

# EcoAI 랩세미나

## 양자배터리 팀 논문리뷰

### ‘The Graph Neural Network Model’

IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 20, NO. 1, JANUARY 2009.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini

○ 2025.11.7 ○

Presenter: 박범도

| 소속: EcoAI Lab

| E-mail: [pbeomdo@gmail.com](mailto:pbeomdo@gmail.com)

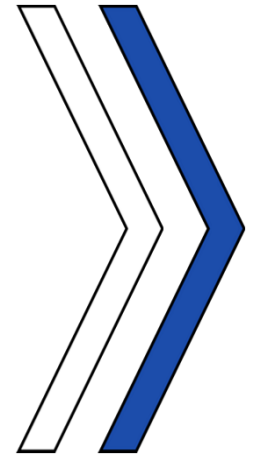
팀원: [박범도, 박준성, 장현석]

# 01

EcoAI 랩세미나 - 양자배터리 팀 논문리뷰

## Paper Review (발표: 박범도)

The Graph Neural Network Model



# Contents

**1-1** Introduction

**1-2** Method

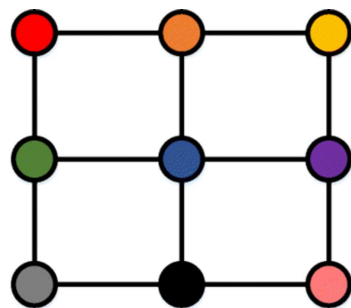
**1-3** Experiments and Results

**1-4** Conclusion

# 1-1 Introduction

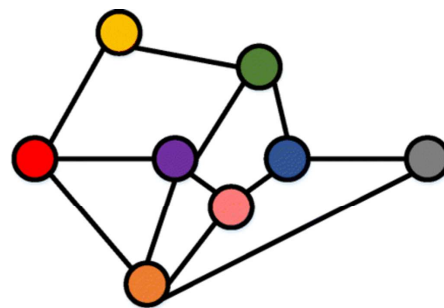
## 문제 정의

- 컴퓨터 비전, 분자 화학, 데이터 마이닝 등 많은 영역의 데이터가 본질적으로 그래프(Graph) 구조로 표현됨.
- 이러한 그래프 구조의 데이터(structured data)를 처리하기 위한 기존 신경망 방법론의 확장이 필요함.



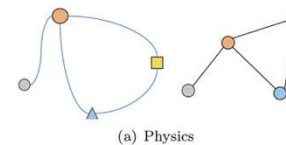
**CNN**

In Euclidean Space

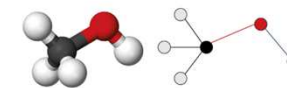


**GNN**

In Non-Euclidean Space



(a) Physics



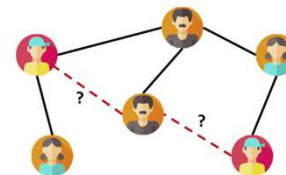
(b) Molecule



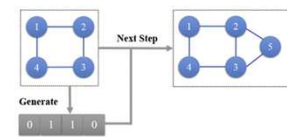
(c) Image



(d) Text



(e) Social Network



(f) Generation

## 1-1 Introduction

### 기존 접근법의 한계

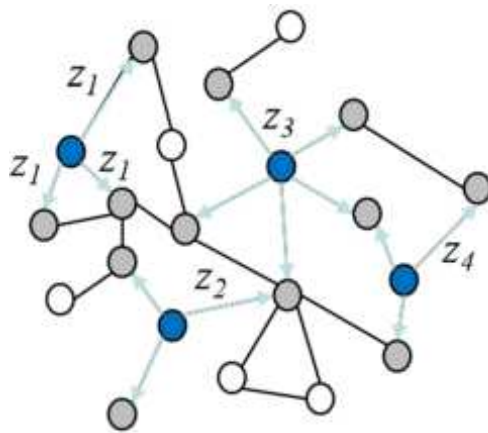
- Traditional Machine Learning: 그래프 구조 정보를 벡터나 리스트로 변환하는 전처리 단계에 의존함. 이 과정에서 그래프의 핵심 정보인 위상적 종속성이 손실될 수 있음.
- Recursive Neural Networks (RecNNs): 그래프 구조를 처리할 수 있지만, 입력 도메인이 유향 비순환 그래프 (DAGs, Directed Acyclic Graphs)로 엄격히 제한됨. 순환 그래프나 노드 중심의 복잡한 작업을 처리하기 위해서는 여전히 별도의 전처리가 필요함.
- Markov Chains: PageRank와 같은 모델이 있으나, 이는 GNN의 선형 모델의 특수한 경우(예: 가중치가 고정된 경우)일 때 가능함. → GNN은 이를 일반화하여 지도 학습을 도입하고 더 복잡한 프로세스를 모델링함.

# 1-1

## Introduction

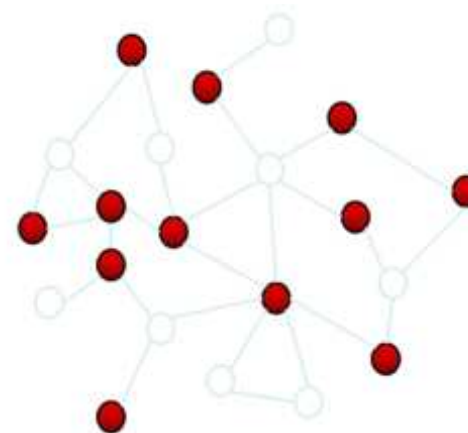
### Graph Neural Network

- 정보 확산 매커니즘
  - 각 노드는 이웃 노드들과 정보를 교환하여 상태를 업데이트함.
  - 해당 업데이트는 모든 노드의 상태가 안정화될 때까지 반복 수행함.



(a) 정보 확산 전

정보 확산  
→

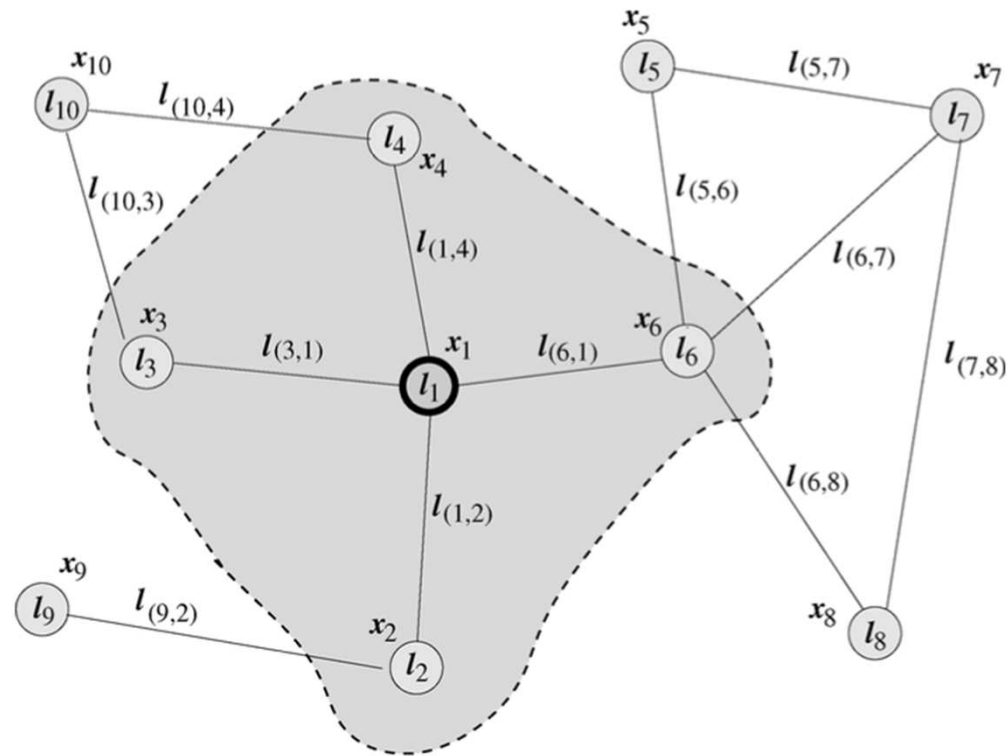


(a) 정보 확산 후

## 1-2 Method

### Graph Neural Network Model Structure

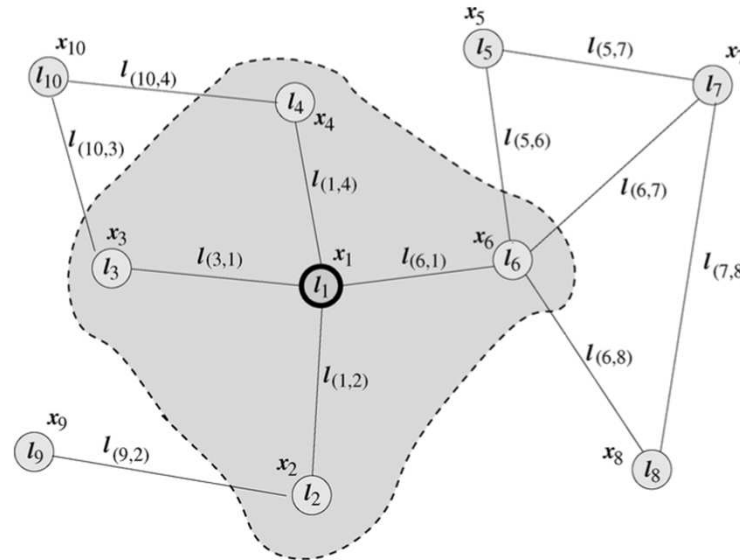
- 노드  $n$ 의 상태  $x_n$ 은 자신과 이웃한 노드들의 특징, 상태에 의해 결정됨.



## 1-2 Method

### Graph Neural Network Model Structure

- Local Transition Function ( $f_w$ ): 각 노드의 상태  $x_n$ 을 업데이트하는 함수
  - $x_n = f_w(l_n, l_{\{co[n]\}}, x_{\{ne[n]\}}, l_{\{ne[n]\}})$
- Local Output Function ( $g_w$ ): 최종 상태  $x_n$ 을 기반으로 노드의 출력  $o_n$ 을 계산하는 함수
  - $o_n = g_w(x_n, l_n)$



$$\rightarrow x_1 = f_w(l_1, l_{(1,2)}, l_{(3,1)}, l_{(1,4)}, l_{(6,1)}, x_2, x_3, x_4, x_6, l_2, l_3, l_4, l_6)$$

$$o_1 = g_w(x_1, l_1)$$



## 1-2 Method

### Graph Neural Network Model Structure

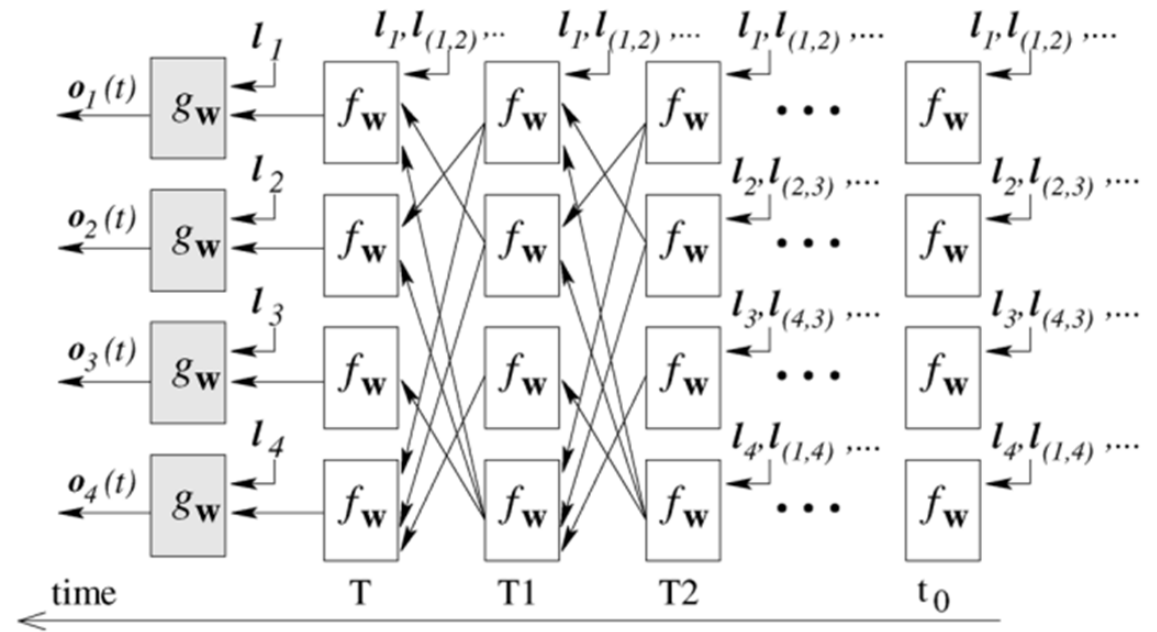
모든 노드의 상태 업데이트를 통합하여  
전체 그래프를 하나의 신경망으로 구성

- Banach 고정점 정리를 통한 수렴 보장.

$$X = F_w(L, X, G)$$

$$O = G_w(X, L)$$

$$X^* = \lim_{t \rightarrow \infty} F_w(L, X_t, G)$$



$$x_n = f_w(l_n, l_{co[n]}, x_{ne[n]}, l_{ne[n]})$$

$$o_n = g_w(x_n, l_n)$$

## 1-2 Method

### Learning Algorithm

- gradient 계산은 Almeida - Pineda (AP) 알고리즘으로 수행.
- AP 알고리즘으로 메모리 효율적인 그래디언트 계산 가능.

$$E = \sum_i \|t_i - o_i\|^2$$

$$\Delta w = -\eta \frac{\partial E}{\partial w}$$

```
MAIN
  initialize  $w$ ;
   $x = \text{Forward}(w)$ ;
  repeat
     $\frac{\partial e_{yw}}{\partial w} = \text{BACKWARD}(x, w)$ ;
     $w = w - \lambda \cdot \frac{\partial e_{yw}}{\partial w}$ ;
     $x = \text{FORWARD}(w)$ ;
  until (a stopping criterion);
  return  $w$ ;
end

FORWARD( $w$ )
  initialize  $x(0)$ ,  $t = 0$ ;
  repeat
     $x(t+1) = F_w(x(t), l)$ ;
     $t = t + 1$ ;
  until  $\|x(t) - x(t-1)\| \leq \varepsilon_f$ 
  return  $x(t)$ ;
end

BACKWARD( $x, w$ )
   $o = G_w(x, l_N)$ ;
   $A = \frac{\partial F_w}{\partial x}(x, l)$ ;
   $b = \frac{\partial e_{yw}}{\partial o} \cdot \frac{\partial G_w}{\partial x}(x, l_N)$ ;
  initialize  $z(0)$ ,  $t=0$ ;
  repeat
     $z(t) = z(t+1) \cdot A + b$ ;
     $t = t - 1$ ;
  until  $\|z(t-1) - z(t)\| \leq \varepsilon_b$ ;
   $c = \frac{\partial e_{yw}}{\partial o} \cdot \frac{\partial G_w}{\partial w}(x, l_N)$ ;
   $d = z(t) \cdot \frac{\partial F_w}{\partial w}(x, l)$ ;
   $\frac{\partial e_{yw}}{\partial w} = c + d$ ;
  return  $\frac{\partial e_{yw}}{\partial w}$ ;
end
```

## 1-2 Method

Linear GNN: • 구조가 단순하고 수렴이 보장되지만, 표현력이 제한적

$$x_n = \sigma \left( A_w \sum_{j \in ne[n]} x_j + b_w(l_n, l_{co[n]}) \right)$$

Nonlinear GNN: • 표현력은 강하지만, 수렴성이 항상 보장되지 않기 때문에 패널티 항(P)을 추가하여 수렴성 보장

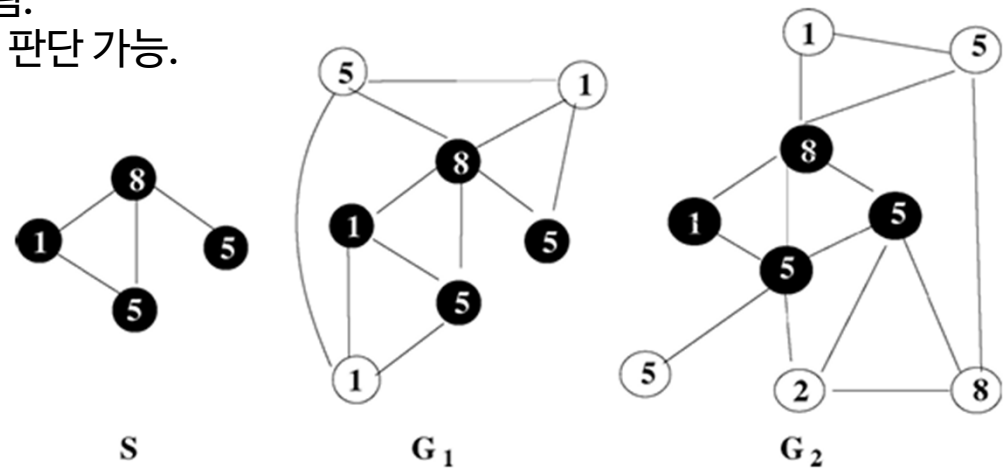
$$x_n = f_{\theta}(l_n, \{x_j, l_{nj}\})$$

$$P = \lambda \max(0, \|J_F\| - \mu)$$

## 1-3 Experiments and Results

### 실험 1: Subgraph Matching

- **목표:** 큰 그래프 내에서 특정 서브그래프의 노드를 식별.
- **설정:** 노드 라벨에 노이즈를 추가한 후, GNN이 구조적 정보와 라벨을 동시에 활용해 서브그래프를 인식하는지 평가.
- **결과:**
  - GNN이 일반적인 모델보다 높은 정확도를 보임.
  - 비선형 GNN이 선형 GNN보다 약간 더 우수.
  - 라벨 수가 적을수록(구분이 쉬울수록) 성능이 향상됨.
  - GNN은 구조적 정보(연결성)를 활용하여 더 정교한 판단 가능.



- 두 개의 큰 그래프( $G_1, G_2$ ) 안에 동일한 서브그래프가 포함되어 있음  
→ GNN은 이 구조 패턴을 이용해 해당 서브그래프에 속하는 노드를 식별하는 문제를 학습

## 1-3 Experiments and Results

### 실험 2: Mutagenesis 데이터셋

• **목표:** 화학 분자의 구조를 그래프로 표현하고, 해당 분자가 특정 생물학적 활동을 보이는지 예측.

• **설정**

- Atom-Graph 형태로 입력 (원자=노드, 결합=엣지)
- 회귀 친화/비친화 subset으로 나누어 일반화 성능 차이 비교

• **결과:**

- GNN이 기존 모델보다 높은 정확도를 달성.
- 특히 비선형 GNN이 복잡한 분자 구조를 더 잘 학습함
- 구조 기반 패턴을 스스로 학습하는 능력을 실증적으로 확인

Method	Knowledge	Reference	Accuracy
non-linear GNN	AB+C+PS		96.0%
$1nn(d_m)$	AB	[87]	72%
$1nn(d_m)$	AB+C	[87]	72%
TILDE	AB	[92]	85%
TILDE	AB+C	[92]	79%
RDBC	AB	[88]	79%
RDBC	AB+C	[88]	79%

➤ 비선형 GNN이 다른 모든 방법 대비 최고 정확도(96%) 달성

## 1-3 Experiments and Results

### 실험 3: 웹 페이지 랭킹

•목표: 웹 페이지 간의 연결 구조를 그래프로 표현하고, 페이지의 중요도를 예측.

•설정

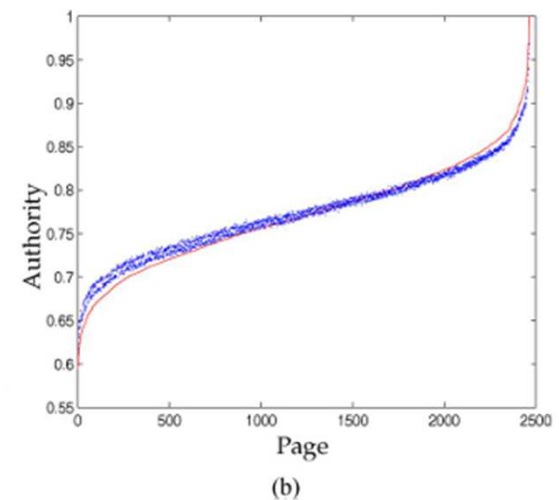
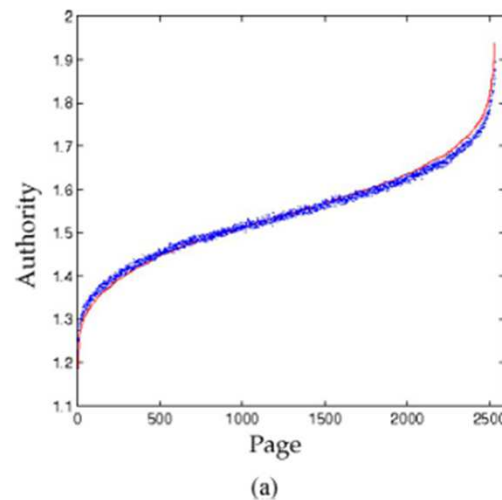
- 5000 node 랜덤 web graph 생성.
- Train / Val / Test 모두 서로 다른 node로 구성.
- 50개 node만 라벨이 주어진 상태에서 학습

•결과:

- GNN이 링크 구조를 효과적으로 학습하여 페이지 랭킹을 예측.
- 기존의 Random Walk 기반 모델보다 더 유연하고 학습 가능한 구조 제공.

➤ 두 곡선이 거의 일치

->GNN의 예측 결과가 실제 정답과 거의 동일하다는 의미



### Implications

- 여러 일반적인 그래프를 처리할 수 있는 GNN(Graph Neural Network) 모델을 최초로 제안.
- 정보 확산 메커니즘을 기반으로 견고한 이론을 제시.
- Almeida-Pineda 알고리즘을 적용하여 메모리 효율적인 학습 방법을 고안.
- 성능 입증: 다양한 실제 문제에서 SOTA 성능을 보이며 GNN의 유효성을 입증.

# 감사합니다

---

○ 2025.11.7 ○

**Presenter:** 박범도 | **소속:** EcoAI Lab | **E-mail:** [pbeomdo@gmail.com](mailto:pbeomdo@gmail.com)

**팀원:** [박범도, 박준성, 장현석]