# Deep Graph InfoMax

Veličković, Petar, et al. (2018)

2021.08.09
Presented by HYOJUN KIM

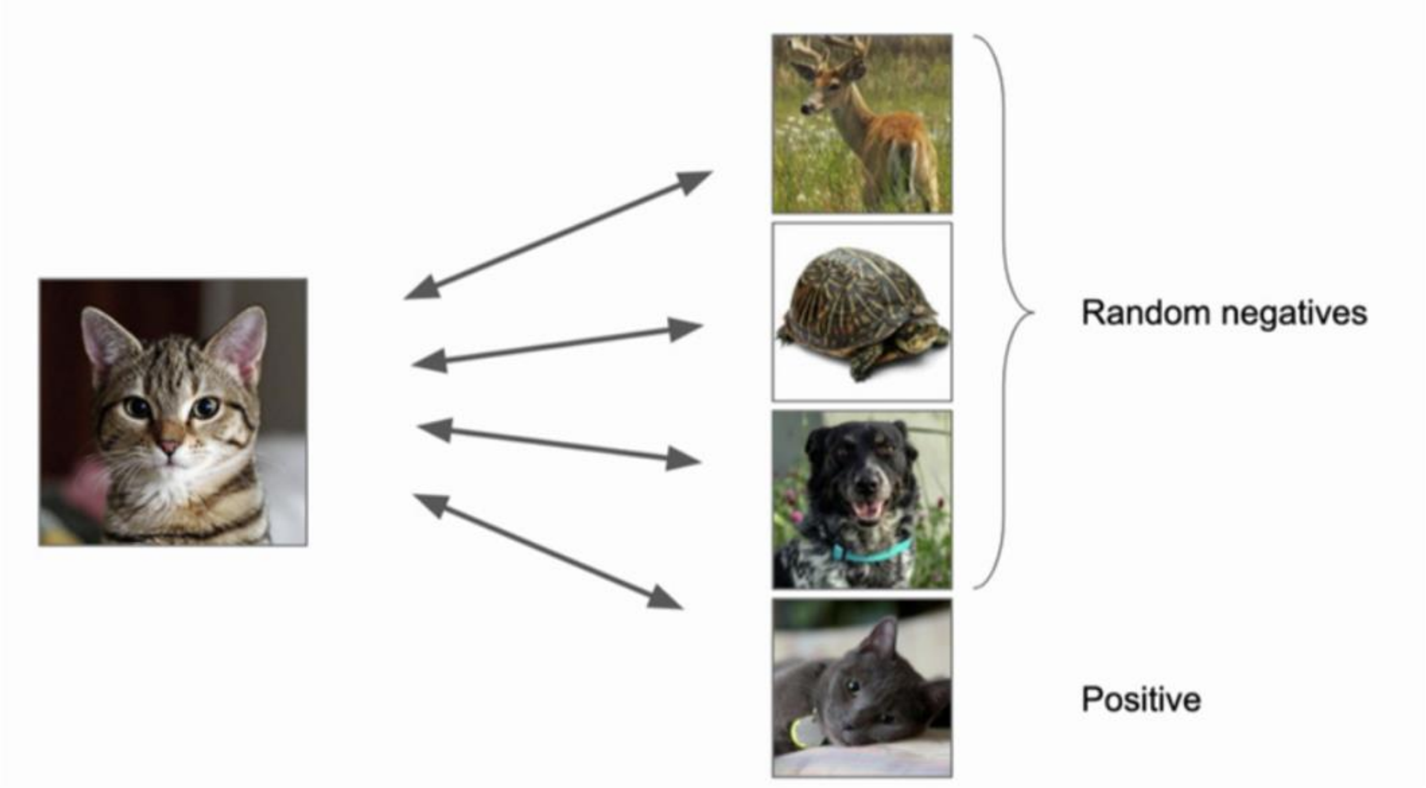
# CONTENTS

# 01 Background



- **Contrastive Self-Supervised Learning**

  - Positive example 과 Negative example을 대조하여 representation을 학습



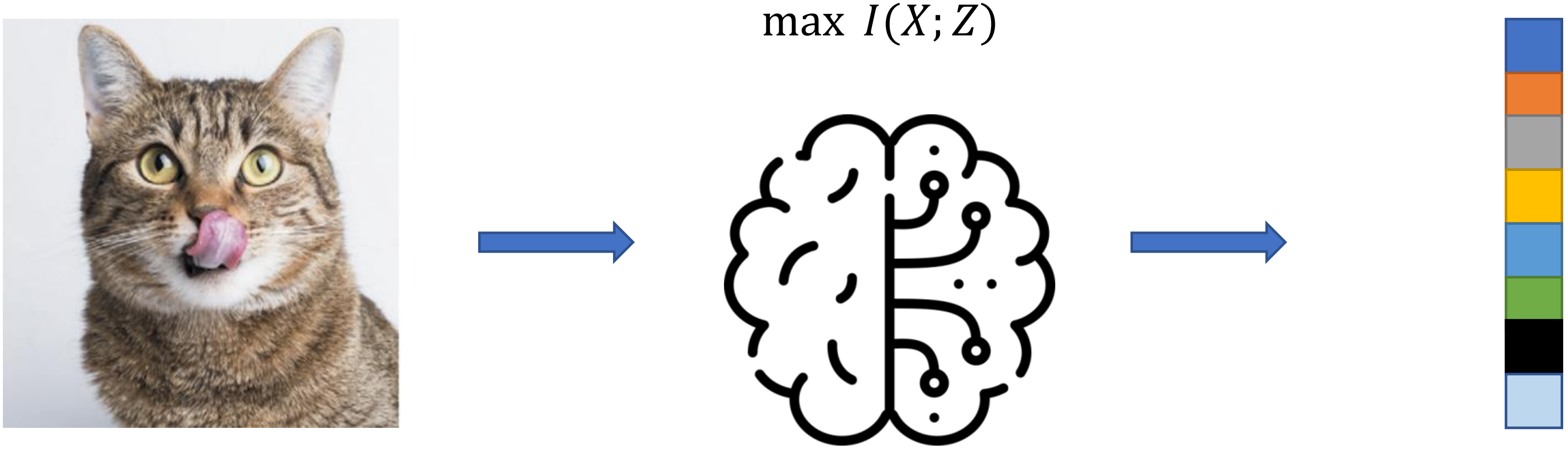

Mutual information - Contrastive (figure from Oord)

# 01 Background

- **Representation learning by InfoMax principle** $I(X; E(Z))$

  - InfoMax principle : maximizing the mutual information between input X and its representation Z

$\max\ I(X;Z)$



Input X

Representation Z

Hjelm, R. Devon, et al. "Learning deep representations by mutual information estimation and maximization." *arXiv preprint arXiv:1808.06670* (2018).

# 01 Background



## ▪ Mutual Information?

- 두 random variable들이 얼마나 dependence한지 measure하는 방법

$$I(X;Z) = D_{KL}(p(x,z)||p(x)p(z)) = \mathbb{E}_{p(x,z)}\left[\log \frac{p(x,z)}{p(x)p(z)}\right]$$

## ▪ Challenge in using InfoMax principle

- $I(X;E(Z)) = D_{KL}(p(x,z)||p(x)p(z))$ : Distribution을 알기가 어렵다.
- Exact computation of mutual information is mostly intractable

➔ Exact mutual information's lower bound를 Maximize 하자!

$$I(X;Z) \geq I_{\Theta}(X,Z),$$

Hjelm, R. Devon, et al. "Learning deep representations by mutual information estimation and maximization." *arXiv preprint arXiv:1808.06670* (2018).

- **Used 3 variational MI Estimators**

1. MINE : random variables을 샘플링 후, Donsker-Varadhan representation을 lower bound로 제안

$$\mathcal{I}(X;Y) := \mathcal{D}_{KL}(\mathbb{J}||\mathbb{M}) \geq \widehat{\mathcal{I}}_{\omega}^{(DV)}(X;Y) := \mathbb{E}_{\mathbb{J}}[T_{\omega}(x,y)] - \log \mathbb{E}_{\mathbb{M}}[e^{T_{\omega}(x,y)}],$$

2. Jensen-Shannon MI estimator

$$\widehat{\mathcal{I}}_{\omega,\psi}^{(\text{JSD})}(X;E_{\psi}(X)) := \mathbb{E}_{\mathbb{P}}[-\text{sp}(-T_{\psi,\omega}(x,E_{\psi}(x)))] - \mathbb{E}_{\mathbb{P}\times\tilde{\mathbb{P}}}[\text{sp}(T_{\psi,\omega}(x',E_{\psi}(x)))],$$

3. CPC : Noise Contrastive Estimation (infoNCE loss를 최소화하는 관점)

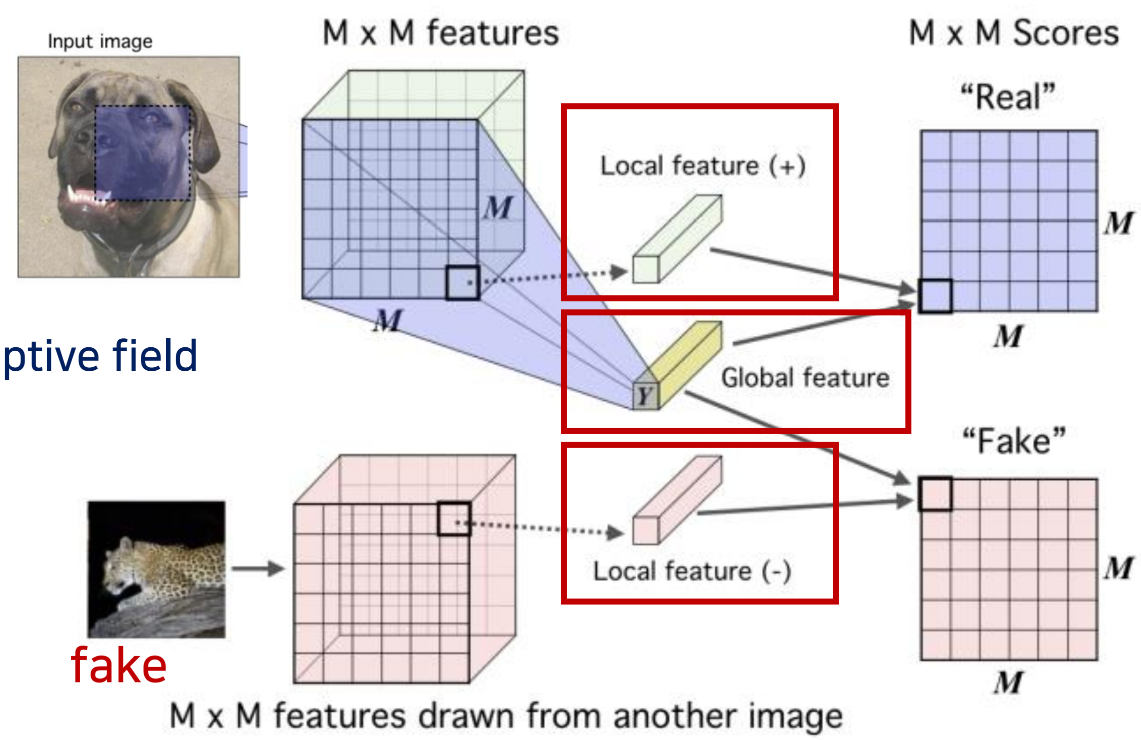
$$\widehat{\mathcal{I}}_{\omega,\psi}^{(\text{infoNCE})}(X;E_{\psi}(X)) := \mathbb{E}_{\mathbb{P}}\left[T_{\psi,\omega}(x,E_{\psi}(x)) - \mathbb{E}_{\tilde{\mathbb{P}}}\left[\log \sum_{x'} e^{T_{\psi,\omega}(x',E_{\psi}(x))}\right]\right].$$

Belghazi, Mohamed Ishmael, et al. "Mutual information neural estimation." *International Conference on Machine Learning*. PMLR, 2018.
Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In Advances in Neural Information Processing Systems, pp. 271–279, 2016.
Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv preprint arXiv:1807.03748* (2018).

- **Main Idea (Well-designed Tasks)**

$$\arg\max_{\omega_1,\omega_2,\psi} \left(\alpha \widehat{\mathcal{I}}_{\omega_1,\psi}(X;E_\psi(X)) + \frac{\beta}{M^2}\sum_{i=1}^{M^2}\widehat{\mathcal{I}}_{\omega_2,\psi}(X^{(i)};E_\psi(X))\right) + \arg\min_{\psi} \arg\max_{\phi} \gamma \widehat{\mathcal{D}}_\phi(\mathbb{V}||\mathbb{U}_{\psi,\mathbb{P}}),$$

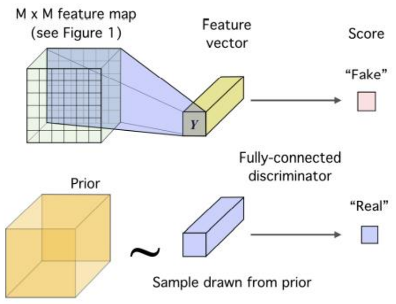
DIM (Global) DIM (Local) Prior Matching




Receptive field

fake

Local DIM Framework




➔ 그래프에 적용시킬 수 없을까??

Hjelm, R. Devon, et al. "Learning deep representations by mutual information estimation and maximization." *arXiv preprint arXiv:1808.06670* (2018).

- **Sum up**

1. Contrastive learning (Positive, Negative)

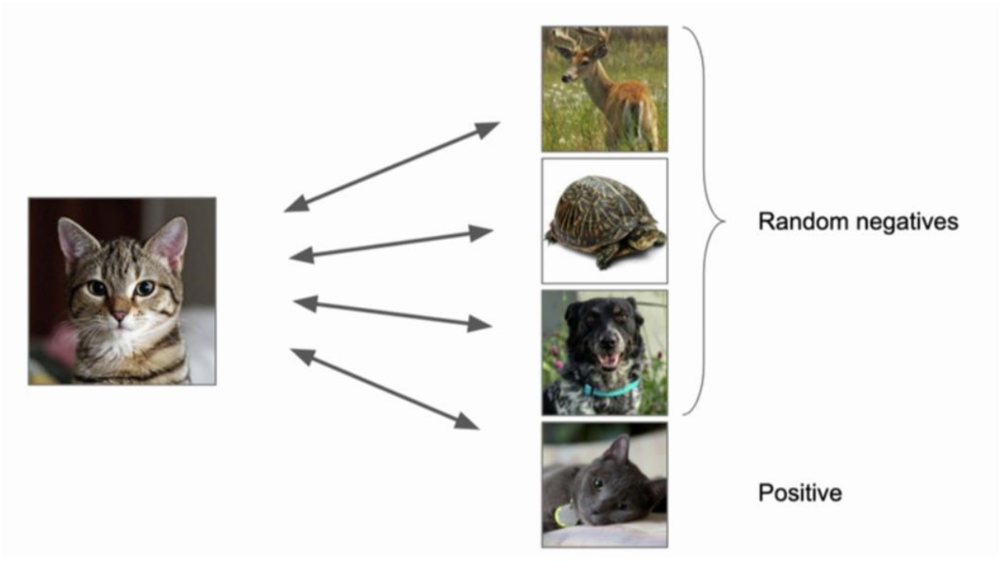
2. local 과 global feature 사이의 MI expectation

   ➔ 효과적인 Representation
   ➔ Robust

# 02 Our model



- **Negative Sampling**

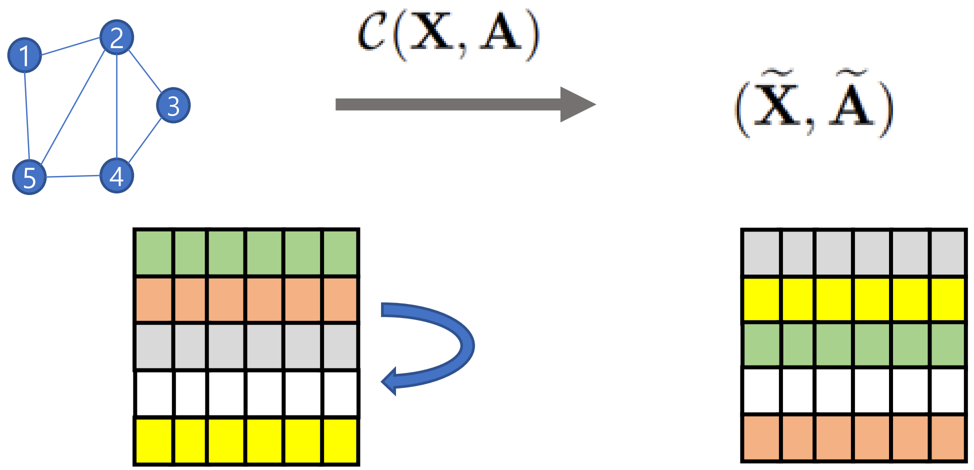



Mutual information - Contrastive (figure from Oord)

- **Use the corruption function!!**

  ① Preserve the original adjacency matrix
  ② Corrupt features X by row-wise shuffling

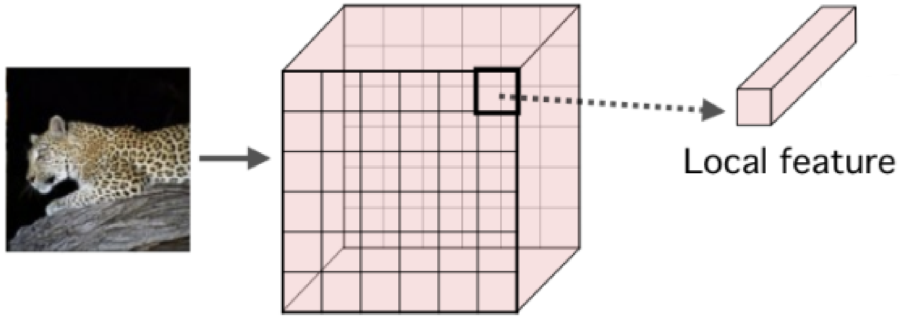
  ➔ has similar levels of connectivity to the positive graph

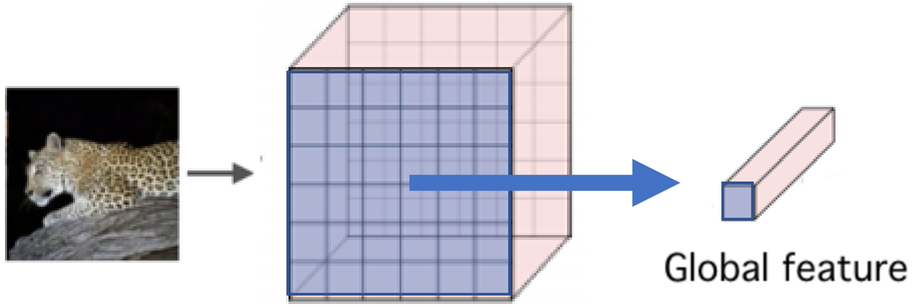


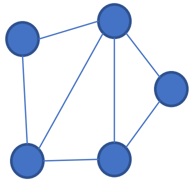
# 02 Our model



- **How to extract feature in Graph**



A set of node features $\mathbf{X} = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\}$
adjacency matrix, $\mathbf{A} \in \mathbb{R}^{N \times N}$

Graph Conv

$\mathcal{E} : \mathbb{R}^{N \times F} \times \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N \times F'}$

**Local Feature** $\boxed{\mathcal{E}(\mathbf{X}, \mathbf{A}) = \mathbf{H} = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}}$

: patch representations

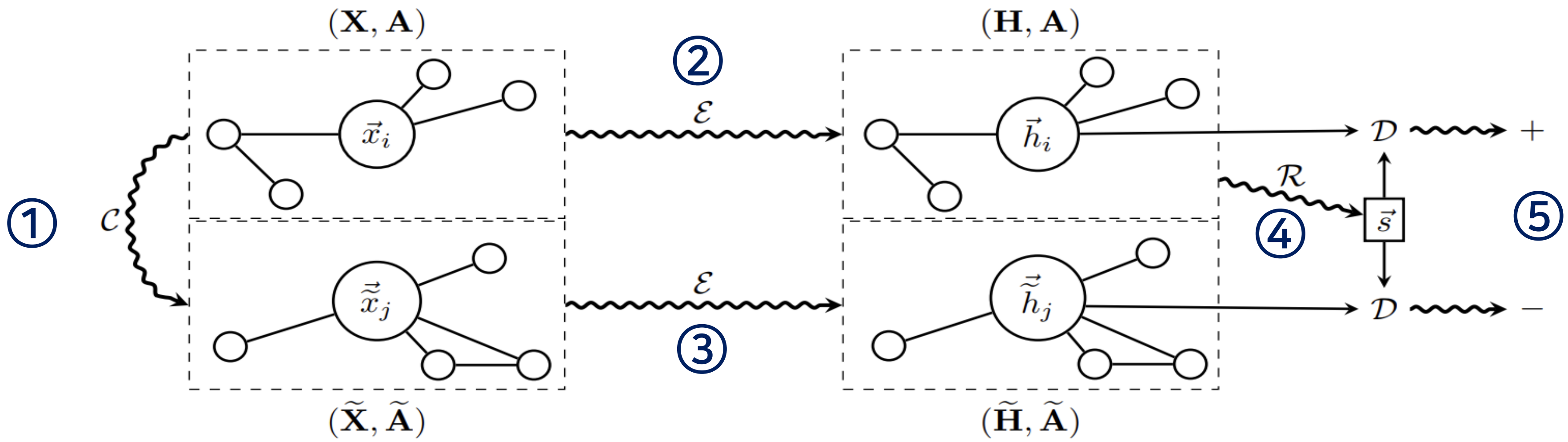
Readout function $\vec{s} = \mathcal{R}(\mathbf{H})$

$= \sigma\left(\frac{1}{N}\sum_{i=1}^{N} \vec{h}_i\right)$

**Global Feature**

: graph-level representations

- **Overview of DGI**




① Negative example by using the corruption function

② Obtain patch representations for input graph

③ Obtain patch representations for negative example

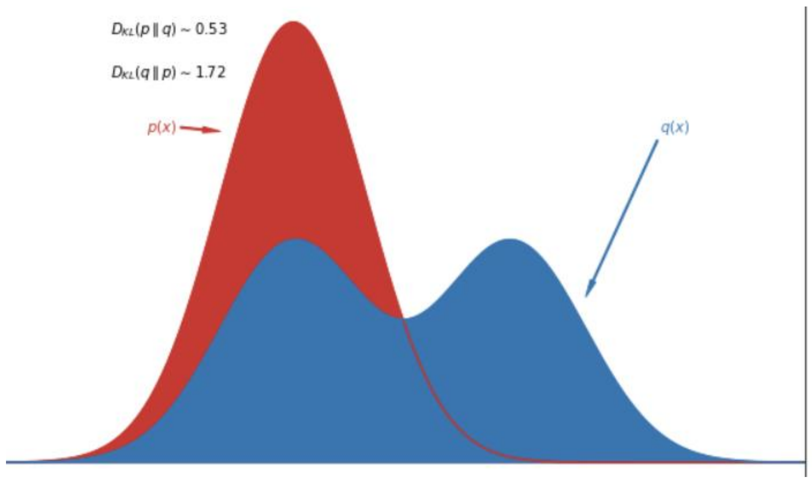
④ Summarize the input graph by passing its patch representations through the readout function

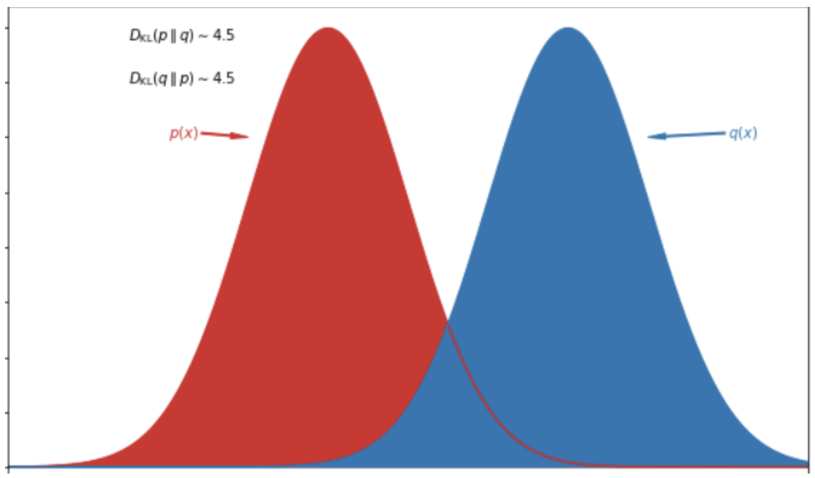
⑤ Update parameters of $\epsilon$, R, D by applying gradient descent

# 02 Our model



## ▪ Discriminator & Mutual Information




Maximize Mutual Information → Distinguish between samples drawn from the joint p(x, y) and those drawn from the product of marginals p(x)p(y),

# 02 Our model



## ▪ Discriminator On DGI

- Discriminator function

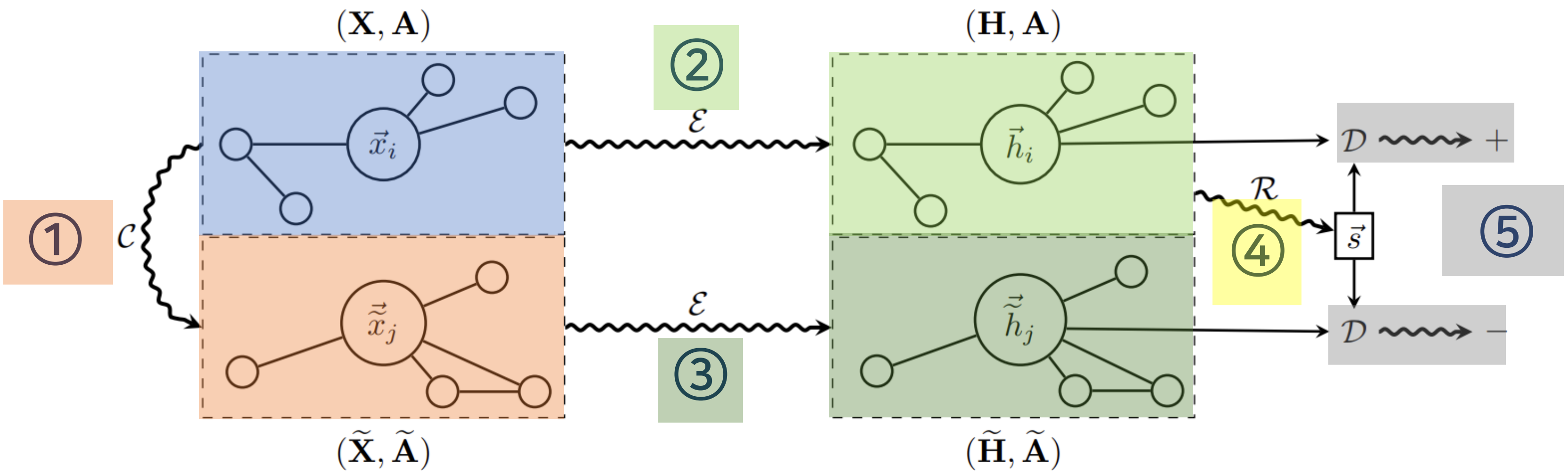
$$\mathcal{D}(\vec{h}_i, \vec{s}) = \sigma\left(\vec{h}_i^T \mathbf{W} \vec{s}\right)$$

In contrastive objective, Discriminator 는 joint (positive examples)로부터 나온 샘플과 marginal의 product로부터 나온 샘플(negative examples)을 구별하는 것

- Objective function

$$\mathcal{L} = \frac{1}{N+M}\left(\underbrace{\sum_{i=1}^{N} \mathbb{E}_{(\mathbf{X},\mathbf{A})}\left[\log \mathcal{D}\left(\vec{h}_i, \vec{s}\right)\right]}_{\text{positive examples}} + \underbrace{\sum_{j=1}^{M} \mathbb{E}_{(\tilde{\mathbf{X}},\tilde{\mathbf{A}})}\left[\log\left(1 - \mathcal{D}\left(\vec{\tilde{h}}_j, \vec{s}\right)\right)\right]}_{\text{negative examples}}\right)$$

- **Overview of DGI functions**




① $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{A}}) \sim \mathcal{C}(\mathbf{X}, \mathbf{A}).$

② ③

$$\mathbf{H} = \mathcal{E}(\mathbf{X}, \mathbf{A}) = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}.$$

$$\widetilde{\mathbf{H}} = \mathcal{E}(\widetilde{\mathbf{X}}, \widetilde{\mathbf{A}}) = \{\vec{\widetilde{h}}_1, \vec{\widetilde{h}}_2, \ldots, \vec{\widetilde{h}}_M\}.$$

④

$$\mathcal{R}(\mathbf{H}) = \sigma\left(\frac{1}{N}\sum_{i=1}^{N}\vec{h}_i\right)$$

⑤

$$\mathcal{D}(\vec{h}_i, \vec{s}) = \sigma\left(\vec{h}_i^T \mathbf{W} \vec{s}\right)$$

# 03 Experiments

## ▪ Dataset Details

① Transductive learning task
(Cora, Citeseer, Pubmed)

- 고정된 Graph에서 추론을 행하는 것
- unlabeled training data도 그들이 가진 특성(ex. 데이터 간 연결 관계, 거리)을 활용해 새로운 prediction을 하는 것
- high computational cost
  : 새로운 데이터가 들어오면, 처음부터 모델을 구축하여야 함

② Inductive learning task
(Reddit, PPI)

- 틀에서 벗어나 새로운 Node에 대해서도 합리적인 추론을 행할 수 있는 경우
- supervised learning으로, 어떤 function parameter를 주어진 labeled training data로 학습하는 것
- less computational cost
  : 모델을 구축하여 새로운 데이터가 입력으로 들어왔을 때 대응 가능 (지금껏 본 적이 없는 Node에 대해 일반화를 하는 것)

# 03 Experiments



## ▪ Experimental Setup

① Transductive learning task
(Cora, Citeseer, Pubmed)

- Encoder : one-layer GCN model

$$\mathcal{E}(\mathbf{X}, \mathbf{A}) = \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{\Theta}\right)$$

➔ 학습된 filter가 fixed and known adjacency matrix에 의존

② Inductive learning task
(Reddit, PPI)

- Encoder : GraphSAGE

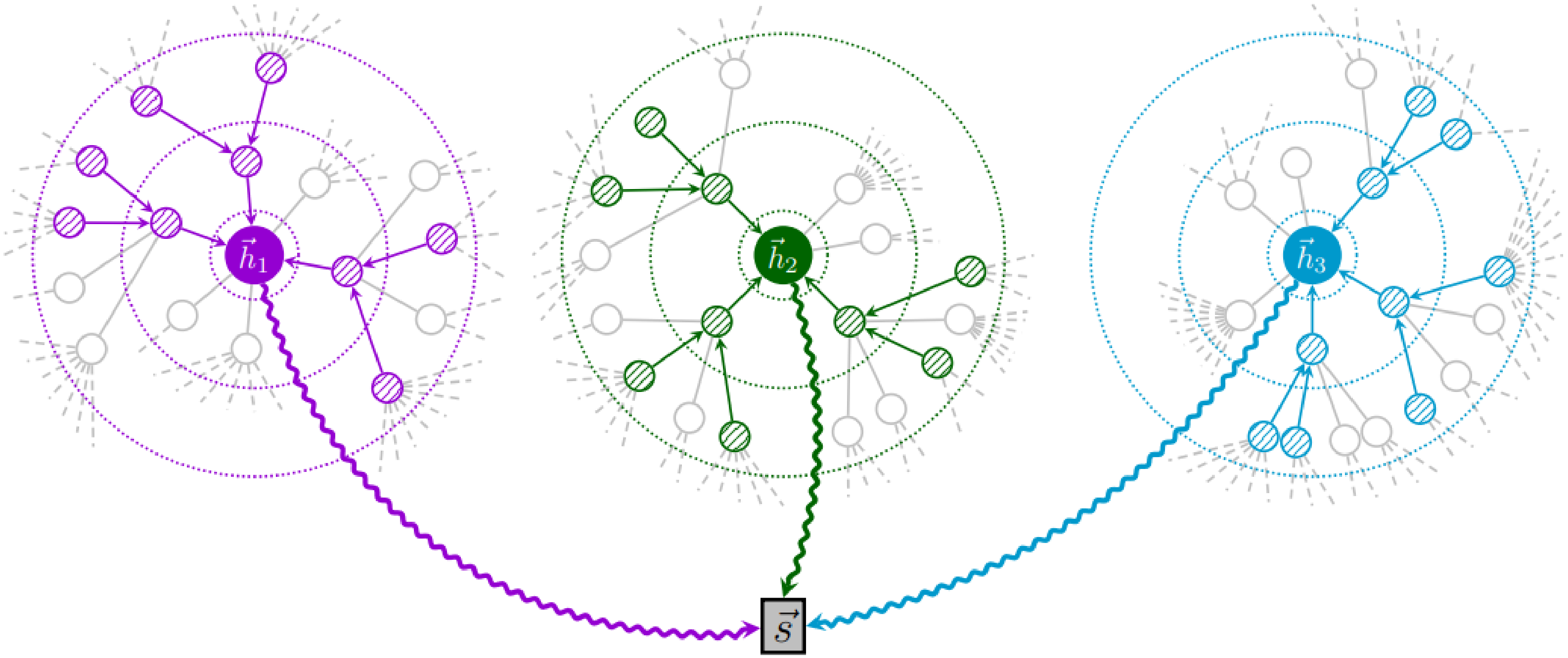
$$\mathrm{MP}(\mathbf{X}, \mathbf{A}) = \hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}\mathbf{X}\mathbf{\Theta}$$

$$\mathcal{E}(\mathbf{X}, \mathbf{A}) = \widetilde{\mathrm{MP}}_3(\widetilde{\mathrm{MP}}_2(\widetilde{\mathrm{MP}}_1(\mathbf{X}, \mathbf{A}), \mathbf{A}), \mathbf{A})$$

➔ efficiently generate node embeddings for previously unseen data.

- **Preview of GraphSAGE**

  - Aggregation 함수를 사용하여 GCN을 일반화하자!



# 03 Experiments



- **Results**

① Transductive learning task (Cora, Citeseer, Pubmed)

② Inductive learning task (Reddit, PPI)

*Transductive*

| Available data | Method | Cora | Citeseer | Pubmed |
|---|---|---|---|---|
| **X** | Raw features | $47.9 \pm 0.4\%$ | $49.3 \pm 0.2\%$ | $69.1 \pm 0.3\%$ |
| **A, Y** | LP (Zhu et al., 2003) | 68.0% | 45.3% | 63.0% |
| **A** | DeepWalk (Perozzi et al., 2014) | 67.2% | 43.2% | 65.3% |
| **X, A** | DeepWalk + features | $70.7 \pm 0.6\%$ | $51.4 \pm 0.5\%$ | $74.3 \pm 0.9\%$ |
| **X, A** | Random-Init (ours) | $69.3 \pm 1.4\%$ | $61.9 \pm 1.6\%$ | $69.6 \pm 1.9\%$ |
| **X, A** | **DGI** (ours) | $\mathbf{82.3} \pm 0.6\%$ | $\mathbf{71.8} \pm 0.7\%$ | $\mathbf{76.8} \pm 0.6\%$ |
| **X, A, Y** | GCN (Kipf & Welling, 2016a) | 81.5% | 70.3% | 79.0% |
| **X, A, Y** | Planetoid (Yang et al., 2016) | 75.7% | 64.7% | 77.2% |

*Inductive*

| Available data | Method | Reddit | PPI |
|---|---|---|---|
| **X** | Raw features | 0.585 | 0.422 |
| **A** | DeepWalk (Perozzi et al., 2014) | 0.324 | — |
| **X, A** | DeepWalk + features | 0.691 | — |
| **X, A** | GraphSAGE-GCN (Hamilton et al., 2017a) | 0.908 | 0.465 |
| **X, A** | GraphSAGE-mean (Hamilton et al., 2017a) | 0.897 | 0.486 |
| **X, A** | GraphSAGE-LSTM (Hamilton et al., 2017a) | 0.907 | 0.482 |
| **X, A** | GraphSAGE-pool (Hamilton et al., 2017a) | 0.892 | 0.502 |
| **X, A** | Random-Init (ours) | $0.933 \pm 0.001$ | $0.626 \pm 0.002$ |
| **X, A** | **DGI** (ours) | $\mathbf{0.940} \pm 0.001$ | $\mathbf{0.638} \pm 0.002$ |
| **X, A, Y** | FastGCN (Chen et al., 2018) | 0.937 | — |
| **X, A, Y** | Avg. pooling (Zhang et al., 2018) | $0.958 \pm 0.001$ | $0.969 \pm 0.002$ |

- Demonstrate strong performance being achieved across all five datasets

# 04 Conclusion



- **Sum up**

  - New approach for learning unsupervised representations on graph-structured data

  - Leveraging local mutual information maximization
    : graph's patch representations + global structural properties of graph

  - transductive, inductive classification task 모두 competitive performance를 보임

- **Discussion**

  - It is very important to tracking trends in various fields such as Vison, NLP, Graph.

  - Readout function : Simple averaging of all the node's features ➔ Global Vector
    - Future work

  - Corruption function : row-wise shuffling on X features
    - Future work

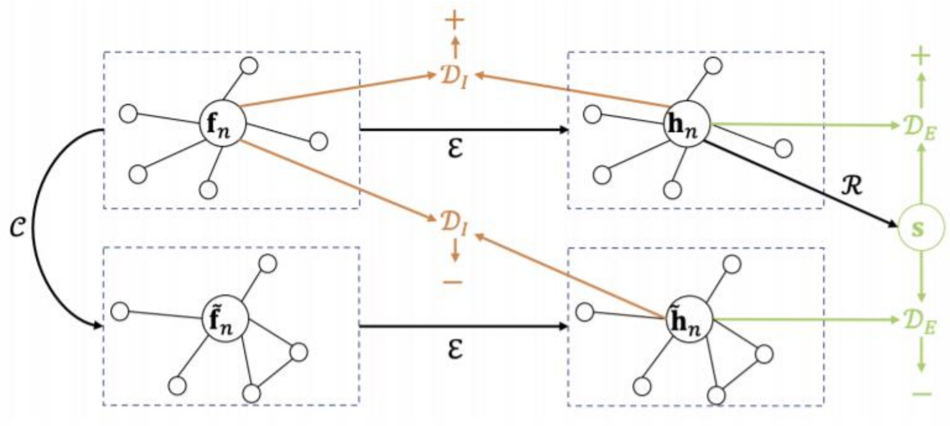
# 04 Conclusion



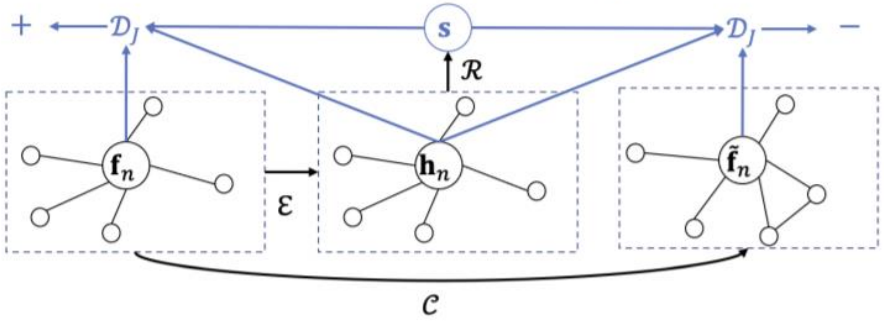
- **Limitation**

1) DGI merely considers the extrinsic supervision signal
   - Ignores intrinsic signal
     (between node embedding, node attributes)

2) DGI is designed for a single attributed network
   - Nodes are connected by multiple relations
     (Multiplex Graph)



(a) Illustration of the extrinsic and intrinsic supervision.



(b) Illustration of the joint supervision.

summary

$$\hat{I}(\mathrm{h}_n; \mathrm{s}; \mathrm{f}_n) = \hat{I}(\mathrm{h}_n; \mathrm{s}) + \hat{I}(\mathrm{h}_n; \mathrm{f}_n) - \hat{I}(\mathrm{h}_n; \mathrm{s}, \mathrm{f}_n)$$

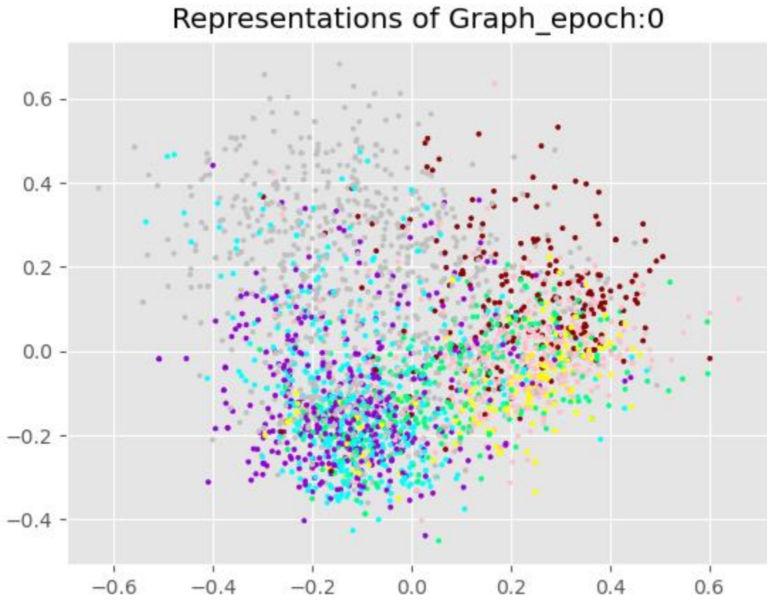
embedding attribute

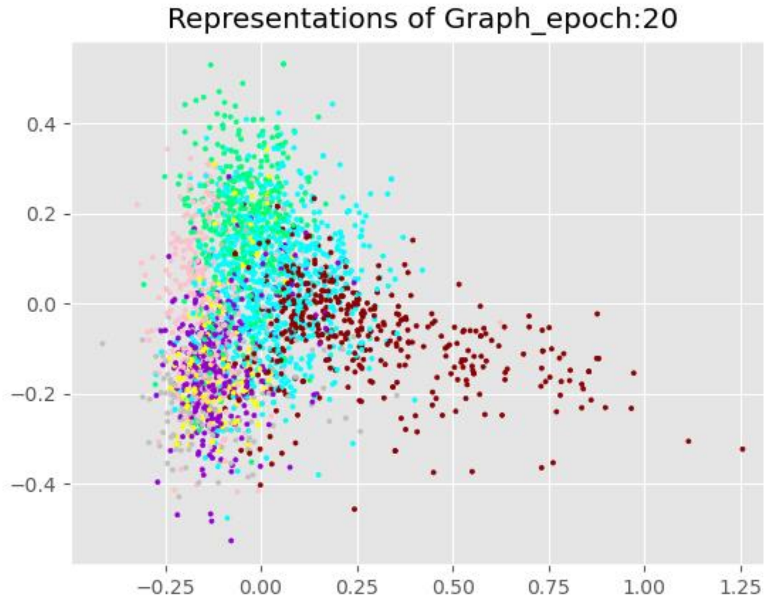
Park, Chanyoung, Jiawei Han, and Hwanjo Yu. "Deep multiplex graph infomax: Attentive multiplex network embedding using global information." Knowledge-Based Systems 197 (2020): 105861.
Jing, Baoyu, Chanyoung Park, and Hanghang Tong. "Hdmi: High-order deep multiplex infomax." Proceedings of the Web Conference 2021. 2021.

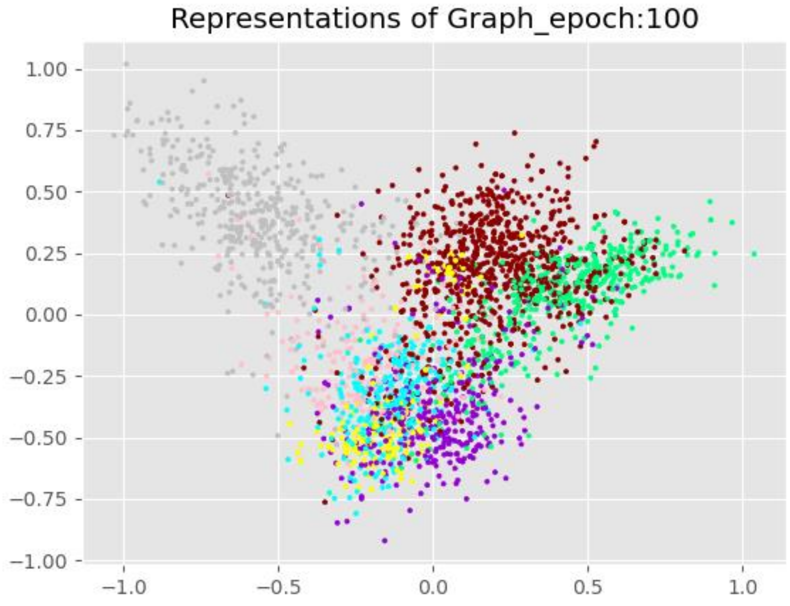
# 05 Implementation

## ▪ Visualization (PCA, t-SNE)

1. PCA



# 05 Implementation



- **Visualization (PCA, t-SNE)**

2. t-SNE

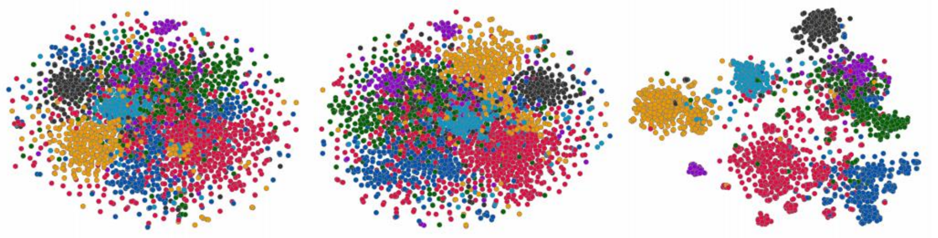
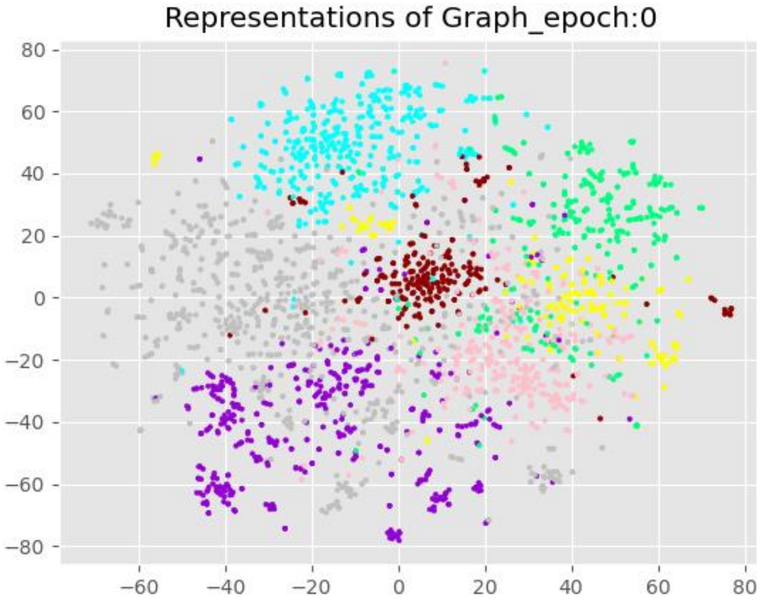
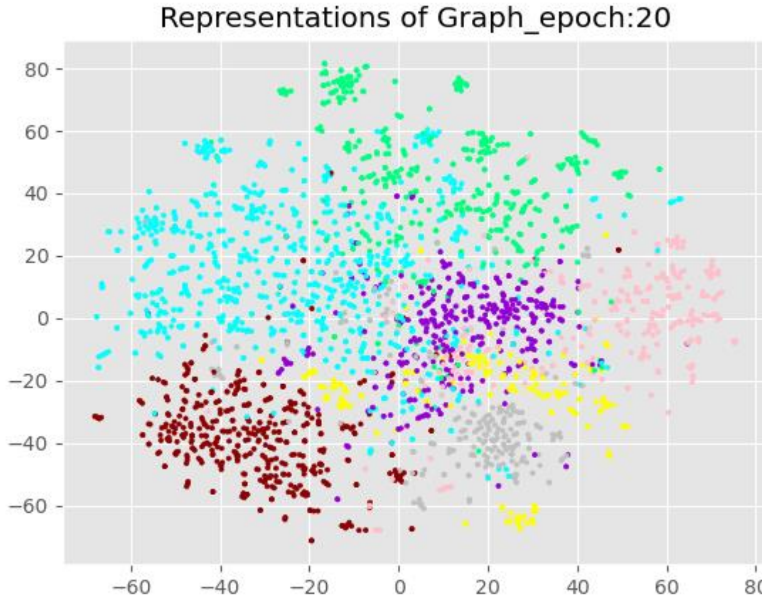
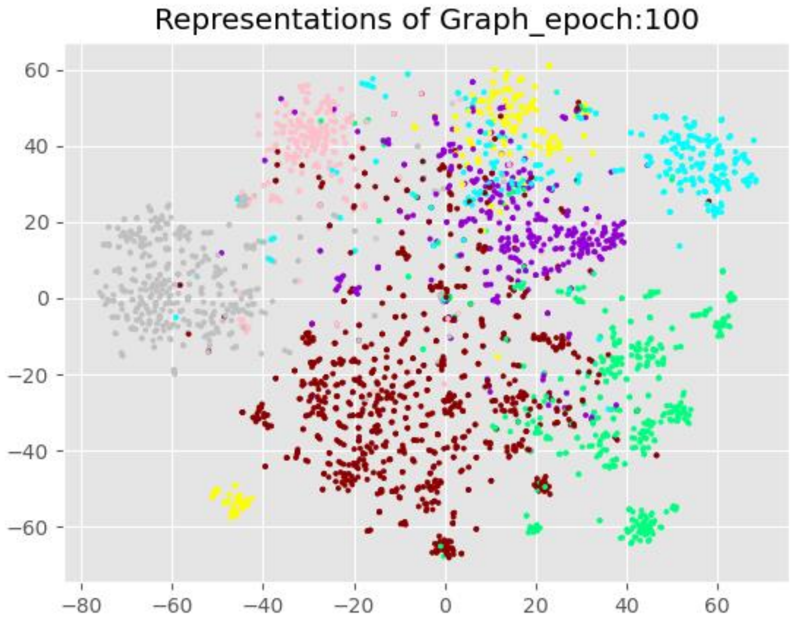


Figure 3: t-SNE embeddings of the nodes in the Cora dataset from the raw features (**left**), features from a randomly initialized DGI model (**middle**), and a learned DGI model (**right**). The clusters of the learned DGI model's embeddings are clearly defined, with a Silhouette score of 0.234.



- **Entropy**

: 확률변수 x의 불확실성을 나타내는 엔트로피

이산 확률분포 $H(x) = -\sum_{i=1,k} P(e_i) log_2 P(e_i)$ 또는 $H(x) = -\sum_{i=1,k} P(e_i) log_e P(e_i)$

연속 확률분포 $H(x) = -\int_R P(x) log_2 P(x)$ 또는 $H(x) = -\int_R P(x) log_e P(x)$

- **Cross Entropy**

: 두 확률분포 P와 Q 사이의 교차 엔트로피

$$
\begin{aligned}
H(P,Q) &= -\sum_X P(x) log_2 Q(x) \\
&= -\sum_X P(x) log_2 P(x) + \sum_X P(x) log_2 P(x) - \sum_X P(x) log_2 Q(x) \\
&= \mathrm{H(P)} + \boxed{\sum_X P(x) log_2 \frac{P(x)}{\mathrm{Q}(x)}}
\end{aligned}
$$

$\mathrm{Plog_2Q} = \mathrm{Plog}(\frac{Q}{P} * P)$

KL divergence

- **Mutual Information and KL Divergence**

$$
\begin{aligned}
I(X;Y) &= H(Y) - H(Y|X) \\
&= -H(Y|X) + H(Y) \\
&= -\sum_x p(x) H(Y|X=x) - \sum_y p(y)\log p(y) \\
&= \sum_x p(x) \Big(\sum_y p(y|x)\log p(y|x)\Big) - \sum_y \log p(y) \Big(\sum_x p(x,y)\Big) \\
&= \sum_{x,y} p(x)p(y|x)\log p(y|x) - \sum_{x,y} p(x,y)\log p(y) \\
&= \sum_{x,y} p(x,y)\log \frac{p(x,y)}{p(x)} - \sum_{x,y} p(x,y)\log p(y) \\
&= \sum_{x,y} p(x,y)\log \frac{p(x,y)}{p(x)p(y)} \\
&= D_{KL}(p(x,y)||p(x)p(y))
\end{aligned}
$$

# Q & A

Github :