Attention Guided Graph Convolutional Networks for Relation Extraction

Zhijiang Guo

Joint work with Yan Zhang, Wei Lu

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Relation Extraction

Sentence-level Relation Extraction

Cross-sentence *n*-ary Relation Extraction

Relation Extraction

Sentence-level Relation Extraction

Input

Carey will succeed Jack, who held the position for 15 years and will take on a new role as chairman.

Relation

per:title

Relation Extraction

Cross-sentence *n*-ary Relation Extraction

Input

The deletion mutation on exon-19 of EGFR gene was present in 16 patients, while the L858E point mutation on exon-21 was noted. All patients were treated with getfitnib and showed a *partial response*.

Relation

sensitivity

Neural Approaches

Sequence-based Model

Dependency-based Model

Neural Approaches

Sequence-based Model

Operates only on the given text sequences

CNNs

Zeng et al., 2014, Wang et al., 2016

RNNs

Zhou et al., 2016, Zhang et al., 2017

CNNs + RNNs

Vu et al., 2016

Neural Approaches

Dependency-based Model

Incorporates the dependency tree into the model

Graph-LSTM

Peng et al., 2017

GCNs

Zhang et al., 2018

GRNs

Song et al., 2018

Dependency-Based Model

Pruning and Encoder

Remove irrelevant information from the tree while keeping relevant content

SDP + RNNs/CNNs

Xu et al., 2015ab

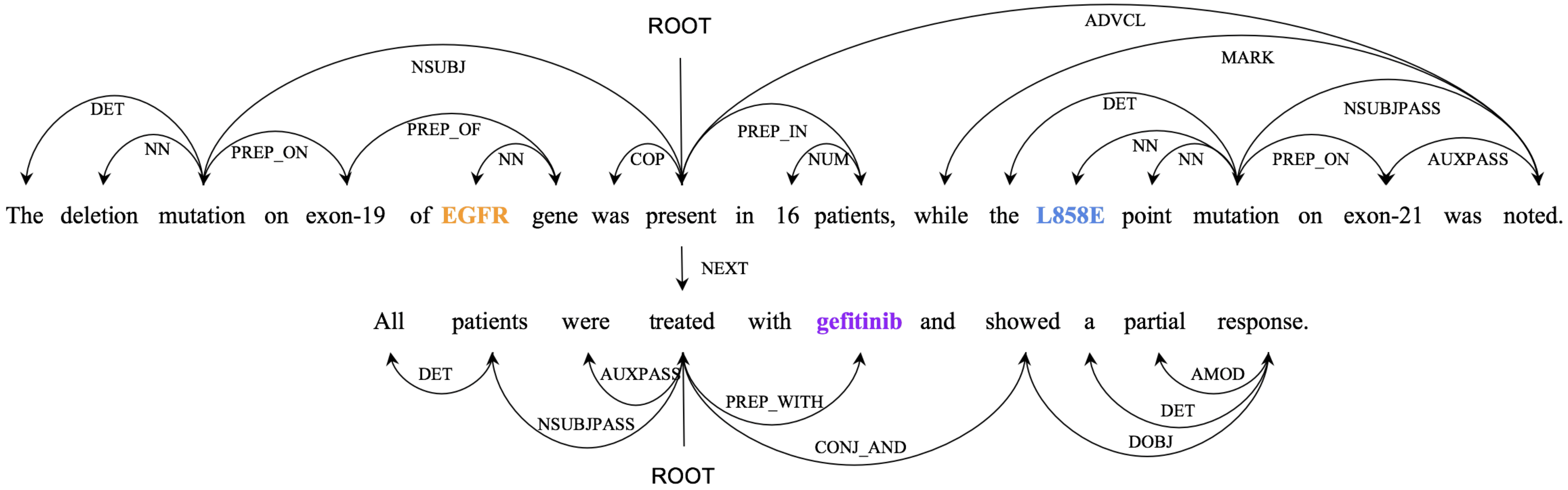
LCA Subtree + Tree-LSTM

Miwa et al., 2016

Pruned Tree + GCNs

Zhang et al., 2018

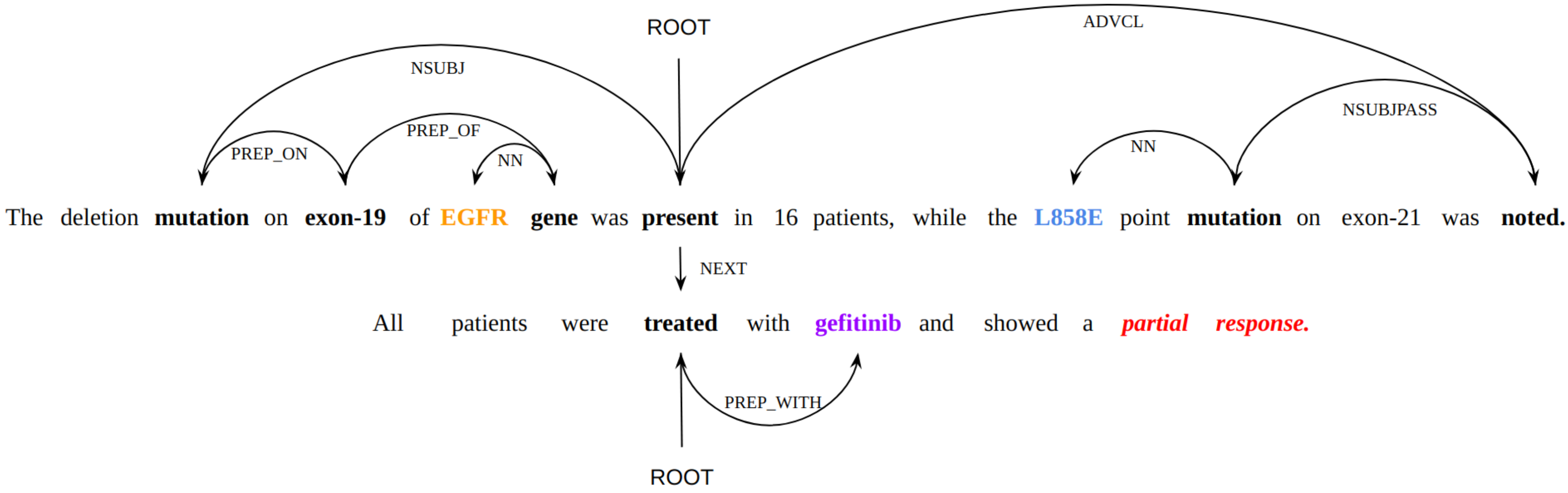
Example Graph



Dependency-Based Model

SDP + RNNs/CNNs
(Xu et al., 2015ab)

Shortest dependency path between entities

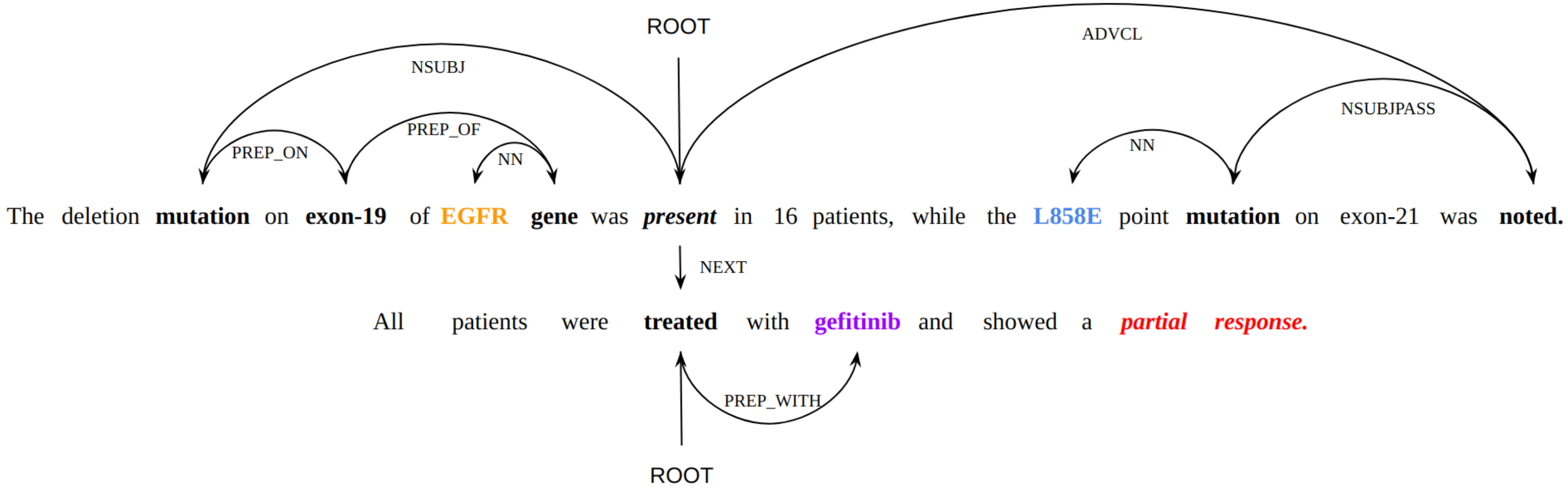


Dependency-Based Model

LCA Subtree + Tree-LSTM

(Miwa et al., 2016)

Subtree below lowest common ancestor (LCA) of entities

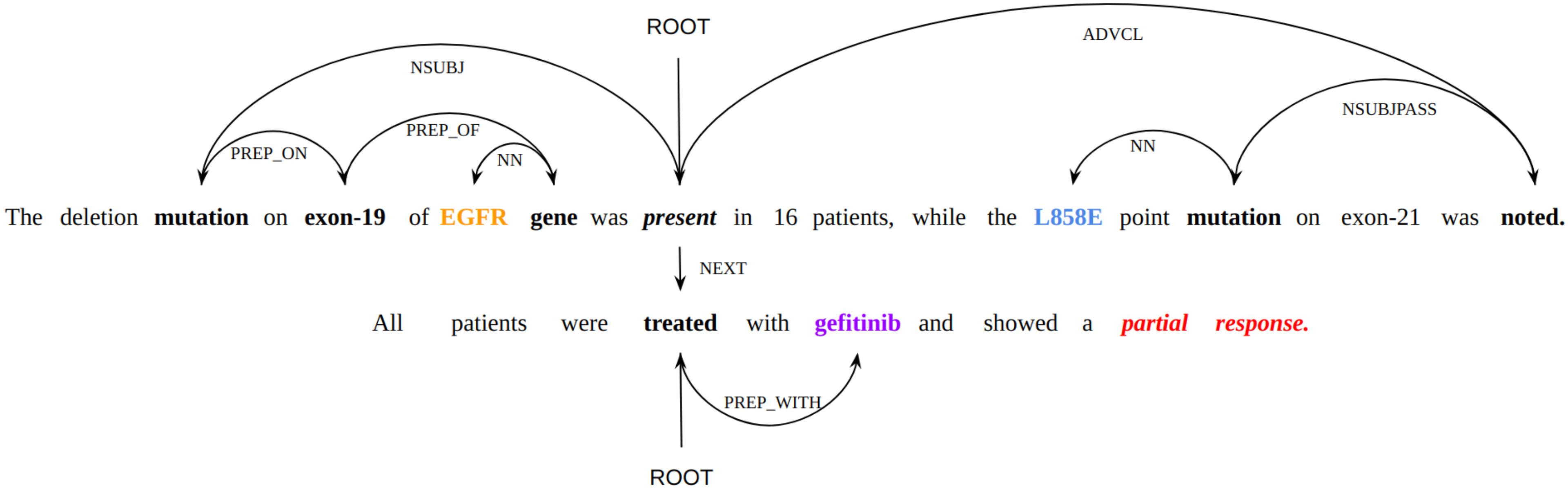


Dependency-Based Model

Pruned Tree + GCNs

(Zhang et al., 2018)

Includes tokens distance *K* away from the LCA subtree

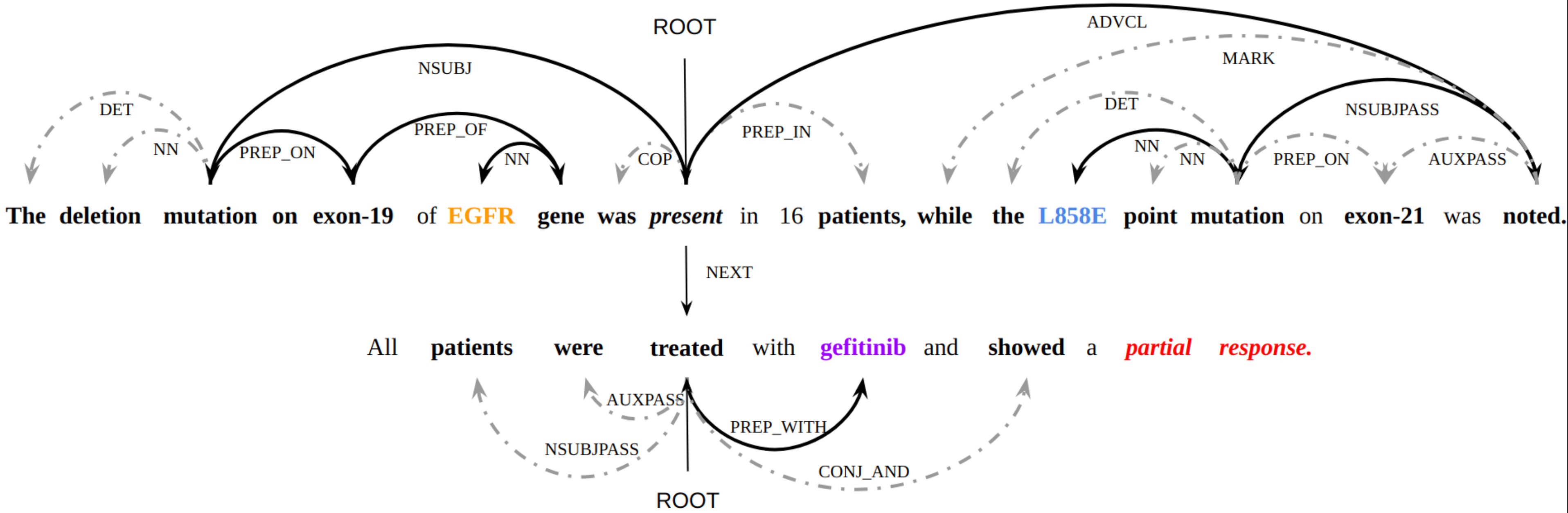


$K = 0$ (LCA subtree)

Dependency-Based Model

Pruned Tree + GCNs

The pruned tree grows when *K* increases

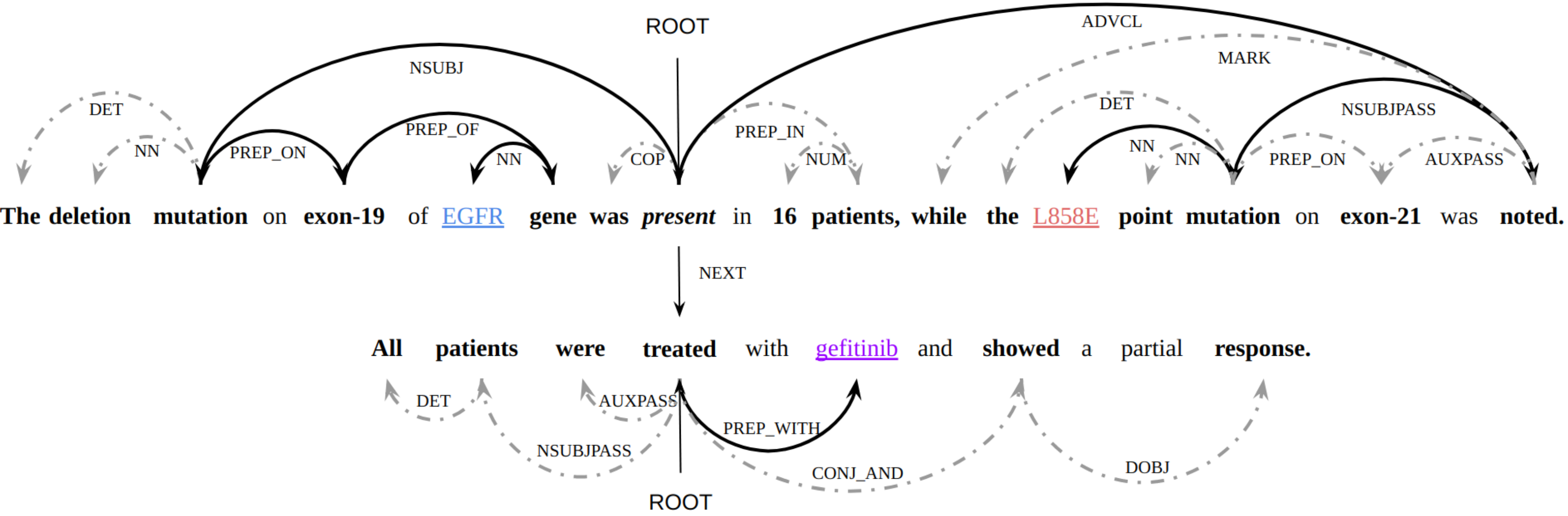


$K = 1$

Dependency-Based Model

Pruned Tree + GCNs

A proper *K* value is required to maintain a balance between keeping and removing information



$K = 2$

Dependency-Based Model

Pruning	**Encoder**	**Pros**	**Cons**
SDP	RNNs/ CNNs	Computationally Efficient	Not a Structural Encoder **May exclude information**
LCA Subtree	Tree-LSTM	Structural Encoder	Hard to Parallelize **May exclude information**
Pruned Tree	GCNs	Computationally Efficient	**Hard to find an optimal *K***

Dependency-Based Model

Pruning	**Encoder**	**Pros**	**Cons**
SDP	RNNs/ CNNs	Computationally Efficient	Not a Structural Encoder **May exclude information**
LCA Subtree	Tree-LSTM	Structural Encoder	Hard to Parallelize **May exclude information**
Pruned Tree	GCNs	Computationally Efficient	**Hard to find an optimal *K***

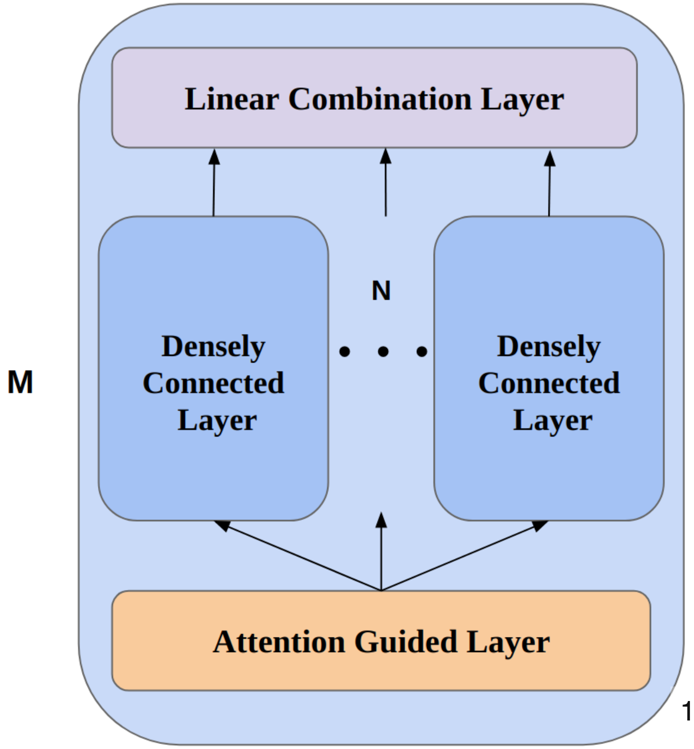
Motivation: Is it possible to *learn* a pruning strategy *without* additional computational overhead?

Model

Attention Guided GCNs (AGGCNs)

Consists of ***M*** identical blocks, each has 3 types of layers

- Attention Guided Layer
- Densely Connected Layer
- Linear Combination Layer

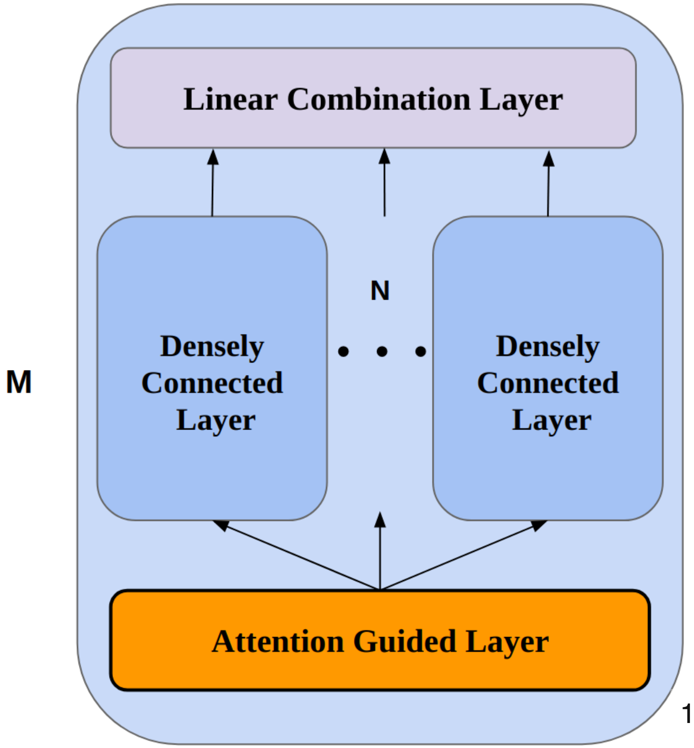


Model

Attention Guided GCNs (AGGCNs)

Consists of ***M*** identical blocks, each has 3 types of layers

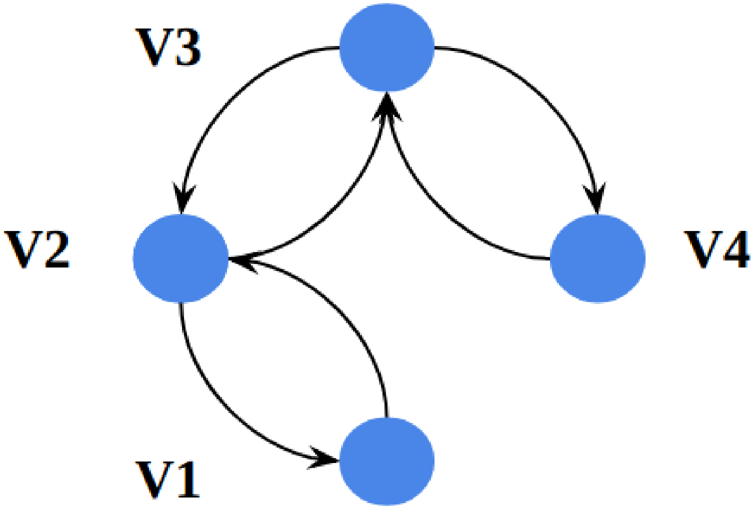
- **Attention Guided Layer**
- Densely Connected Layer
- Linear Combination Layer



Model

GCNs Input

An adjacency matrix that represents the input graph



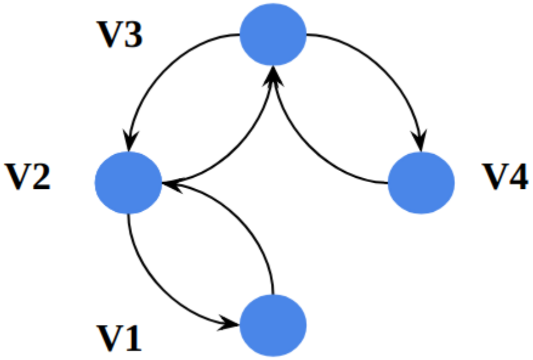
$$
\begin{array}{cc}
 & \begin{array}{cccc} V1 & V2 & V3 & V4 \end{array} \\
\begin{array}{c} V1 \\ V2 \\ V3 \\ V4 \end{array} & \begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix}
\end{array}
$$

Model

Attention Guided Layer

Rule-based pruning can be viewed as **hard attention**

$$
\begin{array}{c|cccc|}
 & V1 & V2 & V3 & V4 \\
\hline
V1 & 1 & 1 & 0 & 0 \\
V2 & 1 & 1 & 1 & 0 \\
V3 & 0 & 1 & 1 & 1 \\
V4 & 0 & 0 & 1 & 1
\end{array}
$$

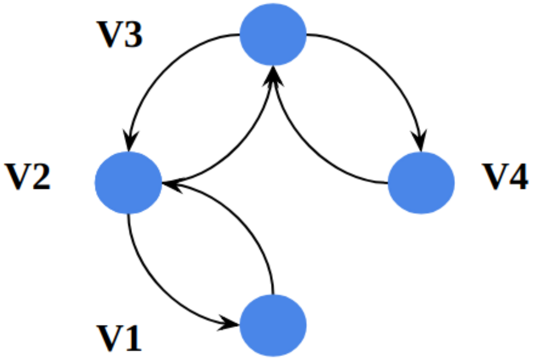


Model

Attention Guided Layer

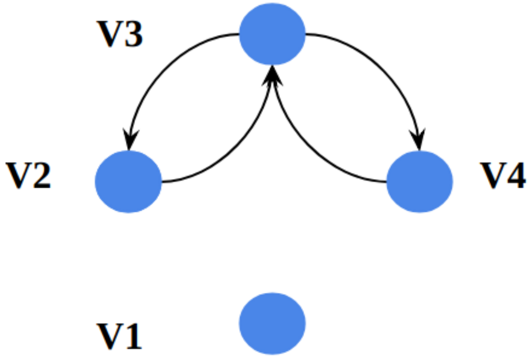
Rule-based pruning can be viewed as **hard attention**

$$\begin{array}{c|cccc|} & V1 & V2 & V3 & V4 \\ \hline V1 & 1 & 1 & 0 & 0 \\ V2 & 1 & 1 & 1 & 0 \\ V3 & 0 & 1 & 1 & 1 \\ V4 & 0 & 0 & 1 & 1 \end{array}$$



Remove Node V1

$$\begin{array}{c|cccc|} & V1 & V2 & V3 & V4 \\ \hline V1 & 1 & \color{red}{0} & 0 & 0 \\ V2 & \color{red}{0} & 1 & 1 & 0 \\ V3 & 0 & 1 & 1 & 1 \\ V4 & 0 & 0 & 1 & 1 \end{array}$$

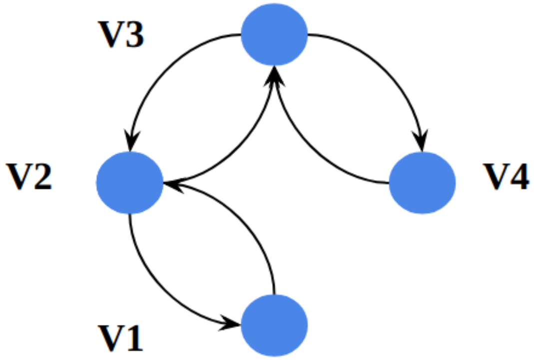


Model

Attention Guided Layer

Soft pruning: assign different weights to different edges

$$
\begin{array}{c|cccc|}
 & \mathbf{V1} & \mathbf{V2} & \mathbf{V3} & \mathbf{V4} \\
\mathbf{V1} & 1 & 1 & 0 & 0 \\
\mathbf{V2} & 1 & 1 & 1 & 0 \\
\mathbf{V3} & 0 & 1 & 1 & 1 \\
\mathbf{V4} & 0 & 0 & 1 & 1
\end{array}
$$

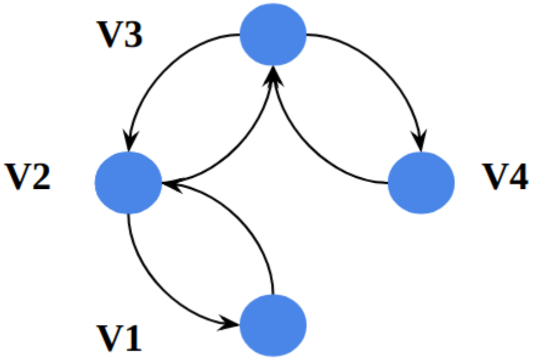


Model

Attention Guided Layer

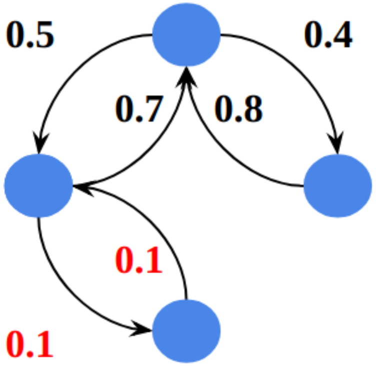
Soft pruning: assign different weights to different edges

$$
\begin{array}{c} \\ V1 \\ V2 \\ V3 \\ V4 \end{array}
\begin{array}{c}
\begin{array}{cccc} V1 & V2 & V3 & V4 \end{array} \\
\left|\begin{array}{cccc}
1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1
\end{array}\right|
\end{array}
$$



Assign Weights

$$
\begin{array}{c} \\ V1 \\ V2 \\ V3 \\ V4 \end{array}
\begin{array}{c}
\begin{array}{cccc} V1 & V2 & V3 & V4 \end{array} \\
\left|\begin{array}{cccc}
0.9 & \color{red}{0.1} & 0.0 & 0.0 \\
\color{red}{0.1} & 0.2 & 0.7 & 0.0 \\
0.0 & 0.5 & 0.1 & 0.4 \\
0.0 & 0.0 & 0.8 & 0.2
\end{array}\right|
\end{array}
$$

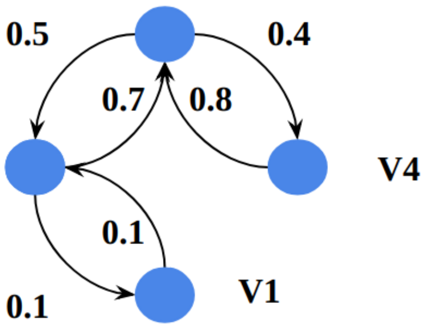


Model

Attention Guided Layer

Fully connected weighted graphs can capture **multi-hop** relations between nodes in a large graph

	V1	V2	V3	V4
V1	0.9	0.1	0.0	0.0
V2	0.1	0.2	0.7	0.0
V3	0.0	0.5	0.1	0.4
V4	0.0	0.0	0.8	0.2

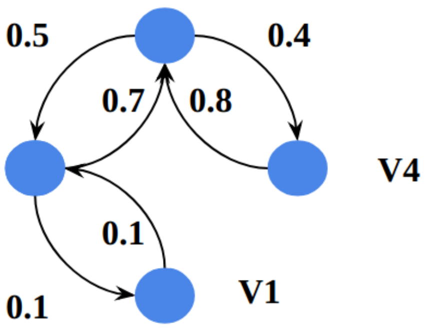


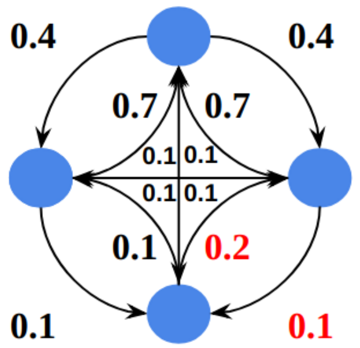
Model

Attention Guided Layer

Fully connected weighted graphs can capture **multi-hop** relations between nodes in a large graph

$$
\begin{array}{c|cccc|}
 & V1 & V2 & V3 & V4 \\
V1 & 0.9 & 0.1 & 0.0 & 0.0 \\
V2 & 0.1 & 0.2 & 0.7 & 0.0 \\
V3 & 0.0 & 0.5 & 0.1 & 0.4 \\
V4 & 0.0 & 0.0 & 0.8 & 0.2
\end{array}
\xrightarrow{\text{Fully Connected}}
\begin{array}{c|cccc|}
 & V1 & V2 & V3 & V4 \\
V1 & 0.6 & 0.1 & 0.1 & 0.2 \\
V2 & 0.1 & 0.1 & 0.7 & 0.1 \\
V3 & 0.1 & 0.4 & 0.1 & 0.4 \\
V4 & 0.1 & 0.0 & 0.8 & 0.2
\end{array}
$$



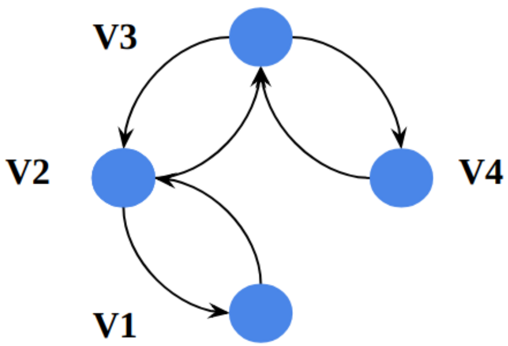


Model

Attention Guided Layer

Use **multi-head** (***N*** head) **attention** (Vaswani et al., 2017) to construct ***N*** fully connected weighted graphs

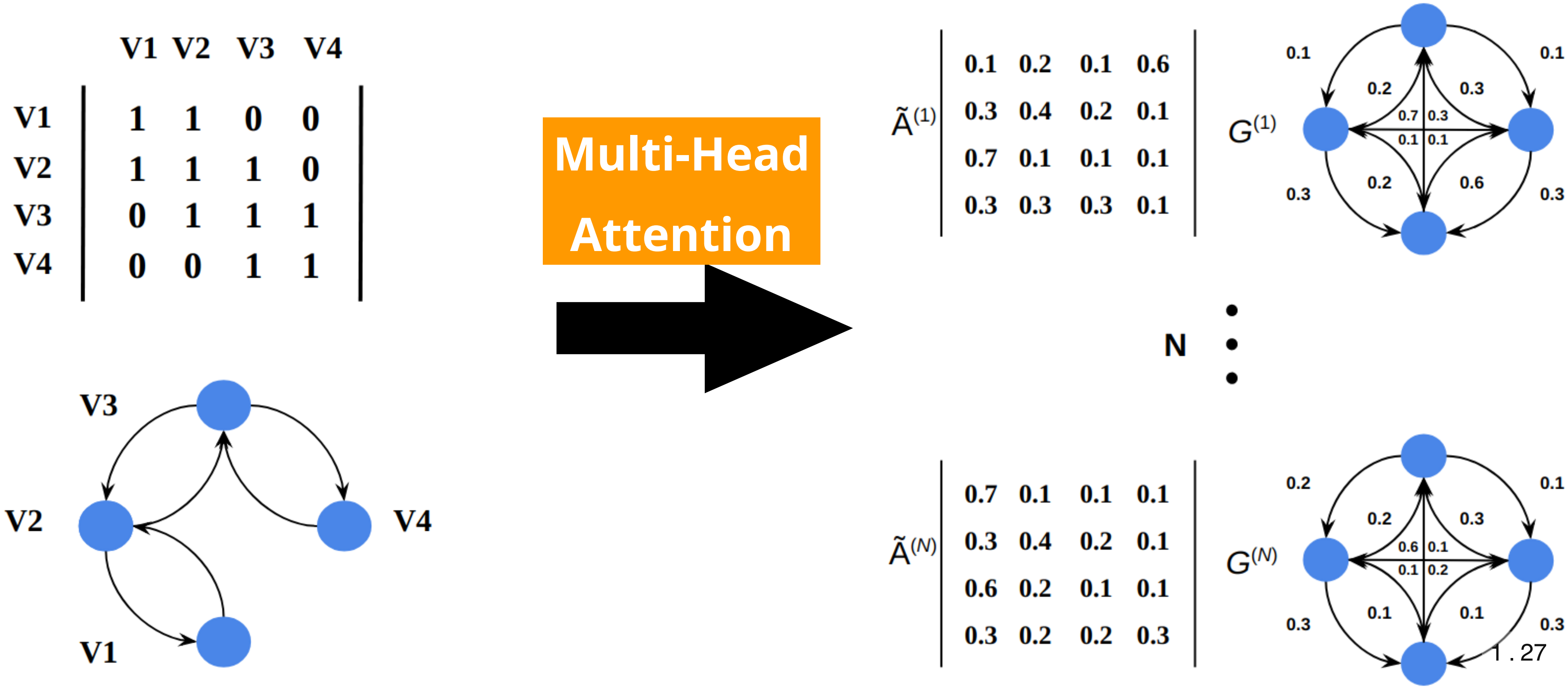
$$
\begin{array}{c|cccc|}
 & V1 & V2 & V3 & V4 \\
V1 & 1 & 1 & 0 & 0 \\
V2 & 1 & 1 & 1 & 0 \\
V3 & 0 & 1 & 1 & 1 \\
V4 & 0 & 0 & 1 & 1
\end{array}
$$



Model

Attention Guided Layer

Use **multi-head** (***N*** head) **attention** (Vaswani et al., 2017) to construct ***N*** fully connected weighted graphs

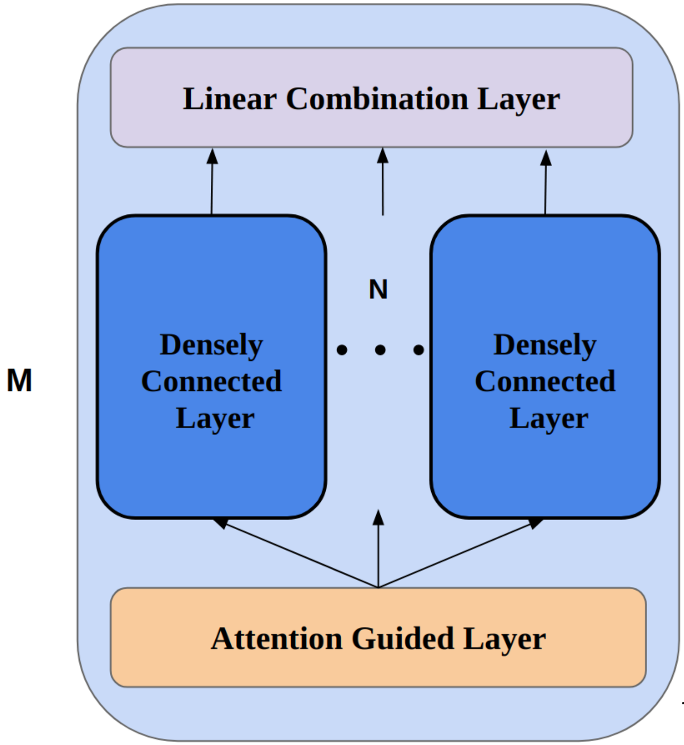


Model

Attention Guided GCNs (AGGCNs)

Consists of ***M*** identical blocks, each has 3 types of layers

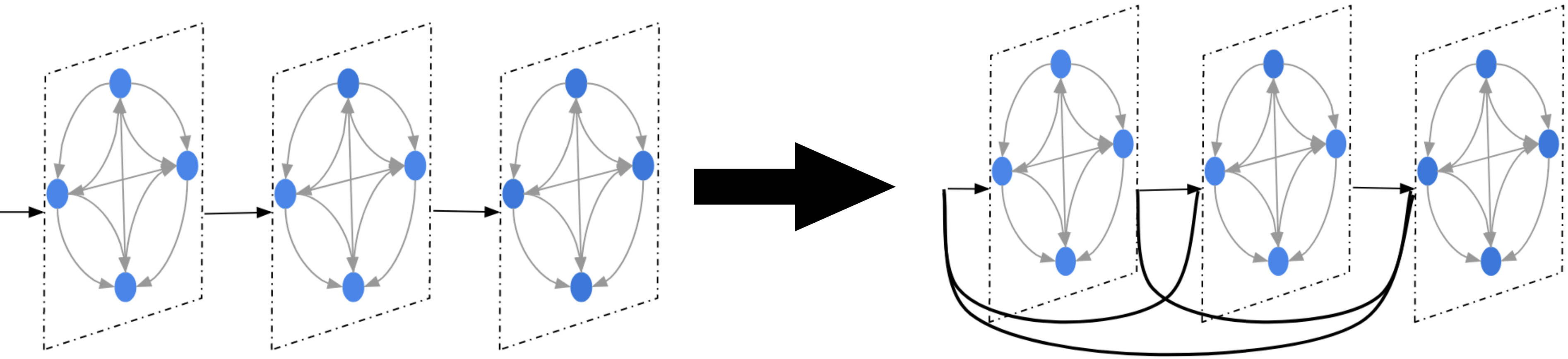
- Attention Guided Layer
- **Densely Connected Layer**
- Linear Combination Layer



Model

Densely Connected Layer

Use densely connected **graph convolutional layers** (Guo et al., 2019) to better encode large graph

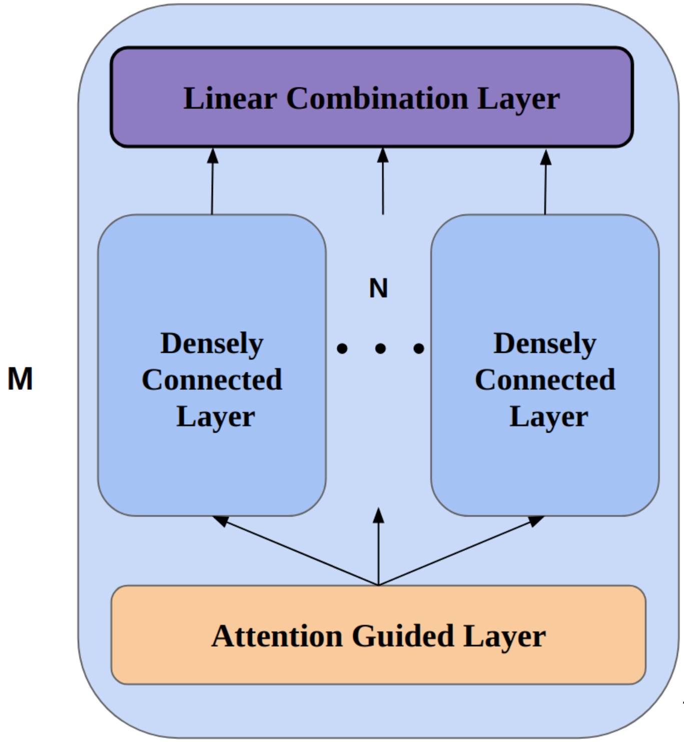


Model

Attention Guided GCNs (AGGCNs)

Consists of ***M*** identical blocks, each has 3 types of layers

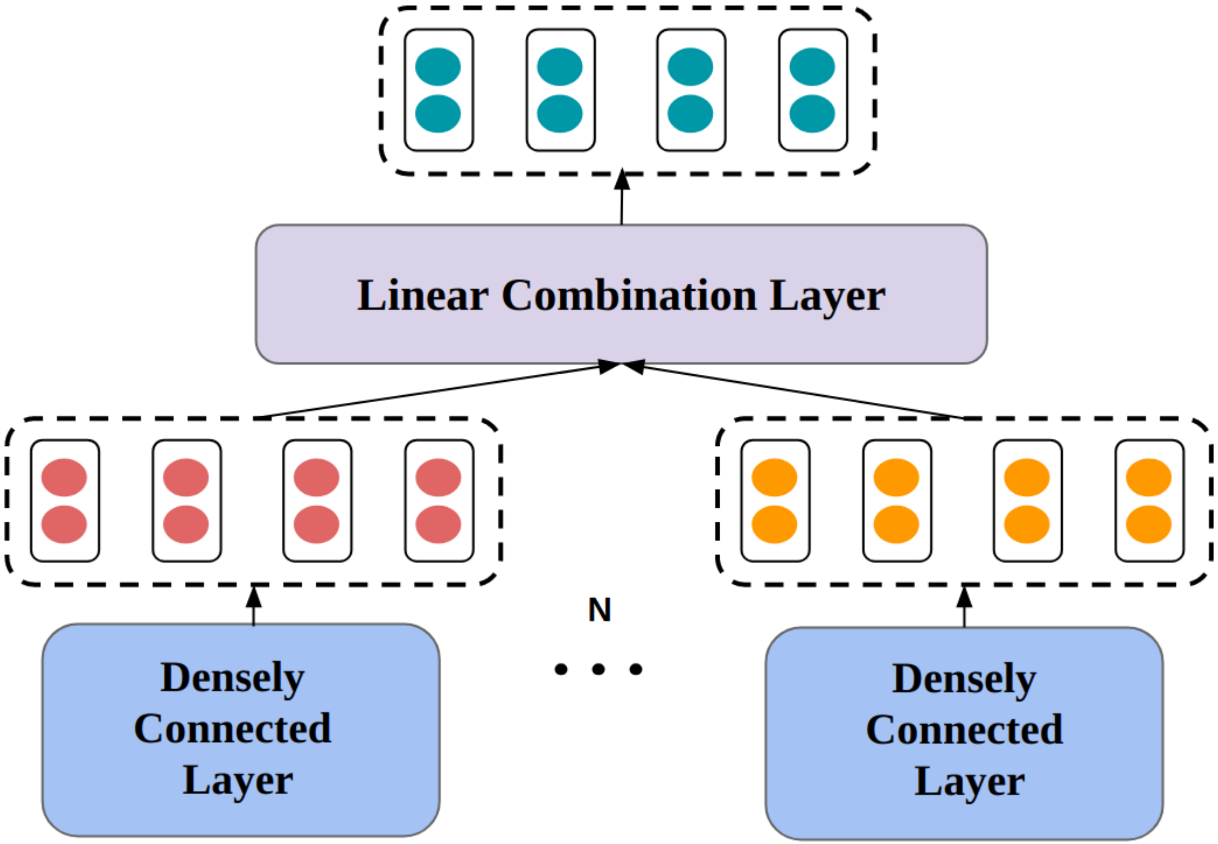
- Attention Guided Layer
- Densely Connected Layer
- **Linear Combination Layer**



Model

Linear Combination Layer

Integrate resulting representations from ***N*** densely connected layers



Experiments

Cross-sentence n-ary relation extraction

PubMed (Peng et al., 2017)

Sentence-level relation extraction

TACRED (Zhang et al., 2017)

SemEval-10 Task 8 (Hendrickx et al., 2010)

Experiments

Cross-sentence n-ary relation extraction

PubMed (Peng et al., 2017)

Sentence-level relation extraction

TACRED (Zhang et al., 2017)

SemEval-10 Task 8 (Hendrickx et al., 2010)

PubMed Settings

Types of Classification

Multi-Class

resistance or non-response, *sensitivity*, *response*, *resistance* and *none*

Binary-Class

binarize labels by grouping

4 relation as ***yes*** and treating none as ***no***

PubMed Settings

Number of Entities Per Relation

Ternary

3 entities are given for each relation

Binary

2 entities are given for each relation

PubMed: Binary-Class
Baselines

Structural Encoder + Full tree/Pruned tree

Model	Input	Tenary-Acc	Binary-Acc
Graph-LSTM	full tree	82.0	78.5
DAG-LSTM	full tree	77.3	76.4
GRNs	full tree	83.2	83.6
GCNs	full tree	84.8	83.6
GCNs	pruned tree (*K*=0)	85.8	82.7
GCNs	pruned tree (*K*=1)	85.7	83.4
GCNs	pruned tree (*K*=2)	85.0	83.7

PubMed: Binary-Class

Pruned tree: hard to find an optimal *K*

Model	Input	Tenary-Acc	Binary-Acc
Graph-LSTM	full tree	82.0	78.5
DAG-LSTM	full tree	77.3	76.4
GRNs	full tree	83.2	83.6
GCNs	full tree	84.8	83.6
GCNs	**pruned tree (*K*=0)**	**85.8**	82.7
GCNs	pruned tree (*K*=1)	85.7	83.4
GCNs	**pruned tree (*K*=2)**	85.0	**83.7**

PubMed: Binary-Class

AGGCNs learns how to automatically select information

Model	Input	Tenary-Acc	Binary-Acc
Graph-LSTM	full tree	82.0	78.5
DAG-LSTM	full tree	77.3	76.4
GRNs	full tree	83.2	83.6
GCNs	full tree	84.8	83.6
GCNs	pruned tree (*K*=0)	85.8	82.7
GCNs	pruned tree (*K*=1)	85.7	83.4
GCNs	pruned tree (*K*=2)	85.0	83.7
AGGCNs	**full tree**	**87.0**	**85.7**

PubMed: Multi-Class

Pruned Tree or Full Tree?

Model	**Input**	**Ternary-Acc**	**Binary-Acc**
DAG-LSTM	full tree	51.7	50.7
GRNs	full tree	71.7	71.7
GCNs	**full tree**	77.5	**74.3**
GCNs	pruned tree (*K*=0)	75.6	72.3
GCNs	**pruned tree (*K*=1)**	**78.1**	73.6
GCNs	pruned tree (*K*=2)	77.9	73.1

PubMed: Multi-Class

AGGCNs: learn how to select and discard information

Model	Input	Tenary-Acc	Binary-Acc
DAG-LSTM	full tree	51.7	50.7
GRNs	full tree	71.7	71.7
GCNs	full tree	77.5	74.3
GCNs	pruned tree (K=0)	75.6	72.3
GCNs	pruned tree (K=1)	78.1	73.6
GCNs	pruned tree (K=2)	77.9	73.1
AGGCNs	**full tree**	**79.7**	**77.4**

Experiments

Cross-sentence n-ary relation extraction

PubMed (Peng et al., 2017)

Sentence-level relation extraction

TACRED (Zhang et al., 2017)

SemEval-10 Task 8 (Hendrickx et al., 2010)

TACRED

Model	**Type**	**Prec**	**Rec**	**F1**
LR (Zhang et al., 2017)	Seq	**73.5**	49.9	59.4
PA-LSTM (Zhang et al., 2017)	Seq	65.7	64.5	65.1
SDP-LSTM (Xu et al., 2015)	Dep	66.3	52.7	58.7
Tree-LSTM (Tai et al., 2016)	Dep	66.0	59.2	62.4
C-GCNs (Zhang et al., 2018)	Dep	69.9	63.3	66.4
C-AGGCNs	Dep	72.3	**64.6**	**68.2**

SemEval

Model	Type	F1
SVM (Rink and Harabagiu, 2010)	Seq	82.2
PA-LSTM (Zhang et al., 2017)	Seq	82.7
SDP-LSTM (Xu et al., 2015)	Dep	83.7
SDPTree (Miwa et al., 2016)	Dep	84.4
C-GCNs (Zhang et al., 2018)	Dep	84.8
C-AGGCNs	Dep	**85.7**

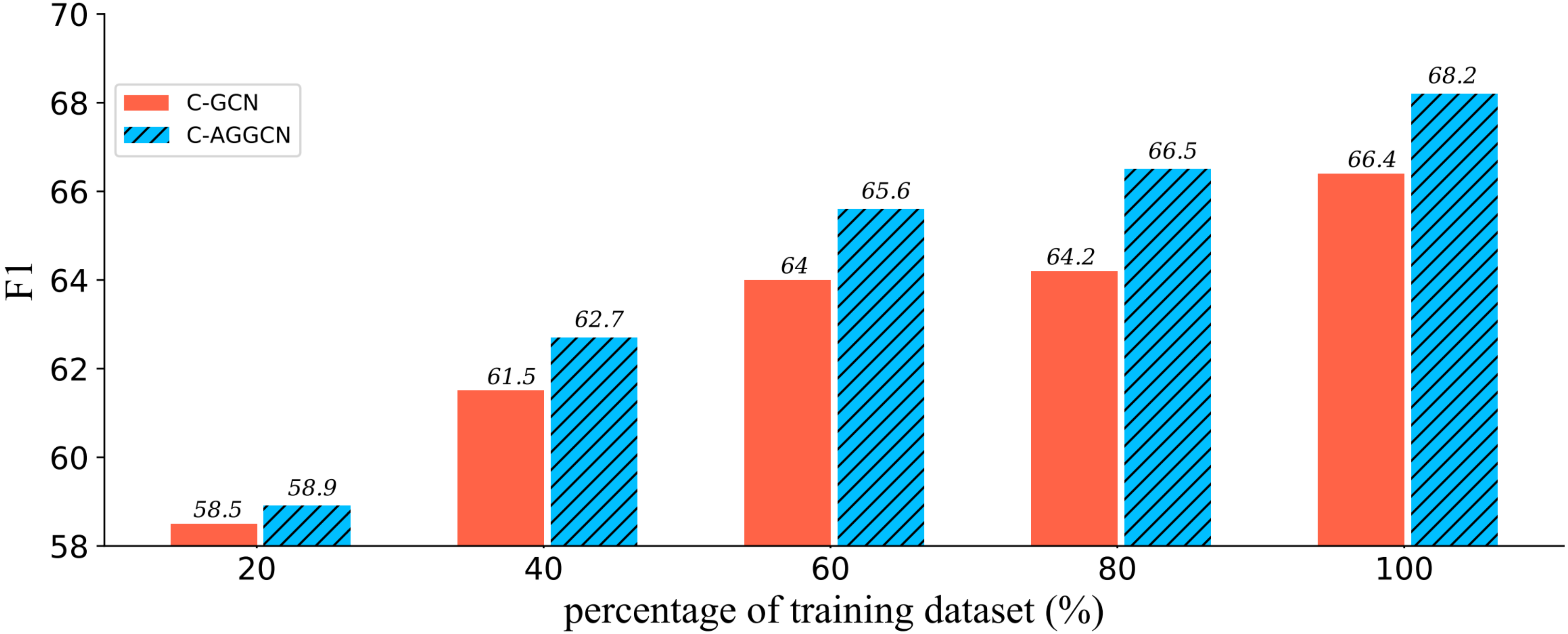
Ablation Test

Model	**F1**
C-AGGCNs	68.2
- Attention Guided Layer (AG)	66.9
- Densely Connected Layer (DC)	67.2
- AG, DC	66.7
- Feed Forward Network	67.8

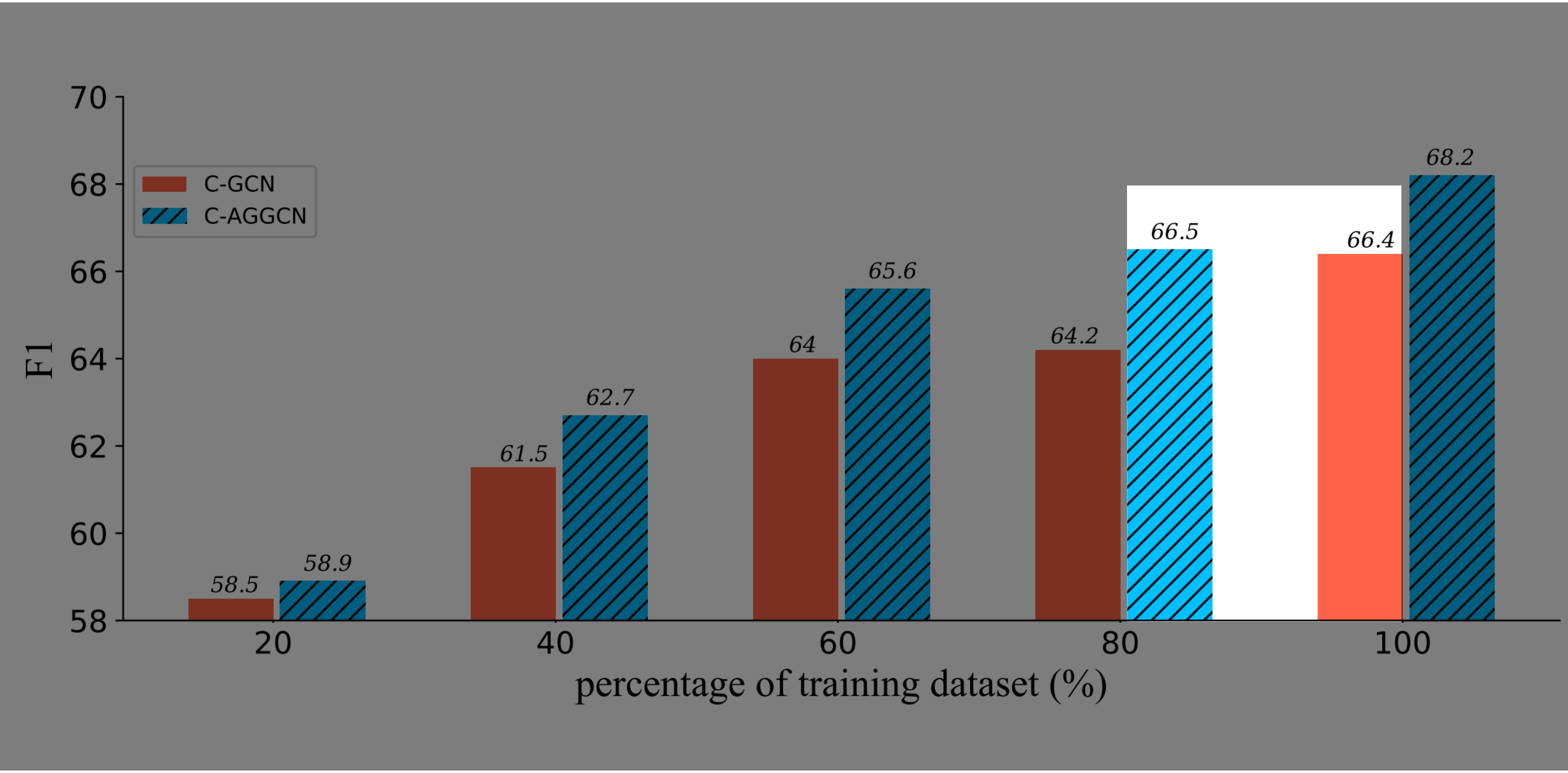
Ablation Test

Model	**F1**
C-AGGCNs	68.2
- Attention Guided Layer (AG)	66.9
- Densely Connected Layer (DC)	67.2
- AG, DC	66.7
- Feed Forward Network	67.8

Results vs Training Size



Results vs Training Size



Conclusion

Contribution

A novel GCN model that is able to learn a soft pruning strategy for better relation extraction.

Future Work

Explore the connections between the proposed model with other neural models for modelling global structural information.

Thank You

Code Available

`http://statnlp.org/research/ie/`

Performance against Sentence Length