Ground Shaking Prediction

with Graph Nueral Network

Phiphat Chomchit

August 6, 2021

Chiang Mai University

Basic Seismic Knowlendge

1. **Earthquake** is the sudden fracture and movement of rocks inside the Earth. Part of the energy released produces seismic waves, like P, S, and surface waves that travel outward in all directions from the point of initial rupture.
2. **Hypocenter** or Focus the point below the epicenter at which an earthquake begins.
3. **Epicenter** the point (map location) on the Earth's surface directly above the hypocenter or focus of an earthquake.

stick-slip model

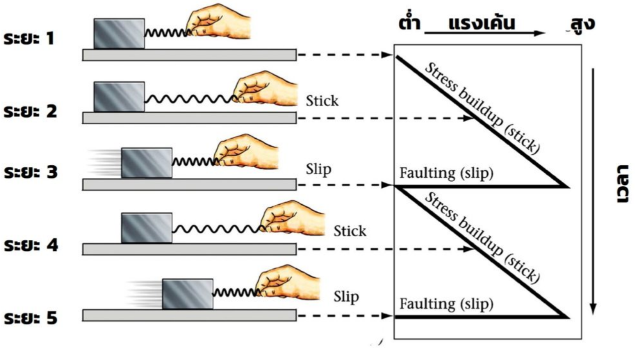


Figure 1: sick-slip model

4. **P Wave** is the primary body wave; the first seismic wave detected by seismographs; able to move through both liquid and solid rock. Also called compressional or longitudinal waves, they compress and expand (oscillate) the ground back and forth in the direction of travel, like sound waves that move back and forth as the waves travel from source to receiver. P wave is the fastest wave.
5. **S Waves** is shear waves that are secondary body waves that oscillate the ground perpendicular to the direction of wave travel. They travel about 1.7 times slower than P waves. Because liquids will not sustain shear stresses, S waves will not travel through liquids like water, molten rock, or the Earth's outer core. S waves produce vertical and horizontal motion in the ground surface.

Body wave and surface wave

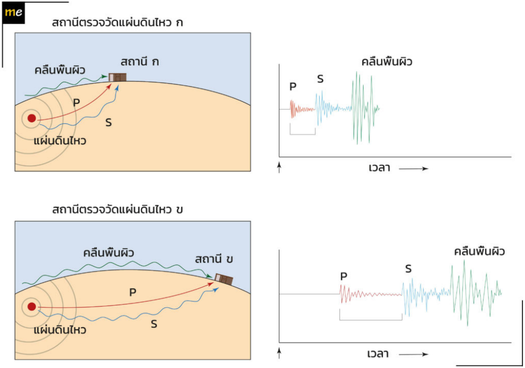


Figure 2: P-wave, S-wave and surface wave

Epicenter and Hypocenter

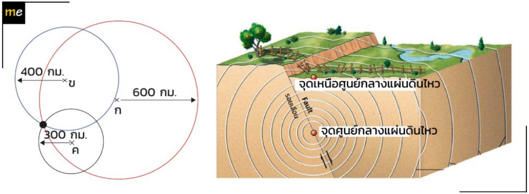


Figure 3: Epicenter and Hypocenter

$V = \frac{S}{T}$

$$T_p - T_s = \frac{S}{V_p} - \frac{S}{V_s}$$

6. **Peak Ground Acceleration** is a largest increase in velocity recorded by a particular station during an earthquake. (Commonly called PGA)

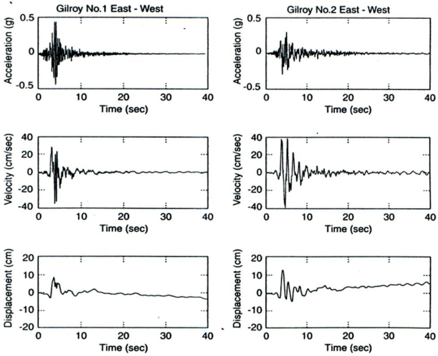


Figure 4: PGA, PGV, PGD

7. **Attenuation** is decrease in wave size, or amplitude, away from source. When you throw a pebble in a pond, it makes waves on the surface that move out from the place where the pebble entered the water. The waves are largest where they are formed and get smaller as they move away.

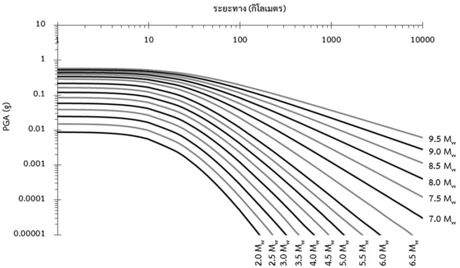


Figure 5: Attenuation

8. **Ground motion prediction model (GMPEs)**, also called ground-motion models (GMMs) and attenuation relations, estimate the shaking (strong ground motion) that may occur at a site if an earthquake of a certain magnitude occurs at a nearby location.

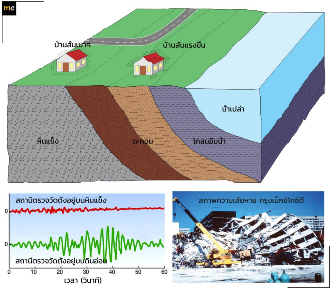


Figure 6: Ground Shaking

9. **ShakeMap** The system is designed to combine instrumental measurements with information about local seismic site conditions and the earthquake source to estimate continuous shaking variations throughout a spatial area. ShakeMap was implemented in other high-to-moderate-risk areas where rapid assessment of earthquakes is critical.

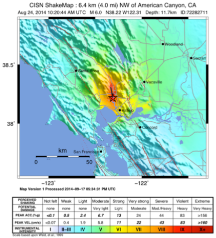


Figure 7: Ground Shaking

Review Papers

Why do we need machine learning?

1. Parameters
   1.1 Multilayer Perceptron
   1.2 Hybrid
   1.3 Genertic Programing

2. Waveform
   2.1 Multilayer Perceptron
   2.2 Convolutional Neural Network
   2.3 Graph Neural Network
   2.4 Self Supervised Learning

Why do we need machine learning?

1. Seismic data
   - Imbalanced data
   - Big data (Volume Velocity and Variety)
2. Purpose of model
   - rapid warning
   - peak ground motion prediction

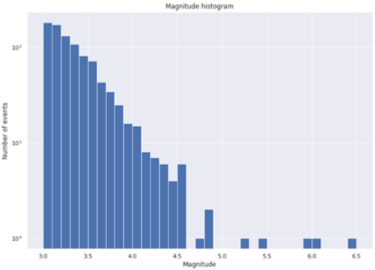


Figure 8: seismic data are long tail data.

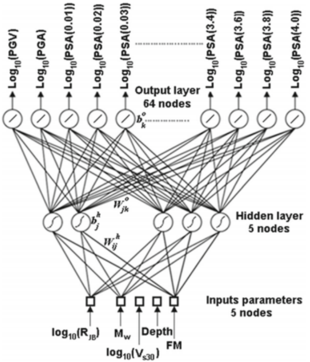


Fig. 4 Architecture of the ANN for *PGA*, *PGV* and *PSA*[0.01–4 s] prediction. The w^h_{ij} is the synaptic weight between the ith neuron of the input layer and the jth neuron in the hidden layer, b^h_j the bias of the jth neuron in the hidden layer. Also the W^o_{ij} is the synaptic weight between the jth neuron of the hidden layer and the kth neuron in the output layer, b^o_j the bias of the kth neuron in the output layer

Figure 9: PGA prediction with ANN

Hybrid

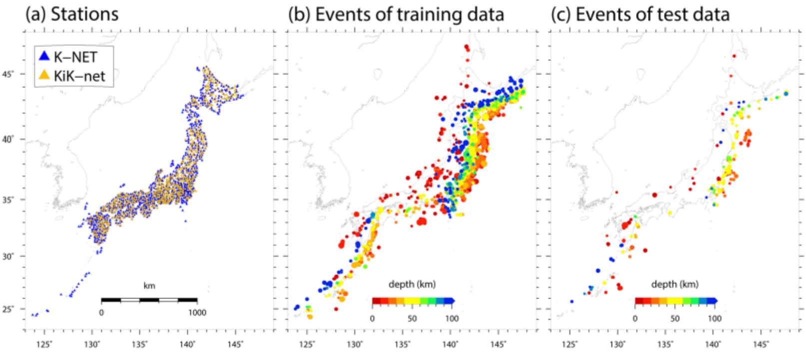


Figure 1. (**a**) Station distribution. Blue and yellow triangles indicate K-NET and KiK-net stations, respectively. (**b**) Spatial distribution of earthquake events in the training data. The circle color indicates the event depth. (**c**) Same as (**b**) but with test data.

Figure 10: Hybrid model(GMPE + ERT)

Convolutional Neural Network

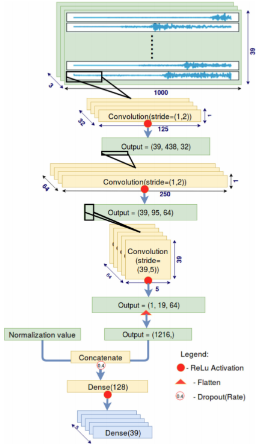


Figure 5. The architecture of the CNN model used. Boxes shaded yellow represent filter banks and operators, whereas boxes shaded green represent activations. Parenthesized vectors denote the dimensions of the outputs in question (height, width and depth).

Figure 11: CNN for rapid earthquake warning

Graph Neural Network

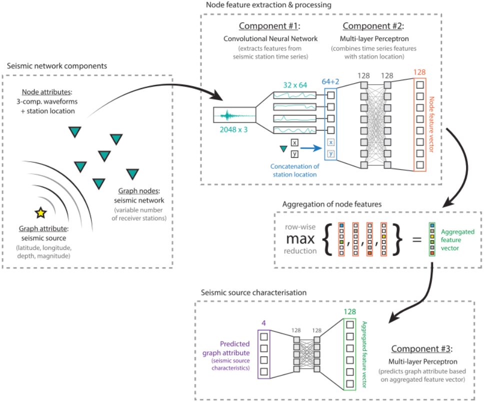


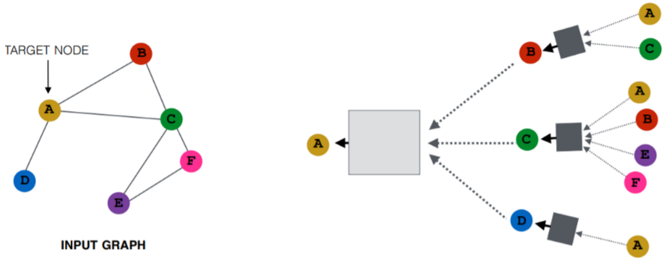
Figure 1. Synoptic overview of the adopted model architecture. The three-component waveforms from a receiver station are fed into a CNN, after which the extracted features are combined with the station's geographic location and further processed by an MLP. The resulting node feature vector of all the stations is aggregated, and this aggregated feature vector is passed through a second MLP that predicts the seismic source characteristics.

Figure 12: Characterize source of earthquake with GNN

Focus on my work

1. Graph Neural Network
   - spatial model
   - temporal model
   - static graph, dynamic signal
2. Semi-Supervised Learning and Self-Supervised Learning
   - Unlabeled data
   - Imbalanced data

Graph Neural Network.



What is Graph Neural Network?

Graph definition

A graph $\mathcal{G}$ is defined as a tuple of a set of nodes/vertices V, and a set of edges /links E: $\mathcal{G} = (V, E)$. Each edge is a pair of two vertices, and represents a connection between them.

For instance, let's look at the following graph:

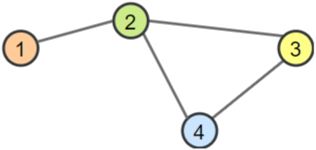


Figure 13: Example graph

The vertices are $V = \{1, 2, 3, 4\}$,

Definition of Adjacency Matrix

The **adjacency matrix** A is a square matrix whose elements indicate whether pairs of vertices are adjacent, i.e. connected, or not. In the simplest case, A_{ij} is 1 if there is a connection from node i to j, and otherwise 0.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

keep in mind that A is a symmetric matrix ($A_{ij} = A_{ji}$)

Graph Convolutions

1. GCNs are similar to convolutions in images in the sense that the ”filter” parameters are typically shared over all locations in the graph.
2. At the same time, GCNs rely on message passing methods, which means that vertices exchange information with the neighbors, and send ”messages” to each other.

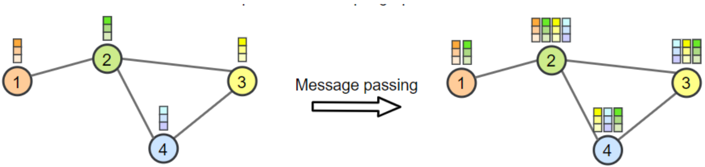


Figure 14: Message passing between nodes

Graph Convolutions

Given the previous features of nodes $H^{(l)}$, the GCN layer is defined as follows:

$$H^{(l+1)} = \sigma\left(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

1. $W^{(l)}$ is the weight parameters with which we transform the input features into messages ($H^{(l)}W^{(l)}$).
2. To the adjacency matrix A we add the identity matrix so that each node sends its own message also to itself: $\hat{A} = A + I$.
3. $\hat{D}$ which is a diagonal matrix with D_{ii} denoting the number of neighbors node i has.
4. σ represents an arbitrary activation function.

Example predict PM2.5 on graph station

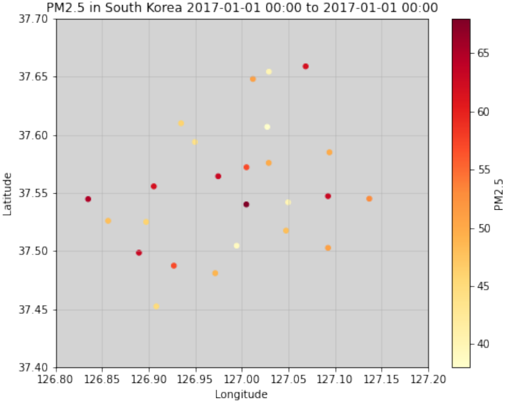


Figure 15: Predict PM2.5 with GNN

Semi-Supervised Learning and Self-Supervised Learning.

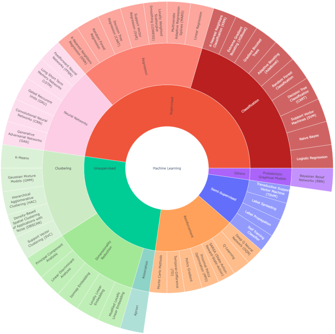


Figure 16: Machine Learning

Self Supervised Learning?

Contrastive self-supervised learning has outperformed supervised pretraining on many downstream tasks like segmentation and object detection.

What if we can get labels for free for unlabelled data and train unsupervised dataset in a supervised manner? We can achieve this by framing a supervised learning task in a special form to predict only a subset of information using the rest. In this way, all the information needed, both inputs and labels, has been provided. This is known as self-supervised learning.

How it work?

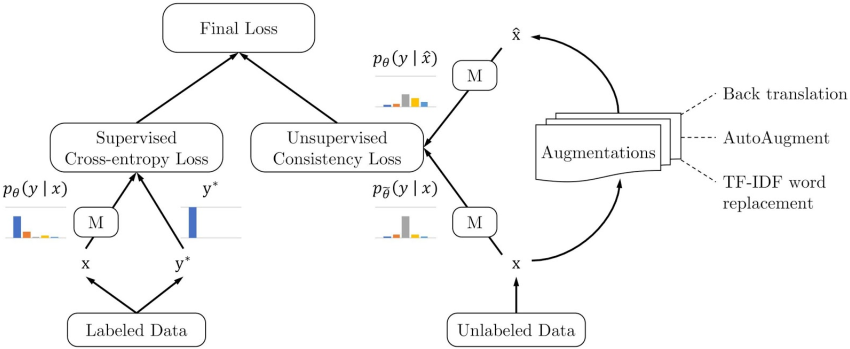


Figure 17: Semi and Self supervised learning

Data Augmentation(Image data)

1. Colorization
2. Placing image patches in the right place
3. Inpainting

Colorization

Figure 18: change color tone

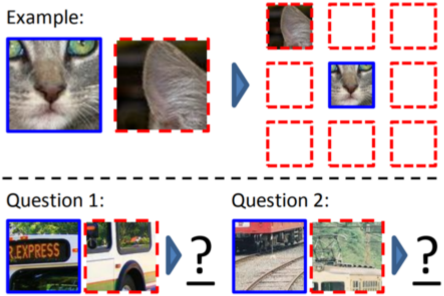


Figure 1. Our task for learning patch representations involves randomly sampling a patch (blue) and then one of eight possible neighbors (red). Can you guess the spatial configuration for the two pairs of patches? Note that the task is much easier once you have recognized the object!

Answer key: Q1: Bottom right Q2: Top center

Figure 19: Location split image

Inpainting

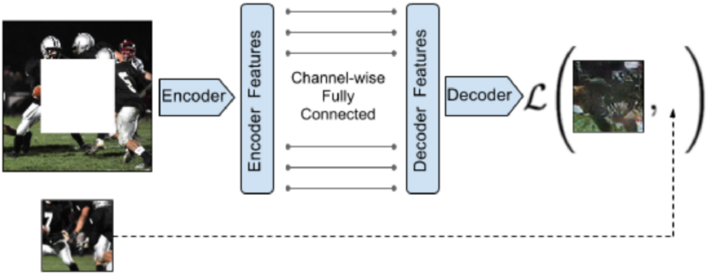


Figure 20: Autoencoder

Fomulation

$$\mathcal{L}(\theta, \eta) = \mathbb{E}_{x,\mathcal{T}}\left[\|\mathcal{F}_\theta(\mathcal{T}(x)) - \eta_x\|_2^2\right]$$

$$\min_{\theta,\eta} \mathcal{L}(\theta, \eta)$$

1. $\mathcal{F}$ is a network parameterized by θ.
2. $\mathcal{T}$ is the augmentation.
3. x is an image.
4. The expectation $\mathbb{E}[\cdot]$ is over the distribution of images and augmentations. For the ease of analysis, here we use the mean squared error $\|\cdot\|_2^2$
5. η_x is the representation of the image x

Clustering Image with Self-Supervised Learning (Birds)

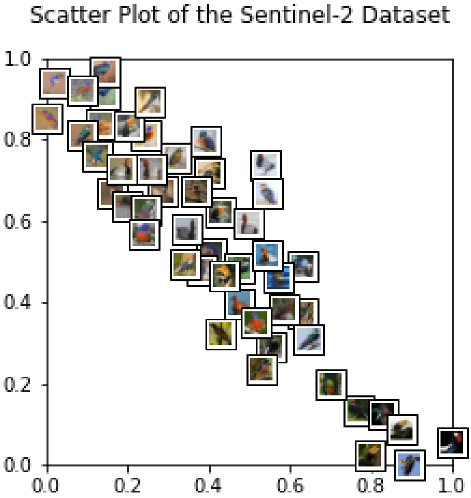


Figure 21: Embeding images on grid

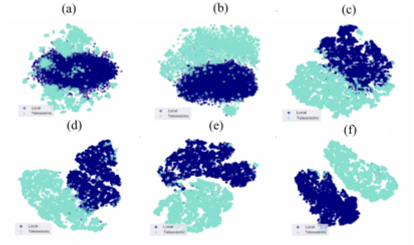


Fig. 3. (a) t-sne visualizations of local and teleseismic data in time domain, (b) time–frequency domain, (c) feature domain after the pretraining, (d) feature domain after 1350 iterations, (e) 1800 iterations, and (f) 5100 iterations during the fine-tuning. The accuracy of the k-mean clustering is given on top of each plot and samples are color-coded based on class labels. Note that as the method is trained, the two populations are increasingly divided into well-separated clusters.

Figure 22: self-supervised learning cluster local and teleseismic

Why Graph and Self-Supervised?

1. Resolve Imbalanced data problem

Table 1: Top-1 test errors (%) of ResNet-32 on long-tailed CIFAR-10 and SVHN. We compare SSL using 5x unlabeled data ($\mathcal{D}_U$@5x) with corresponding supervised baselines. Imbalanced learning can be drastically improved with unlabeled data, which is consistent across different ρ_U and learning strategies.

(a) CIFAR-10-LT

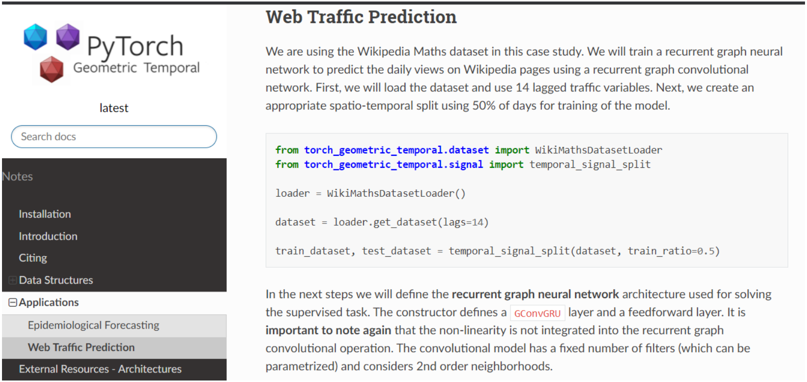
Imbalance Ratio (ρ)	100				50				10			
$\mathcal{D}_U$ Imbalance Ratio (ρ_U)	1	$\rho/2$	ρ	2ρ	1	$\rho/2$	ρ	2ρ	1	$\rho/2$	ρ	2ρ
CE	29.64				25.19				13.61			
CE + $\mathcal{D}_U$@5x	**17.48**	18.42	18.74	20.06	**16.79**	16.88	18.36	19.94	**10.22**	10.48	10.86	11.04
LDAM-DRW [7]	22.97				19.06				11.84			
LDAM-DRW + $\mathcal{D}_U$@5x	**14.96**	15.18	15.33	15.55	**14.33**	14.70	14.93	15.24	8.72	**8.24**	8.68	8.97

(b) SVHN-LT

Imbalance Ratio (ρ)	100				50				10			
$\mathcal{D}_U$ Imbalance Ratio (ρ_U)	1	$\rho/2$	ρ	2ρ	1	$\rho/2$	ρ	2ρ	1	$\rho/2$	ρ	2ρ
CE	19.98				17.50				11.46			
CE + $\mathcal{D}_U$@5x	**13.02**	13.73	14.65	15.04	**13.07**	13.36	13.16	14.54	**10.01**	10.20	10.06	10.71
LDAM-DRW [7]	16.66				14.59				10.27			
LDAM-DRW + $\mathcal{D}_U$@5x	**11.32**	11.70	11.92	12.78	**10.98**	11.14	11.26	11.51	8.94	9.08	**8.70**	9.35

Figure 23: SSL and imbalanced data

2. Represent Temporal and Spatial data


PyTorch Geometric Temporal

latest

Search docs

Notes

Installation

Introduction

Citing

Data Structures

Applications

Epidemiological Forecasting

Web Traffic Prediction

External Resources - Architectures

Web Traffic Prediction

We are using the Wikipedia Maths dataset in this case study. We will train a recurrent graph neural network to predict the daily views on Wikipedia pages using a recurrent graph convolutional network. First, we will load the dataset and use 14 lagged traffic variables. Next, we create an appropriate spatio-temporal split using 50% of days for training of the model.

```
from torch_geometric_temporal.dataset import WikiMathsDatasetLoader
from torch_geometric_temporal.signal import temporal_signal_split

loader = WikiMathsDatasetLoader()

dataset = loader.get_dataset(lags=14)

train_dataset, test_dataset = temporal_signal_split(dataset, train_ratio=0.5)
```

In the next steps we will define the **recurrent graph neural network** architecture used for solving the supervised task. The constructor defines a `GConvGRU` layer and a feedforward layer. It is **important to note again** that the non-linearity is not integrated into the recurrent graph convolutional operation. The convolutional model has a fixed number of filters (which can be parametrized) and considers 2nd order neighborhoods.

Figure 24: web traffic prediction (GNN application example)

3. Self-supervised Training of Graph Convolutional Networks

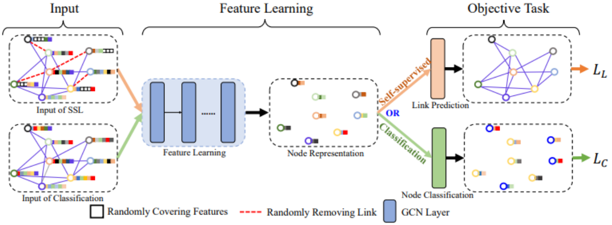


Fig. 1. Overview of the proposed self-supervised training of graph convolutional networks.

Figure 25: Self-supervised Training of Graph Convolutional Networks model

Q/A