Heterogeneous Graph Neural Network

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, Nitesh V. Chawla

KDD 2019

10/29/2019

CONTENTS

1 Background

2 HetGNN

3 Experiments

4 Conclusions

CONTENTS

1 **Background**

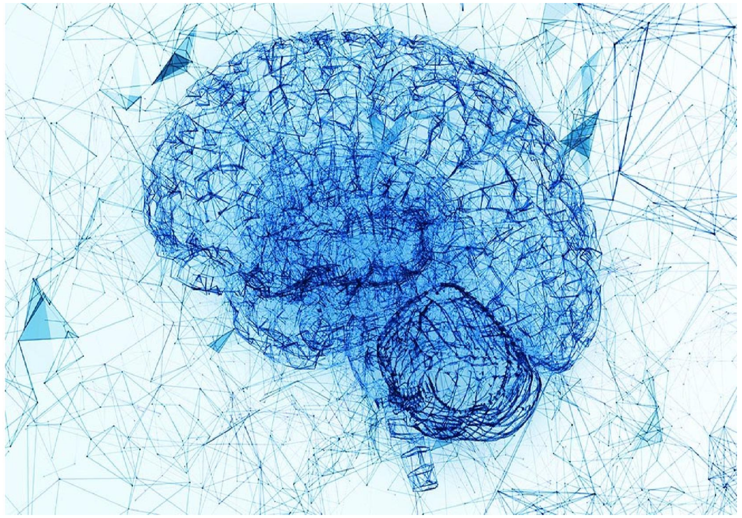
2 **HetGNN**

3 **Experiments**

4 **Conclusions**

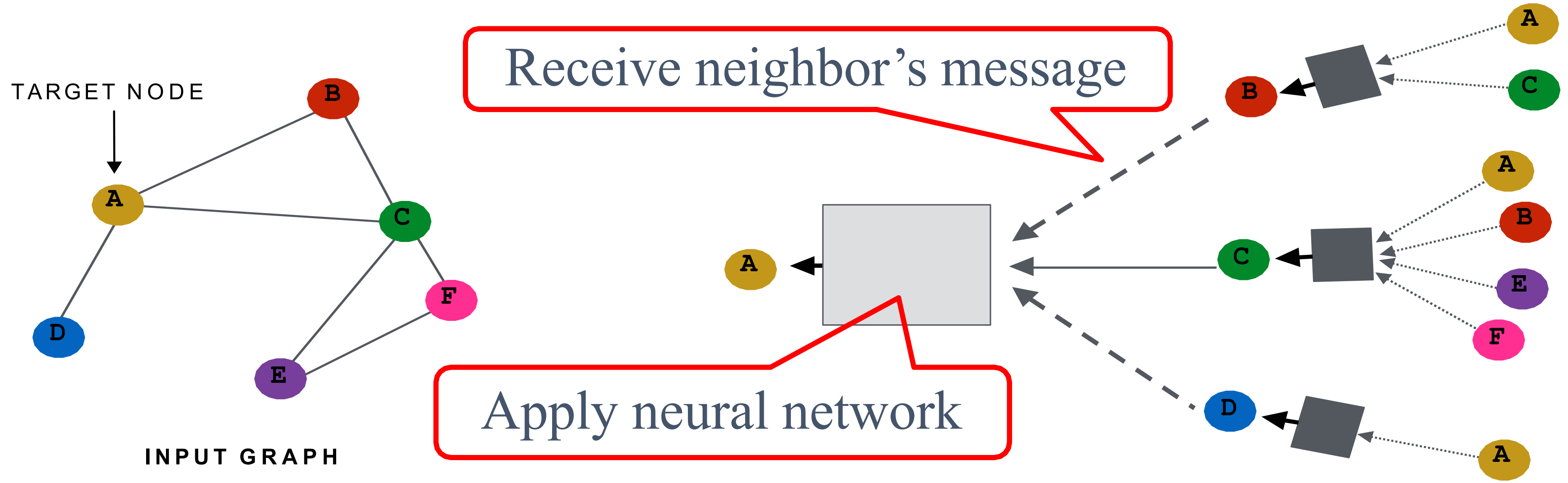
Graph-structured Data

- Graph-structured data are ubiquitous.
- Graph-structured data are flexible to model complex interactions.



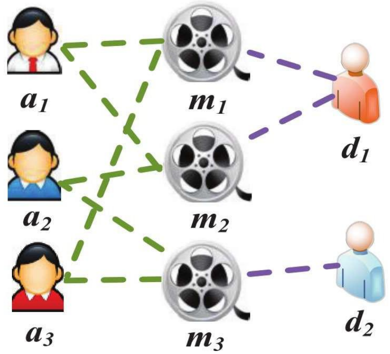
Graph Neural Network

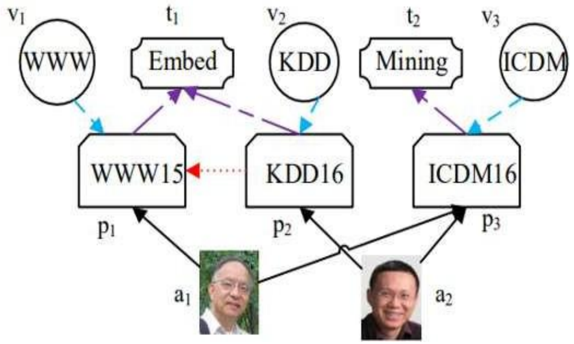
- Neural networks for processing graph-structured inputs.
- Flexible to characterize non-Euclidean data.
- For example, graph convolutional network and graph attention network.

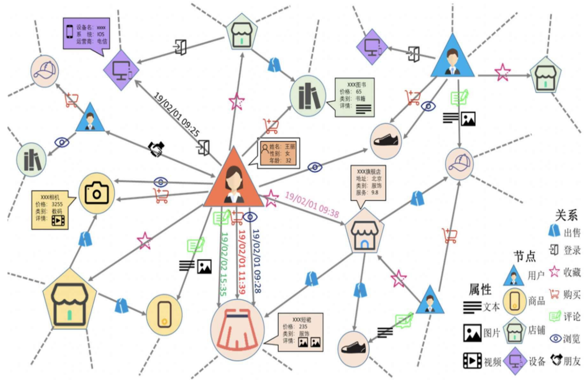


Heterogeneous Graph

- Multiple types of nodes or links





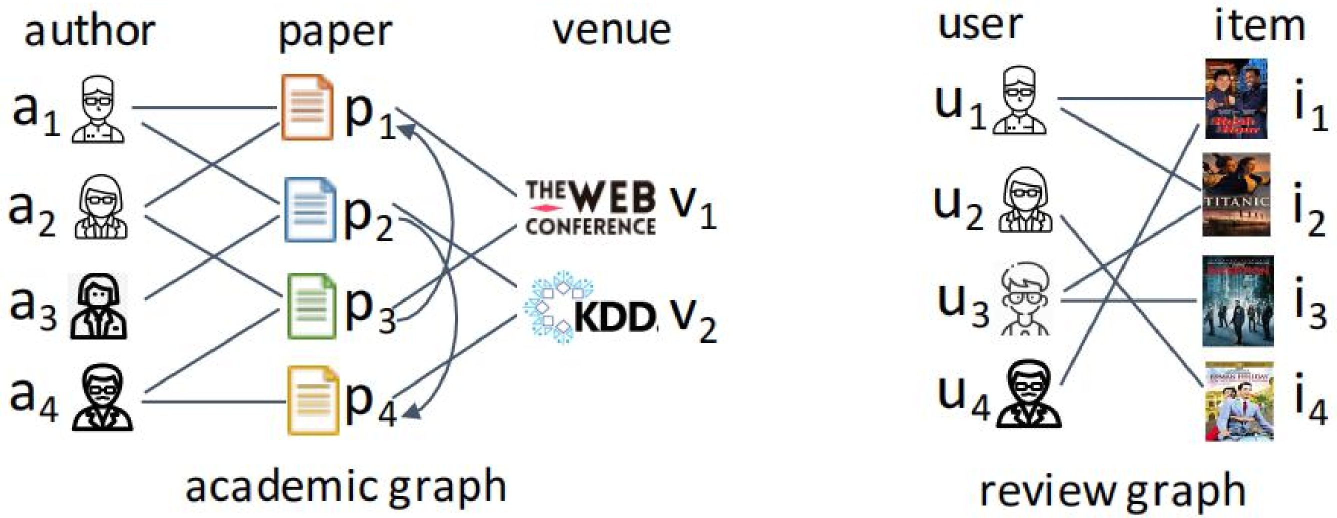


- Rich semantic information
  - Meta-path: a relation sequence connecting two objects (e.g., Movie-Actor-Movie).

Two movies directed by the same director.	Two movies are starred by the same actor.

Existing Graph Neural Networks focus on homogeneous graph

- incorporate heterogeneous structural (graph) information
- considering heterogeneous attributes or contents (e.g, text or image)

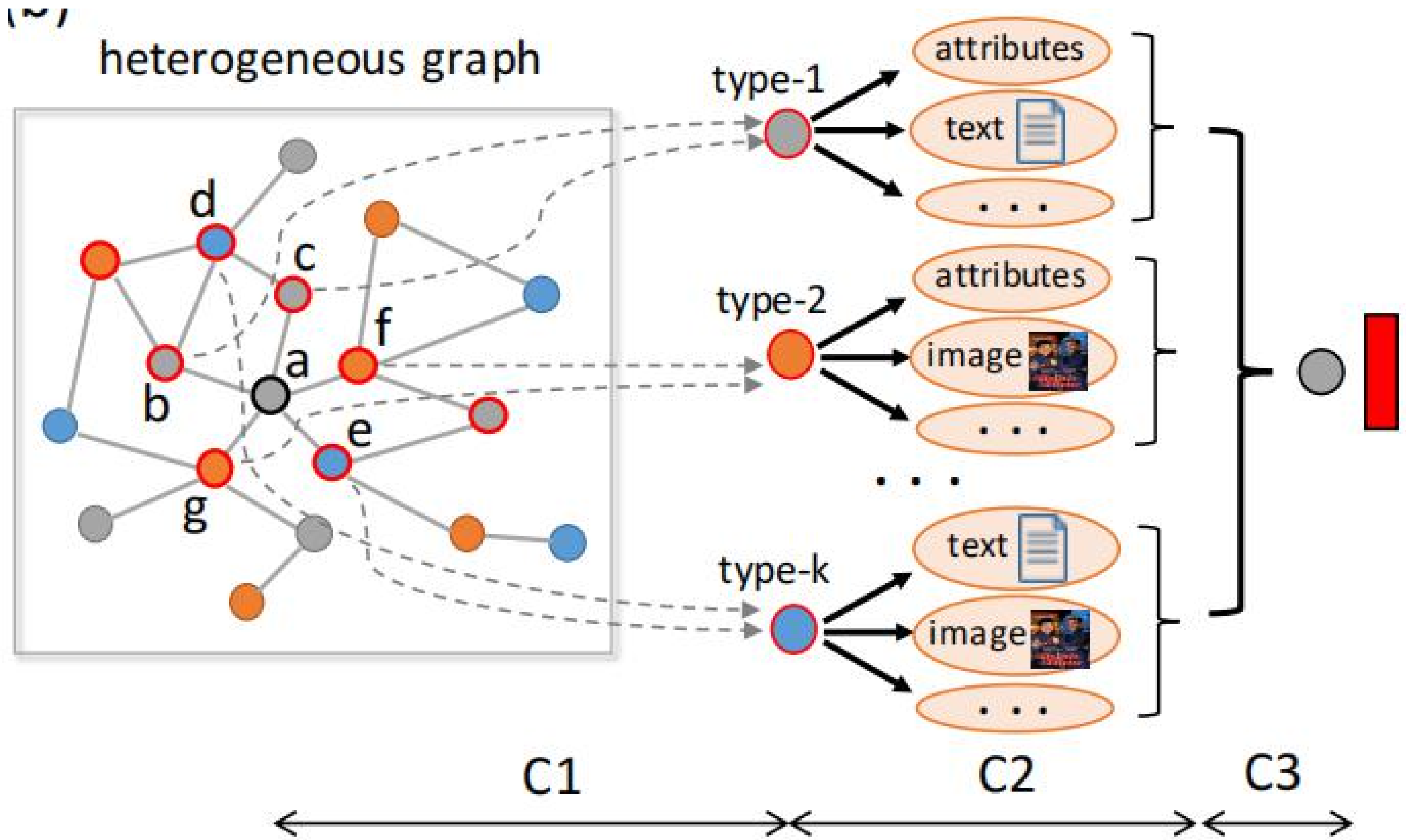


Few of them can jointly consider heterogeneous structural (graph) information as well as heterogeneous contents information of each node effectively.

CONTENTS

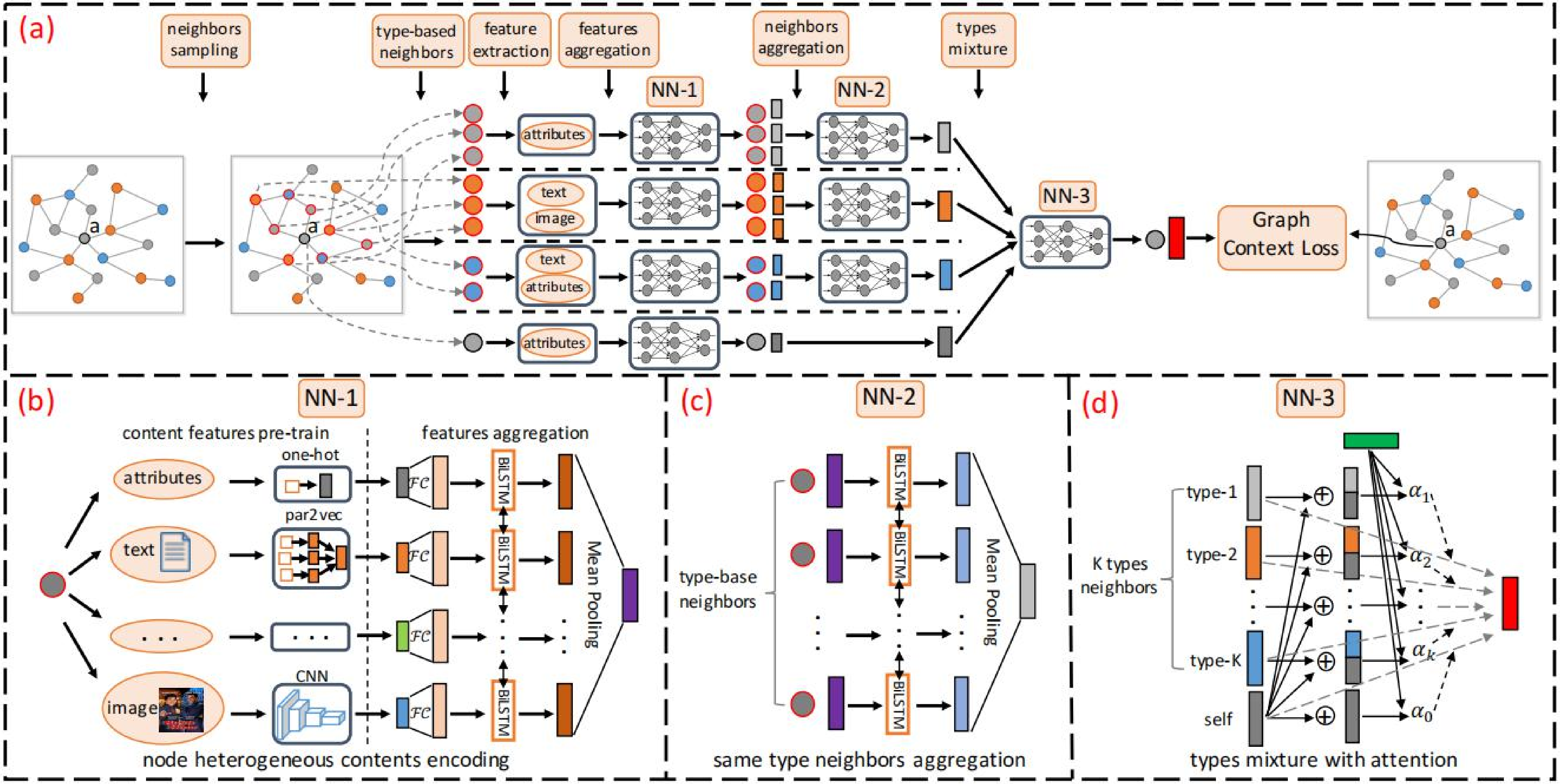
1 Background
2 **HetGNN**
3 Experiments
4 Conclusions

HetGNN



- C1 - **sampling** heterogeneous neighbors;
- C2 - **encoding** heterogeneous contents;
- C3 - **aggregating** heterogeneous neighbors.

Graph-structured Data

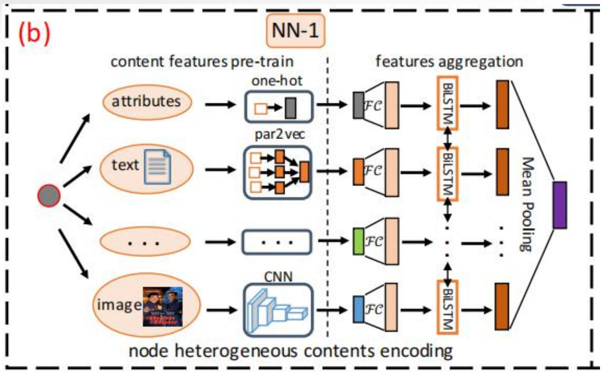


C1 - **sampling** heterogeneous neighbors;

A heterogeneous neighbors sampling strategy based on random walk with restart (RWR):

Step-1: Sampling fixed length RWR, denoted as RWR(v)

Step-2: Grouping different types of neighbors. For each node type t, we select top kt nodes from RWR(v) according to frequency



- C2 - **encoding** heterogeneous contents;

$$f_1(v) = \frac{\sum_{i \in C_v} \left[\overrightarrow{LSTM} \left\{ \mathcal{F}C_{\theta_x}(\mathbf{x}_i) \right\} \bigoplus \overleftarrow{LSTM} \left\{ \mathcal{F}C_{\theta_x}(\mathbf{x}_i) \right\} \right]}{|C_v|} \tag{1}$$

where $f_1(v) \in \mathbb{R}^{d \times 1}$ (d: content embedding dimension), $\mathcal{F}C_{\theta_x}$ denotes feature transformer which can be identity (no transformation), fully connected neural network with parameter θ_x, *etc*. The

C2 - **encoding** heterogeneous contents;

The LSTM is formulated as:

$$
\begin{aligned}
\mathbf{z}_i &= \sigma(\mathcal{U}_z \mathcal{F}C_{\theta_x}(\mathbf{x}_i) + \mathcal{W}_z \mathbf{h}_{i-1} + \mathbf{b}_z) \\
\mathbf{f}_i &= \sigma(\mathcal{U}_f \mathcal{F}C_{\theta_x}(\mathbf{x}_i) + \mathcal{W}_f \mathbf{h}_{i-1} + \mathbf{b}_f) \\
\mathbf{o}_i &= \sigma(\mathcal{U}_o \mathcal{F}C_{\theta_x}(\mathbf{x}_i) + \mathcal{W}_o \mathbf{h}_{i-1} + \mathbf{b}_o) \\
\hat{\mathbf{c}}_i &= \tanh(\mathcal{U}_c \mathcal{F}C_{\theta_x}(\mathbf{x}_i) + \mathcal{W}_c \mathbf{h}_{i-1} + \mathbf{b}_c) \\
\mathbf{c}_i &= \mathbf{f}_i \circ \mathbf{c}_{i-1} + \mathbf{z}_i \circ \hat{\mathbf{c}}_i \\
\mathbf{h}_i &= \tanh(\mathbf{c}_i) \circ \mathbf{o}_i
\end{aligned}
\tag{2}
$$

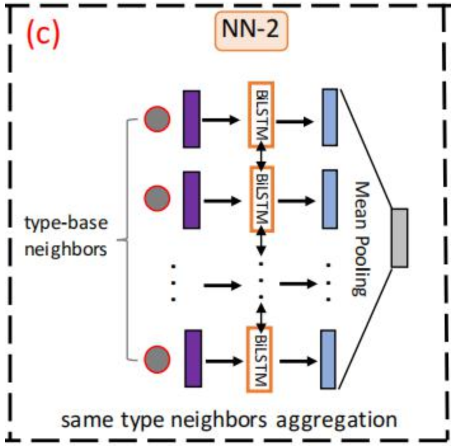
C3 - **aggregating** heterogeneous neighbors.

A type-based neural network:

(1) same type neighbors aggregation;

(2) types combination.

C3 - **aggregating** heterogeneous neighbors.



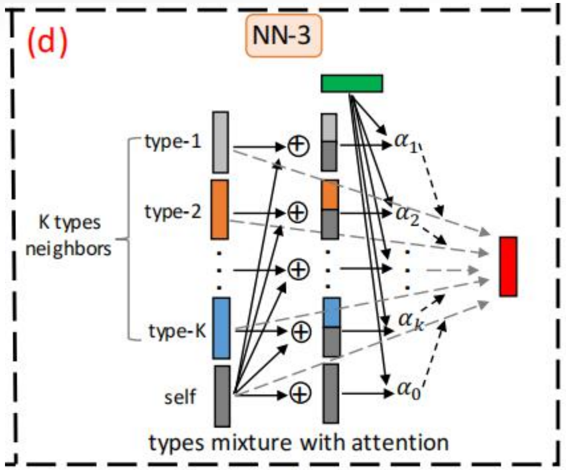
(1) same type neighbors aggregation;

$$f_2^t(v) = \mathcal{AG}^t_{v' \in N_t(v)} \{ f_1(v') \} \quad (3)$$

$$f_2^t(v) = \frac{\sum_{v' \in N_t(v)} \left[\overrightarrow{LSTM} \{ f_1(v') \} \bigoplus \overleftarrow{LSTM} \{ f_1(v') \} \right]}{|N_t(v)|} \quad (4)$$

C3 - **aggregating** heterogeneous neighbors.

(2) types combination.

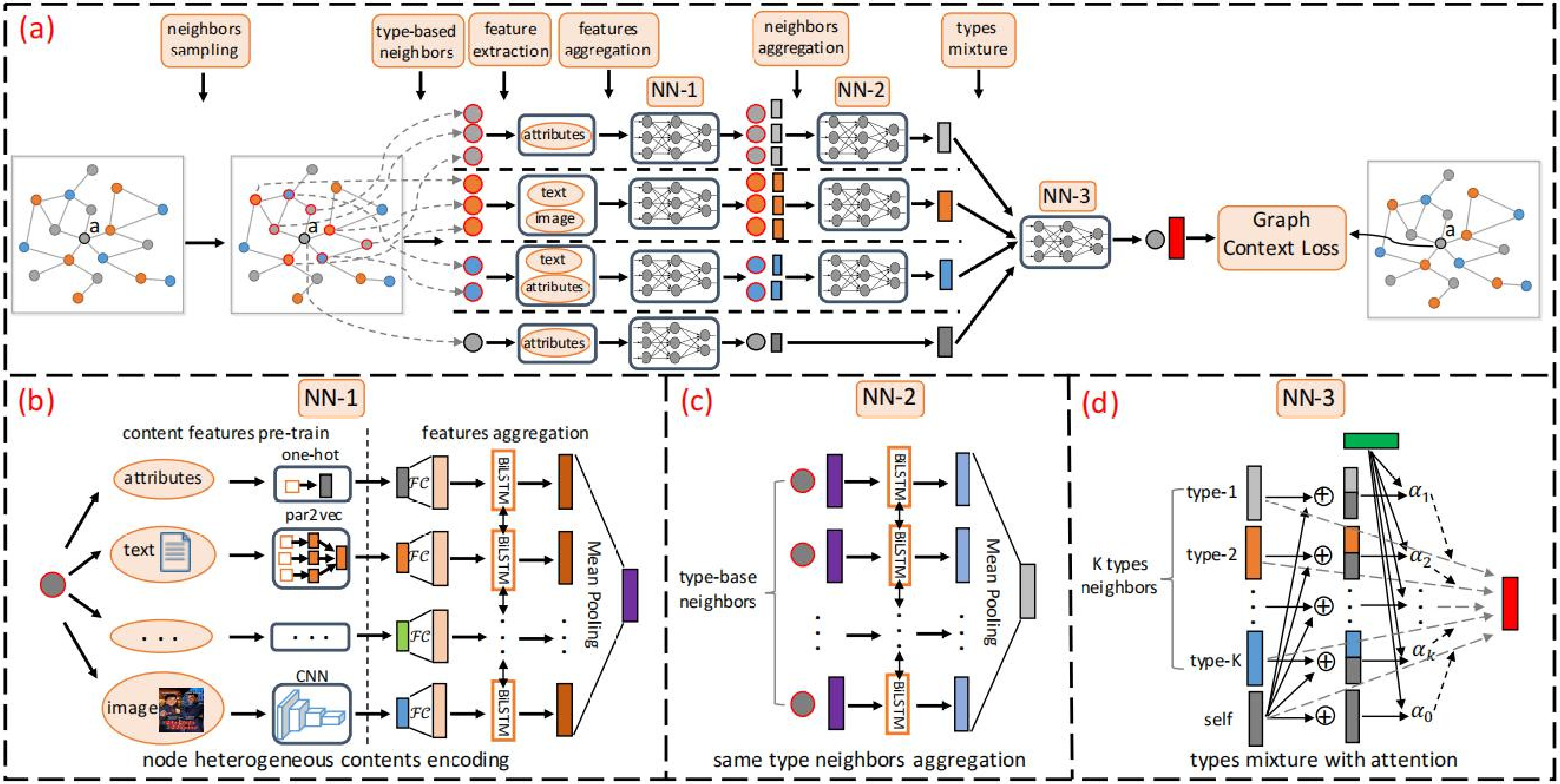


$$\mathcal{E}_v = \alpha^{v,v} f_1(v) + \sum_{t \in O_V} \alpha^{v,t} f_2^t(v) \quad (5)$$

$$\mathcal{E}_v = \sum_{f_i \in \mathcal{F}(v)} \alpha^{v,i} f_i$$

$$\alpha^{v,i} = \frac{exp\left\{LeakyReLU(u^T[f_i \bigoplus f_1(v)])\right\}}{\sum_{f_j \in \mathcal{F}(v)} exp\left\{LeakyReLU(u^T[f_j \bigoplus f_1(v)])\right\}} \quad (6)$$

Graph-structured Data



CONTENTS

1 Background

2 HetGNN

3 Experiments

4 Conclusions

Graph-structured Data

Link prediction results. Split notation in data denotes train/test data split years or ratios.

$Data_{split}$	Metric	MP2V [4]	ASNE [15]	SHNE [34]	GSAGE [7]	GAT [31]	HetGNN
A-I_{2003} (type-1)	AUC F1	0.636 0.435	0.683 0.584	0.696 0.597	0.694 0.586	0.701 0.606	**0.714** **0.620**
A-I_{2003} (type-2)	AUC F1	0.790 0.743	0.794 0.774	0.781 0.755	0.790 0.746	0.821 0.792	**0.837** **0.815**
A-I_{2002} (type-1)	AUC F1	0.626 0.412	0.667 0.554	0.688 0.590	0.681 0.567	0.691 0.589	**0.710** **0.615**
A-I_{2002} (type-2)	AUC F1	0.808 0.770	0.782 0.753	0.795 0.761	0.806 0.772	0.837 0.816	**0.851** **0.828**
A-II_{2013} (type-1)	AUC F1	0.596 0.348	0.689 0.643	0.683 0.639	0.695 0.615	0.678 0.613	**0.717** **0.669**
A-II_{2013} (type-2)	AUC F1	0.712 0.647	0.721 0.713	0.695 0.674	0.714 0.664	0.732 0.705	**0.767** **0.754**
A-II_{2012} (type-1)	AUC F1	0.586 0.318	0.671 0.615	0.672 0.612	0.676 0.573	0.655 0.560	**0.701** **0.642**
A-II_{2012} (type-2)	AUC F1	0.724 0.664	0.726 0.737	0.706 0.692	0.739 0.706	0.750 0.715	**0.775** **0.757**
R-$I_{5:5}$	AUC F1	0.634 0.445	0.623 0.551	0.651 0.586	0.661 0.542	0.683 0.665	**0.749** **0.735**
R-$I_{7:3}$	AUC F1	0.701 0.595	0.656 0.613	0.695 0.660	0.716 0.688	0.706 0.702	**0.787** **0.776**
R-$II_{5:5}$	AUC F1	0.678 0.541	0.655 0.582	0.685 0.593	0.677 0.565	0.712 0.659	**0.736** **0.701**
R-$II_{7:3}$	AUC F1	0.737 0.660	0.695 0.648	0.728 0.685	0.721 0.653	0.742 0.713	**0.772** **0.749**

Tasks: link prediction, recommendation, node classification & clustering and inductive node classification & clustering