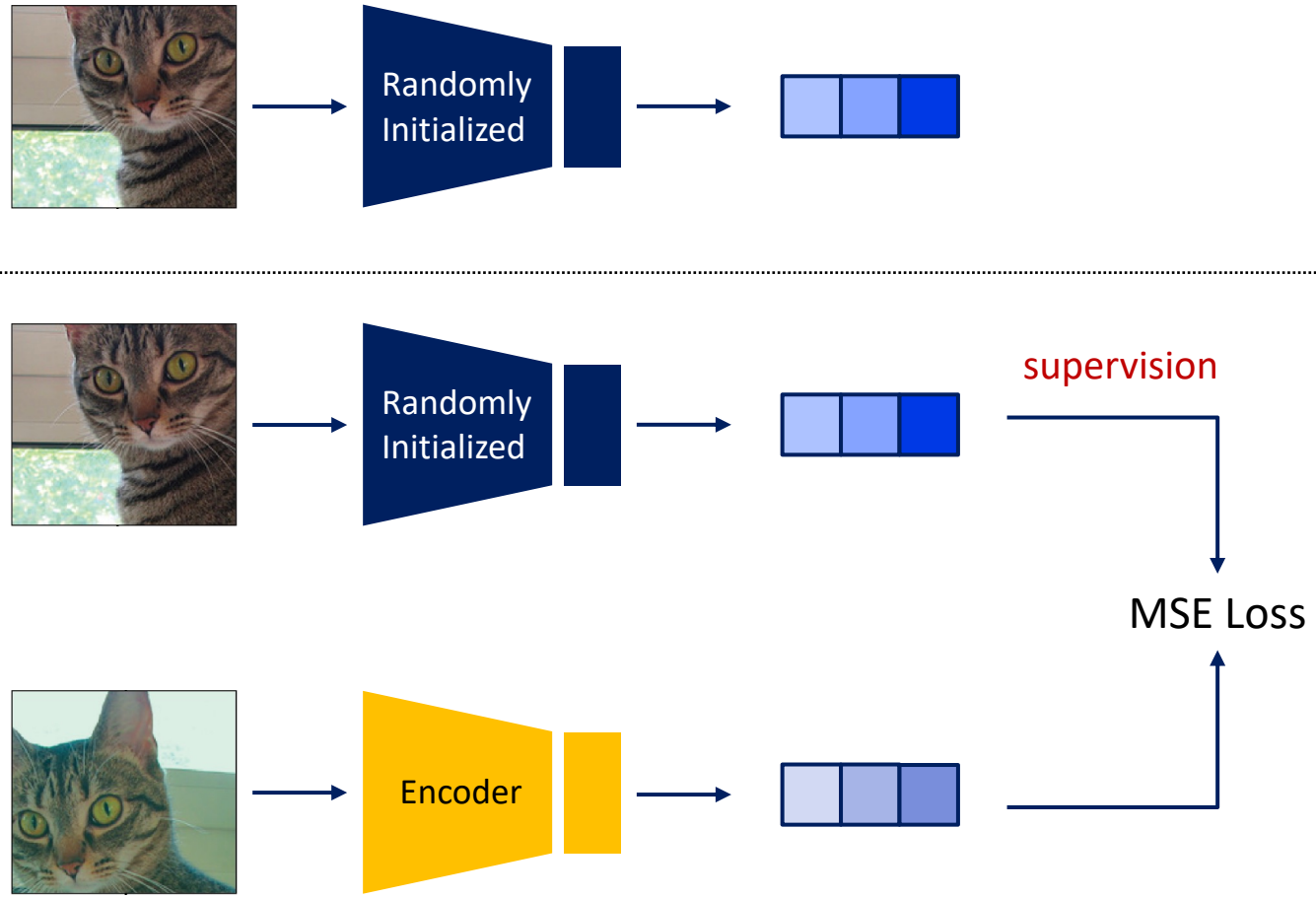
Cross-view prediction framework



Trivial Solution → Constant Vector

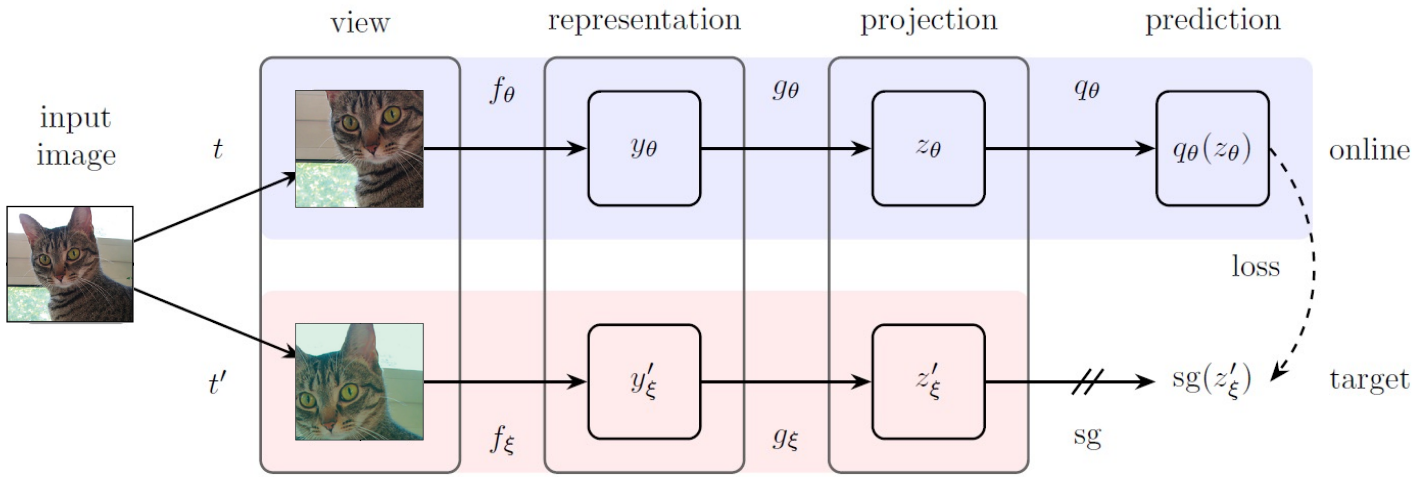
Straightforward Solution to Overcome Collapsed Representation

- **Use a fixed randomly initialized network to produce targets for our predictions**



Bootstrap Your Own Latent (BYOL)

- BYOL uses two neural networks to learn: 1) online and 2) target networks
- From a given target representation, we train a new online representation by predicting the target representation



Only online parameters are updated to reduce the loss, while the target parameters follow a different objective

→ Avoid Collapsed Representation

1) Online Network Update → Gradient-based update

$$\mathcal{L}_{\theta,\xi} \triangleq \left\| \overline{q_\theta}(z_\theta) - \overline{z}'_\xi \right\|_2^2, \qquad \mathcal{L}^{\text{BYOL}}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \widetilde{\mathcal{L}}_{\theta,\xi} \text{ (Symmetrize)}$$

$$\theta \leftarrow \text{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\text{BYOL}}_{\theta,\xi}, \eta\right)$$

Online network

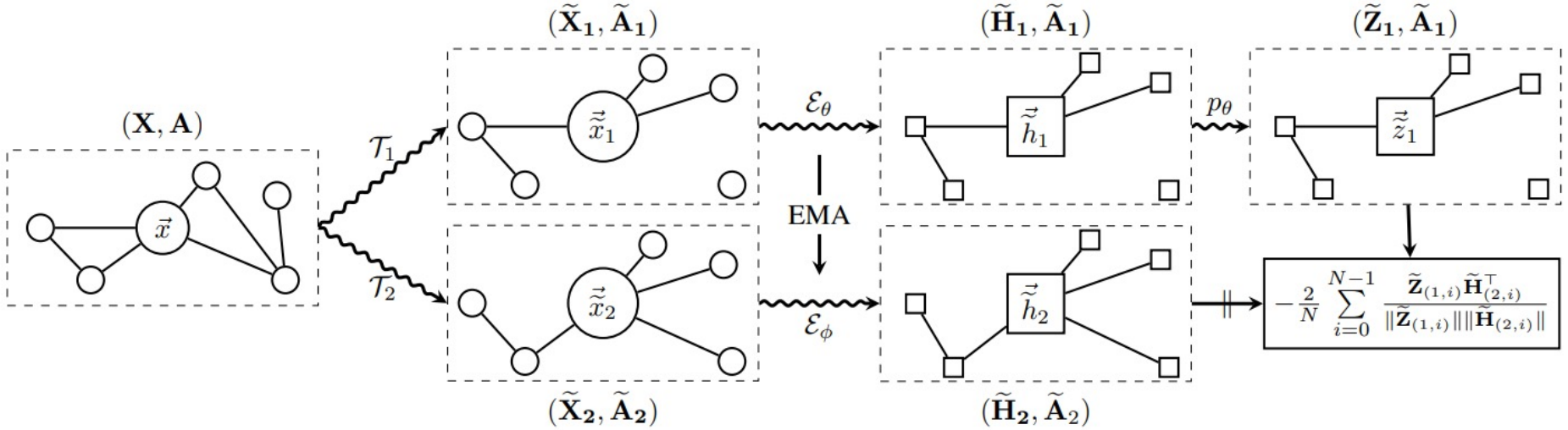
2) Target Network Update → Exponential Moving Average

$$\xi \leftarrow \tau\xi + (1-\tau)\theta$$

Target network Online network

Large-Scale Representation Learning on Graphs via Bootstrapping

- BGRL is a simple extension of BYOL to graph domain → 2nd place solution in KDD cup 2021
- Representations are directly learned by **predicting the representation of each node in one view of the graph, using the representation of the same node in another view**



- Graph Augmentation → Node attribute masking + Edge masking

Shortcomings of Contrastive Methods

- 1) Requires negative samples → **Sampling bias**
  - Treat different image as negative even if they share the semantics
- 2) Requires careful augmentation

Research Question
Is augmentation appropriate for graph-structured data?

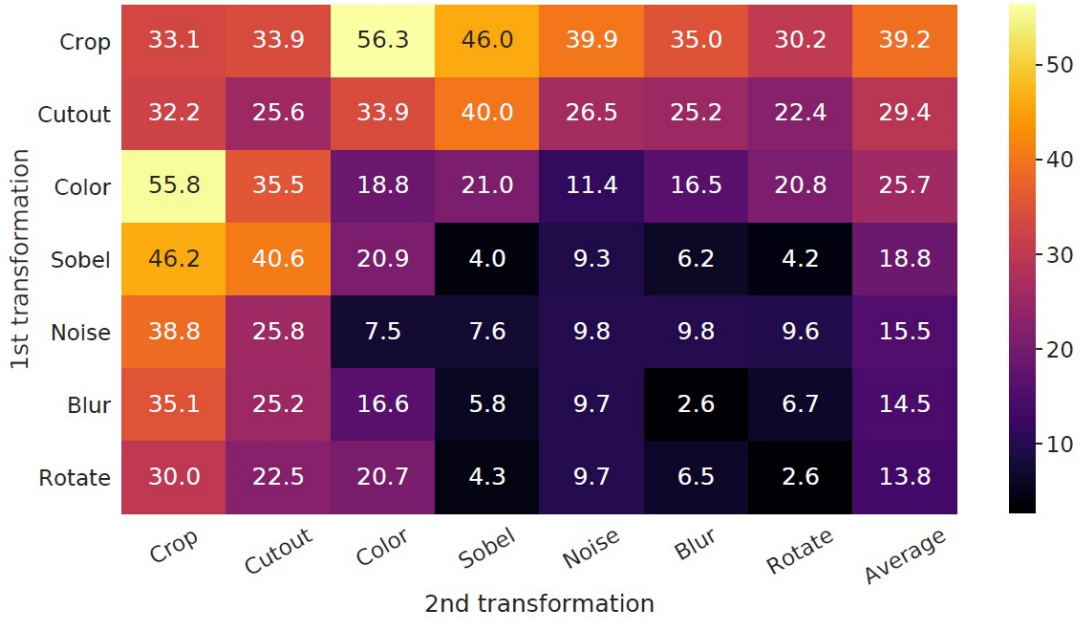
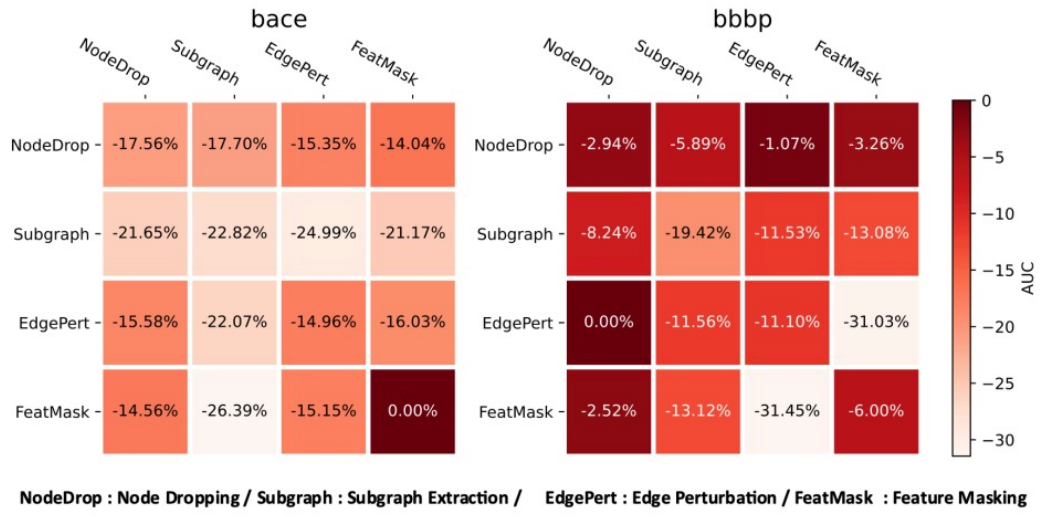


Image classification



Graph classification

A simple framework for contrastive learning of visual representations, ICML 2020
MoCL: Data-driven Molecular Fingerprint via Knowledge-aware Contrastive Learning from Molecular Graph, KDD 2020

KAIST

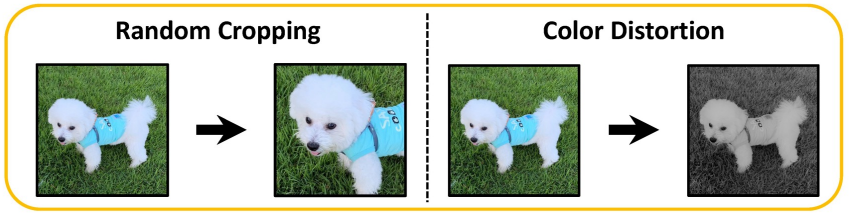
Augmentation-Free Self-Supervised Learning on Graphs

Published in AAAI'22

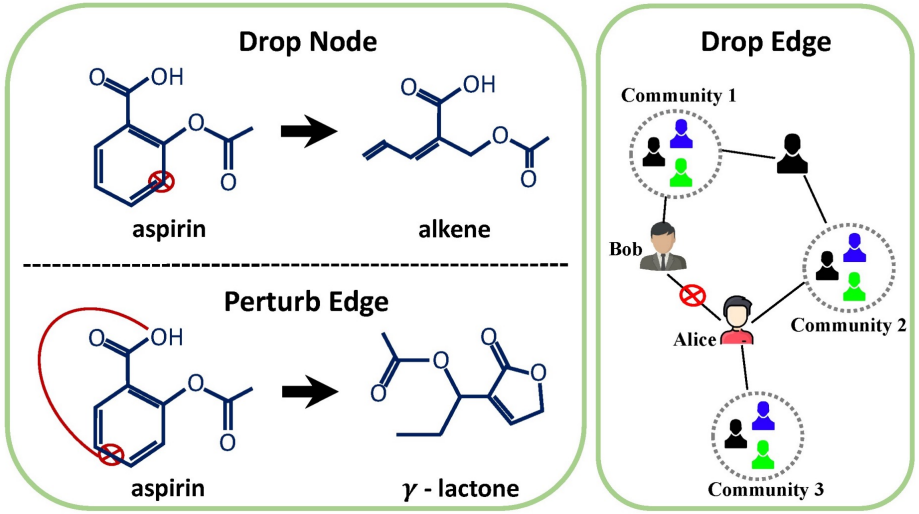
Namkyeong Lee, Junseok Lee, Chanyoung Park

Motivation: Is Augmentation Appropriate for Graph-structured Data?

- Image's underlying semantic is hardly changed after augmentation



- However in the case of graphs, we cannot ascertain whether the augmented graph would be positively related to original graph



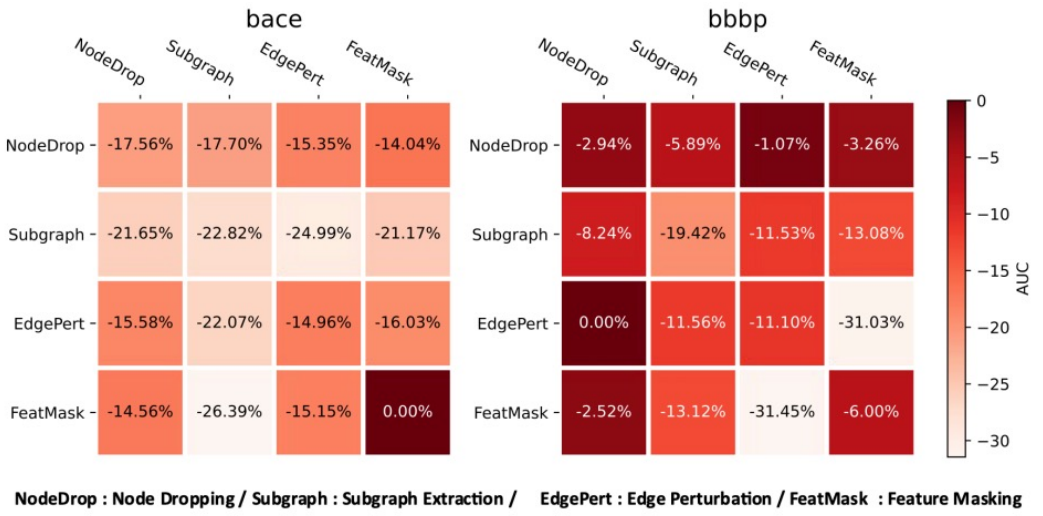
Because graphs contain not only the semantic but also the **structural information**

Motivation: Is Augmentation Appropriate for Graph-structured Data?

- Performance sensitivity according to hyperparameters for augmentations

		Comp.	Photo	CS	Physics
Node Classi.	BGRL	-4.00%	-1.06%	-0.20%	-0.69%
	GCA	-19.18%	-5.48%	-0.27%	OOM
Node Clust.	BGRL	-11.57%	-13.30%	-0.78%	-6.46%
	GCA	-26.28%	-23.27%	-1.64%	OOM

Node-level task



Graph-level task

- The quality of the learned representations relies on the **choice of augmentation scheme**
  - Performance on various downstream tasks varies greatly according to the choice of augmentation hyperparameters

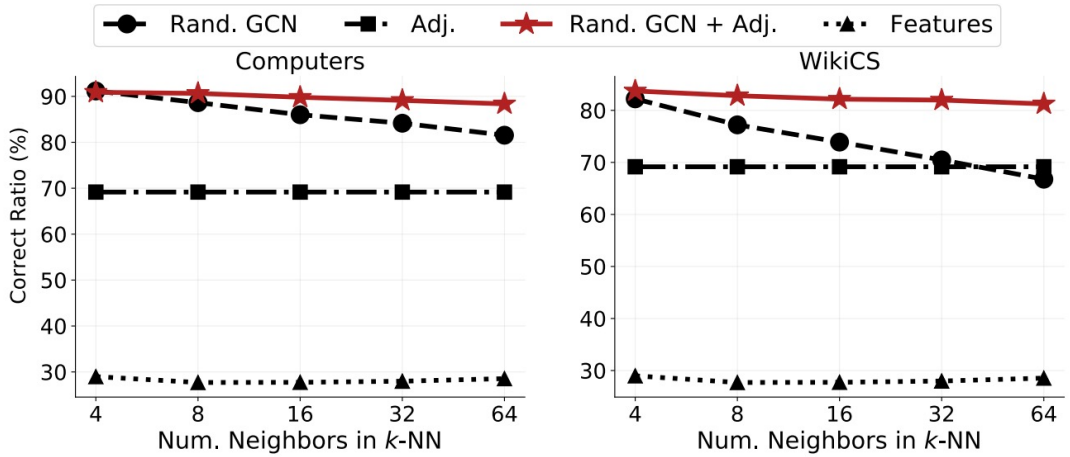
We need more stable and general framework for generating alternative view of the original graph **without relying on augmentation**

+ remove negative sampling process

Augmentation-Free Graph Representation Learning

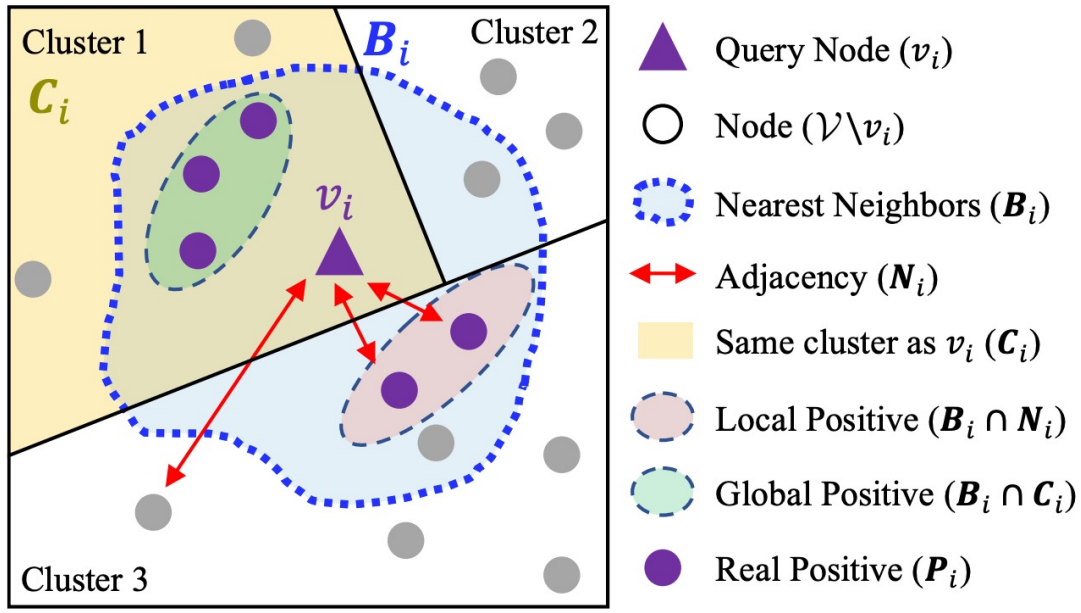
- Instead of creating two arbitrarily augmented views of a graph,
  - Use the original graph as-is as one view, and generate another view by discovering nodes that can serve as positive samples via **k-nearest neighbor search in embedding space**
- However, naively selected positive samples with k-NN includes false positives
  - More than 10% of false negatives

% of same label among neighbors



We need to filter out false positives regarding local and global perspective!

Capturing Local and Global Semantics



- B_i: Set of k-NNs of query v_i
- N_i: Set of adjacent nodes of query v_i
- C_i: Set of nodes that are in the same cluster with query v_i

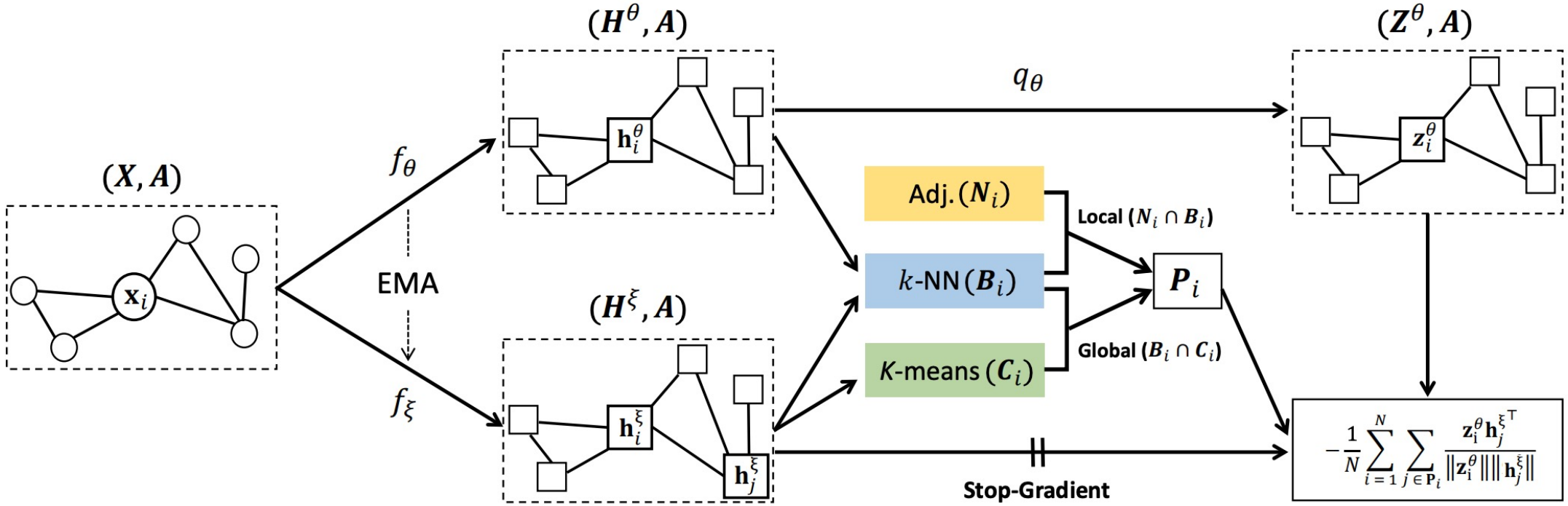
- Obtain real positives for v_i

$$\mathbf{P}_i = (\mathbf{B}_i \cap \mathbf{N}_i) \cup (\mathbf{B}_i \cap \mathbf{C}_i)$$

- Minimize the cosine distance between query and real positives P_i

$$\mathcal{L}_{\theta,\xi} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{v_j \in \mathbf{P}_i} \frac{\mathbf{z}_i^{\theta}\mathbf{h}_j^{\xi\top}}{\left\|\mathbf{z}_i^{\theta}\right\|\left\|\mathbf{h}_j^{\xi}\right\|}$$

Overall Architecture of AFGRL



Experiments

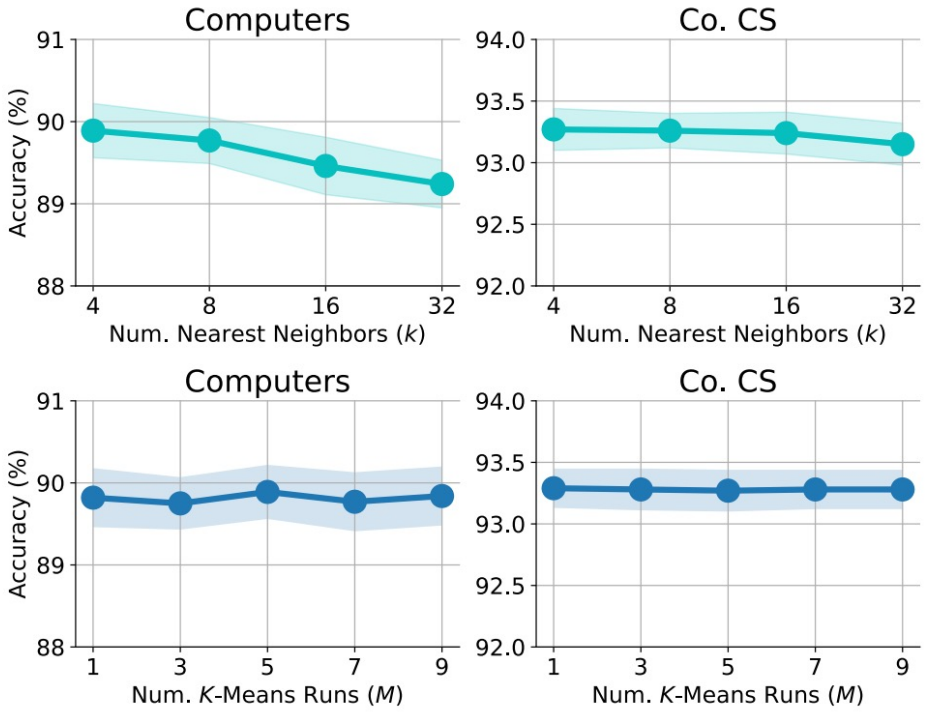
- **Task**: Node classification

	WikiCS	Computers	Photo	Co.CS	Co.Physics
Sup. GCN	77.19 ± 0.12	86.51 ± 0.54	92.42 ± 0.22	93.03 ± 0.31	95.65 ± 0.16
Raw feats.	71.98 ± 0.00	73.81 ± 0.00	78.53 ± 0.00	90.37 ± 0.00	93.58 ± 0.00
node2vec	71.79 ± 0.05	84.39 ± 0.08	89.67 ± 0.12	85.08 ± 0.03	91.19 ± 0.04
DeepWalk	74.35 ± 0.06	85.68 ± 0.06	89.44 ± 0.11	84.61 ± 0.22	91.77 ± 0.15
DW + feats.	77.21 ± 0.03	86.28 ± 0.07	90.05 ± 0.08	87.70 ± 0.04	94.90 ± 0.09
DGI	75.35 ± 0.14	83.95 ± 0.47	91.61 ± 0.22	92.15 ± 0.63	94.51 ± 0.52
GMI	74.85 ± 0.08	82.21 ± 0.31	90.68 ± 0.17	OOM	OOM
MVGRL	77.52 ± 0.08	87.52 ± 0.11	91.74 ± 0.07	92.11 ± 0.12	95.33 ± 0.03
GRACE	**77.97 ± 0.63**	86.50 ± 0.33	92.46 ± 0.18	92.17 ± 0.04	OOM
GCA	77.94 ± 0.67	87.32 ± 0.50	92.39 ± 0.33	92.84 ± 0.15	OOM
BGRL	76.86 ± 0.74	89.69 ± 0.37	93.07 ± 0.38	92.59 ± 0.14	95.48 ± 0.08
AFGRL	77.62 ± 0.49	**89.88 ± 0.33**	**93.22 ± 0.28**	**93.27 ± 0.17**	**95.69 ± 0.10**

Recall…

		Comp.	Photo	CS	Physics
Node Classi.	BGRL	-4.00%	-1.06%	-0.20%	-0.69%
	GCA	-19.18%	-5.48%	-0.27%	OOM
Node Clust.	BGRL	-11.57%	-13.30%	-0.78%	-6.46%
	GCA	-26.28%	-23.27%	-1.64%	OOM

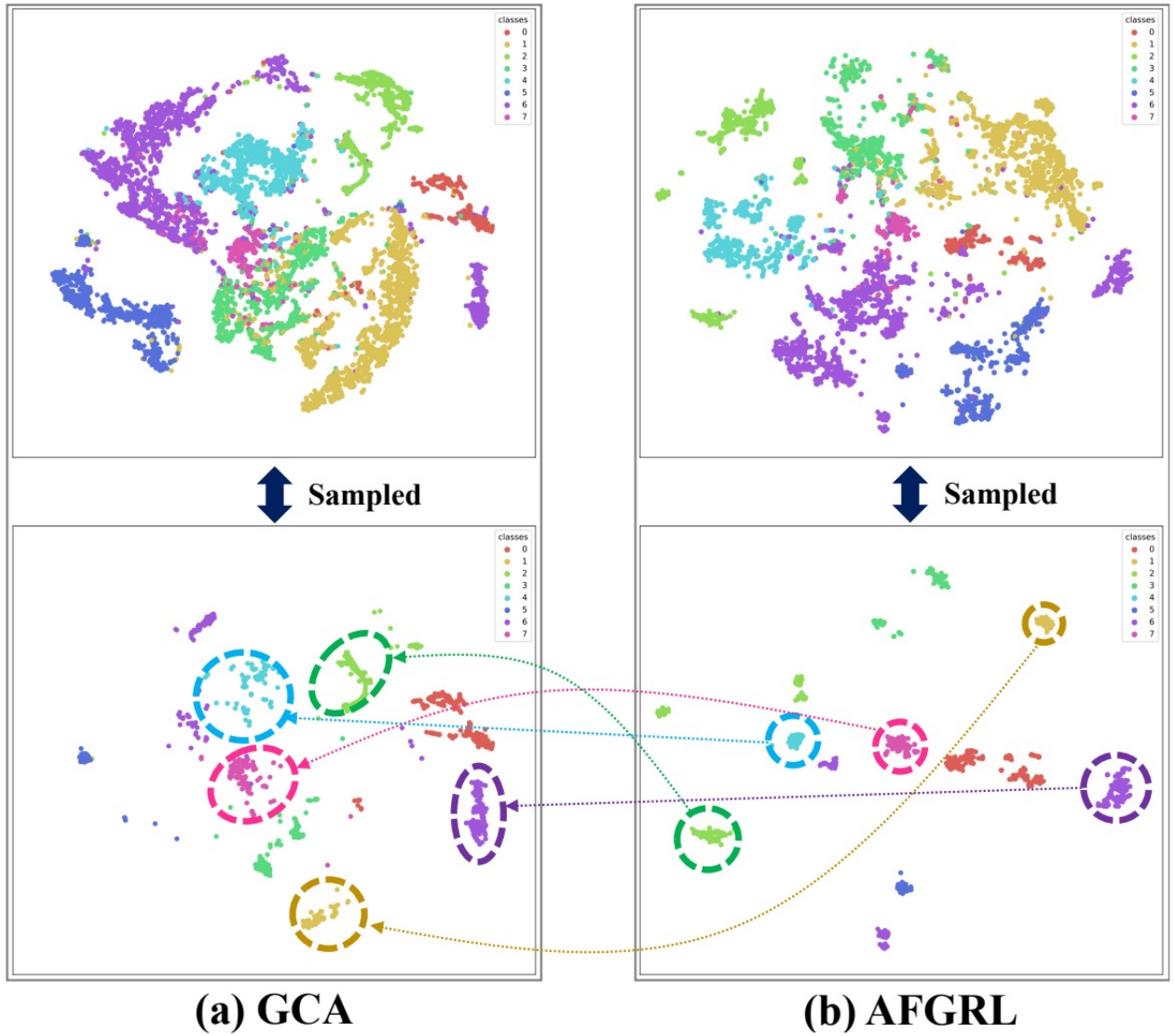
AFGRL outperforms SOTA baselines



AFGRL is stable over hyperparameters
→ Can be easily trained compared with other augmentation-based methods.

Experiments

- **Task**: T-SNE visualization



Nodes are more tightly grouped in AFGRL
→ Captures fine-grained class information

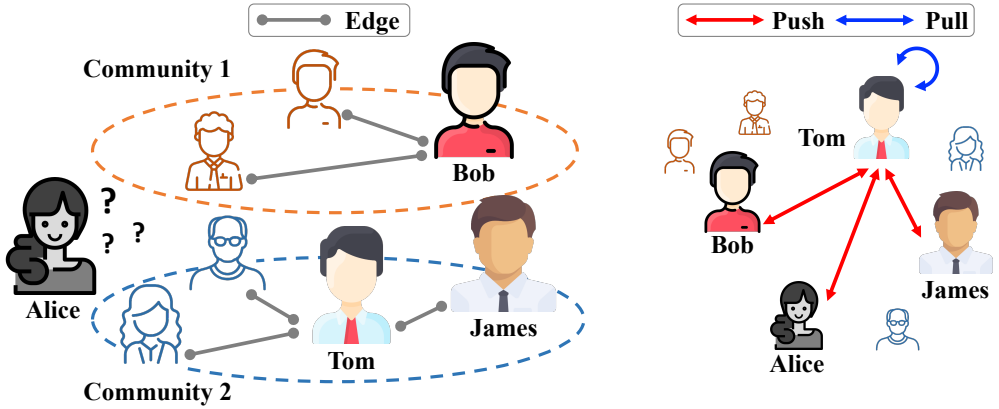
Relational Self-Supervised Learning on Graphs

Published in CIKM'22

Namkyeong Lee, Dongmin Hyun, Junseok Lee, Chanyoung Park

Motivation: Graphs exhibit relational information

- Recent graph representation learning methods do not reflect the **nature of the graph**
  - *Graphs exhibit **relational information among nodes***

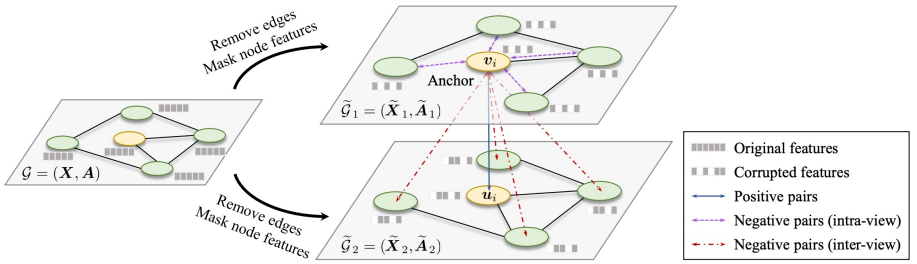


(a) Graph-Structured Data (b) Recent GRL methods

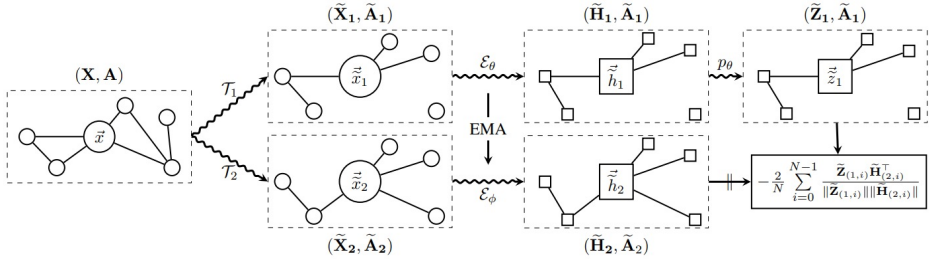
Previous methods (contrastive & non-contrastive) cannot fully benefit from relational information of graph structured data

→ **They learn augmentation-invariant node representation**

Contrastive methods (GRACE, GCA)



Non-contrastive methods (BGRL)

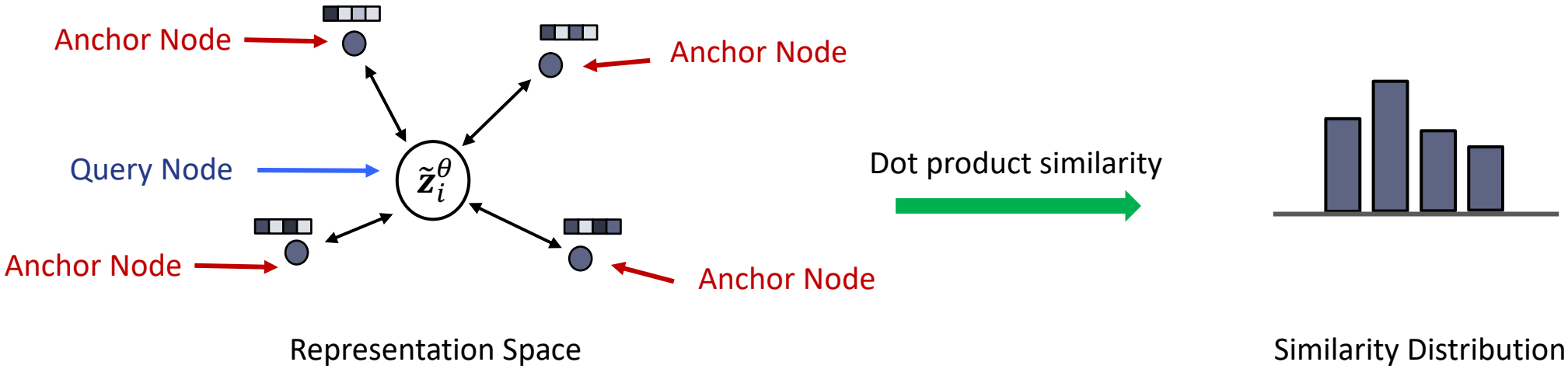


Research question: How can we consider the relational information among nodes?

→ Augmentation-invariant relational information

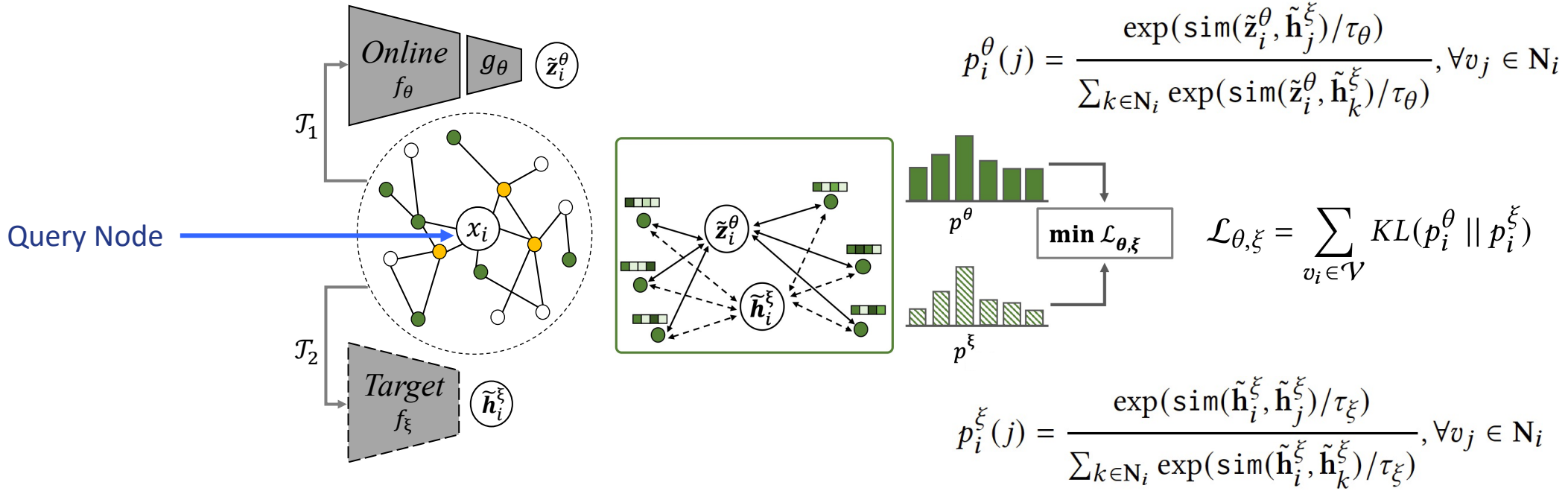
Learning Augmentation-invariant Relationship on Graphs

- How can we define relationship between a query node and anchor nodes?



We define cosine similarity as relationship between a query node and anchor nodes

A Simple Approach



Online network is trained to mimic the relational information captured by target network

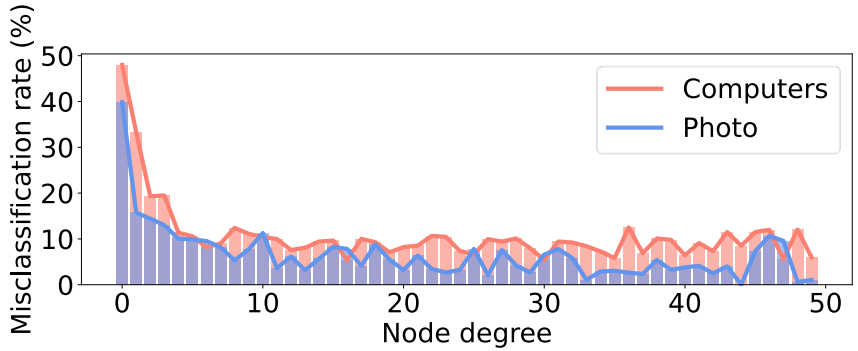
→ **Learning augmentation-invariant relationship!** (Instead of augmentation-invariant node representation)

Next research question: How to sample anchor nodes?

Diverse relational information regarding both global and local perspectives should be considered

Capturing Global Similarity: Sampling Global Anchor Nodes

- Global anchor nodes: Structurally distant nodes



Misclassification rate for certain degree of nodes

Misclassification rate of low-degree nodes is significantly high
→ *Degree-bias* issue!

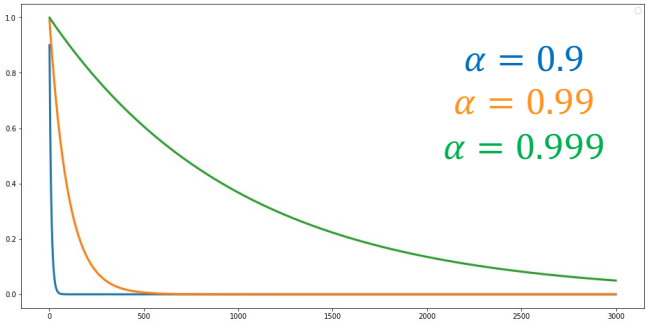
We should focus on low-degree nodes while training RGRL

- Approach: Sample anchor nodes from ***inverse degree-weighted distribution***

→ Sample more from low-degree nodes

$$w_j = \alpha^{\log(\deg_j+1)} + \beta$$

$0 < \alpha < 1$



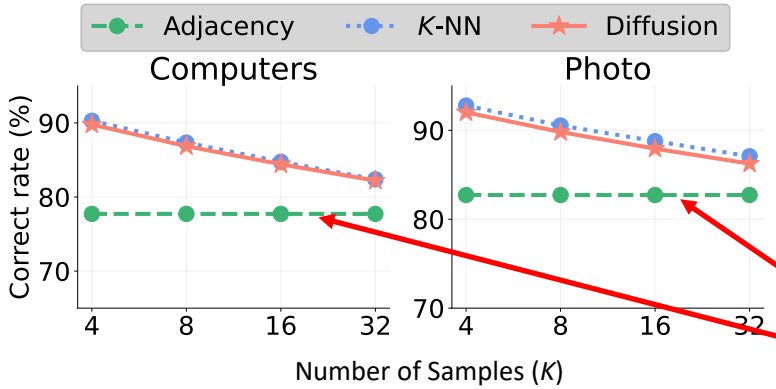
Setting $0 < \alpha < 1$ approximates the misclassification rate

$$p_{sample}(j) = \frac{w_j}{\sum_{v_k \in \mathcal{V}} w_k}, \forall v_j \in \mathcal{V}$$

Inverse degree-weighted distribution

Capturing Local Similarity: Sampling Local Anchor Nodes

- Local anchor nodes: Structurally close nodes



Ratio of its neighboring nodes being the same label

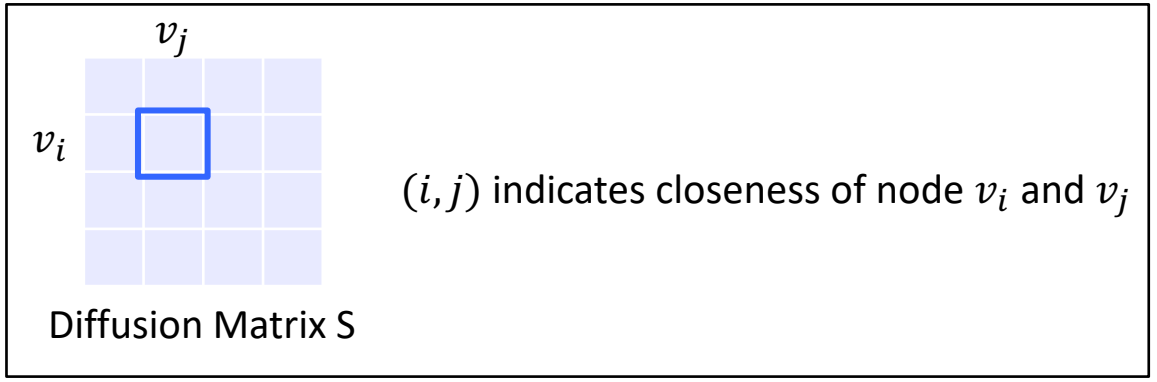
- Adjacency may fail to capture fine-grained relationship among nodes
  - ex) “Data Mining ” vs. “Machine Learning” community
    - Structurally close but different class
- **We should sample anchor nodes that are**
  - 1) Structurally close with query node in the graph structure
  - 2) Share the same label with the query node

- Approach: Sample anchor nodes based on ***diffusion score matrix (Personalized PageRank)***

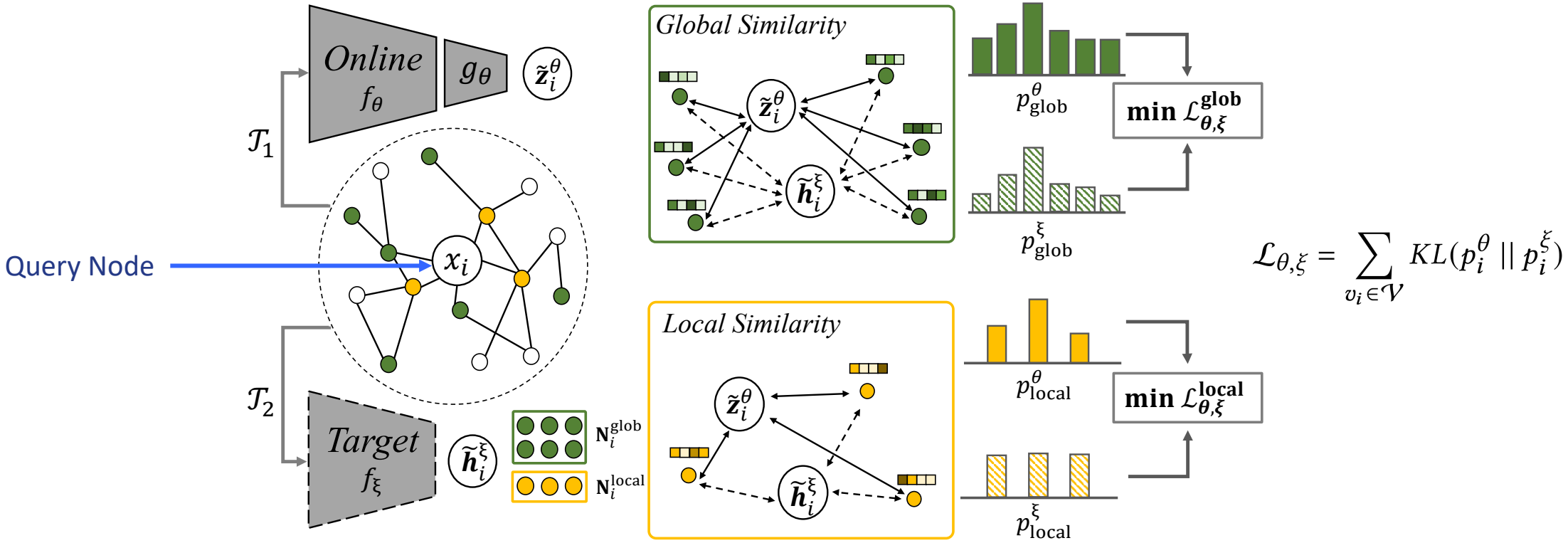
$$\mathrm{S} = \sum_{k=0}^{\infty} t(1-t)^k \mathbf{T}^k$$

t : Teleport probability ($t \in (0,1)$)

$\mathbf{T}$: Symmetric transition matrix



Proposed Method: RGRL (Overview)



$$\mathcal{L}_{\theta,\xi} = \sum_{v_i \in \mathcal{V}} KL(p_i^\theta \,||\, p_i^\xi)$$

$$p_i^\theta(j) = \frac{\exp(\text{sim}(\tilde{\mathbf{z}}_i^\theta, \tilde{\mathbf{h}}_j^\xi)/\tau_\theta)}{\sum_{k\in\mathbf{N}_i} \exp(\text{sim}(\tilde{\mathbf{z}}_i^\theta, \tilde{\mathbf{h}}_k^\xi)/\tau_\theta)}, \forall v_j \in \mathbf{N}_i$$

(Relational information regarding **online network**)

$$p_i^\xi(j) = \frac{\exp(\text{sim}(\tilde{\mathbf{h}}_i^\xi, \tilde{\mathbf{h}}_j^\xi)/\tau_\xi)}{\sum_{k\in\mathbf{N}_i} \exp(\text{sim}(\tilde{\mathbf{h}}_i^\xi, \tilde{\mathbf{h}}_k^\xi)/\tau_\xi)}, \forall v_j \in \mathbf{N}_i$$

(Relational information regarding **target network**)

Discussion: How RGRL overcomes limitations of previous works?

- **Previous works**: **1) Contrastive methods**, **2) Non-contrastive methods**
- 1) Limitation of **Contrastive methods**
  - Sampling bias: Simply treating all other nodes as negatives incurs false negatives
  - Another problem occurs when sampling bias is combined with the **contrastive loss** that is defined as follows [1]:

Positive pair

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Negative pair

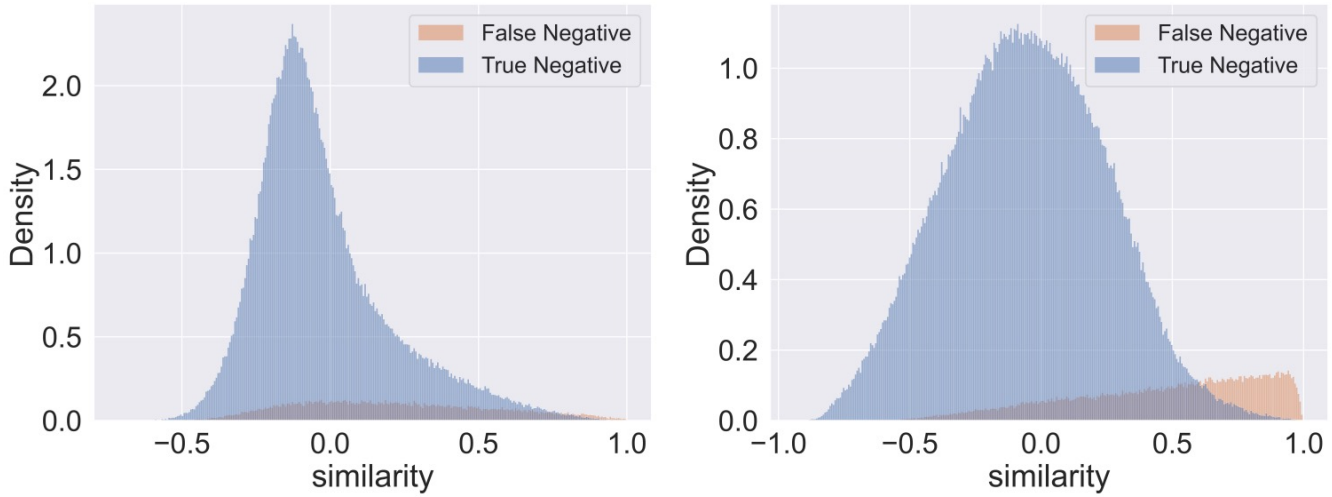
- As τ decreases, the model gives larger penalty to hard negative samples (push away)
  - Makes sense if we know true negatives (supervised setting)
  - But, harmful in self-supervised learning where false negatives exist

- Contrastive loss is “Hardness-aware loss”
  - Gives larger penalties to similar nodes → similar nodes that belong to negative samples become more dissimilar
- **Thus, false negative is trained to be more dissimilar**

[1] Understanding the behaviour of contrastive loss, CVPR 2021

Discussion: How RGRL overcomes limitations of previous works?

- **The problem gets even more severe in graph domain,**
  - In graphs, most “HARD” negatives are indeed “FALSE” negatives



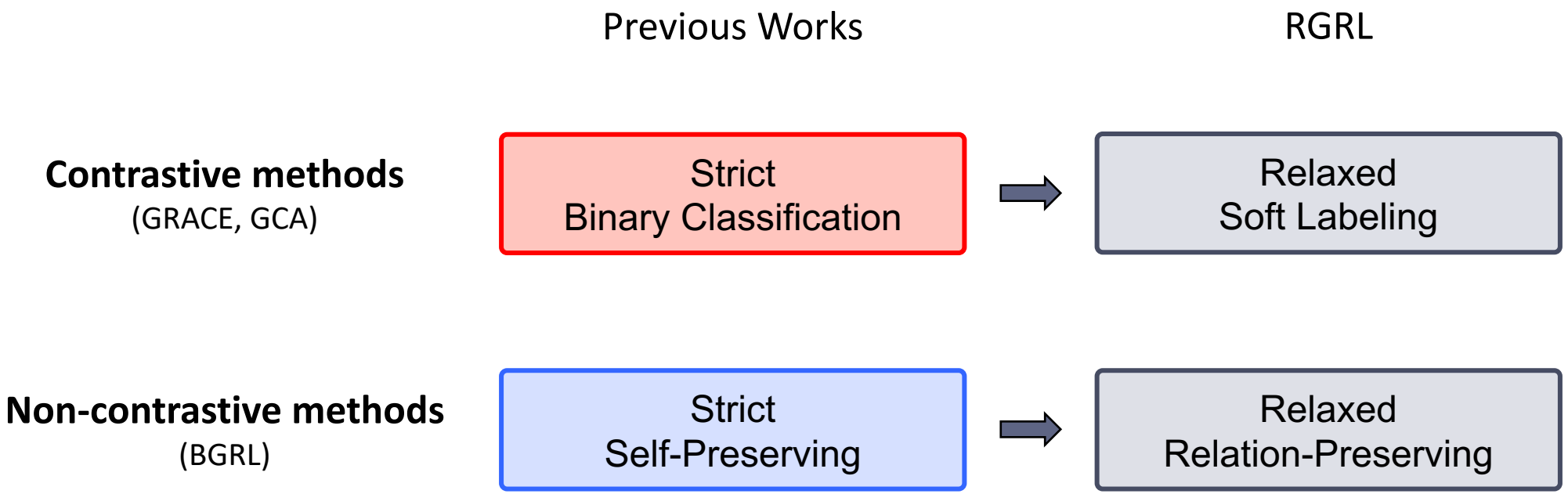
(a) CIFAR-10 (Image) (b) Coauthor-CS (Graph)

- RGRL relaxes the strict binary classification of contrastive methods with **soft labeling**
  - RGRL can decide how much to push or pull other nodes based on the relational information among the nodes with out relying on the binary decisions of positives and negatives

Discussion: How RGRL overcomes limitations of previous works?

- **Previous works**: **1) Contrastive methods**, **2) Non-contrastive methods**
- 2) Limitation of **Non-contrastive methods**
  - Since we don't use any negative samples, **node features should be fully informative**
    - Performance actually degrades if features contain noise (as will be shown later)
    - Overfit to a few non-informative feature
- RGRL alleviates the overfitting problem with a little help from other nodes in the graph
  - Learn from the relationship with other nodes
- RGRL relaxes the strict self-preserving loss with **relation-preserving loss**
  - Allows the representations to vary as long as the **relationship among the representations is preserve**

Summary: How RGRL overcomes limitations of previous works?



RGRL achieves the best of both worlds by relaxing strict constraints of previous works

Experiments: Node Classification

	WikiCS	Computers	Photo	Co.CS	Co.Physics
GCN	77.19 (0.12)	86.51 (0.54)	92.42 (0.22)	93.03 (0.31)	95.65 (0.16)
Feats.	71.98 (0.00)	73.81 (0.00)	78.53 (0.00)	90.37 (0.00)	93.58 (0.00)
n2v	71.79 (0.05)	84.39 (0.08)	89.67 (0.12)	85.08 (0.03)	91.19 (0.04)
DW	74.35 (0.06)	85.68 (0.06)	89.44 (0.11)	84.61 (0.22)	91.77 (0.15)
DW+Feats.	77.21 (0.03)	86.28 (0.07)	90.05 (0.08)	87.70 (0.04)	94.90 (0.09)
DGI	75.35 (0.14)	83.95 (0.47)	91.61 (0.22)	92.15 (0.63)	94.51 (0.52)
GMI	74.85 (0.08)	82.21 (0.31)	90.68 (0.17)	OOM	OOM
MVGRL	77.52 (0.08)	87.52 (0.11)	91.74 (0.07)	92.11 (0.12)	95.33 (0.03)
GRACE	78.25 (0.65)	88.15 (0.43)	92.52 (0.32)	92.60 (0.11)	OOM
GCA	78.30 (0.62)	88.49 (0.51)	92.99 (0.27)	92.76 (0.16)	OOM
CCA-SSG	77.88 (0.41)	87.01 (0.41)	92.59 (0.25)	92.77 (0.17)	95.16 (0.10)
BGRL	79.60 (0.60)	89.23 (0.34)	93.06 (0.30)	92.90 (0.15)	95.43 (0.09)
RGRL	**80.29 (0.72)**	**89.70 (0.44)**	**93.43 (0.31)**	**92.94 (0.13)**	**95.46 (0.10)**

Performance on node classification tasks

	Transductive						Inductive	
	Cora	Cite-seer	Pub-med	Cora Full	ogbn-arXiv Valid	ogbn-arXiv Test	Reddit	PPI
GRACE	83.38 (0.95)	70.79 (0.83)	83.96 (0.29)	64.19 (0.36)	OOM	OOM	94.84 (0.03)	67.12 (0.05)
GCA	82.79 (1.01)	70.70 (0.91)	84.19 (0.32)	64.34 (0.42)	OOM	OOM	94.85 (0.06)	66.72 (0.08)
CCA-SSG	83.01 (0.66)	70.35 (1.23)	84.81 (0.22)	64.09 (0.37)	59.43 (0.05)	58.50 (0.08)	94.89 (0.02)	66.09 (0.01)
BGRL	82.82 (0.86)	69.06 (0.80)	**86.16 (0.19)**	63.94 (0.39)	70.66 (0.06)	69.61 (0.09)	94.90 (0.04)	68.89 (0.08)
RGRL	**83.98 (0.78)**	**71.29 (0.87)**	85.33 (0.20)	**64.62 (0.39)**	**72.34 (0.09)**	**71.49 (0.08)**	**95.04 (0.03)**	**69.28 (0.06)**

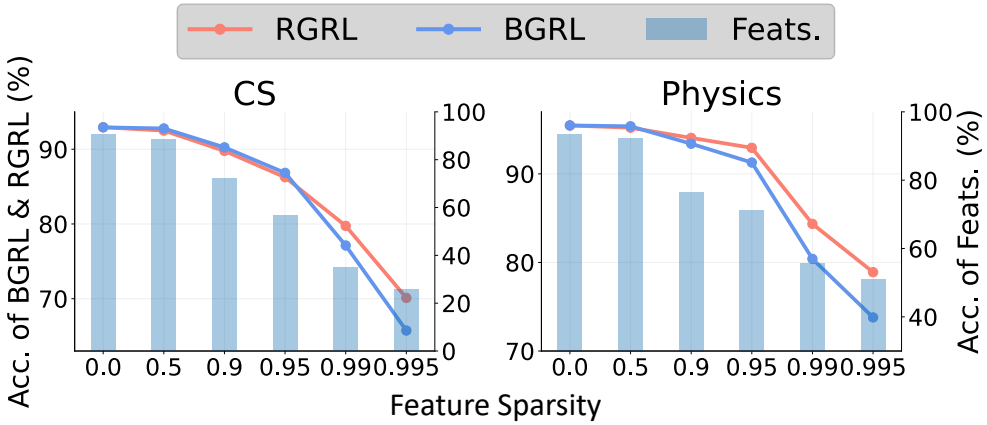
Performance on various datasets (transductive/inductive)

RGRL outperforms previous methods that overlook the relationship among nodes

Experiments: Node Classification

	WikiCS	Computers	Photo	Co.CS	Co.Physics
GCN	77.19 (0.12)	86.51 (0.54)	92.42 (0.22)	93.03 (0.31)	95.65 (0.16)
Feats.	71.98 (0.00)	73.81 (0.00)	78.53 (0.00)	90.37 (0.00)	93.58 (0.00)
n2v	71.79 (0.05)	84.39 (0.08)	89.67 (0.12)	85.08 (0.03)	91.19 (0.04)
DW	74.35 (0.06)	85.68 (0.06)	89.44 (0.11)	84.61 (0.22)	91.77 (0.15)
DW+Feats.	77.21 (0.03)	86.28 (0.07)	90.05 (0.08)	87.70 (0.04)	94.90 (0.09)
DGI	75.35 (0.14)	83.95 (0.47)	91.61 (0.22)	92.15 (0.63)	94.51 (0.52)
GMI	74.85 (0.08)	82.21 (0.31)	90.68 (0.17)	OOM	OOM
MVGRL	77.52 (0.08)	87.52 (0.11)	91.74 (0.07)	92.11 (0.12)	95.33 (0.03)
GRACE	78.25 (0.65)	88.15 (0.43)	92.52 (0.32)	92.60 (0.11)	OOM
GCA	78.30 (0.62)	88.49 (0.51)	92.99 (0.27)	92.76 (0.16)	OOM
CCA-SSG	77.88 (0.41)	87.01 (0.41)	92.59 (0.25)	92.77 (0.17)	95.16 (0.10)
BGRL	79.60 (0.60)	89.23 (0.34)	93.06 (0.30)	92.90 (0.15)	95.43 (0.09)
RGRL	**80.29 (0.72)**	**89.70 (0.44)**	**93.43 (0.31)**	**92.94 (0.13)**	**95.46 (0.10)**

Performance on node classification tasks



Classification accuracy over feature sparsity

Dataset with less informative features

→ Large improvements in performance

→ External self-supervisory signals from other nodes help RGRL to learn from less informative features

Dataset with more informative features

→ RGRL is more robust than BGRL as the quality of input features get worse

Experiments: Qualitative Analysis

Query Author	Model	Top-1 Similar Author	# Co-authored Papers	Student?
Jiawei Han	BGRL	Ke Wang	14	✗
	RGRL	Xifeng Yan	87	✓
Christos Faloutsos	BGRL	Tina Eliassi-Rad	27	✗
	RGRL	Hanghang Tong	47	✓

Case 1) Which author is the most similar?

- RGRL discovers author who have more co-authored papers
- RGRL discovers former Ph.D. students of the query author
  - Advisor-advisee relationship
    → Core relationship in the academia network!

Query Author	Model	Top-1 Similar Author	# Co-authored Papers	Research Keywords
Jiawei Han	BGRL	Zhou Aoying	0	Query Processing
	RGRL	Ee-Peng Lim	0	Data & Text Mining
Christos Faloutsos	BGRL	Michael J. Pazzani	0	Machine Learning
	RGRL	David Jensen	2	Machine Learning

Case 2) Which author will co-work in the future?

- RGRL discovers author of more relevant research area
- RGRL discovers author who actually co-authored in the past (but missing in data)

RGRL discovers core relationship and meaningful knowledge that is not revealed in the given graph

Thank you!

- **Today**: Self-supervised learning on Graphs
  - [AAAI'22] Augmentation-Free Self-Supervised Learning on Graphs → Augmentation-free
  - [CIKM'22] Relational Self-Supervised Learning on Graphs → Augmentation-invariant relationship

- Other topics that my group is working on:
  - Long-tail problem on graphs
  - GNN-based material (chemical) property prediction (Material science)
  - Adversarial robustness of GNN
  - Dealing with noisy features on graphs / Few-shot learning on graphs
  - Continual learning on graphs
  - Anomaly detection on graphs
  - GNN-based scRNA-seq clustering (Bioinformatics)
  - Scene graph generation
  - GNN-based material (chemical) property prediction (Material science)
  - Recommender system
  - …