

# Edge-Labeling Graph Neural Network for Few-shot Learning (2019 CVPR)

**Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D.Yoo**

- Korea Advanced Institute of Science and Technology
- MILA, Universite de Montreal
- Kakao Brain

2017-27619

Hyun Jun Choi

Seoul National University

# Few shot learning



*Amanita muscaria*, commonly known as the **fly agaric** or **fly amanita**, is a **basidiomycete** of the genus *Amanita*. It is also a **muscimol mushroom**. Native throughout the **temperate** and **boreal** regions of the Northern Hemisphere, *Amanita muscaria* has been unintentionally **introduced** to many countries in the Southern Hemisphere, generally as a **symbiont** with pine and birch plantations, and is now a true **cosmopolitan** species. It **associates** with various **deciduous** and **coniferous** trees.

- 광대버섯
- **독성**이 있음!

# Few shot learning



광대버섯(독성 있음!)

- ML supervised learning 관점에서
- 한 개의 data로 그 class에 대한 classifier 학습
- 어려운 task

# Few shot learning (2-way 4 shot learning)

## Episode 1

### Training

Train dataset #1: "cat-bird"

cats



birds



## Episode 2

Train dataset #2: "flower-bike"

flowers



bikes



### Testing

Train 과정에서 없는 새로운 class

## Episode (Test)

Test dataset: "dog-otter"

dogs



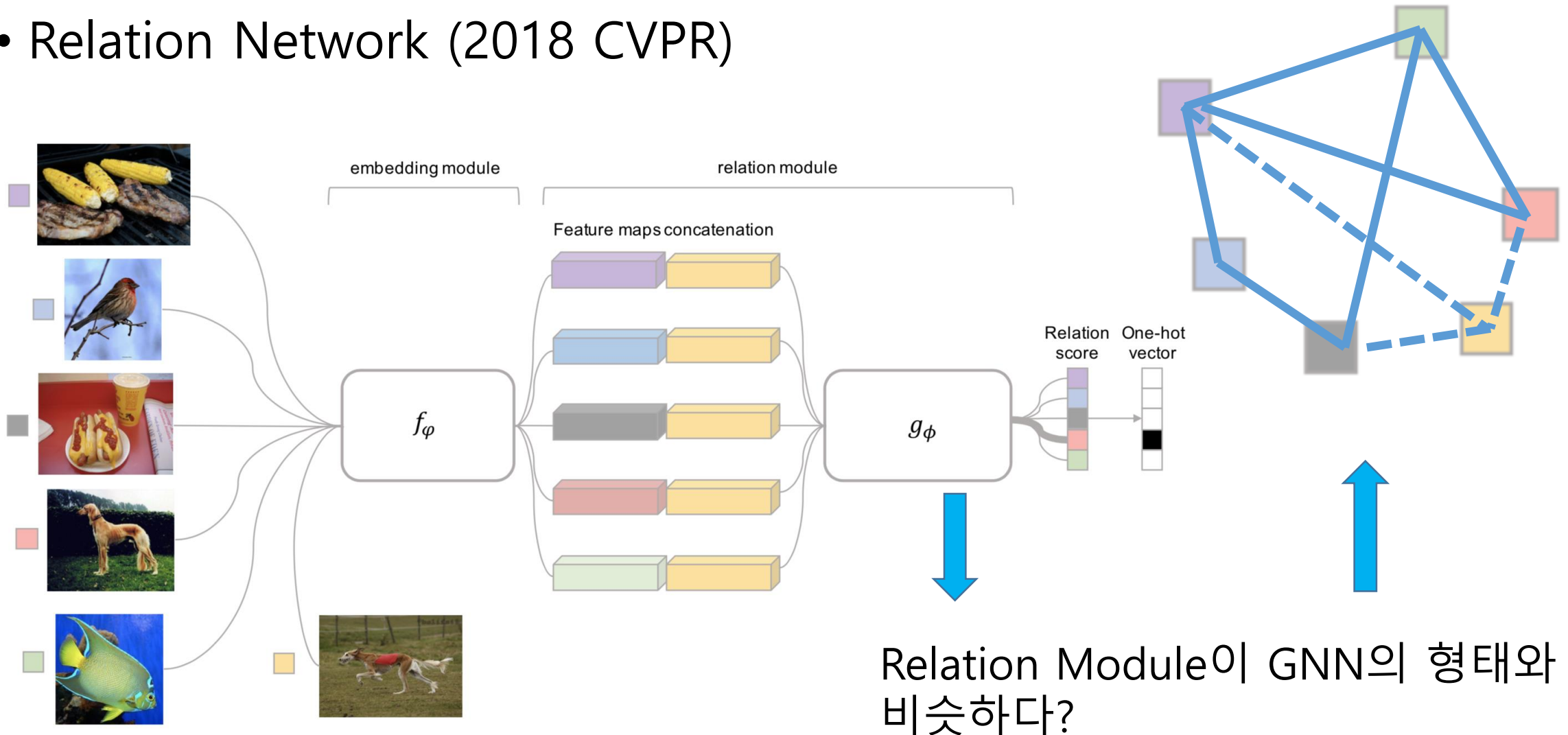
otters





# Metric 기반의 기존 방법 → GNN 기반의 방법

- Relation Network (2018 CVPR)

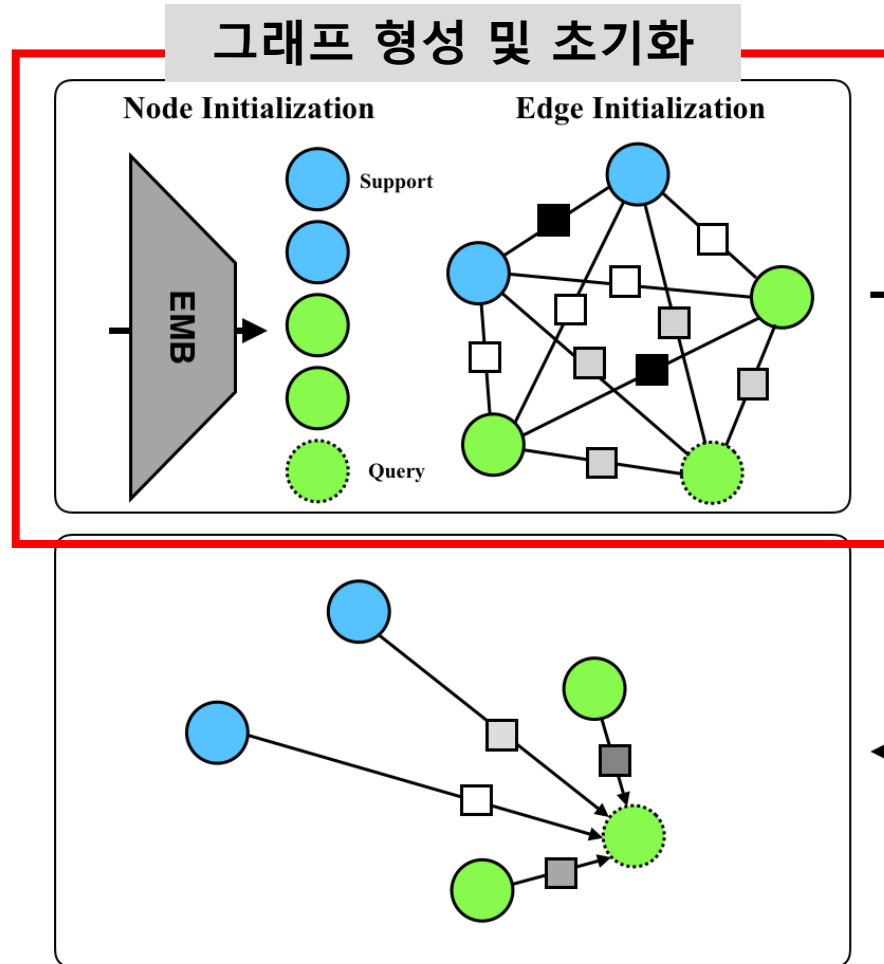


# Proposed Method (EGNN)

	범주 정보를 담는 부분	범주별 군집화 여부	업데이트를 하는 부분	훈련 단계에서 범주 수 변경 가능 여부
GNN	노드	X	노드	X
TPN	노드	X	X	O
EGNN	간선	O	노드와 간선	O

- 기존 GNN의 문제점
  - 노드에 대한 업데이트 및 이에 대한 결과로 바로 노드의 class를 얻음
  - 같은 클래스끼리 근접하게, 다른 클래스끼리 서로 멀어지게 하지 않음
- EGNN에서 제안한 방법
  - Edge를 통해 노드 간의 유사도를 직접적으로(explicitly) 모델링 함
  - Edge의 value를 통해 클래스끼리 근접하게, 다른 클래스끼리 서로 멀어지게 함

# Proposed Method



(c) Query Node Label Prediction

## - Node Initialization

- EMB : Embedding Network 로써, 기존 논문들이 사용하던 단순한 4-block CNN backbone network 사용(image to feature)

## - Edge Initialization (같은 class 1, 다르면 0)

$$y_{ij} = \begin{cases} 1, & \text{if } y_i = y_j, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$$e_{ij}^0 = \begin{cases} [1||0], & \text{if } y_{ij} = 1 \text{ and } i, j \leq N \times K, \\ [0||1], & \text{if } y_{ij} = 0 \text{ and } i, j \leq N \times K, \\ [0.5||0.5], & \text{otherwise,} \end{cases} \quad (2)$$

- 2d vector로 intra-cluster similarity, inter-cluster dissimilarity 표현  
-> 두 개로 분리한 경우 성능이 제일 좋았음

(b-2) EGNN: Edge Feature Update

# Proposed Method

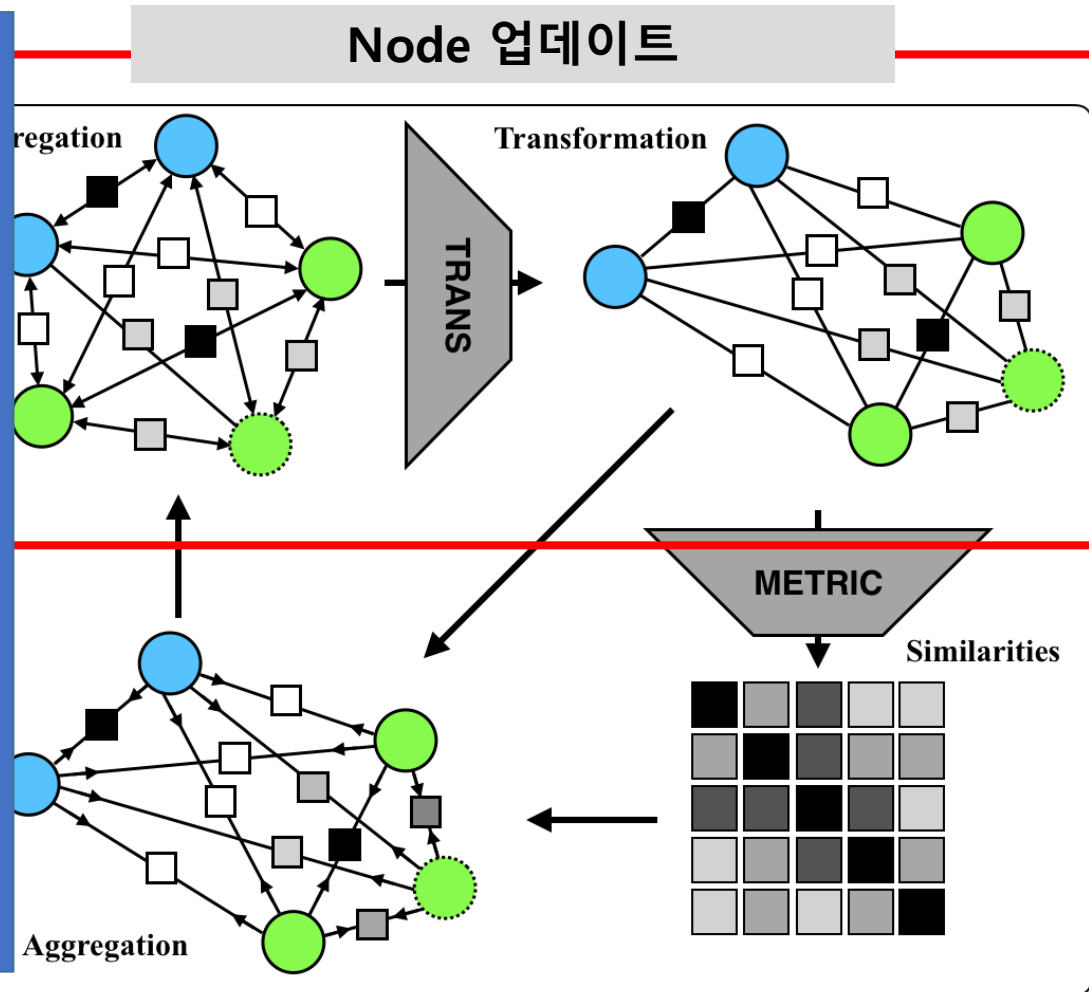
## - Node feature update

- TRANS: 이전의 edge와 node feature로부터 node feature update(1-hop) 함
- Attention mechanism과 유사한 원리 이용
- 대신, Edge feature가 곧 Attention 에 대한 degree of contribution

$$\mathbf{v}_i^\ell = f_v^\ell([\sum_j \tilde{e}_{ij1}^{\ell-1} \mathbf{v}_j^{\ell-1} || \sum_j \tilde{e}_{ij2}^{\ell-1} \mathbf{v}_j^{\ell-1}]; \theta_v^\ell),$$

$$\tilde{e}_{ijd} = \frac{e_{ijd}}{\sum_k e_{ikd}}$$

(c) Query Node Label Prediction



(b-2) EGNN: Edge Feature Update



# Proposed Method

## - Edge feature update

- 업데이트 된 node feature 기반으로 노드 간의 연결성(metric 네트워크)을 구함
- node feature간의 metric(연결성) +이전의 edge feature 둘을 결합하여 새로이 edge feature update 함

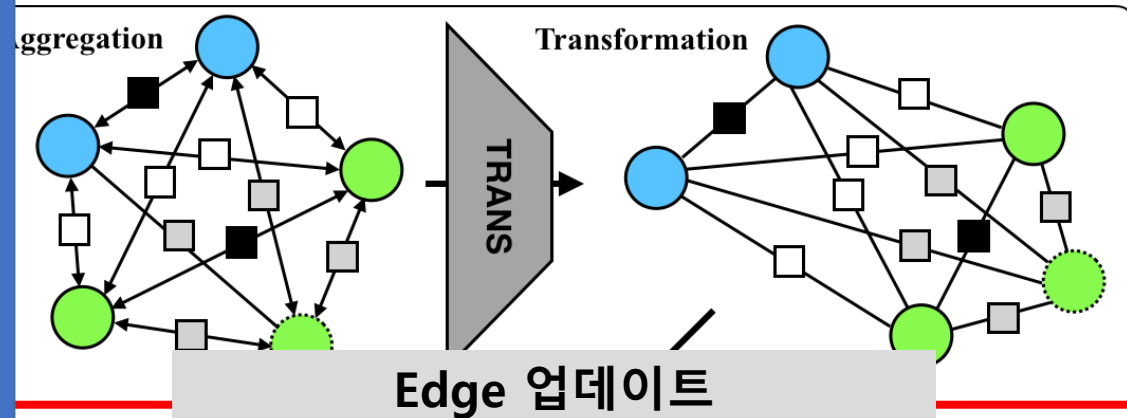
$$\bar{e}_{ij1}^{\ell} = \frac{f_e^{\ell}(\mathbf{v}_i^{\ell}, \mathbf{v}_j^{\ell}; \theta_e^{\ell}) e_{ij1}^{\ell-1}}{\sum_k f_e^{\ell}(\mathbf{v}_i^{\ell}, \mathbf{v}_k^{\ell}; \theta_e^{\ell}) e_{ik1}^{\ell-1} / (\sum_k e_{ik1}^{\ell-1})},$$

$$\bar{e}_{ij2}^{\ell} = \frac{(1 - f_e^{\ell}(\mathbf{v}_i^{\ell}, \mathbf{v}_j^{\ell}; \theta_e^{\ell})) e_{ij2}^{\ell-1}}{\sum_k (1 - f_e^{\ell}(\mathbf{v}_i^{\ell}, \mathbf{v}_k^{\ell}; \theta_e^{\ell})) e_{ik2}^{\ell-1} / (\sum_k e_{ik2}^{\ell-1})},$$

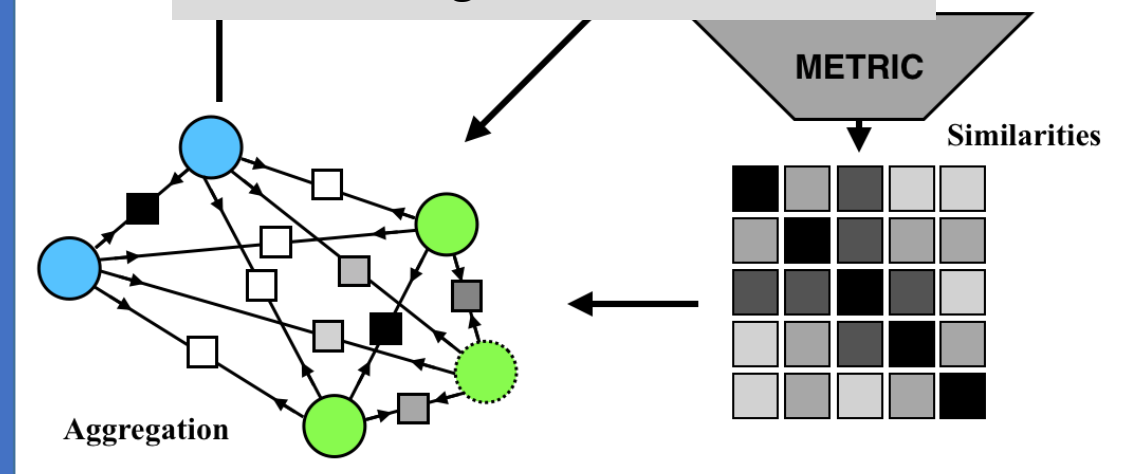
$$\mathbf{e}_{ij}^{\ell} = \bar{\mathbf{e}}_{ij}^{\ell} / \|\bar{\mathbf{e}}_{ij}^{\ell}\|_1,$$

$f_e^{\ell}$  is the metric network

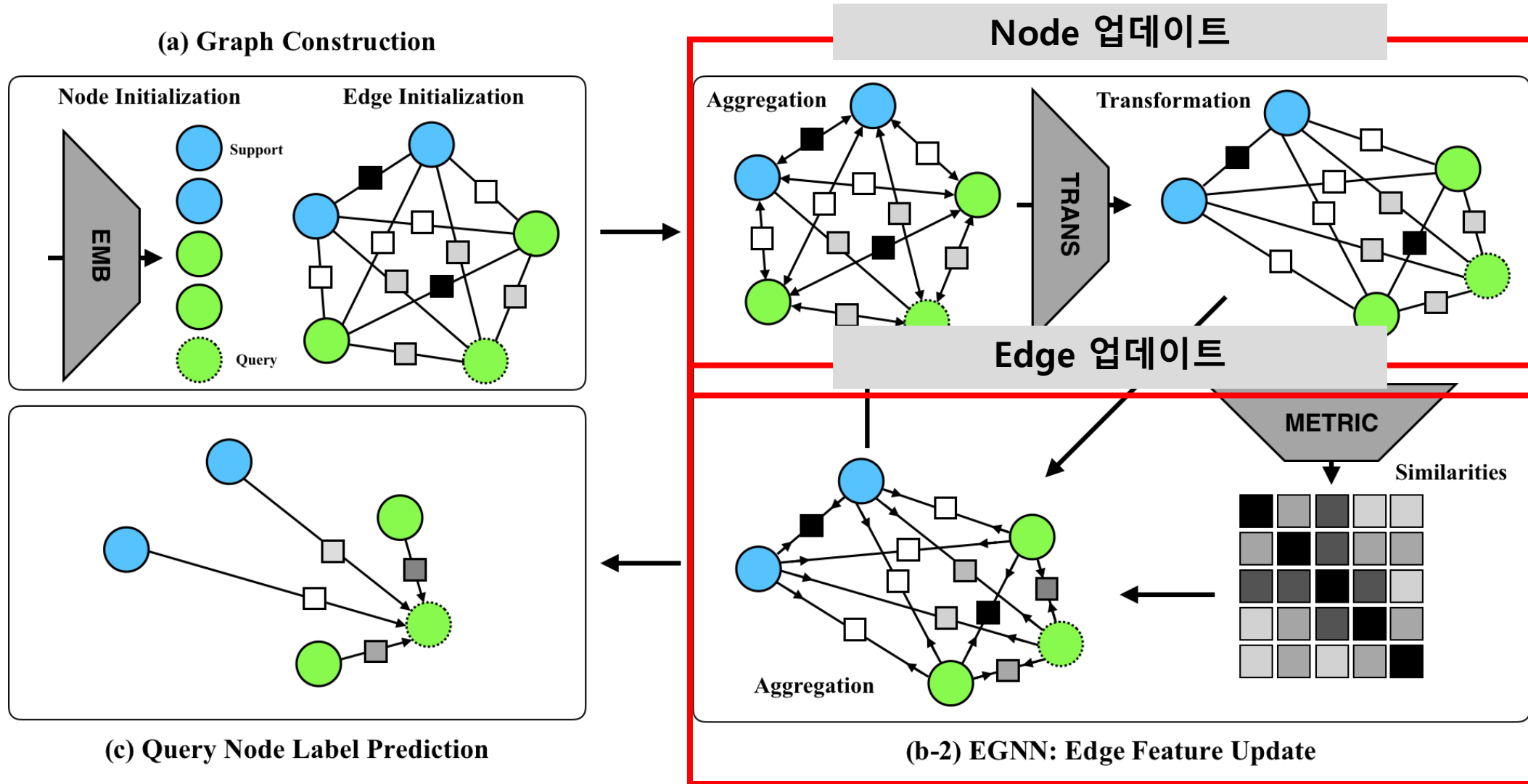
(b-1) EGNN: Node Feature Update



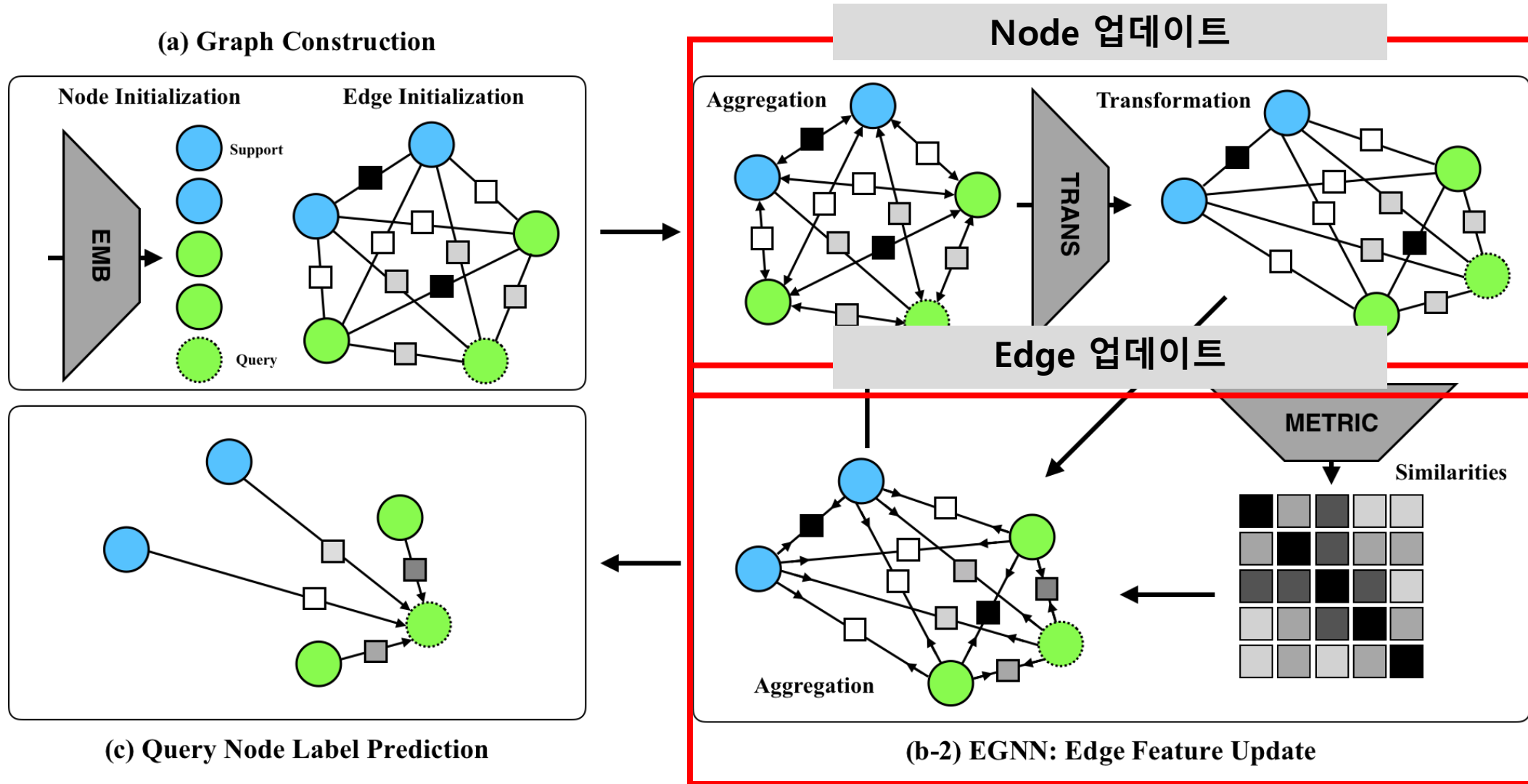
(b-2) EGNN: Edge Feature Update



# Proposed Method

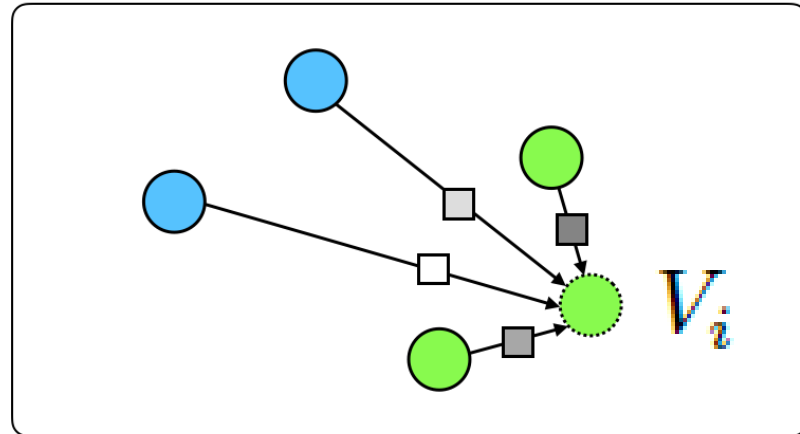
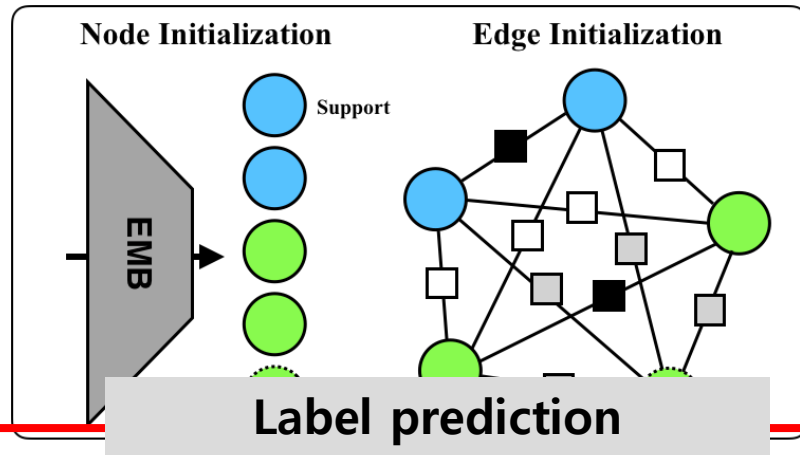


# Proposed Method



# Proposed Method

(a) Graph Construction



(c) Query Node Label Prediction

## - Label prediction

- 인접한 노드 중 label이 있는 것으로 부터 voting을 통해 결정
- Voting에 대한 가중치는 연결된 edge, 즉 similarity가 결정
- 최종적으로 class 별 similarity에 대해 softmax를 취해 확률로 표현

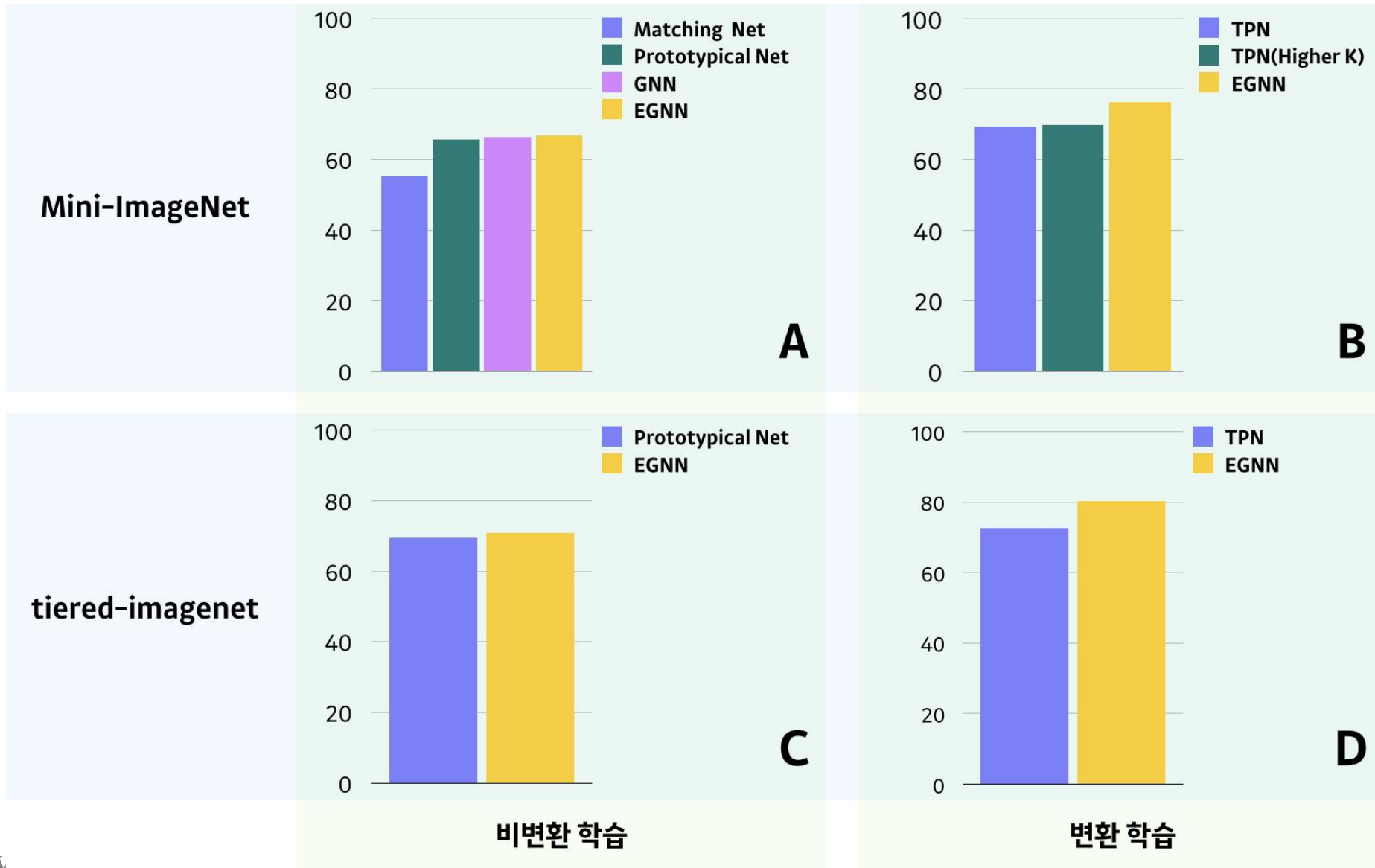
$V_i$  가  $C_k$  class 일 확률

$$P(y_i = C_k | \mathcal{T}) = p_i^{(k)}$$

$$p_i^{(k)} = \text{softmax} \left( \sum_{\{j: j \neq i \wedge (x_j, y_j) \in \mathcal{S}\}} \hat{y}_{ij} \delta(y_j = C_k) \right)$$

(b-2) EGNN: Edge Feature Update

# 성능 평가



기존의 GNN들에  
비해 성능 향상