The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)

Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks

Tian Bian,[1,2] Xi Xiao,[1] Tingyang Xu,[2] Peilin Zhao,[2] Wenbing Huang,[2] Yu Rong,[2] Junzhou Huang[2]
[1]Tsinghua University
[2]Tencent AI Lab
bt18@mails.tsinghua.edu.cn, xiaox@sz.tsinghua.edu.cn, hwenbing@126.com, yu.rong@hotmail.com
{tingyangxu, masonzhao, joehhuang}@tencent.com

AAAI'20

210812 Chia-Chun Ho

Outline

Introduction

Preliminaries

Methodology

Experiments

Conclusions

Comments

Introduction

Conventional detection methods

- Mainly adopted hand-crafted features to train supervised classifiers.
    - User characteristics, text contents, propagation patterns
    - Decision Tree, Random Forest, SVM
- Some studies apply more effective features
    - User comments (Giudice 2010)
    - Temporal-structural features (Wu, Yang, and Zhu 2015)
    - The emotional attitude of posts (Liu et al. 2015)

Introduction

Conventional detection methods

- These method mainly rely on feature engineering
  - Very time-consuming and labor-intensive
- Hand-crafted features are usually lack of high-level representation extracted from the propagation and the dispersion of rumors

Introduction
Recent Studies

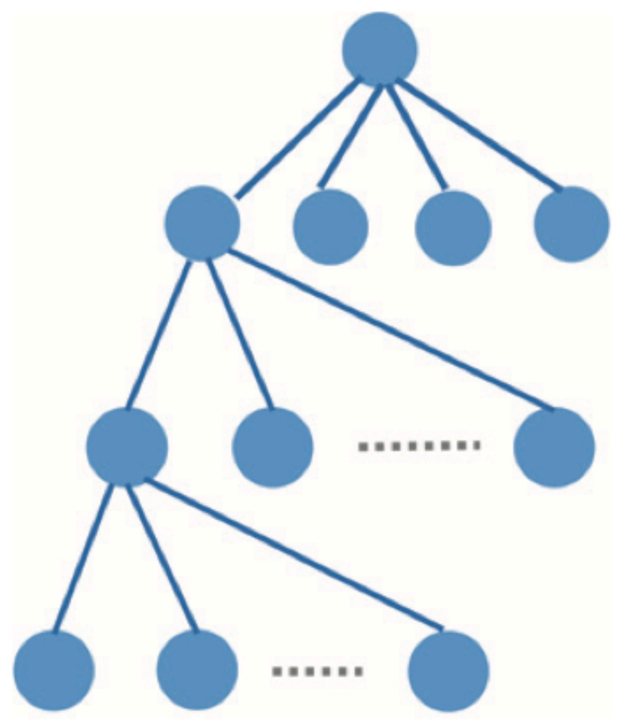
- Exploited deep learning methods that mine high-level representations from propagation path/trees or networks to identify rumors.
  - LSTM, GRU, RvNN(Recursive Neural Networks)
    - Capable to learn sequential features from rumor propagation along time
- These approaches only pay attention on sequential features from propagation of rumors but neglect the influences of rumor dispersion.
- The structures of rumor dispersion also indicate some spreading behaviors of rumors.

Introduction
Recent Studies

- Some studies have tried to involve the information from the structures of rumor dispersion by invoking CNN-based methods.
  - CNN-based methods can obtain the correlation features within local neighbors but cannot handle the global structural relationships in graphs or trees.
  - The global structural features of rumor dispersion are ignored in these approaches.
  - CNN is not designed to learn high-level representations from structured data
    - But GCN is

Introduction

GCN approaches

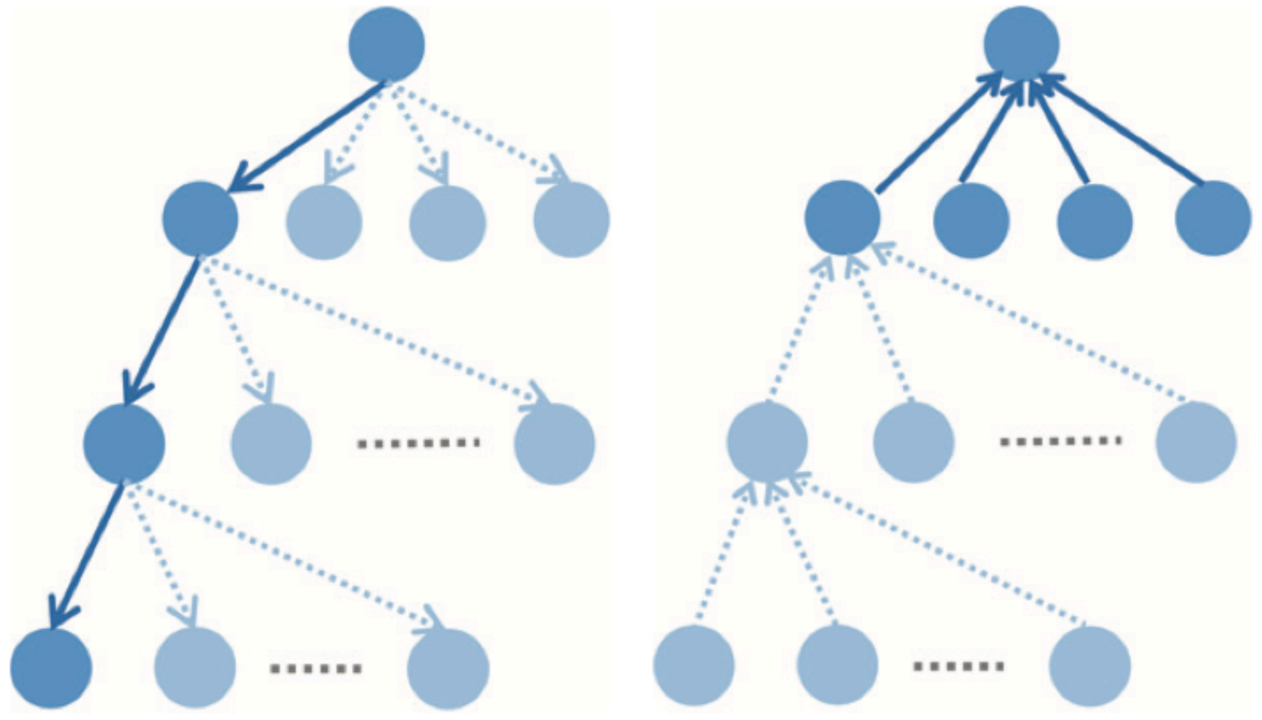


(a) UD-GCN

- GCN (Undirected GCN, UD-GCN) only aggregates information relied on the relationship among relevant posts but loses the sequential order of follows.
  - Although UD-GCN can handle the global structural features of rumor dispersion, it does not consider the direction of the rumor propagation.
- In previous work already prove two major characteristics of rumors
  - deep propagation along a relationship chain (Han et al. 2014)
  - wide dispersion across a social community (Thomas 2007)

Introduction

Bi-directional GCN (Bi-GCN)



(b) TD-GCN (c) BU-GCN

- To deal with both propagation and dispersion of rumors, proposed Bi-GCN.
- Obtains the features of
    - Propagation via Top-Down GCN (TD-GCN)
        - TD-GCN forwards information from the parent node of a node in rumor tree to formulate the rumor propagation
    - Dispersion via Bottom-Up GCN (BU-GCN)
        - BU-GCN aggregates information from the children nodes of a node in a rumor tree to represent rumor dispersion

Introduction

Bi-directional GCN (Bi-GCN)

- Then, the representations of propagation and dispersion pooled from the embedding of TD-GCN and BU-GCN are merged together through full connections to make the final result.

- Meanwhile, concatenate the features of the roots in rumor trees with the hidden features at each GCN layer to enhance the influences from the roots of rumors.

- Employ DropEdge (Rong et al. 2019) in the training phase to avoid over-fitting.

Introduction

Contributions of Bi-directional GCN (Bi-GCN)

- Leverage GCN to detect rumors.
- Proposed Bi-GCN that
  - Not only considers the causal features of rumor propagation along relationship chains from top to down
  - But also obtains the structure features from rumor dispersion within communities through the bottom-up gathering.
- Concatenate the features of the source post with other posts at each GCN to make a comprehensive use of the information from the root feature.

Preliminaries

Notation

- $C = \{c_1, c_2, \cdots, c_m\}$: rumor detection dataset, m: num of events
  - $c_i = \{r_i, w_1^i, w_2^i, \cdots, w_{n_i-1}^i, G_i\}$: i-th event, n_i: num of posts in c_i
    - r_i: source post (root node)
    - w_j^i: j-th relevant responsive post
    - $G_i \rightarrow \left\langle V_i, E_i \right\rangle$: propagation structure
      - $V_i = \{r_i, w_1^i, \cdots, w_{n_i-1}^i\}$
      - $E_i = \{e_{st}^i \mid s, t = 0, \cdots, n_i - 1\}$, i.e., $w_1^i \rightarrow w_2^i$: e_{12}^i, $r_i \rightarrow w_1^i$: e_{01}^i

Preliminaries

Notation

- $\mathbf{A}_i \in \{0,1\}^{n_i \times n_i}$: adjacency matrix where
  - $a^i_{ts} = \begin{cases} 1, & \text{if } e^i_{st} \in E_i \\ 0, & \text{otherwise} \end{cases}$
- $\mathbf{X}_i = \left[\mathbf{x}_0^{i\top}, \mathbf{x}_1^{i\top}, \ldots, \mathbf{x}_{n_i-1}^{i\top}\right]^\top$: feature matrix extracted from c_i
  - $\mathbf{x}^i_0$: feature vector of r_i
  - $\mathbf{x}^i_j$: feature vector of w^i_j

Preliminaries
Notation

- Each c_i is associated with a ground-truth label $y_i \in \{F, T\}$ (False Rumor, True Rumor)
  - In some cases, $y_i \in \{N, F, T, U\}$ (Non-rumor, False Rumor, True Rumor, Unverified Rumor)
- Given the dataset, the goal of rumor detection is to learn a classifier $f : C \to Y$

Preliminaries

Graph Convolutional Networks

- GCN is one of the most effective convolution models
    - Considered as a general "message-passing" architecture
    - $\mathbf{H}_k = M\left(\mathbf{A}, \mathbf{H}_{k-1}; \boldsymbol{W}_{k-1}\right)$: hidden feature matrix computed by k-th GCL
        - $\mathbf{A}$: adjacency matrix
        - $\mathbf{H}_{k-1}$: hidden feature matrix
        - $\boldsymbol{W}_{k-1}$: trainable parameters
        - M: message propagation function for GCN

Preliminaries

Graph Convolutional Networks

- M defined in 1stChebNet (Kipf and Welling 2017) as follow:
- $\mathbf{H}_k = M\left(\mathbf{A}, \mathbf{H}_{k-1}; \boldsymbol{W}_{k-1}\right) = \sigma\left(\hat{\mathbf{A}}\mathbf{H}_{k-1}\boldsymbol{W}_{k-1}\right)$
  - $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$: normalized adjacency matrix
  - $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$: adding self-connection
  - $\tilde{\mathbf{D}}_{ii} = \Sigma_j \tilde{\mathbf{A}}_{ij}$: degree of the i-th node

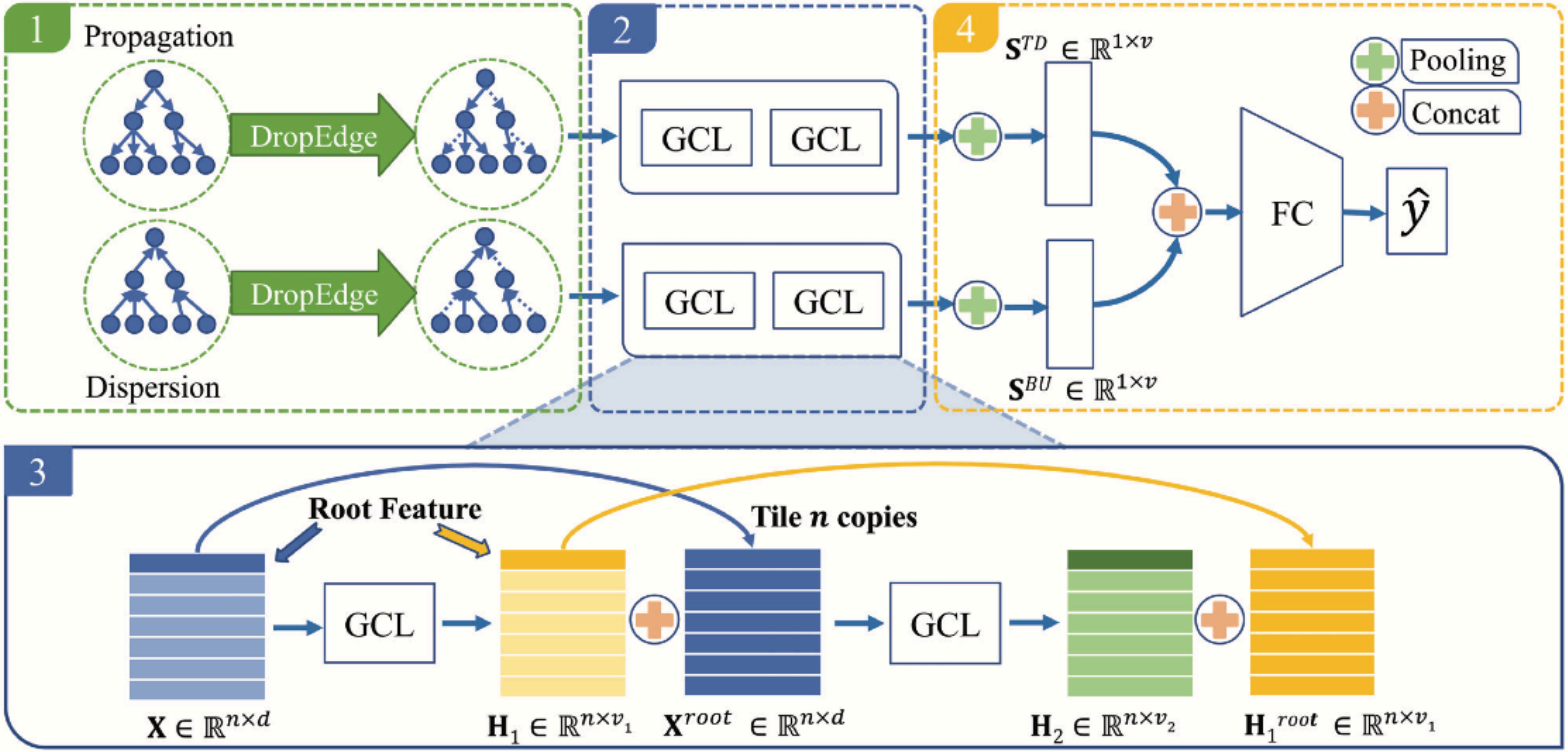
Preliminaries

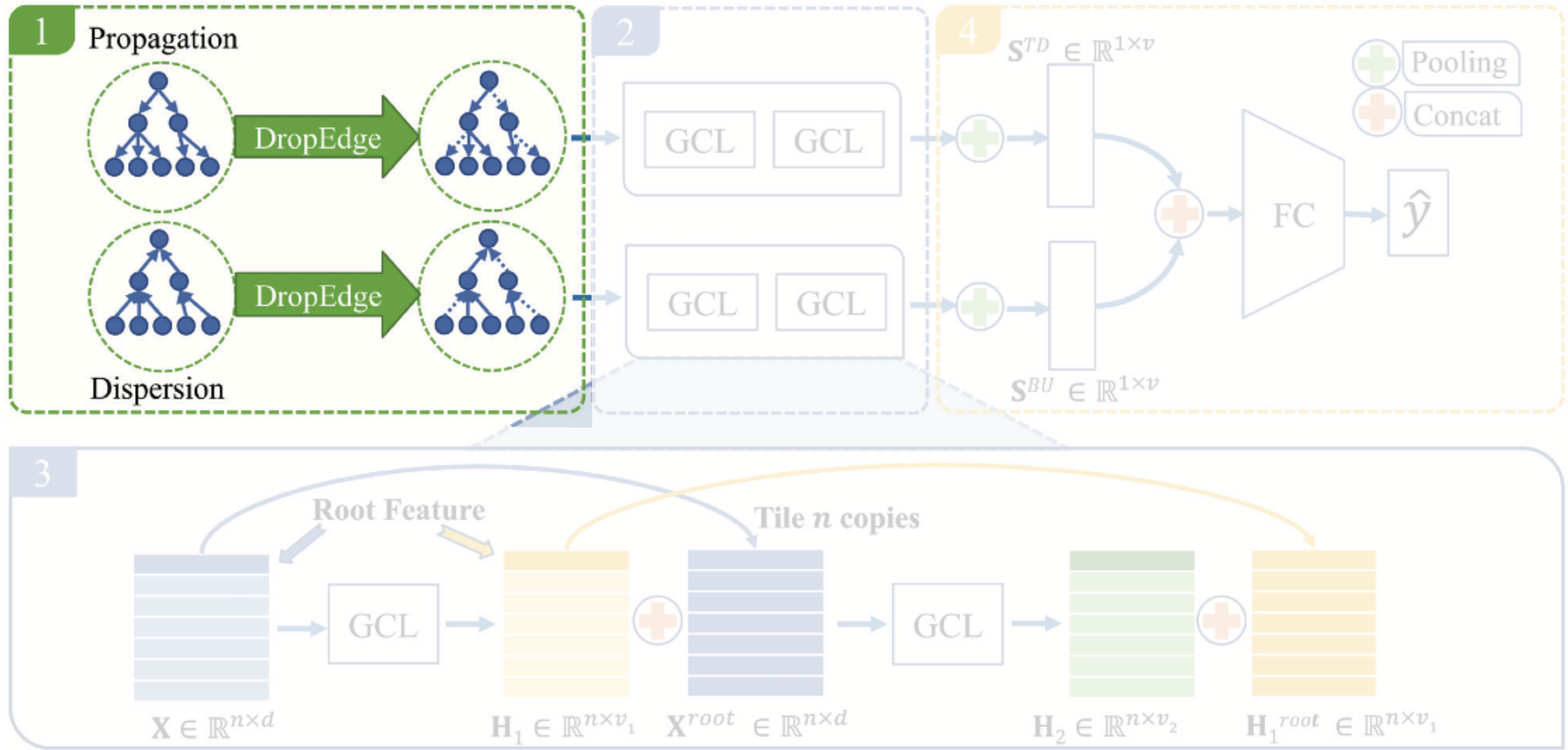
DropEdge

- Novel method to reduce over-fitting for GCN-based models (Rong et al. 2019).
- Randomly drops out edges from input graphs to generate different deformed copies with certain rate at each training epoch.
  - This method augments the randomness and the diversity of input data.
- Formally, suppose the total number of edges in the graph $\mathbf{A}$ is N_e, and the dropping rate is p
  - $\mathbf{A}' = \mathbf{A} - \mathbf{A}_{drop}$: adjacency matrix after DropEdge
  - $\mathbf{A}_{drop}$ is constructed using $N_e \times p$ edges randomly sampled from the original edge set.

Methodology

Bi-GCN Rumor Detection Model

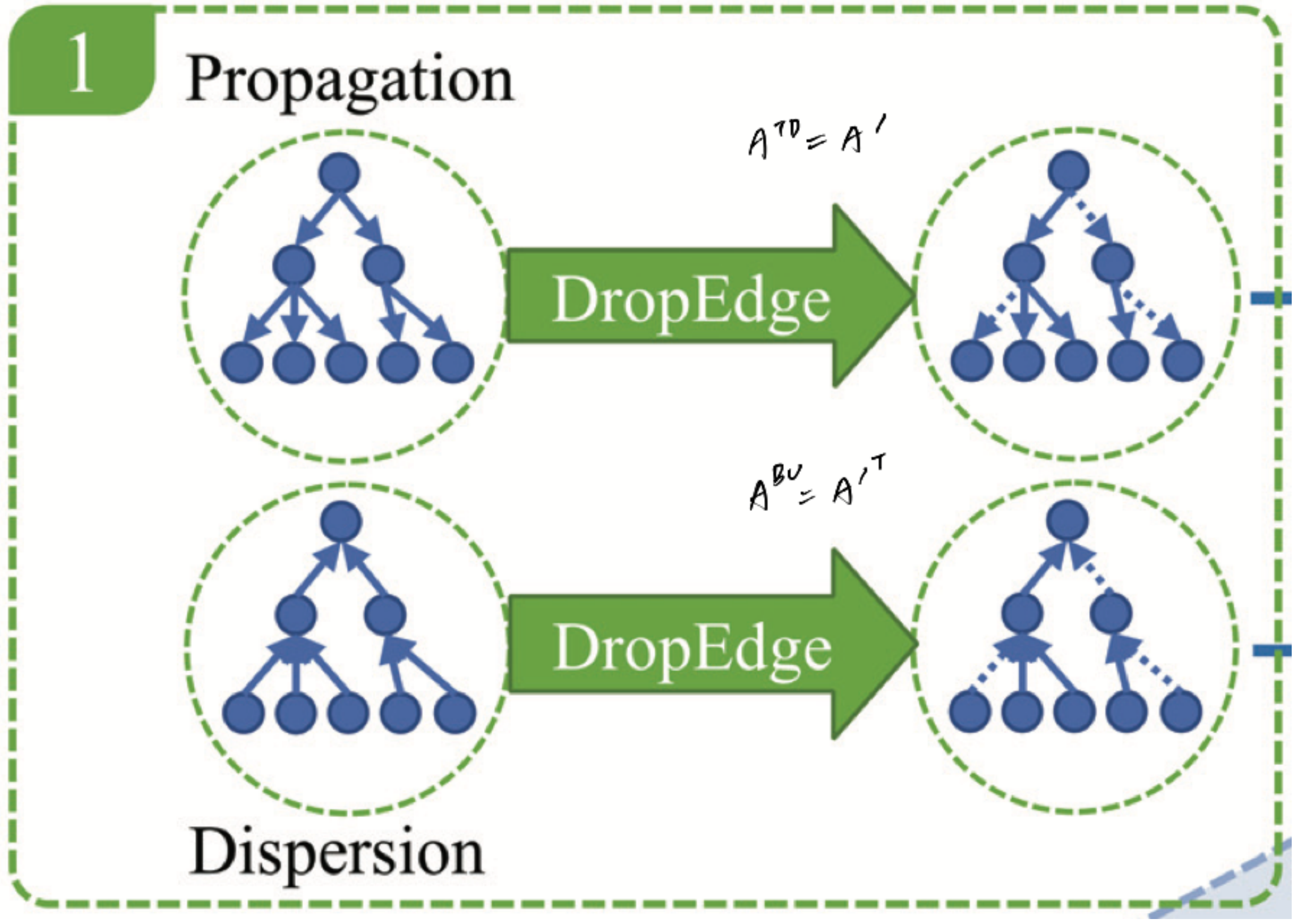




Methodology

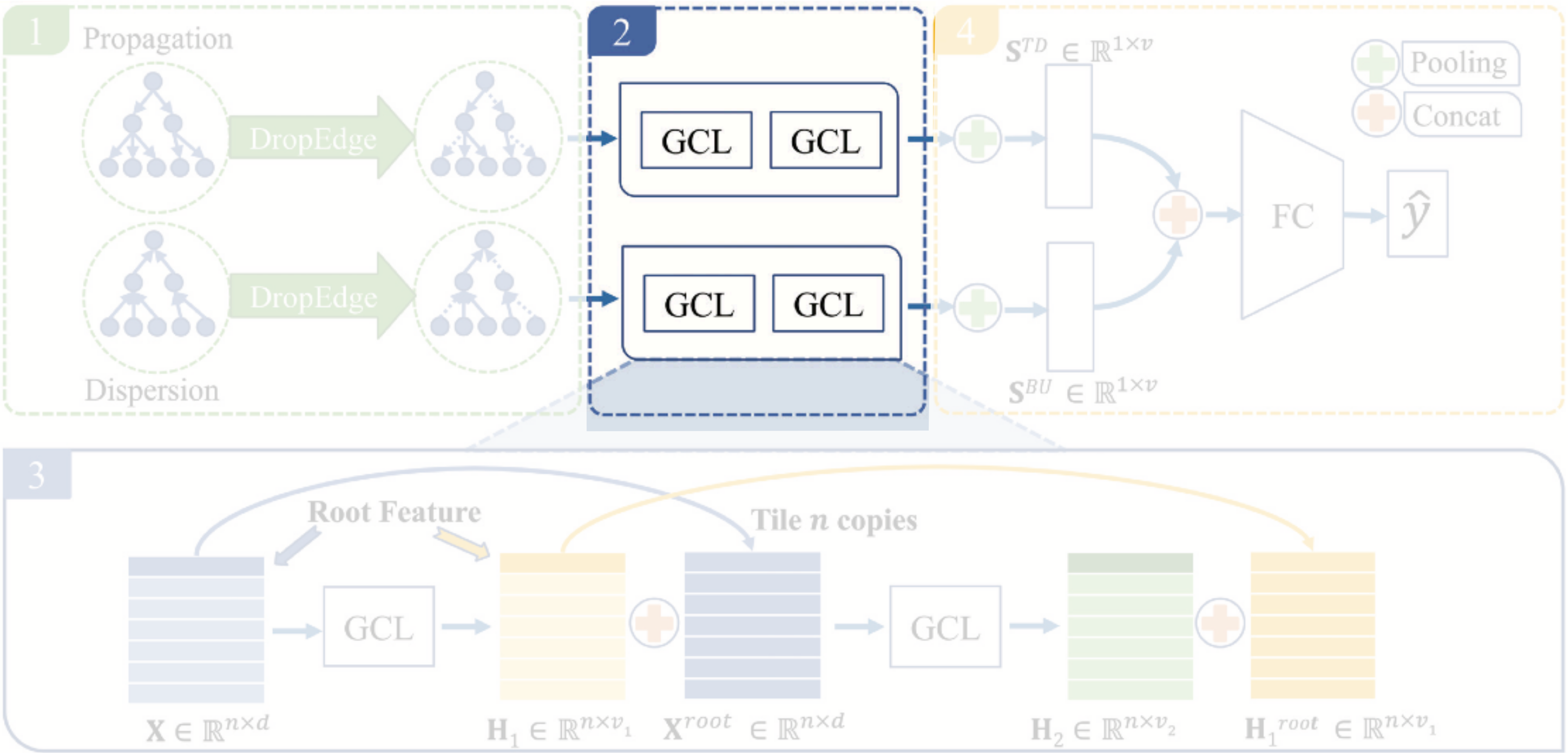
Construct Propagation and Dispersion Graphs

- $\mathbf{A}$ only contains the edges from upper nodes to lower nodes
- At each training epoch, get $\mathbf{A}'$ via DropEdge to avoid overfitting
- Bi-GCN consist of two components, the adjacency matrices are different:
  - TD-GCN: $A^{TD} = \mathbf{A}'$
  - BU-GCN: $A^{BU} = \mathbf{A}'^{\top}$
  - TD-GCN and BU-GCN adopt the same feature matrix $\mathbf{X}$.



Methodology

Calculate the High-level Node Representations

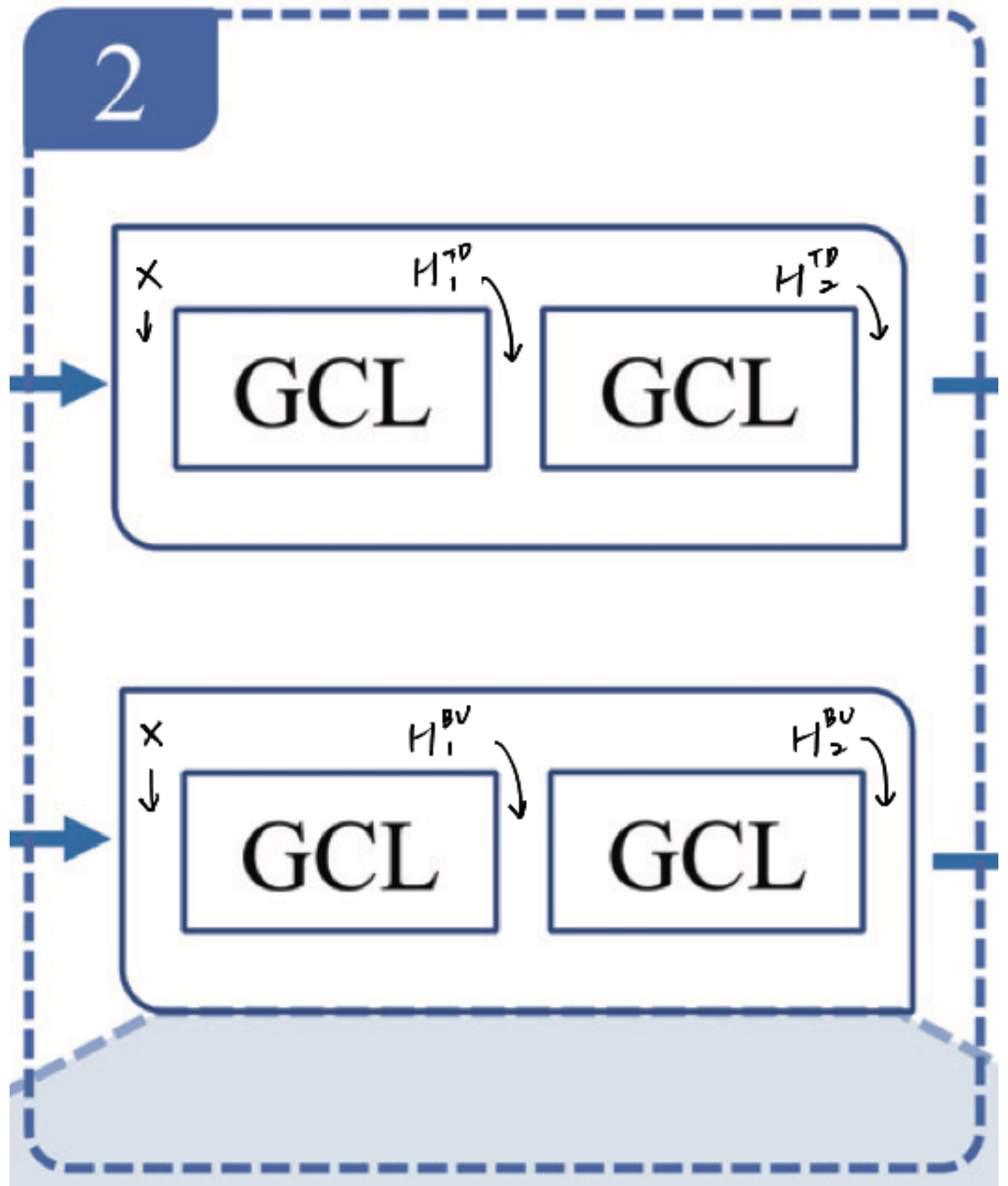


- Top-down propagation and bottom-up propagation features are obtained by TD-GCN and BU-GCN.

- TD-GCN and BU-GCN has two layers, the equations for TD-GCN as below:

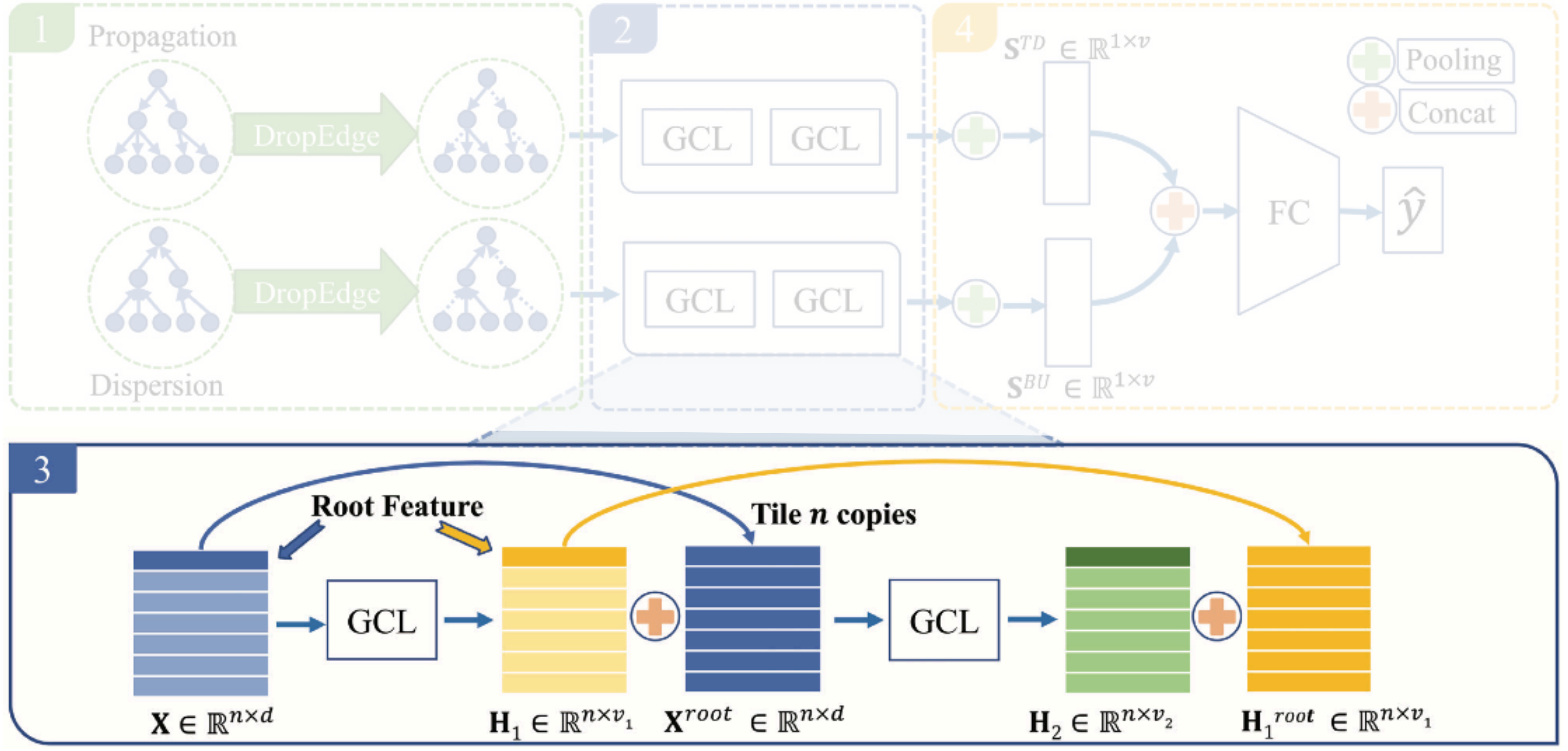
  - $\mathbf{H}_1^{TD} = \sigma\left(\hat{\mathbf{A}}^{TD}\mathbf{X}\boldsymbol{W}_0^{TD}\right)$
  - $\mathbf{H}_2^{TD} = \sigma\left(\hat{\mathbf{A}}^{TD}\mathbf{H}_1^{TD}\boldsymbol{W}_1^{TD}\right)$

- Bottom-up hidden features $\mathbf{H}_1^{BU}, \mathbf{H}_2^{BU}$ for BU-GCN in the same manner as above.



Methodology

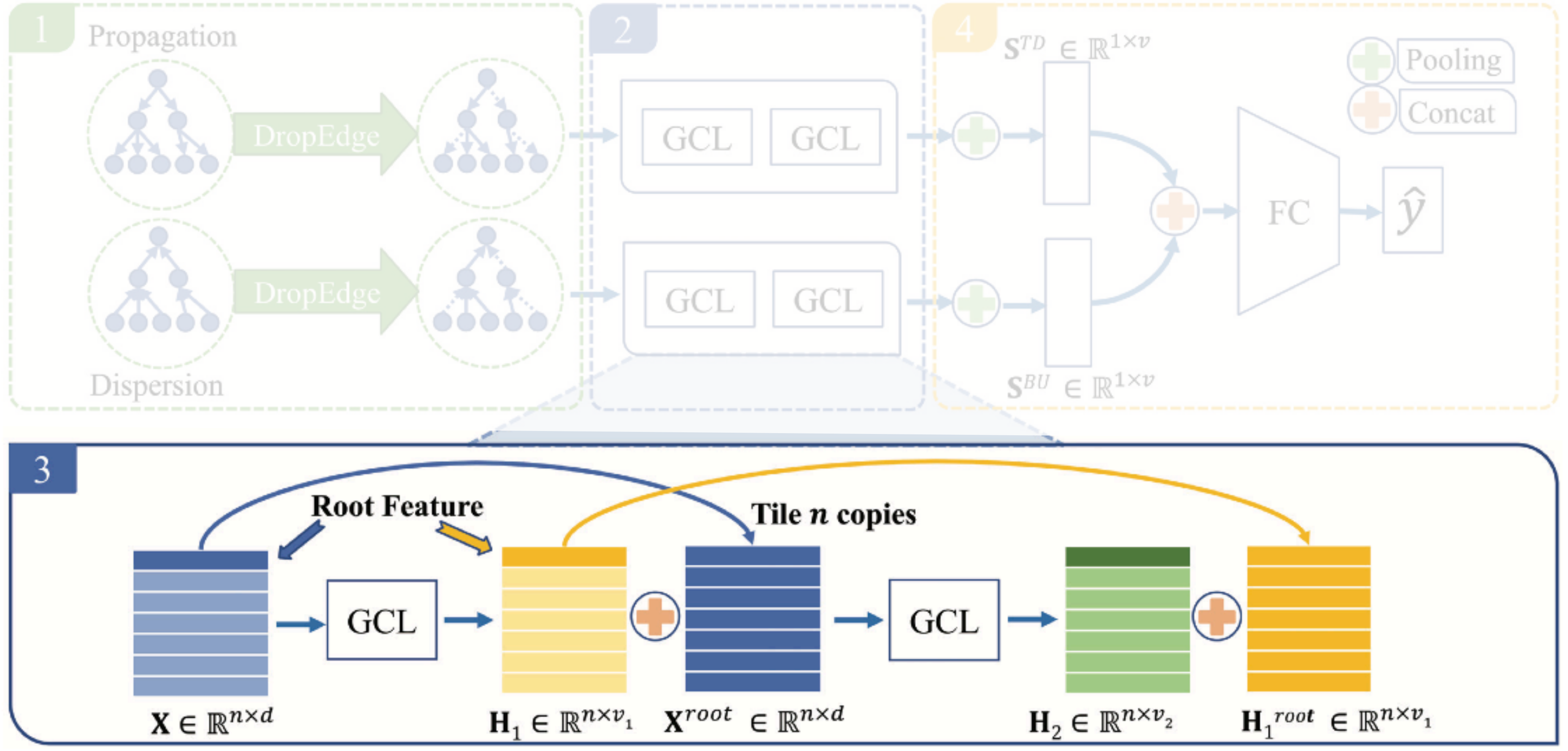
Root Feature Enhancement



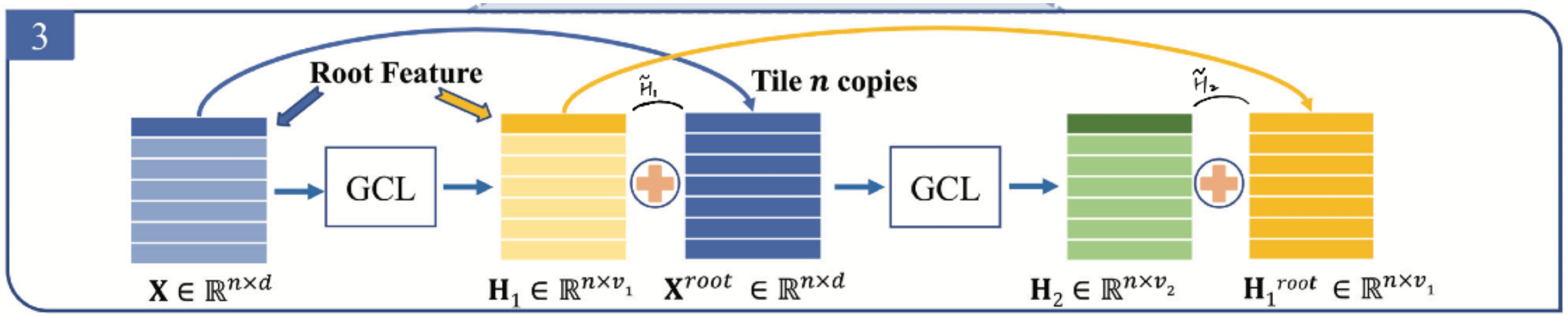
- Source post of a rumor event always has abundant information to make a wide impact.
- Proposed an operation of root feature enhancement to improve the performance of rumor detection.
- For k-th GCL, concatenate the hidden feature vectors of every nodes with the hidden feature vector of the root node from $(k-1)$-th GCL to construct new feature matrix
- $\tilde{\mathbf{H}}_k^{TD} = \text{concat}\left(\mathbf{H}_k^{TD}, \left(\mathbf{H}_{k-1}^{TD}\right)^{root}\right), \mathbf{H}_0^{TD} = \mathbf{X}$

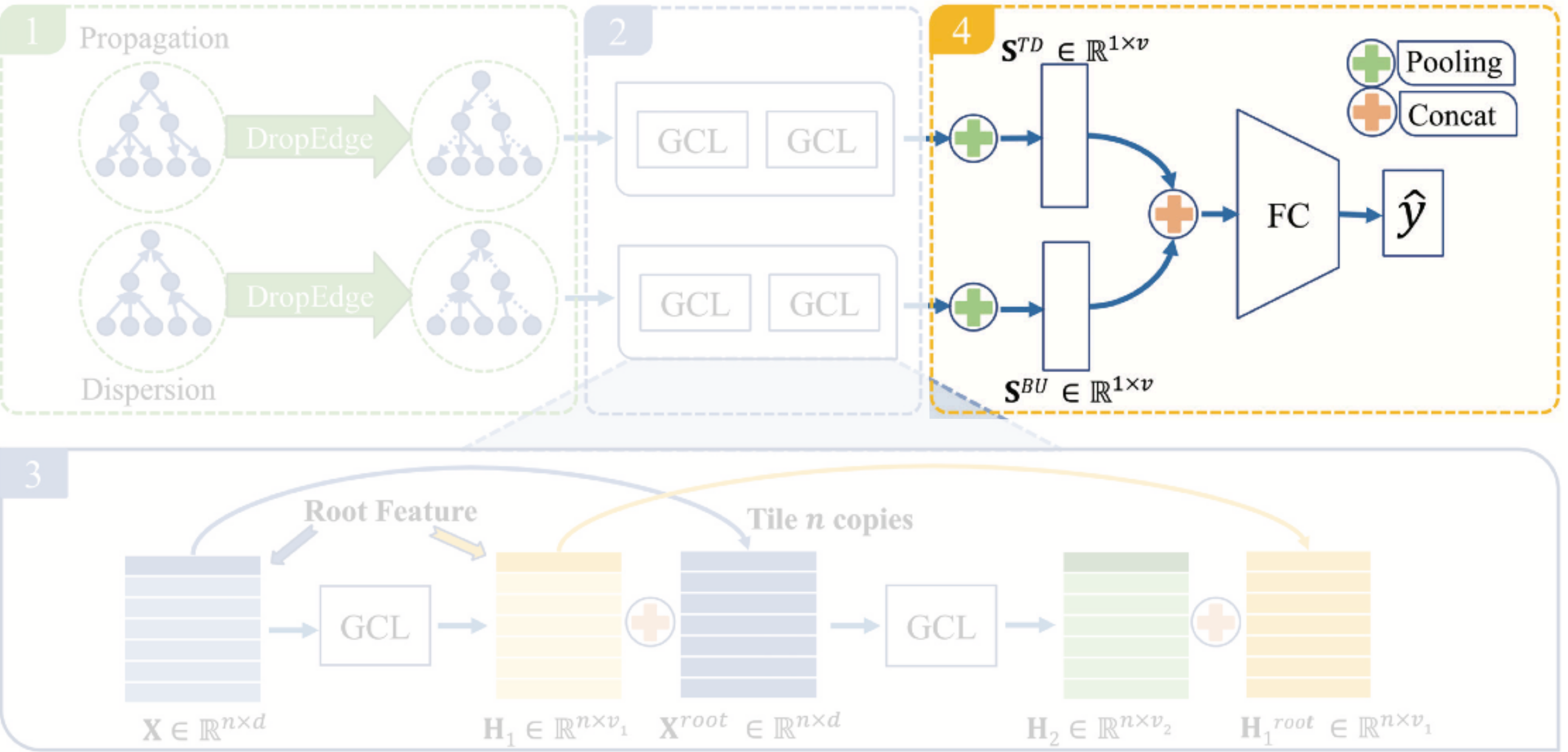
Methodology

Root Feature Enhancement



- $\mathbf{H}_1^{TD} = \sigma\left(\hat{\mathbf{A}}^{TD}\mathbf{X}\boldsymbol{W}_0^{TD}\right)$
- $\tilde{\mathbf{H}}_1^{TD} = \operatorname{concat}\left(\mathbf{H}_1^{TD}, \mathbf{X}^{root}\right)$
- $\mathbf{H}_2^{TD} = \sigma\left(\hat{\mathbf{A}}^{TD}\tilde{\mathbf{H}}_1^{TD}\boldsymbol{W}_1^{TD}\right)$
- $\tilde{\mathbf{H}}_2^{TD} = \operatorname{concat}\left(\mathbf{H}_2^{TD}, \left(\mathbf{H}_1^{TD}\right)^{root}\right)$
- $\tilde{\mathbf{H}}_1^{BU}, \tilde{\mathbf{H}}_2^{BU}$ are obtained in the same manner as above.

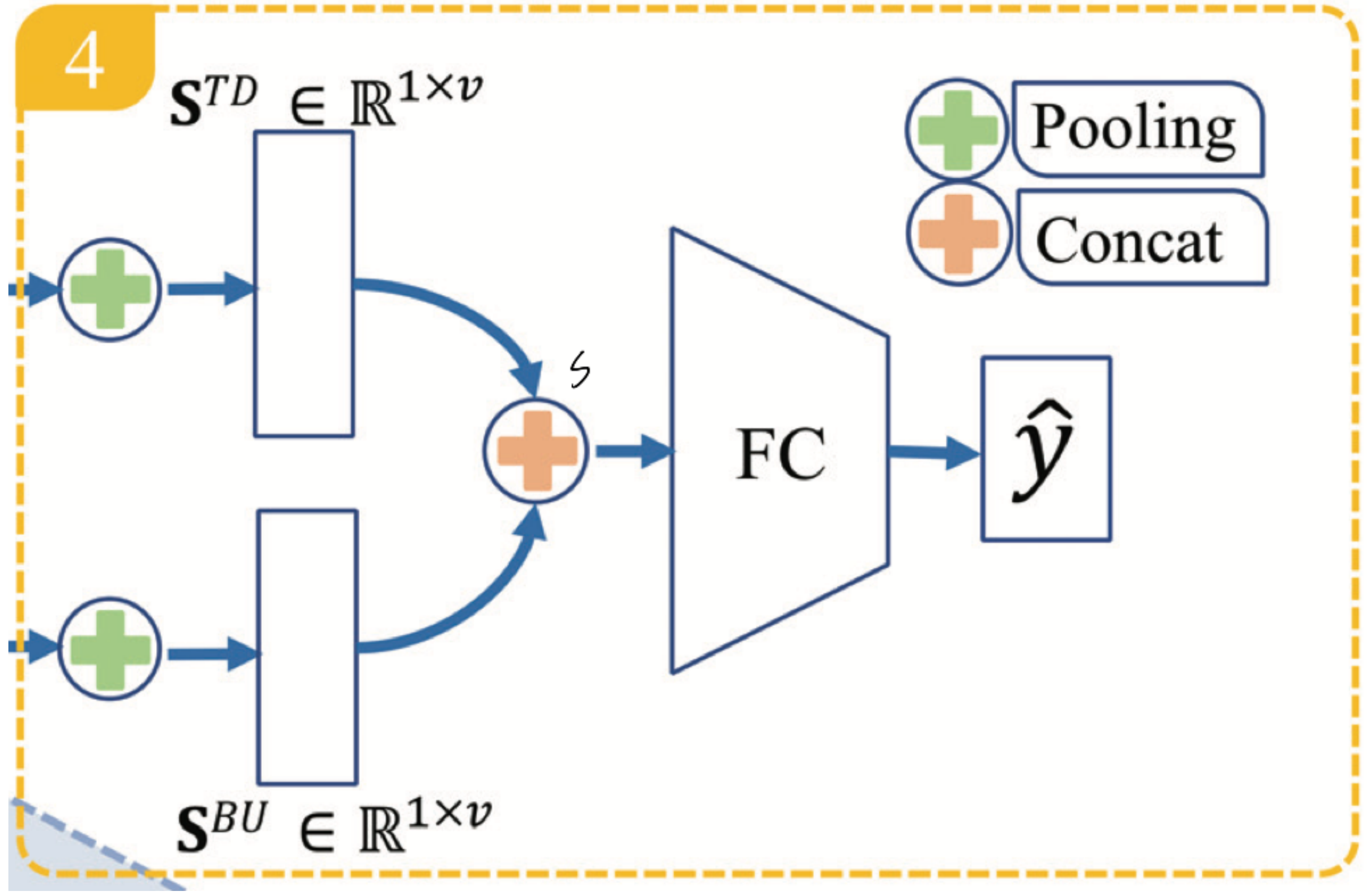




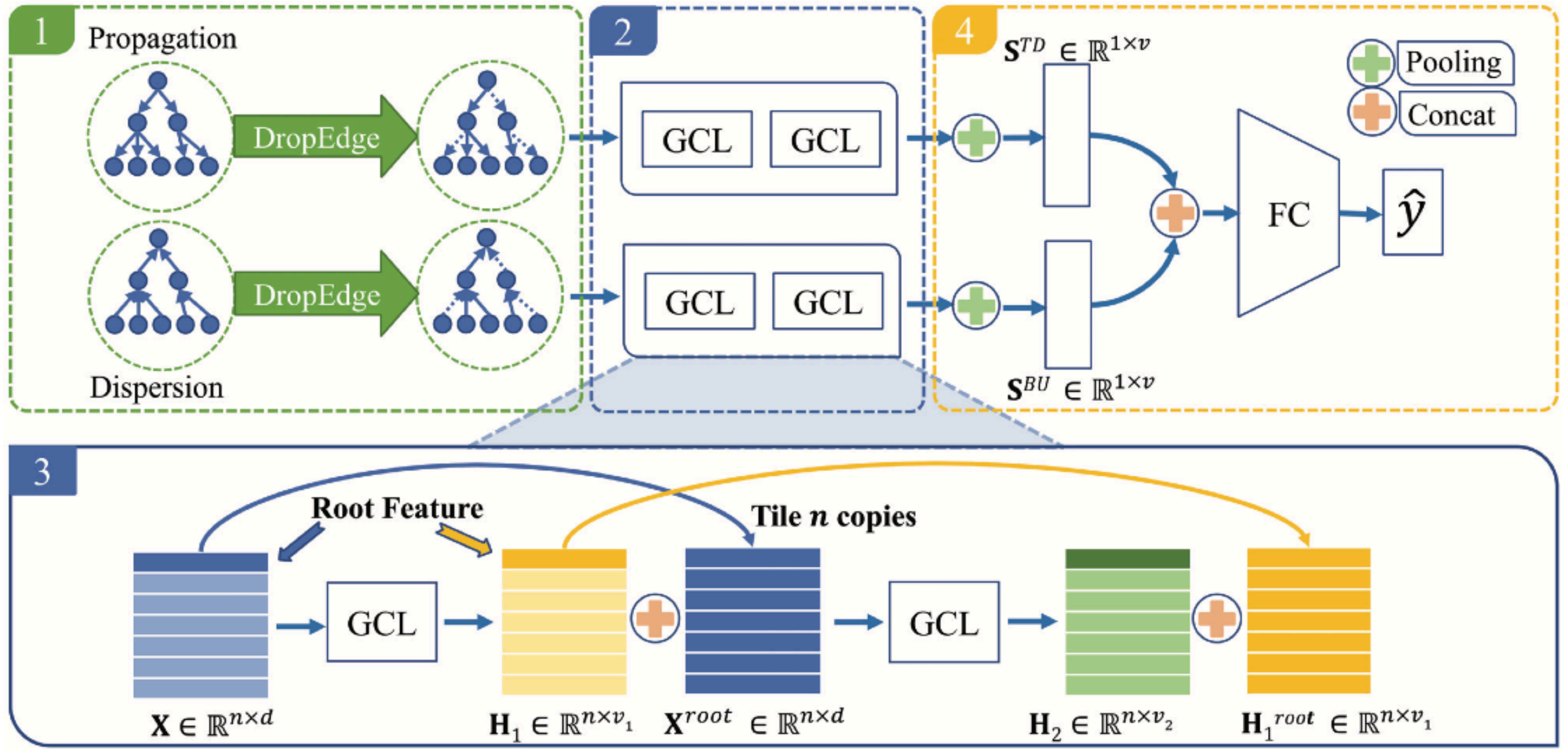
Methodology

Representations of Propagation and Dispersion for Rumor Classification

- Employ mean-pooling operators to aggregate information from these two sets of the node representations.
  - $\mathbf{S}^{TD} = \mathrm{MEAN}\left(\tilde{\mathbf{H}}_2^{TD}\right), \mathbf{S}^{BU} = \mathrm{MEAN}\left(\tilde{\mathbf{H}}_2^{BU}\right)$
- Then concatenate the representations of propagation and dispersion to merge the information as
  - $\mathbf{S} = \mathrm{concat}\left(\mathbf{S}^{TD}, \mathbf{S}^{BU}\right)$
- Finally the label of the event $\mathbf{y}$ is calculated via several fully connected layers and and softmax layer:
  - $\mathbf{y} = Softmax(FC(\mathbf{S}))$



Methodology
Optimizing



- Train all the parameters in the Bi-GCN model by minimizing the cross-entropy of the predictions and ground truth distributions, Y, over all events, C.
- L_2 regularizer is applied in the loss function over all model parameters.

Experiments
Datasets

- Nodes refer to users, edges represent retweet retweet or response relationship.
- Features are the extracted top-5000 words in terms of the TF-IDF values.
- Weibo labels: True (T), False (F)
- Twitter labels: Non-rumor (N), True (T), False (F), Unverified (U)

Table 1: Statistics of the datasets

Statistic	*Weibo*	*Twitter15*	*Twitter16*
# of posts	3,805,656	331,612	204,820
# of Users	2,746,818	276,663	173,487
# of events	4664	1490	818
# of True rumors	2351	374	205
# of False rumors	2313	370	205
# of Unverified rumors	0	374	203
# of Non-rumors	0	372	205
Avg. time length / event	2,460.7 Hours	1,337 Hours	848 Hours
Avg. # of posts / event	816	223	251
Max # of posts / event	59,318	1,768	2,765
Min # of posts / event	10	55	81

Experiments

Baselines

- **DTC (2011):** Decision Tree classifier based on various handcrafted features
- **SVM-RBF (2012):** SVM-based model with RBF kernel, using handcrafted features
- **SVM-TS (2015):** linear SVM classifier that leverages handcrafted features to construct time-series model
- **SVM-TK (2017):** SVM classifier with a propagation Tree Kernel on the basis of the propagation structures
- **RvNN (2018):** tree-structured recursive neural networks with GRU units that learn rumor representations via the propagation structure
- **PPC_RNN+CNN (2018):** combining RNN and CNN, which learns the rumor representations through the characteristics of users in the rumor propagation path

Experiments

Overall Performance

Weibo

Method	Class	Acc.	Prec.	Rec.	F_1
DTC	F T	0.831	0.847 0.815	0.815 0.824	0.831 0.819
SVM-RBF	F T	0.879	0.777 0.579	0.656 0.708	0.708 0.615
SVM-TS	F T	0.885	0.950 0.124	0.932 0.047	0.938 0.059
RvNN	F T	0.908	0.912 0.904	0.897 0.918	0.905 0.911
PPC_RNN+CNN	F T	0.916	0.884 0.955	0.957 0.876	0.919 0.913
Bi-GCN	F T	**0.961**	**0.961** **0.962**	**0.964** **0.962**	**0.961** **0.960**

Twitter15

Method	Acc.	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.454	0.415	0.355	0.733	0.317
SVM-RBF	0.318	0.225	0.082	0.455	0.218
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-TK	0.750	0.804	0.698	0.765	0.733
RvNN	0.723	0.682	0.758	0.821	0.654
PPC_RNN+CNN	0.477	0.359	0.507	0.300	0.640
Bi-GCN	**0.886**	**0.891**	**0.860**	**0.930**	**0.864**

Twitter16

Method	Acc.	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.473	0.254	0.080	0.190	0.482
SVM-RBF	0.553	0.670	0.085	0.117	0.361
SVM-TS	0.574	0.755	0.420	0.571	0.526
SVM-TK	0.732	0.740	0.709	0.836	0.686
RvNN	0.737	0.662	0.743	0.835	0.708
PPC_RNN+CNN	0.564	0.591	0.543	0.394	0.674
Bi-GCN	**0.880**	**0.847**	**0.869**	**0.937**	**0.865**

- Observe that the deep learning methods performs significantly better than those using hand-crafted features.
- Demonstrates the importance and necessity of studying deep learning for rumor detection.

Experiments

Overall Performance

Weibo

Method	Class	Acc.	Prec.	Rec.	F_1
DTC	F T	0.831	0.847 0.815	0.815 0.824	0.831 0.819
SVM-RBF	F T	0.879	0.777 0.579	0.656 0.708	0.708 0.615
SVM-TS	F T	0.885	0.950 0.124	0.932 0.047	0.938 0.059
RvNN	F T	0.908	0.912 0.904	0.897 0.918	0.905 0.911
PPC_RNN+CNN	F T	0.916	0.884 0.955	0.957 0.876	0.919 0.913
Bi-GCN	F T	**0.961**	**0.961** **0.962**	**0.964** **0.962**	**0.961** **0.960**

Twitter15

Method	Acc.	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.454	0.415	0.355	0.733	0.317
SVM-RBF	0.318	0.225	0.082	0.455	0.218
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-TK	0.750	0.804	0.698	0.765	0.733
RvNN	0.723	0.682	0.758	0.821	0.654
PPC_RNN+CNN	0.477	0.359	0.507	0.300	0.640
Bi-GCN	**0.886**	**0.891**	**0.860**	**0.930**	**0.864**

Twitter16

Method	Acc.	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.473	0.254	0.080	0.190	0.482
SVM-RBF	0.553	0.670	0.085	0.117	0.361
SVM-TS	0.574	0.755	0.420	0.571	0.526
SVM-TK	0.732	0.740	0.709	0.836	0.686
RvNN	0.737	0.662	0.743	0.835	0.708
PPC_RNN+CNN	0.564	0.591	0.543	0.394	0.674
Bi-GCN	**0.880**	**0.847**	**0.869**	**0.937**	**0.865**

- Bi-GCN outperforms PPC_RNN+CNN in terms of all the performance measures, indicates the effectiveness of incorporating the dispersion structure for rumor detection.
- Since RNN & CNN can't process data with the graph structure, so ignore important structure features of dispersion.

Experiments

Overall Performance

Weibo

Method	Class	Acc.	Prec.	Rec.	F_1
DTC	F T	0.831	0.847 0.815	0.815 0.824	0.831 0.819
SVM-RBF	F T	0.879	0.777 0.579	0.656 0.708	0.708 0.615
SVM-TS	F T	0.885	0.950 0.124	0.932 0.047	0.938 0.059
RvNN	F T	0.908	0.912 0.904	0.897 0.918	0.905 0.911
PPC_RNN+CNN	F T	0.916	0.884 0.955	0.957 0.876	0.919 0.913
Bi-GCN	F T	**0.961**	**0.961** **0.962**	**0.964** **0.962**	**0.961** **0.960**

Twitter15

Method	Acc.	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.454	0.415	0.355	0.733	0.317
SVM-RBF	0.318	0.225	0.082	0.455	0.218
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-TK	0.750	0.804	0.698	0.765	0.733
RvNN	0.723	0.682	0.758	0.821	0.654
PPC_RNN+CNN	0.477	0.359	0.507	0.300	0.640
Bi-GCN	**0.886**	**0.891**	**0.860**	**0.930**	**0.864**

Twitter16

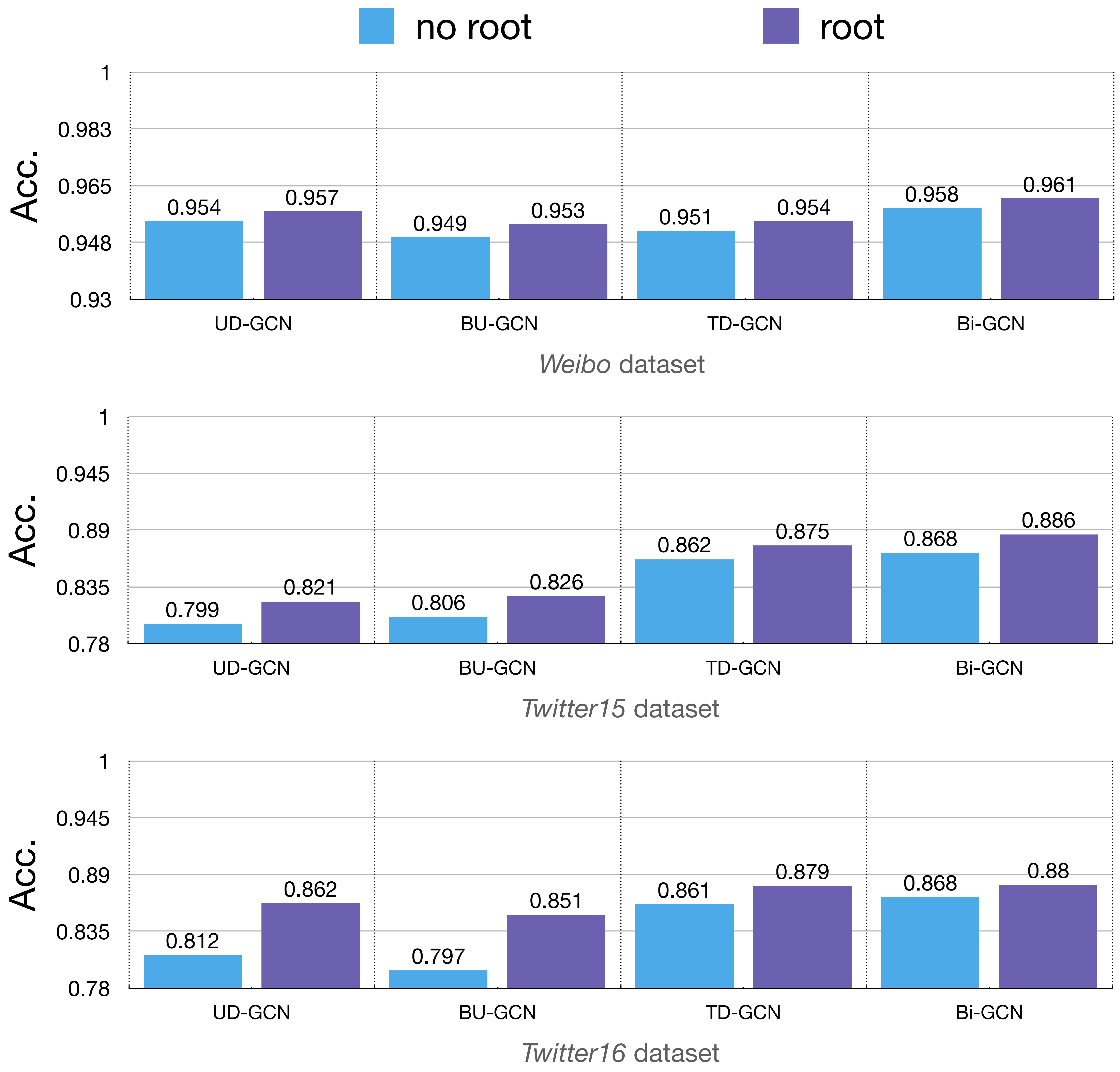
Method	Acc.	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.473	0.254	0.080	0.190	0.482
SVM-RBF	0.553	0.670	0.085	0.117	0.361
SVM-TS	0.574	0.755	0.420	0.571	0.526
SVM-TK	0.732	0.740	0.709	0.836	0.686
RvNN	0.737	0.662	0.743	0.835	0.708
PPC_RNN+CNN	0.564	0.591	0.543	0.394	0.674
Bi-GCN	**0.880**	**0.847**	**0.869**	**0.937**	**0.865**

- Bi-GCN is significantly superior to the RvNN method, RvNN only uses the hidden feature vector of all the leaf nodes so that it's heavily impacted by information of the latest post (lack of information such as comments, and just follow the former posts).
- Root feature enhancement of Bi-GCN to pay attention to the information of the source posts.

Experiments

Ablation Study

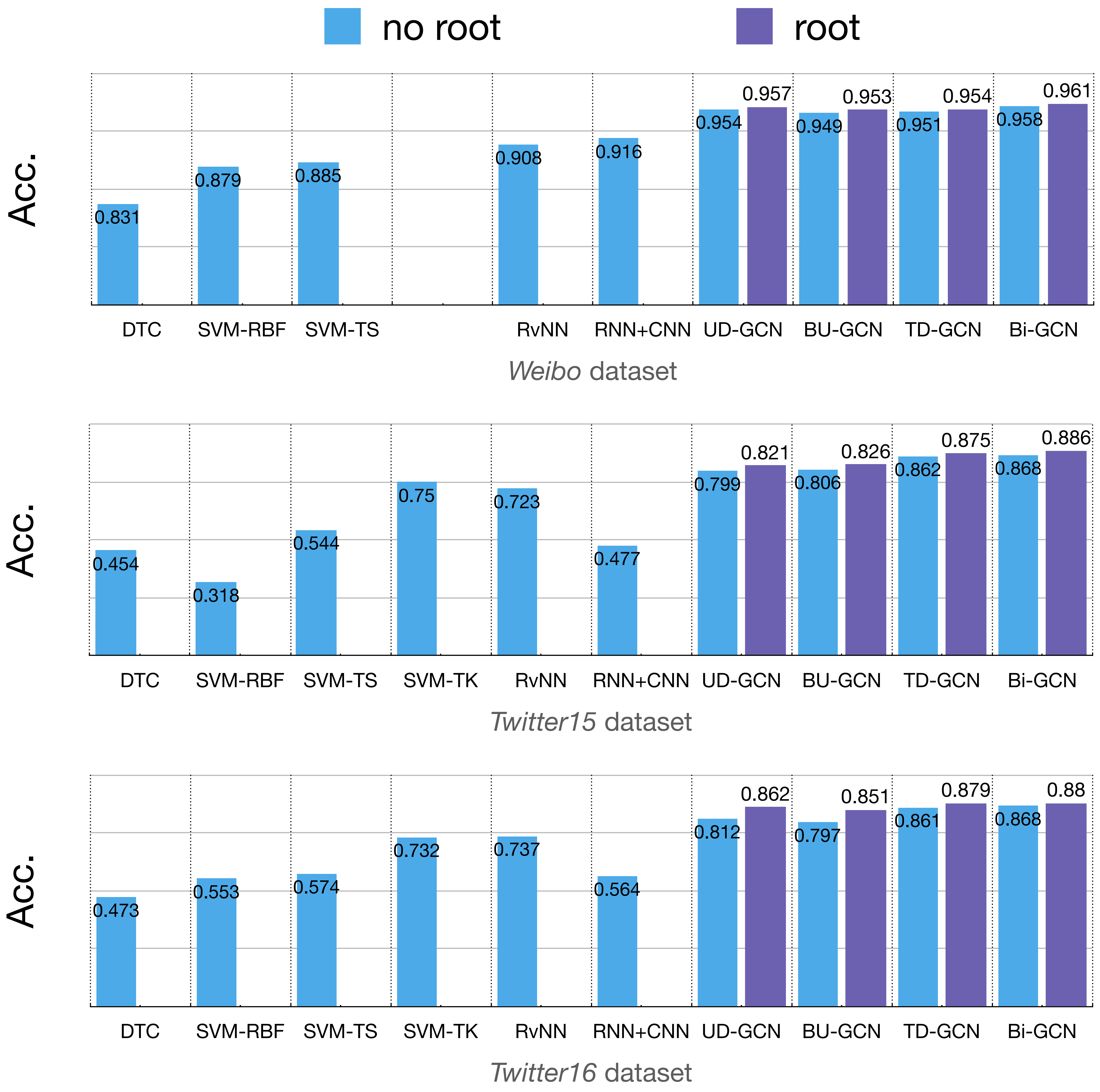
- All variants outperform without root feature enhancement.
- Indicates that the source posts plays an important role in rumor detection.
- TD-GCN and BU-GCN can't always achieve better results that UD-GCN, but Bi-GCN is always superior to them.
- Implies the importance to simultaneously consider both top-down and bottom-up representations.



Experiments

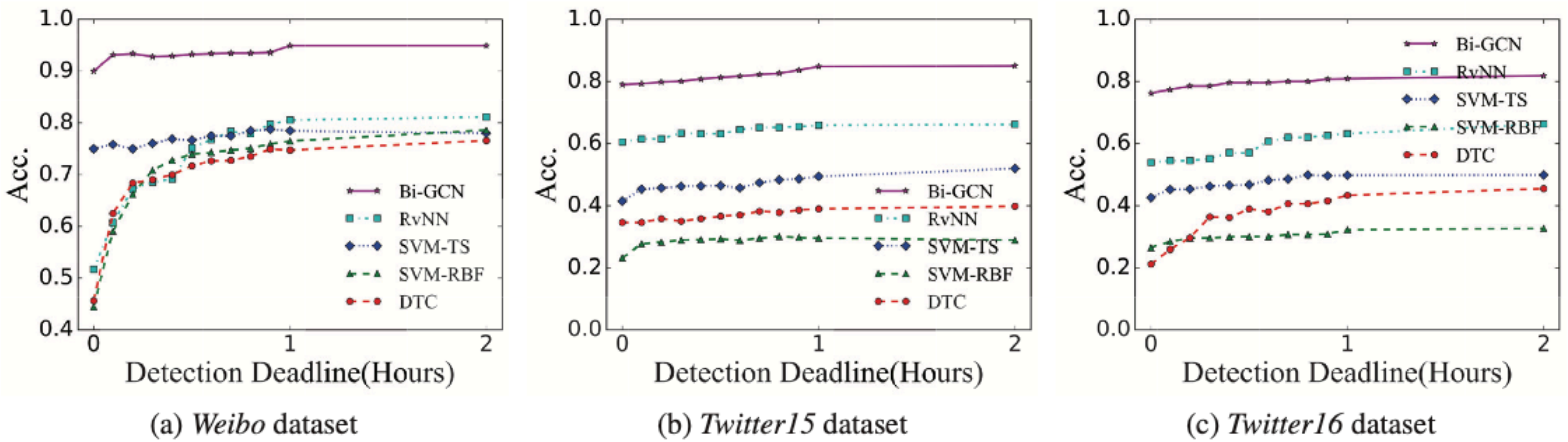
Ablation Study

- Even the worst result in variants are better than those of other baseline methods by a large gap.
- Again verifies the effectiveness of graph convolution for rumor detection.



Experiments

Early Rumor Detection



Result of rumor early detection on three datasets

- Bi-GCN reaches relatively high accuracy at a very early period after the source post initial broadcast.
- Observe that structural features are not only beneficial to long-term rumor detection, but also helpful to the early detection.

Conclusions

- Proposed a GCN-based model of rumor detection on social media, called Bi-GCN.
- Bi-GCN achieves the best performance by considering both
  - Causal features of rumor propagation along relationship chains from top to down propagation pattern
  - Structural features of rumor dispersion within communities through the bottom-up gathering.
- Improve the effectiveness of the model by concatenating the features of the source posts after each GCL of GCN.

Comments
of Bi-GCN

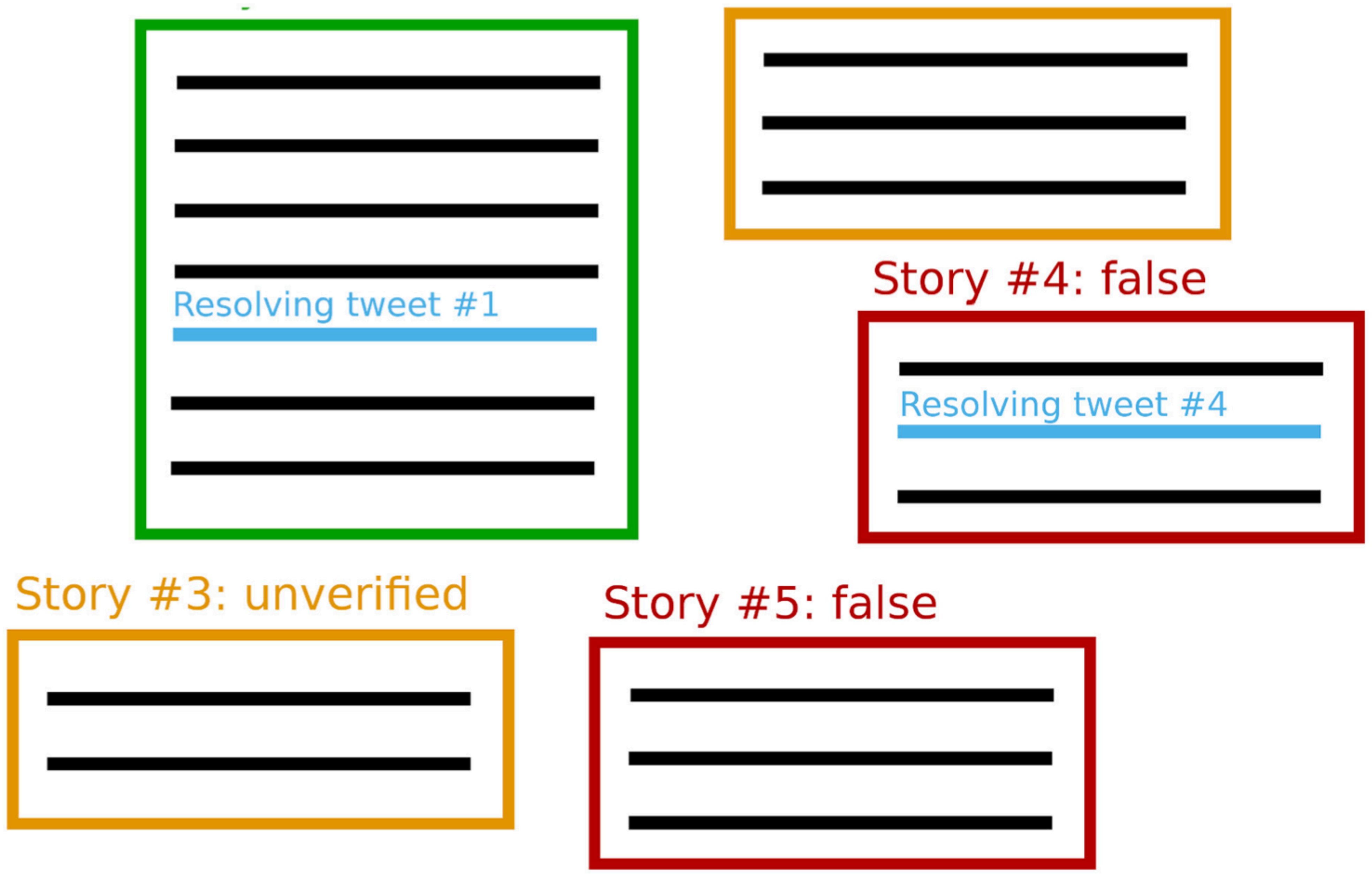
- Consider the dispersion of rumor as feature for learning representation.
- Effective on root feature enhancement.
- RNN+CNN baseline Twitter dataset is awful.
- About event label on Twitter dataset, the unverified rumor and non-rumor may confused during the training.
- Using top-5000 words to get TF-IDF value as representation not informative.
- Competition baseline little outdated and not seen other GCN-based model.

Research

of Event label on Twitter datasets

Moreover, most existing approaches regard rumor detection as a binary classification problem, which predicts a candidate hypothesis as rumor or not. Since a rumor often begins as unverified and later turns out to be confirmed as true or false, or remains unverified (Zubiaga et al., 2016), here we consider a set of more practical, finer-grained classes: false rumor, true rumor, unverified rumor, and non-rumor, which becomes an even more challenging problem.

Ma, Gao, and Wong 2017



Our datasets consist of rumour stories, represented by squares, which can be one of true (green), false (red), or unverified (orange). Each of the rumour stories has a number of rumour threads associated with it, which we represent as black lines that form a timeline where threads are sorted by time. When a story is true or false, the journalists also picked, within the story's timeline, one tweet as the resolving tweet. Note that the resolving tweets cannot always be found within the Twitter timeline, as in example story #5.

Zubiaga et al., 2016

Research

of Event label on Twitter datasets

Finally, we annotated the source tweets by referring to the labels of the events they are from. We first turned the label of each event in Twitter15 and Twitter16 from binary to quaternary according to the veracity tag of the article in rumor debunking websites (e.g., snopes.com, Emergent.info, etc). Then we labeled the source tweets by following these rules: 1) Source tweets from unverified rumor events or non-rumor events are labeled the same as the corresponding event's label; 2) For a source tweet in false rumor event, we flip over the label and assign true to the source tweet if it expresses denial type of stance; otherwise, the label is assigned as false; 3) The analogous flip-over/no-change rule applies to the source tweets from true rumor events.

Ma, Gao, and Wong 2017

True
This rating indicates that the primary elements of a claim are demonstrably true.

Mostly True
This rating indicates that the primary elements of a claim are demonstrably true, but some of the ancillary details surrounding the claim ma

Mixture
This rating indicates that a claim has significant elements of both truth and falsity to it such that it could not fairly be described by any othe

Mostly False
This rating indicates that the primary elements of a claim are demonstrably false, but some of the ancillary details surrounding the claim m

False
This rating indicates that the primary elements of a claim are demonstrably false.

Unproven
This rating indicates that insufficient evidence exists to establish the given claim as true, but the claim cannot be definitively proved false. claims for which there is little or no affirmative evidence, but for which declaring them to be false would require the difficult (if not impossi prove a negative or accurately discern someone else's thoughts and motivations.

https://www.snopes.com/fact-check-ratings/

Unverified

APPLE
Claim: Samsung will supply application processors for Apple Watch
Originating Source: **businesskorea.co.kr** Added Nov 26

True

VIRAL
Claim: A man in England is wanted by police for slapping people who sneeze in public
Originating Source: **newsandstar.co.uk** Added Mar 23

False

VIRAL
Claim: Doctors confirmed the first case of death by genetically modified food
Originating Source: **worldnewsdailyreport.com** Added Mar 9

http://www.emergent.info/